

Untersuchungen zum Einsatz mehrfach überlappender lokaler trigonometrischer Basen in der Bilddatenkompression

Dissertation

zur
Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

der Fakultät für Informatik und Elektrotechnik
der Universität Rostock

vorgelegt von

Alexander Mali, geb. am 1. Mai 1975 in Schwerin
aus Hamburg

Rostock, 7. Dezember 2011

Als Dissertation genehmigt von der
Fakultät für Informatik und Elektrotechnik
der Universität Rostock

Gutachter:

1. Gutachter:
Prof. Dr.-Ing. habil. Erika Müller
Institut für Nachrichtentechnik, Universität Rostock
2. Gutachter:
Prof. Dr. rer. nat. habil. Gerlind Plonka-Hoch
Institut für Numerische und Angewandte Mathematik, Universität Göttingen
3. Gutachter:
Prof. Dr.-Ing. Alfred Mertins
Institut für Signalverarbeitung, Universität zu Lübeck

Tag der Einreichung: 7. Dezember 2011

Tag der Verteidigung: 4. Mai 2012

Kurzfassung

Die vorliegende Arbeit leistet einen Beitrag zur Entwicklung effizienter Kompressionsverfahren auf der Grundlage mehrfach überlappender lokaler trigonometrischer Basen.

Ausgehend von der Frage, wie Signale, die nur auf einem beschränkten Intervall $[0, L]$ definiert sind, durch überlappende lokale trigonometrische Basisfunktionen derart repräsentiert werden können, dass sich auch an den Signalrändern eine gute Signaldarstellung erzielen lässt, wird als Lösungsansatz die Anwendung von Basen auf dem Intervall vorgeschlagen. Es handelt sich hierbei um neuartige Systeme von trigonometrischen Basisfunktionen, die aus bestehenden Kosinus-IV-Basisfunktionen und aus neu entwickelten Kosinus-IV- und Kosinus-II-Randbasisfunktionen gebildet werden. Bei der Anwendung dieser Basisfunktionen in der Bilddatenkompression entstehen keine sichtbaren Randartefakte und es wird kein zusätzlicher Codieraufwand für die gute Darstellung der Bildränder benötigt.

Um die mehrfach überlappende Kosinustransformation von Bildern mit einem möglichst geringen Berechnungsaufwand realisieren zu können, werden schnelle Algorithmen eingeführt. Diese Algorithmen basieren auf schnellen trigonometrischen Blocktransformationen sowie auf effizienten Realisierungen von Klapp- und Aufklappoperatoren, die für die neu eingeführten Basen auf dem Intervall entwickelt wurden.

Schließlich werden verschiedene Strategien zur Codierung der Transformationskoeffizienten untersucht. Anhand üblicher Testbilder wird die Leistungsfähigkeit der vorgeschlagenen Kompressionsverfahren nachgewiesen. Insbesondere wird gezeigt, dass bereits das Ersetzen der Wavelet-Transformation in einem Wavelet-Codec durch die vierfach überlappende Kosinustransformation bei fast allen untersuchten Testbildern zu einer Erhöhung der Rekonstruktionsqualität führt.

Abstract

This thesis is a contribution to the development of efficient compression schemes that are based on multiple overlapping local trigonometric bases.

Starting with the question of how signals that are solely defined on the interval $[0, L]$ can be represented by multiple overlapping local trigonometric bases in such a manner that a good signal representation may be achieved at the boundaries of the signal, the use of bases for the interval is suggested as a solution. These are new systems consisting of trigonometric basis functions, which are constructed from existing cosine basis functions of type IV and newly developed cosine boundary basis functions of type IV and II. By using these basis functions in image compression, no visible boundary artifacts occur, while the good quality at the image boundaries is achieved without extra coding cost.

Fast algorithms are developed in order to realize the multiple overlapping cosine transformation of images with minimal computational complexity. These algorithms are based on fast trigonometric block transformations as well as on efficient realizations of folding and unfolding operators, which are developed for the newly established bases on the interval.

Finally, this thesis explores different strategies for the coding of the transformation coefficients. The efficiency of the proposed compression schemes is demonstrated on common test images. In particular, the results obtained reveal that the replacement of the wavelet transformation in a wavelet codec by the four-overlapping cosine transformation increases the reconstruction quality of almost all of the analyzed test images.

Inhaltsverzeichnis

Abkürzungen und Symbole	V
Einleitung	1
Thematischer Hintergrund	1
Motivation und Ziele	3
Gliederung	4
1 Grundlagen der Transformationscodierung	9
1.1 Signaltheoretische Grundlagen und Definitionen	10
1.1.1 Signale, Signalräume und lineare Operatoren	10
1.1.2 Darstellung von Signalen	13
1.1.3 Kriterien für Basisfunktionen	15
1.2 In der Bilddatenkompression häufig verwendete Transformationen	21
1.2.1 Kosinustransformation	21
1.2.2 Wavelet-Transformation	23
1.3 Skalare Quantisierung	26
1.4 Entropiecodierung	27
1.5 Kriterien zur Beurteilung der Leistungsfähigkeit von Kompressionsverfahren .	30
2 Lokale trigonometrische Basen und Klappoperatoren auf der reellen Achse	33
2.1 Lokale trigonometrische Basen	34
2.1.1 Definitionen und Grundlagen	34
2.1.2 Darstellung von Signalen mit überlappenden lokalen trigonometrischen Basisfunktionen	36
2.2 Klapp- und Aufklappoperatoren	37
2.2.1 Allgemeine Definitionen und Eigenschaften	38
2.2.2 Transformation von Signalen	41
2.2.3 Klappoperatoren für Kosinus-IV-Basen	42
2.2.4 Aufklappoperatoren für Kosinus-IV-Basen	47
2.3 Fenster zur Konstruktion von lokalen trigonometrischen Basisfunktionen . . .	53
2.3.1 Zweifach überlappende Fenster	53
2.3.2 Dreifach überlappende Fenster	55
2.3.3 $2m$ -fach überlappende Fenster	58
3 Lokale trigonometrische Basen und Klappoperatoren auf dem Intervall	61
3.1 Signalfortsetzung	62
3.2 Lokale trigonometrische Basen zur Darstellung von Signalen auf dem Intervall	63

3.3	Klapp- und Aufklappoperatoren zur Transformation von Signalen auf dem Intervall	64
3.3.1	Allgemeine Definitionen	65
3.3.2	Klappoperatoren für Kosinus-IV-Kosinus-II-Basen	67
3.3.3	Aufklappoperatoren für Kosinus-IV-Kosinus-II-Basen	69
3.4	Riesz-Stabilität	72
3.5	Fenster zur Konstruktion von trigonometrischen Randbasisfunktionen	75
3.5.1	Zweifachüberlappung	76
3.5.2	Dreifachüberlappung – Rekonstruktion von Konstanten	79
3.5.3	Dreifachüberlappung – Rekonstruktion von Linearen	81
3.5.4	Vierfachüberlappung – Ansatz 1	90
3.5.5	Vierfachüberlappung – Ansatz 2	92
3.5.6	Sechsfachüberlappung	95
4	Schnelle Transformationsalgorithmen	103
4.1	Diskretisierung	104
4.2	Schnelle Realisierung von Aufklappoperatoren	106
4.3	Schnelle Realisierung von Klappoperatoren	112
4.3.1	Algorithmen auf Grundlage der Standard-LU-Zerlegung	113
4.3.2	Algorithmen auf Grundlage einer Mini-LU-Zerlegung	116
4.4	Leistungsvergleich der Algorithmen	120
5	Lokale trigonometrische Basen in der Bilddatenkompression	123
5.1	Überlappende Kosinustransformation von Bilddaten	125
5.2	LCT-basierte SPIHT-Codierung	128
5.2.1	Suchbäume für blockweise angeordnete Transformationskoeffizienten	128
5.2.2	Kompressionsstrategie SPIHT-Codec-I	129
5.2.3	Kompressionsstrategie SPIHT-Codec-II	130
5.3	LCT-basierte L-CEB-Codierung	134
5.3.1	Kompressionsstrategie L-CEB-Codec-I	134
5.3.2	Kompressionsstrategie L-CEB-Codec-II	135
5.4	Erhöhung der Blockgröße und LCT-basierte AGU-Codierung	138
5.5	Reduktion von Kompressionsartefakten	141
5.6	Matlab-Tools zur Kompression von Bildern	144
6	Kompressionsergebnisse – Analyse und Auswertung	147
6.1	Kompressionsstrategien I und II im Vergleich	148
6.2	Basen auf der reellen Achse mit Signalfortsetzung im Vergleich zu Basen auf dem Intervall	148
6.3	Kosinustransformation	151
6.4	Zweifach überlappende Kosinustransformation	151
6.5	Mehrfach überlappende Kosinustransformation	152
6.6	Kosinus- und überlappende Kosinustransformation im Vergleich	155
6.7	Überlappende Kosinustransformation im Vergleich mit anderen blocküberlappenden Transformationen	158
6.8	Überlappende Kosinustransformation im Vergleich zur Wavelet-Transformation	162

7 Zusammenfassung und Ausblick	167
7.1 Zusammenfassung	167
7.2 Ausblick	169
Literaturverzeichnis	171
Anhang	183
A Klapp- und Aufklappoperatoren	183
A.1 Verarbeitung von Signalen, die auf der reellen Achse definiert sind	184
A.2 Verarbeitung von Signalen, die auf dem Intervall definiert sind	186
B Basisfunktionen	195
B.1 Basisfunktionen auf der reellen Achse	196
B.2 Basisfunktionen auf dem Intervall	200
C Amplitudengänge	207
D Duale Randfenster	211
E Algorithmen	217
E.1 Algorithmen für die Zweifachüberlappung	217
E.2 Algorithmen für die Dreifachüberlappung	225
E.3 Algorithmen für die Vierfachüberlappung	229
F Codierungsalgorithmen	239
F.1 Grundprinzip der SPIHT-Codierung	239
F.2 Grundprinzip der L-CEB-Codierung	243
F.3 Grundprinzip der AGU-Codierung	247
F.4 Filterparameter	251
G Weitere blocküberlappende Transformationen	253
G.1 Lapped Orthogonal Transform (LOT)	253
G.2 Generalized Linear-Phase Lapped Orthogonal Transform (Genlot)	255
G.3 Time-Domain Lapped Transform (TDLT)	259
G.4 Extended Lapped Transform (ELT)	261
H Testbilder und Kompressionsergebnisse	267
H.1 Testbilder, PSNR-Diagramme und PSNR-Tabellen	268
H.2 Quervergleiche zur objektiven Beurteilung der Leistungsfähigkeit	382
H.2.1 Kosinustransformation	382
H.2.2 Zweifach überlappende Kosinustransformation	385
H.2.3 Mehrfach überlappende Kosinustransformation	387
H.2.4 Kosinus- und überlappende Kosinustransformation im Vergleich	390
H.2.5 Vergleich mit anderen blocküberlappenden Transformationen	393
H.2.6 Vergleich mit waveletbasierten Kompressionsverfahren	395
H.3 Testergebnisse zur subjektiven Beurteilung der Rekonstruktionsqualität	399
H.3.1 Vergleich aller untersuchten Kompressionsstrategien am Beispiel des Testbildes „Barbara“ bei 0.1 bpp	399
H.3.2 Codec-I und Codec-II im Vergleich	476
H.3.3 Codec-II und JPEG2000 im Vergleich	482

H.3.4	Reduktion von Kompressionsartefakten	494
H.3.5	Basen auf dem Intervall und Signalfortsetzung im Vergleich	502

Thesen

Nomenklatur und Abkürzungen

Abkürzungen

bpp	bit per pixel
CEB	context-based entropy coding of block transform coefficients
CODEC	enCOder/DECoder
dB	Dezibel (logarithmische Maßeinheit)
DCT	Diskrete Kosinustransformation
DWT	Diskrete Wavelet-Transformation
EBCOT	embedded block coding with optimized truncation
ECECOW	embedded conditional entropy coding of wavelet coefficients
ELT	extended lapped transform
EZDCT	embedded zerotree DCT
EZW	embedded zerotree wavelet
FLWT	fast lifting wavelet transform
GenLOT	generalized lapped orthogonal transform
IEC	international electrotechnical commission
ISO	international organization for standardization
JPEG	joint photographic experts group
LBT	lapped biorthogonal transform
LCT	lapped cosine transform
LOT	lapped orthogonal transform
MOS	mean opinion score
MPEG	moving picture experts group
MSE	mean square error (Mittlerer quadratischer Fehler)
MSE_n	Mittlerer quadratischer Randfehler
n.d.	nicht definiert
PDA	personal digital assistant
pp	post-processing
PSNR	peak signal-to-noise ratio (Spitzen-Signal-Rausch-Verhältnis)
R_n -PSNR	Rand-Spitzen-Signal-Rausch-Verhältnis
SPIHT	set partitioning in hierarchical trees
TDLT	time domain lapped transform

Mathematische und signaltheoretische Notationen

$\mathbb{N}, \mathbb{N}_0, \mathbb{Z}$	natürliche Zahlen, $\mathbb{N} \cup \{0\}$, ganze Zahlen
\mathbb{C}, \mathbb{R}	komplexe Zahlen, reelle Zahlen
Λ_j, \mathcal{I}	Indexmenge

$[a, b]$	abgeschlossenes Intervall
(a, b)	offenes Intervall
$[a, b)$	halboffenes Intervall
S	Signalraum
H	Hilbertraum bzw. in der Codierungstheorie Entropie
L^2	Raum der quadratisch integrierbaren Funktionen
L^2_{loc}	Raum von Funktionen, die über jede kompakte Teilmenge von \mathbb{R} quadratisch integrierbar sind
ℓ^2	Raum der quadratisch summierbaren Folgen
$L^2(\mathbb{R})$	Raum der quadratisch integrierbaren Funktionen, die auf der gesamten reellen Achse definiert sind
$L^2([a, b])$	Raum der quadratisch integrierbaren Funktionen, die auf einem Intervall $[a, b]$ definiert sind
\prod_m	Menge der Polynome vom Grad kleiner oder gleich m
D	Definitionsbereich
$\{\dots\}$	Menge, Folge
$\{x : A\}$	Menge der x , die der Bedingung A genügen
$\overline{\{\dots\}}$	abgeschlossene Hülle einer Menge
$\langle \cdot, \cdot \rangle$	Skalarprodukt
$\lfloor x \rfloor$	größte ganze Zahl kleiner oder gleich x
$ x $	Betrag von x
$\ x\ $	Norm von x
\hat{x}	rekonstruierte Größe
\bar{x}	konjugiert komplexe Zahl
$f, f\tau$	Signal, geklapptes Signal
E_f	Signalenergie
$\text{supp } f$	Träger der Funktion f
χ	Rechteckfenster
r	rising cutoff function (Abschneidefunktion)
w, \tilde{w}	Fenster, duales Fenster
γ	ELT-Fensterparameter
θ	trigonometrische Funktion
$\psi, \tilde{\psi}$	Basisfunktion, duale Basisfunktion
$\delta_{i,j}$	Kronecker-Delta, $\delta_{i,j} = 0$ für $i \neq j$ und $\delta_{i,j} = 1$ für $i = j$
c	Koeffizienten des Signals bzgl. der Basis $\{\psi\}$
$\mathcal{T}_{\tilde{w}}, \mathcal{U}_w$	Klappoperator, Aufklappoperator
Z_λ	Zak-Transformation
ξ	Frequenzvariable
\mathbf{C}_θ	Trigonometrische Blocktransformation
\mathbf{A}	Matrix
\mathbf{A}^{-1}	inverse Matrix
\mathbf{A}^H	adjungierte Matrix
\mathbf{A}^T	transponierte Matrix
$\det \mathbf{A}$	Determinante der Matrix \mathbf{A}
$\sigma(\mathbf{A})$	Spektrum der Matrix \mathbf{A} (Menge aller Eigenwerte)
$\lambda_{\min}, \lambda_{\max}$	größter Eigenwert, kleinster Eigenwert
$\ \mathbf{A}\ _2$	Spektralnorm der Matrix \mathbf{A}
$\inf X$	größte untere Schranke (Infimum) einer Menge X
$\sup X$	kleinste obere Schranke (Supremum) einer Menge X

A, B	Riesz-Konstanten
A_0, B_0	beste untere Riesz-Schranke, beste obere Riesz-Schranke
κ	Konditionszahl
$X(z)$	Eingangssignal der Analysefilterbank im Bildbereich
$Y(z)$	Ausgangssignal der Analysefilterbank im Bildbereich
$\hat{X}(z)$	Ausgangssignal der Synthesefilterbank im Bildbereich
$\downarrow B$	Unterabtastung um den Faktor B
$\uparrow B$	Überabtastung um den Faktor B
$\mathbf{P}(z), \mathbf{Q}(z)$	Filterbankmatrizen
O	Anzahl arithmetischer Operationen
$\text{sgn}(c)$	Signumfunktion
Q	Quantisierungsparameter
e_q	Quantisierungsfehler
η	Thresholding Parameter
σ^2	Varianz
\bar{g}	Mittlerer Grauwert
g_{\min}	Minimaler Grauwert
g_{\max}	Maximaler Grauwert
\mathbb{S}	Symbolalphabet
s_i	Symbol
p_i	Auftretenswahrscheinlichkeit
$I(s_i)$	Informationsgehalt von s_i

Notationen und Abkürzungen zur SPIHT-Codierung

c	Koeffizient
$(i, j), (e, f)$	Koordinate eines Koeffizienten, Knotens bzw. Elements
\mathcal{I}	Koordinatenmenge
\mathcal{H}	Menge der Koordinaten der Baumwurzeln
\mathcal{O}	Menge der direkten Nachkommen
\mathcal{D}	Menge aller Nachkommen
\mathcal{L}	Menge aller Nachkommen ohne die direkten Nachkommen
S	Signifikanzfunktion
k, k_0	Bitschicht, höchstwertige Bitschicht
LIP	list of insignificant pixels
LIS	list of insignificant sets
LSP	list of significant pixels
A, B	Elementtypen

Notationen und Abkürzungen zur L-CEB-Codierung

c	Koeffizient
(x, y)	Koordinate eines Koeffizientenblocks
(i, j)	Koordinate eines Koeffizienten innerhalb eines Koeffizientenblocks
l	Zig-Zag-Scan Parameter
S	Signifikanzfunktion

\mathcal{D}	Koeffizientenmenge
DC	direct current subband
PV	principal vertical subband
PH	principal horizontal subband
LV	low-frequency vertical subband
LH	low-frequency horizontal subband
LD	low-frequency diagonal subband
HP	high-pass subband

Notationen und Abkürzungen zur AGU-Codierung

(x, y)	Koordinate eines Koeffizientenblocks
(i, j)	Koordinate eines Koeffizienten innerhalb eines Koeffizientenblocks
k, k_0	Bitschicht, höchstwertige Bitschicht
C	Entscheidungsregel
PM	Wahrscheinlichkeitsmodell
P	Bitwert

Einleitung

Thematischer Hintergrund

Durch die fortschreitende Entwicklung und Verbreitung digitaler Medien stieg laut einer Studie der International Data Corporation (IDC) die Menge der weltweit gespeicherten und übertragenen Daten im Jahr 2009 um ca. 62 Prozent an [GR10]. Bis zum Jahr 2020 wird ein weltweit vorgehaltenes Datenvolumen prognostiziert, das den Wert aus dem Jahr 2009 um den Faktor 44 übersteigt. Trotz der schnellen Entwicklung größerer Datenspeicher und des ständigen Ausbaus von Übertragungsbandbreiten lässt sich diese Datenflut nur bewältigen, wenn es u.a. gelingt, die Leistungsfähigkeit von Verfahren zur Datenkompression weiter zu verbessern. Da ein hoher Anteil des Datenvolumens durch die Erfassung und Digitalisierung visueller Informationen erzeugt wird, gewinnt insbesondere die Kompression von Bild- und Videodaten immer mehr an Bedeutung.

Bei der Entwicklung neuer Kompressionsverfahren steht neben der Verringerung der Datenmenge bei gleichzeitigem Erhalt der Rekonstruktionsqualität auch die Ressourceneffizienz der Kompressionsalgorithmen im Fokus aktueller Forschungsaktivitäten. Immer mehr Bild- und Videodaten werden durch mobile Systeme erfasst, auf mobilen Systemen verarbeitet und gespeichert sowie zwischen mobilen Systemen übertragen. Da solche Systeme, wie beispielsweise Digicams, Fotohandys, Smartphones, PDAs und Netbooks, im Allgemeinen nur über geringe Ressourcen in Bezug auf Rechenleistung, Arbeitsspeicher und Akku-Kapazität verfügen, sind ressourceneffiziente Algorithmen zur Kompression von Bild- und Videodaten erforderlich. Die vorliegende Arbeit liefert einen Beitrag zur Entwicklung von effizienten und leistungsfähigen Verfahren zur Kompression von Bilddaten.

In der Bilddatenkompression unterscheidet man grundsätzlich zwischen verlustlosen und verlustbehafteten Kompressionsverfahren. Die verlustlose Kompression kommt dort zum Einsatz, wo die rekonstruierten Bilder exakt mit dem Original übereinstimmen sollen. Mit ihr sind im Allgemeinen jedoch nur relativ geringe Kompressionsverhältnisse¹ möglich. Um die Datenmenge weiter zu reduzieren, werden daher bei der verlustbehafteten Kompression Informationen, die für den Empfänger irrelevant oder weniger wichtig sind, entfernt. Da diese Informationen unwiederbringlich verloren sind und somit für die Bildrekonstruktion nicht zur Verfügung stehen, stimmen die rekonstruierten Bilder nur näherungsweise mit dem Original überein. Bei der Entwicklung verlustbehafteter Kompressionsverfahren stellt sich die Frage, welche Informationen wichtig sind und nicht verloren gehen dürfen. In der Bilddatenkompression konzentriert sich dieses Problem in der Regel auf eine von Kompressionsartefakten ungestörte Wahrnehmung der wesentlichen Bildinhalte. Als ein wichtiges Hilfsmittel zur Lösung dieser Aufgabe haben sich Transformationen erwiesen. Sie sollen die zu komprimierenden Bilder in eine Repräsentation überführen, die sich zur Trennung von relevanten und irrelevanten Informationen

¹Das Kompressionsverhältnis ergibt sich aus der Relation der Datenmenge des ursprünglichen Bildes zur Datenmenge des komprimierten Bildes [Str09].

eignet. Die relevanten Bildinformationen sind hierbei auf möglichst wenige Darstellungselemente zu konzentrieren.

Im Allgemeinen lässt sich eine Vielzahl von Bildinhalten als Überlagerung von zeitlich bzw. örtlich lokalisierten Schwingungen auffassen, wobei die wesentlichen Werte einer Zeit-Frequenz-Darstellung auf eine relativ kleine Teilmenge der Zeit-Frequenz-Ebene konzentriert sind. Eine solche dünne Repräsentation von Bildinhalten mit zeit-frequenz-lokalen Funktionen, die im Folgenden als Basisfunktionen bezeichnet werden, ermöglicht eine effiziente und für die verlustbehaftete Datenkompression geeignete Darstellung von Bildern. Die derzeit am häufigsten verwendeten Kompressionsverfahren basieren entweder auf der Kosinustransformation oder der Wavelet-Transformation. In den letzten Jahren wurden durch die Verwendung von waveletbasierten Transformationscodern (u.a. SPIHT, ECECOW, EBCOT/JPEG2000 [SP96b, Xia97, Tau00, JPE00, TOWS02]) beachtliche Kompressionsergebnisse erzielt. Die Basisfunktionen der Wavelet-Transformation werden durch Skalierung und Verschiebung eines zeit-frequenz-lokalen Mutter-Wavelets gebildet [SN96, Dau92]. Man erhält so Basisfunktionen mit guter Zeit-Frequenz-Lokalität, wobei sich mit wachsender Frequenz die Zeitlokalität auf Kosten der Frequenzlokalität verbessert. Besonders glatte Bildbereiche mit vereinzelt auftretenden Sprüngen lassen sich durch Wavelets effizient darstellen. Für eine sparsame Repräsentation von höherfrequenten Bildinhalten mit wiederkehrenden Mustern wie z.B. Texturen sind jedoch harmonische Funktionen wie trigonometrische Basisfunktionen besser geeignet [Bit00b].

Als ein konventioneller Zugang zur zeit-frequenz-lokalen Bildzerlegung in trigonometrische Basisfunktionen gilt die Kosinustransformation [ANR74, RY90]. Sie gehört zur Klasse der sogenannten Blocktransformationen und ist Bestandteil des JPEG-Standards zur Bilddatenkompression [Wal91, Wal92, PM93] sowie verschiedener Standards zur Videokompression (z.B. MPEG-1/-2/-4) [RH96, MPG93, MPG96, MPG04]. Die Basisfunktionen der Kosinustransformation bestehen aus Kosinusfunktionen, die mit einem Rechteckfenster gewichtet werden. Durch die Rechteckfensterung ergeben sich wesentliche Nachteile der so konstruierten Basisfunktionen. Die Unstetigkeit der Basisfunktionen an den Blockgrenzen bewirkt eine schlechte Lokalität im Frequenzbereich. Das führt zu einem langsameren Abklingen der Transformationskoeffizienten und einem geringeren Codiergewinn in der Bilddatenkompression. Bei der verlustbehafteten Datenkompression können im rekonstruierten Bild Sprünge an den Blockgrenzen entstehen. Diese Kompressionsartefakte werden bei höheren Kompressionsverhältnissen als sogenannte Blockeffekte bzw. Schachbrettmuster sichtbar. Da bei der Kosinustransformation nur die Rechteckfensterung zum Einsatz kommt, ist zudem auch keine Anpassung der Transformation an verschiedene Signaleigenschaften möglich.

Einen Ansatz zur Lösung der vorgenannten Probleme stellt die Nutzung von überlappenden lokalen trigonometrischen Basen dar [CM91, DJJ91, AWW92, Mal92, AACI93, Wic93, AACI94, JLS94, JS95, XS95, Mat96, CS97a, BV98, Bit98b, Bit00b, Mey02, LWY06]. Es handelt sich hierbei um Systeme von Basisfunktionen, die aus Produkten von Kosinus- und/oder Sinusfunktionen mit gut lokalisierten, überlappenden Fensterfunktionen gebildet werden [Bit00b]. Man erhält so Basisfunktionen mit guter Zeit-Frequenz-Lokalität und durch die geeignete Wahl von Fensterfunktionen lässt sich eine Anpassung der Basen an Signaleigenschaften erzielen. Die Anwendung der bisher untersuchten zweifach überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression brachte zwar Erfolge bei der Reduktion von Blockartefakten [Mal90, AACI93, AACI94, JLS94, Bit00b, Mal00, MM00, MPM01a, MPM01b], doch blieben die Blockgrenzen bei höheren Kompressionsverhältnissen als schnelle Grauwertwechsel sichtbar. Die Entwicklung von geeigneteren überlappenden lokalen trigonometrischen Basen war daher Bestandteil weiterer Untersuchungen [Bit02a, BC03, BC04].

Motivation und Ziele

Mit den Resultaten in [Bit02a, BC03, BC04] stehen neue Fensterfunktionen zur Konstruktion von überlappenden lokalen trigonometrischen Basen zur Verfügung, die sich nicht mehr auf die Zweifachüberlappung beschränken. Da die Einführung der Mehrfachüberlappung zu mehr Freiheitsgraden für den Entwurf von Fensterfunktionen führt, konnten aufgrund der so gewonnenen Flexibilität Basisfunktionen entwickelt werden, die in der verlustbehafteten Bilddatenkompression eine höhere Codiereffizienz und eine bessere objektive sowie subjektive Qualität der rekonstruierten Bilder erwarten lassen [Bit02a, BC03, BC04]. Die neuen Basen erfüllen u.a. Forderungen, die aus den Ergebnissen in [MM00, Mal00] zur Untersuchung von zweifach überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression abgeleitet wurden. Hierzu zählen insbesondere:

- längere Träger der Basisfunktionen, um schnelle Grauwertwechsel in der Umgebung von Blockgrenzen zu reduzieren,
- Verbesserung der Riesz-Stabilität, damit Störungen im Bild wie z.B. Rauschen und Veränderungen von Transformationskoeffizienten wie z.B. Quantisierungsfehler nicht durch die Transformation bzw. die Rücktransformation verstärkt werden,
- Reproduktion von Polynomen höherer Ordnung ausschließlich mit Basisfunktionen der niedrigsten Frequenz, um lokal glatte Bildbereiche effizienter approximieren zu können und somit eine dünnere Darstellung von Bildinhalten durch eine geringere Anzahl signifikanter Basisfunktionen zu ermöglichen.

Untersuchungen zum Einsatz dieser Basisfunktionen in der Bilddatenkompression existieren nach dem derzeitigen Kenntnisstand nicht. Die vorliegende Arbeit setzt sich daher zum Ziel, mehrfach überlappende lokale trigonometrische Basisfunktionen¹ im Hinblick auf die Anwendung in der Bilddatenkompression zu untersuchen und einen Beitrag zur Entwicklung effizienter Kompressionsverfahren auf Grundlage dieser Basisfunktionen zu leisten. Aus dem bisherigen Entwicklungsstand ergeben sich folgende Problemstellungen:

Verarbeitung von endlich langen Signalen

Sowohl die bisher bekannten zweifach überlappenden trigonometrischen Basen als auch die in [Bit02a, BC04] neu vorgestellten mehrfach überlappenden trigonometrischen Basen beziehen sich auf die gesamte reelle Achse \mathbb{R} . In der Bilddatenkompression werden jedoch Bilder verarbeitet, die eine endliche Bildgröße aufweisen. Bei dimensionsweiser Verarbeitung (vgl. Abschnitt 1.1.1 dieser Arbeit) wird man also mit Signalen konfrontiert, die nur auf einem beschränkten Intervall $[0, L]$ definiert sind. Die Transformation solcher Signale mit Basisfunktionen, die auf ganz \mathbb{R} definiert sind, erfordert im Allgemeinen Signalwerte, die außerhalb des beschränkten Intervalls $[0, L]$ liegen und nicht definiert sind. Es wird daher ein geeignetes Verfahren zur Verarbeitung von Signalen, die nur auf einem beschränkten Intervall $[0, L]$ definiert sind, benötigt. Bei der Auswahl bzw. Entwicklung eines solchen Verfahrens ist der Behandlung der Signالرänder besondere Aufmerksamkeit zu widmen. Trotz der durch das Abschneiden des Signals entstandenen Unstetigkeiten soll auch an den Signالرändern eine gute Signaldarstellung erzielt werden. In der verlustbehafteten Bilddatenkompression können bei

¹Um den Rahmen dieser Arbeit nicht zu sprengen, konzentrieren sich die Untersuchungen auf Systeme, die unter Verwendung von Kosinus-Basisfunktionen [Bit02a, BC04] gebildet werden. Die in [BC03] vorgestellten Basen, die aus einer Mischung von Kosinus- und Sinus-Basisfunktionen bestehen, finden hier keine Berücksichtigung.

der Verwendung von ungeeigneten Verarbeitungsstrategien sichtbare Randartefakte entstehen. Solche Randartefakte sind zu vermeiden und die gute Darstellung der Bildränder soll nicht mit einer Erhöhung des Speicheraufwandes verbunden sein.

Schnelle Transformationsalgorithmen

Die direkte Realisierung von Transformationen mit überlappenden trigonometrischen Basen ist mit einem hohen rechnerischen Aufwand verbunden. Dies gilt insbesondere bei Verwendung der in [Bit02a, BC04] vorgestellten mehrfach überlappenden trigonometrischen Basisfunktionen zur Signalanalyse, da sie im Gegensatz zu den Basisfunktionen zur Signalrekonstruktion keinen beschränkten Träger aufweisen. Für den praktischen Einsatz von transformationsbasierten Kompressionsverfahren in mobilen Systemen, die nur über vergleichsweise geringe Ressourcen verfügen, werden jedoch Transformationsalgorithmen mit geringer Berechnungskomplexität benötigt. Es sind daher im Rahmen dieser Arbeit Algorithmen zur Realisierung der Transformation mit überlappenden lokalen trigonometrischen Basen einzuführen, die einen möglichst geringen Berechnungsaufwand verursachen.

Kompressionsverfahren

Bei der verlustbehafteten Transformationscodierung haben sowohl die Darstellung wesentlicher Bildinhalte durch Überlagerung von wenigen signifikanten Basisfunktionen als auch die effiziente Codierung der quantisierten Wichtungskoeffizienten entscheidenden Einfluss auf die Leistungsfähigkeit des Kompressionsverfahrens. Vorangegangene Untersuchungen zur Anwendung von überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression waren u.a. dadurch motiviert, Blockartefakte in bestehenden DCT-basierten Kompressionsverfahren wie dem JPEG-Codec zu reduzieren [AACI93, AACI94, Bit00b, Mal00]. Dementsprechend wurden die Transformationskoeffizienten nach dem JPEG-Verfahren weiterverarbeitet. Dieser Ansatz eignet sich zwar für einen Vergleich der Effizienz verschiedener trigonometrischer Basisfunktionen, man erhält jedoch gegenüber aktuellen waveletbasierten Kompressionsverfahren keinen leistungsfähigen Transformationscodec. Die Codierungsstrategien, die im JPEG-Codec zur Anwendung kommen, arbeiten deutlich ineffizienter als die kontextbasierten adaptiven Entropiecodierverfahren aktueller waveletbasierter Transformationscodecs [TT02a]. Daher sind im Rahmen dieser Arbeit effizientere Verfahren zur Codierung der Transformationskoeffizienten zu untersuchen, die neben dem Vergleich verschiedener überlappender Kosinustransformationen auch eine Einschätzung der Leistungsfähigkeit gegenüber waveletbasierten State-of-the-Art Kompressionsverfahren zulassen.

Gliederung

Entsprechend der oben erläuterten Problem- und Zielstellungen ist die Arbeit wie folgt gegliedert:

Das erste Kapitel dient der Einführung wesentlicher Grundlagen der Transformationscodierung, auf die in den weiteren Kapiteln dieser Arbeit Bezug genommen wird. Der Schwerpunkt liegt hierbei auf der Darstellung von Signalen mit Basisfunktionen. Es werden wichtige Eigenschaften von Basen wie Stabilität, Trägerlänge, Glattheit und verschwindende Momente vorgestellt und die Bedeutung dieser Eigenschaften für die Bilddatenkompression wird erläutert. Des Weiteren gibt dieses Kapitel eine kurze Einführung zur Kosinus- und Wavelet-Transformation sowie in grundlegende Verfahren zur Quantisierung und Entropiecodierung

von Transformationskoeffizienten. Schließlich werden Kriterien zur Beurteilung der Leistungsfähigkeit von Kompressionsverfahren angegeben.

Das zweite Kapitel befasst sich mit überlappenden lokalen trigonometrischen Basen, die auf der reellen Achse \mathbb{R} definiert sind. Es werden die Fensterfunktionen zur Konstruktion der im Rahmen dieser Arbeit untersuchten lokalen trigonometrischen Basisfunktionen vorgestellt und die Basen im Hinblick auf die Anwendung in der Bilddatenkompression anhand der in Kapitel 1 erläuterten Eigenschaften bewertet. Da wichtige Resultate der vorliegenden Arbeit durch die Verwendung von Klapp- bzw. Aufklappoperatoren erzielt werden, besteht ein weiterer Schwerpunkt des Kapitels darin, wesentliche Eigenschaften dieser Operatoren herauszuarbeiten und die Anwendung dieser Operatoren in der Signalverarbeitung zu erläutern. Ausgehend von den allgemeinen Definitionen in [Bit00b] erfolgt die Herleitung von Klapp- und Aufklappoperatoren für mehrfach überlappende Kosinusbasen. Die hierbei gewonnenen Resultate bilden eine Grundlage für die Entwicklung neuer Klapp- und Aufklappoperatoren in Kapitel 3.

Im dritten Kapitel werden zur effizienten Darstellung von Signalen, die nur auf einem beschränkten Intervall $[0, L]$ definiert sind, überlappende lokale trigonometrische Basen auf dem Intervall eingeführt. Hierbei handelt es sich um neuartige Systeme von Basisfunktionen, die aus bestehenden Basisfunktionen (vgl. Abschnitt 2.3 dieser Arbeit) und aus neu entwickelten Randbasisfunktionen (vgl. Abschnitt 3.5 dieser Arbeit) gebildet werden. Zur Konstruktion von geeigneten Randbasisfunktionen erfolgt die Einführung neuer Randfenster. Die vorgeschlagenen Basen auf dem Intervall erfüllen für die Anwendung in der Bilddatenkompression wichtige Eigenschaften. Es wird u.a. gezeigt, dass die Darstellung des Gleichanteils durch Basisfunktionen der niedrigsten Frequenz auch an den Signlrändern gewährleistet ist. Des Weiteren werden in diesem Kapitel Klapp- und Aufklappoperatoren für Basen auf dem Intervall definiert und unter Verwendung der Resultate aus Kapitel 2 für das neuartige System von trigonometrischen Basisfunktionen hergeleitet. Diese Operatoren ermöglichen es, die Stabilität der neu eingeführten lokalen trigonometrischen Basen auf dem Intervall nachzuweisen. Zudem bilden diese Operatoren die Grundlage für die Entwicklung schneller Transformationsalgorithmen in Kapitel 4.

Im vierten Kapitel werden schnelle Algorithmen eingeführt, um überlappende trigonometrische Transformationen mit einem möglichst geringen Berechnungsaufwand zu realisieren. Da in der Bilddatenkompression die zu verarbeitenden Bilder in diskreter Form vorliegen, erfolgt die Diskretisierung der Transformation von Signalen mit überlappenden trigonometrischen Basisfunktionen. Hierbei wird gezeigt, dass die Transformation eines diskreten Signals mit überlappenden trigonometrischen Basisfunktionen zu einer Darstellung führt, die einer diskreten trigonometrischen Blocktransformation des diskreten geklappten Signals entspricht. Somit ermöglichen die in Kapitel 3 neu eingeführten Klapp- und Aufklappoperatoren für Basen auf dem Intervall die Anwendung schneller trigonometrischer Blocktransformationen. In den weiteren Abschnitten des Kapitels werden effiziente Algorithmen zur Realisierung der Klapp- und Aufklappoperatoren eingeführt.

Die Kapitel 5 und 6 dieser Arbeit setzen sich mit der Kompression von Bilddaten auseinander. Im fünften Kapitel wird zunächst die überlappende trigonometrische Transformation von Bildern auf Grundlage der in Kapitel 3 eingeführten Klappoperatoren bzw. der in Kapitel 4 vorgestellten Algorithmen erläutert. Es schließen sich Untersuchungen verschiedener Ansätze zur Quantisierung und Codierung der Transformationskoeffizienten an, wobei im Ge-

gensatz zu früheren Untersuchungen [Bit00b, Mal00] effizientere Verfahren zur Entropiecodierung Anwendung finden. Um die Leistungsfähigkeit der vorgeschlagenen Kompressionsstrategien beurteilen zu können, werden verschiedene Testbilder komprimiert. Die Untersuchungen konzentrieren sich insbesondere auf

- Vergleiche zwischen DCT, zweifach und mehrfach überlappenden trigonometrischen Basen, um Aussagen über die Reduktion von Blockartefakten zu erhalten,
- Vergleiche mit waveletbasierten Kompressionsverfahren, um die Leistungsfähigkeit mehrfach überlappender trigonometrischer Basisfunktionen gegenüber Wavelets bewerten zu können.

Des Weiteren werden zwei Matlab-Programme zur Kompression von Bildern vorgestellt, die zur Demonstration der Leistungsfähigkeit von überlappenden lokalen trigonometrischen Basen entwickelt wurden. Sie umfassen die im Rahmen dieser Arbeit untersuchten Transformationen, Codierungsstrategien und Testbilder. Im sechsten Kapitel erfolgt die Auswertung der Kompressionsergebnisse in Bezug auf die Zielstellung dieser Arbeit. Es wird gezeigt, dass bei Verwendung der vorgeschlagenen Kompressionsstrategien-II, die mehrfach überlappende lokale trigonometrische Basen auf dem Intervall nutzen, weder Block- noch Randartefakte in den rekonstruierten Bildern auftreten. Besonders hervorzuheben ist das Resultat, dass bereits das Ersetzen der als sehr leistungsfähig geltenden 9/7-Wavelet-Transformation in einem Wavelet-Codec durch die Transformation mit vierfach überlappenden lokalen trigonometrischen Basen auf dem Intervall bei 102 der insgesamt 114 untersuchten Testbild-Bitraten-Kombinationen zu einer Verbesserung der objektiven Rekonstruktionsqualität führt.

Abschließend werden im siebten Kapitel die wesentlichen Erkenntnisse der vorliegenden Arbeit zusammengefasst.

Der Anhang dieser Arbeit enthält eine ausführliche Dokumentation relevanter Kompressionsergebnisse, die zur Beurteilung und Verifikation der Leistungsfähigkeit der vorgeschlagenen Verfahren erstellt wurde.

Es ist zu betonen, dass die Resultate der Kapitel 3 und 4 von grundlegender Bedeutung für die Entwicklung von leistungsfähigen Kompressionsverfahren sind, die mehrfach überlappende lokale trigonometrische Basen nutzen (vgl. Abbildung 1). Die in Kapitel 3 neu eingeführten Basen auf dem Intervall ermöglichen eine effiziente Darstellung von endlich langen Signalen auch an den Signalrändern, so dass in der verlustbehafteten Bilddatenkompression keine Randartefakte auftreten und kein zusätzlicher Codieraufwand entsteht. Die in Kapitel 4 vorgestellten Algorithmen erlauben die mehrfach überlappende trigonometrische Transformation von Bilddaten mit einem geringen Berechnungsaufwand. Die Entwicklung dieser Algorithmen wurde erst durch die Herleitung von bisher nicht bekannten Klapp- und Aufklappoperatoren auf dem Intervall in Kapitel 3 möglich.

Die Einführung von Basen auf dem Intervall und von schnellen Transformationsalgorithmen ist wesentliche Voraussetzung dafür, dass in Kombination mit effizienten Entropiecodierverfahren leistungsfähige Kompressionsstrategien, die mehrfach überlappende trigonometrische Basisfunktionen nutzen, vorgeschlagen werden können.

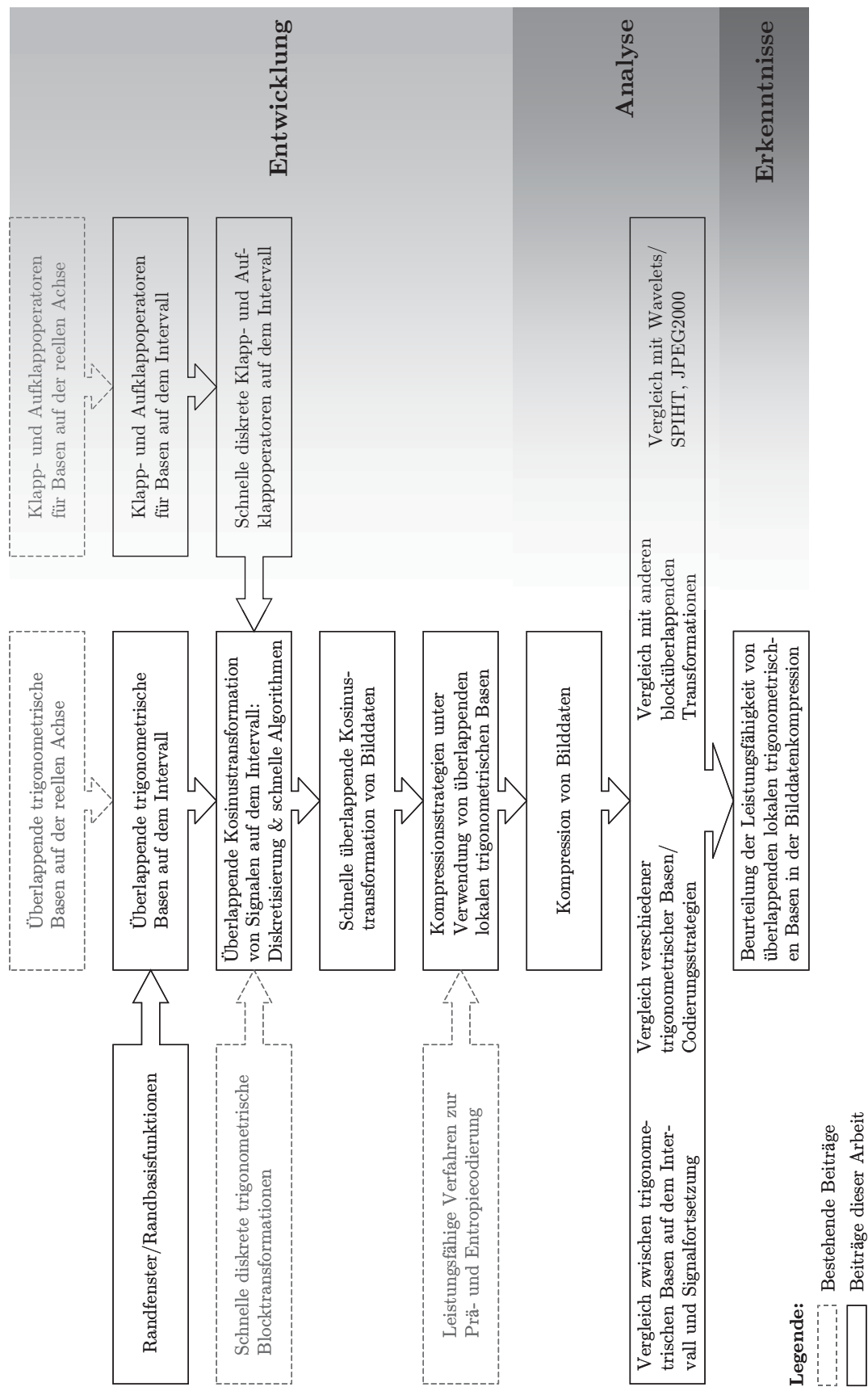


Abbildung 1: Untersuchungen zum Einsatz mehrfach überlappender lokaler trigonometrischer Basen in der Bilddatenkompression.

Kapitel 1

Grundlagen der Transformationscodierung

In diesem Kapitel werden wesentliche signaltheoretische Grundlagen der Transformationscodierung vorgestellt. Der Schwerpunkt liegt hierbei auf der Darstellung von Signalen mit Basisfunktionen. Insbesondere werden Eigenschaften von Basen erläutert, die für die Bilddatenkompression von Bedeutung sind. Des Weiteren gibt dieses Kapitel eine kurze Einführung in grundlegende Verfahren zur Quantisierung und Entropiecodierung von Transformationskoeffizienten.

Der Begriff Transformationscodierung steht für Kompressionsverfahren, die Redundanz unter Einsatz von Signaltransformationen verringern. Die Redundanz gilt als Maß für den Anteil des Datenaufkommens, der überflüssig ist und nicht zur Repräsentation der Information benötigt wird. In der Bilddatenkompression unterscheidet man im Wesentlichen zwischen Codierungsredundanz, Intersymbolredundanz und psychovisuelle Redundanz [GW93, Str09]. Die alleinige Reduktion von Codierungs- und Intersymbolredundanz führt zu verlustfreien Kompressionsverfahren, wobei die zu erzielenden Kompressionsverhältnisse in der Regel nur bei ca. 2:1 bis 4:1 liegen und somit relativ gering ausfallen. Wesentlich höhere Kompressionsverhältnisse sind mit verlustbehafteten Kompressionsverfahren möglich. Hierbei werden Informationen, die für den Empfänger nicht bzw. kaum wahrnehmbar oder nicht von Interesse sind, entfernt. Diese irrelevanten Anteile fallen unter die Bezeichnung psychovisuelle Redundanz.

Abbildung 1.1 zeigt die allgemeine Struktur eines Transformationscodecs zur verlustbehafteten Kompression von Bilddaten. Bei der Transformationscodierung wird das Bildsignal zunächst durch eine Transformation in eine Darstellung überführt, die Vorteile für die weiteren Verarbeitungsschritte bringt. Neben der Signaldekorrelation soll sich diese Repräsentation auch möglichst gut zur Separation von relevanten und irrelevanten Informationen eignen. Die irrelevanten Informationen werden in der Quantisierungsstufe verworfen und können nicht zurückgewonnen werden. In der sich anschließenden Codierstufe wird mit Hilfe der Präcodierung die Intersymbolredundanz reduziert und die Codierungsredundanz mittels Entropiecodierung verringert.

Entsprechend der vorgenannten Verarbeitungskette ist dieses Kapitel wie folgt gegliedert: Abschnitt 1.1 dient der Einführung wichtiger signaltheoretischer Grundlagen und Definitionen. Wesentliche Eigenschaften von Basen, die für die Bilddatenkompression von Bedeutung sind, werden erläutert und anhand von Beispielen veranschaulicht. In Abschnitt 1.2 werden mit der Kosinustransformation und der Wavelet-Transformation die beiden derzeit in der Bilddatenkompression am häufigsten verwendeten Transformationen kurz vorgestellt. Die Abschnitte 1.3

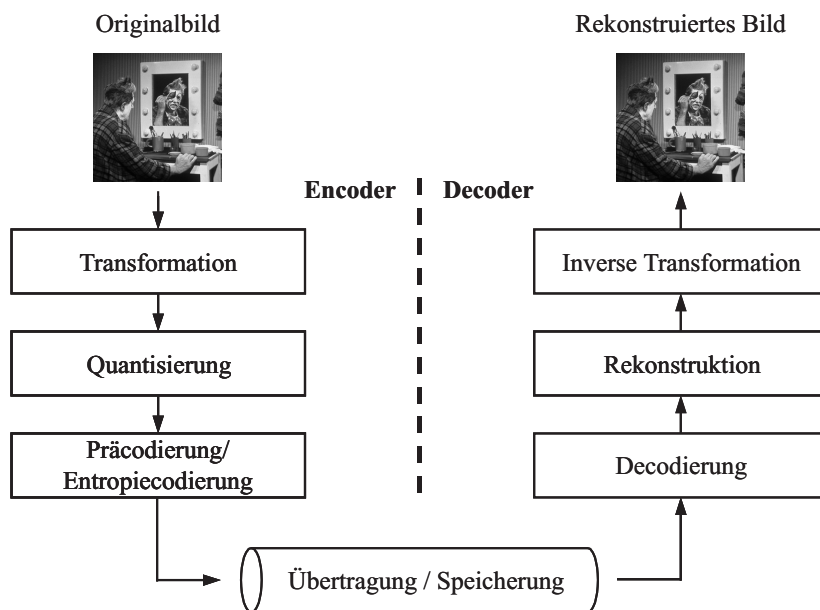


Abbildung 1.1: Allgemeine Struktur eines Transformationscodecs zur verlustbehafteten Kompression und zur Rekonstruktion von Bilddaten.

und 1.4 befassen sich mit Verfahren zur Quantisierung und Codierung von Transformationskoeffizienten. Schließlich führt Abschnitt 1.5 Kriterien zur Beurteilung der Leistungsfähigkeit von Kompressionsverfahren an.

1.1 Signaltheoretische Grundlagen und Definitionen

1.1.1 Signale, Signalräume und lineare Operatoren

In der Systemtheorie werden Signale als Träger von Informationen definiert und mathematisch durch reell- oder komplexwertige Funktionen einer oder mehrerer unabhängiger Variablen modelliert. Zur Vereinfachung der mathematischen Darstellung werden nachfolgend ohne Einschränkung der Allgemeinheit eindimensionale Signale als Funktionen $f(t)$ einer reellen Veränderlichen und eindimensionale Abbildungen bzw. Transformationen betrachtet. Für die Verarbeitung von mehrdimensionalen Signalen sei die Transformation solcher Signale im Folgenden als Tensorprodukt eindimensionaler Transformationen definiert (vgl. z.B. [Mal98, Ten11]). Die Verarbeitung von zweidimensionalen Bildsignalen erfolgt wie in der klassischen Bilddatenkompression üblich durch zeilen- und spaltenweise nacheinander ausgeführte eindimensionale Transformationen.

Definitionsbereich von Signalen

In Abhängigkeit vom Definitionsbereich der unabhängigen Variablen t eines Signals $f(t)$ unterscheidet man zwischen Signalen auf der reellen Achse und Signalen auf dem Intervall¹. Bei der Verarbeitung eines Signals, das nur auf einem beschränkten Intervall definiert ist,

¹In der Literatur wird im Allgemeinen das Intervall $[0, 1]$ betrachtet, das stellvertretend für irgendein endliches Intervall steht [LMR98].

sind im Allgemeinen gewisse Randbetrachtungen erforderlich. Daher wird in den grundlegenden signaltheoretischen Untersuchungen zur Transformation von Signalen zunächst nur der Fall betrachtet, bei dem Signale und Basisfunktionen auf der reellen Achse definiert sind. Für die Anwendung von lokalen trigonometrischen Basen in der Bilddatenkompression erfolgt anschließend der Übergang zu Signalen und Basisfunktionen auf dem Intervall.

Kontinuierliche und diskrete Signale

Die Bilddatenkompression ist ein Teilgebiet der digitalen Signalverarbeitung. Hierbei werden Signale verarbeitet, die sowohl zeit- als auch wertdiskret sind (digitale Signale). Liefert eine Signalquelle nur zeit- und wertkontinuierliche Signale (analoge Signale), so sind diese in einem ersten Verarbeitungsschritt zu digitalisieren. Der Übergang von einem zeitkontinuierlichen zu einem zeitdiskreten Signal $f[k]$ erfolgt durch Abtastung des zeitkontinuierlichen Signals $f(t)$ an äquidistanten Zeitpunkten $t_k = k \cdot \Delta t$, $k \in \mathbb{Z}$, mit der Abtastweite Δt . Um das kontinuierliche Signal verlustfrei aus einer Folge von Abtastwerten rekonstruieren zu können, ist das Nyquist-Shannon'sche Abtasttheorem¹ einzuhalten [Sha49, Kot33]. Dieses besagt, dass die zeitlichen Abtastwerte $f(k \cdot \Delta t)$ ein zeitkontinuierliches und bandbegrenztes Signal $f(t)$ vollständig bestimmen, sofern die Abtastfrequenz $F_s = 1/\Delta t$ mindestens doppelt so groß ist wie die Bandbreite des Signals.

Um in der Digitaltechnik mit einer endlichen Wortlänge zu arbeiten, werden auch die Signalwerte diskretisiert. Der Übergang vom wertkontinuierlichen zum wertdiskreten Signal heißt Quantisierung. Hierbei unterteilt man den Wertebereich des Signals in eine endliche Anzahl von Teilintervallen. Alle Signalwerte f innerhalb eines Intervalles $f_n < f \leq f_{n+1}$ werden durch einen diskreten Signalwert f_Q ersetzt, der dieses Intervall repräsentiert, z.B. $f_Q = \frac{1}{2}(f_n + f_{n+1})$. Die Differenz zwischen f und f_Q bezeichnet man als Quantisierungsfehler.

Zur Vereinfachung der mathematischen Beschreibung wird im Rahmen dieser Arbeit in den grundlegenden theoretischen Untersuchungen zu überlappenden lokalen trigonometrischen Basen sowie Klapp- und Aufklappoperatoren zunächst von kontinuierlichen Signalen und Basisfunktionen ausgegangen. Für die Anwendung in der Bilddatenkompression erfolgt anschließend der Übergang zur diskreten Darstellung.

Signalraum über den reellen Zahlen

Für die theoretischen Untersuchungen in der Signal- und Bildverarbeitung ist es von Vorteil, Signale als Elemente von Signal- bzw. Funktionenräumen² aufzufassen [Neu03]. Ein Signalraum besteht aus Signalen, die addiert oder mit Skalaren $\lambda \in \mathbb{R}$ multipliziert werden können:

- Überlagerung \rightarrow Addition von Signalen: $f + g$,
- Verstärkung oder Dämpfung \rightarrow Multiplikation mit Skalaren: $\lambda \cdot f$,

wobei das Ergebnis wieder ein Signal ist. Die Menge der Signale bildet zusammen mit der Definition der vorgenannten elementaren Signaloperationen einen linearen Raum S über den reellen Zahlen, wobei die folgenden Vektorraumaxiome gelten [Neu03, Heu06]:

1. Kommutativität der Addition: $f + g = g + f$, für alle $f, g \in S$,
2. Assoziativität der Addition: $(f + g) + h = f + (g + h)$, für alle $f, g, h \in S$,

¹Das Abtasttheorem ist häufig auch unter der Bezeichnung WKS-Abtasttheorem (WKS: Whittaker Kotelnikow Shannon) zu finden [Wer09].

²Ein Signal- bzw. Funktionenraum ist ein Vektorraum, dessen Vektoren Signale bzw. Funktionen sind.

3. Existenz des Nullelements: $\exists \mathbf{0} \in S$ mit $f + \mathbf{0} = f$, für alle $f \in S$,
4. Existenz des negativen Elements: $\exists -f \in S$ mit $f + (-f) = \mathbf{0}$, für alle $f \in S$,
5. Assoziativität der Skalarmultiplikation: $(\lambda \cdot \mu) \cdot f = \lambda \cdot (\mu \cdot f)$, für alle $f \in S, \lambda, \mu \in \mathbb{R}$,
6. Existenz des Einselements: $\exists \mathbf{1} \in \mathbb{R}$ mit $\mathbf{1} \cdot f = f$, für alle $f \in S$,
7. Distributivität I: $(\lambda + \mu) \cdot f = \lambda \cdot f + \mu \cdot f$, für alle $f \in S$ und $\lambda, \mu \in \mathbb{R}$,
8. Distributivität II: $\lambda \cdot (f + g) = \lambda \cdot f + \lambda \cdot g$, für alle $f, g \in S$ und $\lambda \in \mathbb{R}$.

Durch die Zuordnung von Norm und Skalarprodukt werden Signalräumen zusätzliche Strukturen aufgeprägt.

Norm

Wird ein Vektorraum S mit einer Norm versehen, so bezeichnet man ihn als normierten Vektorraum. Als Norm wird eine Abbildung $\|\cdot\| : S \rightarrow \mathbb{R}_{\geq 0}$ definiert, die jedem Element f des Vektorraumes S eine reelle nichtnegative Zahl zuordnet und die folgenden Axiome erfüllt [Heu06]:

1. positive Definitheit: $\|f\| \geq 0$ und $\|f\| = 0 \Leftrightarrow f = 0$,
2. Homogenität: $\|\lambda \cdot f\| = |\lambda| \cdot \|f\|$, $\lambda \in \mathbb{R}$,
3. Dreiecksungleichung: $\|f + g\| \leq \|f\| + \|g\|$.

Funktionsräume können mit verschiedenen Normen versehen werden. In der Signalverarbeitung ist die L^2 - bzw. ℓ^2 -Norm die gebräuchlichste Norm, die im Quadrat der Signalenergie

$$\|f\|_{L^2}^2 = E_f = \int_{-\infty}^{\infty} |f(t)|^2 dt, \quad (1.1)$$

$$\|f\|_{\ell^2}^2 = E_f = \sum_{n=-\infty}^{\infty} |f[n]|^2 \quad (1.2)$$

entspricht [KSW08, Mer10]. Der Raum von Funktionen bzw. Folgen mit endlicher Norm wird als $L^2(\mathbb{R})$ -Raum der quadratisch integrierbaren Funktionen bzw. als $\ell^2(\mathbb{Z})$ -Raum der quadratisch summierbaren Folgen bezeichnet. Jede Norm induziert durch $d(f, g) = \|f - g\|$ ein Abstandsmaß, auch Metrik genannt [Heu06]. Somit ist jeder normierte Raum auch ein metrischer Raum. In der Bilddatenkompression wird der euklidische Abstand dazu verwendet, das rekonstruierte Signal mit dem Originalsignal zu vergleichen.

Skalarprodukt

Eine Abbildung $\langle \cdot, \cdot \rangle : S \times S \rightarrow \mathbb{R}$, die zwei Elementen f und g eines Vektorraumes S ein Skalar zuordnet, heißt Skalarprodukt oder inneres Produkt, wenn sie die folgenden Axiome erfüllt [FSV03]:

1. Linearität: $\langle f + g, h \rangle = \langle f, h \rangle + \langle g, h \rangle$ und $\langle \lambda \cdot f, g \rangle = \lambda \cdot \langle f, g \rangle$, $\lambda \in \mathbb{R}$,
2. Symmetrie: $\langle f, g \rangle = \langle g, f \rangle$,
3. positive Definitheit: $\langle f, f \rangle \geq 0$ und $\langle f, f \rangle = 0 \Leftrightarrow f = 0$.

In der Signalverarbeitung wird das Skalarprodukt zwischen zwei reellwertigen kontinuierlichen Signalen $f(t)$ und $g(t)$ als

$$\langle f, g \rangle := \int_{-\infty}^{\infty} f(t) \cdot g(t) dt \quad (1.3)$$

bzw. für diskrete Signale als

$$\langle f, g \rangle := \sum_{n=-\infty}^{\infty} f[n] \cdot g[n] \quad (1.4)$$

definiert. Mit Hilfe des Skalarprodukts lässt sich die durch das Skalarprodukt induzierte L^2 -Norm eines Signals als

$$\|f\|_{L^2} := \sqrt{\langle f, f \rangle} = \sqrt{E_f} \quad (1.5)$$

definieren. Ein Vektorraum mit Skalarprodukt ist aufgrund der induzierten Norm ein normierter Raum und demzufolge auch ein metrischer Raum.

Hilbert-Raum H

Unter dem Begriff Hilbert-Raum werden die mit einem Skalarprodukt versehenen und bzgl. der induzierten Norm vollständigen Vektorräume zusammengefasst [WS05]. Ein normierter Vektorraum heißt vollständig, wenn in ihm jede Cauchy-Folge konvergiert [Hac08]. Der Funktionenraum L^2 und der Folgenraum ℓ^2 sind Hilbert-Räume.

Linearer Operator

In der Signalverarbeitung wird das mathematische Modell, das die Beziehung zwischen Eingangs- und Ausgangssignalen beschreibt, als System bezeichnet. Von besonderem Interesse sind Systeme, die Eingangssignale linear auf Ausgangssignale abbilden. Für zwei Signale S und S' wird eine Abbildung $\mathcal{T}: S \rightarrow S'$ als lineare Abbildung bzw. synonym als linearer Operator bezeichnet, wenn für alle Signale $f, g \in S$ und alle Skalare $\lambda \in \mathbb{R}$ die Eigenschaften der

$$\begin{aligned} \text{Additivität:} \quad \mathcal{T}(f + g) &= \mathcal{T}f + \mathcal{T}g, \\ \text{Homogenität:} \quad \mathcal{T}(\lambda \cdot f) &= \lambda \cdot \mathcal{T}f \end{aligned} \quad (1.6)$$

erfüllt sind. In einem homogenen System bewirkt die Änderung der Amplitude eines Eingangssignals um den Faktor λ eine Änderung des Ausgangssignals um den gleichen Faktor. Das System heißt additiv, wenn addierte Signale im System nicht interagieren.

1.1.2 Darstellung von Signalen

Viele Probleme in der Signal- und Bildverarbeitung lassen sich vereinfachen, indem man die zu verarbeitenden Signale f durch Überlagerung von geeigneten Elementen ψ_n beschreibt:

$$f = \sum_{n \in \mathcal{I}} c_n \cdot \psi_n, \quad c_n \in \mathbb{R}. \quad (1.7)$$

Die Skalare c_n heißen Koordinaten, Koeffizienten oder Wichtungskoeffizienten des Signals f bezüglich der Folge $\{\psi_n : n \in \mathcal{I}\}$ von Elementen. In der Signalverarbeitung bezeichnet man die Bestimmung der Wichtungskoeffizienten als Signalanalyse oder Signalzerlegung und die Rekonstruktion nach Gleichung (1.7) als Signalsynthese bzw. Signalrekonstruktion.

Um für die Bilddatenkompression geeignete Signaldarstellungen zu erhalten, muss das System $\{\psi_n : n \in \mathcal{I}\}$ gewisse grundlegende Eigenschaften wie Vollständigkeit, Eindeutigkeit und numerische Stabilität erfüllen. Zur Beschreibung dieser Eigenschaften wurden Begriffe wie Orthonormalbasis und Riesz-Basis [Bar46, Bar51] eingeführt.

Orthonormalbasen

Ein System $\{\psi_n : n \in \mathcal{I}\}$, $\psi_n \in H$, wird als Orthonormalbasis bezeichnet, wenn es die folgenden Bedingungen erfüllt:

- Orthogonalität: aus $n, m \in \mathcal{I}$ und $n \neq m$ folgt $\langle \psi_n, \psi_m \rangle = 0$,
- Normierung: für alle $n \in \mathcal{I}$ gilt $\|\psi_n\| = 1$,
- Vollständigkeit: aus $f \in H$ und $\langle f, \psi_n \rangle = 0$ für alle $n \in \mathcal{I}$ folgt $f = 0$.

Berücksichtigt man nur die erste Bedingung, so erhält man ein orthogonales System. Es heißt orthonormiert, wenn es zusätzlich die zweite Bedingung erfüllt. Ist das Orthonormalsystem $\{\psi_n\}$ vollständig und somit eine orthonormale Basis von H , so gilt für jedes Signal $f \in H$ die Basisentwicklung [RS80]

$$f = \sum_{n \in \mathcal{I}} \langle \psi_n, f \rangle \psi_n = \sum_{n \in \mathcal{I}} c_n \psi_n \quad (1.8)$$

und die Parsevalsche Gleichung [RS80]

$$\|f\|^2 = \sum_{n \in \mathcal{I}} |\langle \psi_n, f \rangle|^2 = \sum_{n \in \mathcal{I}} |c_n|^2. \quad (1.9)$$

Die Koeffizienten $c_n = \langle \psi_n, f \rangle$ sind die Darstellungskoeffizienten von f bezüglich der Orthonormalbasis $\{\psi_n\}$. Nach Gleichung (1.9) ist die Transformation mit orthonormalen Basen normerhaltend. Die Signalenergie ändert sich bei der Entwicklung in orthonormale Basisfunktionen nicht und kann sowohl im Orts- bzw. Zeitbereich als auch im Frequenzbereich bestimmt werden.

Riesz-Basen

Die Orthonormalität ist eine sehr restriktive Bedingung, welche die Flexibilität beim Entwurf von Basisfunktionen stark einschränkt. Man erhält jedoch Signaldarstellungen, die eindeutig bestimmt, stabil und redundanzfrei sind. Um allgemeinere Systeme von Basisfunktionen zu entwickeln, die ebenfalls zur Darstellung von Signalen dienen und vorteilhafte Eigenschaften für die Anwendung in der Bilddatenkompression aufweisen können, verzichtet man häufig auf Orthonormalität und ersetzt sie durch die schwächere Bedingung der Riesz-Stabilität.

Ein System $\{\psi_n, n \in \mathcal{I}\}$, $\psi_n \in H$, heißt Riesz-Basis von H , wenn es vollständig ist und zwei Konstanten A und B mit $0 < A \leq B < \infty$ existieren, so dass für alle Koeffizienten c_n die Verallgemeinerung der Parsevalschen Gleichung

$$A \sum_{n \in \mathcal{I}} |c_n|^2 \leq \left\| \sum_{n \in \mathcal{I}} c_n \psi_n \right\|^2 \leq B \sum_{n \in \mathcal{I}} |c_n|^2 \quad (1.10)$$

erfüllt ist [Bit00b]. Die Konstanten A und B werden als Riesz-Konstanten bezeichnet. Zu jeder Riesz-Basis $\{\psi_n, n \in \mathcal{I}\}$ existiert ein duales System $\{\tilde{\psi}_n, n \in \mathcal{I}\}$, das ebenfalls eine Riesz-Basis ist und die Bedingung

$$\langle \psi_n, \tilde{\psi}_m \rangle = \delta_{nm}, \quad \text{für alle } m, n \in \mathcal{I} \quad (1.11)$$

erfüllt. Das bedeutet, dass die beiden Basen zueinander biorthogonal sind. Sie werden daher auch als biorthogonale Basen bezeichnet. Die Basis $\{\tilde{\psi}_n\}$ heißt duale Basis und dient in der Signal- und Bildverarbeitung der Signalanalyse. Mit dem Skalarprodukt $c_n = \langle f, \tilde{\psi}_n \rangle$ werden die Transformationskoeffizienten zur Darstellung eines Signals bezüglich der Basis $\{\psi_n\}$

bestimmt. Für die Signalbeschreibung gilt als Verallgemeinerung von Gleichung (1.8) für biorthogonale Basen die Entwicklungsgleichung

$$f = \sum_{n \in \mathcal{I}} \langle f, \tilde{\psi}_n \rangle \psi_n = \sum_{n \in \mathcal{I}} c_n \psi_n. \quad (1.12)$$

Stimmt die Riesz-Basis mit ihrer dualen Basis überein, so ist die Riesz-Basis eine Orthonormalbasis. Jede Orthonormalbasis ist eine Riesz-Basis mit $A = B = 1$.

1.1.3 Kriterien für Basisfunktionen

Wie der Begriff Wavelets umfasst auch der Begriff überlappende lokale trigonometrische Basen eine ganze Familie von Basen mit zeit-frequenz-lokalen Basisfunktionen, die sehr unterschiedliche Eigenschaften aufweisen können. Um in der Bilddatenkompression gute Ergebnisse zu erzielen, sollten die Basisfunktionen gewisse grundlegende Kriterien erfüllen. Da viele Eigenschaften der Basisfunktionen voneinander abhängen und die Verbesserung einer Eigenschaft oftmals zur Verschlechterung anderer Eigenschaften führt, sind bei der Entwicklung von leistungsfähigen Basisfunktionen für die Signal- und Bildverarbeitung geeignete Kompromisse zu finden. In diesem Abschnitt werden wichtige Eigenschaften von Basisfunktionen mit Bezug auf die Bilddatenkompression erläutert.

Stabilität, Kondition

Riesz-Schranken und Konditionszahlen stellen in der Bilddatenkompression wichtige Hilfsmittel zur Einschätzung der Leistungsfähigkeit von Transformationen dar [SN96, Mal00]. Die eingesetzten Transformationen sollen numerisch stabil sein. Kleine Störungen im Bild wie z.B. Rauschen sollen durch die Transformation nicht verstärkt werden, kleine Änderungen von Transformationskoeffizienten wie z.B. Quantisierungsfehler dürfen nicht zu großen Änderungen im rekonstruierten Bild führen.

Um eine stabile Darstellung von Signalen zu erhalten, müssen grundsätzlich Konstanten A und B mit $0 < A \leq B < \infty$ existieren, für welche die Ungleichung in (1.10) erfüllt ist. Von besonderem Interesse sind die größte untere Riesz-Schranke A_0 (Infimum) und die kleinste obere Riesz-Schranke B_0 (Supremum), mit denen die vorgenannte Bedingung erfüllt wird, d.h.

$$0 < A \leq A_0 \leq B_0 \leq B < \infty. \quad (1.13)$$

Die Schranken¹ A_0 und B_0 lassen sich über die Norm des linearen Operators $\mathcal{T}: H \rightarrow H$, der die Orthonormalbasis auf die Riesz-Basis abbildet, bestimmen und werden als $B_0 := \|\mathcal{T}\|^2$ bzw. $A_0 := \|\mathcal{T}^{-1}\|^{-2}$ definiert, wobei [Wer07]

$$\|\mathcal{T}\| := \sup_{\|f\| \neq 0} \frac{\|\mathcal{T}f\|}{\|f\|} = \sup_{\|f\|=1} \|\mathcal{T}f\|, \quad f \in H. \quad (1.14)$$

Ist die lineare Abbildung zwischen endlich dimensionalen normierten Signalräumen durch eine reellwertige Transformationsmatrix \mathbf{T} gegeben, so erhält man die Riesz-Konstanten aus dem Spektrum der Matrix $\mathbf{T}^T \mathbf{T}$. Das Spektrum $\sigma(\mathbf{T}^T \mathbf{T})$ umfasst die Menge aller Eigenwerte λ_i von $\mathbf{T}^T \mathbf{T}$ und die Riesz-Schranken B_0 bzw. A_0 entsprechen dem größten bzw. dem kleinsten Eigenwert:

$$B_0 := \|\mathbf{T}\|_2^2 = \lambda_{\max} = \max \sigma(\mathbf{T}^T \mathbf{T}), \quad (1.15)$$

¹Die Konstanten A_0 und B_0 werden in der Literatur auch als beste Riesz-Konstanten, bestmögliche Riesz-Konstanten oder exakte Riesz-Konstanten bezeichnet [Bit00b, BC04, Pri06].

$$A_0 := \|\mathbf{T}^{-1}\|_2^{-2} = \lambda_{\min} = \min \sigma(\mathbf{T}^T \mathbf{T}). \quad (1.16)$$

Um abzuschätzen, wie sich ein Fehler der Eingabedaten während der Transformation fort-pflanzt, wird die Konditionszahl

$$\kappa = \frac{\sqrt{B_0}}{\sqrt{A_0}} \quad (1.17)$$

definiert. Sie stellt den Faktor dar, um den sich ein relativer Fehler durch die Transformation im ungünstigsten Fall verstärken kann. Für $\kappa \approx 1$ ist die Transformation gut konditioniert. Der relative Fehler wird durch die Transformation nicht oder nur geringfügig verstärkt. Kritisch sind hingegen größere Konditionszahlen $\kappa \gg 1$, da die Transformation zu einer erheblichen Fehlerverstärkung führen kann.

In der verlustbehafteten Bilddatenkompression hat das Verhalten des Quantisierungsfehlers während der Rücktransformation einen wesentlichen Einfluss auf die Leistungsfähigkeit des Kompressionsverfahrens. Beim Quantisierungsfehler handelt es sich um Abweichungen zwischen den rekonstruierten Transformationskoeffizienten \hat{c} und den originalen Transformationskoeffizienten c , die durch die Quantisierung entstehen. Als Fehlermaß wird der mittlere quadratische Fehler (MSE, mean square error)

$$\text{MSE}(x, \hat{x}) := \frac{1}{N} \cdot \sum_{i=1}^N (x_i - \hat{x}_i)^2 \quad (1.18)$$

definiert. Bei der Verwendung von Orthonormalbasen ändert sich der durch die Quantisierung hervorgerufene mittlere quadratische Fehler $\text{MSE}(c, \hat{c})$ während der Signalrekonstruktion nicht und stimmt mit dem $\text{MSE}(f, \hat{f})$ des rekonstruierten Signals überein. Erfüllen die verwendeten Basen die Bedingung der Orthonormalität nicht, so besteht die Möglichkeit, dass sich der MSE während der Signalrekonstruktion verstärkt.

In Abbildung 1.2 ist anhand eines Kompressionsbeispiels [Mal00] das Verhalten des mittleren quadratischen Fehlers während der Rücktransformation für die DCT und für eine beispielhaft

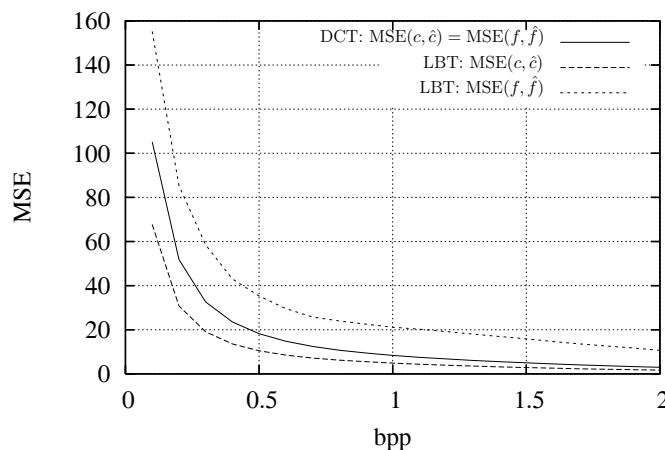


Abbildung 1.2: Verhalten des mittleren quadratischen Fehlers (MSE) bei der Bildrekonstruktion aus quantisierten Transformationskoeffizienten für die orthonormale DCT-Basis und für eine biorthogonale trigonometrische Basis (LBT); Kompressionsbeispiel aus [Mal00].

gewählte LBT¹ dargestellt. Bei einer vorgegebenen Bitrate weisen die rekonstruierten Transformationskoeffizienten der LBT einen geringeren MSE als die Transformationskoeffizienten der DCT auf. Es ist jedoch zu erkennen, dass sich der MSE bei der Signalrekonstruktion mit den biorthogonalen Basisfunktionen der LBT deutlich erhöht, während er bei der Signalrekonstruktion mit den orthonormalen Basisfunktionen der DCT unverändert bleibt. Die Fehlerverstärkung ist bei diesem Kompressionsbeispiel so stark ausgeprägt, dass schließlich mit der gewählten LBT eine schlechtere objektive Rekonstruktionsqualität des Bildes erzielt wird als mit der DCT. Die Eigenschaft der Fehlerverstärkung ist ein wesentlicher Nachteil, den die Einführung der Biorthogonalität mit sich bringen kann. Für die Bilddatenkompression sind daher Basen von Interesse, die eine möglichst gute Kondition aufweisen.

Träger von Basisfunktionen

Unter dem Träger $\text{supp } \psi$ einer Basisfunktion $\psi \in L^2(\mathbb{R})$ versteht man die abgeschlossene Hülle der Menge aller Punkte, an denen die Basisfunktion verschieden von Null ist [For09]:

$$\text{supp } \psi := \overline{\{t \in \mathbb{R} \mid \psi(t) \neq 0\}}. \quad (1.19)$$

Der Träger von Basisfunktionen bzw. dualen Basisfunktionen beschreibt somit den Bereich, in dem die Basisfunktionen zur Rekonstruktion eines Signals bzw. die dualen Basisfunktionen zur Bestimmung der Transformationskoeffizienten beitragen. Eine Basisfunktion besitzt einen kompakten Träger, falls ein endliches abgeschlossenes Intervall existiert, außerhalb dessen die Basisfunktion verschwindet. Für die Entwicklung von schnellen Transformationsalgorithmen sind Basisfunktionen mit kurzem Träger von Vorteil.

Im engen Zusammenhang mit der Trägerlänge steht die zeitliche bzw. örtliche Lokalität der Basisfunktionen. Um in der verlustbehafteten Bilddatenkompression stark lokalisierte Bild-details effizient darzustellen und die Ausdehnung von sichtbaren Rekonstruktionsfehlern zu begrenzen, sind Basisfunktionen mit hinreichend guter Ortslokalität bzw. kurzer Trägerlänge erforderlich. Eine schlechte Lokalität im Ortsbereich macht sich insbesondere an Kanten bemerkbar. Bei hohen Kompressionsfaktoren können in der Umgebung von Kanten deutlich sichtbare Kompressionsartefakte auftreten, die als Ringing bezeichnet werden (vgl. Abbildung 1.3).

Glattheit

Der Begriff Glattheit ist in der verlustbehafteten Bilddatenkompression an die von Kompressionsartefakten ungestörte Wahrnehmung der wesentlichen Bildinhalte gekoppelt [OB96, DN98, Bra99]. Das rekonstruierte Bild soll nicht nur eine möglichst gute Approximation des ursprünglichen Bildes darstellen, sondern vorhandene Rekonstruktionsfehler sollen vom Betrachter möglichst nicht visuell wahrgenommen bzw. als nicht störend empfunden werden. Es geht demzufolge um die Problematik, wie gut ein verlustbehaftetes Kompressionsmodell das subjektive Empfinden nachbildet bzw. das Wahrnehmungsmodell des Empfängers trifft.

Tendenziell bemerkt das menschliche Wahrnehmungssystem bei der Darstellung von Bildern durch Überlagerung von Basisfunktionen Veränderungen von Transformationskoeffizienten, z.B. durch Quantisierung, eher, wenn die Basisfunktionen zur Signalrekonstruktion nicht über eine gewisse Glattheit verfügen [BEL07]. Insbesondere Rekonstruktionsfehler, die sich in glatten tieffrequenten Bildbereichen durch plötzlich auftretende schnelle Grauwertwechsel bemerkbar machen, werden als sehr störend empfunden. Um in der Bilddatenkompression

¹LBT: lapped biorthogonal transform, Verwendung einer zweifach überlappenden biorthogonalen trigonometrischen Basis aus [Bit00b].



Abbildung 1.3: Sichtbare Ringing-Artefakte, die in der Umgebung von Kanten durch die Rekonstruktion mit schlecht lokalisierten Basisfunktionen nach starker Quantisierung der Wichtungskoeffizienten auftreten.

visuell gute Ergebnisse zu erzielen, sollte die Glattheit der Basisfunktionen im Bereich der Glattheit des Signals liegen. Werden glatte Bildbereiche mit unglatten Basisfunktionen wie z.B. dem Haar-Wavelet oder den Basisfunktionen der Block-Kosinustransformation rekonstruiert, so kann die Quantisierung der Transformationskoeffizienten zu Unstetigkeiten und somit deutlich sichtbaren Grauwertsprüngen bzw. Blockartefakten führen.

Verschwindende Momente – Rekonstruktions- und Auflösungseigenschaften

Da sich lokal glatte Bildbereiche gut durch stückweise Polynome approximieren lassen, besteht ein Ansatz zur Konstruktion glatter Basisfunktionen darin, Rekonstruktionsbasen mit polynomreproduzierenden Eigenschaften bzw. duale Basen mit verschwindenden Momenten zu versehen [Dau92, Ode96, DN98]. Der Begriff verschwindende Momente steht in enger Beziehung mit den Rekonstruktions- und Auflösungseigenschaften von Wavelets [BEL07]. Ein duales Wavelet

$$\tilde{\psi}_{j,k}(t) = 2^{j/2} \tilde{\psi}(2^j t - k) \quad (1.20)$$

zur Signalanalyse, das sich durch Skalierung und Verschiebung aus einem Mutter-Wavelet $\tilde{\psi}(t) \in L^2(\mathbb{R})$ ergibt, besitzt M verschwindende Momente, wenn für alle $m = 0, 1, \dots, M-1$ die Bedingung

$$\int_{\mathbb{R}} t^m \tilde{\psi}_{j,k}(t) dt = 0 \quad (1.21)$$

erfüllt ist [BSMM08]. Das heißt, dass z.B. für $M = 4$ das Wavelet orthogonal zu kubischen, quadratischen, linearen und konstanten Funktionen ist. Bei der Wavelet-Transformation werden diese Polynome somit nicht vom Wavelet erfasst und die Wavelet-Koeffizienten sind Null. Die polynomialen Signalanteile gehen jedoch nicht verloren, sondern werden durch die Skalierungskoeffizienten repräsentiert.

Die Existenz von verschwindenden Momenten ist für die verlustbehaftete waveletbasierte Bild-datenkompression von wesentlicher Bedeutung. Je mehr Momente verschwinden, desto besser

lassen sich glatte Bildbereiche durch die Skalierungskoeffizienten der gröbsten Skala approximieren und desto schneller fallen die Wavelet-Koeffizienten ab. Diese Eigenschaft lässt sich mit Hilfe der Taylor-Formel erläutern. Hierzu wird die Wavelet-Analyse eines Signals f auf dem Intervall $I_{j,k} = \text{supp } \tilde{\psi}_{j,k}$ mit einem dualen Wavelet $\tilde{\psi}_{j,k}$ betrachtet:

$$\langle f, \tilde{\psi}_{j,k} \rangle = \int_{I_{j,k}} f(t) \tilde{\psi}_{j,k}(t) dt. \quad (1.22)$$

Ist das Signal f auf dem Intervall $I_{j,k}$ stetig, besitzt es dort stetige Ableitungen bis einschließlich der Ordnung $M - 1$ und existiert im Innern des Intervalls noch die M -te Ableitung, dann gilt die Taylor-Entwicklung [BSMM08]

$$\begin{aligned} f(t) &= f(t_0) + \frac{f'(t_0)}{1!}(t - t_0) + \dots + \frac{f^{(M-1)}(t_0)}{(M-1)!}(t - t_0)^{M-1} + \frac{f^{(M)}(\xi)}{M!}(t - t_0)^M \\ &= \left(\sum_{m=0}^{M-1} \frac{f^{(m)}(t_0)}{m!}(t - t_0)^m \right) + \frac{f^{(M)}(\xi)}{M!}(t - t_0)^M, \quad t_0 = 2^{-j}k. \end{aligned} \quad (1.23)$$

Durch Einsetzen in Gleichung (1.22) erhält man die Wavelet-Koeffizienten

$$\langle f, \tilde{\psi}_{j,k} \rangle = \left(\sum_{m=0}^{M-1} \frac{f^{(m)}(t_0)}{m!} \underbrace{\int_{I_{j,k}} (t - t_0)^m \tilde{\psi}_{j,k} dt}_{=0} \right) + \frac{1}{M!} \int_{I_{j,k}} f^{(M)}(\xi)(t - t_0)^M \tilde{\psi}_{j,k} dt, \quad (1.24)$$

wobei angenommen wird, dass das duale Wavelet M verschwindende Momente besitzt. Hierdurch verschwinden nach Gleichung (1.21) die Integrale der ersten M Glieder der Taylor-Formel und es trägt nur noch das Restglied zur Bestimmung der Wavelet-Koeffizienten bei. Der Betrag dieser Koeffizienten lässt sich durch

$$|\langle f, \tilde{\psi}_{j,k} \rangle| \leq C 2^{-jM} \|f^{(M)}\| \quad (1.25)$$

abschätzen [BD11]. Die Konstante C hängt nur von der Wahl des dualen Wavelets $\tilde{\psi}$ und nicht von den Parametern j und k sowie dem Signal f ab. Die Wavelet-Koeffizienten fallen somit exponentiell mit zunehmendem Verfeinerungslevel j ab. Der exponentielle Abfall ist um so höher, je mehr verschwindende Momente M das duale Wavelet $\tilde{\psi}_{j,k}$ besitzt. Mit steigendem Level j verkleinert sich der Träger des Wavelets und die Wavelet-Koeffizienten streben für stückweise glatte Signale schnell gegen Null. Die Koeffizienten, die fast Null sind, werden in der verlustbehafteten Datenkompression verworfen. Lediglich in den Bereichen, in denen das Signal eine hohe örtliche Aktivität aufweist, erhält man Wavelet-Koeffizienten mit größeren Beträgen. Diese Koeffizienten sollten mit ausreichender Genauigkeit gespeichert werden. Auf der Rekonstruktionsseite werden glatte Signalbestandteile durch wenige Skalierungsfunktionen approximiert und Detailinformationen durch Addition mit Wavelets hinzugefügt.

Da sich der Ansatz, Wavelets mit polynomreproduzierenden Eigenschaften zu versehen, für die Bilddatenkompression als vorteilhaft erwiesen hat, besteht in Analogie zur Wavelet-Repräsentation auch bei den im Rahmen dieser Arbeit untersuchten Systemen $\{\psi_j^k : \text{Ortsparameter } j \in \mathbb{Z}, \text{ Frequenzparameter } k \in \mathbb{N}_0\}$ von zeit-frequenz-lokalen Basisfunktionen das Ziel, Polynome höherer Ordnung mit wenigen Basisfunktionen zu repräsentieren. Dementsprechend sind die dualen Basisfunktionen $\tilde{\psi}^k$, $k > 0$, zur Signalanalyse mit verschwindenden Momenten zu versehen. Das Skalarprodukt zwischen den dualen Basisfunktionen $\tilde{\psi}^k$, $k > 0$, und den Polynomen bis zu einer Ordnung von $M - 1$ soll verschwinden:

$$\int_{\mathbb{R}} t^m \tilde{\psi}^k(t) dt = 0, \quad \text{für alle } k > 0, m = 0, 1, \dots, M - 1, \quad (1.26)$$



Abbildung 1.4: Kompressionsartefakte durch die Verwendung einer trigonometrischen Transformation, die den Gleichanteil nicht ausschließlich mit den Basisfunktionen der niedrigsten Frequenz repräsentiert.

so dass eine sparsame lokale Darstellung dieser Polynome ausschließlich mit den Basisfunktionen ψ_j^0 erfolgt. Lokal glatte Bildbereiche werden somit durch wenige Basisfunktionen ψ_j^0 approximiert und Detailinformationen durch Addition mit Basisfunktionen $\psi_j^{k>0}$ hinzugefügt. In der Bilddatenkompression stellt die Auflösung und Darstellung von Konstanten mit den Basisfunktionen der niedrigsten Frequenz eine Mindestforderung dar. Um die Notwendigkeit dieser Bedingung zu veranschaulichen, ist in Abbildung 1.4 die Kompression eines Bildes gezeigt, wobei die beispielhaft verwendete Transformation nicht die Fähigkeit besitzt, eine Konstante mit den Basisfunktionen der niedrigsten Frequenz zu rekonstruieren. Dadurch werden Basisfunktionen höherer Frequenz zur Darstellung des Gleichanteils benötigt. Die Wichtungskoeffizienten dieser Basisfunktionen stehen jedoch nach der Quantisierung nicht mehr in der erforderlichen Genauigkeit zur Verfügung und so entstehen schon bei geringen Kompressionsfaktoren deutlich sichtbare Kompressionsartefakte.

Eine hohe Anzahl von verschwindenden Momenten bzw. die Fähigkeit, Signale möglichst hoher Polynomordnung mit den Basisfunktionen der niedrigsten Frequenz zu rekonstruieren, führt nicht zwangsläufig zu einer für die Bilddatenkompression geeigneteren Signalrepräsentation. Je höher die Ordnung der Polynome gewählt wird, die ausschließlich mit Basisfunktionen der niedrigsten Frequenz dargestellt werden sollen, desto mehr Freiheitsgrade sind bei der Konstruktion dieser Basisfunktionen erforderlich. Mehr Freiheitsgrade erhält man bei der Verwendung von überlappenden Basisfunktionen durch die Erhöhung des Überlappungsfaktors, wobei sich bei gleichbleibender Blockgröße die Trägerlänge der Basisfunktionen erhöht. Hierdurch verschlechtert sich jedoch die Lokalität der Basisfunktionen im Zeitbereich und im Allgemeinen auch die Riesz-Stabilität, so dass bei der Entwicklung und Auswahl von Basisfunktionen ein Kompromiss zwischen polynomreproduzierender Eigenschaften, Riesz-Stabilität und Trägerlänge einzugehen ist.

Existenz schneller Algorithmen

Eine weitere Eigenschaft von Transformationen, die u.a. auch bei der Beurteilung der Leistungsfähigkeit von Transformationscodern Berücksichtigung findet, ist der Berechnungsauf-

wand, der für die Realisierung der Transformation benötigt wird. Die Existenz schneller Algorithmen entscheidet mit über das Einsatzspektrum einer Transformation. Für die schnelle Realisierung von Transformationen sind z.B. kurze Filterlängen bzw. Basisfunktionen mit kurzen Trägerlängen von Vorteil. Eine andere Möglichkeit der Entwicklung schneller Algorithmen besteht darin, strukturelle Eigenschaften der Basen auszunutzen.

1.2 In der Bilddatenkompression häufig verwendete Transformationen

Die derzeit am häufigsten genutzten Kompressionsverfahren basieren auf der Kosinus- oder der Wavelet-Transformation. Diese Transformationen finden auch bei den im Rahmen dieser Arbeit eingeführten Kompressionsstrategien Anwendung und werden daher im Folgenden kurz vorgestellt.

1.2.1 Kosinustransformation

Die Kosinustransformation [ANR74] kommt in vielen Verfahren zur Signal- und Bildverarbeitung zum Einsatz. Sie ist unter anderem Bestandteil der Kompressionsstandards JPEG und MPEG-1/-2/-4 ([RH96, MPG93, MPG96, MPG04, Ric06]). Die Basisfunktionen der Kosinustransformation bestehen aus Kosinusfunktionen, die mit einem Rechteckfenster gewichtet werden. Die so konstruierten Basisfunktionen besitzen eine gute Zeitlokalität. Aufgrund der Unstetigkeiten durch die Rechteckfensterung sind sie jedoch im Frequenzbereich schlecht lokalisiert. Für die Darstellung eines Signals mit Kosinusbasen wird das Signal in Blöcke eingeteilt und anschließend für jeden Block die Kosinustransformation durchgeführt. In der Bilddatenkompression auf Basis der diskreten Kosinustransformation werden hierbei überwiegend Blöcke der Größe von $B \times B = 8 \times 8$ Pixeln verwendet.

In [RY90, Wic96] sind vier verschiedene diskrete Kosinustransformationen klassifiziert. Von besonderem Interesse sind im Rahmen dieser Arbeit die Kosinustransformationen vom Typ II, III und IV. Für die diskrete Kosinustransformation, die durch Gleichung

$$c[k] = \sum_{m=0}^{B-1} \mathbf{C}[k, m] f[m] \quad (1.27)$$

definiert ist, lassen sich die Transformationsmatrizen der verwendeten DCT-Transformationen in Abhängigkeit vom Typ durch die Gleichungen

- **DCT-II:**
$$\mathbf{C}_B^{II}[k, m] = b[k] \sqrt{\frac{2}{B}} \cos \left(\frac{\pi k(m + \frac{1}{2})}{B} \right), \quad (1.28)$$

- **DCT-III:**
$$\mathbf{C}_B^{III}[k, m] = b[m] \sqrt{\frac{2}{B}} \cos \left(\frac{\pi(k + \frac{1}{2})m}{B} \right), \quad (1.29)$$

- **DCT-IV:**
$$\mathbf{C}_B^{IV}[k, m] = \sqrt{\frac{2}{B}} \cos \left(\frac{\pi(k + \frac{1}{2})(m + \frac{1}{2})}{B} \right) \quad (1.30)$$

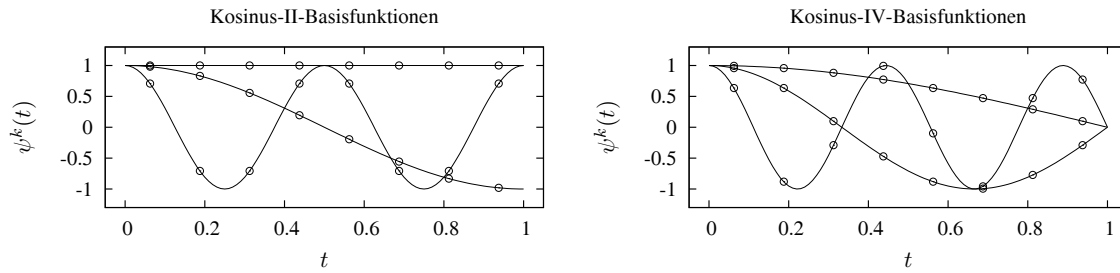


Abbildung 1.5: Kosinus-II-Basisfunktionen (links) und Kosinus-IV-Basisfunktionen (rechts) für die Frequenzparameter $k = 0, 1$ und 4 .

bestimmen, wobei $m = 0, 1, \dots, B - 1$ und $k = 0, 1, \dots, B - 1$ sowie

$$b[n] = \begin{cases} 1/\sqrt{2} & \text{falls } n = 0, \\ 1, & \text{falls } 0 < n < B. \end{cases} \quad (1.31)$$

Für die Bilddatenkompression sind die DCT-II und ihre Inverse, die DCT-III, von besonderem Interesse, da sie den Gleichanteil mit der Basisfunktion der niedrigsten Frequenz (d.h. Frequenzparameter $k = 0$) auflösen bzw. rekonstruieren. Diese Transformationen werden in der Signal- und Bildverarbeitung oft nur als DCT und IDCT bezeichnet. Mit der zu sich selbst inversen DCT-IV ist eine direkte Repräsentation des Gleichanteils mit der Basisfunktion der niedrigsten Frequenz nicht möglich. Diese Transformation findet jedoch in den schnellen Algorithmen der Transformation mit überlappenden lokalen trigonometrische Basen Anwendung (vgl. Kapitel 4).

Für ausgewählte Frequenzparameter sind die kontinuierlichen Kosinus-II- und Kosinus-IV-Basisfunktionen in Abbildung 1.5 dargestellt. Die in den Transformationskernen für $B = 8$ verwendeten Abtastwerte¹ sind ebenfalls in den Diagrammen eingezeichnet.

Schnelle Realisierungen

Seit den 80er Jahren wurden umfangreiche Untersuchungen zur schnellen Realisierung von Kosinustransformationen durchgeführt, so dass heute viele leistungsfähige Algorithmen existieren. Schnelle Algorithmen erhält man durch eine geschickte Faktorisierung der Transformationsmatrizen in dünn besetzte Matrizen. Zum Beispiel sind bei einer Blockgröße von $B = 8$ für die direkte Implementierung der Kosinustransformation als Matrix-Vektor-Produkt 64 Multiplikationen und 56 Additionen notwendig. Das entspricht 15 Operationen pro Transformationskoeffizient. Die schnellsten Algorithmen benötigen jedoch mit 11 Multiplikationen und 29 Additionen für die DCT-II/III bzw. mit 20 Multiplikationen und 36 Additionen für die DCT-IV nur noch 5 bzw. 7 arithmetische Operationen pro Transformationskoeffizient.

Zu den bekanntesten schnellen Algorithmen gehören u.a. die Entwicklungen von Chen, Smith und Fralick [CSF77], Wang [Wan84], Lee [Lee84], Vetterli und Nussbaumer [VN84], Loeffler, Ligtenberg und Moschytz [LLM89], Hou [Hou87] Yip und Rao [YR84], Feig und Winograd [Fei90, FW92], Malvar [Mal92, CG90, DMP91], Steidl [Ste92], Steidl und Tasche [ST91], Plonka und Tasche [PT02], Chan und Ho [CH90] sowie von Makhoul [Mak80]. Es ist zu beachten, dass bei einigen Algorithmen der Wichtungskoeffizient der Basisfunktion der niedrigsten Frequenz nicht normalisiert wird, um eine Multiplikation einzusparen. Bei einigen Algorithmen

¹Die in Abbildung 1.5 eingezeichneten Abtastwerte sind mit $b[k]\sqrt{\frac{2}{B}}$ für die DCT-II bzw. mit $\sqrt{\frac{2}{B}}$ für die DCT-IV zu multiplizieren.

werden auch zwei Multiplikationen eingespart, jedoch mit dem Resultat, dass alle Koeffizienten mit einem Faktor skaliert sind. Für die Anwendung der DCT in der Bilddatenkompression können nach Möglichkeit und in Abhängigkeit von der Quantisierungsstrategie Multiplikationen der Transformationsalgorithmen auch in die Quantisierungsmatrix verschoben werden. Die Transformation wird dann als skalierte DCT bezeichnet. So benötigt z.B. der Algorithmus von Arai, Agui und Nakajima [AAN88] mit diesem Ansatz statt 13 nur noch 5 Multiplikationen.

Es existiert bereits eine Reihe von Bibliotheken und Softwareimplementierungen zur effizienten Realisierung der Kosinustransformationen. Hierzu gehört z.B. die Implementierung der DCT-I bis DCT-IV von Baszenski und Tasche [BT97b], die auf den Algorithmen in [BT97a, ST91] basiert. Der Anwender kann bei dieser Implementierung die Blockgröße für die trigonometrischen Transformationen festlegen und auswählen, ob normierte Transformationen oder zur Einsparung von Rechenoperationen skalierte Transformationen verwendet werden sollen. Eine sehr schnelle Realisierung von diskreten trigonometrischen Transformationen ist auch mit der FFTW¹ C-Bibliothek [FJ98, FJ99, FJ05] möglich. Hier ist zu beachten, dass grundsätzlich skalierte Transformationen verwendet werden. Ein sehr leistungsfähiges Tool zur automatischen Generierung und Optimierung von Algorithmen ist Spiral [PMJ⁺05, EP01, EJP⁺01]. Es erzeugt unter anderem für trigonometrische Transformationen einen Code, der teilweise genauso wenige arithmetische Operationen benötigt wie die schnellsten bisher zitierten Verfahren.

1.2.2 Wavelet-Transformation

In neueren Verfahren zur Bilddatenkompression wurde die DCT durch die Wavelet-Transformation abgelöst. So basiert z.B. der JPEG2000 Standard [JPE00] auf der Wavelet-Transformation. In der klassischen Wavelet-Theorie werden Wavelets $\psi_{j,k}$ durch Skalierung und Verschiebung aus einem Mutter-Wavelet ψ gebildet, wobei sich für viele Anwendungen $\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$ als vorteilhaft erwiesen hat. Die Wavelet-Entwicklung eines Signals ist durch

$$f = \sum_{k \in \mathbb{Z}} c_{j_0,k} \phi_k + \sum_{j=j_0}^{\infty} \sum_{k \in \Lambda_j} d_{j,k} \psi_{j,k} \quad (1.32)$$

mit

$$c_{j_0,k} = \int_{\mathbb{R}} f(t) \tilde{\phi}_{j_0,k}(t) dt, \quad d_{j,k} = \int_{\mathbb{R}} f(t) \tilde{\psi}_{j,k}(t) dt, \quad (1.33)$$

wobei $\langle \psi_{j,k}, \tilde{\psi}_{j',k'} \rangle = \delta_{jj'} \delta_{kk'}$, gegeben [BD11]. Die Funktionen ϕ_k heißen Skalierungsfunktionen. In Abhängigkeit von der Anzahl verschwindender Wavelet-Momente M approximieren sie stückweise glatte Signalbereiche durch Polynome mit einer Ordnung von bis zu $M-1$ (vgl. Abschnitt 1.1.3). Feine Signaldetails werden durch Überlagerung mit Wavelets dargestellt und mit zunehmendem Verfeinerungslevel erhöht sich die Auflösung.

Schnelle Wavelet-Transformation

Das Konzept der Multi-Skalen-Analyse [Mal89, Mey90, Mal98] ermöglicht eine schnelle Filterbankrealisierung der Wavelet-Transformation. Hierbei werden die Wavelet-Koeffizienten nicht rechenaufwändig über Skalarprodukte zwischen Signal und allen dualen Wavelets $\tilde{\psi}_{j,k}$

¹FFTW: Fastest Fourier Transform in the West

bestimmt (vgl. Gleichung (1.33)), sondern durch sukzessive Anwendung von Tief- und Hochpassfiltern berechnet. Die Multi-Skalen-Analyse eines Signals

$$f = \sum_{k \in \mathbb{Z}} c_{J,k} \phi_{J,k} \quad (1.34)$$

basiert auf dualen Skalierungsfunktionen

$$\tilde{\phi}_{j,k}(t) = 2^{j/2} \tilde{\phi}(2^j t - k) \quad (1.35)$$

und der Existenz einer Folge $\{h_{0,k}\}_{k \in \mathbb{Z}} \in \ell^2(\mathbb{Z})$ von Filterkoeffizienten, so dass $\tilde{\phi}$ eine Zweiskalenrelation der Form

$$\tilde{\phi}(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_{0,k} \tilde{\phi}(2t - k) \quad (1.36)$$

erfüllt [Mal98, Pri06]. Die dualen Wavelets $\tilde{\psi}_{j,k}$ zur Signalanalyse erhält man nach

$$\tilde{\psi}_{j,k}(t) = 2^{j/2} \tilde{\psi}(2^j t - k) \quad (1.37)$$

durch Skalierung und Verschiebung eines dualen Mutter-Wavelets $\tilde{\psi}$, das aus der dualen Skalierungsfunktion $\tilde{\phi}$, dem sogenannten Vater-Wavelet, und einer Folge $\{h_{1,k}\}_{k \in \mathbb{Z}} \in \ell^2(\mathbb{Z})$ von Filterkoeffizienten konstruiert wird:

$$\tilde{\psi}(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_{1,k} \tilde{\phi}(2t - k). \quad (1.38)$$

Durch Einsetzen von Gleichung (1.36) in (1.35) und von Gleichung (1.38) in (1.37) ergeben sich die Beziehungen

$$\tilde{\phi}_{j-1,k} = \sum_{l \in \mathbb{Z}} h_{0,l-2k} \tilde{\phi}_{j,l}, \quad \tilde{\psi}_{j-1,k} = \sum_{l \in \mathbb{Z}} h_{1,l-2k} \tilde{\phi}_{j,l}. \quad (1.39)$$

Mit ihrer Hilfe lassen sich für die Bestimmung der Skalierungs- und Wavelet-Koeffizienten die Gleichungen

$$c_{j-1,k} = \langle f, \tilde{\phi}_{j-1,k} \rangle = \sum_{l \in \mathbb{Z}} h_{0,l-2k} \langle f, \tilde{\phi}_{j,l} \rangle = \sum_{l \in \mathbb{Z}} h_{0,l-2k} c_{j,l}, \quad (1.40)$$

$$d_{j-1,k} = \langle f, \tilde{\psi}_{j-1,k} \rangle = \sum_{l \in \mathbb{Z}} h_{1,l-2k} \langle f, \tilde{\phi}_{j,l} \rangle = \sum_{l \in \mathbb{Z}} h_{1,l-2k} c_{j,l} \quad (1.41)$$

herleiten. Somit können die Wavelet- und Skalierungskoeffizienten ausgehend von der Folge $c_J = \{f_k\}_{k \in \mathbb{Z}}$ [SN96] rekursiv für $j = J, J-1, \dots, J-m+1$ nach dem Schema

$$\begin{array}{ccccccc} c_J & \rightarrow & c_{J-1} & \rightarrow & c_{J-2} & \dots & c_{J-m+1} & \rightarrow & c_{J-m} \\ & \searrow & & \searrow & & & & \searrow & \\ & & d_{J-1} & & d_{J-2} & & & & d_{J-m} \end{array}$$

berechnet werden. In Analogie zur Signalanalyse betrachtet man auf der Syntheseseite die Beziehungen

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k), \quad \phi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_{0,k} \phi(2t - k), \quad (1.42)$$

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k), \quad \psi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_{1,k} \phi(2t - k) \quad (1.43)$$

mit den Filterkoeffizientenfolgen $\{g_{0,k}\}_{k \in \mathbb{Z}}$, $\{g_{1,k}\}_{k \in \mathbb{Z}} \in \ell^2(\mathbb{Z})$ und es sei die Dualitätsbedingung $\langle \phi, \tilde{\phi}(t - k) \rangle = \delta_{0,k}$ erfüllt. Aus (1.42) und (1.43) ergibt sich

$$\phi_{j-1,k} = \sum_{l \in \mathbb{Z}} g_{0,l-2k} \phi_{j,l}, \quad \psi_{j-1,k} = \sum_{l \in \mathbb{Z}} g_{1,l-2k} \phi_{j,l}. \quad (1.44)$$

Dann folgt aus

$$\sum_{k \in \mathbb{Z}} c_{j,k} \phi_{j,k} = \sum_{l \in \mathbb{Z}} c_{j-1,l} \phi_{j-1,l} + \sum_{l \in \mathbb{Z}} d_{j-1,k} \psi_{j-1,l} \quad (1.45)$$

unter Ausnutzung der vorgenannten Beziehungen die Rekursionsgleichung

$$c_{j,k} = \sum_{l \in \mathbb{Z}} g_{0,k-2l} c_{j-1,l} + \sum_{l \in \mathbb{Z}} g_{1,k-2l} d_{j-1,l} \quad (1.46)$$

zur Berechnung der Skalierungskoeffizienten $c_{j,k}$ in der Skala 2^j aus den Skalierungs- und Wavelet-Koeffizienten der Skala 2^{j-1} . Für $j = J - m + 1, J - m + 2, \dots, J$ kann die Koeffizientenfolge c_J somit rekursiv nach dem Schema

$$\begin{array}{ccccccc} c_{J-m} & \rightarrow & c_{J-m+1} & \rightarrow & c_{J-m+2} & \cdots & c_{J-1} & \rightarrow & c_J \\ & \nearrow & & \nearrow & & & & \nearrow & \\ d_{J-m} & & d_{J-m+1} & & & & d_{J-1} & & \end{array}$$

zurückgewonnen werden.

Weisen die Filterkoeffizientenfolgen h_0 , h_1 , g_0 und g_1 nur wenige Nichtnullelemente auf (kurze Filterlängen), wie es bei gebräuchlichen Wavelets der Fall ist, dann besitzen die rekursiven Berechnungsalgorithmen zur Realisierung der Wavelet-Transformation eine geringe arithmetische Komplexität.

Filterbankrealisierung

Die Beziehungen in den Gleichungen (1.40) und (1.41) beschreiben die Filterung einer Koeffizientenfolge c_j mit einem Tiefpass \bar{h}_0 , $\bar{h}_0[n] = h_0[-n]$, und einem Hochpass \bar{h}_1 , $\bar{h}_1[n] = h_1[-n]$, mit anschließender Unterabtastung um den Faktor 2, d.h.

$$c_{j-1}[k] = \sum_{l \in \mathbb{Z}} h_0[l - 2k] c_j[l] = \sum_{l \in \mathbb{Z}} \bar{h}_0[2k - l] c_j[l] \implies c_{j-1} = (\downarrow 2)(c_j * \bar{h}_0), \quad (1.47)$$

$$d_{j-1}[k] = \sum_{l \in \mathbb{Z}} h_1[l - 2k] c_j[l] = \sum_{l \in \mathbb{Z}} \bar{h}_1[2k - l] c_j[l] \implies d_{j-1} = (\downarrow 2)(c_j * \bar{h}_1). \quad (1.48)$$

Auf der Rekonstruktionsseite erhält man nach Gleichung (1.46)

$$c_j = g_0 * (\uparrow 2) c_{j-1} + g_1 * (\uparrow 2) d_{j-1}, \quad (1.49)$$

wobei die Wavelet- und Skalierungskoeffizienten vor der Filterung um den Faktor 2 aufwärtsgetastet werden. Somit lässt sich die schnelle Wavelet-Transformation durch eine Zwei-Kanal-Filterbank realisieren, wobei die Mehrfachauflösung durch die weitere Zerlegung des Tiefpassanteils erzielt wird (vgl. Abbildung 1.6).

Zur Bestimmung geeigneter Filterkoeffizienten bzw. Wavelets sei u.a. auf [Dau92] und die Darstellungen in [Str98, Str09] verwiesen. In der verlustbehafteten Bilddatenkompression kommt häufig das in [ABMD92] vorgestellte biorthogonale 9/7-Filterpaar zur Anwendung, z.B. auch bei JPEG2000.

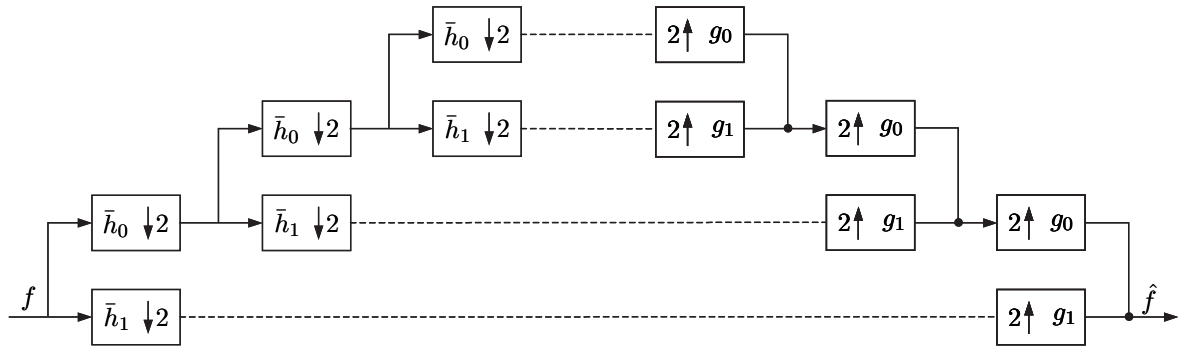


Abbildung 1.6: Filterbank zur schnellen Realisierung der Wavelet-Transformation am Beispiel von drei Zerlegungsstufen.

1.3 Skalare Quantisierung

Nach der Transformation liegen die Koeffizienten zur Darstellung des Signals im Allgemeinen in einer hohen Auflösungsgenauigkeit als Gleitkommazahlen vor. Die Aufgabe der Quantisierung besteht nun darin, für den Empfänger irrelevante Informationen zu entfernen und somit die Datenmenge zu reduzieren. Hierzu erfolgt eine Einteilung des Wertebereiches in Intervalle, wobei jedem Koeffizienten c , der zu einem Intervall q gehört, ein Rekonstruktionswert \hat{c}_q des Intervalls zugeordnet wird. Dieser Vorgang ist verlustbehaftet, die originalen Transformationskoeffizienten lassen sich nach diesem Schritt nicht mehr zurückgewinnen. Die Differenz zwischen dem originalen und dem rekonstruierten Signal wird durch den Quantisierungsfehler $e_q = c - \hat{c}_q$ beschrieben.

In Abhängigkeit von der Unterteilung des Wertebereiches in Intervalle unterscheidet man zwischen gleichmäßiger und ungleichmäßiger Quantisierung [Ohm95, Str09], wobei im Folgenden ausschließlich die gleichmäßige Quantisierung Berücksichtigung findet. Bei der gleichmäßigen Quantisierung besitzen die Intervalle eine feste Breite Q . Gehört der Wert $\hat{c} = 0$ zur Rekonstruktionsmenge, so spricht man von einer Midtread-Quantisierung. Quantisierung und Rekonstruktion erfolgen nach

$$q = \left\lfloor \frac{|c|}{Q} + \frac{1}{2} \right\rfloor \cdot \text{sgn}(c), \quad \hat{c} = q \cdot Q, \quad (1.50)$$

wobei die Signumfunktion durch

$$\text{sgn}(c) := \begin{cases} +1, & \text{falls } c > 0, \\ 0, & \text{falls } c = 0, \\ -1, & \text{falls } c < 0 \end{cases} \quad (1.51)$$

definiert ist. Beinhaltet die Menge der Rekonstruktionswerte nicht den Wert $\hat{c} = 0$, so handelt es sich um die Midrise-Quantisierung

$$q = \left\lfloor \frac{|c|}{Q} + 1 \right\rfloor \cdot \text{sgn}(c), \quad \hat{c} = q \cdot \left(Q - \frac{1}{2} \right). \quad (1.52)$$

Diese Quantisierung wird in der Praxis jedoch selten verwendet [Str09]. Der Quantisierungsfehler liegt bei diesen gleichförmigen Quantisierungsverfahren im Bereich

$$-\frac{Q}{2} \leq e_q \leq \frac{Q}{2}. \quad (1.53)$$

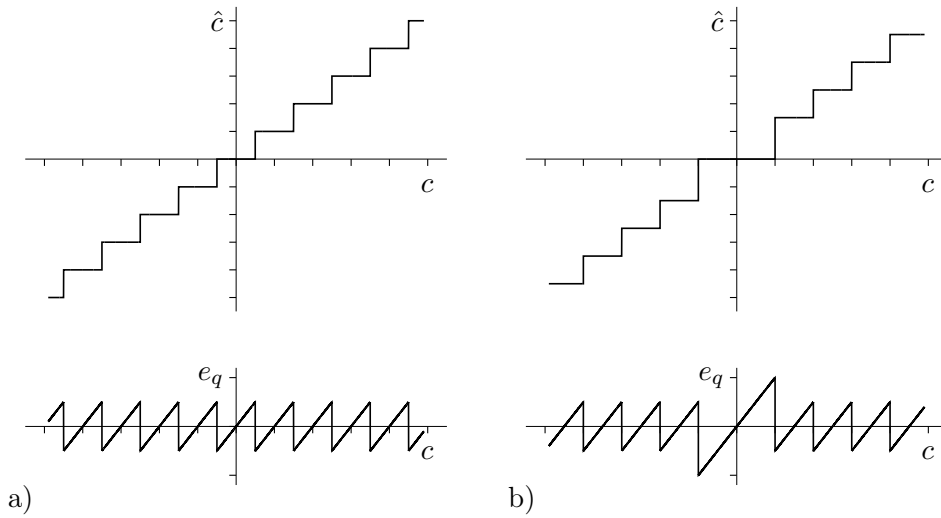


Abbildung 1.7: Quantisierungskennlinie und Quantisierungsfehler, a) gleichmäßige Quantisierung mit konstanter Intervallbreite (Midtread-Quantisierung), b) gleichmäßige Quantisierung mit doppelter Intervallbreite um den Nullpunkt (Totzonen-Quantisierung).

Das Intervall um den Wert Null ist für die Datenkompression von besonderer Bedeutung. In diesem Bereich werden die Signalanteile entfernt, die ausgehend vom Idealfall unter der Wahrnehmungsgrenze des Empfängers liegen. Um die Anzahl der zu codierenden Darstellungskoeffizienten mit geringen Amplitudenwerten zu verringern, wird häufig das Quantisierungsintervall um den Nullpunkt verbreitert. Eine Verdoppelung dieses Intervalls lässt sich durch

$$q = \left\lfloor \frac{|c|}{Q} \right\rfloor \cdot \text{sgn}(c), \quad \hat{c} = Q \cdot \left(q + \frac{1}{2} \cdot \text{sgn}(q) \right) \quad (1.54)$$

erzielen. Diese Form der Quantisierung bezeichnet man als gleichmäßige Quantisierung mit Totzone. In Abbildung 1.7 ist sie der Midtread-Quantisierung gegenübergestellt.

1.4 Entropiecodierung

Der Begriff Entropiecodierung umfasst Methoden zur verlustfreien Kompression von Daten durch Reduktion der Codierungsredundanz. Betrachtet man Daten als Folgen von Zeichen bzw. Symbolen aus einem Alphabet $\mathbb{S} = \{s_i\}$, $i = 0, 1, \dots, N-1$, das die Menge aller vorkommenden Symbole beschreibt, so besitzt ein Symbol s_i in Abhängigkeit von seiner Auftretenswahrscheinlichkeit p_i einen Informationsgehalt von

$$I(s_i) = -\log_2 p_i \quad (\text{Einheit: bit}). \quad (1.55)$$

Je häufiger ein bestimmtes Symbol auftritt, desto geringer ist der mit diesem Symbol verbundene Informationsgehalt. Als Maß für den mittleren Informationsgehalt eines Alphabets \mathbb{S} mit N voneinander unabhängigen Symbolen wird die Entropie

$$H = - \sum_{i=0}^{N-1} p_i \log_2 p_i \quad (\text{Einheit: bit/Symbol}) \quad (1.56)$$

definiert. Um ein und dieselbe Symbolfolge zu repräsentieren, können verschiedene Codes verwendet werden. Die Abweichung zwischen der durchschnittlich benötigten Datenmenge pro Symbol und der Entropie wird als Codierungsredundanz bezeichnet. Häufig bestehen zwischen den Symbolen statistische Bindungen. Die Wahrscheinlichkeit, mit der einzelne Symbole auftreten, ist nicht unabhängig von vorangegangenen Symbolen. Durch die Definition der bedingten Entropie werden diese Abhängigkeiten berücksichtigt. Die Differenz zwischen der bedingten Entropie und der Entropie ohne Berücksichtigung der Abhängigkeiten bezeichnet man als Intersymbolredundanz [Str09].

In der Bilddatenkompression hat die effiziente Codierung der quantisierten Transformationskoeffizienten einen entscheidenden Einfluss auf die Kompressionsleistung des Transformationscoders. Wie bereits erwähnt, setzt sich die Codierung im Allgemeinen aus zwei Teilschritten zusammen, der Präcodierung zur Verminderung der Intersymbolredundanz und der Entropiecodierung zur Reduktion der Codierungsredundanz. Ein Überblick über verschiedene Verfahren zur Prä- und Entropiecodierung ist z.B. in [Str09] zu finden.

Klassische Methoden zur Entropiecodierung sind die Huffman-Codierung [Huf52] und die arithmetische Codierung [RL79, Lan84]. Bei der Huffman-Codierung werden den Symbolen des Quellenalphabets Codewörter mit unterschiedlicher Länge zugeordnet. Häufig auftretende Symbole erhalten kurze, selten auftretende Symbole hingegen lange Codewörter. Im Gegensatz zur Huffman-Codierung wird bei der arithmetischen Codierung der gesamten Symbolfolge ein einziges Codewort zugeordnet. Der Speicheraufwand lässt sich mit diesem Codiervorgang nahezu optimal an den Informationsgehalt der Quelle anpassen. Aufgrund der effizienteren Redundanzreduktion wird im Allgemeinen die arithmetische Codierung bevorzugt¹.

Arithmetische Codierung

Das Prinzip der arithmetischen Codierung besteht darin, die zu verarbeitende Symbolfolge durch ein Intervall von reellen Zahlen zu repräsentieren, wobei sich mit jedem weiteren zu übertragenden Symbol die Intervallbreite in Abhängigkeit von der Auftretenswahrscheinlichkeit des Symbols verkleinert. Hierfür legt man zunächst ein Codebereich $C_0 = [0, 1]$ als Ausgangsintervall fest, unterteilt es in Teilintervalle und weist jedem dieser Teilintervalle ein Symbol zu. Die Größe eines Teilintervalls entspricht dabei prozentual der Auftretenswahrscheinlichkeit des dem Teilintervall zugeordneten Symbols. Jedes Symbol lässt sich somit eindeutig durch die untere und obere Grenze seines Teilintervalls spezifizieren. In Abhängigkeit von dem ersten zu codierenden Symbol wird nun ein Teilintervall ausgewählt und als Codebereich C_1 definiert. Dieses Intervall dient als neues Ausgangsintervall für die Codierung des nachfolgenden Symbols und wird wieder entsprechend der Auftretenswahrscheinlichkeiten der Symbole in Teilintervalle unterteilt. Wiederholt man diesen Vorgang sukzessive für jedes Symbol der Symbolfolge, so erhält man mit dem letzten zu übertragenden Symbol ein Teilintervall, das die gesamte Symbolfolge repräsentiert. Als Codewort können anschließend die Nachkommastellen der kürzesten Binärzahl gewählt werden, die in diesem Teilintervall liegt. Betrachtet man z.B. ein Alphabet $\{a, b, c, d\}$ mit den Auftretenswahrscheinlichkeiten $p_a = 0.1$, $p_b = 0.3$, $p_c = 0.4$ und $p_d = 0.2$, so erhält man für die Codierung einer beispielhaft gewählten Symbolfolge $c a d b c b d b c c$ das Endintervall $C_{10} = [0.4341046016, 0.4341018368]$. Für die ersten 5 zu codierenden Symbole der Symbolfolge ist die Intervallzerlegung in Abbildung 1.8 dargestellt. Das Codewort ergibt sich aus der Binärdarstellung der oberen Intervallgrenze, indem man die Vorkommastelle entfernt und die Binärfolge mit der Ziffer abbricht, in der sich erstmals die Binärzahl der oberen Intervallgrenze von der Binärzahl der unteren Inter-

¹In praktischen Anwendungen wird jedoch häufig auf die Huffman-Codierung zurückgegriffen, da die arithmetische Codierung patentrechtlichen Einschränkungen unterliegt [MS04].

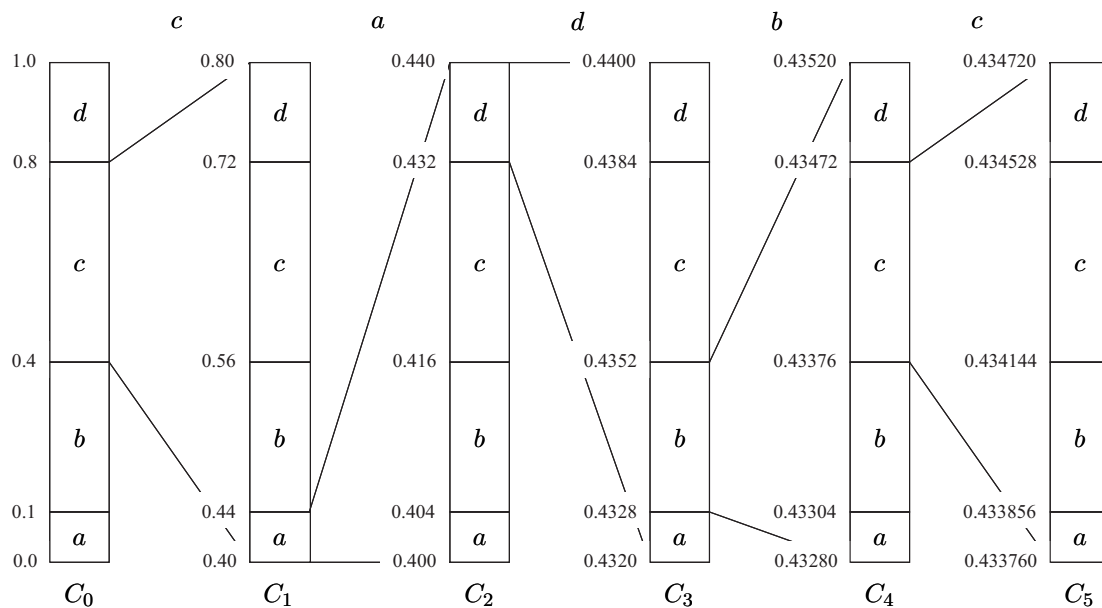


Abbildung 1.8: Arithmetische Codierung, symbolabhängige Intervallzerlegung.

vallgrenze unterscheidet:

obere Intervallgrenze: $0.4341046016 = 0.\{0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ \underline{1}\} \dots$
 untere Intervallgrenze: $0.4341018368 = 0.\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ \underline{0} \dots$

Das Codewort lautet somit 0110111100100001011 und besitzt eine Länge von 19 Bits. Das entspricht einer durchschnittlich benötigten Datenmenge von 1.9 Bits/Symbol. In der allgemeinen binären Schreibweise würde man für eine eindeutige Zuordnung dagegen eine durchschnittliche Datenmenge von 2 Bits/Symbol benötigen. Die Entropie beträgt jedoch nur $H = 1.846$ bit/Symbol.

Die praktische Realisierung der arithmetischen Codierung ist für lange Symbolfolgen mit einigen weiteren Überlegungen verbunden [Wer08]. Zur Bestimmung der Intervallgrenzen ist eine hohe arithmetische Präzision erforderlich. In der Computertechnik erfolgen die Rechenoperationen jedoch nur mit einer begrenzten Genauigkeit. Zudem kann das Codewort erst gesendet werden, wenn die gesamte Nachricht verarbeitet wurde. Lösungen für diese Probleme sind u.a. in [WNC87, Wer08, Str09] dargestellt.

Adaptive und kontextbasierte Modelle

In dem vorangegangenen Codierungsbeispiel wurde davon ausgegangen, dass die Auftretenswahrscheinlichkeiten der einzelnen Symbole a priori bekannt seien und dass das Häufigkeitsmodell statisch im Codec fixiert sei. Da in der Praxis die Wahrscheinlichkeiten bei verschiedenartigen Nachrichten jedoch oft ungewiss sind oder sich innerhalb des Signals ändern können, ist es sinnvoll, adaptive Häufigkeitsmodelle einzuführen. Hierzu wird zunächst von gleich hohen Auftretenswahrscheinlichkeiten der Symbole ausgegangen. Während der Codierung erfolgt dann dynamisch entsprechend der bisher codierten Symbole eine Anpassung der Wahrscheinlichkeiten.

Bereits am Anfang des Abschnitts zur Entropiecodierung wurde darauf hingewiesen, dass zwischen den Symbolen einer Sequenz statistische Bindungen bestehen können und daher

mittels Präcodierung versucht wird, die Intersymbolredundanz zu vermindern (z.B. Lauflängencodierung von Quantisierungssymbolen). Die verbliebene Intersymbolredundanz lässt sich bei der arithmetischen Codierung für eine zusätzliche Verringerung der Datenmenge ausnutzen, indem man bei der Codierung eines Symbols in Abhängigkeit von benachbarten bzw. vorangegangenen Symbolen zwischen verschiedenen Wahrscheinlichkeitsmodellen umschaltet. Die Konstruktion von Modellen in Abhängigkeit von benachbarten bzw. vorangegangenen Symbolen wird als Kontextmodellierung und die Codierung u.a. als kontextbasiert bezeichnet [MBHW01, TT02a].

1.5 Kriterien zur Beurteilung der Leistungsfähigkeit von Kompressionsverfahren

Es existiert eine Reihe von Kriterien, die zur Beurteilung der Leistungsfähigkeit von Kompressionsverfahren und für den Vergleich zwischen verschiedenen Kompressionsstrategien herangezogen werden können. Die vorliegende Arbeit konzentriert sich im Wesentlichen auf die Rekonstruktionsqualität und auf den Berechnungsaufwand (Ressourcenbedarf) der entwickelten Transformationsalgorithmen, die in den vorgeschlagenen Kompressionsstrategien zur Anwendung kommen.

Rekonstruktionsqualität

Die bei einer definierten Bitrate erzielte Rekonstruktionsqualität ist ein wichtiges Kriterium zur Beurteilung der Leistungsfähigkeit von verlustbehafteten Kompressionsverfahren. Die Bitrate wird bei der Codierung von Einzelbildern als durchschnittliche Datenmenge pro Bildpunkt (bpp: bit per pixel) angegeben. Als objektives Maß zur Beurteilung der Rekonstruktionsqualität hat sich in der Bilddatenkompression das Spitzen-Signal-Rausch-Verhältnis (PSNR: peak signal to noise ratio) etabliert. Es berechnet sich nach der Gleichung

$$\text{PSNR} := 10 \cdot \log_{10} \left(\frac{g_{\max}^2}{\text{MSE}(g, \hat{g})} \right) \text{ dB}, \quad (1.57)$$

wobei g_{\max} die maximal mögliche Amplitude darstellt (z.B. $g_{\max} = 255$ bei 8 Bit Auflösung). Der mittlere quadratische Fehler (MSE, mean square error) wurde bereits in Gleichung (1.18) definiert.

Werden in der verlustbehafteten Bilddatenkompression gewisse Bitraten unterschritten, so kann es in den rekonstruierten Bildern zu deutlich sichtbaren Kompressionsartefakten kommen. Es existieren verschiedene Arten von Kompressionsartefakten [YW98, SK98, Pia08], die zudem in Abhängigkeit von der verwendeten Transformation und der Kompressionsrate sehr unterschiedlich stark ausgeprägt sein können. Einige Kompressionsartefakte werden vom Beobachter als weniger, andere hingegen als stark störend empfunden. Da verschiedene Kompressionsstrategien zu sehr unterschiedlichen sichtbaren Kompressionsartefakten führen können und das PSNR das Qualitätsempfinden eines Beobachters nicht im ausreichenden Maße nachbildet [Str09], ist auch eine subjektive Beurteilung der Rekonstruktionsqualität erforderlich. Hierzu werden in der Regel die rekonstruierten Bilder durch mehrere Probanden anhand einer MOS¹-Skala [Ohm95] bewertet.

¹MOS: mean opinion score

Ressourcenbedarf

Neben der Kompressionsleistung sind auch Laufzeit und Speicherbedarf von Kompressionsalgorithmen wichtige Kriterien zur Beurteilung der Leistungsfähigkeit von Kompressionsverfahren. Häufig entscheidet der Ressourcenbedarf eines Transformationscodecs darüber, ob er sich für bestimmte praktische Einsatzzwecke eignet. Es existieren z.B. Anwendungsbereiche (u.a. Echtzeit- und Low-Power-Anwendungen), bei denen ein wesentlich geringerer Ressourcenbedarf wichtiger ist als geringfügige Vorteile in der Kompressionsleistung.

Kapitel 2

Lokale trigonometrische Basen und Klappoperatoren auf der reellen Achse

Dieses Kapitel befasst sich mit grundlegenden theoretischen Betrachtungen zu überlappenden lokalen trigonometrischen Basen sowie Klapp- und Aufklappoperatoren. Die wesentliche Bedeutung von Klapp- und Aufklappoperatoren wird herausgearbeitet und die Anwendung dieser Operatoren in der Signalverarbeitung erläutert. Als Grundlage für die Untersuchungen in Kapitel 3 erfolgt die Herleitung von Klapp- und Aufklappoperatoren für Kosinus-IV-Basen. Des Weiteren werden verschiedene Fensterfunktionen zur Konstruktion von überlappenden Kosinus-IV-Basen beschrieben und die Basen im Hinblick auf die Anwendung in der Bilddatenkompression anhand der in Kapitel 1 erarbeiteten Kriterien bewertet.

Lokale trigonometrische Basen bestehen aus trigonometrischen Funktionen, die mit geeigneten zeitlokalen Fensterfunktionen gewichtet werden. Zu den bekanntesten überlappenden lokalen trigonometrischen Basen zählen die zweifach überlappenden Orthonormalbasen von Malvar [Mal90, Mal92]. Das Besondere an dem Konstruktionsansatz von Malvar ist, dass die Orthonormalität durch leicht zu erfüllende Bedingungen an die Fensterfunktionen erreicht wird [Bit00b]. Die von Malvar hergeleiteten Basen lassen sich als Spezialfall der Wilson-Basen von Daubechies, Jaffard und Journé [DJJ91] interpretieren. Auf Grundlage der Ergebnisse in [Mal90, Lae90, DJJ91] stellten Coifman und Meyer [CM91] fest, dass sich die Konstruktion von Malvar auf allgemeinere Systeme zweifach überlappender Fensterfunktionen erweitern lässt [Bit00b]. Eine detaillierte Betrachtung solcher Basen, die unter die Bezeichnungen Malvar-Basen, Malvar-Wavelets, Coifman-Meyer-Basen oder einfach bzw. zweifach überlappende lokale trigonometrische Basen fallen, ist z.B. in [AWW92, Aus94, Bit00b] zu finden.

Die Ansätze von Daubechies, Jaffard und Journé sowie Coifman und Meyer beschränken sich auf die Konstruktion von orthonormalen Basen. Die Bedingung der Orthonormalität schränkt jedoch die Anzahl der möglichen Fensterfunktionen stark ein. So existiert lediglich eine orthonormale zweifach überlappende trigonometrische Basis, welche die für die Bilddatenkompression grundlegend geforderte Eigenschaft besitzt, den Gleichanteil von Signalen mit Basisfunktionen der niedrigsten Frequenz aufzulösen bzw. zu rekonstruieren (vgl. Abschnitt 1.1.3). Es handelt sich hierbei um die mit einem Sinusfenster konstruierte Malvar-Basis [Mal90, Mal92]. Diese Basis erfüllt jedoch nicht die Eigenschaft der stetigen Differenzierbarkeit. Dadurch wird ein schnelles Abklingen der Nebenmaxima im Spektralbereich verhindert. Die Frequenzselektivität sowie die Konzentration der Signalenergie auf wenige Transformationskoeffizienten verschlechtern sich.

Damit bei der Konstruktion von überlappenden lokalen trigonometrischen Basen mehr Freiheitsgrade für die Realisierung bestimmter Eigenschaften zur Verfügung stehen, verzichtet man häufig auf die restriktive Bedingung der Orthogonalität und betrachtet stattdessen biorthogonale Basen mit hinreichend guter Riesz-Stabilität (vgl. Abschnitte 1.1.2 und 1.1.3). Zur Entwicklung von leistungsfähigen Kompressionsverfahren auf der Grundlage von überlappenden lokalen trigonometrischen Basen sind insbesondere die in Abschnitt 1.1.3 erläuterten Kriterien bei der Konstruktion von Basisfunktionen zu berücksichtigen.

Zweifach überlappende biorthogonale trigonometrische Basen wurden zunächst von Jawerth, Sweldens und Liu [JS95, JLS94, Swe94] sowie von Matviyenko [Mat96] vorgestellt. Sie beschränken sich jedoch auf Fenster, die im Überlappungsbereich gewisse Symmetrieeigenschaften zu erfüllen haben [Bit00b]. Eine allgemeinere Theorie zweifach überlappender biorthogonaler trigonometrischer Basisfunktionen wurde von Chui und Shi [CS96, CS97b, CS97a] entwickelt. Diese Arbeiten bildeten in [Bit98b, Bit99b, Bit00b] die Grundlage für die Herleitung von Fensterfunktionen zur Konstruktion von stetig differenzierbaren zweifach überlappenden Basen, die neben der Darstellung von Konstanten mit Basisfunktionen der niedrigsten Frequenz auch linear verlaufende Signalinhalte mit einer geringen Anzahl von Basisfunktionen niedriger Frequenz repräsentieren können. Die Rekonstruktion von Linearen und Polynomen höherer Ordnung mit Basisfunktionen der niedrigsten Frequenz ist jedoch erst durch die Einführung von mehrfach überlappenden lokalen trigonometrischen Basen durch Bittner und Chui in [Bit02a, BC03, BC04] möglich.

In den folgenden Abschnitten werden Grundlagen der Theorie überlappender lokaler trigonometrischer Basen dargelegt, soweit sie zum Verständnis der sich anschließenden Kapitel erforderlich sind. Abschnitt 2.1 dient der Einführung wesentlicher Definitionen zu überlappenden lokalen trigonometrischen Basen. Als wichtige Hilfsmittel zur Untersuchung von überlappenden lokalen trigonometrischen Basen werden in Abschnitt 2.2 Klapp- und Aufklappoperatoren beschrieben und die Bedeutung dieser Operatoren sowie ihre Funktionsweise erläutert. Es erfolgt die Herleitung von Klapp- und Aufklappoperatoren für überlappende Kosinus-IV-Basen. Die hierbei gewonnenen Erkenntnisse bilden eine wesentliche Grundlage für die Entwicklung neuer Klapp- und Aufklappoperatoren in Kapitel 3. Abschnitt 2.3 gibt einen Überblick über Fensterfunktionen zur Konstruktion von überlappenden lokalen trigonometrischen Basen, die aufgrund ihrer Eigenschaften für die Anwendung in der Bilddatenkompression von Interesse sind.

2.1 Lokale trigonometrische Basen

2.1.1 Definitionen und Grundlagen

Zur Darstellung von Signalen $f \in L^2(\mathbb{R})$ mit lokalen trigonometrischen Basen erfolgt zunächst eine Einteilung der Zeitachse in Intervalle. Im Rahmen dieser Arbeit wird die Zerlegung in äquidistante Intervalle $[j, j+1)$, $j \in \mathbb{Z}$, vorgenommen¹. Die Intervalle haben somit eine Breite von $N_j = 1$. Bei trigonometrischen Blocktransformationen wird auf dem Intervall $[j, j+1)$ das Signal bezüglich der Orthonormalbasis $\{\chi_{[j,j+1)}\theta_j^k : j \in \mathbb{Z}, k \in \mathbb{N}_0\}$ repräsentiert. Die Funktion $\chi_{[j,j+1)}$ beschreibt das Rechteckfenster

$$\chi_{[j,j+1)}(t) := \begin{cases} 1, & \text{für } t \in [j, j+1), \\ 0, & \text{für } t \notin [j, j+1) \end{cases} \quad (2.1)$$

¹Es sei darauf hingewiesen, dass in einigen der zugrunde liegenden Publikationen, wie z.B. in [Bit02a, BC04], eine andere Einteilung der reellen Achse definiert wird.

und wird als charakteristische Funktion bezeichnet [Bit00b]. Bei der konventionellen Kosinustransformation werden mit diesem Fenster Kosinus-II-Funktionen

$$\theta_j^k(t) := \begin{cases} 1, & \text{falls } k = 0, \\ \sqrt{2} \cos(k(t-j)\pi), & \text{falls } k \geq 1 \end{cases} \quad (2.2)$$

gewichtet. Die so konstruierten Basisfunktionen besitzen eine gute Zeitlokalität. Durch die Verwendung des Rechteckfensters sind die Basisfunktionen jedoch an den Intervallgrenzen nicht stetig und somit im Frequenzbereich schlecht lokalisiert. Die Reproduktion von Signalen mit unstetigen Basisfunktionen kann im rekonstruierten Signal zu Unstetigkeiten an den Intervallgrenzen führen. In der Bilddatenkompression kommt es bei einer zu hohen Quantisierung der Wichtungskoeffizienten zu sichtbaren Blockartefakten. Um diese Probleme zu lösen, werden im Folgenden überlappende lokale trigonometrische Basen

$$\{\psi_j^k\} := \{\psi_j^k : j \in \mathbb{Z}, k \in \mathbb{N}_0\} \quad (2.3)$$

genutzt. Die Basisfunktionen $\psi_j^k \in L^2(\mathbb{R})$ sind frequenzlokale trigonometrische Funktionen θ_j^k , die mit zeitlokalen überlappenden Fensterfunktionen $w_j \in L^2(\mathbb{R})$ gewichtet werden, d.h.

$$\psi_j^k := w_j \theta_j^k \quad (2.4)$$

mit dem Frequenzindex $k \in \mathbb{N}_0$ und dem Intervall- bzw. Blockindex $j \in \mathbb{Z}$. Die Basisfunktionen sind nach steigendem Frequenzparameter k geordnet. Geeignete trigonometrische Funktionen [Bit00b] zur Konstruktion von überlappenden Basisfunktionen sind Kosinus-IV-Funktionen

$$\theta_j^k(t) = C_j^k(t) = \sqrt{2} \cos\left(\left(k + \frac{1}{2}\right)(t-j)\pi\right) \quad (2.5)$$

und Sinus-IV-Funktionen

$$\theta_j^k(t) = S_j^k(t) = \sqrt{2} \sin\left(\left(k + \frac{1}{2}\right)(t-j)\pi\right). \quad (2.6)$$

Aber auch eine Kombination von verschiedenen trigonometrischen Funktionen wie z.B. Kosinus-II-Funktionen und Sinus-II-Funktionen

$$\theta_j^k(t) = D_j^k(t) = \begin{cases} 1, & \text{falls } k = 0, j \text{ gerade,} \\ \sqrt{2} \cos(k(t-j)\pi), & \text{falls } k \geq 1, j \text{ gerade,} \\ \sqrt{2} \sin((k+1)(t-j)\pi), & \text{falls } k \geq 0, j \text{ ungerade} \end{cases} \quad (2.7)$$

ist möglich. Dieses Kapitel konzentriert sich auf die Untersuchung von Kosinus-IV-Basen, die auf der reellen Achse definiert sind. Für die Darstellung von Signalen, die nur auf einem beschränkten Intervall definiert sind, werden in Kapitel 3 Basen auf dem Intervall eingeführt, die aus einer neuartigen Kombination von Kosinus-IV-Funktionen und Kosinus-II-Funktionen bestehen.

Zur Konstruktion von überlappenden trigonometrischen Basisfunktionen ψ_j^k nach Gleichung (2.4) werden den Intervallen $[j, j+1)$ Fensterfunktionen $w_j \in L^2(\mathbb{R})$ mit beschränktem Träger zugeordnet. Die im weiteren Kapitelverlauf betrachteten Fensterfunktionen sollen einen Träger von $\text{supp } w_j \subset [j - (u-1)\frac{1}{2}, j+1 + (u-1)\frac{1}{2}]$ besitzen. Der Parameter $u \in \mathbb{N}$ beschreibt den Überlappungsfaktor und stimmt mit der Länge des Trägers der Fenster- und Basisfunktionen zur Signalrekonstruktion überein. In Abbildung 2.1 sind für einige Überlappungsfaktoren die Fenster schematisiert. Der Parameter $u = 1$ entspricht einer Rechteckfensterung, wie sie bei der konventionellen Kosinustransformation verwendet wird. Weiterhin ist mit $u = 2$ die

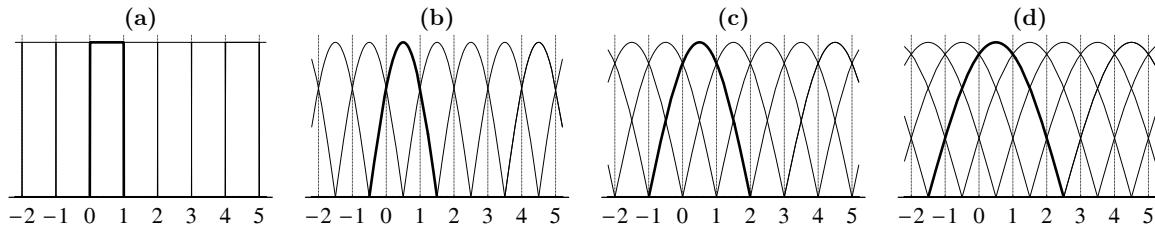


Abbildung 2.1: Überlappungsfaktoren: **a)** Rechteckfensterung, keine Überlappung; **b)** Zweifachüberlappung; **c)** Dreifachüberlappung; **d)** Vierfachüberlappung.

Zweifachüberlappung veranschaulicht. Hier überschneiden sich in der Umgebung der Intervallgrenzen jeweils zwei benachbarte Fensterfunktionen. Der Träger des Fensters verdoppelt sich im Vergleich zum Rechteckfenster. Eine weitere Vergrößerung des Trägers führt zu höheren Überlappungsfaktoren, wie beispielsweise der Dreifach- und Vierfachüberlappung.

2.1.2 Darstellung von Signalen mit überlappenden lokalen trigonometrischen Basisfunktionen

Mit den im vorangegangenen Abschnitt betrachteten lokalen trigonometrischen Basen $\{\psi_j^k\}$ wird das Signal $f \in L^2(\mathbb{R})$ als Überlagerung von gewichteten Basisfunktionen durch

$$f = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} c_{j,k} \psi_j^k = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} c_{j,k} w_j \theta_j^k \quad (2.8)$$

beschrieben. Die Skalare $c_{j,k} \in \mathbb{R}$ sind die Koordinaten des Signals f bezüglich der Darstellung mit lokalen trigonometrischen Basisfunktionen ψ_j^k . Sie werden in der Signal- und Bildverarbeitung auch als Transformationskoeffizienten bezeichnet. In Abbildung 2.2 sind Beispiele für überlappende trigonometrische Basisfunktionen dargestellt. Es handelt sich um Kosinus-IV-Funktionen, die mit einem stetig differenzierbaren Fenster gewichtet wurden. Durch die Wichtung der trigonometrischen Funktionen θ_j^k mit blocküberlappenden Fensterfunktionen w_j überschneiden sich die resultierenden Basisfunktionen ψ_j^k an den Block- bzw. Intervallgrenzen. Die Überlagerung dieser Basisfunktionen ergibt ein Signal, das keine Sprünge aufweist. Durch die überlappenden Fenster- und Basisfunktionen ist die Rekonstruktion eines Intervalles $[j, j+1)$ für sich allein betrachtet unvollständig. Es ist nicht möglich, das Signal durch

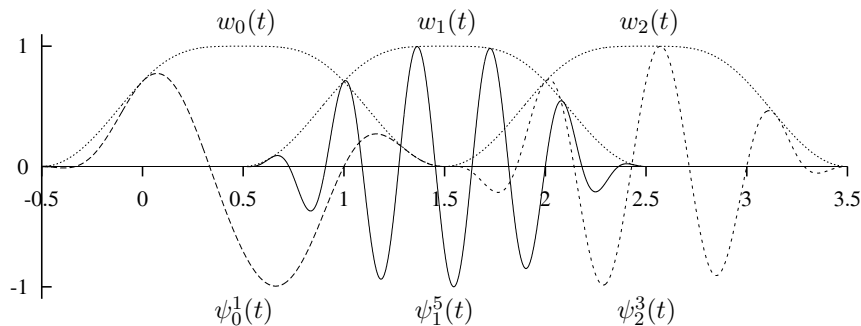


Abbildung 2.2: Überlappende lokale trigonometrische Basisfunktionen.

die Überlagerung der gewichteten Basisfunktionen ψ_j^k eines Intervalls j zu rekonstruieren. Die Informationen über die noch fehlenden Signalanteile eines Blockes sind in den Wichtungskoeffizienten der benachbarten Intervalle enthalten. Erst durch die Blocküberlappung wird die Signalrekonstruktion vollständig.

Die Bestimmung der Wichtungskoeffizienten $c_{j,k}$ zur Darstellung des Signals f mit lokalen trigonometrischen Basen kann mit dem Skalarprodukt

$$c_{j,k} = \langle f, \tilde{\psi}_j^k \rangle \quad (2.9)$$

erfolgen. Die Koeffizientenfunktionalen $\tilde{\psi}_j^k$ werden als duale Basisfunktionen oder Analysebasisfunktionen bezeichnet. Wie die Synthesebasisfunktionen ψ_j^k bestehen auch die dualen Basisfunktionen $\tilde{\psi}_j^k$ aus gefensterten trigonometrischen Funktionen

$$\tilde{\psi}_j^k = \tilde{w}_j \theta_j^k, \quad (2.10)$$

wobei \tilde{w}_j das duale Fenster darstellt. Ist die Basis $\{\psi_j^k\}$ orthonormal, so stimmen die dualen Basisfunktionen $\tilde{\psi}_j^k$ zur Signalanalyse mit den Basisfunktionen ψ_j^k zur Signalrekonstruktion überein. Bei biorthogonalen Basen sind die Fensterfunktionen und somit auch die Basisfunktionen für die Signalanalyse und die Signalrekonstruktion voneinander verschieden und sie können auch unterschiedliche Überlappungsfaktoren aufweisen. Während bei der Zweifachüberlappung automatisch auch die dualen Fensterfunktionen zweifach überlappend sind und somit einen beschränkten Träger aufweisen, ist diese Eigenschaft bei mehrfach überlappenden Fensterfunktionen nicht mehr erfüllt. Für die im Rahmen dieser Arbeit betrachteten mehrfach überlappenden Basisfunktionen erhält man keine dualen Fenster- und Basisfunktionen mit beschränktem Träger. Diese Eigenschaft ist insbesondere bei der Entwicklung schneller Transformationsalgorithmen in Kapitel 4 zu berücksichtigen.

2.2 Klapp- und Aufklappoperatoren

Mit der Einführung von Klapp- und Aufklappoperatoren durch Wickerhauser [Wic93, Wic96] und den Verallgemeinerungen dieser Operatoren in [Swe94, JLS94, JS95, Mat96, CS96, CS97a, CS97b, Bit00a, BC99, Bit98b, Bit99b, Bit00b] stehen sehr leistungsfähige Werkzeuge zur Untersuchung von überlappenden lokalen trigonometrischen Basen zur Verfügung. Die wesentliche Bedeutung von Klapp- und Aufklappoperatoren besteht darin, dass sie die Beziehung zwischen den überlappenden trigonometrischen Basisfunktionen $\psi_j^k = w_j \theta_j^k$, $j \in \mathbb{Z}$, $k \in \mathbb{N}_0$, und den einfachen nicht überlappenden trigonometrischen Funktionen der Orthonormalbasis $\{\chi_{[j,j+1)} \theta_j^k : j \in \mathbb{Z}, k \in \mathbb{N}_0\}$ beschreiben. Aufgrund dieser Zusammenhänge lassen sich Eigenschaften von überlappenden trigonometrischen Basen mit bestimmten Eigenschaften von Klapp- und Aufklappoperatoren identifizieren [Bit00b]. Hierzu gehören unter anderem die Auflösungs- und Rekonstruktionseigenschaften sowie die Riesz-Stabilität. Die Beschreibung von lokalen trigonometrischen Basen mit Klapp- und Aufklappoperatoren erweist sich als äußerst nützlich für die:

- Herleitung von Fensterfunktionen zur Konstruktion von überlappenden lokalen trigonometrischen Basen,
- Bestimmung von dualen Basen,
- Berechnung von Riesz-Konstanten,

- Entwicklung von rechen- und speichereffizienten Algorithmen auf der Grundlage schneller trigonometrischer Blocktransformationen.

2.2.1 Allgemeine Definitionen und Eigenschaften

Die bisher am allgemeinsten gehaltene Definition von Klapp- und Aufklappoperatoren wurde von Bittner in [Bit00b] eingeführt. Sie bildet die Grundlage für die weiteren Untersuchungen im Rahmen dieser Arbeit.

Analog zu den Ausführungen im Abschnitt 2.1.1 wird im Folgenden die Einteilung der Zeitachse in äquidistante Intervalle $[j, j+1)$, $j \in \mathbb{Z}$, betrachtet. Für die dualen überlappenden trigonometrischen Basisfunktionen $\tilde{\psi}_j^k = \tilde{w}_j \theta_j^k$ zur Signalanalyse soll gelten, dass für jedes Signal $f \in L^2(\mathbb{R})$ die Bedingung

$$\sum_{k \in \mathbb{N}_0} |\langle f, \tilde{\psi}_j^k \rangle|^2 < \infty, \quad j \in \mathbb{Z}, \quad \tilde{\psi}_j^k \in L^2(\mathbb{R}) \quad (2.11)$$

erfüllt sei [Bit00b]. Der Klappoperator $\mathcal{T}_{\tilde{w}} : L^2(\mathbb{R}) \rightarrow L_{loc}^2(\mathbb{R})$ wird dann durch

$$f_{\mathcal{T}} = \mathcal{T}_{\tilde{w}} f := \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \langle f, \tilde{\psi}_j^k \rangle \chi_{[j, j+1)} \theta_j^k, \quad f \in L^2(\mathbb{R}) \quad (2.12)$$

definiert [Bit00b]. Das Skalarprodukt $\langle f, \tilde{\psi}_j^k \rangle$ ergibt die Darstellungskoeffizienten $c_{j,k}$ des Signals $f \in L^2(\mathbb{R})$ bezüglich der überlappenden trigonometrischen Basisfunktionen $\tilde{\psi}_j^k = \tilde{w}_j \theta_j^k$. Der Klappoperator in Gleichung (2.12) kann daher auch durch

$$f_{\mathcal{T}} = \mathcal{T}_{\tilde{w}} f = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} c_{j,k} \chi_{[j, j+1)} \theta_j^k \quad (2.13)$$

beschrieben werden. Das geklappte Signal $f_{\mathcal{T}}$ ist somit die Linearkombination der mit den Koeffizienten $c_{j,k}$ gewichteten trigonometrischen Funktionen $\chi_{[j, j+1)} \theta_j^k$. Die Transformationskoeffizienten stellen daher zum einen das Signal $f \in L^2(\mathbb{R})$ bezüglich der überlappenden trigonometrischen Basisfunktionen $\tilde{\psi}_j^k$ dar und zum anderen das geklappte Signal $f_{\mathcal{T}} \in L_{loc}^2(\mathbb{R})$ bezüglich der nicht überlappenden trigonometrischen Funktionen $\chi_{[j, j+1)} \theta_j^k$. Da das System von trigonometrischen Funktionen $\chi_{[j, j+1)} \theta_j^k$, $k \in \mathbb{N}_0$, eine Orthonormalbasis auf dem Intervall $[j, j+1)$ bildet, erhält man den Zusammenhang

$$\int_j^{j+1} \mathcal{T}_{\tilde{w}} f(t) \theta_j^k(t) dt = \int_j^{j+1} f_{\mathcal{T}}(t) \theta_j^k(t) dt = \langle f, \tilde{\psi}_j^k \rangle = \langle \mathcal{T}_{\tilde{w}} f, \chi_{[j, j+1)} \theta_j^k \rangle = c_{j,k}. \quad (2.14)$$

Die Analyse von Signalen mit überlappenden trigonometrischen Basisfunktionen $\tilde{\psi}_j^k$ lässt sich demnach durch zwei nacheinander auszuführende lineare Abbildungen beschreiben. Zunächst werden Signalanteile von f , die auf der Zeitachse im Bereich des Trägers der überlappenden Basisfunktionen $\tilde{\psi}_j^k$ liegen, durch den Klappoperator $\mathcal{T}_{\tilde{w}}$ auf das Intervall $[j, j+1)$ abgebildet. Man spricht hierbei von einem „in das Intervall Klappen der überlappenden Signalanteile“. Anschließend erfolgt durch die Koordinatentransformation des geklappten Signals $f_{\mathcal{T}} = \mathcal{T}_{\tilde{w}} f \in L_{loc}^2(\mathbb{R})$ mit den trigonometrischen Basisfunktionen $\chi_{[j, j+1)} \theta_j^k$ die Bestimmung der Transformationskoeffizienten $c_{j,k}$.

Der zum Klappoperator inverse Operator $\mathcal{T}_{\tilde{w}}^{-1}$ wird als Aufklappoperator \mathcal{U}_w bezeichnet. Mit ihm erhält man aus dem geklappten Signal $f_{\mathcal{T}}$ das ursprüngliche Signal f zurück. Aus der Darstellung des Signals durch Linearkombination der Basisfunktionen

$$f = \mathcal{U}_w f_{\mathcal{T}} = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} c_{j,k} \psi_j^k \quad (2.15)$$

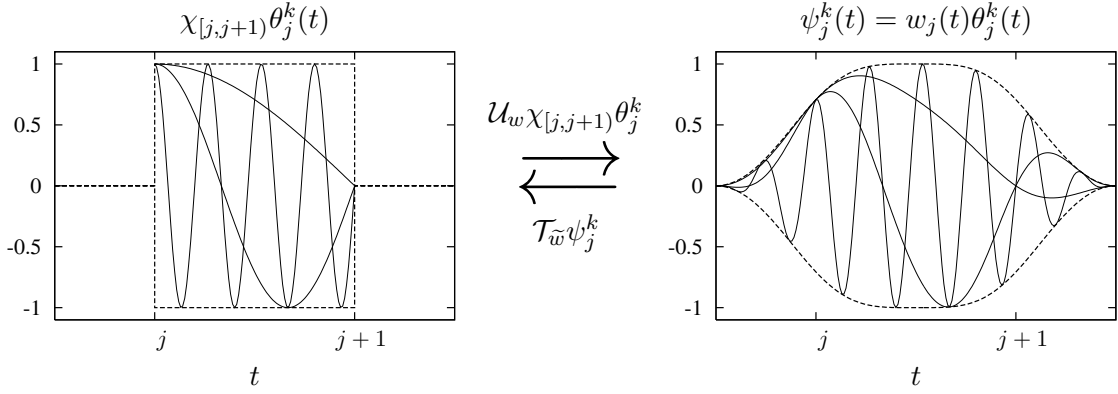


Abbildung 2.3: Beziehung zwischen überlappenden lokalen trigonometrischen Basisfunktionen $\psi_j^k = w_j \theta_j^k$ (rechts) und einfachen orthogonalen trigonometrischen Basisfunktionen $\chi_{[j,j+1)} \theta_j^k$ (links), die durch Klapp- bzw. Aufklappoperatoren beschrieben wird.

erhält man durch Substitution mit Gleichung (2.14) den Aufklappoperator [Bit00b]

$$f = \mathcal{U}_w f_{\mathcal{T}} = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_j^{j+1} f_{\mathcal{T}}(t) \theta_j^k(t) dt \psi_j^k. \quad (2.16)$$

Da $\{\chi_{[j,j+1)} \theta_j^k : j \in \mathbb{Z}, k \in \mathbb{N}_0\}$ eine Orthonormalbasis von $L^2(\mathbb{R})$ darstellt, bildet der Aufklappoperator die Basisfunktionen dieser einfachen trigonometrischen Basis auf die Basisfunktionen der überlappenden trigonometrischen Basis $\{\psi_j^k : j \in \mathbb{Z}, k \in \mathbb{N}_0\}$ ab, d.h.

$$\mathcal{U}_w(\chi_{[j,j+1)} \theta_j^k) = \psi_j^k, \quad \text{wobei } \langle \theta_j^k, \theta_j^l \rangle = \delta_{lk} = \begin{cases} 1 & \text{für } k = l, \\ 0 & \text{für } k \neq l. \end{cases} \quad (2.17)$$

Der Klappoperator $\mathcal{T}_{\tilde{w}}$ bildet hingegen nach Gleichung (2.12) die überlappenden Basisfunktionen $\psi_j^k = w_j \theta_j^k$ auf die trigonometrischen Funktionen $\chi_{[j,j+1)} \theta_j^k$ ab, d.h.

$$\mathcal{T}_{\tilde{w}} \psi_j^k = \chi_{[j,j+1)} \theta_j^k, \quad \text{wobei } \langle \psi_j^k, \tilde{\psi}_j^l \rangle = \delta_{lk} = \begin{cases} 1 & \text{für } k = l, \\ 0 & \text{für } k \neq l. \end{cases} \quad (2.18)$$

Diese Zusammenhänge sind in Abbildung 2.3 veranschaulicht. Klapp- und Aufklappoperatoren beschreiben demzufolge die Beziehung zwischen der überlappenden trigonometrischen Basis $\{\psi_j^k : j \in \mathbb{Z}, k \in \mathbb{N}_0\}$ und der einfachen nicht überlappenden trigonometrischen Orthonormalbasis $\{\chi_{[j,j+1)} \theta_j^k : j \in \mathbb{Z}, k \in \mathbb{N}_0\}$. Diese Operatoren spiegeln somit wesentliche Eigenschaften der überlappenden trigonometrischen Basen $\{\psi_j^k\}$ wider und sind daher ein wichtiges Werkzeug zur Untersuchung der Basiseigenschaften von $\{\psi_j^k\}$.

Riesz-Stabilität

In Abschnitt 1.1 wurden die Begriffe Riesz-Basis und Riesz-Konstanten eingeführt, um bei der Darstellung von Signalen durch die Funktionen ψ_j^k Eigenschaften wie Vollständigkeit, Eindeutigkeit und numerische Stabilität zu beschreiben. Wie in [Bit00b] gezeigt wird, können diese Eigenschaften auch durch Klapp- und Aufklappoperatoren beschrieben werden.

Das System $\{\psi_j^k\}$ mit $\psi_j^k \in L^2(\mathbb{R})$, $j \in \mathbb{Z}$, $k \in \mathbb{N}_0$, ist eine Riesz-Basis mit den Riesz-Konstanten A und B genau dann, wenn der Operator $\mathcal{U}_w : L^2(\mathbb{R}) \rightarrow L^2(\mathbb{R})$ eineindeutig (bijektiv) ist und $0 < A \leq A_0 \leq B_0 \leq B < \infty$ mit

$$A_0 := \|\mathcal{U}_w^{-1}\|_{L^2(\mathbb{R}) \rightarrow L^2(\mathbb{R})}^{-2} \quad \text{und} \quad B_0 := \|\mathcal{U}_w\|_{L^2(\mathbb{R}) \rightarrow L^2(\mathbb{R})}^2 \quad (2.19)$$

erfüllt wird [Bit00b, BC03] (vgl. Abschnitt 1.1.3). Für den Fall $A_0 = B_0 = 1$ sind die Operatoren unitär (normerhaltend). Die trigonometrische Basis $\{\psi_j^k\}$ ist dann eine Orthonormalbasis. Für mehrfach überlappende lokale trigonometrische Basen werden in [Bit00b] Methoden zur Bestimmung der Riesz-Konstanten vorgestellt. Ein wichtiges Werkzeug stellt hierbei die Zak-Transformation [Gel50, Wei64, Zak67, Bit00b] dar:

$$Z_2 f(t, \xi) := \sqrt{2} \sum_{j \in \mathbb{Z}} f(2(t+j)) e^{2\pi i j \xi}, \quad t, \xi \in \mathbb{R}. \quad (2.20)$$

Mit Hilfe der Zak-Transformation kann der Aufklappoperator für mehrfach überlappende Kosinus-IV-Basen als

$$\begin{pmatrix} Z_2 \mathcal{U}_w f_{\mathcal{T}}(t, \xi) \\ Z_2 \mathcal{U}_w f_{\mathcal{T}}(-t, \xi) \end{pmatrix} = \mathbf{M} \left(t, \xi + \frac{1}{2} \right) \begin{pmatrix} Z_2 f_{\mathcal{T}}(t, \xi) \\ Z_2 f_{\mathcal{T}}(-t, \xi) \end{pmatrix}, \quad (t, \xi) \in Q^+ := (0, \frac{1}{2}) \times [-\frac{1}{2}, \frac{1}{2}), \quad (2.21)$$

mit der 2×2 Matrix

$$\mathbf{M}(t, \xi) = \frac{1}{\sqrt{2}} \begin{pmatrix} Z_2 w(t, \xi) & -Z_2 w(\frac{1}{2} + t, \xi) \\ Z_2 w(-t, \xi) & Z_2 w(\frac{1}{2} - t, \xi) \end{pmatrix} \quad (2.22)$$

beschrieben werden [Bit02a]. Die größte untere Riesz-Konstante A_0 und die kleinste obere Riesz-Konstante B_0 lassen sich dann mit den Gleichungen [BC04]

$$A_0 = \inf_{(t, \xi) \in Q^+} \|\mathbf{M}^{-1}(t, \xi)\|_2^{-2} = \inf_{(t, \xi) \in Q^+} \frac{\Delta(t, \xi)}{2} - \sqrt{\frac{\Delta(t, \xi)^2}{4} - |\det \mathbf{M}(t, \xi)|^2}, \quad (2.23)$$

$$B_0 = \sup_{(t, \xi) \in Q^+} \|\mathbf{M}(t, \xi)\|_2^2 = \sup_{(t, \xi) \in Q^+} \frac{\Delta(t, \xi)}{2} + \sqrt{\frac{\Delta(t, \xi)^2}{4} - |\det \mathbf{M}(t, \xi)|^2}, \quad (2.24)$$

bestimmen, wobei

$$\Delta(t, \xi) = \frac{1}{2} (|Z_2 w(t, \xi)|^2 + |Z_2 w(-t, \xi)|^2 + |Z_2 w(\frac{1}{2} + t, \xi)|^2 + |Z_2 w(\frac{1}{2} - t, \xi)|^2). \quad (2.25)$$

Auflösungs- und Rekonstruktionseigenschaften

Bei der Anwendung des Klappoperators auf glatte Signale $f \in L^2(\mathbb{R})$ werden diese in stückweise glatte Signale $f_{\mathcal{T}} \in L_{loc}^2(\mathbb{R})$ transformiert. Für die glatten Teilstücke in den Intervallen $[j, j+1)$ lassen sich nun Aussagen über die Darstellung mittels orthonormaler trigonometrischer Funktionen $\chi_{[j, j+1)} \theta_j^k$ ableiten. Durch die Entwicklung und Nutzung von effizienten Klappoperatoren $\mathcal{T}_{\tilde{w}}$ bzw. dualen Fensterfunktionen \tilde{w}_j soll eine möglichst dünn besetzte Repräsentation des geklappten Signals mit den trigonometrischen Funktionen $\chi_{[j, j+1)} \theta_j^k$ erzielt werden. In der Bilddatenkompression hat sich der Ansatz der verschwindenden Momente als sehr leistungsfähig erwiesen. Das bedeutet, dass Polynome mit möglichst hoher Polynomordnung ausschließlich durch die Basisfunktionen $\tilde{\psi}_j^0$ aufgelöst werden sollen (vgl. Abschnitt 1.1.3). Um geeignete Klappoperatoren bzw. duale Fensterfunktionen zur Signalanalyse zu

entwickeln, ist bei diesem Ansatz für den Klappoperator die Abbildung eines Polynoms der Ordnung m auf die Basisfunktion $\chi_{[j,j+1)}\theta_j^0$ als Entwurfskriterium vorzugeben, d.h.

$$\mathcal{T}_{\tilde{w}}t^m = \sum_{j \in \mathbb{Z}} c_{j,0}\chi_{[j,j+1)}(t)\theta_j^0(t). \quad (2.26)$$

Betrachtet man die Entwicklung der Fensterfunktionen auf der Rekonstruktionsseite, so ist die Bedingung

$$t^m = \sum_{j \in \mathbb{Z}} \mathcal{U}_w(c_{j,0}\chi_{[j,j+1)}(t)\theta_j^0(t)) = \sum_{j \in \mathbb{Z}} c_{j,0}w_j(t)\theta_j^0(t) \quad (2.27)$$

zu erfüllen.

2.2.2 Transformation von Signalen

Durch die Verwendung von Klapp- und Aufklappoperatoren wird die Transformation von Signalen mit überlappenden trigonometrischen Basen in zwei nacheinander auszuführende Abbildungen zerlegt. Bei der Signalanalyse wird nach Gleichung (2.14) in einem ersten Teilschritt das Signal mit dem Klappoperator $\mathcal{T}_{\tilde{w}}$ transformiert. Anschließend erfolgt mit der ortho-normalen trigonometrischen Blocktransformation \mathbf{C}_θ des geklappten Signals die Bestimmung der Transformationskoeffizienten. Auf der Syntheseseite wird zunächst durch die Blocktransformation \mathbf{C}_θ^T das geklappte Signal rekonstruiert. Anschließend kann durch die Anwendung des Aufklappoperators \mathcal{U}_w das ursprüngliche Signal zurückgewonnen werden. Die Äquivalenz zwischen

- der Transformation mit überlappenden lokalen trigonometrischen Basisfunktionen und
- der Transformation mit Klapp- und Aufklappoperatoren sowie trigonometrischen Blocktransformationen

ist in Abbildung 2.4 dargestellt. Bevor sich die nachfolgenden Abschnitte mit der Herleitung von Klapp- und Aufklappoperatoren für mehrfach überlappende Kosinus-IV-Basen befassen,

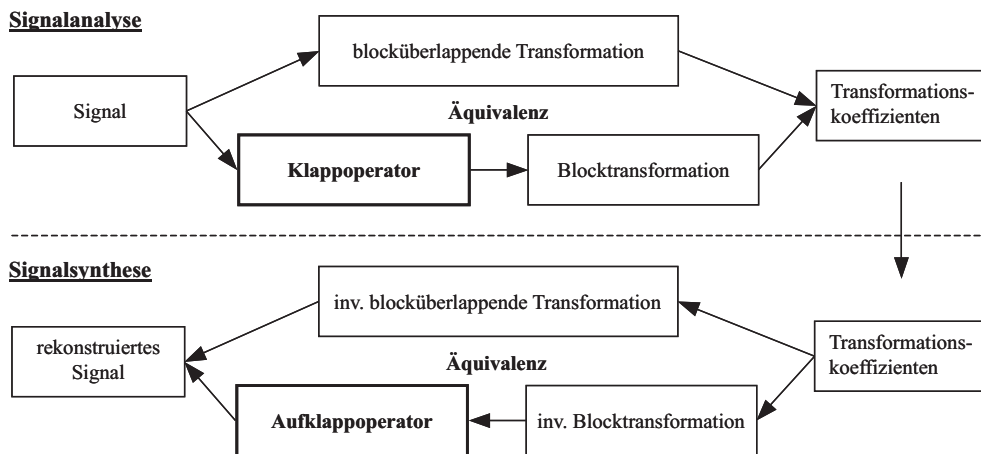


Abbildung 2.4: Äquivalenz zwischen der Transformation mit überlappenden lokalen trigonometrischen Basen und der Transformation unter Verwendung von Klapp- bzw. Aufklappoperatoren sowie trigonometrischen Blocktransformationen.

soll zunächst die prinzipielle Funktionsweise von Klappoperatoren am Beispiel einer zweifach überlappenden orthogonalen Kosinus-IV-Basis unter Verwendung des von Wickerhauser in [Wic93, Wic96] eingeführten Klappoperators veranschaulicht werden.

Für die Transformation von Signalen $f \in L^2(\mathbb{R})$ mit Klappoperatoren wird die Zeitachse in äquidistante Intervalle $[j, j+1)$, $j \in \mathbb{Z}$, eingeteilt. Der Überlappungsradius benachbarter Fensterfunktionen bzw. Signalanteile soll für die Zweifachüberlappung $\epsilon_j = \frac{1}{2}$ betragen. Wickerhausers Klappoperator ist dann durch

$$\mathcal{T}f(j+t) := \begin{cases} r(j+t)f(j+t) + r(j-t)f(j-t), & \text{für } t > 0, \\ r(j-t)f(j+t) - r(j+t)f(j-t), & \text{für } t < 0 \end{cases} \quad (2.28)$$

definiert [Wic93]. Die Funktion $r(t)$ stellt die Abschneidefunktion

$$r(t) := \begin{cases} 0, & \text{für } t \leq j - \frac{1}{2}, \\ w(t-j), & \text{für } j - \frac{1}{2} < t < j + \frac{1}{2}, \\ 1, & \text{für } t \geq j + \frac{1}{2}, \end{cases} \quad \text{mit } |r(j+t)|^2 + |r(j-t)|^2 = 1 \quad (2.29)$$

dar. Bei der Klappoperation wird das Signal $f(t) \in L^2(\mathbb{R})$ mit der Abschneidefunktion $r(t)$ multipliziert und das Intervall $(-\infty, j)$ durch Spiegelung auf das Intervall $(j, +\infty)$ geklappt. Anschließend bildet man die Summe der beiden Anteile. Das Ergebnis ist das geklappte Signal $\mathcal{T}f(t)$ für $t > j$. Um $\mathcal{T}f(t)$ für $t < j$ zu erhalten, wird das Signal $f(t)$ mit der gespiegelten Abschneidefunktion $r(2j-t)$ multipliziert. Dann erfolgt das Klappen des Intervalls (j, ∞) auf das Intervall $(-\infty, j)$ und die Bildung der Differenz. Die einzelnen Verarbeitungsschritte sind in Abbildung 2.5 am Beispiel einer Abschneidefunktion veranschaulicht, die sich aus dem zweifach überlappenden Malvar-Fenster ergibt (vgl. Abschnitt 2.3.1). In Abbildung 2.6 ist zu erkennen, dass mit der verwendeten Abschneidefunktion bei der Verarbeitung einer Konstanten in den Blockintervallen jeweils die Basisfunktion $\chi_{[j,j+1)}\theta_j^0$ der niedrigsten Frequenz der Kosinus-IV-Transformation entsteht (vgl. Abschnitt 1.2.1). Die sich anschließende Kosinus-IV-Transformation des geklappten Signals ergibt daher für jedes Intervall Wichtungskoeffizienten $c_{j,0} \neq 0$ und $c_{j,k>0} = 0$. Für die Rekonstruktion einer Konstanten werden demzufolge bei der Verwendung des Malvar-Fensters nur die Basisfunktionen ψ_j^0 benötigt.

Anhand des dargestellten Beispiels wird eine wichtige Eigenschaft von Klappoperatoren deutlich. Durch die Wahl eines geeigneten Klappoperators bzw. durch die Wahl einer geeigneten Fensterfunktion ist eine effiziente Anpassung des Signals an die nachfolgende trigonometrische Blocktransformation und somit eine sparsame Signalrepräsentation möglich. Die Bedingung $|r(j+t)|^2 + |r(j-t)|^2 = 1$ in Gleichung (2.29) schränkt jedoch die Anzahl geeigneter Fensterfunktionen stark ein. In verschiedenen Arbeiten [Swe94, JLS94, JS95, Mat96, CS96, CS97a, CS97b, Bit00a, BC99, Bit98b, Bit99b, Bit00b] wurden daher allgemeinere Klappoperatoren zur Beschreibung von überlappenden lokalen trigonometrischen Basen eingeführt. Sie bieten mehr Freiheiten bei der Konstruktion von überlappenden trigonometrischen Basisfunktionen.

2.2.3 Klappoperatoren für Kosinus-IV-Basen

Die Gleichungen (2.21) und (2.22) beschreiben bereits den Aufklappoperator für Kosinus-IV-Basen im Zak-Transformationsbereich. Durch die Bestimmung der inversen Matrix aus Gleichung (2.22) lässt sich sofort eine zu Gleichung (2.21) äquivalente Matrix-Vektor-Beschreibung des Klappoperators im Zak-Transformationsbereich finden. Die Anwendung der inversen Zak-Transformation überführt die Zak-Transformierte des geklappten Signals in den Zeit- bzw. Ortsbereich. Bei dieser Betrachtungsweise von Klapp- und Aufklappoperatoren wird vorausgesetzt, dass allen Teilintervallen Kosinus-IV-Funktionen und Translate einer einzigen dualen

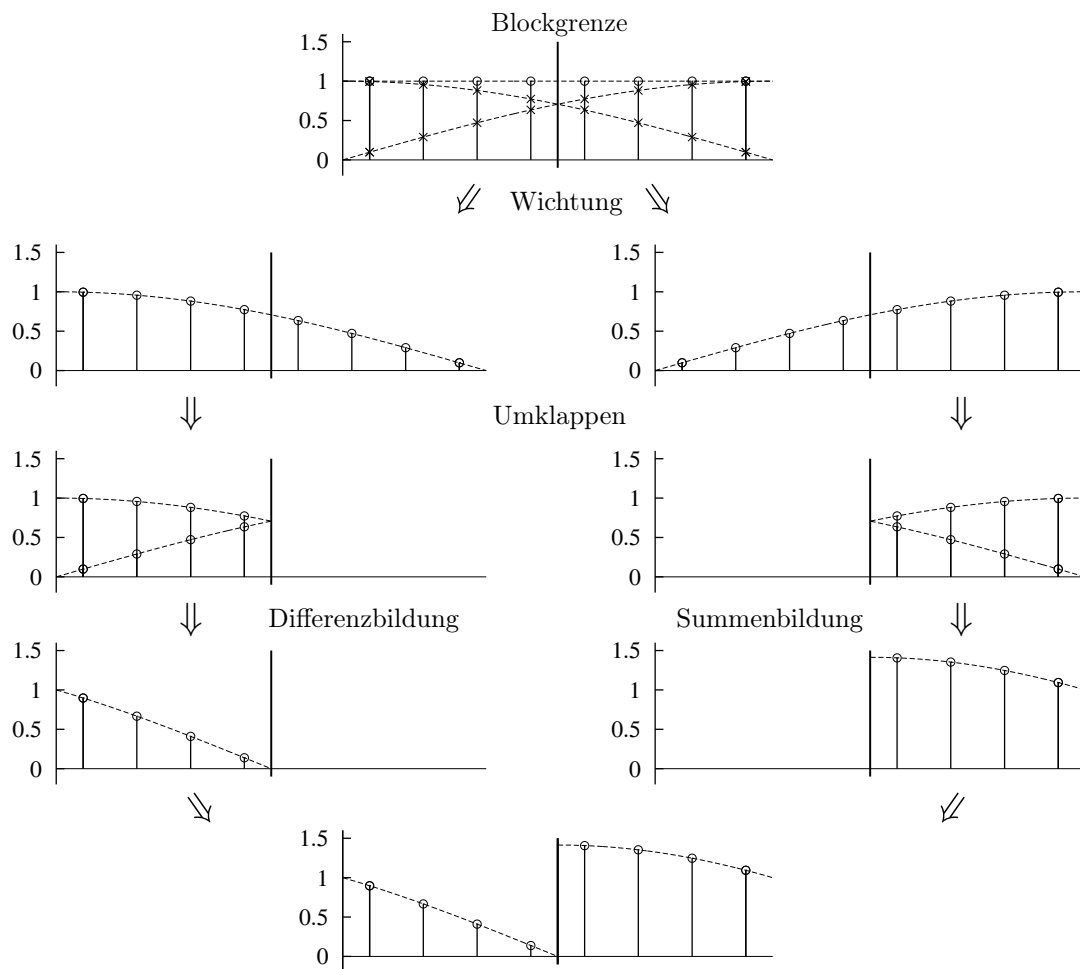


Abbildung 2.5: Klappoperation am Beispiel des Klappoperators von Wickerhauser für die zweifach überlappende orthogonale trigonometrische Malvar-Basis.

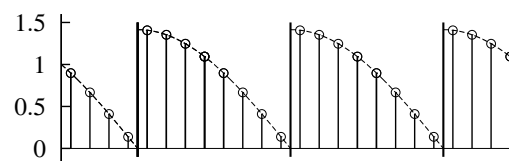


Abbildung 2.6: Anwendung des Klappoperators auf eine Konstante an den Blockgrenzen mehrerer aufeinanderfolgender Intervalle unter Verwendung des Malvar-Fensters.

Fensterfunktion zugeordnet sind. Diese Bedingung ist jedoch bei der Darstellung von endlich langen Signalen mit dem System von Basisfunktionen, das in Kapitel 3 neu eingeführt wird, nicht erfüllt. Im Folgenden werden daher als wichtige Grundlage für die Untersuchungen in Kapitel 3 die Klapp- und Aufklappoperatoren für mehrfach überlappende Kosinus-IV-Basen im Zeitbereich hergeleitet.

Ausgangspunkt für die Herleitung von Klappoperatoren für mehrfach überlappende Kosinus-IV-Basen ist die verallgemeinerte Definition nach Gleichung (2.13), wobei sich die Darstellungskoeffizienten $c_{j,k}$ aus dem Skalarprodukt zwischen dem zu klappenden Signal $f \in L^2(\mathbb{R})$ und den dualen Basisfunktionen $\tilde{\psi}_j^k \in L^2(\mathbb{R})$ zur Signalanalyse ergeben, d.h.

$$c_{j,k} = \langle f, \tilde{\psi}_j^k \rangle = \langle f, \tilde{w}_j \theta_j^k \rangle = \int_{\mathbb{R}} f(\tau) \tilde{w}_j(\tau) \theta_j^k(\tau) d\tau, \quad j \in \mathbb{Z}, k \in \mathbb{N}_0. \quad (2.30)$$

Die trigonometrischen Funktionen θ_j^k sind nachfolgend die Kosinus-IV-Funktionen

$$\theta_j^k(t) = \sqrt{2} \cos\left((k + \tfrac{1}{2})\pi(t - j)\right). \quad (2.31)$$

Sie sind antiperiodisch mit der Periodenlänge $\lambda = 2$, d.h.

$$\theta_j^k(t) = -\theta_j^k(t - 2). \quad (2.32)$$

Bei der Einteilung des Signals in Intervalle der Länge $N_j = 1$ und der Verarbeitung mit antiperiodischen Basisfunktionen der Periodenlänge $\lambda = 2$ erweist es sich als nützlich, das System $\{\tilde{\psi}_j^k : j \in \mathbb{Z}, k \in \mathbb{N}_0\}$ mit den Basisfunktionen

$$\tilde{\psi}_{2j} = \tilde{w}_{2j} \theta_{2j}^k \quad \text{und} \quad \tilde{\psi}_{2j-1} = \tilde{w}_{2j-1} \theta_{2j-1}^k \quad (2.33)$$

sowie die geklappten Teilsignale $\mathcal{T}_{\tilde{w}} f(2j + t)$ und $\mathcal{T}_{\tilde{w}} f(2j - t)$ mit $t \in [0, 1)$ zu betrachten. Hierfür wird das zu transformierende Signal in Intervalle $[2j - 1, 2j)$ und $[2j, 2j + 1)$ eingeteilt. Bei der Herleitung des Klappoperators werden des Weiteren die für Kosinus-IV-Funktionen gültigen Beziehungen

$$\begin{aligned} \theta_{2j}^k(t) &= (-1)^j \theta_0^k(t), & \theta_0^k(-t) &= \theta_0^k(t), \\ \theta_{2j-1}^k(t) &= (-1)^j \theta_{-1}^k(t), & \theta_{-1}^k(-t) &= -\theta_{-1}^k(t) \end{aligned} \quad (2.34)$$

ausgenutzt. Mit Hilfe der dargestellten Eigenschaften lassen sich nun für Kosinus-IV-Basen aus der allgemeinen Definition des Klappoperators in Gleichung (2.13) und aus der Beschreibung der Darstellungskoeffizienten in Gleichung (2.30) sehr gut handhabbare Gleichungen zur Realisierung von Klappoperatoren herleiten. Für $t \in [0, 1)$ kann man den Klappoperator durch

$$\begin{aligned} \mathcal{T}_{\tilde{w}} f(2j + t) &= \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r + \tau) \tilde{w}_{2j}(2r + \tau) \theta_{2j}^k(2r + \tau) d\tau \theta_{2j}^k(2j + t) \\ &\quad + \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_{-1}^0 f(2r + \tau) \tilde{w}_{2j}(2r + \tau) \theta_{2j}^k(2r + \tau) d\tau \theta_{2j}^k(2j + t), \end{aligned} \quad (2.35)$$

$$\begin{aligned} \mathcal{T}_{\tilde{w}} f(2j - t) &= \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r - \tau) \tilde{w}_{2j-1}(2r - \tau) \theta_{2j-1}^k(2r - \tau) d\tau \theta_{2j-1}^k(2j - t) \\ &\quad + \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_{-1}^0 f(2r - \tau) \tilde{w}_{2j-1}(2r - \tau) \theta_{2j-1}^k(2r - \tau) d\tau \theta_{2j-1}^k(2j - t) \end{aligned} \quad (2.36)$$

beschreiben. Aufgrund der in Gleichung (2.34) angegebenen Beziehungen gelten für die trigonometrischen Funktionen die Zusammenhänge

$$\begin{aligned} \theta_{2j}^k(2r + t) &= (-1)^{r+j} \theta_0^k(t), & \theta_{2j}^k(2j + t) &= \theta_0^k(t), \\ \theta_{2j-1}^k(2r - t) &= (-1)^{r+j} \theta_{-1}^k(-t), & \theta_{2j-1}^k(2j - t) &= \theta_{-1}^k(-t). \end{aligned} \quad (2.37)$$

Durch Einsetzen in die Gleichungen (2.35) und (2.36) erhält man

$$\begin{aligned} \mathcal{T}_{\tilde{w}} f(2j+t) &= \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r+\tau) \tilde{w}_{2j}(2r+\tau) (-1)^{r+j} \theta_0^k(\tau) d\tau \theta_0^k(t) \\ &\quad + \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_{-1}^0 f(2r+\tau) \tilde{w}_{2j}(2r+\tau) (-1)^{r+j} \theta_0^k(\tau) d\tau \theta_0^k(t), \end{aligned} \quad (2.38)$$

$$\begin{aligned} \mathcal{T}_{\tilde{w}} f(2j-t) &= \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r-\tau) \tilde{w}_{2j-1}(2r-\tau) (-1)^{r+j} \theta_{-1}^k(-\tau) d\tau \theta_{-1}^k(-t) \\ &\quad + \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_{-1}^0 f(2r-\tau) \tilde{w}_{2j-1}(2r-\tau) (-1)^{r+j} \theta_{-1}^k(-\tau) d\tau \theta_{-1}^k(-t). \end{aligned} \quad (2.39)$$

Mit dem Zusammenhang

$$\int_{-1}^0 f(a+t) dt = \int_0^1 f(a-t) dt \quad (2.40)$$

ergeben sich für $t \in [0, 1)$ und $j \in \mathbb{Z}$ die Gleichungen

$$\begin{aligned} \mathcal{T}_{\tilde{w}} f(2j+t) &= \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r+\tau) \tilde{w}_{2j}(2r+\tau) (-1)^{r+j} \theta_0^k(\tau) d\tau \theta_0^k(t) \\ &\quad + \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r-\tau) \tilde{w}_{2j}(2r-\tau) (-1)^{r+j} \theta_0^k(-\tau) d\tau \theta_0^k(t), \end{aligned} \quad (2.41)$$

$$\begin{aligned} \mathcal{T}_{\tilde{w}} f(2j-t) &= \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r-\tau) \tilde{w}_{2j-1}(2r-\tau) (-1)^{r+j} \theta_{-1}^k(-\tau) d\tau \theta_{-1}^k(-t) \\ &\quad + \sum_{r \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r+\tau) \tilde{w}_{2j-1}(2r+\tau) (-1)^{r+j} \theta_{-1}^k(\tau) d\tau \theta_{-1}^k(-t). \end{aligned} \quad (2.42)$$

Um die Integrale zu eliminieren, wird zum Abschluss der Umformungen die Beziehung

$$\sum_{k \in \mathbb{N}_0} \int_0^1 f(\tau) \tilde{w}(\tau) \theta^k(\tau) d\tau \theta^k(t) = \sum_{k \in \mathbb{N}_0} \langle f(t) \tilde{w}(t), \theta^k(t) \rangle \theta^k(t) = f(t) \tilde{w}(t), \quad \text{für } t \in [0, 1) \quad (2.43)$$

ausgenutzt, wobei die Funktionen θ^k , $k \in \mathbb{N}_0$, eine Orthonormalbasis von $L^2([0, 1])$ bilden. Man erhält somit aus den Gleichungen (2.41) und (2.42) die geklappten Teilsignale

$$\mathcal{T}_{\tilde{w}} f(2j+t) = \sum_{r \in \mathbb{Z}} (-1)^{r+j} (\tilde{w}_{2j}(2r+t) f(2r+t) + \tilde{w}_{2j}(2r-t) f(2r-t)), \quad (2.44)$$

$$\mathcal{T}_{\tilde{w}} f(2j-t) = \sum_{r \in \mathbb{Z}} (-1)^{r+j} (\tilde{w}_{2j-1}(2r-t) f(2r-t) - \tilde{w}_{2j-1}(2r+t) f(2r+t)). \quad (2.45)$$

Diese Beziehungen lassen sich zu einer verallgemeinerten Matrix-Vektor-Beschreibung

$$\begin{pmatrix} \mathcal{T}_{\tilde{w}} f(2j+t) \\ \mathcal{T}_{\tilde{w}} f(2j-t) \end{pmatrix} = \sum_{r \in \mathbb{Z}} (-1)^{r+j} \begin{pmatrix} \tilde{w}_{2j}(2r+t) & \tilde{w}_{2j}(2r-t) \\ -\tilde{w}_{2j-1}(2r+t) & \tilde{w}_{2j-1}(2r-t) \end{pmatrix} \begin{pmatrix} f(2r+t) \\ f(2r-t) \end{pmatrix} \quad (2.46)$$

zusammenfassen, die in Abbildung A.1 (Anhang A) dargestellt ist. Es sei darauf hingewiesen, dass die Einträge der Abbildungsmatrix des Klappoperators Funktionen von $t \in [0, 1)$ sind.

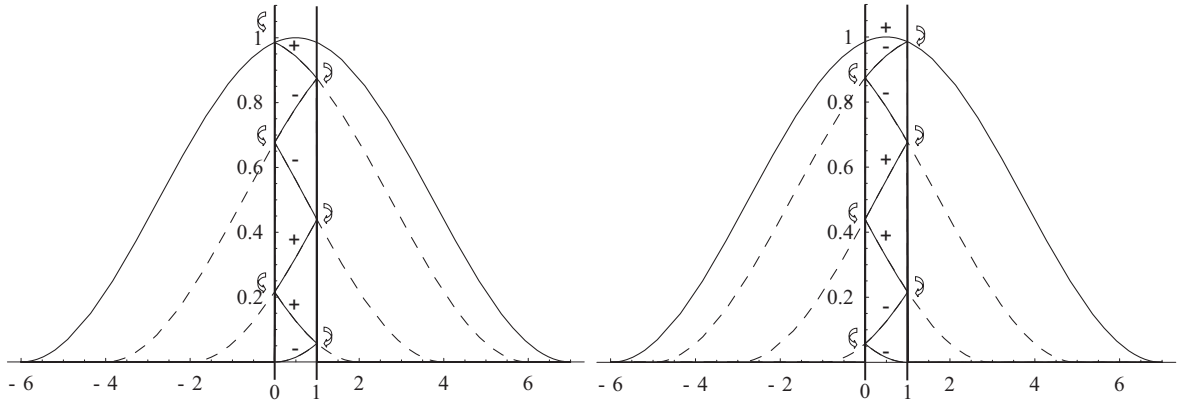


Abbildung 2.8: In das Intervall Klappen der überlappenden Signalanteile, die sich aus der Wichtung des Signals mit dem dualen Fenster ergeben; Klappvorgang an der linken Intervallgrenze (links) und der rechten Intervallgrenze (rechts).

Nimmt man in Analogie zur Beschreibung im Zak-Transformationsbereich nach Gleichung (2.21) an, dass die Fensterfunktionen \tilde{w}_{2j} und \tilde{w}_{2j-1} Translate eines Fensters \tilde{w} sind, so erhält man aus Gleichung (2.46) für das geklappte Signal $\mathcal{T}_{\tilde{w}}f(t)$, $j \in \mathbb{Z}$, $t \in [0, 1)$, den Ausdruck

$$\begin{pmatrix} \mathcal{T}_{\tilde{w}}f(2j+t) \\ \mathcal{T}_{\tilde{w}}f(2j-t) \end{pmatrix} = \sum_{r \in \mathbb{Z}} (-1)^r \begin{pmatrix} \tilde{w}(2r+t) & \tilde{w}(2r-t) \\ -\tilde{w}(2r+1+t) & \tilde{w}(2r+1-t) \end{pmatrix} \begin{pmatrix} f(2j+2r+t) \\ f(2j+2r-t) \end{pmatrix}. \quad (2.47)$$

Der Parameter r wurde bei den Umformungen durch $r+j$ substituiert und die Beziehung $(-1)^{r+2j} = (-1)^r$ ausgenutzt.

In Abbildung 2.7 ist zunächst Gleichung (2.47) als verallgemeinerte Matrix-Vektor-Beschreibung veranschaulicht. Abbildung 2.8 demonstriert schließlich den Klappvorgang, wie er sich aus den vorangegangenen Herleitungen bzw. aus Abbildung 2.7 ergibt. Ähnlich wie beim klassischen Klappoperator von Wickerhauser wird das mit dem Fenster gewichtete Signal $\tilde{w}_j f$ an den Intervallgrenzen umgeklappt. Bei mehrfach überlappenden trigonometrischen Basen wird wiederholt umgeklappt. Durch Addition und Subtraktion der einzelnen Anteile erhält man für das betrachtete Intervall $[j, j+1)$ das in dieses Intervall geklappte Signal $\mathcal{T}_{\tilde{w}_j}f$.

Wie sich in Abschnitt 2.3 bei der Einführung der im Rahmen dieser Arbeit untersuchten Fensterfunktionen zur Konstruktion von mehrfach überlappenden trigonometrischen Basen zeigen wird, besitzen die dualen Fensterfunktionen \tilde{w} zur Signalanalyse im Gegensatz zu den Fensterfunktionen w für die Signalrekonstruktion keinen beschränkten Träger. Das hat zur Folge, dass die Abbildungsmatrizen des Klappoperators bei der Mehrfachüberlappung voll besetzt sind. Die Multiplikation mit voll besetzten Matrizen bedeutet einen hohen Rechenaufwand und ist zu vermeiden. Zur Lösung dieses Problems sind Aufklappoperatoren, die im nachfolgenden Abschnitt für Kosinus-IV-Basen hergeleitet werden, von wesentlicher Bedeutung.

2.2.4 Aufklappoperatoren für Kosinus-IV-Basen

Um zunächst einen Einblick in die Funktionsweise des Aufklappoperators für mehrfach überlappende Kosinus-IV-Basen zu geben, ist in Abbildung 2.9 das Prinzip der Rekonstruktion eines Signals $f \in L^2(\mathbb{R})$ aus dem geklappten Signal $f_{\mathcal{T}} \in L^2_{loc}(\mathbb{R})$ dargestellt. Das geklappte Signal $f_{\mathcal{T}}$ besteht aus den stückweise stetigen Funktionen $\chi_{[j,j+1)}h_j$. Sie resultieren in dem jeweilig betrachteten Intervall aus der Linearkombination der mit den Transformationskoeffi-

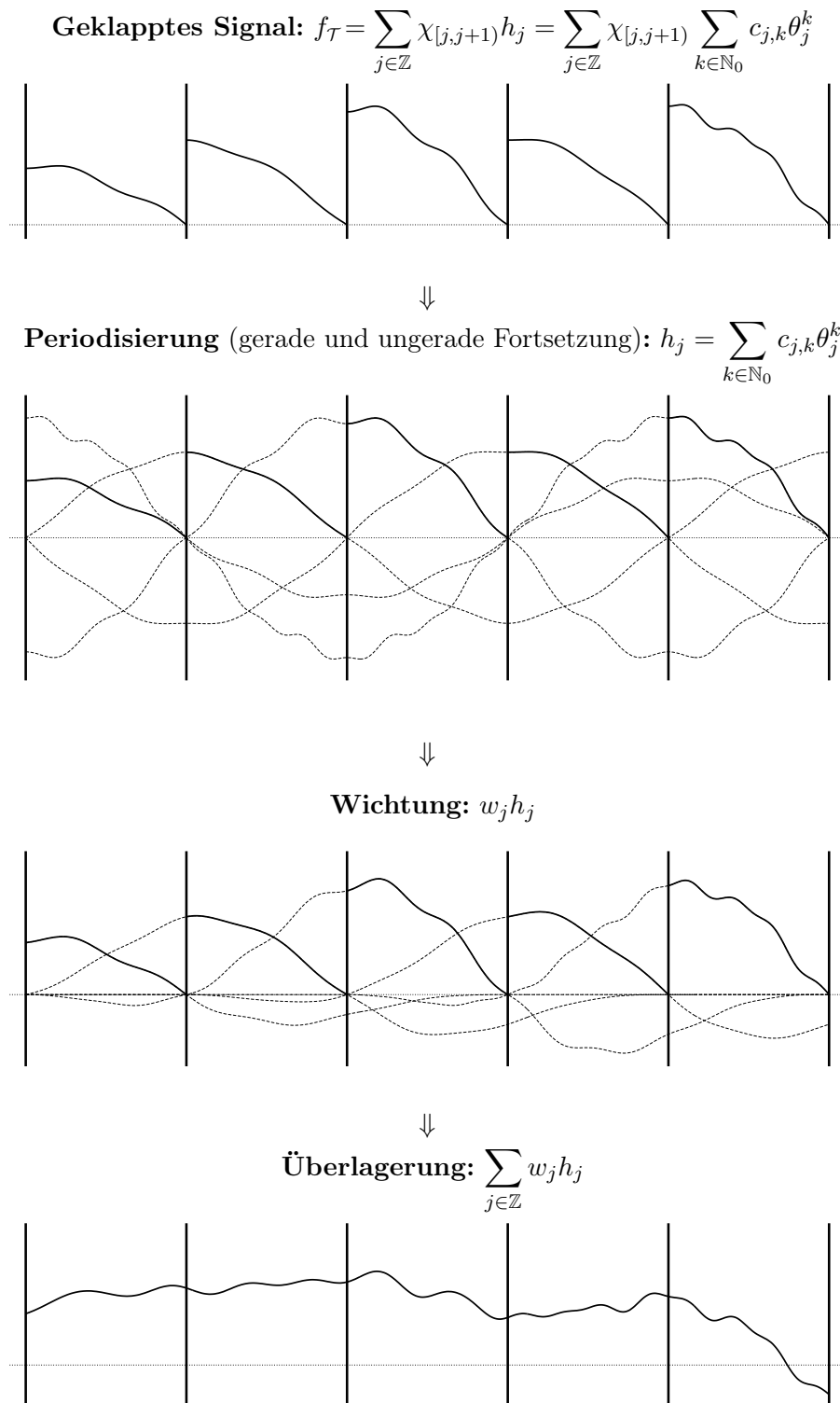


Abbildung 2.9: Veranschaulichung der prinzipiellen Funktionsweise des Aufklappoperators.

zienten $c_{j,k}$ gewichteten orthonormalen Basisfunktionen $\chi_{[j,j+1)}\theta_j^k$, $j \in \mathbb{Z}$, $k \in \mathbb{N}_0$, d.h.

$$f_{\mathcal{T}}(t) = \sum_{j \in \mathbb{Z}} \chi_{[j,j+1)} h_j = \sum_{j \in \mathbb{Z}} \chi_{[j,j+1)} \sum_{k \in \mathbb{N}_0} c_{j,k} \theta_j^k, \quad (2.48)$$

wobei

$$h_j = \sum_{k \in \mathbb{N}_0} c_{j,k} \theta_j^k. \quad (2.49)$$

Betrachtet man die Rekonstruktion eines Signals f als Linearkombination von gewichteten überlappenden Basisfunktionen ψ_j^k , so ergeben sich für den Aufklappoperator folgende Beziehungen:

$$\begin{aligned} \mathcal{U}_w f_{\mathcal{T}} &= \mathcal{U}_w \left(\sum_{j \in \mathbb{Z}} \chi_{[j,j+1)} h_j \right) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} c_{j,k} \psi_j^k, \\ &= \sum_{j \in \mathbb{Z}} w_j \sum_{k \in \mathbb{N}_0} c_{j,k} \theta_j^k = \sum_{j \in \mathbb{Z}} w_j h_j. \end{aligned} \quad (2.50)$$

Das aufgeklappte Signal $\mathcal{U}_w f_{\mathcal{T}}$ erhält man durch Überlagerung aller gewichteter Funktionen $w_j h_j$. Um in einem ersten Schritt die Funktionen h_j zu ermitteln, werden die stückweisen Funktionen $\chi_{[j,j+1)} h_j$ an den Intervallgrenzen gerade oder ungerade fortgesetzt. Anschließend sind die Funktionen h_j mit den Fensterfunktionen w_j zu wichten und aufzusummieren. Um diese Funktionsweise in gut handhabbare Gleichungen zu überführen, erweist es sich in Analogie zum Klappoperator als hilfreich, für die Basis $\{\psi_j^k : j \in \mathbb{Z}, k \in \mathbb{N}_0\}$ vom Typ Kosinus-IV die Basisfunktionen

$$\psi_{2j} = w_{2j} \theta_{2j}^k \quad \text{und} \quad \psi_{2j-1} = w_{2j-1} \theta_{2j-1}^k \quad (2.51)$$

sowie die aufgeklappten Teilsignale $\mathcal{U}_w f_{\mathcal{T}}(2r+t)$ und $\mathcal{U}_w f_{\mathcal{T}}(2r-t)$ mit $t \in [0, 1)$ zu betrachten. Der Aufklappoperator kann daher nach Gleichung (2.50) durch

$$\mathcal{U}_w f_{\mathcal{T}}(2r+t) = \sum_{j \in \mathbb{Z}} w_{2j}(2r+t) h_{2j}(2r+t) + w_{2j-1}(2r+t) h_{2j-1}(2r+t), \quad t \in [0, 1), \quad (2.52)$$

$$\mathcal{U}_w f_{\mathcal{T}}(2r-t) = \sum_{j \in \mathbb{Z}} w_{2j}(2r-t) h_{2j}(2r-t) + w_{2j-1}(2r-t) h_{2j-1}(2r-t), \quad t \in [0, 1) \quad (2.53)$$

beschrieben werden. Die Funktionen h_j sind Linearkombinationen von gewichteten Kosinus-IV-Funktionen. Sie ergeben sich, wie in Abbildung 2.9 dargestellt, durch gerades oder ungerades Fortsetzen (Periodisierung) der Funktionen $\chi_{[j,j+1)} h_j = \chi_{[j,j+1)} f_{\mathcal{T}}$ über die Intervallgrenzen hinaus. Aufgrund der Symmetrie- und Periodizitätseigenschaften von Kosinus-IV-Funktionen erhält man für $t \in [0, 1)$ die Beziehungen

$$\begin{aligned} h_{2j}(2r+t) &= (-1)^{j+r} f_{\mathcal{T}}(2j+t), & h_{2j-1}(2r+t) &= -(-1)^{j+r} f_{\mathcal{T}}(2j-t), \\ h_{2j}(2r-t) &= (-1)^{j+r} f_{\mathcal{T}}(2j+t), & h_{2j-1}(2r-t) &= (-1)^{j+r} f_{\mathcal{T}}(2j-t). \end{aligned} \quad (2.54)$$

Durch Einsetzen in die Gleichungen (2.52) und (2.53) kann der Aufklappoperator für $t \in [0, 1)$ durch

$$\mathcal{U}_w f_{\mathcal{T}}(2r+t) = \sum_{j \in \mathbb{Z}} (-1)^{j+r} (w_{2j}(2r+t) f_{\mathcal{T}}(2j+t) - w_{2j-1}(2r+t) f_{\mathcal{T}}(2j-t)), \quad (2.55)$$

$$\mathcal{U}_w f_{\mathcal{T}}(2r - t) = \sum_{j \in \mathbb{Z}} (-1)^{j+r} (w_{2j}(2r - t) f_{\mathcal{T}}(2j + t) + w_{2j-1}(2r - t) f_{\mathcal{T}}(2j - t)) \quad (2.56)$$

beschrieben werden. Diese Beziehungen lassen sich zu der verallgemeinerten Matrix-Vektor-Darstellung

$$\begin{pmatrix} \mathcal{U}_w f_{\mathcal{T}}(2r + t) \\ \mathcal{U}_w f_{\mathcal{T}}(2r - t) \end{pmatrix} = \sum_{j \in \mathbb{Z}} (-1)^{j+r} \begin{pmatrix} w_{2j}(2r + t) & -w_{2j-1}(2r + t) \\ w_{2j}(2r - t) & w_{2j-1}(2r - t) \end{pmatrix} \begin{pmatrix} f_{\mathcal{T}}(2j + t) \\ f_{\mathcal{T}}(2j - t) \end{pmatrix} \quad (2.57)$$

zusammenfassen, die in Abbildung A.2 (Anhang A) veranschaulicht ist. Sind die Fensterfunktionen w_{2j} und w_{2j-1} Translate einer Fensterfunktion w und wird j mit $r - j$ substituiert, wobei $(-1)^{2r-j} = (-1)^j$, so ergibt sich für das aufgeklappte Signal $\mathcal{U}_w f$ die in Abbildung 2.10 dargestellte verallgemeinerte Matrix-Vektor-Beschreibung

$$\begin{pmatrix} \mathcal{U}_w f_{\mathcal{T}}(2r + t) \\ \mathcal{U}_w f_{\mathcal{T}}(2r - t) \end{pmatrix} = \sum_{j \in \mathbb{Z}} (-1)^j \begin{pmatrix} w(2j + t) & -w(2j + 1 + t) \\ w(2j - t) & w(2j + 1 - t) \end{pmatrix} \begin{pmatrix} f_{\mathcal{T}}(2r - 2j + t) \\ f_{\mathcal{T}}(2r - 2j - t) \end{pmatrix}, \quad (2.58)$$

wobei $t \in [0, 1)$. Für die im Rahmen dieser Arbeit untersuchten mehrfach überlappenden trigonometrischen Basen besitzen die Fensterfunktionen w zur Signalrekonstruktion im Gegensatz zu den dualen Analysefenstern \tilde{w} einen beschränkten Träger. Die Abbildungsmatrizen des Aufklappoperators sind daher nicht wie beim Klappoperator voll besetzt, sondern weisen eine Bandstruktur auf. Die Bandbreite hängt direkt von der Trägerlänge des Rekonstruktionsfensters ab. So besitzen z.B. die dreifach überlappenden Fenster mit $\text{supp } w \subset [-1, 2]$ eine Trägerlänge von drei. Die Bandbreite der Abbildungsmatrix des Aufklappoperators ist ebenfalls drei. Für $t \in [0, 1)$ können nur die Koeffizienten mit den Funktionswerten $w(t)$, $w(-1+t)$, $w(1+t)$, $w(-t)$, $w(1-t)$ und $w(2-t)$ ungleich Null werden. Alle anderen Koeffizienten sind Null. Für die Dreifachüberlappung ergibt sich daher das aufgeklappte Signal aus

$$\begin{aligned} \mathcal{U}_w f_{\mathcal{T}}(2j - t) &= -w(2-t) f_{\mathcal{T}}(2j - 2 + t) + w(1-t) f_{\mathcal{T}}(2j - t) + w(-t) f_{\mathcal{T}}(2j + t), \\ \mathcal{U}_w f_{\mathcal{T}}(2j + t) &= -w(1+t) f_{\mathcal{T}}(2j - t) + w(t) f_{\mathcal{T}}(2j + t) + w(-1+t) f_{\mathcal{T}}(2j + 2 - t) \end{aligned} \quad (2.59)$$

mit $t \in [0, 1)$. Man erhält für den Aufklappoperator die in Abbildung 2.11 dargestellte verallgemeinerte Matrix-Vektor-Beschreibung. Da auch die vierfach und sechsfach überlappenden Fensterfunktionen einen beschränkten Träger besitzen, erhält man hier für die Abbildungsmatrizen des Aufklappoperators ebenfalls Bandmatrizen. So ergibt sich z.B. für die Vierfachüberlappung das aufgeklappte Signal aus

$$\begin{aligned} \mathcal{U}_w f_{\mathcal{T}}(2j - t) &= -w(3-t) f_{\mathcal{T}}(2j - 2 - t) - w(2-t) f_{\mathcal{T}}(2j - 2 + t) + w(1-t) f_{\mathcal{T}}(2j - t) \\ &\quad + w(-t) f_{\mathcal{T}}(2j + t) - w(-1-t) f_{\mathcal{T}}(2j + 2 - t), \\ \mathcal{U}_w f_{\mathcal{T}}(2j + t) &= -w(2+t) f_{\mathcal{T}}(2j - 2 + t) - w(1+t) f_{\mathcal{T}}(2j - t) + w(t) f_{\mathcal{T}}(2j + t) \\ &\quad + w(-1+t) f_{\mathcal{T}}(2j + 2 - t) - w(-2+t) f_{\mathcal{T}}(2j + 2 + t) \end{aligned} \quad (2.60)$$

mit $t \in [0, 1)$. Da für gerade Überlappungsfaktoren der Träger der jeweiligen Fensterfunktion in der Mitte eines Intervalls beginnt und in der Mitte eines Intervalls endet, besitzen die äußeren Nebendiagonalen des Matrixbandes in Abhängigkeit von t Nullkoeffizienten. Aufgrund der Bandstruktur und der dünnen Besetzung der Transformationsmatrizen der Aufklappoperatoren eignen sich diese für die Entwicklung von schnellen Algorithmen.

[illegible]

Abbildung 2.10: Rekonstruktion von Signalen, die sich auf die gesamte reelle Achse beziehen; Darstellung der Aufklappoperation für mehrfach überlappende Kosinus-IV-Basen als verallgemeinerte Matrix-Vektor-Beschreibung, $t \in [0, 1)$. Alle Fenster sind Translate einer Fensterfunktion $w(t)$.

2.3 Fenster zur Konstruktion von lokalen trigonometrischen Basisfunktionen

Bezugnehmend auf die in Abschnitt 1.1.3 beschriebenen Kriterien für Basisfunktionen wird gefordert, dass die überlappenden lokalen trigonometrischen Basen mindestens den Gleichanteil von Signalen mit den Basisfunktionen ψ_j^0 reproduzieren bzw. mit den dualen Basisfunktionen $\tilde{\psi}_j^0$ auflösen, d.h.

$$\sum_{j \in \mathbb{Z}} \psi_j^0(t) = 1, \quad t \in \mathbb{R} \quad \text{bzw.} \quad \int_{\mathbb{R}} \tilde{\psi}_j^k(t) dt = 0, \quad k \in \mathbb{N}, \quad k > 0. \quad (2.61)$$

Nachfolgend werden Fensterfunktionen zur Konstruktion von überlappenden Kosinus-IV-Basen vorgestellt, die diese Bedingung erfüllen und daher für die Anwendung in der Bilddatenkompression von Interesse sind.

2.3.1 Zweifach überlappende Fenster

Wie bereits in der Einleitung zu diesem Kapitel dargelegt wurde, gehört das Sinusfenster von Malvar [Mal90, Mal92], das durch

$$w(t) := \begin{cases} \sin\left(\frac{1}{2}\pi\left(t + \frac{1}{2}\right)\right), & t \in \left[-\frac{1}{2}, \frac{3}{2}\right), \\ 0, & \text{sonst} \end{cases} \quad (2.62)$$

definiert ist, zu dem bekanntesten Fenster zur Konstruktion von zweifach überlappenden lokalen trigonometrischen Basisfunktionen. Es handelt sich hierbei um die einzige Fensterfunktion, mit der sich orthogonale zweifach überlappende trigonometrische Basen konstruieren lassen, die den Gleichanteil mit den Basisfunktionen ψ_j^0 darstellen. Das Sinusfenster und die Basisfunktionen für die Frequenzparameter $k = 0$ und $k = 3$ sind in Abbildung 2.12 dargestellt. Die Fensterfunktion und daraus resultierend auch die Basisfunktionen besitzen den Nachteil, dass sie an den Intervallgrenzen $t = -\frac{1}{2}$ und $t = \frac{3}{2}$ nicht stetig differenzierbar sind. Dadurch wird bei der Anwendung dieser Basis ein schnelleres Abklingen der Transformationskoeffizienten verhindert. Dieses Problem lässt sich mit Hilfe von Klappoperatoren veranschaulichen. Abbildung 2.13 zeigt die Klappung einer Linearen mit dem nicht stetig differenzierbaren Malvar-Fenster. Bei Betrachtung des kontinuierlichen Falls (gestrichelt eingezeichnet) sind im

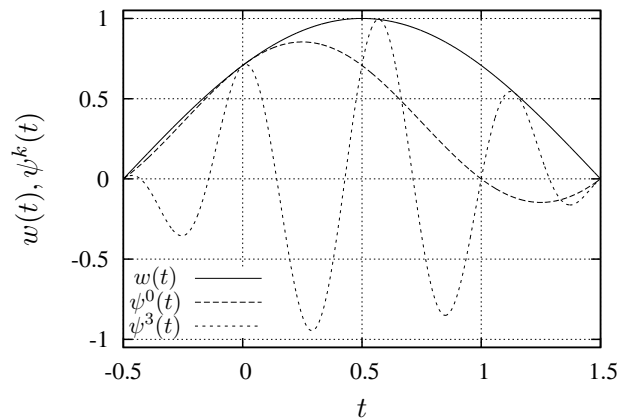


Abbildung 2.12: Sinusfenster zur Konstruktion der zweifach überlappenden orthogonalen Malvar-Basis.

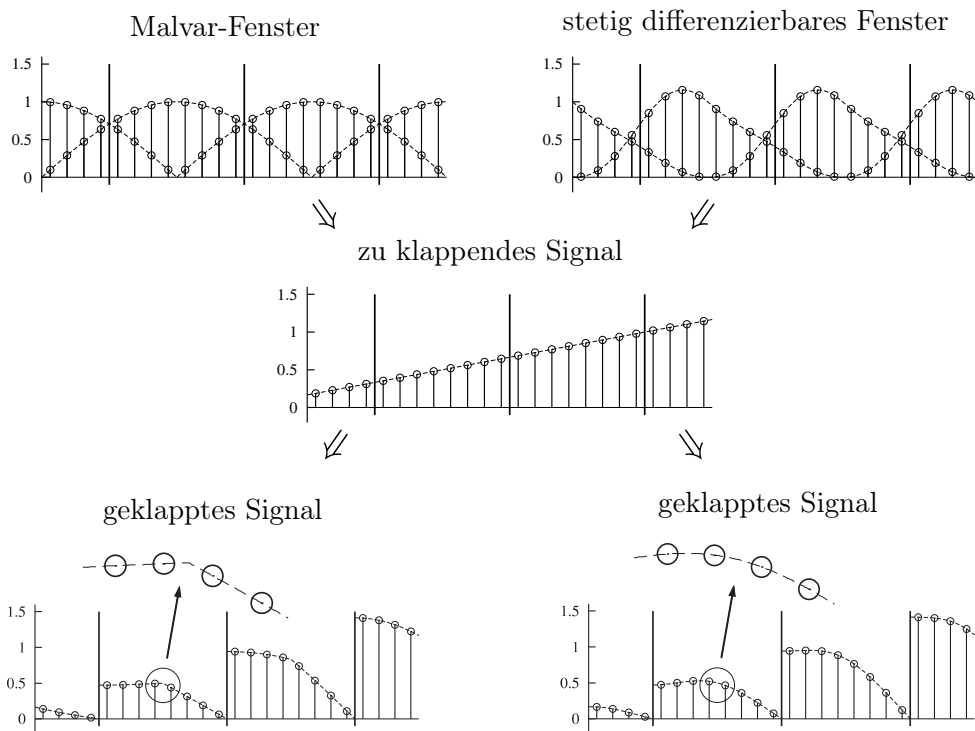


Abbildung 2.13: Klappung eines Signals unter Verwendung verschiedener zweifach überlappender Fensterfunktionen. Linke Seite: Verwendung des nicht stetig differenzierbaren Malvar-Fensters, das geklappte kontinuierliche Signal ist in der Intervallmitte nicht stetig differenzierbar. Rechte Seite: Verwendung einer stetig differenzierbaren Fensterfunktion aus [Bit00b], das geklappte kontinuierliche Signal ist in der Intervallmitte stetig differenzierbar.

geklappten Signal nicht stetig differenzierbare Stellen in den Intervallmitten zu erkennen. Um diese „Knicke“ zu rekonstruieren, sind eine Vielzahl von hochfrequenten Kosinus-IV-Funktionen erforderlich. Auch im diskreten Fall sind im Vergleich zur Darstellung eines geklappten Signals, das in der Intervallmitte einen glatteren Kurvenverlauf aufweist, mehr hochfrequente Basisfunktionen erforderlich. Diese fallen in der verlustbehafteten Datenkompression jedoch im Allgemeinen der Quantisierung zum Opfer. Hohe Kompressionsfaktoren können daher zu sichtbaren Kompressionsartefakten innerhalb der Intervalle führen, die sich im rekonstruierten Bild als Gitterstruktur bemerkbar machen. Bei der Verwendung von stetig differenzierbaren Fensterfunktionen [Bit00b], ebenfalls in Abbildung 2.13 dargestellt, tritt das diskutierte Problem nicht auf. Es sind daher weniger hochfrequente Basisfunktionen zur Signalrepräsentation notwendig, was zunächst eine Verbesserung der Kompressionsleistung in der Bilddatenkompression vermuten lässt.

Um zweifach überlappende trigonometrische Basisfunktionen zu konstruieren, die neben der Auflösung bzw. Darstellung des Gleichanteils mit den Basisfunktionen der niedrigsten Frequenz auch die Bedingung der stetigen Differenzierbarkeit erfüllen, muss man sich von der strengen Bedingung der Orthogonalität trennen. In [Bit00b] werden solche stetig differenzierbaren Kosinus-IV-Basen vorgestellt, die darüber hinaus auch lineare Funktionen mit wenigen Basisfunktionen reproduzieren. Untersuchungen in [Mal00] zeigen jedoch, dass diese Basisfunktionen in der Bilddatenkompression sowohl in Bezug auf die objektive als auch auf die

subjektive Rekonstruktionsqualität nicht so leistungsfähig sind wie die Malvar-Basis. Insbesondere die Blockgrenzen bleiben als schnelle Grauwertwechsel deutlich sichtbar. Die Untersuchungen im Rahmen dieser Arbeit konzentrieren sich daher bei der Zweifachüberlappung auf das Malvar-Fenster.

2.3.2 Dreifach überlappende Fenster

Für die Reproduktion von linearen Signalen mit zweifach überlappenden Kosinus-IV-Basen werden mindestens die Basisfunktionen ψ^0 , ψ^1 und ψ^2 sowie deren ganzzahlige Translate benötigt [Bit00b]. Diese Schranke lässt sich überwinden, indem man sich von der Zweifachüberlappung trennt und das Überlappen von mindestens drei Fensterfunktionen zulässt. Hierdurch entstehen mehr Freiheitsgrade beim Fensterentwurf, die zur Verbesserung der Rekonstruktionseigenschaften genutzt werden können. Das ermöglicht es, trigonometrische Basen zu entwickeln, die eine lineare Funktion ausschließlich mit den Basisfunktionen ψ_j^0 der niedrigsten Frequenz rekonstruieren:

$$f(t) = at + b = \sum_{j \in \mathbb{Z}} (aj + b) \psi_j^0(t). \quad (2.63)$$

Basisfunktionen, die diese Reproduktionseigenschaft erfüllen, lassen sich mit den Fensterfunktionen [Bit02a]

$$w(t) := \begin{cases} g(t+1), & \text{für } t \in (-1, 0], \\ \frac{1+t-2g(t)\sqrt{2}\sin(\frac{1}{2}\pi t)}{\sqrt{2}\cos(\frac{1}{2}\pi t)}, & \text{für } t \in (0, 1), \\ \frac{\sqrt{2}}{\pi}, & \text{für } t = 1, \\ \frac{1-t}{\sqrt{2}\cos(\frac{1}{2}\pi t)} - g(t-1), & \text{für } t \in (1, 2), \\ 0, & \text{sonst} \end{cases} \quad (2.64)$$

konstruieren. Die Funktion g hat die Bedingungen $g(0) = 0$, $g(1) = \frac{1}{\sqrt{2}}$ und $g'(1) = \frac{1}{\sqrt{2}}$ zu erfüllen [Bit02a]. Die Lineare¹

$$g^{(1)}(t) = \frac{1}{\sqrt{2}}t \quad (2.65)$$

genügt diesen Forderungen. Das hieraus resultierende Fenster $w^{(1)}$ ist stetig, jedoch nicht stetig differenzierbar. Dadurch wird ein schnelleres Abklingen der Transformationskoeffizienten verhindert. Ein stetig differenzierbares dreifach überlappendes Fenster erhält man mit der Funktion

$$g^{(2)}(t) = \frac{t^2(24 + \pi^2 - 2(12 + \pi^2)t + (8 + \pi^2)t^2)}{8\sqrt{2}}. \quad (2.66)$$

Das so rekonstruierte Fenster $w^{(2)}$ weist jedoch im Vergleich zum Fenster $w^{(1)}$ schlechtere Riesz-Konstanten auf. Ein weiteres Fenster $w^{(3)}$ kann mit der Funktion

$$g^{(3)}(t) = \frac{\pi(1+t) + (-2t^2 + \pi(-1+t)(1+2t))\cos(\frac{1}{2}\pi t)}{2\sqrt{2}\pi\sin(\frac{1}{2}\pi t)} \quad (2.67)$$

erzeugt werden. Das so konstruierte Fenster $w^{(3)}$ ist zwar auch nicht stetig differenzierbar, es besitzt jedoch im Vergleich zu den beiden vorangegangenen Fensterfunktionen die besten

¹Hinweis: Die hochgestellten Klammernausdrücke kennzeichnen hier keine Ableitung sondern dienen im Rahmen dieser Arbeit zur Unterscheidung der drei verschiedenen dreifach überlappenden Fensterfunktionen.

Riesz-Schranken und somit die geringste Konditionszahl. Die numerische Bestimmung der Riesz-Konstanten erfolgt nach den Gleichungen (2.23) sowie (2.24) und ist in Abbildung 2.15 am Beispiel von $w^{(2)}$ veranschaulicht. Die ermittelten Riesz-Konstanten der dreifach überlappenden Kosinus-IV-Basen auf der reellen Achse sind in Tabelle 2.1 zusammengefasst. Hieraus kann abgeleitet werden, dass in der Bilddatenkompression mit dem Fenster $w^{(3)}$ im Vergleich zu den Fenstern $w^{(2)}$ und $w^{(1)}$ die besten Rekonstruktionsergebnisse zu erwarten sind. Die dualen Fensterfunktionen \tilde{w} zur Konstruktion der Basisfunktionen für die Signalanalyse werden nach Gleichung [BC04]

$$\tilde{w}(t) = \frac{1}{\sqrt{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{Z_2 w\left(\frac{1}{2} - \frac{t}{2}, \xi\right)}{\det \mathbf{M}\left(\frac{t}{2}, \xi\right)} d\xi \quad (2.68)$$

aus den Fensterfunktionen w abgeleitet. Im Gegensatz zu den Fensterfunktionen w besitzen die dualen Fensterfunktionen \tilde{w} und somit auch die dualen Basisfunktionen $\tilde{\psi}_j^k$ keinen beschränkten Träger. Die Fensterfunktionen und die dualen Fensterfunktionen sind in Abbildung 2.14 dargestellt. Des Weiteren wird an einem Beispiel die Rekonstruktion einer Linearen mit den Basisfunktionen der niedrigsten Frequenz demonstriert. Es ist zu erkennen, dass nach einem Einschwingvorgang eine Lineare perfekt reproduziert wird.

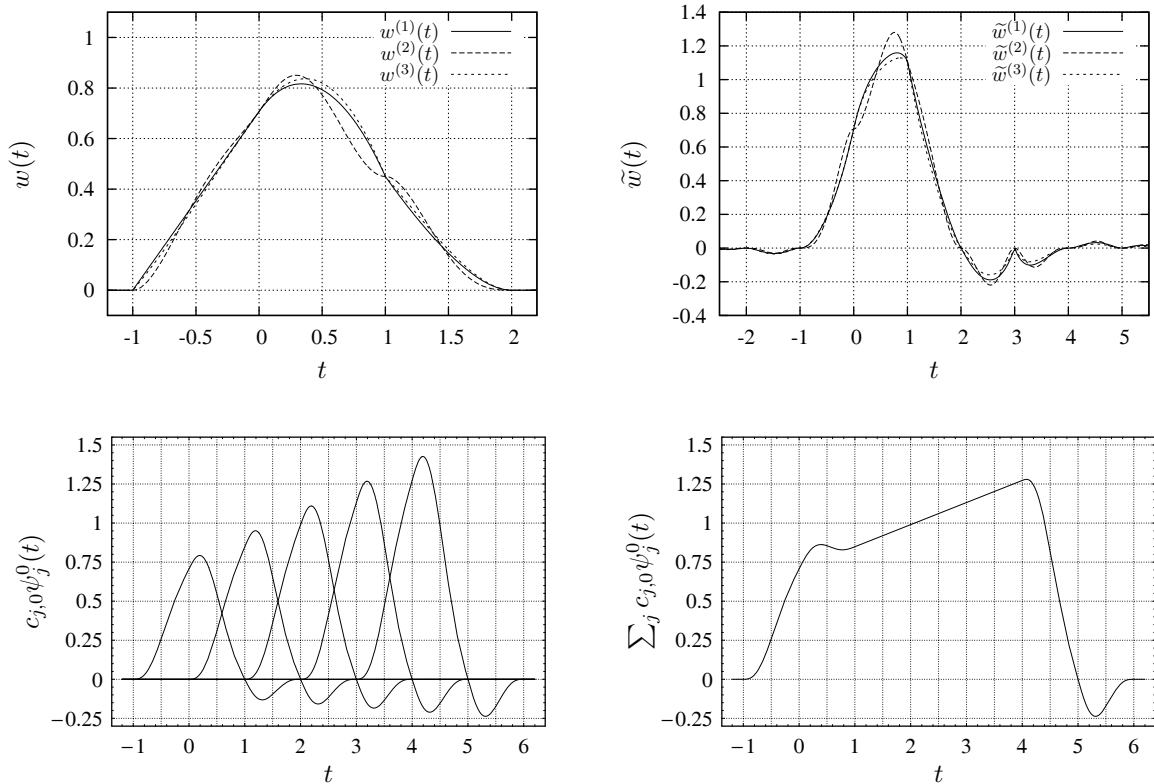
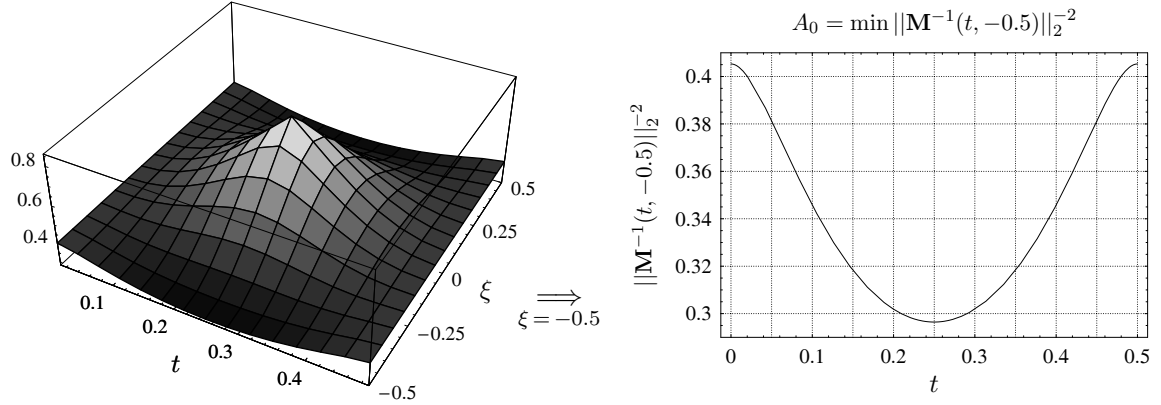


Abbildung 2.14: Dreifachüberlappung: Fensterfunktionen w (oben links) und ihre dualen Fensterfunktionen \tilde{w} (oben rechts); mit dem Fenster $w^{(2)}$ erzeugte Basisfunktionen zur Repräsentation einer Linearen (unten links) und stückweise Rekonstruktion einer Linearen (unten rechts).

Bestimmung der unteren Riesz-Schranke A_0 durch Suche des globalen Minimums von $\|\mathbf{M}^{-1}(t, \xi)\|_2^{-2}$ für $(t, \xi) \in Q^+ := (0, \frac{1}{2}) \times [-\frac{1}{2}, \frac{1}{2}]$:



Bestimmung der oberen Riesz-Schranke B_0 durch Suche des globalen Maximums von $\|\mathbf{M}(t, \xi)\|_2^2$ für $(t, \xi) \in Q^+ := (0, \frac{1}{2}) \times [-\frac{1}{2}, \frac{1}{2}]$:

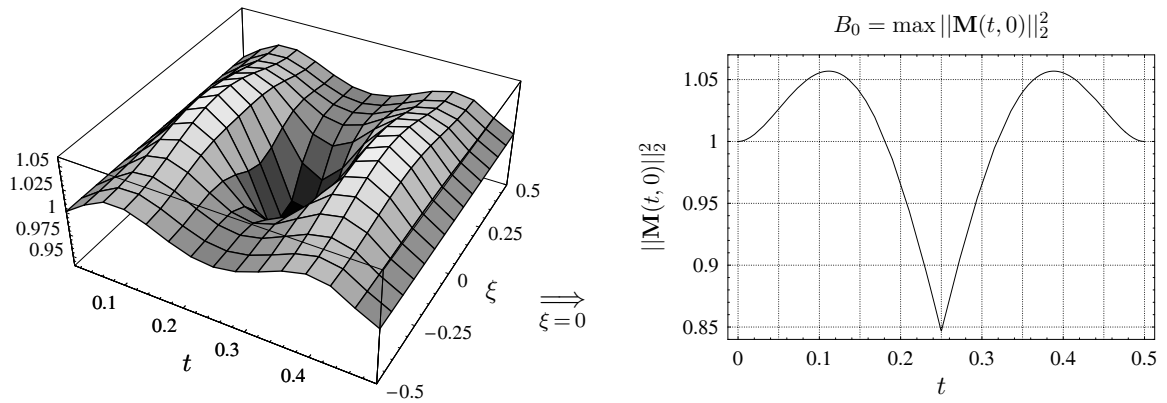


Abbildung 2.15: Bestimmung der Riesz-Schranken am Beispiel der dreifach überlappenden Fensterfunktion $w^{(2)}$.

	A_0	B_0	κ
Fenster $w^{(1)}$	0.3431	1.0089	1.7147
Fenster $w^{(2)}$	0.2965	1.0568	1.8881
Fenster $w^{(3)}$	0.4053	1.0223	1.5881

Tabelle 2.1: Riesz-Konstanten und Konditionszahlen der dreifach überlappenden Kosinus-IV-Basen.

2.3.3 $2m$ -fach überlappende Fenster

Eine für die Bilddatenkompression sehr interessante Klasse von mehrfach überlappenden Fensterfunktionen wurde von Bittner und Chui in [BC04] eingeführt. Die Fensterfunktionen zur Konstruktion von Kosinus-IV-Basen sind durch

$$w(t) := \sum_{k=0}^{m-1} (2\pi)^{-k} \binom{m-1+k}{m-1} M_m^{(k)}\left(\frac{1}{2}\left(t - \frac{1}{2}\right)\right) \cos\left(\frac{1}{2}\pi\left(t - \frac{1}{2} + k\right)\right) \quad (2.69)$$

definiert. Der Ausdruck $M_m^{(k)}(t)$ stellt die k -te Ableitung eines um Null zentrierten B-Splines [Chu92] der Ordnung m dar, der durch

$$M_1(t) := \frac{1}{2} \left(\chi_{[-\frac{1}{2}, \frac{1}{2})} + \chi_{(-\frac{1}{2}, \frac{1}{2}]}\right), \quad (2.70)$$

$$M_m(t) := M_{m-1}(t) * M_1(t) = \int_{-\frac{1}{2}}^{\frac{1}{2}} M_{m-1}(t - \tau) d\tau, \quad m \geq 2 \quad (2.71)$$

definiert ist. Man erhält aus dieser Beschreibung Fensterfunktionen mit einem Überlappungsfaktor von $2m$. Für $m = 1$ ergibt sich das zweifach überlappende Fenster von Malvar [Mal90, Mal92], das bereits in Abschnitt 2.3.1 eingeführt wurde. Die Untersuchungen im Rahmen dieser Arbeit konzentrieren sich für die Anwendung in der Bilddatenkompression auf die Zweifach-, Vierfach- und Sechsfachüberlappung dieser Klasse von $2m$ -fach überlappenden Basisfunktionen und wird am Ende des Abschnitts kurz begründet. Abbildung 2.16 stellt die ausgewählten Fensterfunktionen w und die korrespondierenden dualen Fensterfunktionen \tilde{w} gegenüber. Bei der Mehrfachüberlappung besitzen die dualen Fenster \tilde{w} für die Konstruktion von Basisfunktionen zur Signalanalyse keinen kompakten Träger.

Die mehrfach überlappenden lokalen trigonometrischen Basen, die mit den nach Gleichung (2.69) definierten Fensterfunktionen konstruiert werden, weisen im Vergleich zu den bisher bekannten und untersuchten Basisfunktionen Eigenschaften auf, die für bestimmte Parameter m beim Einsatz in der Bilddatenkompression eine bessere Leistungsfähigkeit erwarten lassen. So gelingt es mit den Fensterfunktionen für $m > 1$ trigonometrische Basen zu entwickeln, die auch Polynome höherer Ordnungen ($p \in \prod_{m-1}$) mit den Basisfunktionen der niedrigsten

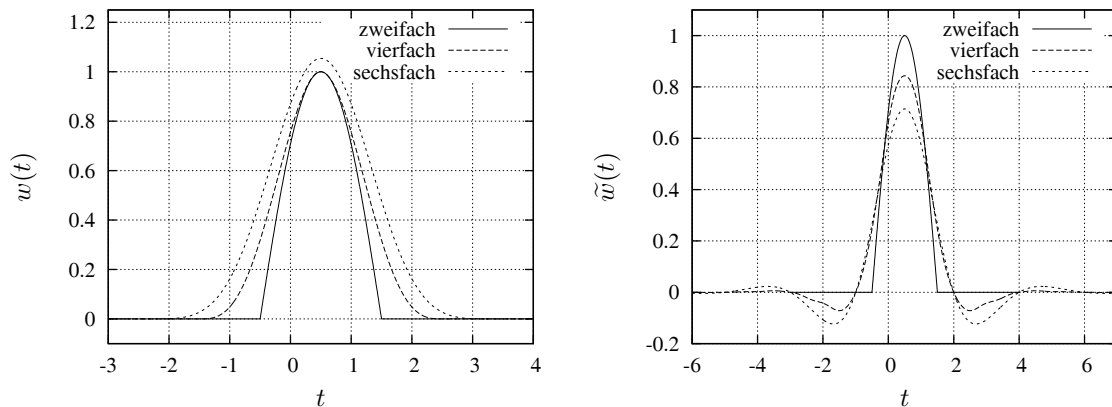


Abbildung 2.16: Fenster (links) zur Konstruktion von zweifach, vierfach und sechsfach überlappenden lokalen trigonometrischen Basen und ihre dualen Fenster, die keinen beschränkten Träger aufweisen (rechts).

	A_0	B_0	κ
Zweifachüberlappung, $m = 1$	1	1	1
Vierfachüberlappung, $m = 2$	1	1.4053	1.1855
Sechsfachüberlappung, $m = 3$	1	2.1394	1.4627

Tabelle 2.2: Riesz-Konstanten der Klasse $2m$ -fach überlappender Basisfunktionen für ausgewählte m .

Frequenz rekonstruieren, d.h.

$$p(t) = \sum_{j \in \mathbb{Z}} a_j w(t - j) \sqrt{2} \cos \left(\frac{1}{2} \pi (t - j) \right). \quad (2.72)$$

Das bedeutet, dass für die Reproduktion einer Linearen mit z.B. vierfach überlappenden trigonometrischen Basen ($m = 2$) wie bei der Dreifachüberlappung nur die Basisfunktionen der niedrigsten Frequenz benötigt werden. Darüber hinaus können im Gegensatz zur Dreifachüberlappung jedoch auch mit Polynomen der Ordnung $m - 1$ modulierte Harmonische durch jeweils zwei Basisfunktionen pro Frequenzparameter k nach

$$p(t) \cos(k\pi t - \alpha) = \sum_{j \in \mathbb{Z}} \sum_{l=k-1}^k a_{j,l} w(t - j) \sqrt{2} \cos \left((2l + 1) \frac{1}{2} \pi (t - j) \right) \quad (2.73)$$

rekonstruiert werden. Diese Eigenschaft soll eine sparsame Repräsentation von Texturen mit oszillierendem Charakter ermöglichen. Ein weiterer Vorteil dieser Fenster liegt in der hohen Glattheit dieser Funktionen. Die Fensterfunktionen sind $(2m - 2)$ -fach stetig differenzierbar. Neben den bereits genannten Eigenschaften besitzen die konstruierten Basisfunktionen im Vergleich zu den bisher betrachteten biorthogonalen trigonometrischen Basen für kleine Parameter m sehr gute Riesz-Konstanten (vgl. Tabelle 2.2).

Anhang B.1 gibt einen Überblick über Basisfunktionen, die mit den in diesem Kapitel beschriebenen zweifach, dreifach, vierfach und sechsfach überlappenden Fensterfunktionen konstruiert wurden. Des Weiteren sind im Anhang C die Amplitudengänge der Übertragungsfunktionen dargestellt, die sich aus der Interpretation der trigonometrischen Transformationen als Teilbandsysteme [Ohm95] ableiten. Es ist zu erkennen, dass sich mit steigendem Überlappungsfaktor die Nebenmaxima im Spektralbereich verringern. Somit wird eine bessere Frequenzselektivität und Dekorrelation der Transformationskoeffizienten erzielt.

Bewertet man alle dargestellten Eigenschaften in ihrer Gesamtheit, so ist zu erwarten, dass für die Anwendung von lokalen trigonometrischen Basen in der Bilddatenkompression die Vierfachüberlappung einen geeigneten Kompromiss zwischen Rekonstruktions- und Auflösungseigenschaften, Frequenzselektivität, Riesz-Stabilität, Glattheit und Trägerlänge der Basisfunktionen darstellt. Bei größeren Überlappungsfaktoren wie beispielsweise der Sechsfachüberlappung verbessern sich zwar die Dekorrelations- und polynomialen Reproduktionseigenschaften, jedoch verschlechtern sich die Riesz-Schranken und es sind bei höheren Kompressionsfaktoren aufgrund der langen Träger der Basisfunktionen deutlich sichtbare Ringing-Artefakte zu erwarten. Daher finden im Rahmen dieser Arbeit höhere Überlappungsfaktoren als die Sechsfachüberlappung keine Berücksichtigung.

Kapitel 3

Lokale trigonometrische Basen und Klappoperatoren auf dem Intervall

Dieses Kapitel befasst sich mit der Repräsentation von Signalen, die nur auf einem beschränkten Intervall definiert sind. Zur effizienten Darstellung solcher Signale werden überlappende lokale trigonometrische Basen vorgeschlagen, die auf einer neu eingeführten Kombination von trigonometrischen Basisfunktionen beruhen. Ausgehend von den im vorangegangenen Kapitel gewonnenen Resultaten zu Klapp- und Aufklappoperatoren erfolgt die Herleitung dieser Operatoren für das neuartige System von Basisfunktionen auf dem Intervall. Schließlich werden neu entwickelte Fensterfunktionen zur Konstruktion von Randbasisfunktionen eingeführt. Es wird gezeigt, dass die vorgeschlagenen Basen auf dem Intervall die grundlegenden Kriterien der Rekonstruktion von Konstanten mit den Basisfunktionen der niedrigsten Frequenz und der Riesz-Stabilität erfüllen.

In vielen praktischen Anwendungen der Signal- und Bildverarbeitung werden Signale $f \in L^2([0, L])$ verarbeitet, die zeitlich bzw. örtlich nur auf einem beschränkten Intervall $[0, L]$ definiert sind. Hierdurch ergibt sich für die Signaldarstellung mit überlappenden Basisfunktionen, die sich auf die gesamte reelle Achse \mathbb{R} beziehen, folgendes zu lösendes Problem. Liegt der Träger einer Basisfunktion nicht vollständig im Intervall $[0, L]$, so überlappt die Basisfunktion eine oder bei entsprechender Trägerlänge auch beide Signalränder. Die Transformation benötigt somit Signalwerte, die außerhalb des beschränkten Intervalls $[0, L]$ liegen und nicht definiert sind. Da die in Kapitel 2 betrachteten mehrfach überlappenden dualen Basisfunktionen $\tilde{\psi}_j^k$ zur Signalanalyse keinen beschränkten Träger aufweisen, müsste für die Koordinatentransformation mit diesen Basisfunktionen das zu verarbeitende Signal auf der gesamten reellen Achse \mathbb{R} definiert sein. Die bisher betrachteten Basisfunktionen ψ_j^k zur Signalrekonstruktion besitzen zwar einen beschränkten Träger, durch die Blocküberlappung werden jedoch bei der Signalrekonstruktion Basisfunktionen $\psi_{j<0}^k$, $\psi_{j>L-1}^k$ und somit auch Wichtungskoeffizienten $c_{j<0,k}$, $c_{j>L-1,k}$ benötigt, die Intervallen außerhalb des Signalträgers zugeordnet und ebenfalls nicht definiert sind. Es müssen daher geeignete Methoden zur Verarbeitung von Signalen, die nur auf einem endlichen langen Intervall $[0, L]$ definiert sind, entwickelt werden. Die Ansätze zur Lösung des geschilderten Problems lassen sich grundsätzlich in zwei Gruppen klassifizieren:

- Definition von Signalwerten außerhalb des Intervalls $[0, L] \Rightarrow$ Signalfortsetzung,
- Anpassung der Transformation \Rightarrow Basen auf dem Intervall $[0, L]$.

Abschnitt 3.1 setzt sich zunächst mit Verfahren zur Signalfortsetzung auseinander und gelangt zu dem Resultat, dass die betrachteten Verfahren für die Anwendung von überlappenden Kosinus-IV-Basen in der verlustbehafteten Datenkompression keine geeignete Lösung darstellen. Somit besteht in diesem Kapitel das Ziel, als geeignetere Methode lokale trigonometrische Basen auf dem Intervall einzuführen. Dieser Ansatz entspricht einer Anpassung der Transformation an die Trägerlänge des Signals und zeichnet sich dadurch aus, dass keine Basisfunktionen die Signalränder überlappen. Entsprechend der Zielsetzung ist das Kapitel im Weiteren wie folgt gegliedert: Für Signale, die auf einem beschränkten Intervall $[0, L]$ gegeben sind, wird in Abschnitt 3.2 die Darstellung mit überlappenden lokalen trigonometrischen Basisfunktionen definiert. Des Weiteren wird mit der Mischung von Kosinus-IV-Kosinus-II-Basisfunktionen eine neue Kombination von trigonometrischen Basisfunktionen eingeführt. Abschnitt 3.3 gibt eine allgemeine Definition von Klapp- und Aufklappoperatoren für Basen, die auf einem Intervall $[0, L]$ beschränkt sind. Die Herleitung dieser Operatoren für die neu eingeführten überlappenden Kosinus-IV-Kosinus-II-Basen ist ebenfalls Gegenstand von Abschnitt 3.3. Anschließend befasst sich Abschnitt 3.4 mit der Frage der Riesz-Stabilität und beschreibt, wie sich die Riesz-Schranken des auf dem Intervall definierten Systems von Basisfunktionen mit Hilfe des neu eingeführten Aufklappoperators bestimmen lassen. Die im Rahmen dieser Arbeit entworfenen Fensterfunktionen zur Konstruktion von überlappenden trigonometrischen Randbasisfunktionen werden in Abschnitt 3.5 eingeführt. Es wird gezeigt, dass die mit diesen Fensterfunktionen konstruierten Basen die Eigenschaft der Rekonstruktion von Konstanten mit den Basisfunktionen ψ_j^0 an den Signalrändern gewährleisten. Die Riesz-Stabilität wird mit der Bestimmung der Riesz-Schranken nachgewiesen.

3.1 Signalfortsetzung

Eine in der Signal- und Bildverarbeitung häufig verwendete Methode zur Transformation von Signalen mit sich überlappenden Basisfunktionen ist das Fortsetzen des Signals. Die Signalfortsetzung kommt u.a. auch bei waveletbasierten Bilddatenkompressionsverfahren zum Einsatz. Für die Definition von Signalwerten außerhalb eines beschränkten Intervalls $[0, L]$ existiert eine Reihe von verschiedenen Verfahren [KV89, Bri96, SN96, PM99b, PM99a, Mal98, Bit02b]. Zu den bekanntesten Techniken gehören das Fortsetzen mit Nullen (zero padding), die periodische Fortsetzung (periodic extension) und die symmetrische periodische Fortsetzung (symmetric periodic extension).

Das Fortsetzen des Signals mit Nullen und die periodische Signalfortsetzung besitzen für die Datenkompression einige wesentliche Nachteile. An den Signalrändern können Sprungstellen entstehen. Diese Unstetigkeitsstellen erzeugen zusätzliche hochfrequente Signalkomponenten, die zu einer Verschlechterung des Codiergewinns führen. Bei der Anwendung in der Bilddatenkompression können bei starker Quantisierung der Transformationskoeffizienten sichtbare Ringing-Artefakte an den Rändern des rekonstruierten Bildes auftreten. Weiterhin erhält man durch das Fortsetzen mit Nullen zusätzliche Transformationskoeffizienten und es müssen somit mehr Daten übertragen werden. Dieser Effekt ist in der Datenkompression unerwünscht. Bei der periodischen Signalfortsetzung sind dagegen keine zusätzlichen Transformationskoeffizienten zu übertragen. Hier wird die Eigenschaft ausgenutzt, dass die Transformation eines periodischen Signals mit der Periodenlänge L zu Transformationskoeffizienten mit der gleichen Periodenlänge führt. Zur Gewährleistung der perfekten Signalrekonstruktion genügt es daher, nur die dem Intervall $[0, L]$ zugeordneten Transformationskoeffizienten zu speichern und zu übertragen.

Das Problem der Sprungstellen an den Signalrändern lässt sich durch die symmetrische peri-

odische Fortsetzung lösen. Bei diesem Verfahren wird das Signal $f \in L^2([0, L])$ im Intervall $[-L, L]$ symmetrisch und dann auf der reellen Achse periodisch mit einer Periodenlänge von $2L$ fortgesetzt. Durch die Symmetrisierung entstehen keine Diskontinuitäten an den Intervallgrenzen und man erhält eine stetige Fortsetzung des Signals. Die Transformationskoeffizienten weisen wie das fortgesetzte Signal eine Periodenlänge von $2L$ auf. In Analogie zur periodischen Signalfortsetzung genügt daher zur Gewährleistung der perfekten Rekonstruktion die Speicherung und Übertragung der dem Intervall $[-L, L]$ zugeordneten Transformationskoeffizienten. Im Vergleich zur periodischen Signalfortsetzung hat sich jedoch die Anzahl der Transformationskoeffizienten verdoppelt. Für den besonderen Fall, dass die Basisfunktionen gewisse Symmetrieeigenschaften aufweisen, erhält man für das symmetrisch fortgesetzte Signal im Intervall $[-L, L]$ Transformationskoeffizienten, die ebenfalls gewisse Symmetrieeigenschaften aufweisen. In diesem Fall genügt zur Gewährleistung der perfekten Rekonstruktion auch bei der symmetrischen periodischen Fortsetzung die Speicherung bzw. Übertragung der dem Intervall $[0, L]$ zugeordneten Transformationskoeffizienten. Das heißt, dass durch die Verwendung von symmetrischen Basisfunktionen bei der symmetrischen Signalfortsetzung ebenfalls keine zusätzlichen Transformationskoeffizienten erzeugt werden. Für die Datenkompression sind daher Transformationen mit symmetrischen Basisfunktionen und Verfahren zur symmetrischen Signalfortsetzung¹ von besonderem Interesse. Die von der periodischen Signalfortsetzung bekannten sichtbaren Ringing-Artefakte an den Signalländern treten in der verlustbehafteten Bilddatenkompression unter Verwendung der symmetrischen Signalfortsetzung nicht auf und es wird erfahrungsgemäß im Vergleich zur periodischen Signalfortsetzung eine Erhöhung der objektiven Rekonstruktionsqualität um ca. 0.1 – 0.5 dB erzielt.

Die im Rahmen dieser Arbeit untersuchten überlappenden Kosinus-IV-Basisfunktionen sind nicht symmetrisch. Die symmetrische periodische Signalfortsetzung eignet sich daher nicht für die Anwendung in der Datenkompression mit überlappenden Kosinus-IV-Basen. Die periodische Signalerweiterung eignet sich ebenfalls nicht für die Anwendung in der verlustbehafteten Datenkompression, da sie bei hohen Kompressionsfaktoren zu sichtbaren Randartefakten führt. Diese können in Abhängigkeit vom Überlappungsfaktor weit in das rekonstruierte Bild hineinreichen. Um eine Signalfortsetzung zu vermeiden, werden daher für die Darstellung von Signalen, die auf einem endlich langen Intervall definiert sind, in den nachfolgenden Abschnitten lokale trigonometrische Basen auf dem Intervall eingeführt.

3.2 Lokale trigonometrische Basen zur Darstellung von Signalen auf dem Intervall

Für die Darstellung von Signalen $f \in L^2([0, 2N])$ mit überlappenden lokalen trigonometrischen Basen wird die Einteilung der Zeitachse in äquidistante Intervalle $[j, j+1)$, $j = 0, 1, \dots, 2N-1$, betrachtet. Zur Repräsentation der Signale wird das System

$$\{\psi_j^k\} := \{\psi_j^k : \psi_j^k \in L^2([0, 2N]), j = 0, 1, \dots, 2N-1, k \in \mathbb{N}_0\} \quad (3.1)$$

eingeführt und die Signalrekonstruktion durch Gleichung

$$f = \sum_{j=0}^{2N-1} \sum_{k \in \mathbb{N}_0} c_{j,k} \psi_j^k = \sum_{j=0}^{2N-1} \sum_{k \in \mathbb{N}_0} \langle f, \tilde{\psi}_j^k \rangle \psi_j^k \quad (3.2)$$

¹Es sei darauf hingewiesen, dass in Abhängigkeit von der Transformation verschiedene Verfahren zur symmetrischen Fortsetzung existieren [Bar94, Sch00].

beschrieben. Die Basisfunktionen ψ_j^k bestehen aus trigonometrischen Funktionen θ_j^k , die mit überlappenden Fensterfunktionen w_j gewichtet werden, d.h.

$$\psi_j^k := w_j \theta_j^k, \quad j = 0, 1, \dots, 2N-1, \quad k \in \mathbb{N}_0. \quad (3.3)$$

Damit die Basisfunktionen nicht die Intervallgrenzen des Signals bei $t = 0$ und $t = 2N$ überschneiden, sollen die u -fach überlappenden Fensterfunktionen w_j die Bedingung

$$\text{supp } w_j \subset \left[\max \left(0, j - (u-1)\frac{1}{2} \right), \min \left(2N, j + 1 + (u-1)\frac{1}{2} \right) \right] \quad (3.4)$$

erfüllen. Bevor geeignete Fensterfunktionen sowie Klapp- und Aufklappoperatoren für die Anwendung in der Bilddatenkompression eingeführt werden, ist die Frage zu diskutieren, von welchem trigonometrischen Typ die Funktionen θ_j^k in den Randintervallen $j = 0$ und $j = 2N-1$ gewählt werden. Die Kosinus-IV-Funktionen θ_j^k , die dem Intervall $[j, j+1)$ zugeordnet sind, besitzen an der Stelle $t = j+1$ Nulldurchgänge. Für das Randintervall $j = 0$ und die inneren Intervalle $j = 1, \dots, 2N-2$ stellt diese Eigenschaft für die Rekonstruktion von Signalen $f \neq 0$ an den Intervallgrenzen im Bereich $[j+1-\varepsilon, j+1)$, $\varepsilon > 0$, $\varepsilon \rightarrow 0$, kein Problem dar. Die Signalanteile für diesen Bereich werden in hohem Maße durch die überlappenden Basisfunktionen ψ_{j+1}^k , die dem benachbarten Intervall $[j+1, j+2)$ zugeordnet sind, repräsentiert. Am rechten Signalrand $t = 2N$ sind jedoch keine solche benachbarten Basisfunktionen ψ_{2N}^k definiert. Setzt man voraus, dass die Fenster- und Basisfunktionen, deren Träger im Intervall $[0, 2N]$ liegen, unverändert bleiben, so führt dies zu Randfenstern w_{2N-1} mit einer Polstelle bei $t = 2N$. In Abbildung 3.1 ist dieses Problem anhand der Zweifach- und der Dreifachüberlappung veranschaulicht. Bei der Verwendung von Randfenstern mit Polstellen erhält man keine Riesz-Basis und keine stabile Signalrepräsentation.

Als Resultat der Untersuchungen im Rahmen dieser Arbeit haben sich für die Konstruktion von geeigneten Basisfunktionen $\psi_{2N-1}^k = w_{2N-1} \theta_{2N-1}^k$ sowohl bei der Zweifachüberlappung als auch bei der Mehrfachüberlappung trigonometrische Funktionen vom Kosinus-II-Typ als sehr praktikabel und effizient erwiesen. Zur Darstellung von Signalen, die nur auf einem Intervall $t \in [0, 2N]$ definiert sind, wird daher im Folgenden das System $\{\psi_j^k : \psi_j^k \in L^2([0, 2N]), j = 0, 1, \dots, 2N-1, k \in \mathbb{N}_0\}$ mit den überlappenden Kosinus-IV-Kosinus-II-Basisfunktionen

$$\psi_j^k(t) := \begin{cases} w_j(t) \sqrt{2} \cos \left(\left(k + \frac{1}{2} \right) (t - j) \pi \right), & \text{für } j = 0, 1, \dots, 2N-2, \quad k \in \mathbb{N}_0, \\ w_j(t) \epsilon_k \cos k\pi(t - j), & \text{für } j = 2N-1, \quad k \in \mathbb{N}_0, \end{cases} \quad (3.5)$$

eingeführt, wobei

$$\epsilon_k := \begin{cases} 1, & \text{für } k = 0, \\ \sqrt{2}, & \text{sonst.} \end{cases} \quad (3.6)$$

Bei der Mischung von Kosinus-IV- und Kosinus-II-Basisfunktionen zur Darstellung von Signalen $f \in L^2([0, 2N])$ handelt es sich um eine neu entwickelte Kombination von trigonometrischen Basisfunktionen. Für das System vom Kosinus-IV-Kosinus-II-Typ werden im nachfolgenden Abschnitt Klapp- und Aufklappoperatoren zur Analyse und Rekonstruktion von Signalen auf dem Intervall hergeleitet.

3.3 Klapp- und Aufklappoperatoren zur Transformation von Signalen auf dem Intervall

Nachdem in den Abschnitten 2.2.3 und 2.2.4 Klapp- und Aufklappoperatoren für mehrfach überlappende Kosinus-IV-Basen auf der reellen Achse hergeleitet wurden, sollen nachfolgend

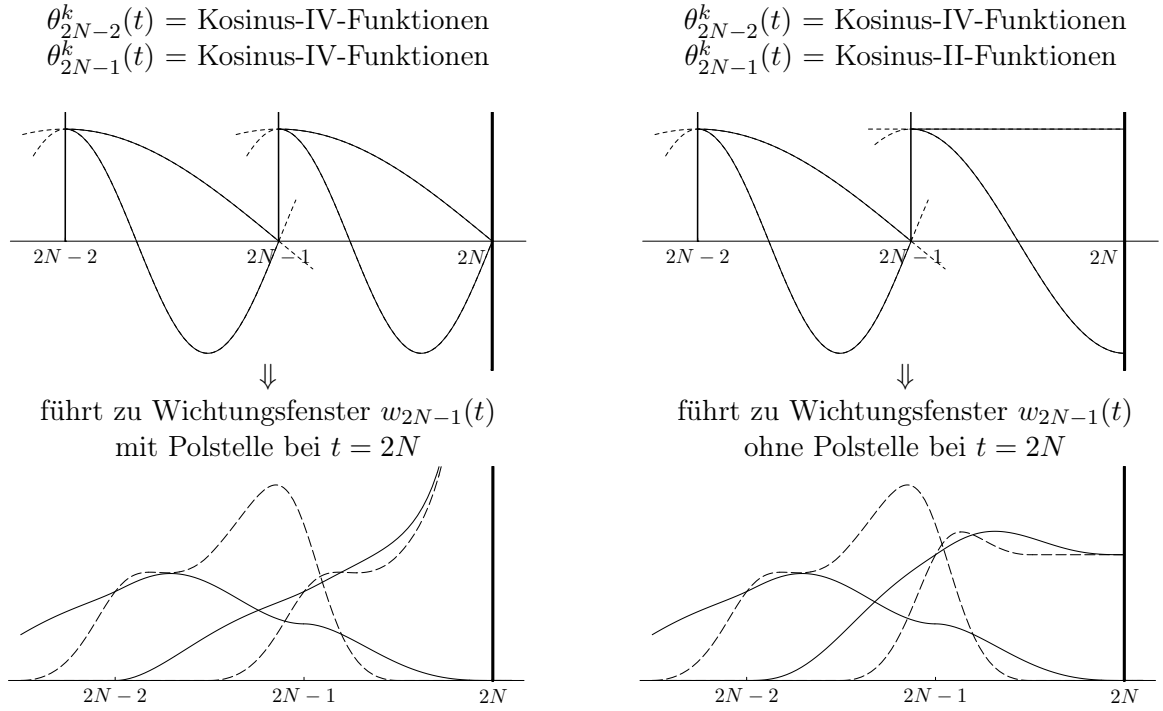


Abbildung 3.1: **oben:** trigonometrische Funktionen der Orthonormalbasis $\{\theta_j^k\}$ am Beispiel der Frequenzparameter $k = 0$ und $k = 1$; **unten:** Fensterfunktionen $w_{2N-2}(t)$ und $w_{2N-1}(t)$ am Beispiel der Zweifachüberlappung (gestrichelte Linie) und der Dreifachüberlappung (durchgehende Linie). Für Kosinus-IV-Funktionen weisen die Fenster $w_{2N-1}(t)$ eine Polstelle bei $t = 2N$ auf, bei der Verwendung von Kosinus-II-Funktionen besitzen die Fenster $w_{2N-1}(t)$ hingegen keine Polstelle bei $t = 2N$.

diese Operatoren für die neu eingeführten Kosinus-IV-Kosinus-II-Basen entwickelt werden. Im Unterschied zu den Ausführungen in den Abschnitten 2.2.3 und 2.2.4 werden nun Signale $f \in L^2([0, 2N])$ und Basisfunktionen $\psi_j^k \in L^2([0, 2N])$ sowie duale Basisfunktionen $\tilde{\psi}_j^k \in L^2([0, 2N])$ betrachtet, die nur auf einem Intervall $[0, 2N]$ definiert sind. Des Weiteren tritt mit den Kosinus-IV-Kosinus-II-Basen eine neue Kombination von trigonometrischen Basisfunktionen am rechten Signalrand auf.

3.3.1 Allgemeine Definitionen

Für die Repräsentation von Signalen, die auf einem Intervall $[0, 2N]$ definiert sind, erfolgt zunächst eine Unterteilung des Gesamtintervalls in äquidistante Teilintervalle $[j, j+1)$, $j = 0, 1, \dots, 2N-1$. Des Weiteren wird zur Analyse von Signalen das System

$$\{\tilde{\psi}_j^k\} := \{\tilde{\psi}_j^k : \tilde{\psi}_j^k \in L^2([0, 2N]), j = 0, 1, \dots, 2N-1, k \in \mathbb{N}_0\} \quad (3.7)$$

eingeführt, wobei die dualen Basisfunktionen $\tilde{\psi}_j^k \in L^2[0, 2N]$ durch

$$\tilde{\psi}_j^k = \tilde{w}_j \theta_j^k, \quad j = 0, 1, \dots, 2N-1, k \in \mathbb{N}_0 \quad (3.8)$$

gegeben sind. Die trigonometrischen und somit frequenzlokalen Funktionen $\theta_j^k, k \in \mathbb{N}_0$, die für jedes $j = 0, 1, \dots, 2N - 1$ eine Orthonormalbasis von $L^2([j, j+1])$ bilden, werden mit zeitlokalen dualen Fensterfunktionen $\tilde{w}_j \in L^2([0, 2N])$ gewichtet. Zur Rekonstruktion von Signalen, die auf dem Intervall $[0, 2N]$ definiert sind, wird das in Gleichung (3.1) eingeführte System $\{\psi_j^k\}$ betrachtet. In Analogie zu den Untersuchungen von Basisfunktionen auf der reellen Achse in Abschnitt 2.2.1 sollen nun Klapp- und Aufklappoperatoren auf dem Intervall $[0, 2N]$ eingeführt werden, die eine Beziehung zwischen den Basisfunktionen ψ_j^k der überlappenden trigonometrischen Basis $\{\psi_j^k : \psi_j^k \in L^2([0, 2N]), j = 0, 1, \dots, 2N - 1, k \in \mathbb{N}_0\}$ und den Basisfunktionen $\chi_{[j,j+1)}\theta_j^k$ der Orthonormalbasis $\{\chi_{[j,j+1)}\theta_j^k : j = 0, 1, \dots, 2N - 1, k \in \mathbb{N}_0\}$ von $L^2([0, 2N])$ herstellen.

Der Übergang von der reellen Achse zur Beschreibung auf dem Intervall erfolgt durch Anpassung der Integrationsgrenzen. Bei der Berechnung des Skalarprodukts zwischen Signal und dualen überlappenden Basisfunktionen wird aufgrund der Beschränktheit dieser Funktionen auf das Intervall $[0, 2N]$ nur noch über dieses Intervall integriert und nicht mehr über die gesamte reelle Achse. Des Weiteren wird sich auf die Summanden beschränkt, die sich auf die in $[0, 2N]$ enthaltenen Teilintervalle beziehen.

Für die dualen überlappenden trigonometrischen Basisfunktionen $\tilde{\psi}_j^k = \tilde{w}_j\theta_j^k$ zur Signalanalyse sei für jedes Signal $f \in L^2([0, 2N])$ die Bedingung

$$\sum_{k \in \mathbb{N}_0} |\langle f, \tilde{\psi}_j^k \rangle|^2 < \infty, \quad j = 0, 1, \dots, 2N - 1, \quad \tilde{\psi}_j^k \in L^2([0, 2N]) \quad (3.9)$$

erfüllt. Der Klappoperator $\mathcal{T}_{\tilde{w}} : L^2([0, 2N]) \rightarrow L_{loc}^2([0, 2N])$ wird dann durch

$$f_{\mathcal{T}} = \mathcal{T}_{\tilde{w}} f := \sum_{j=0}^{2N-1} \sum_{k \in \mathbb{N}_0} \langle f, \tilde{\psi}_j^k \rangle \chi_{[j,j+1)} \theta_j^k = \sum_{j=0}^{2N-1} \sum_{k \in \mathbb{N}_0} c_{j,k} \chi_{[j,j+1)} \theta_j^k, \quad f \in L^2([0, 2N]) \quad (3.10)$$

definiert. Die Wichtungskoeffizienten $c_{j,k}$ beschreiben zum einen das Signal $f \in L^2([0, 2N])$ mit den überlappenden Basisfunktionen $\psi_j^k \in L^2([0, 2N])$ und zum anderen das geklappte Signal $f_{\mathcal{T}} \in L_{loc}^2([0, 2N])$ mit den orthonormalen Basisfunktionen $\chi_{[j,j+1)}\theta_j^k, j = 0, 1, \dots, 2N - 1$. Es gelten die Zusammenhänge

$$c_{j,k} = \langle \mathcal{T}_{\tilde{w}} f, \chi_{[j,j+1)} \theta_j^k \rangle = \int_j^{j+1} \mathcal{T}_{\tilde{w}} f(t) \theta_j^k(t) dt = \langle f, \tilde{\psi}_j^k \rangle = \langle f, \tilde{w}_j \theta_j^k \rangle = \int_0^{2N} f(t) \tilde{w}_j(t) \theta_j^k(t) dt. \quad (3.11)$$

Betrachtet man nun die Darstellung des Signals $f \in L^2([0, 2N])$ als Linearkombination von gewichteten Basisfunktionen gemäß Gleichung (3.2), so erhält man mit Hilfe von Gleichung (3.11) den Ausdruck

$$f = \mathcal{U}_w f_{\mathcal{T}} := \sum_{j=0}^{2N-1} \sum_{k \in \mathbb{N}_0} \int_j^{j+1} f_{\mathcal{T}}(t) \theta_j^k(t) dt \psi_j^k, \quad \psi_j^k \in L^2([0, 2N]), \quad (3.12)$$

durch den der Aufklappoperator definiert wird. Da das System $\{\chi_{[j,j+1)}\theta_j^k : j = 0, 1, \dots, 2N - 1, k \in \mathbb{N}_0\}$ eine Orthonormalbasis von $L^2([0, 2N])$ darstellt, bildet der Aufklappoperator die Basisfunktionen dieser einfachen trigonometrischen Basis auf die Basisfunktionen der überlappenden trigonometrischen Basis $\{\psi_j^k : \psi_j^k \in L^2([0, 2N]), j = 0, 1, \dots, 2N - 1, k \in \mathbb{N}_0\}$ ab:

$$\mathcal{U}_w(\chi_{[j,j+1)}\theta_j^k) = \psi_j^k, \quad j = 0, 1, \dots, 2N - 1, \quad k \in \mathbb{N}_0. \quad (3.13)$$

Der Klappoperator $\mathcal{T}_{\tilde{w}}$ bildet hingegen nach Gleichung (3.10) die auf dem Intervall $[0, 2N]$ definierten überlappenden Basisfunktionen $\psi_j^k = w_j \theta_j^k$ auf die trigonometrischen Funktionen $\chi_{[j, j+1)} \theta_j^k$ ab, d.h.

$$\mathcal{T}_{\tilde{w}} \psi_j^k = \chi_{[j, j+1)} \theta_j^k, \quad j = 0, 1, \dots, 2N-1, \quad k \in \mathbb{N}_0. \quad (3.14)$$

Diese Operatoren spiegeln daher wesentliche Eigenschaften der auf dem Intervall $[0, 2N]$ definierten Basisfunktionen wider. Hierzu gehören u.a. die Riesz-Stabilität sowie die polynomialen Auflösungs- bzw. Rekonstruktionseigenschaften.

3.3.2 Klappoperatoren für Kosinus-IV-Kosinus-II-Basen

Aufgrund der Periodenlänge von $\lambda = 2$ der periodischen Kosinus-II-Funktionen und der antiperiodischen Kosinus-IV-Funktionen erweist es sich als nützlich, zur Analyse von Signalen der Länge $[0, 2N]$ das System $\{\tilde{\psi}_j^k : \tilde{\psi}_j^k \in L^2([0, 2N]), j = 0, 1, \dots, N-1, k \in \mathbb{N}_0\}$ mit den Basisfunktionen

$$\tilde{\psi}_{2j} = \tilde{w}_{2j} \theta_{2j}^k \quad \text{und} \quad \tilde{\psi}_{2j+1} = \tilde{w}_{2j+1} \theta_{2j+1}^k, \quad (3.15)$$

sowie die geklappten Teilsignale $\mathcal{T}_{\tilde{w}} f(2j+t)$ und $\mathcal{T}_{\tilde{w}} f(2j+2-t)$ mit $t \in [0, 1)$ zu betrachten. Bei der Entwicklung von Klappoperatoren für Kosinus-IV-Kosinus-II-Basen auf dem Intervall sind die bereits in Abschnitt 2.2.3 gewonnenen Resultate zu Klappoperatoren für überlappende Kosinus-IV-Basen auf der reellen Achse äußerst hilfreich. Die Herleitungen für die Teilintervalle $[2j, 2j+1) \subset [0, 2N-1)$ und $[2j+1, 2j+2) \subset [0, 2N-1)$, denen Kosinus-IV-Basisfunktionen zugeordnet sind, erfolgen analog zu den dort beschriebenen Umformungen. Hierbei wird sich die Eigenschaft zunutze gemacht, dass bei der Signalanalyse für jedes Intervall ausschließlich Basisfunktionen betrachtet werden, die diesem Intervall zugeordnet sind. Das heißt, dass bei der Transformation mit Klappoperatoren für die Berechnung des in ein Intervall $[2j, 2j+1)$ geklappten Signals nur die Fensterfunktion \tilde{w}_{2j} Berücksichtigung findet, die diesem Intervall zugeordnet ist¹. Zur Analyse von Signalen $f \in L^2([0, 2N])$ lassen sich die Klappoperatoren für die Kosinus-IV-Intervalle somit direkt aus der in Abschnitt 2.2.3 hergeleiteten Gleichung (2.46) ableiten. Für die Intervallindizes $j = 0, 1, \dots, N-2$, die den Intervallbereich $[0, 2N-2)$ beschreiben, erhält man die geklappten Teilsignale

$$\begin{aligned} \mathcal{T}_{\tilde{w}} f(2j+t) &= \sum_{r=0}^{N-1} (-1)^{r+j} \tilde{w}_{2j}(2r+t) f(2r+t) + (-1)^{r+j+1} \tilde{w}_{2j}(2(r+1)-t) f(2(r+1)-t) \\ &= \sum_{r=0}^{N-1} (-1)^{r+j} (\tilde{w}_{2j}(2r+t) f(2r+t) - \tilde{w}_{2j}(2r+2-t) f(2r+2-t)) \end{aligned} \quad (3.16)$$

und

$$\begin{aligned} \mathcal{T}_{\tilde{w}} f(2(j+1)-t) &= \sum_{r=0}^{N-1} (-1)^{r+j+1} \tilde{w}_{2(j+1)-1}(2(r+1)-t) f(2(r+1)-t) \\ &\quad - (-1)^{r+j+1} \tilde{w}_{2(j+1)-1}(2r+t) f(2r+t) \\ &= \sum_{r=0}^{N-1} (-1)^{r+j} (\tilde{w}_{2j+1}(2r+2-t) f(2r+2-t) + \tilde{w}_{2j+1}(2r+t) f(2r+t)), \end{aligned} \quad (3.17)$$

¹Bei der Rekonstruktion eines Intervalls mit dem Aufklappoperator finden hingegen auch die Fenster benachbarter Intervalle Berücksichtigung, da das Signal bei der Darstellung mit überlappenden Basisfunktionen erst durch die Überlagerung mit den benachbarten Basisfunktionen vollständig rekonstruiert wird.

wobei $t \in [0, 1)$. Für $j = N - 1$ erhält man im Intervall $[2N - 2, 2N - 1)$

$$\mathcal{T}_{\tilde{w}} f(2N - 2 + t) = \sum_{r=0}^{N-1} (-1)^{r+N} (\tilde{w}_{2N-2}(2r + 2 - t)f(2r + 2 - t) - \tilde{w}_{2N-2}(2r + t)f(2r + t)) \quad (3.18)$$

mit $t \in [0, 1)$. Um schließlich den Klappoperator für die Kosinus-II-Basisfunktionen im Intervall $[2N - 1, 2N]$ herzuleiten, kann ebenfalls auf die in Abschnitt 2.2.3 gewonnenen Resultate zurückgegriffen werden. Aus Gleichung (2.36), die noch nicht an einen bestimmten Typ trigonometrischer Funktionen gebunden ist, erhält man für $t \in [0, 1)$

$$\begin{aligned} \mathcal{T}_{\tilde{w}} f(2N - t) &= \sum_{r=0}^{N-1} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r + 2 - \tau) \tilde{w}_{2N-1}(2r + 2 - \tau) \\ &\quad \cdot \theta_{2N-1}^k(2r + 2 - \tau) d\tau \theta_{2N-1}^k(2N - t) \\ &\quad + \sum_{r=0}^{N-1} \sum_{k \in \mathbb{N}_0} \int_{-1}^0 f(2r - \tau) \tilde{w}_{2N-1}(2r - \tau) \theta_{2N-1}^k(2r - \tau) d\tau \theta_{2N-1}^k(2N - t). \end{aligned} \quad (3.19)$$

Durch Ausnutzung der Beziehung in Gleichung (2.40) folgt

$$\begin{aligned} \mathcal{T}_{\tilde{w}} f(2N - t) &= \sum_{r=0}^{N-1} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r + 2 - \tau) \tilde{w}_{2N-1}(2r + 2 - \tau) \\ &\quad \cdot \theta_{2N-1}^k(2r + 2 - \tau) d\tau \theta_{2N-1}^k(2N - t) \\ &\quad + \sum_{r=0}^{N-1} \sum_{k \in \mathbb{N}_0} \int_0^1 f(2r + \tau) \tilde{w}_{2N-1}(2r + \tau) \theta_{2N-1}^k(2r + \tau) d\tau \theta_{2N-1}^k(2N - t). \end{aligned} \quad (3.20)$$

Aufgrund der Periodizitätseigenschaften der Kosinus-II-Funktionen gelten die Zusammenhänge

$$\theta_{2N-1}^k(2r + 2 - t) = \theta_{2N-1}^k(2N - t) = \theta_{2N-1}^k(2r + t) = \theta_1^k(t). \quad (3.21)$$

Nutzt man weiterhin die Orthonormalität der Funktionen $\chi_{[0,1)} \theta_1^k$ und den hieraus resultierenden Zusammenhang nach Gleichung (2.43) aus, so ergibt sich im Intervall $[2N - 1, 2N]$ der Kosinus-II-Basis das geklappte Teilsignal

$$\mathcal{T}_{\tilde{w}} f(2N - t) = \sum_{r=0}^{N-1} (\tilde{w}_{2N-1}(2r + 2 - t)f(2r + 2 - t) + \tilde{w}_{2N-1}(2r + t)f(2r + t)). \quad (3.22)$$

Fasst man die Resultate der Gleichungen (3.16 - 3.18) und (3.22) zusammen, so erhält man zur Analyse von Signalen f , die auf einem Intervall $[0, 2N]$ definiert sind, die Realisierung des

Klappoperators für Kosinus-IV-Kosinus-II-Basen als Matrix-Vektor-Produkt

$$\begin{pmatrix} \mathcal{T}_{\tilde{w}} f(2j+t) \\ \mathcal{T}_{\tilde{w}} f(2j+2-t) \end{pmatrix} = \begin{cases} \sum_{r=0}^{N-1} (-1)^{r+j} \begin{pmatrix} \tilde{w}_{2j}(2r+t) & -\tilde{w}_{2j}(2r+2-t) \\ \tilde{w}_{2j+1}(2r+t) & \tilde{w}_{2j+1}(2r+2-t) \end{pmatrix} \cdot \begin{pmatrix} f(2r+t) \\ f(2r+2-t) \end{pmatrix}, & \text{für } j = 0, 1, \dots, N-2, \\ \sum_{r=0}^{N-1} \begin{pmatrix} -(-1)^{N+r} \tilde{w}_{2N-2}(2r+t) & (-1)^{N+r} \tilde{w}_{2N-2}(2r+2-t) \\ \tilde{w}_{2N-1}(2r+t) & \tilde{w}_{2N-1}(2r+2-t) \end{pmatrix} \cdot \begin{pmatrix} f(2r+t) \\ f(2r+2-t) \end{pmatrix}, & \text{für } j = N-1 \end{cases} \quad (3.23)$$

mit $t \in [0, 1)$. Das Matrix-Vektor-Produkt ist in Abbildung 3.2 dargestellt. Wie in Kapitel 2 herausgearbeitet wurde, besitzen die dualen Fenster der im Rahmen dieser Arbeit betrachteten mehrfach überlappenden Basen auf der reellen Achse keinen beschränkten Träger. Für die Verarbeitung von Signalen auf dem Intervall müsste somit für jedes Teilintervall $j = 0, 1, \dots, L$ ein duales Fenster $\tilde{w}_j \in L([0, L])$ konstruiert werden, wobei der Träger eines dualen Fensters genauso lang ist wie der Träger des Signals. Die Transformationsmatrizen der Klappoperatoren sind folglich voll besetzt. Die explizite Realisierung des Klappoperators als Matrix-Vektor-Produkt ist sowohl bei der Berechnung der dualen Fenster in Abhängigkeit von der Trägerlänge des Signals als auch bei der Klappoperation mit diesen Fenstern mit einem hohen rechnerischen Aufwand verbunden. Lösungsansätze für dieses Problem werden in Kapitel 4 vorgestellt.

3.3.3 Aufklappoperatoren für Kosinus-IV-Kosinus-II-Basen

Analog zu den Herleitungen in Abschnitt 2.2.4 wird angenommen, dass das geklappte Signal $f_{\mathcal{T}} \in L^2_{loc}([0, 2N])$ durch die stückweise stetigen Hilfsfunktionen $\chi_{[j, j+1)} h_j$, $j = 0, 1, \dots, 2N-1$, dargestellt wird (vgl. Abbildung 2.9). Die Hilfsfunktionen ergeben sich aus der Linearkombination der orthonormalen Basisfunktionen $\chi_{[j, j+1)} \theta_j^k$, wobei die Wichtungskoeffizienten die Transformationskoeffizienten $c_{j,k}$ sind. Für das geklappte Signal gelten daher die Beziehungen

$$f_{\mathcal{T}} = \sum_{j=0}^{2N-1} \chi_{[j, j+1)} h_j = \sum_{j=0}^{2N-1} \chi_{[j, j+1)} \sum_{k \in \mathbb{N}_0} c_{j,k} \theta_j^k. \quad (3.24)$$

Ausgehend von Gleichung (3.2) erhält man die Beschreibung des aufgeklappten Signals als Linearkombination der mit den Rekonstruktionsfenstern $w_j \in L^2([0, 2N])$ gewichteten Hilfsfunktionen h_j , d.h.

$$\mathcal{U}_w f_{\mathcal{T}} = \sum_{j=0}^{2N-1} \sum_{k \in \mathbb{N}_0} c_{j,k} \psi_j^k = \sum_{j=0}^{2N-1} w_j \sum_{k \in \mathbb{N}_0} c_{j,k} \theta_j^k = \sum_{j=0}^{2N-1} w_j h_j, \quad \text{mit } h_j = \sum_{k \in \mathbb{N}_0} c_{j,k} \theta_j^k. \quad (3.25)$$

Um die Hilfsfunktionen $h_j \in L^2([0, 2N])$ zu bestimmen, sind analog zu der in Abbildung 2.9 dargestellten Vorgehensweise die stückweise gegebenen Hilfsfunktionen $\chi_{[j, j+1)} h_j$ an den Intervallgrenzen gerade oder ungerade fortzusetzen. Um aus diesen Vorbetrachtungen gut handhabbare Gleichungen zur Realisierung des Aufklappoperators für Kosinus-IV-Kosinus-II-

$$\begin{pmatrix} \mathcal{T}_{\tilde{w}} f(t) \\ \mathcal{T}_{\tilde{w}} f(2-t) \\ \mathcal{T}_{\tilde{w}} f(2+t) \\ \mathcal{T}_{\tilde{w}} f(4-t) \\ \mathcal{T}_{\tilde{w}} f(4+t) \\ \mathcal{T}_{\tilde{w}} f(2N-4-t) \\ \vdots \\ \mathcal{T}_{\tilde{w}} f(2N-4+t) \\ \mathcal{T}_{\tilde{w}} f(2N-2-t) \\ \mathcal{T}_{\tilde{w}} f(2N-2+t) \\ \mathcal{T}_{\tilde{w}} f(2N-t) \end{pmatrix} = \begin{pmatrix} \tilde{w}_0(t) & -\tilde{w}_0(2-t) & -\tilde{w}_0(2+t) & \tilde{w}_0(4-t) & (-1)^N \tilde{w}_0(2N-4+t) & (-1)^{N-1} \tilde{w}_0(2N-2-t) & (-1)^N \tilde{w}_0(2N-t) \\ \tilde{w}_1(t) & \tilde{w}_1(2-t) & -\tilde{w}_1(2+t) & -\tilde{w}_1(4-t) & (-1)^N \tilde{w}_1(2N-4+t) & (-1)^{N-1} \tilde{w}_1(2N-2-t) & (-1)^{N-1} \tilde{w}_1(2N-t) \\ -\tilde{w}_2(t) & \tilde{w}_2(2-t) & \tilde{w}_2(2+t) & -\tilde{w}_2(4-t) & (-1)^{N-1} \tilde{w}_2(2N-4+t) & (-1)^N \tilde{w}_2(2N-2-t) & (-1)^{N-1} \tilde{w}_2(2N-t) \\ -\tilde{w}_3(t) & -\tilde{w}_3(2-t) & \tilde{w}_3(2+t) & \tilde{w}_3(4-t) & (-1)^{N-1} \tilde{w}_3(2N-4+t) & (-1)^{N-1} \tilde{w}_3(2N-2-t) & (-1)^N \tilde{w}_3(2N-t) \\ \tilde{w}_4(t) & -\tilde{w}_4(2-t) & -\tilde{w}_4(2+t) & \tilde{w}_4(4-t) & (-1)^N \tilde{w}_4(2N-4+t) & (-1)^{N-1} \tilde{w}_4(2N-2-t) & (-1)^{N-1} \tilde{w}_4(2N-t) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (-1)^{N-1} \tilde{w}_{2N-4}(t) & (-1)^{N-1} \tilde{w}_{2N-4}(2-t) & (-1)^N \tilde{w}_{2N-4}(2+t) & (-1)^N \tilde{w}_{2N-4}(4-t) & -\tilde{w}_{2N-4}(2N-4+t) & -\tilde{w}_{2N-4}(2N-2-t) & \tilde{w}_{2N-4}(2N-t) \\ (-1)^N \tilde{w}_{2N-4}(t) & (-1)^{N-1} \tilde{w}_{2N-4}(2-t) & (-1)^N \tilde{w}_{2N-4}(2+t) & (-1)^N \tilde{w}_{2N-4}(4-t) & \tilde{w}_{2N-4}(2N-4+t) & -\tilde{w}_{2N-4}(2N-2-t) & \tilde{w}_{2N-4}(2N-t) \\ (-1)^N \tilde{w}_{2N-3}(t) & (-1)^N \tilde{w}_{2N-3}(2-t) & (-1)^{N-1} \tilde{w}_{2N-3}(2+t) & (-1)^{N-1} \tilde{w}_{2N-3}(4-t) & \tilde{w}_{2N-3}(2N-4+t) & -\tilde{w}_{2N-3}(2N-2-t) & -\tilde{w}_{2N-3}(2N-t) \\ (-1)^{N-1} \tilde{w}_{2N-3}(t) & (-1)^{N-1} \tilde{w}_{2N-3}(2-t) & (-1)^N \tilde{w}_{2N-3}(2+t) & (-1)^N \tilde{w}_{2N-3}(4-t) & -\tilde{w}_{2N-3}(2N-4+t) & \tilde{w}_{2N-3}(2N-2-t) & -\tilde{w}_{2N-3}(2N-t) \\ (-1)^{N-1} \tilde{w}_{2N-2}(t) & (-1)^N \tilde{w}_{2N-2}(2-t) & (-1)^{N-1} \tilde{w}_{2N-2}(2+t) & (-1)^{N-1} \tilde{w}_{2N-2}(4-t) & -\tilde{w}_{2N-2}(2N-4+t) & \tilde{w}_{2N-2}(2N-2-t) & -\tilde{w}_{2N-2}(2N-t) \\ +\tilde{w}_{2N-1}(t) & +\tilde{w}_{2N-1}(2-t) & +\tilde{w}_{2N-1}(2+t) & +\tilde{w}_{2N-1}(4-t) & +\tilde{w}_{2N-1}(2N-4+t) & +\tilde{w}_{2N-1}(2N-2-t) & +\tilde{w}_{2N-1}(2N-t) \end{pmatrix} \cdot \begin{pmatrix} f(t) \\ f(2-t) \\ f(2+t) \\ f(4-t) \\ f(4+t) \\ f(2N-4-t) \\ \vdots \\ f(2N-4+t) \\ f(2N-2-t) \\ f(2N-2+t) \\ f(2N-t) \end{pmatrix}$$

Abbildung 3.2: Analyse von Signalen, die nur auf einem Intervall $[0, 2N]$ definiert sind; Darstellung der Klappoperation für mehrfach überlappende Kosinus-IV-Kosinus-II-Basen als Matrix-Vektor-Produkt, $t \in [0, 1)$.

Basen zu entwickeln, werden für die Rekonstruktion von Signalen $f \in L^2([0, 2N])$ im Folgenden die Basisfunktionen

$$\psi_{2j} = w_{2j}\theta_{2j}^k, \quad \psi_{2j+1} = w_{2j+1}\theta_{2j+1}^k, \quad w_{2j}, w_{2j+1} \in L^2([0, 2N]), \quad j = 0, 1, \dots, N-1, \quad k \in \mathbb{N}_0, \quad (3.26)$$

sowie die aufgeklappten Teilsignale $\mathcal{U}_w f_{\mathcal{T}}(2r+t)$ und $\mathcal{U}_w f_{\mathcal{T}}(2r+2-t)$ für $t \in [0, 1)$ und $r = 0, 1, \dots, N-1$ betrachtet. Der Aufklappoperator kann dann nach Gleichung (3.25) für $t \in [0, 1)$ durch

$$\mathcal{U}_w f_{\mathcal{T}}(2r+t) = \sum_{j=0}^{N-1} w_{2j}(2r+t)h_{2j}(2r+t) + w_{2j+1}(2r+t)h_{2j+1}(2r+t), \quad (3.27)$$

$$\mathcal{U}_w f_{\mathcal{T}}(2r+2-t) = \sum_{j=0}^{N-1} w_{2j}(2r+2-t)h_{2j}(2r+2-t) + w_{2j+1}(2r+2-t)h_{2j+1}(2r+2-t) \quad (3.28)$$

beschrieben werden. Aufgrund der in Abschnitt (2.2.3) angegebenen Periodizitäts- und Antisymmetrieeigenschaften der Kosinus-IV-Funktionen gelten für die Hilfsfunktionen h_{2j} und h_{2j+1} für $j = 0, 1, \dots, N-2$ die Beziehungen

$$\begin{aligned} h_{2j}(2r+t) &= (-1)^{j+r} f_{\mathcal{T}}(2j+t), & h_{2j+1}(2r+t) &= (-1)^{j+r} f_{\mathcal{T}}(2j+2-t), \\ h_{2j}(2r+2-t) &= -(-1)^{j+r} f_{\mathcal{T}}(2j+t), & h_{2j+1}(2r+2-t) &= (-1)^{j+r} f_{\mathcal{T}}(2j+2-t), \end{aligned} \quad (3.29)$$

sowie für $j = N-1$ die Beziehungen

$$h_{2N-2}(2r+t) = -(-1)^{N+r} f_{\mathcal{T}}(2N-2+t), \quad h_{2N-2}(2r+2-t) = (-1)^{N+r} f_{\mathcal{T}}(2N-2+t). \quad (3.30)$$

Die im rechten Randintervall eingeführten Kosinus-II-Funktionen sind 2-periodisch und es gelten somit die Beziehungen

$$h_{2N-1}(2r+t) = f_{\mathcal{T}}(2N-t), \quad h_{2N-1}(2r+2-t) = f_{\mathcal{T}}(2N-t). \quad (3.31)$$

Durch Einsetzen in die Gleichungen (3.27) und (3.28) erhält man für $r = 0, 1, \dots, N-1$ und $t \in [0, 1)$ das aufgeklappte Signal aus

$$\begin{aligned} \mathcal{U}_w f_{\mathcal{T}}(2r+t) &= \sum_{j=0}^{N-2} (-1)^{j+r} (w_{2j}(2r+t)f_{\mathcal{T}}(2j+t) + w_{2j+1}(2r+t)f_{\mathcal{T}}(2j+2-t)) \\ &\quad - (-1)^{N+r} w_{2N-2}(2r+t)f_{\mathcal{T}}(2N-2+t) + w_{2N-1}(2r+t)f_{\mathcal{T}}(2N-t), \end{aligned} \quad (3.32)$$

$$\begin{aligned} \mathcal{U}_w f_{\mathcal{T}}(2r+2-t) &= \\ &\sum_{j=0}^{N-2} (-1)^{j+r} (-w_{2j}(2r+2-t)f_{\mathcal{T}}(2j+t) + w_{2j+1}(2r+2-t)f_{\mathcal{T}}(2j+2-t)) \\ &\quad + (-1)^{N+r} w_{2N-2}(2r+2-t)f_{\mathcal{T}}(2N-2+t) + w_{2N-1}(2r+2-t)f_{\mathcal{T}}(2N-t). \end{aligned} \quad (3.33)$$

Hieraus ergibt sich die Realisierung des Aufklappoperators für Kosinus-IV-Kosinus-II-Basen auf dem Intervall $[0, 2N]$ als Matrix-Vektor-Produkt $\vec{f} = \mathbf{T}_u \cdot \vec{f}_{\mathcal{T}}$ mit

$$\begin{aligned} &\begin{pmatrix} \mathcal{U}_w f_{\mathcal{T}}(2r+t) \\ \mathcal{U}_w f_{\mathcal{T}}(2r+2-t) \end{pmatrix} \\ &= \sum_{j=0}^{N-2} (-1)^{j+r} \begin{pmatrix} w_{2j}(2r+t) & w_{2j+1}(2r+t) \\ -w_{2j}(2r+2-t) & w_{2j+1}(2r+2-t) \end{pmatrix} \begin{pmatrix} f_{\mathcal{T}}(2j+t) \\ f_{\mathcal{T}}(2j+2-t) \end{pmatrix} \\ &\quad + \begin{pmatrix} -(-1)^{N+r} w_{2N-2}(2r+t) & w_{2N-1}(2r+t) \\ (-1)^{N+r} w_{2N-2}(2r+2-t) & w_{2N-1}(2r+2-t) \end{pmatrix} \begin{pmatrix} f_{\mathcal{T}}(2N-2+t) \\ f_{\mathcal{T}}(2N-t) \end{pmatrix} \end{aligned} \quad (3.34)$$

für $t \in [0, 1)$, das in Abbildung 3.3 dargestellt ist. Für die Untersuchungen zur Leistungsfähigkeit von überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression werden im Folgenden in den inneren Intervallen die Fensterfunktionen w_{2j+1} und w_{2j} möglichst als Translate einer Fensterfunktion w angenommen, wobei die Träger dieser Fensterfunktionen vollständig in $[0, 2N]$ liegen. Die im Rahmen dieser Arbeit untersuchten Fensterfunktionen für die inneren Intervalle wurden in Abschnitt 2.3 vorgestellt. In den Randintervallen kommen hingegen in Abhängigkeit vom Überlappungsfaktor und vom Fensterentwurf jeweils ein oder mehrere Randfenster w_{l_i} am linken Signalrand und w_{r_i} am rechten Signalrand zum Einsatz. Diese im Rahmen dieser Arbeit neu entwickelten Fensterfunktionen werden in Abschnitt 3.5 eingeführt. Aus Gleichung (3.34) erhält man z.B. bei der Dreifachüberlappung für den Fall, dass an den Signalrändern jeweils zwei Randfenster Anwendung finden (vgl. Abschnitt 3.5.3), am linken Signalrand unter Nutzung der Randfenster w_{l_0} und w_{l_1} die aufgeklappten Teilsignale aus

$$\begin{aligned}\mathcal{U}_w f_{\mathcal{T}}(t) &= w_{l_0}(t) f_{\mathcal{T}}(t) + w_{l_1}(-1+t) f_{\mathcal{T}}(2-t), \\ \mathcal{U}_w f_{\mathcal{T}}(2-t) &= -w_{l_0}(2-t) f_{\mathcal{T}}(t) + w_{l_1}(1-t) f_{\mathcal{T}}(2-t) + w(-t) f_{\mathcal{T}}(2+t), \\ \mathcal{U}_w f_{\mathcal{T}}(2+t) &= -w_{l_1}(1+t) f_{\mathcal{T}}(2-t) + w(t) f_{\mathcal{T}}(2+t) + w(-1+t) f_{\mathcal{T}}(4-t),\end{aligned}\tag{3.35}$$

wobei $t \in [0, 1)$. In den inneren Intervallen kommt für den Aufklappoperator die Beschreibung

$$\begin{aligned}\mathcal{U}_w f_{\mathcal{T}}(2j-t) &= -w(2-t) f_{\mathcal{T}}(2j-2+t) + w(1-t) f_{\mathcal{T}}(2j-t) + w(-t) f_{\mathcal{T}}(2j+t), \\ \mathcal{U}_w f_{\mathcal{T}}(2j+t) &= -w(1+t) f_{\mathcal{T}}(2j-t) + w(t) f_{\mathcal{T}}(2j+t) + w(-1+t) f_{\mathcal{T}}(2j+2-t)\end{aligned}\tag{3.36}$$

zur Anwendung, wobei $t \in [0, 1)$, $j = 2, 3, \dots, N-2$. Am rechten Signalrand erfolgt in den Randintervallen unter Verwendung der Randfenster w_{r_0} und w_{r_1} die Berechnung des aufgeklappten Signals für $t \in [0, 1)$ nach

$$\begin{aligned}\mathcal{U}_w f_{\mathcal{T}}(2N-2-t) &= -w(2-t) f_{\mathcal{T}}(2N-4+t) + w(1-t) f_{\mathcal{T}}(2N-2-t) \\ &\quad + w_{r_1}(-t) f_{\mathcal{T}}(2N-2+t), \\ \mathcal{U}_w f_{\mathcal{T}}(2N-2+t) &= -w(1+t) f_{\mathcal{T}}(2N-2-t) + w_{r_1}(t) f_{\mathcal{T}}(2N-2+t) \\ &\quad + w_{r_0}(-1+t) f_{\mathcal{T}}(2N-t), \\ \mathcal{U}_w f_{\mathcal{T}}(2N-t) &= -w_{r_1}(2-t) f_{\mathcal{T}}(2N-2+t) + w_{r_0}(1-t) f_{\mathcal{T}}(2N-t).\end{aligned}\tag{3.37}$$

Die Realisierung des Aufklappoperators zur Transformation von Signalen auf dem Intervall $[0, 2N]$ ist in Abbildung 3.4 am Beispiel der Dreifachüberlappung auf Grundlage der Gleichungen (3.35 - 3.37) dargestellt¹. Es wird sichtbar, dass die Transformationsmatrix des Aufklappoperators eine Bandstruktur aufweist. Des Weiteren sind im Anhang A die Matrix-Vektor-Produkte zur Realisierung des Aufklappoperators auf dem Intervall $[0, 2N]$ für die Zweifach- und für die Vierfachüberlappung veranschaulicht. Hierbei wird berücksichtigt, dass für gerade Überlappungsfaktoren in Abhängigkeit vom Argument t Nullen in den äußeren Nebendiagonalen auftreten. In Analogie zur Zweifach- und Vierfachüberlappung lässt sich auch für sechsfach überlappende Fensterfunktionen eine äquivalente Beschreibung aus Gleichung (3.34) ableiten.

3.4 Riesz-Stabilität

Die Frage, ob das System $\{\psi_j^k : \psi_j^k \in L^2([0, 2N]), j = 0, 1, \dots, 2N-1, k \in \mathbb{N}_0\}$ von überlappenden Kosinus-IV-Kosinus-II-Funktionen eine Riesz-Basis auf dem Intervall $[0, 2N]$

¹Für den Fall, dass bei der Dreifachüberlappung an den Signalgrenzen jeweils nur ein Randfenster zur Anwendung kommt (vgl. Abschnitt 3.5.2), ist der Klappoperator in Abbildung A.7 dargestellt.

$$\begin{pmatrix} \mathcal{U}_w f_T(t) \\ \mathcal{U}_w f_T(2-t) \\ \mathcal{U}_w f_T(2+t) \\ \mathcal{U}_w f_T(4-t) \\ \mathcal{U}_w f_T(4+t) \\ \cdot \\ \cdot \\ \cdot \\ \mathcal{U}_w f_T(2N-4-t) \\ \mathcal{U}_w f_T(2N-4+t) \\ \mathcal{U}_w f_T(2N-2-t) \\ \mathcal{U}_w f_T(2N-2+t) \\ \mathcal{U}_w f_T(2N-t) \end{pmatrix} = \begin{pmatrix} w_0(t) & w_1(t) & -w_2(t) & -w_3(t) & (-1)^N w_{2N-4}(t) & (-1)^N w_{2N-3}(t) & (-1)^{N-1} w_{2N-2}(t) & +w_{2N-1}(t) \\ -w_0(2-t) & w_1(2-t) & w_2(2-t) & -w_3(2-t) & (-1)^{N-1} w_{2N-4}(2-t) & (-1)^N w_{2N-3}(2-t) & (-1)^N w_{2N-2}(2-t) & +w_{2N-1}(2-t) \\ -w_0(2+t) & -w_1(2+t) & w_2(2+t) & w_3(2+t) & (-1)^{N-1} w_{2N-4}(2+t) & (-1)^{N-1} w_{2N-3}(2+t) & (-1)^N w_{2N-2}(2+t) & +w_{2N-1}(2+t) \\ w_0(4-t) & -w_1(4-t) & -w_2(4-t) & w_3(4-t) & (-1)^N w_{2N-4}(4-t) & (-1)^{N-1} w_{2N-3}(4-t) & (-1)^{N-1} w_{2N-2}(4-t) & +w_{2N-1}(4-t) \\ w_0(4+t) & w_1(4+t) & -w_2(4+t) & -w_3(4+t) & (-1)^N w_{2N-4}(4+t) & (-1)^N w_{2N-3}(4+t) & (-1)^{N-1} w_{2N-2}(4+t) & +w_{2N-1}(4+t) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathcal{U}_w f_T(2N-4-t) & (-1)^{N-1} w_1(2N-4-t) & (-1)^{N-1} w_2(2N-4-t) & (-1)^N w_3(2N-4-t) & w_{2N-4}(2N-4-t) & -w_{2N-3}(2N-4-t) & -w_{2N-2}(2N-4-t) & +w_{2N-1}(2N-4-t) \\ \mathcal{U}_w f_T(2N-4+t) & (-1)^N w_1(2N-4+t) & (-1)^{N-1} w_2(2N-4+t) & (-1)^{N-1} w_3(2N-4+t) & w_{2N-4}(2N-4+t) & w_{2N-3}(2N-4+t) & -w_{2N-2}(2N-4+t) & +w_{2N-1}(2N-4+t) \\ \mathcal{U}_w f_T(2N-2-t) & (-1)^N w_1(2N-2-t) & (-1)^N w_2(2N-2-t) & (-1)^{N-1} w_3(2N-2-t) & -w_{2N-4}(2N-2-t) & w_{2N-3}(2N-2-t) & w_{2N-2}(2N-2-t) & +w_{2N-1}(2N-2-t) \\ \mathcal{U}_w f_T(2N-2+t) & (-1)^{N-1} w_1(2N-2+t) & (-1)^N w_2(2N-2+t) & (-1)^N w_3(2N-2+t) & -w_{2N-4}(2N-2+t) & -w_{2N-3}(2N-2+t) & w_{2N-2}(2N-2+t) & +w_{2N-1}(2N-2+t) \\ \mathcal{U}_w f_T(2N-t) & (-1)^{N-1} w_1(2N-t) & (-1)^{N-1} w_2(2N-t) & (-1)^N w_3(2N-t) & w_{2N-4}(2N-t) & -w_{2N-3}(2N-t) & -w_{2N-2}(2N-t) & +w_{2N-1}(2N-t) \end{pmatrix} \begin{pmatrix} f_T(t) \\ f_T(2-t) \\ f_T(2+t) \\ f_T(4-t) \\ f_T(4+t) \\ \cdot \\ \cdot \\ \cdot \\ f_T(2N-4-t) \\ f_T(2N-4+t) \\ f_T(2N-2-t) \\ f_T(2N-2+t) \\ f_T(2N-t) \end{pmatrix}$$

Abbildung 3.3: Rekonstruktion von Signalen, die auf ein Intervall $[0, 2N]$ begrenzt sind; Darstellung der Aufklappoperation für mehrfach überlappende Kosinus-IV-Kosinus-II-Basen als Matrix-Vektor-Produkt, $t \in [0, 1)$.

$$\begin{pmatrix} \mathcal{U}_w f \hat{r}(t) \\ \mathcal{U}_w f \hat{r}(2-t) \\ \mathcal{U}_w f \hat{r}(2+t) \\ \mathcal{U}_w f \hat{r}(4-t) \\ \mathcal{U}_w f \hat{r}(4+t) \\ \vdots \\ \mathcal{U}_w f \hat{r}(2N-4-t) \\ \mathcal{U}_w f \hat{r}(2N-4+t) \\ \mathcal{U}_w f \hat{r}(2N-2-t) \\ \mathcal{U}_w f \hat{r}(2N-2+t) \\ \mathcal{U}_w f \hat{r}(2N-t) \end{pmatrix} = \begin{pmatrix} \begin{array}{cc|cc|cc|cc|cc} w_{i_0}(t) & w_{i_1}(-1+t) & 0 & & & & & & & \\ -w_{i_0}(2-t) & w_{i_1}(1-t) & w(-t) & 0 & & & & & & \\ 0 & -w_{i_1}(1+t) & w(t) & w(-1+t) & 0 & & & & & \\ & 0 & -w(2-t) & w(1-t) & w(-t) & 0 & & & & \\ & & 0 & -w(1+t) & w(t) & w(-1+t) & 0 & & & \\ & & & \ddots & \ddots & \ddots & & & & \\ & & & & -w(2-t) & w(1-t) & w(-t) & 0 & & \\ & & & & -w(1+t) & w(t) & w(-1+t) & 0 & & \\ & & & & 0 & -w(2-t) & w(1-t) & 0 & & \\ & & & & & 0 & -w(1+t) & w(t) & w(-1+t) & 0 \\ & & & & & & 0 & -w(2-t) & w(1-t) & 0 \\ & & & & & & & 0 & -w(1+t) & 0 \\ & & & & & & & & 0 & -w_{i_1}(2-t) \end{array} & \begin{array}{cc} w_{i_1}(-t) & 0 \\ w_{i_1}(t) & \\ -w_{i_1}(2-t) & \end{array} & \begin{array}{cc} w_{i_0}(-1+t) \\ w_{i_0}(1-t) \end{array} \end{pmatrix} \cdot \begin{pmatrix} f \hat{r}(t) \\ f \hat{r}(2-t) \\ f \hat{r}(2+t) \\ f \hat{r}(4-t) \\ f \hat{r}(4+t) \\ \vdots \\ f \hat{r}(2N-4-t) \\ f \hat{r}(2N-4+t) \\ f \hat{r}(2N-2-t) \\ f \hat{r}(2N-2+t) \\ f \hat{r}(2N-t) \end{pmatrix}$$

Abbildung 3.4: Rekonstruktion von Signalen, die auf ein Intervall $[0, 2N]$ begrenzt sind; Darstellung der Aufklappoperation für dreifach überlappende Kosinus-IV-Kosinus-II-Basen als Matrix-Vektor-Produkt, $t \in [0, 1)$. Die inneren Fenster sind Translate einer Fensterfunktion $w(t)$.

bildet, lässt sich nicht mit der in Abschnitt 2.2.1 beschriebenen Methode zur Bestimmung der Riesz-Schranken unter Verwendung der Zak-Transformation beantworten (vgl. Gleichungen (2.22 - 2.25)). Die dort zugrundeliegende Voraussetzung, dass alle Fensterfunktionen w_j Translate einer einzigen Fensterfunktion w sind, ist für das in diesem Kapitel neu eingeführte System von überlappenden Kosinus-IV-Kosinus-II-Funktionen nicht erfüllt. Das Problem der Bestimmung der Riesz-Schranken lässt sich jedoch mit dem in Abschnitt 3.3.3 entwickelten Aufklappoperator bzw. mit der für diesen Operator hergeleiteten Matrix-Vektor-Beschreibung (vgl. Gleichung (3.34)) lösen.

Das System $\{\psi_j^k\}$ mit $\psi_j^k \in L^2([0, 2N])$, $j = 0, 1, \dots, 2N - 1$, $k \in N_0$, ist eine Riesz-Basis mit den Riesz-Konstanten A und B , wenn der Operator $\mathcal{U}_w : L^2([0, 2N]) \rightarrow L^2([0, 2N])$ eineindeutig ist und $0 < A \leq A_0 \leq B_0 \leq B < \infty$ mit

$$A_0 := \|\mathcal{U}_w^{-1}\|_{L^2([0, 2N]) \rightarrow L^2([0, 2N])}^{-2} \quad \text{und} \quad B_0 := \|\mathcal{U}_w\|_{L^2([0, 2N]) \rightarrow L^2([0, 2N])}^2 \quad (3.38)$$

erfüllt wird. Mit Hilfe der $2N \times 2N$ Abbildungsmatrizen der Aufklappoperatoren, die im Anhang A.2 für die untersuchten Überlappungsfaktoren¹ und in Abbildung 3.3 für allgemeinere Überlappungsfaktoren angegeben sind, lassen sich nun in Abhängigkeit von der Trägerlänge $2N$ des betrachteten Signals die größte untere Riesz-Schranke A_0 und die kleinste obere Riesz-Schranke B_0 der Kosinus-IV-Kosinus-II-Basen durch

$$\begin{aligned} B_0 &= \max_{t \in [0, 1)} \|\mathbf{T}_u(t)\|_2^2 = \max_{t \in [0, 1)} \left(\max \sigma(\mathbf{T}_u^T(t) \mathbf{T}_u(t)) \right), \\ A_0 &= \min_{t \in [0, 1)} \|\mathbf{T}_u^{-1}(t)\|_2^{-2} = \min_{t \in [0, 1)} \left(\min \sigma(\mathbf{T}_u^T(t) \mathbf{T}_u(t)) \right) \end{aligned} \quad (3.39)$$

ermitteln (vgl. Abschnitt 1.1.3). Die Koeffizienten der Abbildungsmatrizen der Aufklappoperatoren sind Funktionen der Zeit, wobei $t \in [0, 1)$. Zur Bestimmung der Riesz-Konstanten werden daher zunächst die beiden Normen $\|\mathbf{T}_u(t)\|_2^2$ und $\|\mathbf{T}_u^{-1}(t)\|_2^{-2}$ in Abhängigkeit von t dargestellt. Anschließend erfolgt durch Suche des globalen Minimums von $\|\mathbf{T}_u^{-1}(t)\|_2^{-2}$ die Bestimmung der größten unteren Riesz-Schranke A_0 und durch Suche des globalen Maximums von $\|\mathbf{T}_u(t)\|_2^2$ die Bestimmung der kleinsten oberen Riesz-Schranke B_0 .

3.5 Fenster zur Konstruktion von trigonometrischen Randbasisfunktionen

Um geeignete überlappende trigonometrische Basen auf dem Intervall für die Anwendung in der Bilddatenkompression zu entwickeln, sei gefordert, dass der Gleichanteil eines Signals mit dem Träger $\text{supp } f = [0, 2N]$ ausschließlich durch die Basisfunktionen der niedrigsten Frequenz rekonstruiert werde, d.h. nach Gleichung (3.5) und (3.6)

$$f(t) = 1 = \sum_{j=0}^{2N-1} \psi_j^0(t) = \sum_{j=0}^{2N-2} w_j(t) \sqrt{2} \cos\left(\frac{1}{2}\pi(t-j)\right) + w_{2N-1}(t), \quad \text{für } t \in [0, 2N]. \quad (3.40)$$

Für $j = 0, 1, \dots, 2N - 2$ sind die Basisfunktionen gewichtete Kosinus-IV-Funktionen. Als Resultat des in Abschnitt 3.2 erarbeiteten Lösungsvorschlags zur Darstellung von endlich langen

¹Die Abbildungsmatrix der Sechsfachüberlappung ist im Anhang A.2 nicht explizit angegeben. Sie lässt sich analog zu den dargestellten Überlappungsfaktoren aus der in Abbildung 3.3 gezeigten allgemeinen Abbildungsmatrix für beliebige Überlappungsfaktoren herauslesen (Bandbreite = 7, jeweils 3 Randfenster (vgl. Abschnitt 3.5.6)).

Signalen mit überlappenden lokalen trigonometrischen Basen werden am rechten Signalrand dem Intervall $j = 2N - 1$ Kosinus-II-Basisfunktionen zugeordnet.

Die Entwicklung von geeigneten Fensterfunktionen zur Konstruktion von Basisfunktionen für die Darstellung des endlich langen Signals in den Randintervallen geht von den inneren Fensterfunktionen aus, die auf der reellen Achse definiert sind (vgl. Abschnitt 2.3). Hierbei wird angenommen, dass die inneren Fensterfunktionen w , also die Fensterfunktionen, deren Träger vollständig im Intervall $[0, 2N]$ liegt, möglichst unverändert bleiben sollen. In Abhängigkeit vom Entwurfsansatz kann es jedoch auch erforderlich sein, dass innere Fensterfunktionen angepasst werden müssen.

3.5.1 Zweifachüberlappung

In diesem Abschnitt werden für die Zweifachüberlappung Fensterfunktionen zur Konstruktion von trigonometrischen Randbasisfunktionen eingeführt. Um am linken Signalrand im Intervall $[0, \frac{1}{2})$ eine Konstante mit den Basisfunktionen der niedrigsten Frequenz zu repräsentieren, ist die Bedingung

$$1 = \psi_0^0(t) = w_{l_0}(t)\sqrt{2}\cos\left(\frac{1}{2}\pi t\right), \quad \text{für } t \in \left[0, \frac{1}{2}\right) \quad (3.41)$$

zu erfüllen. Im Intervall $[\frac{1}{2}, \frac{3}{2})$ gilt hingegen $w_{l_0} = w$. Ausgehend von dem in Abschnitt 2.3.1 vorgestellten Malvar-Fenster w für die inneren Intervalle lässt sich durch Umformen und Einsetzen die Fensterfunktion

$$w_{l_0}(t) = \begin{cases} \frac{1}{\sqrt{2}\cos(\frac{1}{2}\pi t)}, & \text{für } t \in [0, \frac{1}{2}), \\ \sin(\frac{1}{2}\pi(t + \frac{1}{2})), & \text{für } t \in [\frac{1}{2}, \frac{3}{2}), \\ 0, & \text{sonst} \end{cases} \quad (3.42)$$

zur Konstruktion von Randbasisfunktionen am linken Signalrand herleiten. Am rechten Signalrand ist im Intervall $[2N - \frac{3}{2}, 2N - \frac{1}{2})$ für die Rekonstruktion einer Konstanten mit den Basisfunktionen der niedrigsten Frequenz der Kosinus-IV-Kosinus-II-Basen die Bedingung

$$\begin{aligned} \psi_{2N-2}^0(t) + \psi_{2N-1}^0(t) = 1 = w(t - 2N + 2)\sqrt{2}\cos\left(\frac{1}{2}\pi(t - 2N + 2)\right) \\ + w_{r_0}(t - 2N + 1), \quad \text{für } t \in \left[2N - \frac{3}{2}, 2N - \frac{1}{2}\right) \end{aligned} \quad (3.43)$$

zu erfüllen. Im Intervall $[2N - \frac{1}{2}, 2N)$, dem Kosinus-II-Funktionen zugeordnet sind, gilt die Bedingung

$$\psi_{2N-1}^0(t) = 1 = w_{r_0}(t - 2N + 1), \quad \text{für } t \in \left[2N - \frac{1}{2}, 2N\right). \quad (3.44)$$

Betrachtet man für w wieder das Malvar-Fenster, so ergibt sich durch Substitution von t mit $t + 2N - 1$ und Auflösen nach w_{r_0} zur Konstruktion von Randbasisfunktionen am rechten Signalrand die Fensterfunktion

$$w_{r_0}(t) = \begin{cases} \frac{1}{2}(1 + \cos(\pi t) + \sin(\pi t)), & \text{für } t \in [-\frac{1}{2}, \frac{1}{2}), \\ 1, & \text{für } t \in [\frac{1}{2}, 1), \\ 0, & \text{sonst.} \end{cases} \quad (3.45)$$

Die Fensterfunktionen sind in Abbildung 3.5 dargestellt. In Abbildung 3.6 wird gezeigt, wie die mit diesen Fenstern erzeugten Basen an den Signalrändern eine Konstante mit den Basisfunktionen der niedrigsten Frequenz reproduzieren.

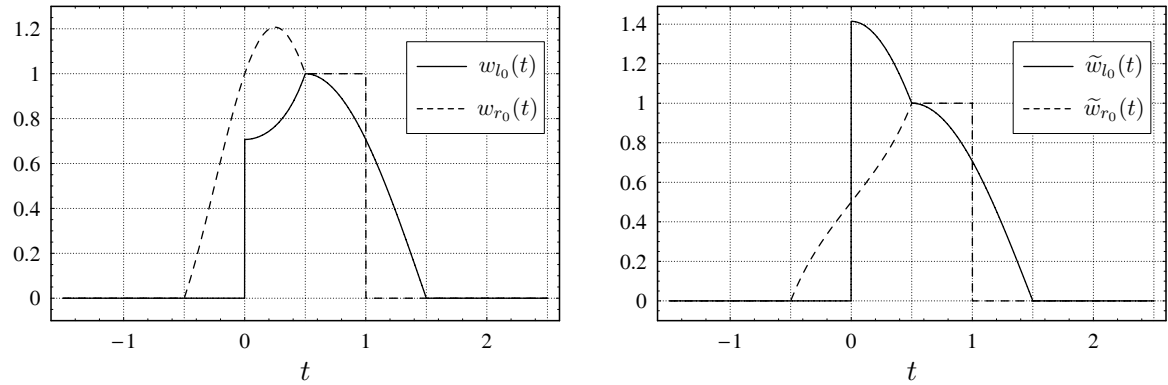


Abbildung 3.5: Randfenster (links) und duale Randfenster (rechts) zur Konstruktion von Randbasisfunktionen für die Rekonstruktion und Analyse von Signalen endlicher Länge; Verwendung des Malvar-Fensters in den inneren Intervallen.

Bei der Zweifachüberlappung besitzen die dualen Fensterfunktionen zur Signalanalyse die gleiche Trägerlänge wie die Fensterfunktionen zur Signalrekonstruktion. Daraus resultierend weist die Transformationsmatrix des Klappoperators wie die Transformationsmatrix des Aufklappoperators eine Bandstruktur mit gleicher Bandbreite auf (vgl. Abbildungen A.5 und A.6 mit A.3 und A.4). Im Gegensatz zur Mehrfachüberlappung eignet sich daher bei der Zweifachüberlappung die Beschreibung des Klappoperators als Matrix-Vektor-Produkt für eine schnelle algorithmische Realisierung. Die hierzu benötigten dualen Fensterfunktionen lassen sich aus den Dualitätsbeziehungen

$$\tilde{w}_{l_0}(t) = \begin{cases} \frac{1}{w_{l_0}(t)}, & \text{für } t \in [0, \frac{1}{2}), \\ \frac{w(t-1)}{w_{l_0}(2-t)w(t-1) + w_{l_0}(t)w(1-t)}, & \text{für } t \in [\frac{1}{2}, \frac{3}{2}) \end{cases} \quad (3.46)$$

und

$$\tilde{w}_{r_0}(t) = \begin{cases} \frac{w(1-t)}{w_{r_0}(-t)w(1+t) + w_{r_0}(t)w(1-t)}, & \text{für } t \in [-\frac{1}{2}, \frac{1}{2}), \\ \frac{1}{w_{r_0}(t)}, & \text{für } t \in [\frac{1}{2}, 1) \end{cases} \quad (3.47)$$

bestimmen. Ausgehend von den in den Gleichungen (3.42) und (3.45) eingeführten Randfenstern erhält man zur Konstruktion von Randbasisfunktionen für die Signalanalyse die dualen Fensterfunktionen

$$\tilde{w}_{l_0}(t) = \begin{cases} \sqrt{2} \cos(\frac{1}{2}\pi t), & \text{für } t \in [0, \frac{1}{2}), \\ \sin(\frac{1}{2}\pi(t + \frac{1}{2})), & \text{für } t \in [\frac{1}{2}, \frac{3}{2}), \\ 0, & \text{sonst,} \end{cases} \quad (3.48)$$

und

$$\tilde{w}_{r_0}(t) = \begin{cases} \frac{1}{2} + \frac{1}{2} \tan(\frac{1}{2}\pi t), & \text{für } t \in [-\frac{1}{2}, \frac{1}{2}), \\ 1, & \text{für } t \in [\frac{1}{2}, 1), \\ 0, & \text{sonst.} \end{cases} \quad (3.49)$$

Die dualen Randfenster zur Konstruktion von Randbasisfunktionen für die Signalanalyse sind ebenfalls in Abbildung 3.5 dargestellt.

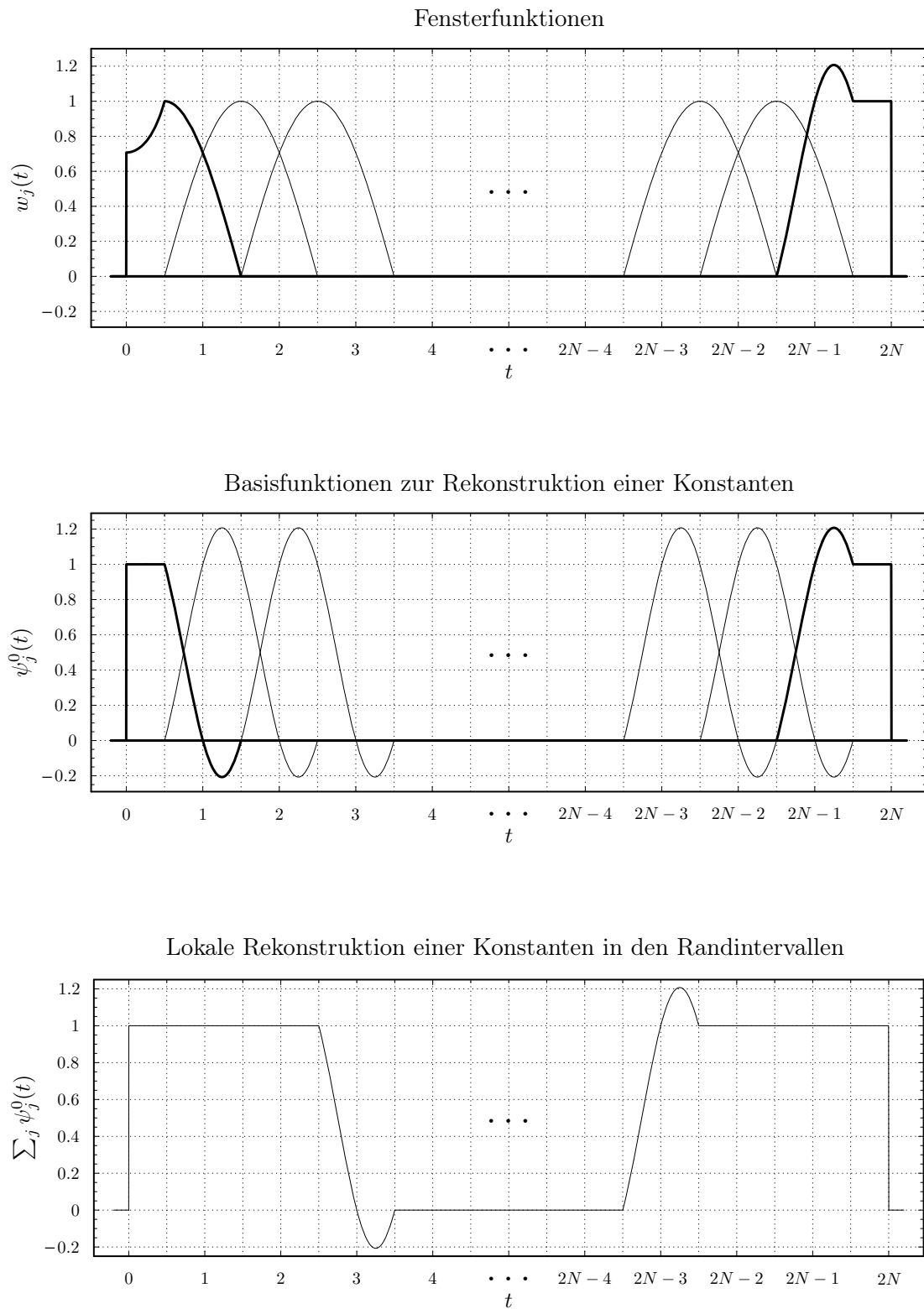


Abbildung 3.6: Zweifachüberlappung: Fenster- und Basisfunktionen zur Verarbeitung von endlich langen Signalen, lokale Rekonstruktion von Konstanten.

Durch die Verwendung der in diesem Abschnitt eingeführten Randbasisfunktionen zur Repräsentation von Signalen, die nur auf einem endlich langen Intervall definiert sind, erfüllt die Signaltransformation nicht mehr die strenge Bedingung der Orthogonalität. Mit den Riesz-Konstanten $A_0 = 0.5$ und $B_0 = 2$ wird jedoch die Bedingung der Riesz-Stabilität erfüllt und man erhält eine Konditionszahl von $\kappa = 2$. Die Bestimmung der Riesz-Schranken ist am Ende von Abschnitt 3.5.6 in Abbildung 3.21 veranschaulicht.

3.5.2 Dreifachüberlappung – Rekonstruktion von Konstanten

Ausgehend von einer gegebenen dreifach überlappenden Fensterfunktion w , die den inneren Intervallen zugeordnet ist, wird zur Konstruktion von Kosinus-IV-Randbasisfunktionen am linken Signalrand die Fensterfunktion

$$w_{l_0}(t) := \begin{cases} \frac{1 - \sqrt{2}w(t-1)\sin(\frac{1}{2}\pi t)}{\sqrt{2}\cos(\frac{1}{2}\pi t)}, & \text{für } t \in [0, 1), \\ w(t), & \text{für } t \in [1, 2), \\ 0, & \text{sonst} \end{cases} \quad (3.50)$$

definiert. Zur Konstruktion von Kosinus-II-Randbasisfunktionen am rechten Signalrand wird die Fensterfunktion

$$w_{r_0}(t) := \begin{cases} 1 + w(t+1)\sqrt{2}\sin(\frac{1}{2}\pi t) + w(t+2)\sqrt{2}\cos(\frac{1}{2}\pi t), & \text{für } t \in [-1, 0), \\ 1 + w(t+1)\sqrt{2}\sin(\frac{1}{2}\pi t), & \text{für } t \in [0, 1), \\ 0, & \text{sonst} \end{cases} \quad (3.51)$$

definiert. Die beiden Randfenster sind in Abbildung 3.7 für die in Abschnitt 2.3.2 beschriebene Klasse von dreifach überlappenden Fenstern am Beispiel von $w = w^{(3)}$ dargestellt. Dass die Basen auf dem Intervall unter Verwendung der Randbasisfunktionen, die mit den Fensterfunktionen nach Gleichung (3.50) und (3.51) konstruiert werden, eine Konstante lokal auch in den Randintervallen ausschließlich mit Basisfunktionen ψ_j^0 reproduzieren, lässt sich durch nachfolgende Umformungen zeigen. Hierzu ist die Signalrekonstruktion in den Intervallen $[0, 1)$,

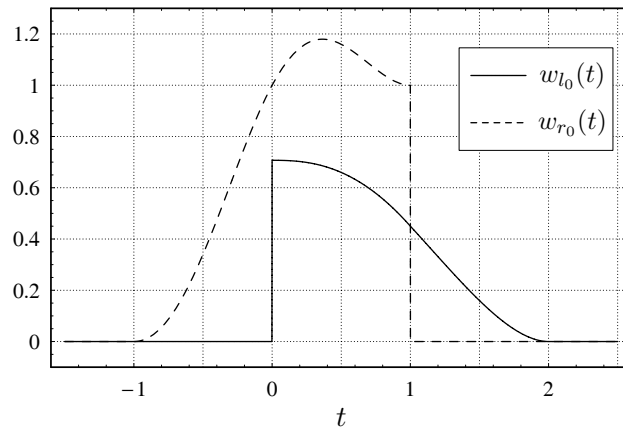


Abbildung 3.7: Dreifachüberlappung: Fensterfunktionen zur Konstruktion von Randbasisfunktionen für die Darstellung von Signalen mit einer endlichen Länge, wobei $w = w^{(3)}$.

$[2N-2, 2N-1)$ und $[2N-1, 2N)$ zu betrachten. In allen anderen Intervallen $[j, j+1)$ für $j = 1, 2, \dots, 2N-3$ überlagern sich hingegen ausschließlich Bestandteile bzw. Abschnitte von Basisfunktionen, die nur durch die Wichtung mit der in Gleichung (2.64) beschriebenen dreifach überlappenden Fensterfunktion erzeugt werden. In den inneren Intervallen ist somit bereits die Bedingung der Rekonstruktion einer Konstanten mit den Basisfunktionen ψ_j^0 erfüllt. Am linken Signalrand überlagern sich im Intervall $[0, 1)$ die Basisfunktionen

$$\psi_0^0(t) + \psi_1^0(t) = w_{l_0}(t)\sqrt{2}\cos\left(\frac{1}{2}\pi t\right) + w(t-1)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-1)\right), \quad \text{für } t \in [0, 1). \quad (3.52)$$

Ersetzt man w_{l_0} nach Gleichung (3.50), so erhält man für $t \in [0, 1)$

$$\psi_0^0(t) + \psi_1^0(t) = \frac{1 - w(t-1)\sqrt{2}\sin\left(\frac{1}{2}\pi t\right)}{\sqrt{2}\cos\left(\frac{1}{2}\pi t\right)}\sqrt{2}\cos\left(\frac{1}{2}\pi t\right) + w(t-1)\sqrt{2}\sin\left(\frac{1}{2}\pi t\right) = 1. \quad (3.53)$$

Die Bedingung der Rekonstruktion einer Konstanten durch Überlagerung der Basisfunktionen ψ_j^0 ist somit in diesem Randintervall erfüllt. Am rechten Signalrand überlagern sich im Intervall $[2N-2, 2N-1)$ von der Kosinus-IV-Kosinus-II-Basis die Funktionen

$$\begin{aligned} \psi_{2N-3}^0(t) + \psi_{2N-2}^0(t) + \psi_{2N-1}^0(t) &= w(t-2N+3)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+3)\right) \\ &+ w(t-2N+2)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+2)\right) + w_{r_0}(t-2N+1), \quad \text{für } t \in [2N-2, 2N-1). \end{aligned} \quad (3.54)$$

Durch Substitution von $w_{r_0}(t-2N+1)$ für $t \in [2N-2, 2N-1)$ mit Gleichung (3.51) erhält man

$$\begin{aligned} \psi_{2N-3}^0(t) + \psi_{2N-2}^0(t) + \psi_{2N-1}^0(t) &= w(t-2N+3)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+3)\right) \\ &+ w(t-2N+2)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+2)\right) + 1 + w(t-2N+2)\sqrt{2}\sin\left(\frac{1}{2}\pi(t-2N+1)\right) \\ &+ w(t-2N+3)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+1)\right), \quad \text{für } t \in [2N-2, 2N-1). \end{aligned} \quad (3.55)$$

Die Ausnutzung der trigonometrischen Beziehungen

$$\sin\left(\frac{1}{2}\pi(t-2N+1)\right) = -\cos\left(\frac{1}{2}\pi(t-2N+2)\right) \quad (3.56)$$

und

$$\cos\left(\frac{1}{2}\pi(t-2N+3)\right) = -\cos\left(\frac{1}{2}\pi(t-2N+1)\right) \quad (3.57)$$

führt zu $\psi_{2N-3}^0 + \psi_{2N-2}^0 + \psi_{2N-1}^0 = 1$. Auch in diesem Intervall ist daher die Konstruktion einer Konstanten mit den Basisfunktionen ψ_j^0 nachgewiesen. Bleibt abschließend zu zeigen, dass im Intervall $[2N-1, 2N)$ ebenfalls die Forderung der Rekonstruktion einer Konstanten mit den Basisfunktionen ψ_j^0 erfüllt ist. In diesem Intervall wird die Überlagerung der Basisfunktionen

$$\begin{aligned} \psi_{2N-2}^0(t) + \psi_{2N-1}^0(t) &= w(t-2N+2)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+2)\right) \\ &+ w_{r_0}(t-2N+1), \quad \text{für } t \in [2N-1, 2N) \end{aligned} \quad (3.58)$$

betrachtet. Die Substitution von $w_{r_0}(t - 2N + 1)$ für $t \in [2N - 1, 2N)$ mit Gleichung (3.51) ergibt

$$\begin{aligned} \psi_{2N-2}^0(t) + \psi_{2N-1}^0(t) &= w(t - 2N + 2)\sqrt{2} \cos\left(\frac{1}{2}\pi(t - 2N + 2)\right) + 1 \\ &\quad + w(t - 2N + 2)\sqrt{2} \sin\left(\frac{1}{2}\pi(t - 2N + 1)\right), \quad \text{für } t \in [2N - 1, 2N). \end{aligned} \quad (3.59)$$

Die Anwendung der in Gleichung (3.56) angegebenen trigonometrischen Beziehung führt zu dem Resultat $\psi_{2N-2}^0 + \psi_{2N-1}^0 = 1$, womit auch im Intervall $[2N - 1, 2N)$ die Repräsentation einer Konstanten mit den Basisfunktionen ψ_j^0 nachgewiesen ist.

Die Konstruktion von Basisfunktionen zur Darstellung einer Konstanten sowie die lokale Repräsentation einer Konstanten in den Randintervallen wird in Abbildung 3.8 gezeigt. Des Weiteren sind in Abbildung D.1 (Anhang D) für eine ausgewählte Anzahl von Intervallen an den Signalrändern die dualen Fensterfunktionen zur Konstruktion von Basisfunktionen für die Signalanalyse dargestellt. Die dualen Fenster lassen sich numerisch durch Invertierung der Transformationsmatrix des Aufklappoperators (vgl. Abbildung A.7) und Koeffizientenvergleich mit der Transformationsmatrix des Klappoperators (vgl. Abbildung 3.2) bestimmen. Für dreifach überlappende Kosinus-IV-Kosinus-II-Basen auf dem Intervall unter Verwendung der in diesem Abschnitt neu eingeführten Randfenster werden in Tabelle 3.1 Riesz-Konstanten und Konditionszahlen in Abhängigkeit von den zugrunde liegenden inneren Fensterfunktionen $w^{(\cdot)}$ angegeben. Es sei darauf hingewiesen, dass die Riesz-Schranken auch von der Anzahl der Teilintervalle abhängen, in die das auf einem endlich langen Intervall definierte Signal eingeteilt wird (vgl. Abschnitt 3.4). Tabelle 3.1 zeigt jedoch, dass sich bei Erhöhung von N die Riesz-Schranken nur noch geringfügig verändern. Abbildung 3.9 veranschaulicht die Bestimmung der Riesz-Schranken und es ist zu erkennen, dass die Bedingung der Riesz-Stabilität erfüllt wird. Im Vergleich mit den Basisfunktionen zur Darstellung von Signalen, die auf der reellen Achse definiert sind, verschlechtern sich durch die Verwendung von Randbasisfunktionen die Riesz-Konstanten und somit auch die Konditionszahlen geringfügig (vgl. Tabelle 2.1).

3.5.3 Dreifachüberlappung – Rekonstruktion von Linearen

In Abschnitt 3.5.2 wurden für die Dreifachüberlappung Randfenster zur Verarbeitung von Signalen endlicher Länge definiert. Da bezüglich der Rekonstruktionseigenschaften lediglich die Darstellung von konstanten Signalanteilen mit den Basisfunktionen der niedrigsten Frequenz als Bedingung gefordert wurde, genügte es, neue Fensterfunktionen w_{l_0} am linken Signalrand und w_{r_0} am rechten Signalrand einzuführen. Hierbei wurde vorausgesetzt, dass alle Basisfunktionen, deren Träger vollständig im Intervall $[0, 2N]$ liegen, unverändert bleiben. Die inneren dreifach überlappenden Basisfunktionen der niedrigsten Frequenz können jedoch neben Konstanten auch lineare Bestandteile von Signalen perfekt rekonstruieren. Es liegt daher nahe, diese Eigenschaft auch in den Randintervallen zu gewährleisten. Hierzu ist es erforderlich, sich von der oben genannten Voraussetzung zu lösen und auch innere Basisfunktionen durch geeignete Randbasisfunktionen zu ersetzen. Es kann gezeigt werden, dass sich Fensterfunktionen w_{l_0} und w_{l_1} am linken Signalrand sowie Fensterfunktionen w_{r_0} und w_{r_1} am rechten Signalrand zur Konstruktion von trigonometrischen Basen auf dem Intervall bestimmen lassen, die in Analogie zu Gleichung (2.63) die Darstellung von Linearen mit den Basisfunktionen der niedrigsten Frequenz auch in den Randintervallen ermöglichen. Die Herleitung der hierzu benötigten Fensterfunktionen ist Gegenstand dieses Abschnitts.

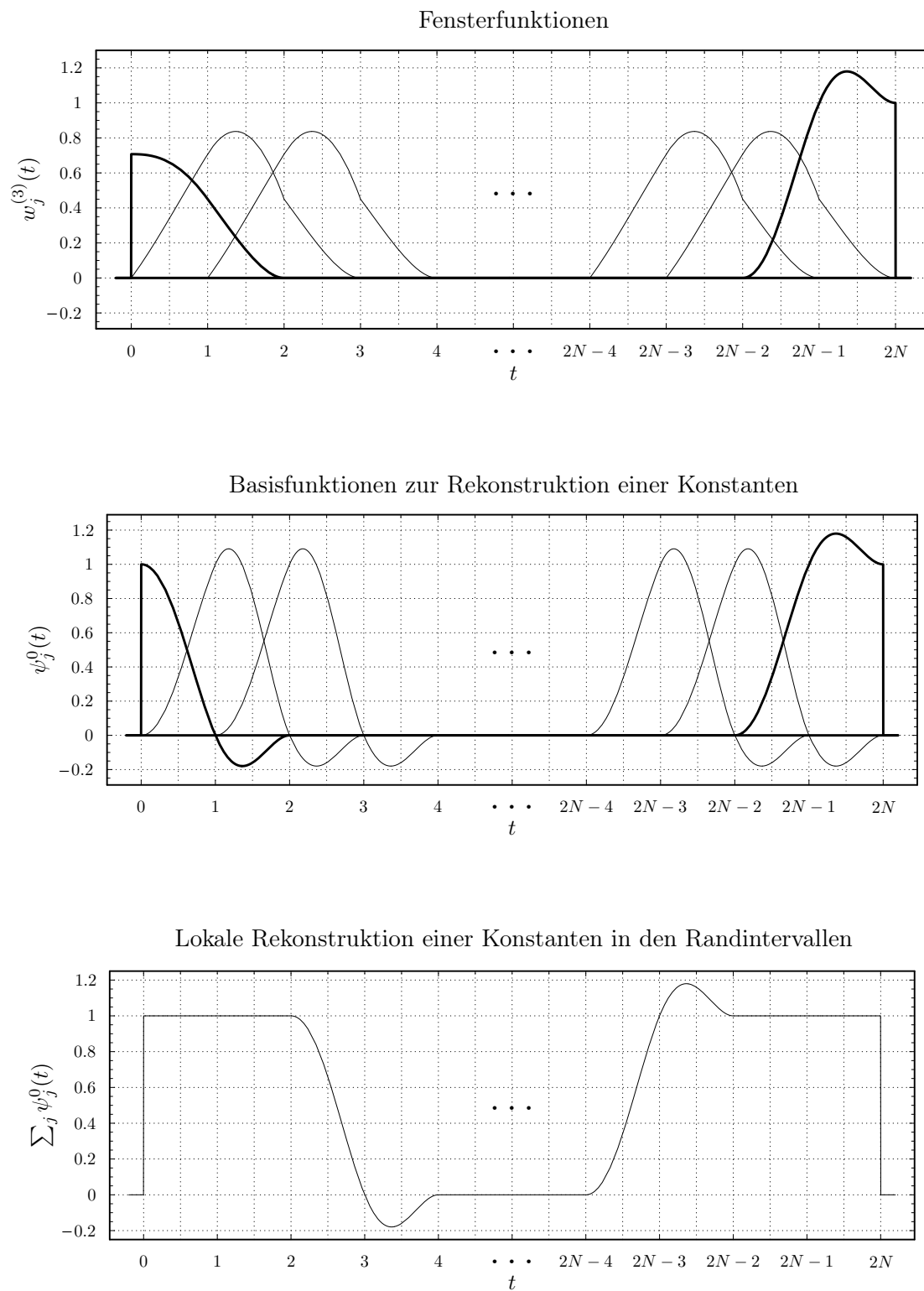


Abbildung 3.8: Dreifachüberlappung: Fenster- und Basisfunktionen zur Verarbeitung von endlich langen Signalen, lokale Rekonstruktion von Konstanten.

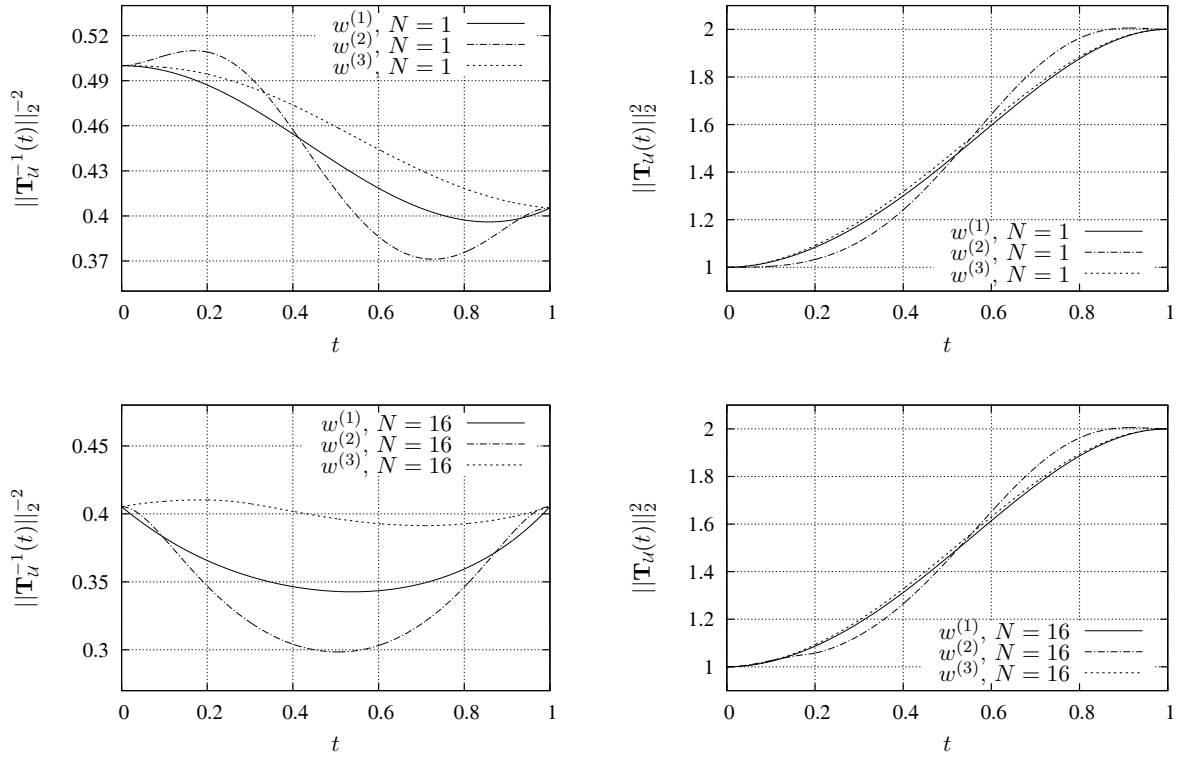


Abbildung 3.9: Dreifachüberlappung: Bestimmung der größten unteren Riesz-Schranke $A_0 = \min \|\mathbf{T}_U^{-1}(t)\|_2^{-2}$ (links) und der kleinsten oberen Riesz-Schranke $B_0 = \max \|\mathbf{T}_U(t)\|_2^2$ (rechts) für $2N$ Teilintervalle, $t \in [0, 1]$ (vgl. Tabelle 3.1).

	A_0	B_0	κ
Fenster $w^{(1)}, N = 1$	0.3960	2	2.2472
Fenster $w^{(2)}, N = 1$	0.3712	2.0056	2.3244
Fenster $w^{(3)}, N = 1$	0.4053	2	2.2214
Fenster $w^{(1)}, N = 16$	0.3427	2	2.4159
Fenster $w^{(2)}, N = 16$	0.2984	2.0053	2.5921
Fenster $w^{(3)}, N = 16$	0.3914	2	2.2606
Fenster $w^{(1)}, N = 32$	0.3425	2	2.4165
Fenster $w^{(2)}, N = 32$	0.2971	2.0053	2.5981
Fenster $w^{(3)}, N = 32$	0.3914	2	2.2606

Tabelle 3.1: Riesz-Konstanten von dreifach überlappenden Kosinus-IV-Kosinus-II-Basen auf dem Intervall, wobei eine Konstante mit den Basisfunktionen ψ_j^0 darstellbar ist. Die mit $w^{(\cdot)}$ bezeichneten Fensterfunktionen beziehen sich auf die Fenster, die in den inneren Intervallen zur Anwendung kommen (vgl. Abschnitt 2.3.2). $2N$ = Anzahl der Teilintervalle, in die das auf einem endlich langen Intervall definierte Signal unterteilt wird.

Fensterfunktionen am linken Signalrand

Im Intervall $t \in [0, 1)$ sind durch die Basisfunktionen der niedrigsten Frequenz folgende Bedingungen zu erfüllen:

$$\psi_0^0(t) + \psi_1^0(t) = w_{l_0}(t)\sqrt{2} \cos\left(\frac{1}{2}\pi t\right) + w_{l_1}(t-1)\sqrt{2} \cos\left(\frac{1}{2}\pi(t-1)\right) = 1, \quad \text{für } t \in [0, 1) \quad (3.60)$$

$$a\psi_0^0(t) + b\psi_1^0(t) = aw_{l_0}(t)\sqrt{2} \cos\left(\frac{1}{2}\pi t\right) + bw_{l_1}(t-1)\sqrt{2} \cos\left(\frac{1}{2}\pi(t-1)\right) = t, \quad \text{für } t \in [0, 1). \quad (3.61)$$

Die Bedingung (3.60) gewährleistet die Rekonstruktion einer Konstanten mit den Basisfunktionen ψ_0^0 und ψ_1^0 . Die Darstellung von linear verlaufenden Signalen bzw. Signalanteilen mit diesen beiden Basisfunktionen wird durch die Bedingung (3.61) sichergestellt. Um die Fensterfunktionen $w_{l_0}(t)$ und $w_{l_1}(t-1)$ aus den Gleichungen (3.60) und (3.61) herzuleiten, sind die Parameter a und b anhand von Randbedingungen zu bestimmen. Aus Gleichung (3.61) erhält man für $t = 0$ den Parameter $a = 0$ und damit für die Fensterfunktion $w_{l_1}(t-1)$ den Ausdruck

$$w_{l_1}(t-1) = \frac{1}{\sqrt{2}} \cdot \frac{t}{b \cos\left(\frac{1}{2}\pi(t-1)\right)}, \quad \text{für } t \in [0, 1). \quad (3.62)$$

Für die Dreifachüberlappung ist die Anpassung der Fenster $w_{l_0}(t)$ und $w_{l_1}(t-1)$ lediglich im Intervall $[0, 1)$ erforderlich. In den Intervallbereichen für $t \geq 1$ können die Fensterfunktionen

$$w_{l_0}(t) = w(t) \quad \text{und} \quad w_{l_1}(t-1) = w(t-1), \quad \text{für } t \geq 1 \quad (3.63)$$

angewendet werden. Um hierbei eine stetige Fensterfunktion $w_{l_1}(t-1)$ zu erhalten, ist für $t = 1$ die Bedingung $w_{l_1}(0) = 1/\sqrt{2}$ zu erfüllen. So ergibt sich aus Gleichung (3.62) der Parameter $b = 1$ und folglich

$$w_{l_1}(t-1) = \frac{1}{\sqrt{2}} \cdot \frac{t}{\cos\left(\frac{1}{2}\pi(t-1)\right)}, \quad \text{für } t \in [0, 1). \quad (3.64)$$

Durch Einsetzen der Fensterfunktion $w_{l_1}(t-1)$ in Gleichung (3.60) und Umformen erhält man

$$w_0(t) = \frac{1}{\sqrt{2}} \cdot \frac{1-t}{\cos\left(\frac{1}{2}\pi t\right)}, \quad \text{für } t \in [0, 1). \quad (3.65)$$

Fasst man die in den Gleichungen (3.63), (3.64) und (3.65) gewonnenen Resultate zusammen, so erhält man für die Konstruktion von Randbasisfunktionen die in Abbildung (3.10) dargestellten Fensterfunktionen

$$w_{l_0}(t) = \begin{cases} \frac{1-t}{\sqrt{2} \cos\left(\frac{1}{2}\pi t\right)}, & \text{für } t \in [0, 1), \\ w(t), & \text{für } t \in [1, 2), \\ 0, & \text{sonst} \end{cases} \quad (3.66)$$

und

$$w_{l_1}(t) = \begin{cases} \frac{t+1}{\sqrt{2} \cos\left(\frac{1}{2}\pi t\right)}, & \text{für } t \in [-1, 0), \\ w(t), & \text{für } t \in [0, 2), \\ 0, & \text{sonst.} \end{cases} \quad (3.67)$$

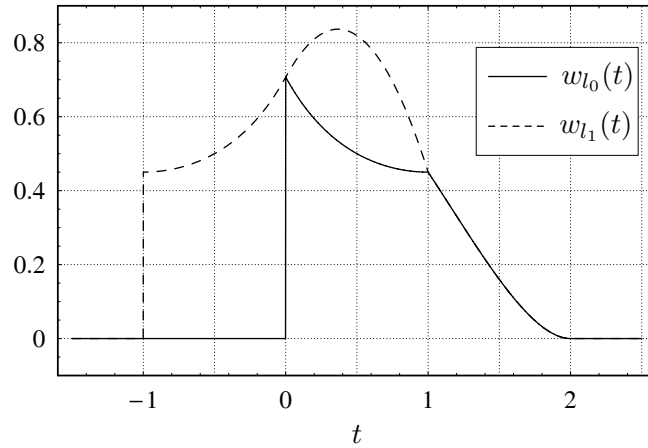


Abbildung 3.10: Dreifachüberlappung: Fensterfunktionen zur Konstruktion von Randbasisfunktionen für die Darstellung von endlich langen Signalen am linken Signalrand, wobei $w = w^{(3)}$.

Fensterfunktionen am rechten Signalrand

Für die Rekonstruktion einer Konstanten mit den Funktionen ψ_j^0 der Kosinus-IV-Kosinus-II-Basen ist im Intervall $[2N - 2, 2N - 1)$ die Bedingung

$$\begin{aligned} \psi_{2N-3}(t) + \psi_{2N-2}(t) + \psi_{2N-1}(t) &= w(t-2N+3)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+3)\right) + w_{r_1}(t-2N+2) \\ &\cdot \sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+2)\right) + w_{r_0}(t-2N+1) = 1, \quad \text{für } t \in [2N-2, 2N-1) \end{aligned} \quad (3.68)$$

zu erfüllen. Da in diesem Intervall $w_{r_1} = w$ vorausgesetzt wird, erhält man nach Substitution von t mit $t + 2N - 1$ und Umformen für w_{r_0} den Ausdruck

$$w_{r_0}(t) = 1 + w(t+1)\sqrt{2}\sin\left(\frac{1}{2}\pi t\right) + w(t+2)\sqrt{2}\cos\left(\frac{1}{2}\pi t\right), \quad \text{für } t \in [-1, 0). \quad (3.69)$$

Dass im Intervall $[2N - 2, 2N - 1)$ auch die Bedingung der Rekonstruktion einer Linearen mit den Basisfunktionen ψ_j^0 erfüllt ist, lässt sich durch nachfolgende Umformungen zeigen. Für $t \in [2N - 2, 2N - 1)$ folgt aus Gleichung (2.63)

$$\begin{aligned} f(t) &= (a(2N-3) + b)\psi_{2N-3}^0(t) + (a(2N-2) + b)\psi_{2N-2}^0(t) + (a(2N-1) + b)\psi_{2N-1}^0(t) \\ &= (a(2N-3) + b)w(t-2N+3)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+3)\right) + (a(2N-2) + b) \\ &\cdot w_{r_1}(t-2N+2)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+2)\right) + (a(2N-1) + b)w_{r_0}(t-2N+1). \end{aligned} \quad (3.70)$$

Entsprechend der oben genannten Voraussetzung wird w_{r_1} mit w substituiert und für w_{r_0} Gleichung (3.69) eingesetzt. Durch weiteres Umformen erhält man

$$\begin{aligned} f(t) &= a\left(-2w(t-2N+3)\sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+3)\right) - w(t-2N+2)\right. \\ &\quad \cdot \sqrt{2}\cos\left(\frac{1}{2}\pi(t-2N+2)\right) + 2N-1) + b, \quad \text{für } t \in [2N-2, 2N-1). \end{aligned} \quad (3.71)$$

Die Fensterfunktionen $w(t - 2N + 3)$ und $w(t - 2N + 2)$ ergeben sich aus der Definition des dreifach überlappenden Fensters in Abschnitt 2.3.2. Gemäß Gleichung (2.64) gelten im Intervall $[2N - 2, 2N - 1]$ für $t \in [2N - 2, 2N - 1]$ die Beziehungen

$$w(t - 2N + 3) = \frac{-2 - t + 2N}{\sqrt{2} \cos(\frac{1}{2}\pi(t - 2N + 3))} - g(t - 2N + 2) \quad (3.72)$$

und

$$w(t - 2N + 2) = \frac{3 + t - 2N - 2g(t - 2N + 2)\sqrt{2} \sin(\frac{1}{2}\pi(t - 2N + 2))}{\sqrt{2} \cos(\frac{1}{2}\pi(t - 2N + 2))}. \quad (3.73)$$

Durch Einsetzen in Gleichung (3.71) folgt sofort $f(t) = at + b$, womit gezeigt ist, dass die konstruierten Basen im Intervall $[2N - 2, 2N - 1]$ eine Lineare mit den Basisfunktionen ψ_j^0 reproduzieren.

Im Intervall $[2N - 1, 2N)$ ist für die Rekonstruktion einer Konstanten die Bedingung

$$\begin{aligned} \psi_{2N-2}(t) + \psi_{2N-1}(t) &= w_{r_1}(t - 2N + 2)\sqrt{2} \cos\left(\frac{1}{2}\pi(t - 2N + 2)\right) \\ &\quad + w_{r_0}(t - 2N + 1) = 1, \quad \text{für } t \in [2N - 1, 2N) \end{aligned} \quad (3.74)$$

zu erfüllen. Die zusätzliche Bedingung

$$\begin{aligned} a\psi_{2N-2}(t) + b\psi_{2N-1}(t) &= aw_{r_1}(t - 2N + 2)\sqrt{2} \cos\left(\frac{1}{2}\pi(t - 2N + 2)\right) \\ &\quad + bw_{r_0}(t - 2N + 1) = t - 2N, \quad \text{für } t \in [2N - 1, 2N) \end{aligned} \quad (3.75)$$

soll die Darstellung einer Linearen mit den Basisfunktionen der niedrigsten Frequenz sicherstellen. Hieraus ergibt sich nach Substitution mit $t = t + 2N - 1$ das Gleichungssystem

$$w_{r_1}(t + 1)\sqrt{2} \cos\left(\frac{1}{2}\pi(t + 1)\right) + w_{r_0}(t) = 1, \quad \text{für } t \in [0, 1), \quad (3.76)$$

$$aw_{r_1}(t + 1)\sqrt{2} \cos\left(\frac{1}{2}\pi(t + 1)\right) + bw_{r_0}(t) = t - 1, \quad \text{für } t \in [0, 1). \quad (3.77)$$

Durch Lösen des Gleichungssystems erhält man die Fensterfunktionen

$$w_{r_0}(t) = 1 - \frac{t - 1 - b}{a - b}, \quad \text{für } t \in [0, 1) \quad (3.78)$$

und

$$w_{r_1}(t + 1) = \frac{1}{\sqrt{2}} \cdot \frac{t - 1 - b}{\cos(\frac{1}{2}\pi(t + 1))(a - b)}, \quad \text{für } t \in [0, 1). \quad (3.79)$$

Die Parameter a und b werden aus Randbedingungen ermittelt. Hierzu sind die Fensterfunktionen $w_{r_0}(t)$ und $w_{r_1}(t + 1)$ an der Stelle $t = 0$ zu betrachten. Es wird Stetigkeit der Fensterfunktionen gefordert, d.h. $w_{r_0}(0) = 1$ und $w_{r_1}(1) = \frac{\sqrt{2}}{\pi}$. Aus Gleichung (3.77) lässt sich dann der Parameter $b = -1$ bestimmen. Aus Gleichung (3.79) erhält man durch Grenzwertberechnung mittels Bernoulli-L'Hospitalscher Regel [BSMM08] den Parameter $a = -2$, so dass

$$w_{r_0}(t) = t + 1, \quad \text{für } t \in [0, 1) \quad (3.80)$$

und

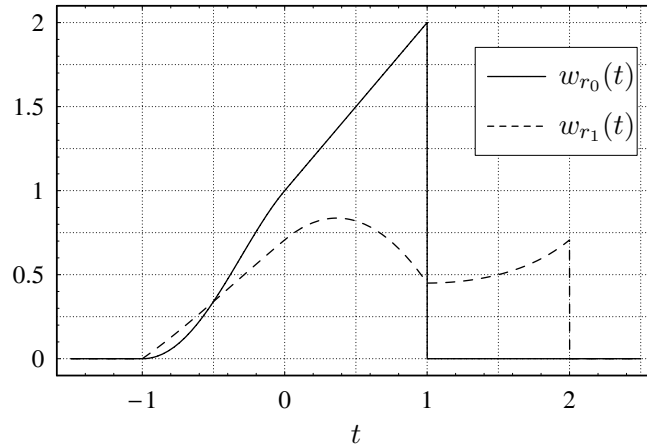


Abbildung 3.11: Dreifachüberlappung: Fensterfunktionen zur Konstruktion von Randbasisfunktionen für die Darstellung von endlich langen Signalen am rechten Signalrand, wobei $w = w^{(3)}$.

$$w_{r1}(t+1) = \frac{1}{\sqrt{2}} \cdot \frac{-t}{\cos\left(\frac{1}{2}\pi(t+1)\right)}, \quad \text{für } t \in [0, 1). \quad (3.81)$$

Durch Zusammenfassen der in den Gleichungen (3.69), (3.80) und (3.81) gewonnenen Resultate erhält man für die Konstruktion von Randbasisfunktionen am rechten Signalrand die in Abbildung 3.11 dargestellten Fensterfunktionen

$$w_{r0}(t) = \begin{cases} 1 + w(t+1)\sqrt{2}\sin\left(\frac{1}{2}\pi t\right) + w(t+2)\sqrt{2}\cos\left(\frac{1}{2}\pi t\right), & \text{für } t \in [-1, 0), \\ t+1, & \text{für } t \in [0, 1), \\ 0, & \text{sonst} \end{cases} \quad (3.82)$$

und

$$w_{r1}(t) = \begin{cases} w(t), & \text{für } t \in [-1, 1), \\ \frac{1-t}{\sqrt{2}\cos\left(\frac{1}{2}\pi t\right)}, & \text{für } t \in [1, 2), \\ 0, & \text{sonst.} \end{cases} \quad (3.83)$$

Abbildung 3.12 zeigt die lokale Repräsentation von Linearen unter Verwendung der mit diesen Fenstern konstruierten Randbasisfunktionen. Es wird sichtbar, dass sich auch in den Randintervallen linear verlaufende Signalbestandteile bereits mit den Basisfunktionen ψ_j^0 lokal reproduzieren lassen. In Abbildung D.2 (Anhang D) sind für ausgewählte Intervalle an den Signalrändern die dualen Fensterfunktionen zur Konstruktion von Analysebasisfunktionen dargestellt. Die Riesz-Schranken verschiedener dreifach überlappender Kosinus-IV-Kosinus-II-Basen sind in Tabelle 3.2 angegeben. Im Vergleich zu den Riesz-Schranken der Basen auf dem Intervall aus Abschnitt 3.5.2 (siehe Tabelle 3.1), die in den Randintervallen lediglich eine Konstante mit den Basisfunktionen ψ_j^0 rekonstruieren, ist zu erkennen, dass die Verbesserung der polynomialen Reproduktionseigenschaft mit einer Verschlechterung der Riesz-Konstanten einhergeht. Es sei vorweggenommen, dass die Verwendung dieser Randfenster im Vergleich zu den Randfenstern aus Abschnitt 3.5.2 zu geringfügig schlechteren Kompressionsergebnissen bei den im Rahmen dieser Arbeit untersuchten Kompressionsstrategien und Testbildern führten.

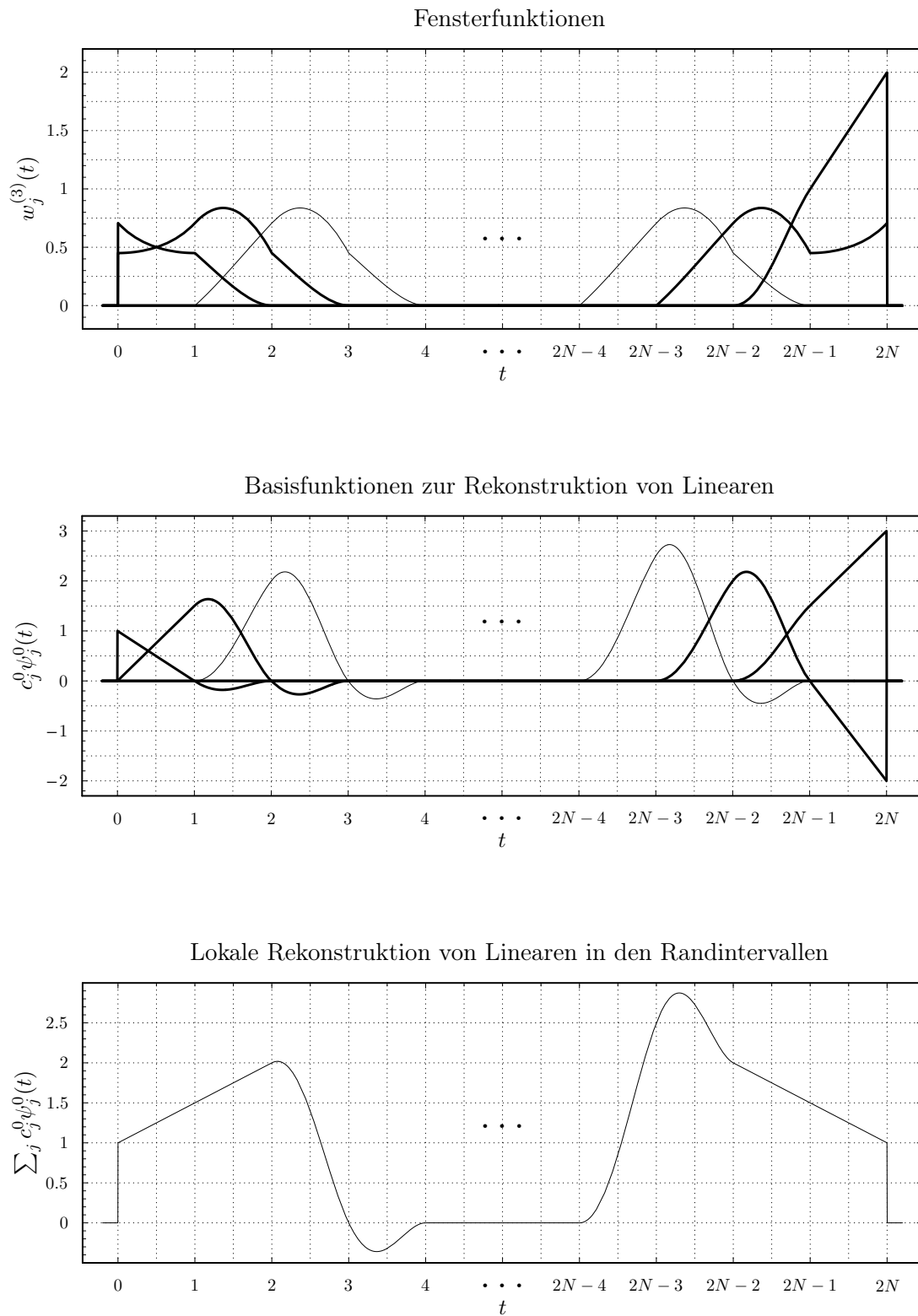


Abbildung 3.12: Dreifachüberlappung: Fenster- und Basisfunktionen zur Verarbeitung von endlich langen Signalen, lokale Rekonstruktion von Linearen.

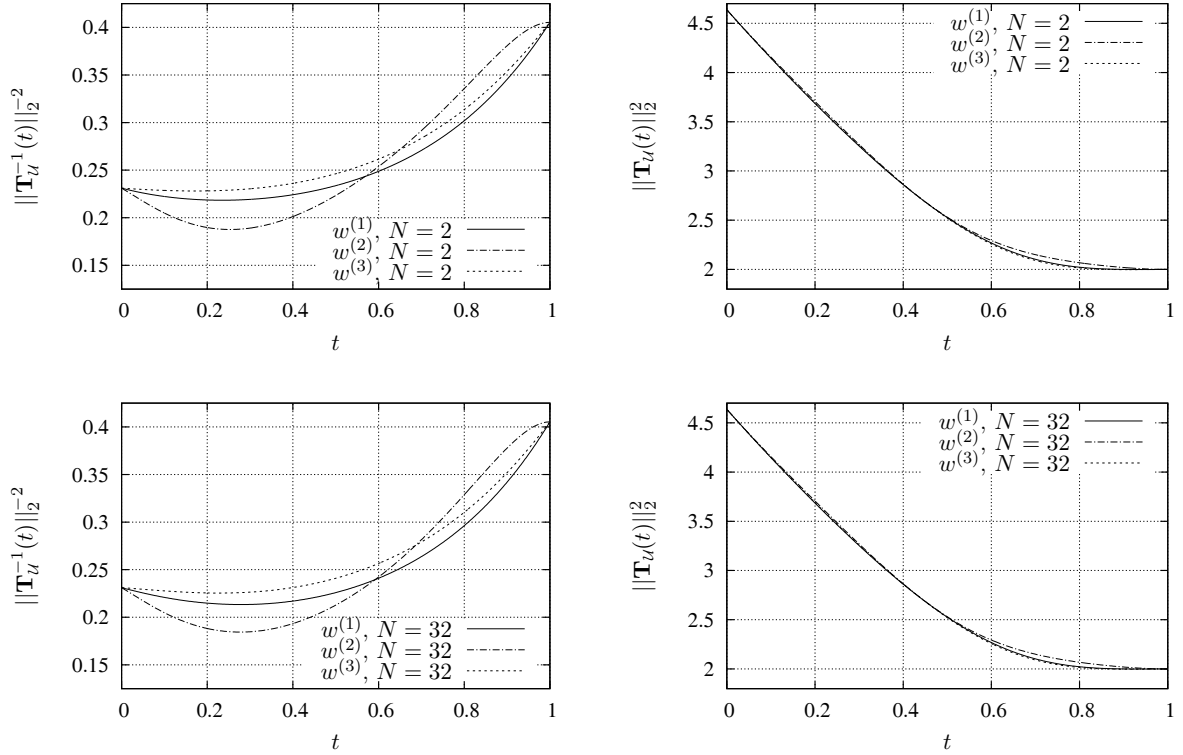


Abbildung 3.13: Dreifachüberlappung: Bestimmung der größten unteren Riesz-Schranke $A_0 = \min \|\mathbf{T}_U^{-1}(t)\|_2^{-2}$ (links) und der kleinsten oberen Riesz-Schranke $B_0 = \max \|\mathbf{T}_U(t)\|_2^2$ (rechts) für $2N$ Teilintervalle, $t \in [0, 1]$ (vgl. Tabelle 3.2).

	A_0	B_0	κ
Fenster $w^{(1)}$, $N = 2$	0.2184	4.6375	4.6080
Fenster $w^{(2)}$, $N = 2$	0.1877	4.6375	4.9711
Fenster $w^{(3)}$, $N = 2$	0.2281	4.6375	4.5088
Fenster $w^{(1)}$, $N = 16, 32$	0.2134	4.6375	4.6620
Fenster $w^{(2)}$, $N = 16, 32$	0.1844	4.6375	5.0143
Fenster $w^{(3)}$, $N = 16, 32$	0.2254	4.6375	4.5355

Tabelle 3.2: Riesz-Konstanten von dreifach überlappenden Kosinus-IV-Kosinus-II-Basen auf dem Intervall, wobei eine Lineare mit den Basisfunktionen ψ_j^0 darstellbar ist. Die Fensterfunktionen $w^{(\cdot)}$ beziehen sich auf die Fenster, die in den inneren Intervallen zur Anwendung kommen (vgl. Abschnitt 2.3.2). $2N =$ Anzahl der Teilintervalle, in die das auf einem endlich langen Intervall definierte Signal unterteilt wird.

3.5.4 Vierfachüberlappung – Ansatz 1

Betrachtet man alle Wichtungsfenster w_j als Translate einer Fensterfunktion w , so überschneiden ab einem Überlappungsfaktor von Vier neben den Basisfunktionen ψ_0^k und ψ_{2N-1}^k nun auch die Basisfunktionen ψ_1^k und ψ_{2N-2}^k die Signalgrenzen bei $t = 0$ bzw. $t = 2N$. Ausgehend von der Klasse $2m$ -fach überlappende Fensterfunktionen w aus Abschnitt 2.3.3 werden daher im Folgenden für den Fall der Vierfachüberlappung Fensterfunktionen $w_{l_0}, w_{l_1}, w_{r_0}$ und w_{r_1} zur Konstruktion von überlappenden trigonometrischen Basen auf dem Intervall eingeführt, die der Forderung der Reproduktion einer Konstanten mit den Basisfunktionen ψ_j^0 genügen. Analog zur Dreifachüberlappung wurden für die Vierfachüberlappung ebenfalls Randfenster zur Konstruktion von Basisfunktionen entwickelt und untersucht, die in den Randintervallen neben Konstanten auch linear verlaufende Signalanteile mit den Basisfunktionen ψ_j^0 repräsentieren können. Die so entwickelten Basen führten jedoch wie bei der Dreifachüberlappung im Vergleich zu den Basen, die in den Randintervallen lediglich eine Konstante mit den Basisfunktionen der niedrigsten Frequenz reproduzieren können, zu einer Verschlechterung der Riesz-Stabilität und brachten keine Vorteile in der Bilddatenkompression. Daher wird sich nachfolgend auf Fensterfunktionen zur Konstruktion von Basen auf dem Intervall konzentriert, die in den Randintervallen ausschließlich die Bedingung der Reproduktion einer Konstanten mit den Basisfunktionen ψ_j^0 erfüllen.

Fensterfunktionen am linken Signalrand

Zur Konstruktion von Randbasisfunktionen für die Darstellung von Signalen, die auf einem endlich langen Intervall gegeben sind, werden am linken Signalrand die beiden Fensterfunktionen

$$w_{l_0}(t) := \begin{cases} \frac{1 - g_{l_1}(t-1)\sqrt{2}\sin(\frac{1}{2}\pi t)}{\sqrt{2}\cos(\frac{1}{2}\pi t)}, & \text{für } t \in [0, \frac{1}{2}), \\ \frac{1 - g_{l_1}(t-1)\sqrt{2}\sin(\frac{1}{2}\pi t) + w(t-2)\sqrt{2}\cos(\frac{1}{2}\pi t)}{\sqrt{2}\cos(\frac{1}{2}\pi t)}, & \text{für } t \in [\frac{1}{2}, 1), \\ w(t), & \text{für } t \in [1, \frac{5}{2}), \\ 0, & \text{sonst} \end{cases} \quad (3.84)$$

und

$$w_{l_1}(t) := \begin{cases} g_{l_1}(t), & \text{für } t \in [-1, 0), \\ \frac{1 + (w(t+1) - w(t-1))\sqrt{2}\sin(\frac{1}{2}\pi t)}{\sqrt{2}\cos(\frac{1}{2}\pi t)}, & \text{für } t \in [0, \frac{1}{2}), \\ w(t), & \text{für } t \in [\frac{1}{2}, \frac{5}{2}), \\ 0, & \text{sonst} \end{cases} \quad (3.85)$$

definiert. Zur Gewährleistung von Stetigkeit und stetiger Differenzierbarkeit der Fensterfunktionen w_{l_1} sind von der Funktion g_{l_1} die Bedingungen

$$\begin{aligned} g_{l_1}(-1) &= 0, & g'_{l_1}(-1) &= 0, \\ g_{l_1}(0) &= w_{l_1}(0), & g'_{l_1}(0) &= w'_{l_1}(0), & g''_{l_1}(0) &= w''_{l_1}(0), & \text{wobei} \\ w_{l_1}(0) &= \frac{1}{\sqrt{2}}, & w'_{l_1}(0) &= \frac{\pi}{2\sqrt{2}}, & w''_{l_1}(0) &= -\frac{\pi^2}{4\sqrt{2}} \end{aligned} \quad (3.86)$$

zu erfüllen. Die einfache stetige Differenzierbarkeit an der Stelle $t = 0$ genügt zwar für die Realisierung einer stetig differenzierbaren Fensterfunktion $w_{l_1}(t)$. Dies führt dann aber zu

dem unerwünschten Resultat, dass die Fensterfunktion $w_{l_0}(t)$ an der Stelle $t = 1$ nicht stetig differenzierbar ist. Bei dem Polynom

$$g_{l_1}(t) = -\frac{(1+t)^2(-8+t(16-24t+\pi(-4+(8+\pi)t)))}{8\sqrt{2}} \quad (3.87)$$

handelt es sich um eine Funktion, die den Forderungen in Gleichung (3.86) genügt. Die hieraus resultierenden Fensterfunktionen w_{l_0} und w_{l_1} zur Konstruktion von Randbasisfunktionen für den linken Signalrand sind in Abbildung 3.14 dargestellt.

Fensterfunktionen am rechten Signalrand

Am rechten Signalrand werden zur Konstruktion der Randbasisfunktionen für vierfach überlappende Kosinus-IV-Kosinus-II-Basen auf dem Intervall die beiden Fensterfunktionen

$$w_{r_0}(t) = \begin{cases} 1 + w(t+2)\sqrt{2}\cos(\frac{1}{2}\pi t) + (w(t+1) - w(t+3))\sqrt{2}\sin(\frac{1}{2}\pi t), & \text{für } t \in [-\frac{3}{2}, 0), \\ 1 + w(t+2)\sqrt{2}\cos(\frac{1}{2}\pi t) + g_{r_1}(t+1)\sqrt{2}\sin(\frac{1}{2}\pi t), & \text{für } t \in [0, \frac{1}{2}), \\ 1 + g_{r_1}(t+1)\sqrt{2}\sin(\frac{1}{2}\pi t), & \text{für } t \in [\frac{1}{2}, 1), \\ 0, & \text{sonst} \end{cases} \quad (3.88)$$

und

$$w_{r_1}(t) = \begin{cases} w(t), & \text{für } t \in [-\frac{3}{2}, 1), \\ g_{r_1}(t), & \text{für } t \in [1, 2), \\ 0, & \text{sonst} \end{cases} \quad (3.89)$$

definiert. Die Funktion g_{r_1} hat zur Gewährleistung von Stetigkeit und stetiger Differenzierbarkeit der Fensterfunktion w_{r_1} die Bedingungen

$$\begin{aligned} g_{r_1}(2) &= 0, & g'_{r_1}(2) &= 0, \\ g_{r_1}(1) &= w(1), & g'_{r_1}(1) &= w'(1), \quad \text{wobei} \\ w(1) &= \frac{4+3\pi}{4\sqrt{2\pi}}, & w'(1) &= -\frac{3\pi}{8\sqrt{2}} \end{aligned} \quad (3.90)$$

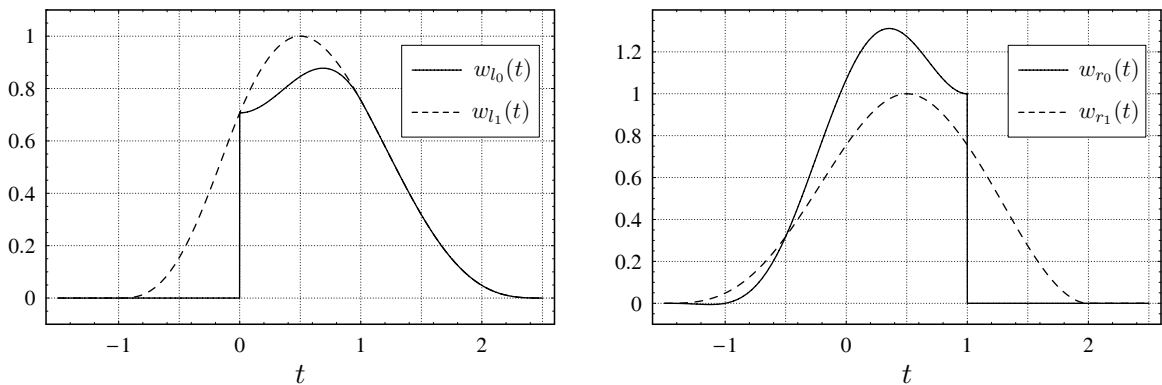


Abbildung 3.14: Vierfachüberlappung: Fensterfunktionen zur Konstruktion von Randbasisfunktionen für die Darstellung von endlich langen Signalen am linken und am rechten Signalrand.

zu erfüllen. Damit ist gleichzeitig auch die stetige Differenzierbarkeit der Fensterfunktion w_{r_0} gewährleistet. Die Funktion

$$g_{r_1}(t) = -\frac{(8 + 3\pi(2 + \pi(-1 + t) - 4t) - 16t)(-2 + t)^2}{8\sqrt{2}\pi} \quad (3.91)$$

erfüllt die genannten Forderungen. In Abbildung 3.14 sind die hieraus resultierenden Fensterfunktionen w_{r_0} und w_{r_1} dargestellt. Analog zu den Betrachtungen für den Fall der Dreifachüberlappung in Abschnitt 3.5.2 lässt sich auch für die Vierfachüberlappung durch einfache Umformungen nachweisen, dass die trigonometrischen Basen auf dem Intervall, die mit den in diesem Abschnitt neu eingeführten Randfenstern konstruiert werden, eine Konstante mit den Basisfunktionen ψ_j^0 reproduzieren. Abbildung 3.15 zeigt schließlich für die Randintervalle die Konstruktion der Basisfunktionen ψ_j^0 und die lokale Darstellung einer Konstanten mit diesen Basisfunktionen.

Die Bestimmung der Riesz-Schranken zum Nachweis der Riesz-Stabilität für ausgewählte (verschiedene Parameter N) trigonometrische Basen auf dem Intervall ist zusammen mit der Zweifach- und der Sechsfachüberlappung in Abbildung 3.21 dargestellt. Tabelle 3.3 fasst die ermittelten Riesz-Schranken und Konditionszahlen zusammen. Des Weiteren sind in Abbildung D.3 (Anhang D) für eine ausgewählte Anzahl von Intervallen in der Umgebung der Signlränder die dualen Fensterfunktionen \tilde{w}_j zur Konstruktion von Basisfunktionen ψ_j für die Signalanalyse veranschaulicht.

3.5.5 Vierfachüberlappung – Ansatz 2

In diesem Abschnitt wird der Frage nachgegangen, ob sich geeignete Randbasisfunktionen auch durch ein einfaches Abschneiden der Basisfunktionen ψ_1^k und ψ_{2N-2}^k bzw. der Fensterfunktionen w_1 und w_{2N-2} an den Signlrändern konstruieren lassen, dh.

$$w_{l_1}(t) := \begin{cases} w(t), & \text{für } t \in [-1, \frac{5}{2}) \\ 0, & \text{sonst} \end{cases} \quad (3.92)$$

und

$$w_{r_1}(t) := \begin{cases} w(t), & \text{für } t \in [-\frac{3}{2}, 2) \\ 0, & \text{sonst.} \end{cases} \quad (3.93)$$

Die Fensterfunktionen w_{l_0} und w_{r_0} können dann direkt aus Gleichung (3.40) durch Umformen abgeleitet werden. Man erhält

$$w_{l_0}(t) = \begin{cases} \frac{1 - w_{r_0}(t-1)\sqrt{2}\cos(\frac{1}{2}\pi(t-1)) - w(t-2)\sqrt{2}\cos(\frac{1}{2}\pi(t-2))}{\sqrt{2}\cos(\frac{1}{2}\pi t)}, & \text{für } t \in [0, \frac{3}{2}), \\ w(t), & \text{für } t \in [\frac{3}{2}, \frac{5}{2}), \\ 0, & \text{sonst} \end{cases} \quad (3.94)$$

und

$$w_{r_0}(t) = \begin{cases} 1 - w_{r_1}(t+1)\sqrt{2}\cos\left(\frac{1}{2}\pi(t+1)\right) - w(t+2)\sqrt{2} \\ \quad \cdot \cos\left(\frac{1}{2}\pi(t+2)\right) - w(t+3)\sqrt{2}\cos\left(\frac{1}{2}\pi(t+3)\right), & \text{für } t \in [-\frac{3}{2}, 1), \\ 0, & \text{sonst.} \end{cases} \quad (3.95)$$

Während sich mit den so bestimmten Fensterfunktionen am rechten Signalrand geeignete

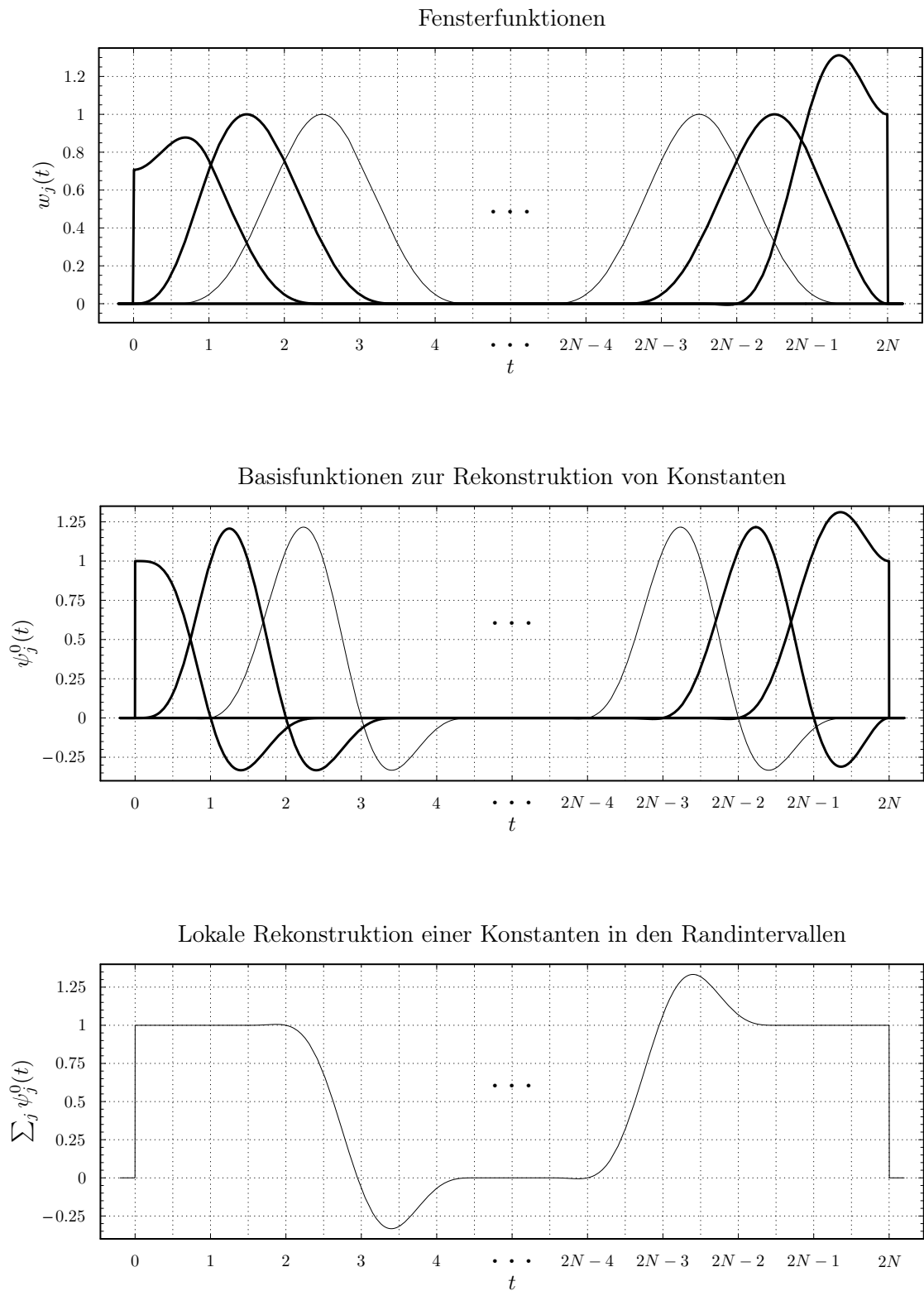


Abbildung 3.15: Vierfachüberlappung, Ansatz 1: Fenster- und Basisfunktionen zur Verarbeitung von endlich langen Signalen, lokale Rekonstruktion von Konstanten.

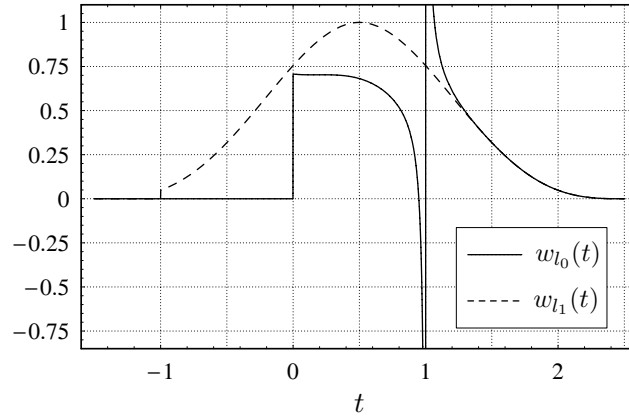


Abbildung 3.16: Vierfachüberlappung: Randfenster am linken Signalrand mit Polstelle.

Randbasisfunktionen konstruieren lassen, ist hingegen am linken Signalrand die Bedingung der Riesz-Stabilität nicht erfüllt. Die Fensterfunktion w_{l_0} besitzt, wie in Abbildung 3.16 dargestellt, bei $t = 1$ eine Polstelle. Das aufgezeigte Problem kann durch einen sehr einfachen Konstruktionsansatz einer Lösung zugeführt werden. Betrachtet man Basisfunktionen auf der reellen Achse, so sind bei der Vierfachüberlappung zur Reproduktion einer Konstanten $f(t) = 1$ im Intervall $t \in [0, \frac{3}{2})$ die Basisfunktionen

$$\underbrace{\psi_{-2}^0 + \psi_{-1}^0}_{\text{n.d.}} + \psi_0^0 + \psi_1^0 + \psi_2^0 = 1 \quad (3.96)$$

erforderlich. Da bei der Beschreibung von Signalen auf dem Intervall $[0, 2N]$ die Basisfunktionen ψ_{-2}^0 und ψ_{-1}^0 bzw. die Wichtungskoeffizienten dieser Basisfunktionen jedoch nicht definiert sind, werden zur Konstruktion einer Konstanten die in das Intervall $[0, \frac{3}{2})$ überlappenden Anteile dieser nur auf der reellen Achse definierten Basisfunktionen mit den neu einzuführenden Randbasisfunktionen

$$\begin{aligned} \psi_{l_1}^0 &:= \psi_1^0 + \psi_{-1}^0 = w_{l_1}(t-1)\sqrt{2} \cos\left(\frac{1}{2}\pi(t-1)\right), \\ &= w(t-1)\sqrt{2} \cos\left(\frac{1}{2}\pi(t-1)\right) + w(t+1)\sqrt{2} \cos\left(\frac{1}{2}\pi(t+1)\right), \end{aligned} \quad (3.97)$$

$$\begin{aligned} \psi_{l_0}^0 &:= \psi_0^0 + \psi_{-2}^0 = w_{l_0}(t)\sqrt{2} \cos\left(\frac{1}{2}\pi t\right), \\ &= w(t)\sqrt{2} \cos\left(\frac{1}{2}\pi t\right) + w(t+2)\sqrt{2} \cos\left(\frac{1}{2}\pi(t+2)\right) \end{aligned} \quad (3.98)$$

repräsentiert. Durch Umformen erhält man schließlich zur Konstruktion von stabilen Basen auf dem Intervall die beiden Randfenster

$$w_{l_0}(t) = \begin{cases} w(t) - w(t+2), & \text{für } t \in [0, \frac{5}{2}), \\ 0, & \text{sonst,} \end{cases} \quad (3.99)$$

$$w_{l_1}(t) = \begin{cases} w(t) - w(t+2), & \text{für } t \in [-1, \frac{5}{2}), \\ 0, & \text{sonst.} \end{cases} \quad (3.100)$$

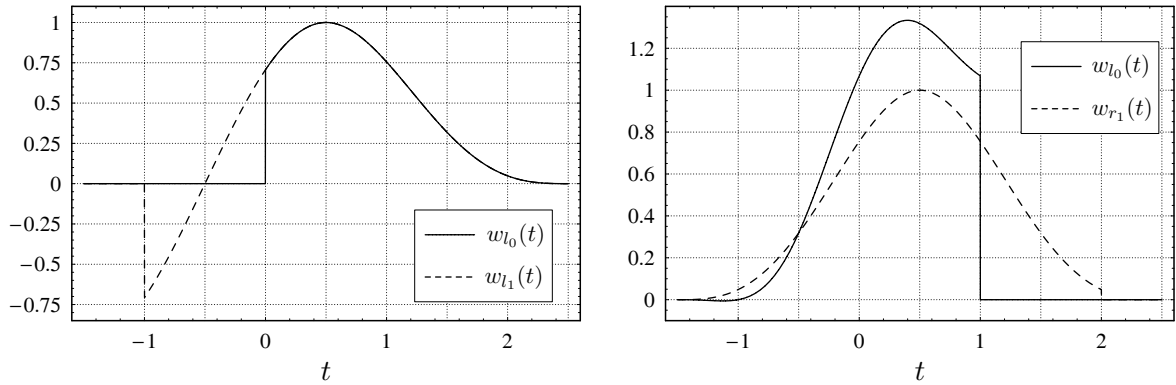


Abbildung 3.17: Vierfachüberlappung: Fensterfunktionen zur Konstruktion von Randbasisfunktionen für die Darstellung von endlich langen Signalen am linken und rechten Signalrand (Hinweis: $w_{l_1}(t) = w_{l_0}(t)$ für $t \in [0, \frac{5}{2})$).

Sie sind in Abbildung 3.17 dargestellt, ebenso wie die Fensterfunktionen nach den Gleichungen (3.93) und (3.95) zur Konstruktion von Randbasisfunktionen am rechten Signalrand. Abbildung 3.18 zeigt für die Randintervalle die Konstruktion der Basisfunktionen ψ_j^0 sowie die lokale Reproduktion von Konstanten mit diesen Basisfunktionen. Im Vergleich zu den trigonometrischen Basen aus Abschnitt 3.5.4 verschlechtert sich die Riesz-Stabilität geringfügig (vgl. Abbildung 3.21 und Tabelle 3.3) und die Fensterfunktionen zur Signalanalyse weisen Sprungstellen auf (vgl. Abbildung D.4). Dennoch erhält man mit diesem Lösungsansatz Randbasisfunktionen, die sich für die Anwendung in der Bilddatenkompression eignen.

Das in diesem Abschnitt vorgestellte Konstruktionsprinzip lässt sich grundsätzlich auch auf höhere Überlappungsfaktoren übertragen. Die Bestimmung der Randbasisfunktionen am rechten Signalrand erfolgt analog zu den Gleichungen (3.93) und (3.95). Am linken Signalrand werden in Abhängigkeit vom Überlappungsfaktor in der Gleichung (3.97) weitere Basisfunktionen $\psi_{-3}^0, \psi_{-5}^0, \dots$ und in der Gleichung (3.98) weitere Basisfunktionen $\psi_{-4}^0, \psi_{-6}^0, \dots$ aufsummiert. Die Basisfunktionen $\psi_{j>1}$, die den Signalrand bei $t = 0$ überlappen, sind an dieser Stelle abzuschneiden.

3.5.6 Sechsfachüberlappung

Fensterfunktionen am linken Signalrand

Zur Konstruktion von Randbasisfunktionen werden bei der Sechsfachüberlappung ausgehend von der in Abschnitt 2.3.3 für den Parameter $m = 3$ beschriebenen Fensterfunktion w die drei Randfenster

$$w_{l_0}(t) := \begin{cases} \frac{1}{\sqrt{2} \cos(\frac{1}{2}\pi t)} - g_{l_1}(t-1) \tan(\frac{1}{2}\pi t) + g_{l_2}(t-2), & \text{für } t \in [0, \frac{1}{2}), \\ \frac{1}{\sqrt{2} \cos(\frac{1}{2}\pi t)} + (w(t-3) - g_{l_1}(t-1)) \tan(\frac{1}{2}\pi t) + g_{l_2}(t-2), & \text{für } t \in [\frac{1}{2}, 1), \\ w(t), & \text{für } t \in [1, \frac{7}{2}), \\ 0, & \text{sonst,} \end{cases} \quad (3.101)$$

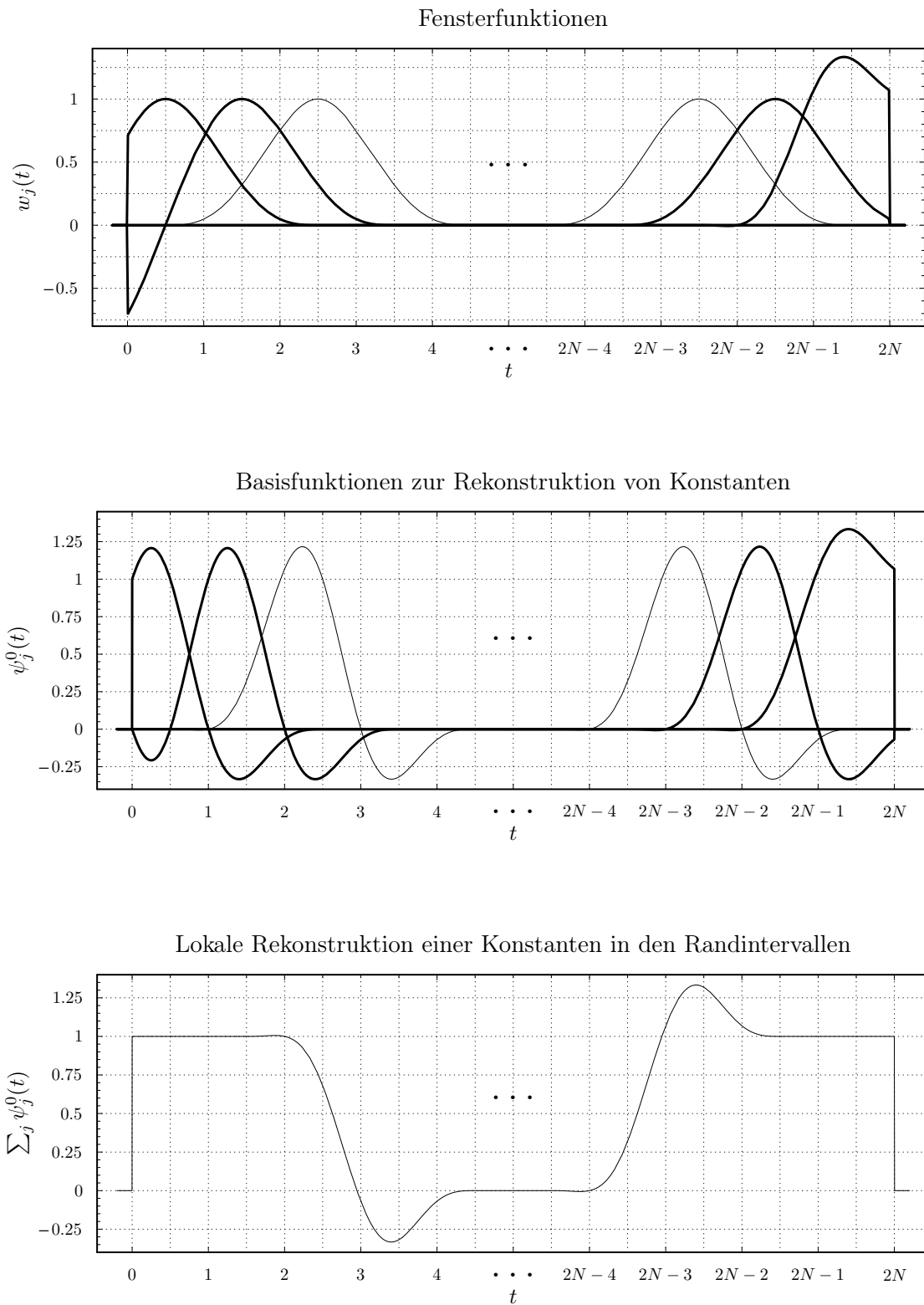


Abbildung 3.18: Vierfachüberlappung, Ansatz 2: Fenster- und Basisfunktionen zur Verarbeitung von endlich langen Signalen, lokale Rekonstruktion von Konstanten.

$$w_{l_1}(t) := \begin{cases} g_{l_1}(t), & \text{für } t \in [-1, 0), \\ \frac{1}{\sqrt{2} \cos(\frac{1}{2}\pi t)} + (w(t+1) - w(t-1)) \tan(\frac{1}{2}\pi t) + w(t-2), & \text{für } t \in [0, \frac{1}{2}), \\ \frac{1}{\sqrt{2} \cos(\frac{1}{2}\pi t)} + (w(t+1) - w(t-1) + w(t-3)) \tan(\frac{1}{2}\pi t) \\ \quad + w(t-2), & \text{für } t \in [\frac{1}{2}, \frac{3}{2}), \\ w(t), & \text{für } t \in [\frac{3}{2}, \frac{7}{2}), \\ 0, & \text{sonst} \end{cases} \quad (3.102)$$

und

$$w_{l_2}(t) := \begin{cases} g_{l_2}(t), & \text{für } t \in [-2, -1), \\ w(t), & \text{für } t \in [-1, \frac{7}{2}), \\ 0, & \text{sonst} \end{cases} \quad (3.103)$$

definiert. Zur Gewährleistung von Stetigkeit und stetiger Differenzierbarkeit hat die Funktion g_{l_1} die Bedingungen

$$\begin{aligned} g_{l_1}(-1) = 0, \quad g_{l_1}(0) &= \frac{48 + \pi(-12 + 31\pi)}{32\sqrt{2}\pi^2}, \\ g'_{l_1}(-1) = 0, \quad g'_{l_1}(0) &= \frac{24 + \pi(-4 + 15\pi)}{32\sqrt{2}\pi}, \quad g''_{l_1}(0) = \frac{16 - \pi(12 + 29\pi)}{128\sqrt{2}} \end{aligned} \quad (3.104)$$

zu erfüllen. Die einfache stetige Differenzierbarkeit an der Stelle $t = 0$ genügt zwar für die Realisierung einer stetig differenzierbaren Fensterfunktion $w_{l_1}(t)$. Dies führt aber in Analogie zur Vierfachüberlappung ebenfalls zu dem unerwünschten Resultat, dass die Fensterfunktion $w_{l_0}(t)$ bei $t = 1$ nicht stetig differenzierbar ist. Das Polynom

$$\begin{aligned} g_{l_1} = -\frac{1}{256\sqrt{2}\pi^2} &((1+t)^2(-8(48 + \pi(-12 + 31\pi)) - 24(-4 + \pi)(8 + \pi(-2 + 5\pi))t \\ &+ (-1152 + \pi(672 + \pi(-824 + \pi(252 + 29\pi))))t^2)) \end{aligned} \quad (3.105)$$

genügt den in Gleichung (3.104) angegebenen Forderungen. Die Funktion g_{l_2} hat die Bedingungen

$$\begin{aligned} g_{l_2}(-2) = 0, \quad g_{l_2}(-1) &= \frac{48 - 9(-4 + \pi)\pi}{32\sqrt{2}\pi^2}, \\ g'_{l_2}(-2) = 0, \quad g'_{l_2}(-1) &= \frac{3(4 + 3\pi)}{64\sqrt{2}} \end{aligned} \quad (3.106)$$

zu erfüllen. Das Polynom

$$g_{l_2}(t) = \frac{3(2+t)^2(3\pi^3(1+t) - 32(1+2t) - 24\pi(1+2t) + 2\pi^2(5+8t))}{64\sqrt{2}\pi^2} \quad (3.107)$$

genügt den genannten Forderungen. Die hieraus resultierenden Fensterfunktionen zur Konstruktion von Randbasisfunktionen am linken Signalrand sind in Abbildung 3.19 dargestellt.

Fensterfunktionen am rechten Signalrand

Am rechten Signalrand werden zur Konstruktion von Randbasisfunktionen der Kosinus-IV-Kosinus-II-Basen auf dem Intervall die Fensterfunktionen

$$w_{r_0}(t) := \begin{cases} 1 + (w(t+5) - w(t+3) + w(t+1))\sqrt{2}\sin\left(\frac{1}{2}\pi t\right) \\ \quad - (w(t+4) - w(t+2))\sqrt{2}\cos\left(\frac{1}{2}\pi t\right), & \text{für } t \in \left[-\frac{5}{2}, 0\right), \\ 1 + (-w(t+3) + g_{r_1}(t+1))\sqrt{2}\sin\left(\frac{1}{2}\pi t\right) \\ \quad + g_{r_2}(t+2)\sqrt{2}\cos\left(\frac{1}{2}\pi t\right), & \text{für } t \in \left[0, \frac{1}{2}\right), \\ 1 + g_{r_1}(t+1)\sqrt{2}\sin\left(\frac{1}{2}\pi t\right) + g_{r_2}(t+2)\sqrt{2}\cos\left(\frac{1}{2}\pi t\right), & \text{für } t \in \left[\frac{1}{2}, 1\right), \\ 0, & \text{sonst,} \end{cases} \quad (3.108)$$

$$w_{r_1}(t) := \begin{cases} w(t), & \text{für } t \in \left[-\frac{5}{2}, 1\right), \\ g_{r_1}(t), & \text{für } t \in [1, 2), \\ 0, & \text{sonst} \end{cases} \quad (3.109)$$

und

$$w_{r_2}(t) := \begin{cases} w(t), & \text{für } t \in \left[-\frac{5}{2}, 2\right), \\ g_{r_2}(t), & \text{für } t \in [2, 3), \\ 0, & \text{sonst} \end{cases} \quad (3.110)$$

definiert. Die Funktionen g_{r_1} und g_{r_2} haben zur Gewährleistung von Stetigkeit und stetiger Differenzierbarkeit der Fensterfunktion w_{r_0} , w_{r_1} und w_{r_2} die Bedingungen

$$\begin{aligned} g_{r_1}(2) &= 0, \quad g_{r_1}(1) = \frac{48 + \pi(12 + 11\pi)}{16\sqrt{2}\pi^2}, \\ g'_{r_1}(2) &= 0, \quad g'_{r_1}(1) = \frac{4 - 11\pi}{32\sqrt{2}} \end{aligned} \quad (3.111)$$

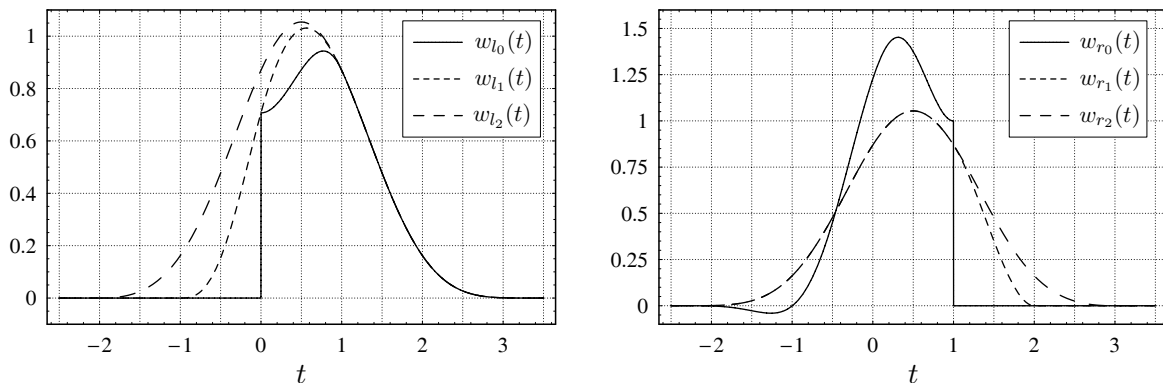


Abbildung 3.19: Sechsfachüberlappung: Fensterfunktionen zur Konstruktion von Randbasisfunktionen für die Darstellung von endlich langen Signalen am linken und am rechten Signalrand.

und

$$\begin{aligned} g_{r_2}(3) &= 0, & g_{r_2}(2) &= \frac{48 - 9(-4 + \pi)\pi}{32\sqrt{2}\pi^2}, \\ g'_{r_2}(3) &= 0, & g'_{r_2}(2) &= -\frac{3(4 + 3\pi)}{64\sqrt{2}} \end{aligned} \quad (3.112)$$

zu erfüllen. Die Polynome

$$g_{r_1}(t) = -\frac{(96 + \pi(24 + \pi(26 + 11\pi(-1 + t) - 48t) - 48t) - 192t)(-2 + t)^2}{32\sqrt{2}\pi^2} \quad (3.113)$$

und

$$g_{r_2}(t) = -\frac{3(-3 + t)^2(96 - 64t + \pi(72 - 48t + \pi(-26 + 3\pi(-2 + t) + 16t)))}{64\sqrt{2}\pi^2} \quad (3.114)$$

genügen den genannten Bedingungen. Die hieraus resultierenden Fensterfunktionen sind ebenfalls in Abbildung 3.19 dargestellt. In Analogie zum Nachweis der Rekonstruktionseigenschaften für die Dreifachüberlappung in Abschnitt 3.5.2 kann auch für die Sechsfachüberlappung durch einfache Umformungen gezeigt werden, dass die trigonometrischen Basen auf dem Intervall, die unter Verwendung der in diesem Abschnitt eingeführten Randfenster konstruiert werden, eine Konstante ausschließlich mit den Basisfunktionen ψ_j^0 reproduzieren. In Abbildung 3.20 ist für die Randintervalle die Konstruktion der Basisfunktionen ψ_j^0 und die lokale Darstellung einer Konstanten mit diesen Basisfunktionen veranschaulicht. Ausgewählte duale Fensterfunktionen zur Konstruktion von Randbasisfunktionen für die Signalanalyse sind in Abbildung D.5 (Anhang D) dargestellt.

Tabelle 3.3 fasst für die Zweifach- Vierfach- und Sechsfachüberlappung der Klasse $2m$ -fach überlappender Fensterfunktionen die u.a. nach Abbildung 3.21 bestimmten Riesz-Konstanten von trigonometrischen Basen auf dem Intervall bei verschiedenen Parametern N zusammen. Im Vergleich mit Tabelle 2.2 ist zu erkennen, dass die entwickelten Basen nur eine geringfügig schlechtere Riesz-Stabilität aufweisen.

Anhang B.2 gibt für ausgewählte Fensterfunktionen und Frequenzparameter einen Überblick über die in diesem Kapitel neu entwickelten Randbasisfunktionen zur Reproduktion von Signalen, die nur auf einem endlich langen Intervall definiert sind. Im Vergleich zu den im Anhang B.1 gezeigten Basisfunktionen auf der reellen Achse ist durch die Verwendung der neu vorgeschlagenen trigonometrischen Basen auf dem Intervall ein Fortsetzen des endlich langen Signals auf die gesamte reelle Achse nicht mehr erforderlich.

Schwerpunkt des nächsten Kapitels ist die Entwicklung schneller Transformationsalgorithmen zur Darstellung von endlich langen Signalen bzgl. der neu eingeführten Systeme von Basisfunktionen.

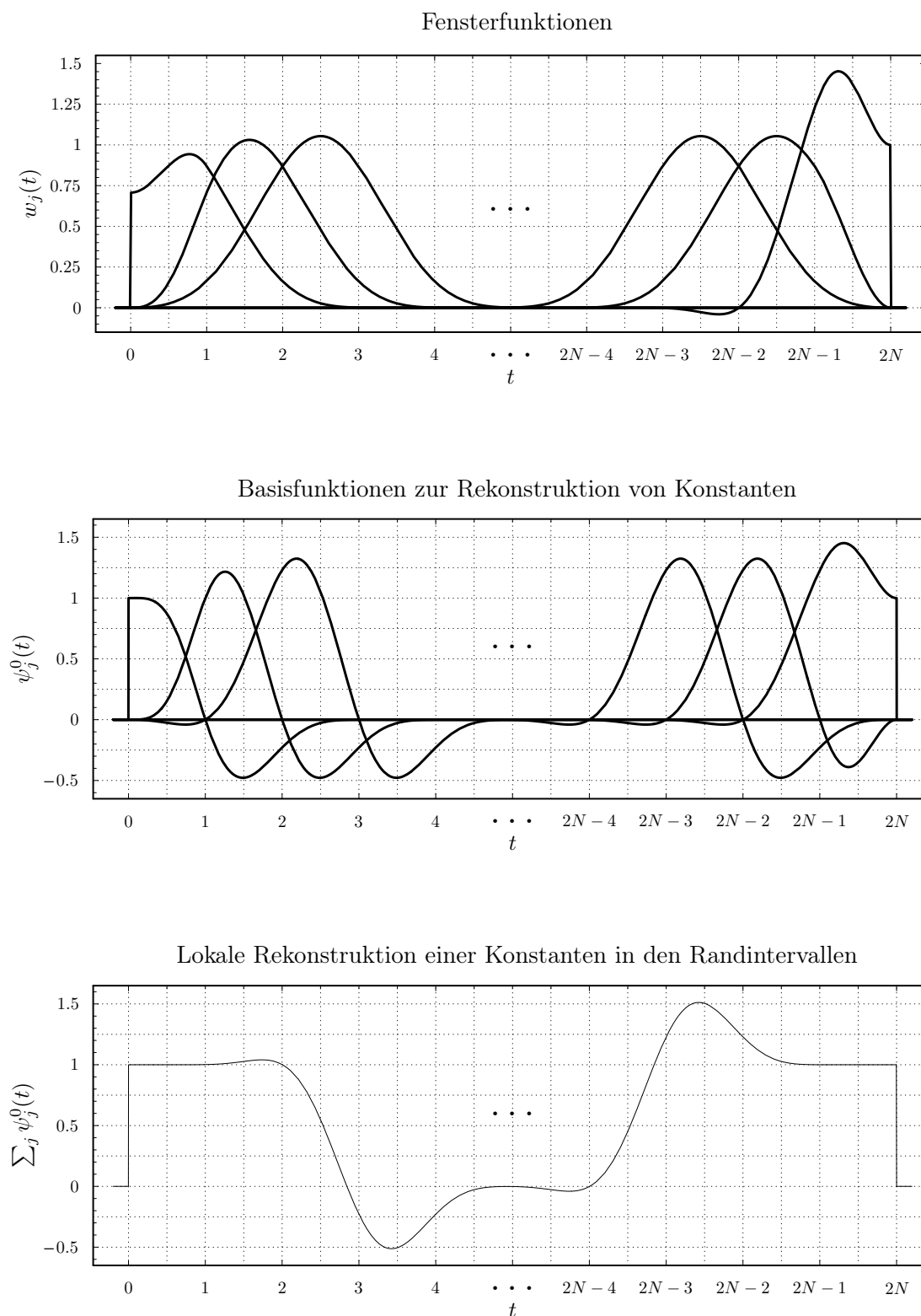


Abbildung 3.20: Sechsfachüberlappung: Fenster- und Basisfunktionen zur Verarbeitung von endlich langen Signalen, lokale Rekonstruktion von Konstanten.

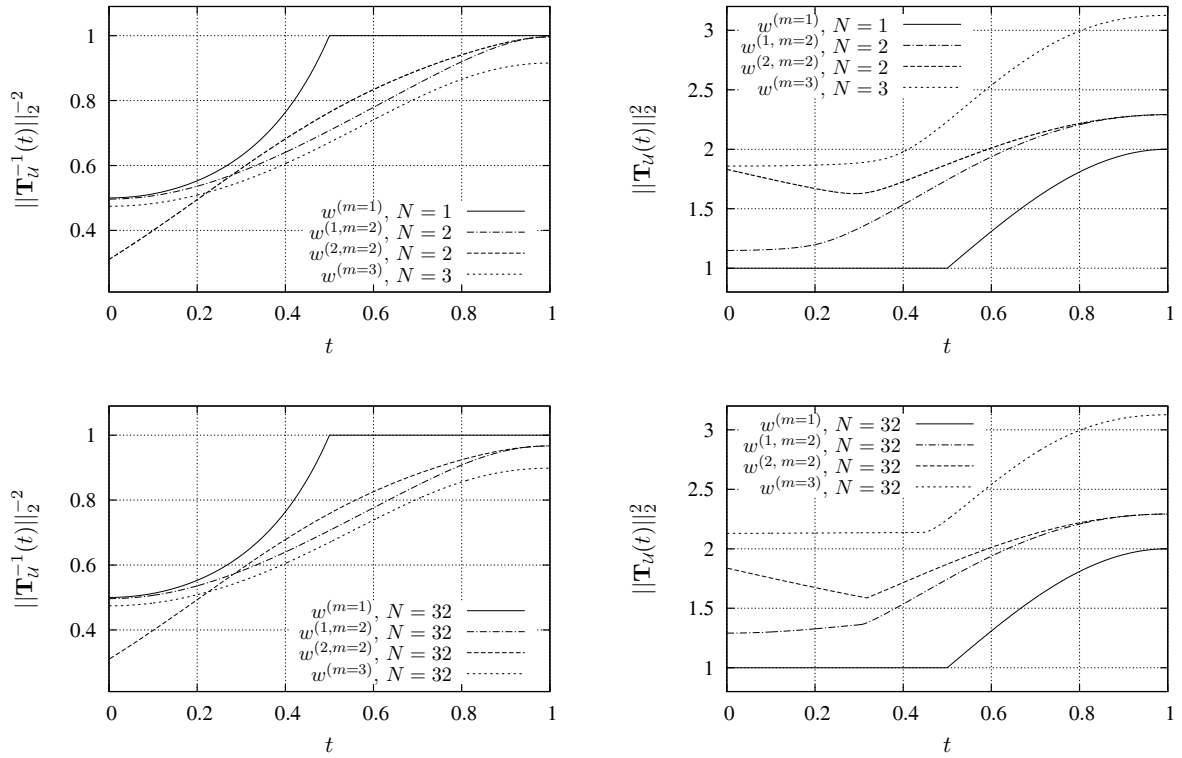


Abbildung 3.21: Zweifach-, Vierfach- und Sechsfachüberlappung (Klasse $2m$ -fach überlappender trigonometrischer Basisfunktionen für $m = 1, 2, 3$): Bestimmung der größten unteren Riesz-Schranke $A_0 = \min ||\mathbf{T}_U^{-1}(t)||_2^{-2}$ (links) und der kleinsten oberen Riesz-Schranke $B_0 = \max ||\mathbf{T}_U(t)||_2^2$ (rechts) für $t \in [0, 1]$ (vgl. Tabelle 3.3).

	A_0	B_0	κ
$w^{(m=1)}, N = 1$	0.5	2	2
$w^{(1, m=2)}$ (Ansatz 1), $N = 2$	0.4964	2.2909	2.1482
$w^{(2, m=2)}$ (Ansatz 2), $N = 2$	0.3104	2.2909	2.7167
$w^{(m=3)}, N = 3$	0.4743	3.1253	2.5670
$w^{(m=1)}, N = 16, 32$	0.5	2	2
$w^{(1, m=2)}$ (Ansatz 1), $N = 16, 32$	0.4964	2.2919	2.1488
$w^{(2, m=2)}$ (Ansatz 2), $N = 16, 32$	0.3096	2.2919	2.7208
$w^{(m=3)}, N = 16, 32$	0.4741	3.1263	2.5678

Tabelle 3.3: Zweifach-, Vierfach- und Sechsfachüberlappung: Riesz-Schranken von Basen auf dem Intervall für die Klasse $2m$ -fach überlappender trigonometrischer Basisfunktionen, $m = 1, 2, 3$.

Kapitel 4

Schnelle Transformationsalgorithmen

Für die Anwendung von überlappenden lokalen trigonometrischen Basen in der Bild-datenkompression erfolgt in diesem Kapitel der Übergang von der Betrachtungsweise kontinuierlicher Signale und Basisfunktionen zu einer diskreten Beschreibung der Signale und Transformationen. Auf der Grundlage von Klapp- und Aufklappoperatoren sowie von schnellen diskreten trigonometrischen Blocktransformationen werden effiziente Verfahren zur Realisierung der überlappenden Kosinustransformation eingeführt. Das Hauptaugenmerk der Untersuchungen richtet sich hierbei auf die Entwicklung und Analyse von schnellen Algorithmen für Klapp- und Aufklappoperatoren.

Neben der Kompressionsleistung, die mit einem Transformationscoder auf Grundlage der Darstellung von Signalen mit den im Rahmen dieser Arbeit untersuchten überlappenden lokalen trigonometrischen Basen erzielt werden kann, ist der Ressourcenbedarf zur Realisierung der überlappenden Kosinustransformation ein weiteres wesentliches Kriterium zur Beurteilung der Leistungsfähigkeit des Kompressionsverfahrens (vgl. Abschnitt 1.5). Daher stellt die Entwicklung und Untersuchung von effizienten Transformationsalgorithmen eine wichtige Aufgabe dar. Wenn bei Algorithmen von Effizienz gesprochen wird, betrachtet man im Allgemeinen die Speichereffizienz und die Laufzeiteffizienz [Wac06]. Der Speicherbedarf eines implementierten Algorithmus darf die zur Verfügung stehenden Ressourcen nicht übersteigen und die Laufzeit sollte möglichst kurz sein. Laufzeit- und Speichereffizienz einer Implementierung stehen zwar wesentlich im direkten Zusammenhang mit den Eigenschaften des Algorithmus, sie werden aber auch von der konkreten Implementierung und der verwendeten Hardware beeinflusst [Wac06, Neu96]. Da die Untersuchung unterschiedlicher Hardwarearchitekturen nicht Bestandteil dieser Arbeit ist, erfolgt die Bewertung der in diesem Kapitel vorgeschlagenen Algorithmen unabhängig von der rechentechnischen Realisierung und der zugrundeliegenden Hardware. Hierzu wird üblicherweise die Anzahl der arithmetischen Operationen bestimmt, die zur Berechnung des transformierten bzw. rekonstruierten Signals erforderlich sind.

Die direkte Realisierung der überlappenden Kosinustransformation als Matrix-Vektor-Produkt benötigt eine Vielzahl von Additionen und Multiplikationen. Wie in diesem Kapitel gezeigt wird, reduziert die Verwendung von Klapp- und Aufklappoperatoren den Berechnungsaufwand und erlaubt die Nutzung schneller trigonometrischer Blocktransformationen. Anstatt das Signal mit einer direkten Realisierung der überlappenden Kosinustransformation zu verarbeiten, werden mit dem Klappoperator zunächst die überlappenden Signalanteile, die sich aus der Wichtung des Signals mit den Fensterfunktionen ergeben, in das jeweils betrachtete Intervall geklappt. Anschließend ist das geklappte Signal entsprechend der in Kapitel 3 neu eingeführten Kombination von Kosinus-IV-Kosinus-II-Basisfunktionen mit der DCT-IV

und am Signalende mit der DCT-II zu transformieren. Für die Realisierung der trigonometrischen Blocktransformationen kann auf bekannte schnelle Algorithmen zurückgegriffen werden (vgl. Abschnitt 1.2.1). Die Rekonstruktion des Signals aus den Transformationskoeffizienten erfolgt durch die Anwendung der inversen Kosinustransformationen und der anschließenden Verarbeitung mit dem Aufklappoperator.

Für die Herleitung von schnellen Algorithmen zur Realisierung der Klapp- und Aufklappoperatoren ist es von Vorteil, wenn die Fensterfunktionen einen kurzen beschränkten Träger aufweisen. Die im Rahmen dieser Arbeit untersuchten Rekonstruktionsfenster besitzen einen kurzen Träger und die Signalrekonstruktion mit mehrfach überlappenden lokalen trigonometrischen Basen benötigt durch die Einführung des Aufklappoperators in Abhängigkeit vom Überlappungsfaktor nur geringfügig mehr Rechenoperationen als die zugrunde liegenden trigonometrischen Blocktransformationen. Die Träger der dualen Fensterfunktionen der mehrfach überlappenden lokalen trigonometrischen Basisfunktionen zur Transformation eines Signals auf dem Intervall sind dagegen genauso lang wie der Träger des Signals und es ist in Abhängigkeit von der Trägerlänge für jedes Teilintervall eine eigene individuelle duale Fensterfunktion zu berechnen. Die direkte Realisierung der Klappoperation ist somit im Vergleich zur direkten Realisierung der Aufklappoperation mit einem vielfach höheren Berechnungsaufwand verbunden. Es besteht daher primär die Notwendigkeit, Verfahren zu entwickeln, die den Rechenaufwand zur Bestimmung des geklappten Signals reduzieren.

Entsprechend der Problemstellungen ist das Kapitel wie folgt gegliedert: In Abschnitt 4.1 wird die überlappende Kosinustransformation von Signalen diskretisiert. Mit Hilfe der numerischen Integration wird gezeigt, dass sich die Darstellungskoeffizienten eines diskreten Signals bzgl. der überlappenden trigonometrischen Basis aus der diskreten trigonometrischen Blocktransformation eines diskreten geklappten Signals ergeben. In den Abschnitten 4.2 und 4.3 werden schnelle Algorithmen zur Realisierung der Klapp- und Aufklappoperatoren entwickelt. Abschnitt 4.4 fasst die Ergebnisse des Kapitels zusammen und vergleicht die für die weiteren Untersuchungen wesentlich relevanten Transformationsverfahren bzgl. des benötigten Berechnungsaufwandes. Für die Anwendung in der Bild- und Signalverarbeitung werden die hergeleiteten Algorithmen zur Realisierung der Klapp- und Aufklappoperatoren in der Programmiersprache *C* implementiert. Darüber hinaus erfolgt die Einbindung als Dynamic Link Library in Matlab.

Hinweis:

Im Folgenden wird für die überlappende Kosinustransformation die Bezeichnung LCT bzw. LCT- u (lapped cosine transform) eingeführt, wobei u den Überlappungsfaktor beschreibt.

4.1 Diskretisierung

Bisher wurden in den theoretischen Untersuchungen und Herleitungen dieser Arbeit im Wesentlichen kontinuierliche Signale, Fensterfunktionen und Basisfunktionen betrachtet. Da bei der Bildcodierung die zu komprimierenden Bilddaten jedoch als diskrete Datensätze vorliegen und digital verarbeitet werden, ist die Diskretisierung der überlappenden Kosinustransformation erforderlich.

Für die Darstellung von diskreten Signalen mit lokalen trigonometrischen Basen wird die Zeit- bzw. Ortsachse in Intervalle $[j, j + 1)$ mit $j = 0, 1, \dots, 2N - 1$ zerlegt. Das kontinuierliche Signal $f(t)$ sei an den Stützstellen

$$t = j + \frac{1}{2B} + \frac{m}{B}, \quad m = 0, 1, \dots, B - 1 \quad (4.1)$$

abgetastet. Der Parameter $B = 2^s$, $s \in \mathbb{N}$, beschreibt die Anzahl der Stützstellen pro Intervall und wird als Blockgröße bezeichnet. Für die Untersuchungen im Rahmen dieser Arbeit werden Signallängen von $2NB$ diskreten Abtastwerten angenommen. Um die Wichtungskoeffizienten $c_{j,k}$ bzgl. der Darstellung des Signals mit überlappenden lokalen trigonometrischen Basisfunktionen ψ_j^k zu ermitteln, ist nach Gleichung (3.11) das Skalarprodukt

$$c_{j,k} = \langle f, \tilde{\psi}_j^k \rangle = \langle \mathcal{T}_{\tilde{w}} f, \chi_{[j,j+1)} \theta_j^k \rangle = \int_0^1 \mathcal{T}_{\tilde{w}} f(j+t) \theta_j^k(j+t) dt \quad (4.2)$$

zu berechnen. Da der Signalterm $f(t)$ nicht bekannt ist, sondern lediglich diskrete Abtastwerte des Signals an den in Gleichung (4.1) angegebenen Stützstellen vorliegen, lässt sich das Integral in Gleichung (4.2) nur mit Hilfe numerischer Quadratur bestimmen. Als Näherung eignet sich die summierte Tangententrapezformel [BSMM08]

$$\int_a^b f(x) dx \approx h \cdot \sum_{m=0}^{B-1} f\left(a + h \cdot \frac{2m+1}{2}\right), \quad h = \frac{b-a}{B}. \quad (4.3)$$

Ausgehend von Gleichung (4.1) unterteilt man das Intervall $[a, b]$ in B gleich große Teilintervalle der Länge h . Als Stützstellen für die Tangenten werden die Stellen verwendet, an denen das Signal nach Gleichung (4.1) abgetastet wurde. Für die trigonometrischen Funktionen vom Kosinus-IV-Typ in den Intervallen $[j, j+1)$ mit $j = 0, 1, \dots, 2N-2$, die durch

$$\theta_j^k(t) := \sqrt{2} \cos\left(\left(k + \frac{1}{2}\right)(t-j)\pi\right) \quad (4.4)$$

definiert sind, ist nach Einsetzen in Gleichung (4.2) das Integral

$$\langle f, \tilde{\psi}_j^k \rangle = \int_0^1 \mathcal{T}_{\tilde{w}} f(j+t) \sqrt{2} \cos\left(\left(k + \frac{1}{2}\right)t\pi\right) dt \quad (4.5)$$

zu berechnen. Die numerische Integration nach der in Gleichung (4.3) angegebenen summierten Tangententrapezformel führt zu dem Resultat

$$\langle f, \tilde{\psi}_j^k \rangle \approx \frac{1}{B} \sum_{m=0}^{B-1} \mathcal{T}_{\tilde{w}} f\left(j + \frac{2m+1}{2B}\right) \sqrt{2} \cos\left(\left(k + \frac{1}{2}\right)\left(m + \frac{1}{2}\right)\frac{\pi}{B}\right) =: c_{j,k}. \quad (4.6)$$

Die Koeffizienten $c_{j,k}$ stellen somit näherungsweise das Skalarprodukt zwischen dem kontinuierlichen Signal f und der kontinuierlichen dualen Basisfunktion $\tilde{\psi}_j^k$ dar. Vergleicht man jedoch Gleichung (4.6) mit der Definition der diskreten Kosinustransformation in den Gleichungen (1.27) und (1.30), so ist festzustellen, dass es sich bei den Koeffizienten $c_{j,k}$ um die Transformationskoeffizienten der DCT-IV des nach Gleichung (4.1) abgetasteten und anschließend geklappten Signals $\mathcal{T}_{\tilde{w}} f\left(j + \frac{2m+1}{2B}\right)$ handelt. Die an den Stützstellen abgetasteten Signalwerte lassen sich aus den Koeffizienten $c_{j,k}$ durch die Anwendung der selbstinversen DCT-IV und des Aufklappoperators wieder exakt rekonstruieren. Analog zu den Herleitungen für den Kosinus-IV-Fall ergibt sich für Kosinus-II-Funktionen

$$\theta_j^k(t) := \sqrt{2} \cos(k(t-j)\pi), \quad (4.7)$$

die im Intervall $[2N-1, 2N)$ Anwendung finden, das Skalarprodukt

$$\langle f, \tilde{\psi}_j^k \rangle = \int_0^1 \mathcal{T}_{\tilde{w}} f(j+t) \sqrt{2} \cos(kt\pi) dt. \quad (4.8)$$

Durch numerische Integration unter Nutzung der summierten Tangententrapezformel erhält man

$$\langle f, \tilde{\psi}_j^k \rangle \approx \frac{1}{B} \sum_{m=0}^{B-1} \mathcal{T}_{\tilde{w}} f \left(j + \frac{2m+1}{2B} \right) \sqrt{2} \cos \left(k \left(m + \frac{1}{2} \right) \frac{\pi}{B} \right) =: c_{j,k}. \quad (4.9)$$

Nach den Gleichungen (1.27) und (1.28) sind die Koeffizienten $c_{j,k}$ die Transformationskoeffizienten der DCT-II des nach Gleichung (4.1) diskret abgetasteten und anschließend geklappten Signals $\mathcal{T}_{\tilde{w}} f \left(j + \frac{2m+1}{2B} \right)$. Die an den Stützstellen abgetasteten Signalwerte werden durch die Anwendung der DCT-III und des Aufklappoperators wieder exakt rekonstruiert.

Die Verwendung von Klapp- und Aufklappoperatoren ermöglicht daher für die Realisierung von überlappenden Kosinustransformationen die Nutzung schneller diskreter trigonometrischer Blocktransformationen vom Typ II, III und IV (vgl. Abschnitt 1.2.1).

4.2 Schnelle Realisierung von Aufklappoperatoren

Um Algorithmen zur schnellen Realisierung von Aufklappoperatoren für die im Rahmen dieser Arbeit untersuchten Fenster mit kurzem Träger herzuleiten, wird nachfolgend der Aufklappoperator durch das Matrix-Vektor-Produkt

$$\vec{f}^{(m)} = \mathbf{T}_u^{(m)} \vec{f}_\tau^{(m)} \quad (4.10)$$

beschrieben. Diese Betrachtungsweise bildet insbesondere auch die Grundlage für die Entwicklung von leistungsfähigen Algorithmen zur Realisierung des Klappoperators in Abschnitt 4.3. Für die Transformation von Signalen, die auf einem Intervall $[0, 2N]$ definiert sind, lassen sich die Koeffizienten der Abbildungsmatrizen $\mathbf{T}_u^{(m)}$ aus der in Abschnitt 3.3 neu eingeführten Beschreibung des Aufklappoperators für endliche Träger ableiten. In Analogie zu den Ausführungen in Abschnitt 4.1 werden nachfolgend Signale auf dem Intervall betrachtet, die an den Stützstellen

$$t = 2j + \frac{2m+1}{2B}, \quad t = 2j+1 + \frac{2m+1}{2B}, \quad j = 0, 1, \dots, N-1, \quad m = 0, 1, \dots, B-1 \quad (4.11)$$

abgetastet wurden und dementsprechend eine Länge von $L = 2NB$ Abtastwerten aufweisen. Aus Gleichung (3.34) ergibt sich für den Aufklappoperator

$$\begin{aligned} & \begin{pmatrix} \mathcal{U}_w f_\tau[2Br + m] \\ \mathcal{U}_w f_\tau[2B(r+1) - m - 1] \end{pmatrix} \\ &= \sum_{j=0}^{N-2} (-1)^{j+r} \begin{pmatrix} w_{2j}[2Br + m] & w_{2j+1}[2Br + m] \\ -w_{2j}[2B(r+1) - m - 1] & w_{2j+1}[2B(r+1) - m - 1] \end{pmatrix} \\ & \quad \cdot \begin{pmatrix} f_\tau[2Bj + m] \\ f_\tau[2B(j+1) - m - 1] \end{pmatrix} \\ & \quad + \begin{pmatrix} -(-1)^{N+r} w_{2N-2}[2Br + m] & w_{2N-1}[2Br + m] \\ (-1)^{N+r} w_{2N-2}[2B(r+1) - m - 1] & w_{2N-1}[2B(r+1) - m - 1] \end{pmatrix} \\ & \quad \cdot \begin{pmatrix} f_\tau[2B(N-1) + m] \\ f_\tau[2NB - m - 1] \end{pmatrix}, \end{aligned} \quad (4.12)$$

wobei folgende Wertezuweisungen Anwendung finden:

$$\begin{aligned} f_{\mathcal{T}}[2Bj + m] &= f_{\mathcal{T}}\left(2j + \frac{2m+1}{2B}\right), & f_{\mathcal{T}}[2B(j+1) - m - 1] &= f_{\mathcal{T}}\left(2j + 2 - \frac{2m+1}{2B}\right), \\ w_s[2Br + m] &= w_s\left(2r + \frac{2m+1}{2B}\right), & w_s[2B(r+1) - m - 1] &= w_s\left(2r + 2 - \frac{2m+1}{2B}\right) \end{aligned} \quad (4.13)$$

mit $s \in \{2j, 2j+1, 2N-2, 2N-1\}$. Bei dieser Beschreibung überführt die Abbildungsmatrix $\mathbf{T}_u^{(m)}$ des Aufklappoperators den Vektor

$$\vec{f}_{\mathcal{T}}^{(m)} = (f_{\mathcal{T}}[m], f_{\mathcal{T}}[2B - m - 1], f_{\mathcal{T}}[2B + m], f_{\mathcal{T}}[4B - m - 1], \dots, f_{\mathcal{T}}[2N - m - 1])^T, \quad (4.14)$$

der das in diskreten Werten vorliegende geklappte Signal repräsentiert, in den Vektor

$$\vec{f}^{(m)} = (f[m], f[2B - m - 1], f[2B + m], f[4B - m - 1], \dots, f[2N - m - 1])^T, \quad (4.15)$$

der das diskrete aufgeklappte Signal repräsentiert. Das Matrix-Vektor-Produkt ist für jeden Parameter $m = 0, 1, \dots, B-1$ zu bestimmen. Bei einer Blockgröße von z.B. $B = 8$ sind somit insgesamt 8 Matrix-Vektor-Multiplikationen durchzuführen. Unter der Bedingung, dass in den inneren Intervallen Translate einer Fensterfunktion w verwendet werden, erhält man z.B. für dreifach überlappende Fenster die in Abbildung 4.1 dargestellte Realisierung des Aufklappoperators als Matrix-Vektor-Produkt.

Die Abbildungsmatrizen der Aufklappoperatoren weisen Bandstruktur auf, wobei die Bandbreite β vom Überlappungsfaktor u der Rekonstruktionsfenster abhängt. Sie ergibt sich aus der Beziehung $\beta = \lfloor \frac{u}{2} \rfloor 2 + 1$. Aufgrund der Bandstruktur besitzen die Abbildungsmatrizen des Aufklappoperators eine hohe Anzahl von Nullelementen. Um überflüssige Operationen mit Nullen zu vermeiden, ist die Realisierung des Aufklappoperators mit der exemplarisch gewählten Abbildungsmatrix

$$\mathbf{T}_u^{(m)} = \begin{pmatrix} \overbrace{\begin{matrix} a_{0,0}^{(m)} & a_{0,1}^{(m)} & a_{0,2}^{(m)} & 0 \\ a_{1,0}^{(m)} & a_{1,1}^{(m)} & a_{1,2}^{(m)} & a_{1,3}^{(m)} \\ a_{2,0}^{(m)} & a_{2,1}^{(m)} & \ddots & \ddots & \ddots \end{matrix}}^o & & & \\ 0 & a_{3,1}^{(m)} & \ddots & & a_{2N-4,2N-2}^{(m)} & 0 \\ & \ddots & & & a_{2N-3,2N-2}^{(m)} & a_{2N-3,2N-1}^{(m)} \\ & & a_{2N-2,2N-4}^{(m)} & a_{2N-2,2N-3}^{(m)} & a_{2N-2,2N-2}^{(m)} & a_{2N-2,2N-1}^{(m)} \\ & & 0 & a_{2N-1,2N-3}^{(m)} & a_{2N-1,2N-2}^{(m)} & a_{2N-1,2N-1}^{(m)} \end{pmatrix}, \quad (4.16)$$

wobei $o = \frac{\beta-1}{2}$ die obere und untere Bandbreite der Bandmatrix beschreibt, an die Bandstruktur anzupassen. Zur recheneffizienten Implementierung des Aufklappoperators für fast beliebige Überlappungsfaktoren u lässt sich der in Abbildung 4.2 dargestellte allgemeine Algorithmus nutzen. Eine einfache Abschätzung des Berechnungsaufwandes pro Signalwert ergibt sich direkt aus dem Überlappungsfaktor der verwendeten Fensterfunktion. Zur Bestimmung der oberen Schranke für die Anzahl arithmetischer Rechenoperationen pro Signalwert erhält man

$$O_u = 4 \left\lfloor \frac{u}{2} \right\rfloor + 1, \quad u = \text{Überlappungsfaktor}. \quad (4.17)$$

$$\begin{pmatrix} f[m] \\ f[2B-m-1] \\ f[2B+m] \\ f[4B-m-1] \\ f[4B+m] \\ \vdots \\ f[L-4B-m-1] \\ f[L-4B+m] \\ f[L-2B-m-1] \\ f[L-2B+m] \\ f[L-m-1] \end{pmatrix} = \begin{pmatrix} \begin{matrix} w_{n_0}[m] & w_{n_1}[-B+m] & 0 \\ -w_{n_0}[2B-m-1] & w_{n_1}[B-m-1] & w[-m-1] \\ 0 & -w_{n_1}[B+m] & 0 \end{matrix} & \begin{matrix} 0 \\ w[m] \\ w[-B+m] \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & -w[2B-m-1] & w[B-m-1] \\ 0 & -w[B+m] & w[m] \end{matrix} & \begin{matrix} w[-m-1] \\ w[-B+m] \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \vdots & \vdots & \vdots \\ \begin{matrix} 0 & -w[2B-m-1] & w[B-m-1] \\ 0 & -w[B+m] & w[m] \end{matrix} & \begin{matrix} w[-m-1] \\ w[-B+m] \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & -w[2B-m-1] & w[B-m-1] \\ 0 & -w[B+m] & w[m] \end{matrix} & \begin{matrix} w[-m-1] \\ w[-B+m] \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \end{pmatrix} \cdot \begin{pmatrix} f_1[m] \\ f_1[2B-m-1] \\ f_1[2B+m] \\ f_1[4B-m-1] \\ f_1[4B+m] \\ \vdots \\ f_1[L-4B-m-1] \\ f_1[L-4B+m] \\ f_1[L-2B-m-1] \\ f_1[L-2B+m] \\ f_1[L-m-1] \end{pmatrix}$$

Abbildung 4.1: Bandmatrix zur Realisierung der Aufklappoperation als Matrix-Vektor-Produkt für dreifach überlappende Fenster, $m = 0, 1, \dots, B-1$ mit $B = \text{Blockgröße}$, $L = 2NB = \text{Signallänge}$ als Anzahl der diskreten Abtastwerte.

```

For  $i = 0$  To  $2N - 1$ 
     $k = \max(0, i - o)$ 
     $f_i^{(m)} = f_{\mathcal{T}k}^{(m)} a_{i,k}^{(m)}$ 
    For  $j = k + 1$  To  $\min(i + o, 2N - 1)$ 
         $f_i^{(m)} = f_i^{(m)} + f_{\mathcal{T}j}^{(m)} a_{i,j}^{(m)}$ 
    End
End

```

Abbildung 4.2: Allgemein gehaltener Algorithmus zur Realisierung des Aufklappoperators für fast beliebige Überlappungsfaktoren u , $m = 0, 1, \dots, B - 1$ mit $B =$ Blocklänge, $L = 2NB =$ Signallänge als Anzahl der diskreten Abtastwerte (vgl. Matrix-Vektor-Produkt für Bandmatrizen u.a. in [Sch03]).

Um die exakte Anzahl der Rechenoperationen zu bestimmen, ist der reduzierte Berechnungsaufwand für die Randintervalle zu berücksichtigen (vgl. Tabelle 4.1).

Ausgehend von dem allgemeinen Verfahren zur schnellen Realisierung des Matrix-Vektor-Produktes mit Bandmatrizen kann für die in dieser Arbeit untersuchten Aufklappoperatoren unter Verwendung geradzahlicher Überlappungsfaktoren der Berechnungsaufwand um zwei weitere arithmetische Operationen pro Signalwert reduziert werden. Für geradzahlige Überlappungsfaktoren u lassen sich die Aufklappoperatoren als Spezialfall der Aufklappoperatoren für ungeradzahlige Überlappungsfaktoren $u + 1$ interpretieren. Bei beiden Überlappungsfaktoren besitzt das Matrixband die gleiche Bandbreite β . Das Matrixband weist jedoch für geradzahlige Überlappungsfaktoren eine Besonderheit auf. In Abhängigkeit vom Parameter m treten in den äußeren Nebendiagonalen Koeffizienten mit dem Wert Null auf. Für die Vierfachüberlappung gelten z.B. die Beziehungen

$$\begin{aligned}
 w[-2B + m] &= 0, & w_{r_1}[-2B + m] &= 0, & \text{für } m &= 0, 1, \dots, \frac{B}{2} - 1, \\
 w[3B - m - 1] &= 0, & w_{l_1}[3B - m - 1] &= 0, & & \\
 w[2B + m] &= 0, & w_{l_0}[2B + m] &= 0, & \text{für } m &= \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1. \\
 w[-B - m - 1] &= 0, & w_{r_0}[-B - m - 1] &= 0, & &
 \end{aligned}
 \tag{4.18}$$

Die beiden hieraus resultierenden Matrix-Vektor-Produkte sind in den Abbildungen 4.3 und 4.4 veranschaulicht. Da die Nullkoeffizienten in den äußeren Nebendiagonalen in Abhängigkeit vom Parameter m auftreten, sind bei der Implementierung des Aufklappoperators die Fälle $m = 0, 1, \dots, \frac{B}{2} - 1$ und $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$ zu unterscheiden.

Neben der Einsparung von arithmetischen Operationen kommt für große, schwach besetzte Abbildungsmatrizen von Aufklappoperatoren auch der Reduktion des Speicheraufwandes eine wesentliche Bedeutung zu. Zielstellung ist hierbei die günstige Speicherung der von Null verschiedenen Matrixelemente sowie ein effizienter Zugriff auf diese Elemente. In [Mae88] werden verschiedene Kompaktspeichertechniken in Abhängigkeit von der Struktur schwach besetzter Matrizen vorgestellt. Für Bandmatrizen gestaltet sich eine kompakte Speicherung der von Null verschiedenen Elemente einfach. Statt der Abbildungsmatrix $\mathbf{T}_u^{(m)}$ des Aufklappoperators, die

$$\begin{pmatrix} f[m] \\ f[2B-m-1] \\ f[2B+m] \\ f[4B-m-1] \\ f[4B+m] \\ \vdots \\ f[L-4B-m-1] \\ f[L-4B+m] \\ f[L-2B-m-1] \\ f[L-2B+m] \\ f[L-m-1] \end{pmatrix} = \begin{pmatrix} \begin{matrix} w_{b_0}[m] & w_{t_1}[-B+m] \\ -w_{b_0}[2B-m-1] & w_{t_1}[B-m-1] \\ -w_{b_0}[2B+m] & -w_{t_1}[B+m] \end{matrix} & \begin{matrix} 0 & 0 \\ w[-m-1] & -w[-B-m-1] \\ w[m] & w[-B+m] \end{matrix} \\ \begin{matrix} 0 & -w[2B-m-1] & w[B-m-1] & -w[-B-m-1] & 0 \\ 0 & -w[2B+m] & -w[B+m] & w[m] & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & -w[2B-m-1] & w[B-m-1] & -w[-B-m-1] & 0 \\ 0 & -w[2B+m] & -w[B+m] & w[m] & 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} w_{t_1}[m] & w[-B+m] & 0 \\ w[m] & w[-B+m] & 0 \\ \vdots & \vdots & \vdots \\ w_{t_1}[m] & w[-B+m] & 0 \\ w[m] & w[-B+m] & 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} w_{t_1}[-m-1] & w[B-m-1] & -w[-B-m-1] & 0 \\ w[m] & w[-B+m] & w[m] & 0 \\ \vdots & \vdots & \vdots & \vdots \\ w_{t_1}[-m-1] & w[B-m-1] & -w[-B-m-1] & 0 \\ w[m] & w[-B+m] & w[m] & 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} w_{t_1}[m] & w[-B+m] & 0 \\ w[m] & w[-B+m] & 0 \\ \vdots & \vdots & \vdots \\ w_{t_1}[m] & w[-B+m] & 0 \\ w[m] & w[-B+m] & 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} w_{t_1}[-m-1] & w[B-m-1] & -w[-B-m-1] & 0 \\ w[m] & w[-B+m] & w[m] & 0 \\ \vdots & \vdots & \vdots & \vdots \\ w_{t_1}[-m-1] & w[B-m-1] & -w[-B-m-1] & 0 \\ w[m] & w[-B+m] & w[m] & 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{matrix} \end{pmatrix} \begin{pmatrix} f[m] \\ f[2B-m-1] \\ f[2B+m] \\ f[4B-m-1] \\ f[4B+m] \\ \vdots \\ f[L-4B-m-1] \\ f[L-4B+m] \\ f[L-2B-m-1] \\ f[L-2B+m] \\ f[L-m-1] \end{pmatrix}$$

Abbildung 4.3: Bandmatrix zur Realisierung der Aufklappoperation als Matrix-Vektor-Produkt für vierfach überlappende Fenster, $m = 0, 1, \dots, \frac{B}{2} - 1$ mit $B = \text{Blockgröße}$, $L = 2NB$ Signallänge.

$$\begin{pmatrix} f[m] \\ f[2B-m-1] \\ f[2B+m] \\ f[4B-m-1] \\ f[4B+m] \\ \vdots \\ \vdots \\ \vdots \\ f[L-4B-m-1] \\ f[L-4B+m] \\ f[L-2B-m-1] \\ f[L-2B+m] \\ f[L-m-1] \end{pmatrix} = \begin{pmatrix} w_{i_0}[m] & w_{i_1}[-B+m] & 0 & -w[-2B+m] & 0 & \dots & 0 \\ -w_{i_0}[2B-m-1] & w_{i_1}[B-m-1] & 0 & w[-m-1] & 0 & \dots & 0 \\ 0 & -w_{i_1}[B+m] & w[-B+m] & -w[-2B+m] & 0 & \dots & 0 \\ 0 & -w_{i_1}[3B-m-1] & w[B-m-1] & w[-m-1] & 0 & \dots & 0 \\ 0 & 0 & -w[B+m] & w[m] & w[-B+m] & -w[-2B+m] & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & w[B-m-1] & w[-m-1] & 0 \\ f[L-4B-m-1] & 0 & -w[3B-m-1] & -w[2B-m-1] & w[B-m-1] & w[-m-1] & 0 \\ f[L-4B+m] & 0 & 0 & 0 & 0 & w[m] & w[-B+m] \\ f[L-2B-m-1] & 0 & 0 & 0 & -w[3B-m-1] & -w[2B-m-1] & w[B-m-1] \\ f[L-2B+m] & 0 & 0 & 0 & 0 & w_{i_1}[m] & w_{i_1}[-m-1] \\ f[L-m-1] & 0 & 0 & 0 & 0 & -w[3B-m-1] & -w[2B-m-1] \end{pmatrix} \begin{pmatrix} f[m] \\ f[2B-m-1] \\ f[2B+m] \\ f[4B-m-1] \\ f[4B+m] \\ \vdots \\ \vdots \\ \vdots \\ f[L-4B-m-1] \\ f[L-4B+m] \\ f[L-2B-m-1] \\ f[L-2B+m] \\ f[L-m-1] \end{pmatrix}$$

Abbildung 4.4: Bandmatrix zur Realisierung der Aufklappoperation als Matrix-Vektor-Produkt für vierfach überlappende Fenster, $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$ mit $B = \text{Blockgröße}$, $L = 2NB = \text{Signallänge}$.

eine Größe von $2N \times 2N$ Koeffizienten besitzt, wird mit der Kompaktspeicherung

$$\begin{pmatrix} \times & + & + & & & \\ - & \times & + & + & & \\ - & - & \times & + & + & \\ & - & - & \times & + & + \\ & & - & - & \times & + \\ & & & - & - & \times \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & \times & + & + \\ 0 & - & \times & + & + \\ - & - & \times & + & + \\ - & - & \times & + & + \\ - & - & \times & + & 0 \\ - & - & \times & 0 & 0 \end{pmatrix} \quad (4.19)$$

nach [Mae88] nur das $2N \times \beta$ - Rechteck des Matrixbandes gespeichert. Durch die Indexabbildung

$$(v, w) \mapsto (v, w - v + o) \quad (4.20)$$

der Matrixkoeffizienten $a_{v,w}$ wird der in Abbildung 4.2 dargestellte Algorithmus zur Realisierung des Aufklappoperators an die kompakte Speicherung der Koeffizienten angepasst. Werden nach Kapitel 3 den inneren Signalintervallen Translate einer Fensterfunktion zugeordnet, so treten innerhalb des Matrixbandes der Abbildungsmatrizen des Aufklappoperators Koeffizienten mit den gleichen Funktionswerten auf. Der Speicheraufwand für diese Matrizen lässt sich folglich weiter reduzieren, indem man die benötigten Koeffizienten nur einmal abspeichert.

4.3 Schnelle Realisierung von Klappoperatoren

Unter der Voraussetzung, dass zu der Abbildungsmatrix $\mathbf{T}_u^{(m)}$ des Aufklappoperators eine Inverse $\mathbf{T}_\tau^{(m)}$ existiert, kann das geklappte Signal $\vec{f}_\tau^{(m)}$ mit dem Matrix-Vektor-Produkt

$$\vec{f}_\tau^{(m)} = \mathbf{T}_\tau^{(m)} \vec{f}^{(m)} \quad (4.21)$$

aus dem Signalvektor $\vec{f}^{(m)}$ bestimmt werden. Bei der Verwendung von zweifach überlappenden Fenstern weist die Matrix $\mathbf{T}_\tau^{(m)}$ des Klappoperators ebenso wie die Matrix $\mathbf{T}_u^{(m)}$ des Aufklappoperators eine Bandstruktur mit der Bandbreite $\beta = 3$ auf. Der Klappoperator lässt sich in diesem Fall mit wenigen Rechenoperationen durch ein Matrix-Vektor-Produkt realisieren, indem die überflüssigen Operationen mit den Nullelementen außerhalb des Matrixbandes vermieden werden (vgl. Anhang E.1). Für die im Rahmen dieser Arbeit untersuchten mehrfach überlappenden Fensterfunktionen ist die inverse Matrix $\mathbf{T}_\tau^{(m)}$ hingegen voll besetzt, da die Träger der dualen Fensterfunktionen mit $L = 2NB$ Abtastwerten genauso lang sind wie der Träger des zu verarbeitenden Signals. Bei der Realisierung des Klappoperators über das Matrix-Vektor-Produkt nach Gleichung (4.21) wird somit eine Vielzahl von Additionen und Multiplikationen benötigt. Die Bestimmung der dualen Fensterfunktionen bzw. die Bestimmung der inversen Matrix $\mathbf{T}_\tau^{(m)}$ ist ebenfalls mit einem hohen Rechenaufwand verbunden. Es gilt gemeinhin als nicht effizient, die Inverse einer Matrix zu bestimmen, wenn lediglich das Ergebnis der linearen Abbildung, die durch die inverse Matrix dargestellt wird, berechnet werden soll. Zur schnellen Berechnung des geklappten Signals wird folglich das Matrix-Vektor-Produkt $\vec{f}^{(m)} = \mathbf{T}_u^{(m)} \vec{f}_\tau^{(m)}$ des Aufklappoperators als lineares Gleichungssystem betrachtet und die Bandstruktur der Abbildungsmatrix des Aufklappoperators ausgenutzt. Das geklappte Signal $\vec{f}_\tau^{(m)}$ erhält man durch Lösen des Gleichungssystems für jedes $m = 0, 1, \dots, B - 1$. Bei einer Blockgröße von z.B. $B = 8$ sind somit insgesamt 8 Gleichungssysteme aufzustellen und anschließend zu lösen.

Die Realisierung mit linearen Gleichungssystemen zeichnet sich durch folgende Vorteile aus:

- Eine Berechnung der an die Signallänge angepassten dualen Fensterfunktionen \tilde{w}_j , $j = 0, 1, \dots, 2N - 1$, mit langem Träger ist nicht erforderlich.
- Es werden keine Matrix-Vektor-Multiplikationen mit vollbesetzten Transformationsmatrizen $\mathbf{T}_{\mathcal{T}}^{(m)}$, $m = 0, 1, \dots, B - 1$, durchgeführt.

Es existieren verschiedene Verfahren zum Lösen von linearen Gleichungssystemen. Weisen die Matrizen eine Bandstruktur mit relativ geringer Bandbreite auf, so wie es bei den Abbildungsmatrizen der Aufklappoperatoren der Fall ist, dann führen angepasste Gauß-Eliminationsverfahren zu sehr leistungsfähigen Algorithmen. In der Bildverarbeitung spielt die Tatsache, dass Gleichungssysteme mit derselben Koeffizientenmatrix $\mathbf{T}_{\mathcal{T}}^{(m)}$, aber mehreren verschiedenen rechten Seiten¹ $\vec{f}^{(m)}$ zu lösen sind, eine entscheidende Rolle bei der Auswahl des Lösungsverfahrens. Unter Berücksichtigung der vorgenannten Faktoren wird zur effizienten Realisierung von Klappoperatoren die LU-Zerlegung [Mae88] gewählt.

4.3.1 Algorithmen auf Grundlage der Standard-LU-Zerlegung

Bei der Realisierung des Klappoperators durch Lösen von linearen Gleichungssystemen $\vec{f}^{(m)} = \mathbf{T}_{\mathcal{U}}^{(m)} \vec{f}_{\mathcal{T}}^{(m)}$ mittels LU-Zerlegung wird die in Gleichung (4.16) beispielhaft dargestellte Abbildungsmatrix $\mathbf{T}_{\mathcal{U}}^{(m)}$ des Aufklappoperators in eine linke untere Dreiecksmatrix

$$\mathbf{L}^{(m)} = \begin{pmatrix} 1 & & & & \\ l_{1,0}^{(m)} & 1 & & & \\ l_{2,0}^{(m)} & l_{2,1}^{(m)} & 1 & & \\ & \ddots & & \ddots & \\ & & l_{2N-1,2N-3}^{(m)} & l_{2N-1,2N-2}^{(m)} & 1 \end{pmatrix} \quad (4.22)$$

und in eine rechte obere Dreiecksmatrix

$$\mathbf{U}^{(m)} = \begin{pmatrix} u_{0,0}^{(m)} & u_{0,1}^{(m)} & u_{0,2}^{(m)} & & \\ & \ddots & \ddots & \ddots & \\ & & & & u_{2N-3,2N-1}^{(m)} \\ & & & & u_{2N-2,2N-1}^{(m)} \\ & & & & u_{2N-1,2N-1}^{(m)} \end{pmatrix} \quad (4.23)$$

faktoriisiert, wobei sich die Bandstruktur von $\mathbf{T}_{\mathcal{U}}^{(m)}$ vollständig auf die Matrizen $\mathbf{L}^{(m)}$ und $\mathbf{U}^{(m)}$ überträgt [Kan05, MNW04]. Im Anschluss an die LU-Zerlegung folgt als eigentlicher Klappvorgang für jede Bildzeile bzw. Bildspalte durch Vorwärtseinsetzen das Lösen des Gleichungssystems $\mathbf{L}^{(m)} \vec{x}^{(m)} = \vec{f}^{(m)}$ mit dem Zwischenergebnis $\vec{x}^{(m)}$ und schließlich durch Rückwärtseinsetzen das Lösen des Gleichungssystems $\mathbf{U}^{(m)} \vec{f}_{\mathcal{T}}^{(m)} = \vec{x}^{(m)}$ mit dem Endergebnis des geklappten Signals $\vec{f}_{\mathcal{T}}^{(m)}$. Die wesentlichen Vorteile dieser Verarbeitungsstrategie sind:

- Bei den Gleichungssystemen $\mathbf{L}^{(m)} \vec{x}^{(m)} = \vec{f}^{(m)}$ und $\mathbf{U}^{(m)} \vec{f}_{\mathcal{T}}^{(m)} = \vec{x}^{(m)}$ handelt es sich um Dreieckssysteme, die sich sehr einfach und direkt durch Vorwärts- bzw. Rückwärts-substitution lösen lassen.

¹Die Gleichungssysteme sind für jeweils mehrere Bildzeilen bzw. für jeweils mehrere Bildspalten zu lösen.

- Für die Verarbeitung von mehrdimensionalen Signalen ist die LU-Faktorisierung für jeden Parameter $m = 0, 1, \dots, B - 1$ nur einmal je Dimensionsrichtung erforderlich.

Eine Reduzierung des Berechnungsaufwandes lässt sich sowohl bei der LU-Zerlegung als auch beim Klappvorgang durch die Berücksichtigung der Bandstruktur der Matrix $\mathbf{T}_u^{(m)}$ sowie der Bandstrukturen der LU-Matrizen $\mathbf{L}^{(m)}$ und $\mathbf{U}^{(m)}$ erzielen, indem die überflüssigen Rechenoperationen mit den Nullelementen außerhalb der Matrixbänder vermieden werden. Hierfür geeignete Algorithmen sind in den Abbildungen 4.5 und 4.6 dargestellt.

Da für geradzahlige Überlappungsfaktoren auch in den äußeren Nebendiagonalen der Matrixbänder Nullkoeffizienten auftreten, besteht in Analogie zu den Betrachtungen in Abschnitt 4.2 die Möglichkeit, zwei weitere arithmetische Operationen pro Signalwert einzusparen. Darüberhinaus treten bei der LU-Zerlegung für geradzahlige Überlappungsfaktoren unter der Voraussetzung, dass den inneren Signalintervallen Translate einer Fensterfunktion zugeordnet sind, wiederholt identische Berechnungen auf, so dass sich die Anzahl der arithmetischen Operationen weiter reduzieren lässt.

Um bei der LU-Zerlegung Speicherplatz einzusparen, erfolgt die Faktorisierung der Koeffizientenmatrizen inplace [Kan05]. Die LU-Koeffizienten werden hierbei direkt im Speicher der Abbildungsmatrix $\mathbf{T}_u^{(m)}$ des Aufklappoperators abgelegt. Zur effizienten Speicherung von $\mathbf{T}_u^{(m)}$ eignet sich wiederum die durch die Gleichungen (4.19) und (4.20) beschriebene Kompaktspeichertechnik [Mae88], bei der nur das $2N \times \beta$ -Rechteck des Matrixbandes gespeichert wird. Die Realisierung der Klappoperationen erfolgt ebenfalls inplace, indem das geklappte Signal f_τ direkt in den Speicher von f geschrieben wird.

Im Anhang E sind an die Zweifach-, Dreifach- und Vierfachüberlappung angepasste Algorithmen¹ zur effizienten Realisierung von Klapp- und Aufklappoperatoren angegeben. Durch die

```

(1) LU-Zerlegung
  For k = 0 To 2N - 2
    For i = k + 1 To min(k + o, 2N - 1)
       $a_{i,k}^{(m)} = a_{i,k}^{(m)} / a_{k,k}^{(m)}$ 
      For j = k + 1 To min(k + o, 2N - 1)
         $a_{i,j}^{(m)} = a_{i,j}^{(m)} - a_{i,k}^{(m)} a_{k,j}^{(m)}$ 
      End
    End
  End
End

```

Abbildung 4.5: Allgemein gehaltener Algorithmus zur schnellen Realisierung der LU-Zerlegung als Vorverarbeitungsschritt des Klappoperators für fast beliebige Überlappungsfaktoren u , $m = 0, 1, \dots, B - 1$ mit $B = \text{Blocklänge}$, $L = 2NB = \text{Signallänge}$ (vgl. LU-Zerlegung von Bandmatrizen u.a. in [Kan05, MNW04]).

¹Da in den Untersuchungen zur Anwendung von überlappenden lokalen trigonometrischen Basen in der Bildatenkompression mit der LCT-6 im Vergleich zur LCT-4 keine besseren Rekonstruktionsergebnisse erzielt werden konnten, wird sich bei der LCT-6 auf den allgemeinen schnellen Algorithmus beschränkt. Dieser benötigt für die Klappoperation zwei arithmetische Operationen pro Signalwert mehr als ein speziell an die LCT-6 angepasster Algorithmus.

```

(2) Vorwärtseinsetzen
  For  $j = 0$  To  $2N - 2$ 
    For  $i = j + 1$  To  $\min(j + o, 2N - 1)$ 
       $f_i^{(m)} = f_i^{(m)} - a_{i,j}^{(m)} f_j^{(m)}$ 
    End
  End

(3) Rückwärtseinsetzen
   $f_{2N-1}^{(m)} = f_{2N-1}^{(m)} / a_{2N-1,2N-1}^{(m)}$ 
  For  $i = 2N - 2$  To  $0$ 
    For  $j = i + 1$  To  $\min(i + o, 2N - 1)$ 
       $f_i^{(m)} = f_i^{(m)} - a_{i,j}^{(m)} f_j^{(m)}$ 
    End
     $f_i^{(m)} = f_i^{(m)} / a_{i,i}^{(m)}$ 
  End

```

Abbildung 4.6: Allgemein gehaltener Algorithmus zur schnellen Realisierung des Klappoperators durch Vorwärts- und Rückwärtseinsetzen für fast beliebige Überlappungsfaktoren u , $m = 0, 1, \dots, B - 1$ mit $B = \text{Blocklänge}$, $L = 2NB = \text{Signallänge}$ (vgl. LU-Zerlegung von Bandmatrizen u.a. in [Kan05, MNW04]). Das geklappte Signal $f_{\mathcal{T}}$ wird inplace in f gespeichert.

Ausnutzung der in diesem und in Abschnitt 4.2 aufgezeigten Eigenschaften verwenden sie im Vergleich zu den allgemein gehaltenen schnellen Algorithmen (vgl. Abbildungen 4.2, 4.5 und 4.6) eine kompaktere Speicherung der Nicht-Nullkoeffizienten und benötigen bei geradzahli- gen Überlappungsfaktoren weniger Rechenoperationen.

Tabelle 4.1 fasst die Gleichungen zur Bestimmung des Berechnungsaufwandes für die allgemein gehaltenen und angepassten Algorithmen zur schnellen Realisierung der Klapp- und Aufklappoperationen sowie der LU-Zerlegung zusammen. Für mehrfach überlappende Fenster ist die Bestimmung des geklappten Signals unter Verwendung des Matrix-Vektor-Produkts nach Gleichung (4.21), das $O_{\mathcal{T}} = 2\frac{L}{B} - 1$ arithmetische Operationen pro Signalwert benötigt, mit einem sehr hohen Berechnungsaufwand verbunden. Wie anhand von Tabelle 4.1 zu erkennen ist, kommen hingegen die schnellen Realisierungen der Klappoperatoren mit der gleichen geringen Anzahl arithmetischer Operationen aus wie die korrespondierenden Realisierungen der Aufklappoperatoren. Ein zusätzlicher Berechnungsaufwand entsteht lediglich durch die LU-Faktorisierung. Bei der Verarbeitung von mehrdimensionalen Signalen reduziert sich jedoch der relative Berechnungsaufwand erheblich, da die LU-Zerlegung für jeden Parameter m nur einmal pro Dimensionsrichtung durchgeführt werden muss.

Betrachtet man z.B. ein Bildsignal der Größe von 256 Bildzeilen und 512 Bildspalten, so benötigt die LU-Zerlegung bei der Dreifachüberlappung unter Verwendung einer Blockgröße von $B = 8$ zur Verarbeitung einer Bildzeilenlänge von 512 Bildpunkten 2.953125 arithmetische Operationen pro Signalwert. Durch die Transformation von 256 Bildzeilen verringert sich jedoch der relative Berechnungsaufwand um den Faktor 256 auf 0.0115 arithmetische

Überlappungsfaktor		Signalanalyse		Signalrekonstruktion
		LU-Zerlegung	Klappoperation	Aufklappoperation
$u = 2$	allg. Algo.	$O_{LU} = 3 - 3\frac{B}{L}$	$O_{\mathcal{T}} = 5 - 4\frac{B}{L}$	$O_{\mathcal{U}} = 5 - 4\frac{B}{L}$
	angep. Algo.	keine	$O_{\mathcal{T}} = 3 - 2\frac{B}{L}$	$O_{\mathcal{U}} = 3 - 2\frac{B}{L}$
$u = 3$	allg./angep. Algo.	$O_{LU} = 3 - 3\frac{B}{L}$	$O_{\mathcal{T}} = 5 - 4\frac{B}{L}$	$O_{\mathcal{U}} = 5 - 4\frac{B}{L}$
$u = 4$	allg. Algo.	$O_{LU} = 10 - 17\frac{B}{L}$	$O_{\mathcal{T}} = 9 - 12\frac{B}{L}$	$O_{\mathcal{U}} = 9 - 12\frac{B}{L}$
	angep. Algo.	$O_{LU} = 4 + 5\frac{B}{L}$	$O_{\mathcal{T}} = 7 - 8\frac{B}{L}$	$O_{\mathcal{U}} = 7 - 8\frac{B}{L}$
$u = 6$	allg. Algo.	$O_{LU} = 21 - 50\frac{B}{L}$	$O_{\mathcal{T}} = 13 - 24\frac{B}{L}$	$O_{\mathcal{U}} = 13 - 24\frac{B}{L}$

Tabelle 4.1: Anzahl der arithmetischen Operationen pro Signalwert für die allgemein gehaltenen schnellen Algorithmen (vgl. Abbildungen 4.2, 4.5 und 4.6) und für die angepassten schnellen Algorithmen (vgl. Anhang E) der Klapp- und Aufklappoperatoren, B = Blockgröße, $L = 2NB$ = Signallänge in Abtastwerten.

Operationen pro Signalwert. Die LU-Zerlegung für eine Bildspaltenlänge von 256 Bildpunkten benötigt 2.90625 arithmetische Operationen pro Signalwert. Durch die Transformation von 512 Bildspalten reduziert sich der relative Berechnungsaufwand auf 0.0057 Operationen pro Signalwert. Somit ergibt sich bei der Dreifachüberlappung für die Klappung eines zweidimensionalen Signals der Größe 256×512 durch die LU-Zerlegung ein zusätzlicher relativer Berechnungsaufwand von lediglich 0.0172 arithmetischen Operationen pro Signalwert.

Für zweidimensionale Signale der Größe $L_v \times L_h$ mit $L_v = 2N_v B$ und $L_h = 2N_h B$ lässt sich die Anzahl der arithmetischen Operationen allgemein durch die Gleichung

$$O_{LU} = \frac{O_{LU}(L_v)}{L_h} + \frac{O_{LU}(L_h)}{L_v} \quad (4.24)$$

beschreiben.

4.3.2 Algorithmen auf Grundlage einer Mini-LU-Zerlegung

Die im vorangegangenen Abschnitt vorgestellten Algorithmen zur schnellen Realisierung des Klappoperators führen eine LU-Faktorisierung der in Gleichung (4.16) dargestellten Koeffizientenmatrix $\mathbf{T}_u^{(m)}$ des Aufklappoperators durch. Diese Faktorisierung bewirkt im Vergleich zur schnellen Realisierung des Aufklappoperators einen zusätzlichen Berechnungsaufwand. Wie nachfolgend am Beispiel der Dreifachüberlappung gezeigt wird, lässt sich unter der Bedingung, dass den inneren Signalintervallen Translate einer Fensterfunktion w zugeordnet werden, die Anzahl der arithmetischen Operationen für die LU-Zerlegung im Rahmen einer definierten Zahlengenauigkeit signifikant reduzieren. Es sei darauf hingewiesen, dass der in diesem Abschnitt vorgestellte Ansatz insbesondere für die Transformation von eindimensionalen

	$u_{0,0}^{(3)} = 0.514543 \dots$	$u_{0,1}^{(3)} = 0.487645 \dots$
$l_{1,0}^{(3)} = -0.248954 \dots$	$u_{1,1}^{(3)} = 0.921222 \dots$	$u_{1,2}^{(3)} = 0.386445 \dots$
$l_{2,1}^{(3)} = -0.209910 \dots$	$u_{2,2}^{(3)} = 0.913058 \dots$	$u_{2,3}^{(3)} = 0.294271 \dots$
$l_{3,2}^{(3)} = -0.140295 \dots$	$u_{3,3}^{(3)} = 0.841106 \dots$	$u_{3,4}^{(3)} = 0.386445 \dots$
$l_{4,3}^{(3)} = -0.229904 \dots$	$u_{4,4}^{(3)} = 0.920785 \dots$	$u_{4,5}^{(3)} = 0.294271 \dots$
$l_{5,4}^{(3)} = -0.139118 \dots$	$u_{5,5}^{(3)} = 0.840759 \dots$	$u_{5,6}^{(3)} = 0.386445 \dots$
<hr/>		
$l_{6,5}^{(3)} = -0.229999 \dots$	$u_{6,6}^{(3)} = 0.920822 \dots$	$u_{6,7}^{(3)} = 0.294271 \dots$
$l_{7,6}^{(3)} = -0.139112 \dots$	$u_{7,7}^{(3)} = 0.840758 \dots$	$u_{7,8}^{(3)} = 0.386445 \dots$
$l_{8,7}^{(3)} = -0.229999 \dots$	$u_{8,8}^{(3)} = 0.920822 \dots$	$u_{8,9}^{(3)} = 0.294271 \dots$
$l_{9,8}^{(3)} = -0.139112 \dots$	$u_{9,9}^{(3)} = 0.840758 \dots$	$u_{9,10}^{(3)} = 0.386445 \dots$
$l_{10,9}^{(3)} = -0.229999 \dots$	$u_{10,10}^{(3)} = 0.920822 \dots$	$u_{10,11}^{(3)} = 0.294271 \dots$
\vdots	\vdots	\vdots
$l_{2N-3,2N-4}^{(3)} = -0.139112 \dots$	$u_{2N-3,2N-3}^{(3)} = 0.840758 \dots$	$u_{2N-3,2N-2}^{(3)} = 0.386445 \dots$
<hr/>		
$l_{2N-2,2N-3}^{(3)} = -0.229999 \dots$	$u_{2N-2,2N-2}^{(3)} = 0.920822 \dots$	$u_{2N-2,2N-1}^{(3)} = 0.264010 \dots$
$l_{2N-1,2N-2}^{(3)} = -0.558787 \dots$	$u_{2N-1,2N-1}^{(3)} = 1.710025 \dots$	

Tabelle 4.2: Matrixkoeffizienten der LU-Zerlegung für den Parameter $m = 3$ am Beispiel der Dreifachüberlappung unter Verwendung der Randfenster aus Abschnitt 3.5.3 und der Fensterfunktion $w^{(3)}$ aus Abschnitt 2.3.2.

Signalen, beispielsweise Audio, von Interesse ist, da bei der Verarbeitung von mehrdimensionalen Signalen der relative Berechnungsaufwand bereits sehr gering ausfällt (vgl. Abschnitt 4.3.1).

In Tabelle 4.2 sind unter Verwendung des dreifach überlappenden Fensters $w^{(3)}$ aus Abschnitt 2.3.2 sowie der Randfenster aus Abschnitt 3.5.3 für den Parameter $m = 3$ die berechneten Koeffizienten $l_{i+1,i}^{(3)}$, $u_{i,i}^{(3)}$ und $u_{i,i+1}^{(3)}$ des Matrixbandes der Matrizen $\mathbf{L}^{(3)}$ und $\mathbf{U}^{(3)}$ dargestellt. Wie zu erkennen ist, ergeben sich bei diesem Berechnungsbeispiel für die inneren Intervalle unter der Annahme einer gewählten Zahlengenauigkeit von sechs Nachkommastellen ab dem Index $i = 7$ betragsmäßig keine neuen Koeffizienten. Somit konvergieren die LU-Koeffizienten – und zwar sehr schnell. Lediglich am rechten Signalrand erhält man durch die Randfenster neue Koeffizienten. Bei der Entwicklung von schnellen Algorithmen lässt sich daher die Anzahl der Rechenoperationen reduzieren, indem die im Rahmen einer geforderten Zahlengenauigkeit konvergierten Koeffizienten nicht wiederholt neu berechnet werden.

Während die bisherigen Algorithmen zur Realisierung der Klappoperatoren auf Koeffizientenmatrizen $\mathbf{T}_u^{(m)}$ basieren, die in Abhängigkeit von der Signallänge von $L = 2NB$ Abtastwerten eine Größe von $2N \times 2N$ aufweisen, wird im Folgenden eine um die konvergierten Bereiche reduzierte und von der realen Signallänge unabhängige Matrix der Größe $2K \times 2K$ mit $K < N$

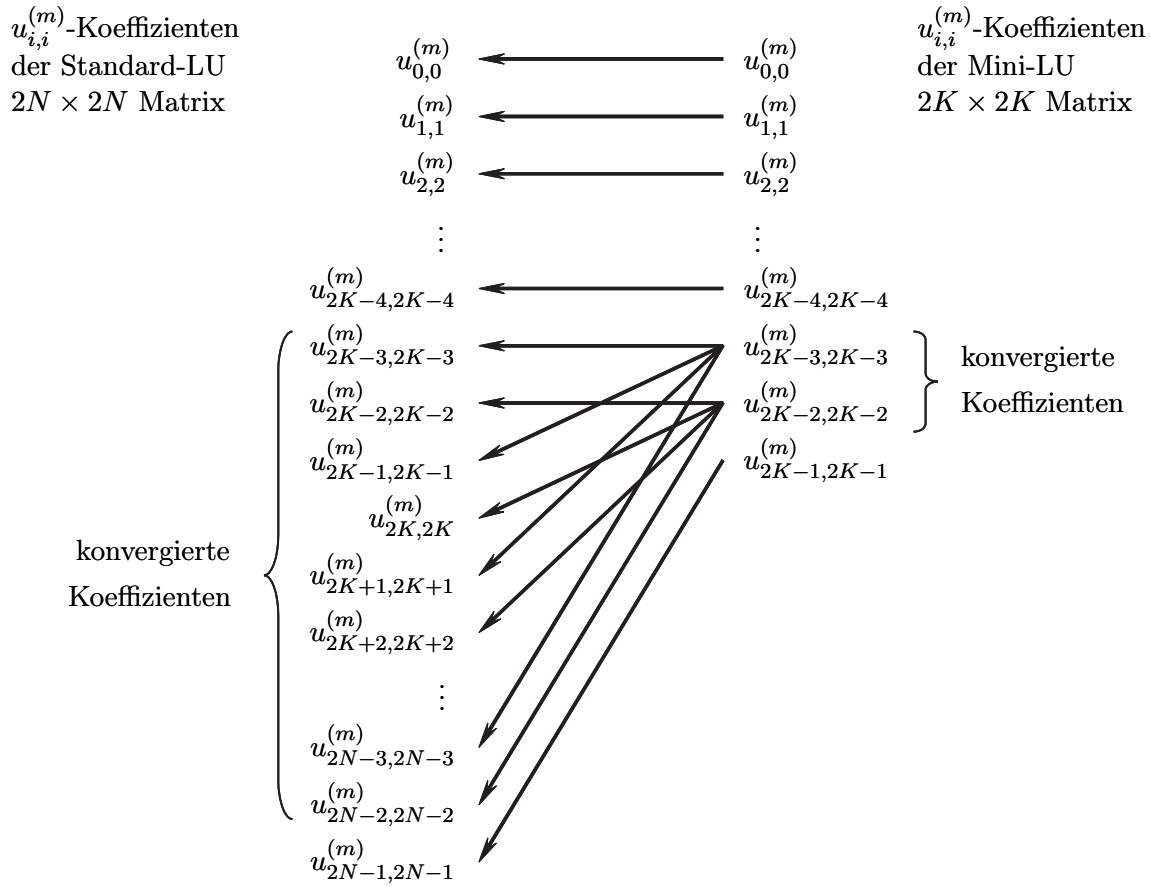


Abbildung 4.7: Koeffizienten-Mapping zur Realisierung einer minimierten LU-Zerlegung am Beispiel der $u_{i,i}^{(m)}$ -Matrixkoeffizienten.

eingeführt. Für die LU-Zerlegung, die nicht mit Signalwerten operiert, sei also angenommen, dass das zu transformierende Signal nur eine minimierte Länge von $L = 2KB$ Abtastwerten aufweist. Das Verfahren soll daher als Mini-LU bezeichnet werden. Die zu dieser Signallänge korrespondierende $2K \times 2K$ Transformationsmatrix $\mathbf{T}_u^{(m)}$ wird entsprechend den Betrachtungen im vorangegangenen Abschnitt mittels LU-Zerlegung in eine linke untere Dreiecksmatrix und eine rechte obere Dreiecksmatrix faktorisiert. Beim sich anschließenden Klappvorgang eines Signals der realen Länge von $2NB$ Abtastwerten wird dann während des Vorwärts- und des Rückwärtseinsetzens wiederholt auf die konvergierten Koeffizienten zugegriffen. Dieses im Folgenden als Koeffizienten-Mapping bezeichnete Verfahren ist in Abbildung 4.7 am Beispiel der $u_{i,i}^{(m)}$ -Matrixkoeffizienten dargestellt. Abbildung 4.8 zeigt den auf Grundlage der Mini-LU und des Koeffizienten-Mappings entwickelten Algorithmus zur schnellen Realisierung des Klappoperators am Beispiel der Dreifachüberlappung.

Praktischen Untersuchungen für die im Anhang H dargestellten Testbilder zeigten, dass z.B. bei der Dreifachüberlappung die Transformation auf Basis der Mini-LU-Zerlegung und die anschließende Rücktransformation mit Runden auf Integerwerte bereits bei $K = 6$ eine perfekte Rekonstruktion lieferte. Auch nach der verlustbehafteten Kompression mit den im Rahmen dieser Arbeit untersuchten Kompressionsverfahren stimmten die rekonstruierten Bilder

Koeffizienten der Mini-LU	Klappoperation
$\vec{a}_{-1}^{(m)} = (a_{-1,0}^{(m)} \quad a_{-1,1}^{(m)} \quad \dots \quad a_{-1,2K-2}^{(m)})^T$ $\vec{a}_0^{(m)} = (a_{0,0}^{(m)} \quad a_{0,1}^{(m)} \quad \dots \quad a_{0,2K-1}^{(m)})^T$ $\vec{a}_1^{(m)} = (a_{1,0}^{(m)} \quad a_{1,1}^{(m)} \quad a_{1,2}^{(m)} \quad a_{1,3}^{(m)})^T$	(2) Vorwärtseinsetzen
<i>Initialisierung</i>	$f_1^{(m)} = f_1^{(m)} - a_{-1,0}^{(m)} f_0^{(m)}$
$a_{-1,2K-4}^{(m)} = -w[2B - m - 1]$ $a_{-1,2K-3}^{(m)} = -w[B + m]$ $a_{-1,2K-2}^{(m)} = -w_{r1}[2B - m - 1]$ $a_{0,0}^{(m)} = w_{l0}[m]$ $a_{0,2K-3}^{(m)} = w[B - m - 1]$ $a_{0,2K-2}^{(m)} = w[m]$ $a_{0,2K-1}^{(m)} = w_{r0}[B - m - 1]$ $a_{1,0}^{(m)} = w_{l1}[-B + m]$ $a_{1,1}^{(m)} = w[-m - 1]$ $a_{1,2}^{(m)} = w[-B + m]$ $a_{1,3}^{(m)} = w_{r0}[-B + m]$	For $n = 1$ To $K - 2$
	$f_{2n}^{(m)} = f_{2n}^{(m)} - a_{-1,2n-1}^{(m)} f_{2n-1}^{(m)}$ $f_{2n+1}^{(m)} = f_{2n+1}^{(m)} - a_{-1,2n}^{(m)} f_{2n}^{(m)}$
	End
	For $n = K - 1$ To $N - 2$
	$f_{2n}^{(m)} = f_{2n}^{(m)} - a_{-1,2K-3}^{(m)} f_{2n-1}^{(m)}$ $f_{2n+1}^{(m)} = f_{2n+1}^{(m)} - a_{-1,2K-4}^{(m)} f_{2n}^{(m)}$
	End
	$f_{2N-2}^{(m)} = f_{2N-2}^{(m)} - a_{-1,2K-3}^{(m)} f_{2N-3}^{(m)}$ $f_{2N-1}^{(m)} = f_{2N-1}^{(m)} - a_{-1,2K-2}^{(m)} f_{2N-2}^{(m)}$
(1) <i>LU-Zerlegung</i>	(3) Rückwärtseinsetzen
$a_{-1,0}^{(m)} = a_{-1,2K-4}^{(m)} / a_{0,0}^{(m)}$ $a_{0,1}^{(m)} = a_{0,2K-3}^{(m)} - a_{-1,0}^{(m)} a_{1,0}^{(m)}$	$f_{2N-1}^{(m)} = f_{2N-1}^{(m)} / a_{0,2K-1}^{(m)}$ $f_{2N-2}^{(m)} = \frac{f_{2N-2}^{(m)} - a_{1,3}^{(m)} f_{2N-1}^{(m)}}{a_{0,2K-2}^{(m)}}$
For $n = 1$ To $K - 2$	For $n = N - 2$ (-1) $K - 1$
$a_{-1,2n-1}^{(m)} = a_{-1,2K-3}^{(m)} / a_{0,2n-1}^{(m)}$ $a_{0,2n}^{(m)} = a_{0,2K-2}^{(m)} - a_{-1,2n-1}^{(m)} a_{1,1}^{(m)}$ $a_{-1,2n}^{(m)} = a_{-1,2K-4}^{(m)} / a_{0,2n}^{(m)}$ $a_{0,2n+1}^{(m)} = a_{0,2K-3}^{(m)} - a_{-1,2n}^{(m)} a_{1,2}^{(m)}$	$f_{2n+1}^{(m)} = \frac{f_{2n+1}^{(m)} - a_{1,1}^{(m)} f_{2n+2}^{(m)}}{a_{0,2K-3}^{(m)}}$ $f_{2n}^{(m)} = \frac{f_{2n}^{(m)} - a_{1,2}^{(m)} f_{2n+1}^{(m)}}{a_{0,2K-2}^{(m)}}$
End	End
$a_{-1,2K-3}^{(m)} = a_{-1,2K-3}^{(m)} / a_{0,2K-3}^{(m)}$ $a_{0,2K-2}^{(m)} = a_{0,2K-2}^{(m)} - a_{-1,2K-3}^{(m)} a_{1,1}^{(m)}$ $a_{-1,2K-2}^{(m)} = a_{-1,2K-2}^{(m)} / a_{0,2K-2}^{(m)}$ $a_{0,2K-1}^{(m)} = a_{0,2K-1}^{(m)} - a_{-1,2K-2}^{(m)} a_{1,3}^{(m)}$	For $n = K - 2$ (-1) 1
	$f_{2n+1}^{(m)} = \frac{f_{2n+1}^{(m)} - a_{1,1}^{(m)} f_{2n+2}^{(m)}}{a_{0,2n+1}^{(m)}}$ $f_{2n}^{(m)} = \frac{f_{2n}^{(m)} - a_{1,2}^{(m)} f_{2n+1}^{(m)}}{a_{0,2n}^{(m)}}$
	End
	$f_1^{(m)} = (f_1^{(m)} - a_{1,1}^{(m)} f_2^{(m)}) / a_{0,1}^{(m)}$ $f_0^{(m)} = (f_0^{(m)} - a_{1,0}^{(m)} f_1^{(m)}) / a_{0,0}^{(m)}$

Abbildung 4.8: Mini-LU: Schnelle Realisierung der Mini-LU und schnelle Realisierung der Klappoperation durch Vorwärts- und Rückwärtseinsetzen am Beispiel dreifach überlappender Fenster, $m = 0, 1, \dots, B - 1$; $B = \text{Blocklänge}$, $L = 2NB = \text{Signallänge}$. Das geklappte Signal f_T wird inplace in f gespeichert.

	1×512			256×512		
	Operationen	$\frac{\text{Operationen}}{\text{Koeffizient}}$	Einsparung	Operationen	$\frac{\text{Operationen}}{\text{Koeffizient}}$	Einsparung
LU	1512	2.953125		2256	0.017212	
Mini-LU	264	0.515625	82.5 %	264	0.002014	88.3 %

Tabelle 4.3: Vergleich der Algorithmen zur LU-Zerlegung für dreifach überlappende Fenster am Beispiel eines eindimensionalen Signals und eines zweidimensionalen Signals; Anzahl der arithmetischen Operationen zur Bestimmung der LU-Koeffizienten für die nachfolgende Realisierung des Klappoperators durch Vorwärts- und Rückwärtseinsetzen, Blockgröße $B = 8$.

beim Vergleich zwischen der Verwendung der Standard-LU-Zerlegung und der Verwendung der Mini-LU-Zerlegung bei $K = 6$ überein. Für $K = 6$ benötigt die Mini-LU-Zerlegung unter der Verwendung von dreifach überlappenden Fensterfunktionen für jeden Parameter $m = 0, 1, \dots, B - 1$ nur 33 arithmetische Operationen. Bei einer Blockgröße von $B = 8$ ergeben sich somit insgesamt 264 Operationen – und zwar unabhängig von der Anzahl der Dimensionen sowie der Größe bzw. der Länge des zu verarbeitenden Signals. Durch die Verwendung der Mini-LU reduziert sich bei der Dreifachüberlappung der relative Berechnungsaufwand für die LU-Zerlegung auf

$$O_{\text{Mini-LU}} = (6K - 3) \frac{B}{L} \quad (4.25)$$

arithmetische Operationen pro Signalwert. In Tabelle 4.3 werden anhand von zwei Signalbeispielen die vorgeschlagenen schnellen Algorithmen miteinander verglichen. Aufgrund der signifikanten Reduktion des relativen Berechnungsaufwandes pro Signalwert lassen sich durch die Einführung der Mini-LU-Zerlegung nun auch eindimensionale Signale sehr effizient verarbeiten.

Da die Mini-LU-Zerlegung unabhängig von der eigentlichen Signallänge ist und sich die vergleichsweise kleinen bandstrukturierten Koeffizientenmatrizen $\mathbf{L}^{(m)}$ und $\mathbf{U}^{(m)}$ sehr kompakt speichern lassen, bietet sich die Realisierung einer vorberechneten LU-Zerlegung an. Anstelle der Fensterfunktionen können die einmalig zu ermittelnden Koeffizienten der Mini-LU-Zerlegung vorberechnet im Encoder implementiert werden, so dass im Encoder die LU-Zerlegung entfällt.

Analog zu den Herleitungen in diesem Abschnitt lässt sich die Mini-LU-Zerlegung in Kompaktsteichertechnik und die vorberechnete Mini-LU-Koeffizientenmatrix auch für größere Überlappungsfaktoren herleiten. Im Rahmen der Untersuchungen für diese Arbeit wurde so auch die Mini-LU für die Vierfachüberlappung realisiert und verifiziert.

4.4 Leistungsvergleich der Algorithmen

In diesem Kapitel wurden Algorithmen zur schnellen Realisierung der überlappenden Kosinustransformation entwickelt und analysiert. Tabelle 4.4 vergleicht für ausgewählte Überlappungsfaktoren und Blockgrößen den Berechnungsaufwand, der bei der direkten und der schnellen Realisierung der überlappenden Kosinustransformation anfällt. Der Berechnungsaufwand für die schnelle überlappende Kosinustransformation ergibt sich aus der Anzahl der arithmetischen Operationen für den Klapp- bzw. Aufklappvorgang zuzüglich der Anzahl der

arithmetischen Operationen für die schnelle trigonometrische Blocktransformation. Zur Bestimmung der Anzahl arithmetischer Operationen für den Klapp- bzw. Aufklappvorgang bei einem gegebenen Überlappungsfaktor u wird in Tabelle 4.4 vereinfachend¹ angenommen, dass die schnellen Realisierungen der Klapp- bzw. Aufklappoperatoren u Punktoperationen (Multiplikationen und Divisionen) und $u - 1$ Strichoperationen (Additionen und Subtraktionen) pro Signalwert benötigen. Betrachtet man z.B. die schnelle zweifach überlappende Kosinustransformation 16×32 LCT-2, für die das Signal in Intervalle mit einer Länge von 16 Abtastwerten eingeteilt wird, so ergibt sich folgender Berechnungsaufwand: Der Klappvorgang benötigt bei der Zweifachüberlappung zwei Punktoperationen und eine Strichoperation pro Signalwert. Bezogen auf ein Intervall von 16 Abtastwerten ergeben sich folglich 32 Punktoperationen und 16 Strichoperationen pro Intervall. Die sich anschließende Verarbeitung des in das Intervall geklappten Signals mit der schnellen 16×16 DCT-IV nach [Mal92] benötigt 48 Multiplikationen und 96 Additionen pro Intervall (vgl. Tabelle 4.4), so dass man einen Berechnungsaufwand von insgesamt 80 Punktoperationen und 112 Strichoperationen pro Intervall erhält. Demzufolge ergibt sich für die schnelle 16×32 LCT-2 ein durchschnittlicher Berechnungsaufwand von 12 arithmetischen Operationen pro Transformationskoeffizient. Im Vergleich zur direkten Realisierung der 16×32 LCT-2, die 63 Operationen pro Transformationskoeffizient benötigt, reduziert sich der Berechnungsaufwand um 80.95 %.

Anhand von Tabelle 4.4 ist zu erkennen, dass sich für alle untersuchten Überlappungsfaktoren der Berechnungsaufwand signifikant reduziert, wenn die überlappenden Kosinustransformationen mit schnellen Klapp- bzw. Aufklappoperatoren und schnellen trigonometrischen Blocktransformationen realisiert werden. Im Vergleich mit den schnellen Algorithmen der klassischen DCT-II/III, wie sie z.B. im JPEG-Kompressionsverfahren zur Anwendung kommen, benötigen die schnellen Algorithmen der überlappenden Kosinustransformationen mehr arithmetische Operationen. Zum einen ist im Verhältnis zur schnellen DCT-II/III die schnelle DCT-IV des geklappten Signals mit einem höheren Berechnungsaufwand verbunden, zum anderen entstehen zusätzliche Rechenoperationen durch den Klapp- und Aufklappvorgang.

Da in den nachfolgenden Kapiteln ein Vergleich zwischen LCT-basierten und waveletbasierten Kompressionsstrategien erfolgt, werden in Tabelle 4.4 auch die arithmetischen Operationen für die schnelle Realisierung in Lifting-Struktur des verwendeten 9/7-DWT [DS98, LCC01] angegeben. Es ist zu erkennen, dass die schnellen Algorithmen für die 16×32 LCT-2, 8×24 LCT-3, 8×32 LCT-4 und 9/7-DWT mit 12 – 14 arithmetischen Operationen pro Koeffizienten einen vergleichbaren Berechnungsaufwand benötigen. Dieses Resultat ist insofern von Interesse, da die zu erwartende Verbesserung der Kompressionsleistung durch die Verwendung der überlappenden Kosinustransformationen in Bezug auf die Wavelet-Transformation nicht mit einer wesentlichen Erhöhung des Berechnungsaufwandes verbunden ist.

Zusammenfassend lässt sich feststellen, dass die in diesem Kapitel für die LCT-u vorgeschlagenen Algorithmen eine schnelle Transformation von Signalen ermöglichen. Da sich die Algorithmen sehr effizient implementieren lassen, eignen sich die untersuchten überlappenden Kosinustransformationen auch für Applikationen, denen nur geringe Ressourcen zur Verfügung stehen.

Der Schwerpunkt des nächsten Kapitels besteht in der Untersuchung der Leistungsfähigkeit von Bildkompressionsverfahren, die zur Darstellung der Bildsignale die neu eingeführten überlappenden lokalen trigonometrischen Basen auf dem Intervall verwenden.

¹Die geringfügige Reduktion des Berechnungsaufwandes durch die Randbasisfunktionen bzw. durch die Randfenster wird nicht berücksichtigt. Sie wird durch die in Tabelle 4.1 angegebenen $\frac{B}{L}$ -Subtrahenden und die Differenz zwischen dem Berechnungsaufwand für die DCT-II und DCT-IV am Signalende quantifiziert. Der Berechnungsaufwand für die LU-Zerlegung wird ebenfalls nicht berücksichtigt, da er, wie in Abschnitt 4.3.1 erläutert, bei der Transformation von Bildern relativ gering ausfällt (vgl. auch Tabelle 4.3).

Transformation	Mul.+ Div. Intervall	Add.+Sub. Intervall	Operationen Koeffizient	Einsparung
Keine Überlappung				
8×8 DCT, direkt	64	56	15	
8×8 DCT-II/III [Mal92]	13	29	5.25	65 %
8×8 DCT-IV [Mal92]	20	36	7	53 %
16×16 DCT, direkt	256	240	31	
16×16 DCT-II/III [Mal92]	33	81	7.125	77.02 %
16×16 DCT-IV [Mal92]	48	96	9	70.97 %
32×32 DCT, direkt	1024	992	63	
32×32 DCT-II/III [Mal92]	81	209	9.0625	85.62 %
Zweifachüberlappung				
8×16 LCT-2, direkt	128	120	31	
8×16 LCT-2, schnell	36	44	10	67.74 %
16×32 LCT-2, direkt	512	496	63	
16×32 LCT-2, schnell	80	112	12	80.95 %
Dreifachüberlappung				
8×24 LCT-3, direkt	192	184	47	
8×24 LCT-3, schnell	44	52	12	74.47 %
Vierfachüberlappung				
8×32 LCT-4, direkt	256	248	63	
8×32 LCT-4, schnell	52	60	14	77.78 %
Sechsfachüberlappung				
8×48 LCT-6, direkt	384	376	95	
8×48 LCT-6, schnell	68	76	18	81.05 %
Wavelet-Transformation				
FLWT, 3-level, 8 Pixel	42	56	12.25	
FLWT, 4-level, 16 Pixel	90	120	13.125	
FLWT, 5-level, 32 Pixel	186	248	13.5625	
FLWT, 6-level, 64 Pixel	378	504	13.78125	

Tabelle 4.4: Berechnungsaufwand für die Realisierung der überlappenden Kosinustransformationen ohne Klapp- bzw. Aufklappoperatoren (direkt) und für die Realisierung der überlappenden Kosinustransformationen unter Verwendung von Klapp- bzw. Aufklappoperatoren und schnellen trigonometrischen Blocktransformationen (schnell) sowie Vergleich mit der schnellen 9/7-DWT (FLWT¹).

¹FLWT: fast lifting wavelet transform

Kapitel 5

Lokale trigonometrische Basen in der Bilddatenkompression

Dieses Kapitel untersucht die Anwendung von überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression. Im ersten Abschnitt des Kapitels wird die überlappende Kosinustransformation von Bilddaten, welche die im Rahmen dieser Arbeit neu entwickelten Klappoperatoren für Kosinus-IV-Kosinus-II-Basen auf dem Intervall verwendet, erläutert. Anschließend werden verschiedene Ansätze zur Codierung der LCT-Transformationskoeffizienten untersucht. Des Weiteren werden in diesem Kapitel zwei Matlab-Toolboxen vorgestellt, die zur Demonstration der Leistungsfähigkeit der LCT-basierten Kompressionsstrategien entwickelt wurden.

Die Einführung von überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression wurde u.a. mit dem Ziel verfolgt, zur Reduktion von Blockartefakten die DCT in bestehenden Kompressionsverfahren wie z.B. dem JPEG-Codec durch eine geeignete überlappende trigonometrische Transformation zu ersetzen. Diese Möglichkeit besteht, da die überlappenden trigonometrischen Transformationen ebenso wie die klassische Kosinustransformation zu blockweise angeordneten Transformationskoeffizienten führen und die Koeffizientenblöcke der Größe $B \times B$ eine vergleichbare Teilbandstruktur aufweisen wie die Transformationskoeffizienten der DCT. Die weiteren Verarbeitungsstufen wie Quantisierung und Entropiecodierung können somit aus dem JPEG-Verfahren übernommen werden. In vorangegangenen Untersuchungen zum Einsatz von zweifach überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression [AACI93, AACI94, Mal00, Bit00b] wurde daher der JPEG-Algorithmus zur Codierung der quantisierten Transformationskoeffizienten verwendet. Dieser Ansatz eignet sich zwar für einen Vergleich der Effizienz verschiedener trigonometrischer Basisfunktionen untereinander, man erhält jedoch gegenüber aktuellen waveletbasierten Kompressionsverfahren keinen leistungsfähigen Transformationscodec. Die Rekonstruktionsqualität bei der JPEG-Codierung fällt auch unter Verwendung von überlappenden trigonometrischen Transformationen immer noch deutlich schlechter aus als bei den gegenwärtig in Leistungsanalysen am häufigsten verwendeten Wavelet-Codecs. Die schlechteren Kompressionsergebnisse liegen u.a. in der vergleichsweise ineffizienten Codierung der quantisierten Transformationskoeffizienten nach dem JPEG-Verfahren begründet. Die verbliebene Korrelation der Transformationskoeffizienten innerhalb eines Blocks wird nicht vollständig aufgelöst. Des Weiteren erfolgt mit Ausnahme der DC-Koeffizienten eine von den benachbarten Blöcken unabhängige Codierung der AC-Koeffizienten, obwohl diese ebenfalls miteinander korreliert sein können. Daher werden nachfolgend leistungsfähigere Verfahren zur Codierung der Transformationskoeffizienten untersucht, die

neben dem Vergleich der verschiedenen überlappenden lokalen trigonometrischen Basisfunktionen auch eine Einschätzung der Leistungsfähigkeit gegenüber waveletbasierten State-of-the-Art Kompressionsverfahren zulassen.

Mit der Entwicklung von effizienten Wavelet-Codern haben sich auch einige interessante Ansätze zur Codierung der Transformationskoeffizienten DCT-basierter Verfahren ergeben. Insbesondere konnte in verschiedenen Publikationen gezeigt werden, dass sich Codierungsstrategien, die ursprünglich für waveletbasierte Kompressionsverfahren entwickelt wurden, auch zur Codierung der Transformationskoeffizienten von Blocktransformationen eignen. Eine der ersten Arbeiten, die sich mit der Codierung von DCT-Koeffizienten auf Basis eines Wavelet-Codex beschäftigt, ist die Publikation von Xiong, Guleryuz und Orchard [XGO96]. Der von ihnen vorgestellte EZDCT¹-Codec basiert auf dem EZW²-Wavelet-Codec von Shapiro [Sha93]. Um den Wavelet-Codec nutzen zu können, werden die Transformationskoeffizienten der DCT in eine waveletartige Teilbandstruktur überführt. Abbildung 5.1 zeigt am Beispiel der diskreten Kosinustransformation das Prinzip der Umsortierung von Koeffizientenblöcken der Blockgröße $B = 2^s$ in eine Teilbandrepräsentation, die strukturmäßig äquivalent zu einer s -level Wavelet-Transformation ist. Der Ansatz, umsortierte DCT-Koeffizienten mit einem für Wavelets entwickelten Transformationscoder zu komprimieren, erscheint zunächst aufgrund der unterschiedlichen Transformationseigenschaften als wenig effizient. Die Vergleichsstudie von Xiong, Ramchandran, Orchard und Zhang [XROZ99] zeigt jedoch, dass mit diesem Codieransatz im Vergleich zur DCT-basierten JPEG-Komprimierung ein Codiergewinn erzielt werden kann. Nach der Entwicklung des SPIHT³-Wavelet-Codex durch Said und Pearlman [SP96b, PS08] wurde das Umsortierungsverfahren auch mit diesem Codec realisiert [TN98, TN99, TN00]. Die Ergebnisse dieser Arbeiten motivieren, in Abschnitt 5.2 für die Darstellung von Bildsignalen mit überlappenden lokalen trigonometrischen Basen die Komprimierung der Transformationskoeffizienten nach dem SPIHT-Algorithmus zu untersuchen. Insbesondere soll dieser Ansatz auch dazu dienen, mit Hilfe einer einheitlichen und etablierten Codierungsstrategie Aussagen über die Leistungsfähigkeit von überlappenden lokalen trigonometrischen Basen im Vergleich zur Leistungsfähigkeit des 9/7-Wavelets abzuleiten.

Um das Potential überlappender lokaler trigonometrischer Basen weiter auszuschöpfen, befassen sich die Abschnitte 5.3 und 5.4 mit Codierverfahren, die an die Eigenschaften der blockweise vorliegenden Transformationskoeffizienten trigonometrischer Transformationen angepasst sind. In Abschnitt 5.3 werden für überlappende lokale trigonometrische Basen Kompressionsstrategien auf Grundlage der CEB⁴-Codierung eingeführt und untersucht. Die Entwicklung der CEB-Codierung durch Tu und Tran [TT02b, TT02a] resultierte aus der Erkenntnis heraus, dass die Leistungsfähigkeit waveletbasierter Kompressionsverfahren, wie z.B. der ECECOW⁵-Codierung [Xia97] oder der EBCOT⁶-Codierung [Tau99, Tau00], die auch die Grundlage für den JPEG2000 Standard [JPE00, TM01, TOWS02] bildet, in hohem Maße das Ergebnis leistungsfähiger Kontextmodelle und adaptiver Entropiecodierung ist. Die CEB-Codierung realisiert eine kontextbasierte Entropiecodierung von Transformationskoeffizienten, die in Blöcken der Größe 8×8 vorliegen und das Resultat von Blocktransformationen oder überlappenden Blocktransformationen sind. Zur Codierung von quantisierten Koeffizientenblöcken ab der Größe 16×16 , die z.B. das Ergebnis einer 16×16 DCT oder einer 16×32 LCT-2 sind, werden in Abschnitt 5.4 Kompressionsstrategien unter Verwendung der kontextbasierten AGU-

¹EZDCT: embedded zerotree DCT

²EZW: embedded zerotree Wavelet

³SPIHT: set partitioning in hierarchical trees

⁴CEB: context-based entropy coding of block transform coefficients

⁵ECECOW: embedded conditional entropy coding of wavelet coefficients

⁶EBCOT: embedded block coding with optimized truncation

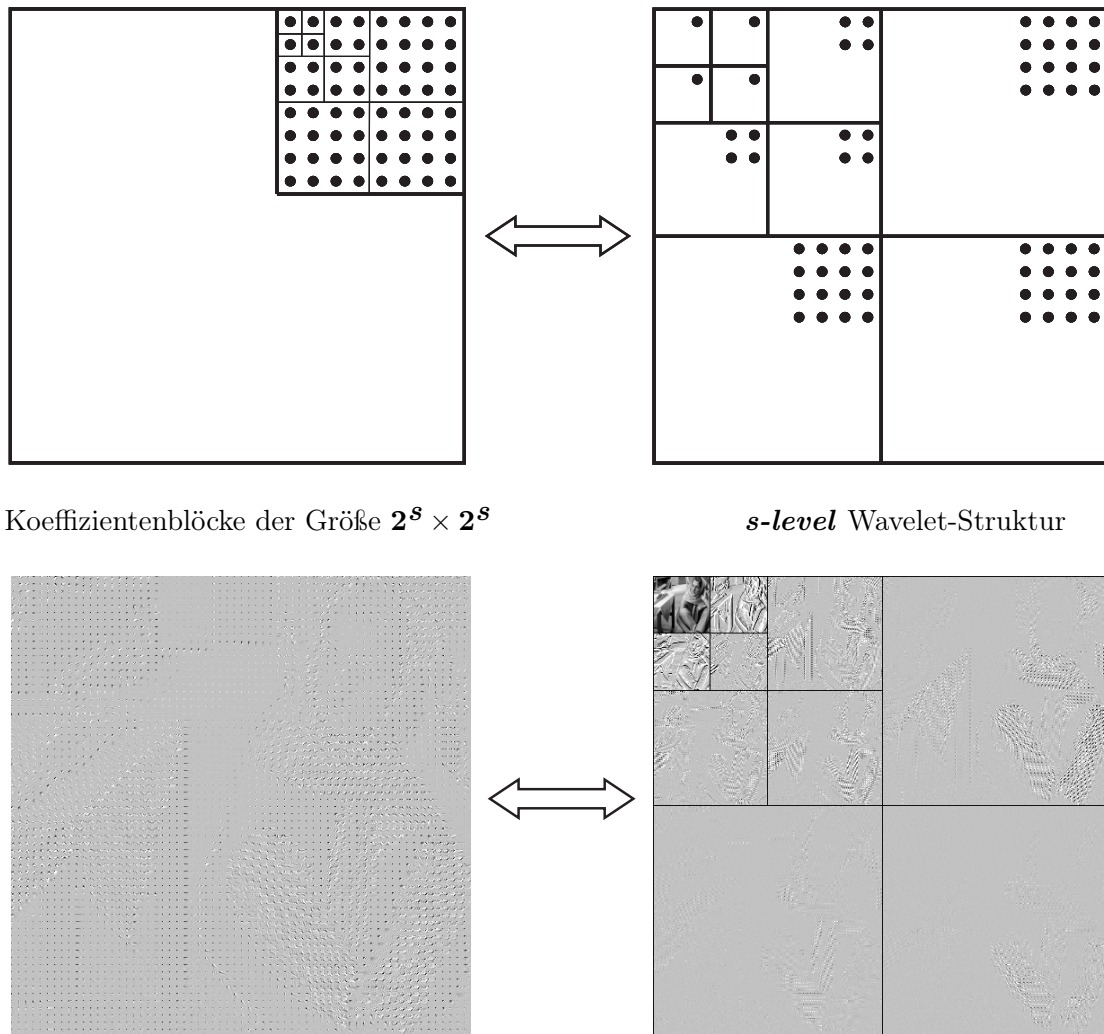
Koeffizientenblöcke der Größe $2^s \times 2^s$ *s-level* Wavelet-Struktur

Abbildung 5.1: Umsortierung von blockweise angeordneten DCT-Transformationskoeffizienten mit einer Blockgröße von $B \times B = 2^s \times 2^s$ in eine waveletartige *s-level* Teilbandstruktur am Beispiel von $s = 3$ (vgl. [XGO96, TN98, XROZ99]).

Codierung von Ponomarenko, Egiazarian, Lukin und Astola [PELA05] vorgestellt. Abschnitt 5.5 befasst sich hauptsächlich mit der Reduktion von Ringing-Artefakten. Es handelt sich hierbei um Kompressionsartefakte, die in den rekonstruierten Bildern in der Umgebung von Kanten auftreten und bei hohen Kompressionsfaktoren um so deutlicher sichtbar werden, je länger die Träger der verwendeten trigonometrischen Basisfunktionen sind. In Abschnitt 5.6 werden zwei Matlab-Toolboxen vorgestellt, die zur Veranschaulichung der Leistungsfähigkeit von überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression entwickelt und implementiert wurden.

5.1 Überlappende Kosinustransformation von Bilddaten

Die überlappende trigonometrische Transformation von mehrdimensionalen Signalen wird durch separiert ausgeführte eindimensionale Transformationen realisiert. Bei der Verarbeitung von Bilddaten erfolgt z.B. zuerst die Transformation aller Zeilen und darauf die Transforma-

tion aller Spalten oder umgekehrt. Zur effizienten Durchführung der überlappenden Kosinustransformation werden zunächst die überlappenden Signalanteile, die sich aus der Wichtung des Signals mit den Analysefenstern ergeben, spalten- und zeilenweise in den jeweils betrachteten Block geklappt. Für die Berechnung des geklappten Signals kommen die in Kapitel 4 entwickelten Algorithmen zur schnellen Realisierung des Klappoperators zum Einsatz. In Abbildung 5.2 ist z.B. das Resultat der Klappoperationen unter Verwendung von vierfach überlappenden Fensterfunktionen auf dem Intervall anhand des Bildes „Barbara“ veranschaulicht. Im Anschluß an die Klappoperationen wird das geklappte Bild mit den schnellen trigonometrischen Blocktransformationen weiterverarbeitet. Bis auf die gesonderte Behandlung des rechten Bildrandes und des unteren Bildrandes kommt in allen Intervallen die DCT-IV zur



Abbildung 5.2: Transformation von zweidimensionalen Signalen mit Klappoperatoren unter Verwendung von vierfach überlappenden Fensterfunktionen auf dem Intervall am Beispiel des Bildes „Barbara“, Blockgröße $B = 16$.

Anwendung. Mit Ausnahme des untersten rechten Blocks wird am rechten Bildrand in horizontaler Richtung die DCT-II und in vertikaler Richtung die DCT-IV verwendet, während hingegen am unteren Bildrand in vertikaler Richtung die DCT-II und in horizontaler Richtung die DCT-IV zum Einsatz kommt. Der rechte untere Block wird sowohl in horizontaler als auch in vertikaler Richtung mit der DCT-II transformiert.

Abbildung 5.3 zeigt am Beispiel des Bildes „Barbara“ die Transformationskoeffizienten der vorstehend beschriebenen DCT-IV/DCT-II des geklappten Bildsignals, wobei für die Klappoperationen vierfach überlappende Fenster genutzt wurden. Zum Vergleich sind die Trans-

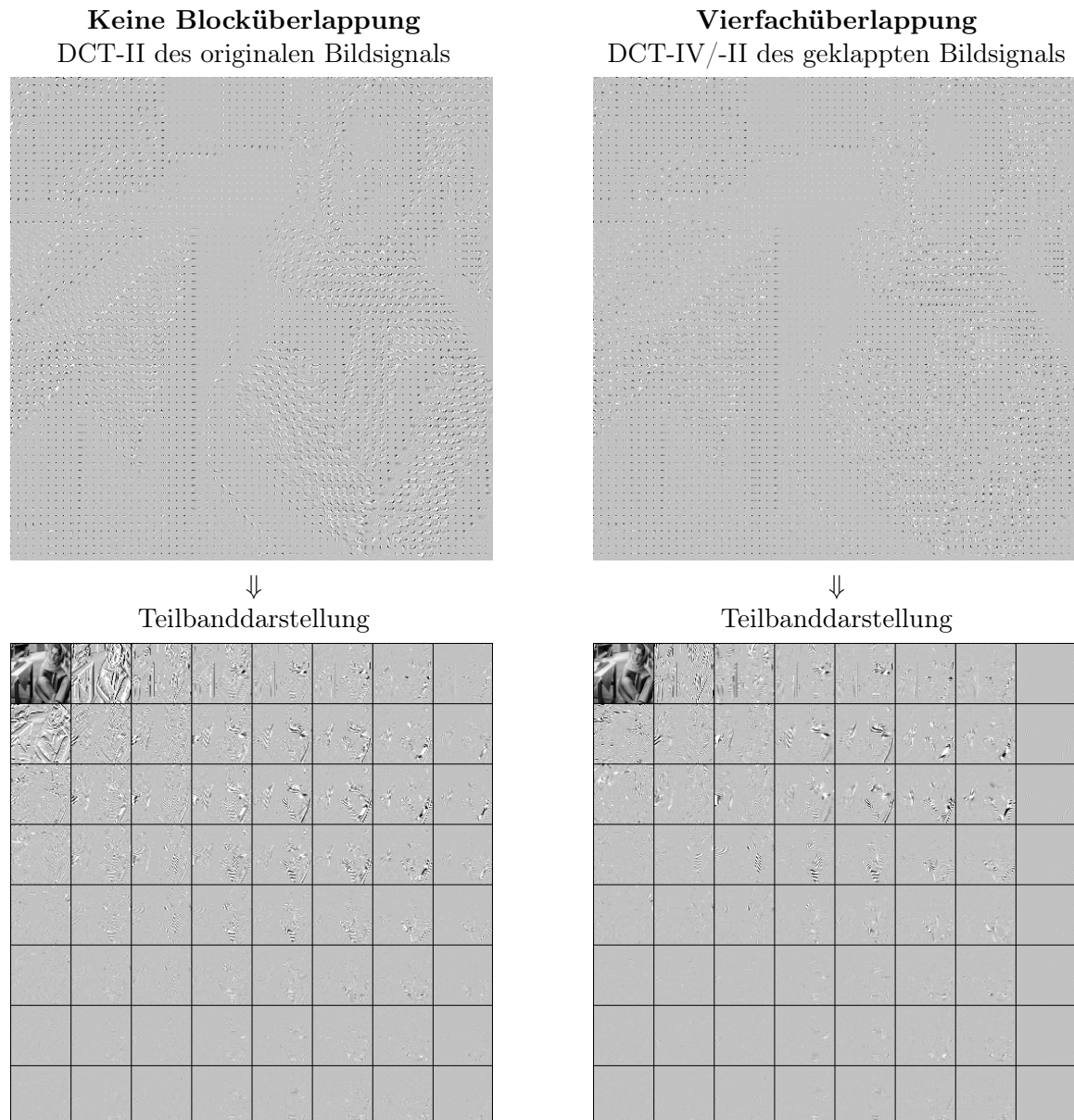


Abbildung 5.3: DCT-II-Transformierte des originalen Bildsignals (links) und DCT-IV/-II-Transformierte des geklappten Bildsignals (rechts); Umsortierung der Transformationskoeffizienten nach ihrer Frequenz (Teilbanddarstellung), Blockgröße $B = 8$.

formationskoeffizienten der Standard-DCT (DCT-II) des originalen und somit ungeklappten Bildes ebenfalls in Abbildung 5.3 dargestellt. Sortiert man die Transformationskoeffizienten aller Blöcke nach ihrer Frequenz, so wird durch die Gegenüberstellung der hieraus resultierenden Teilbänder beider Transformationen sichtbar, dass die vierfach überlappende Kosinustransformation eine wesentlich bessere Informationsverdichtung erzielt als die klassische Kosinustransformation (vgl. Teilbanddarstellung in Abbildung 5.3). Die Bildinformationen konzentrieren sich auf deutlich weniger Teilbänder und Transformationskoeffizienten, so dass das Bild durch Überlagerung einer geringeren Anzahl signifikanter Basisfunktionen rekonstruiert wird. Es ist daher zu erwarten, dass in der Bilddatenkompression bei diesem Bildbeispiel die Verwendung von vierfach überlappenden lokalen trigonometrischen Basen im Vergleich zur konventionellen 8×8 Block-DCT zu einem erheblichen Codiergewinn führt.

5.2 LCT-basierte SPIHT-Codierung

Der von Said und Pearlman entwickelte SPIHT-Algorithmus [SP96b, PS08] zählt zu den leistungsfähigsten Codierv Verfahren in der Bilddatenkompression. Dieser Algorithmus realisiert eine sukzessive Quantisierung der Transformationskoeffizienten und erzeugt ein progressives Übertragungsschema [Fey01]. Die Signalinformationen werden hierbei in absteigender Wichtigkeit übertragen, wobei sich der Datenstrom an einer beliebigen Stelle abbrechen lässt. Je mehr Daten übertragen werden, desto genauer kann der Empfänger das Bild rekonstruieren. Die Minimierung des Rekonstruktionsfehlers schreitet am schnellsten voran, wenn für alle Transformationskoeffizienten zuerst die höchst signifikanten Bits übertragen werden [Fey01]. Es folgen zur schrittweisen Verfeinerung der Signalinformationen die nächstniedrig signifikanten Bits. Mit jeder übertragenen Bitschicht verdoppelt sich durch die Halbierung des Quantisierungsintervalls die Auflösung der Transformationskoeffizienten.

Bei der SPIHT-Codierung werden durch die Definition einer Baumstruktur die Transformationskoeffizienten miteinander verbunden, die Informationen zu den gleichen Bildregionen enthalten. Das Grundprinzip der SPIHT-Codierung besteht darin, Abhängigkeiten zwischen korrespondierenden Transformationskoeffizienten verschiedener Auflösungsstufen bzw. verschiedener Frequenzbänder auszunutzen. Ist z.B. ein Transformationskoeffizient in einer bestimmten Auflösungsstufe insignifikant, so sind häufig auch die korrespondierenden Transformationskoeffizienten der feineren Auflösungsstufen insignifikant. Insignifikant bedeutet, dass der Betrag des Transformationskoeffizienten unter einer vorgegebenen Schwelle liegt. Sind auch für den Unterbaum alle Koeffizienten insignifikant, so wird dieser als Zerotree bzw. Nullbaum bezeichnet. Entscheidend für die Leistungsfähigkeit der SPIHT-Codierung ist die geschickte Gruppierung von insignifikanten Koeffizienten zu Nullbäumen, um diese Koeffizienten mit möglichst wenigen Bits kodieren zu können [Sch06]. Durch die sukzessive Halbierung der Schwelle entsteht schließlich der nach Bitebenen sortierte Datenstrom, wobei die höchst signifikanten Bits zuerst übertragen werden. Anhang F.1 gibt einen detaillierteren Überblick über die Funktionsweise der SPIHT-Codierung.

5.2.1 Suchbäume für blockweise angeordnete Transformationskoeffizienten

Bereits bei der Einführung des Zerotree-Codecs [Sha93] hat Shapiro hervorgehoben, dass die Codierungsstrategie, insignifikante Koeffizienten zu Nullbäumen zusammenzufassen, auch zur Verarbeitung von DCT-Koeffizienten verwendet werden kann. Der von ihm hierfür vorgeschlagene Suchbaum ist in Abbildung 5.4a) dargestellt. Der effizientere SPIHT-Algorithmus ist ebenfalls nicht auf den Wavelet-Baum beschränkt und lässt sich grundsätzlich auch für andere Baumstrukturen verwenden. Daher wurde der in Abbildung F.2 (Anhang F.1) angege-

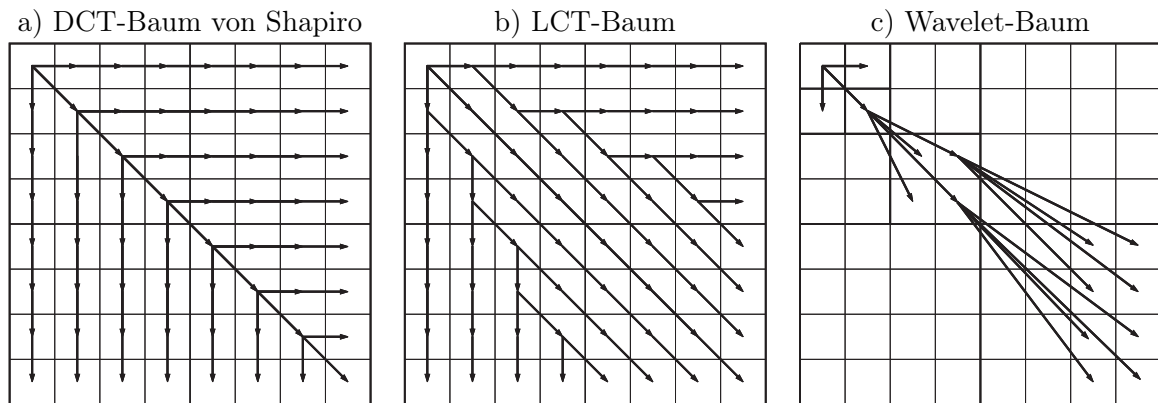


Abbildung 5.4: Baumstrukturen zur Codierung von DCT- und LCT-Koeffizienten.

bene SPIHT-Algorithmus so implementiert, dass für die Untersuchung verschiedener Baumstrukturen ein einfacher Austausch der Suchbäume vorgenommen werden kann. Im Rahmen dieser Arbeit wurden verschiedene Suchbäume entworfen und analysiert. Ein Beispiel ist in Abbildung 5.4b) dargestellt. Im Vergleich zum DCT-Baum von Shapiro können mit diesem Baum zwar geringfügig bessere Codierungsergebnisse erzielt werden, da diese Bäume im Vergleich zum Wavelet-Baum (vgl. Abbildung 5.4c)) jedoch mehr Knoten aufweisen, ist insgesamt eine größere Anzahl von sogenannten Entscheidungsbits zu übertragen. Hierdurch verringert sich die Codiereffizienz des SPIHT-Algorithmus. Obwohl der an die Oktavebandstruktur der Wavelet-Transformation angepasste Wavelet-Suchbaum die Eigenschaften der Transformationskoeffizienten uniformer Signalzerlegungen nicht optimal repräsentiert, erlaubt er dennoch eine leistungsfähigere Codierung dieser Koeffizienten. Daher konzentrieren sich die nachfolgenden Untersuchungen zur Kompression von Bilddaten unter Verwendung von überlappenden lokalen trigonometrischen Basen und der Codierung nach dem SPIHT-Algorithmus auf die Nutzung des Wavelet-Baumes.

5.2.2 Kompressionsstrategie SPIHT-Codec-I

Die in der Einleitung zu diesem Kapitel dargestellte Umsortierung von Transformationskoeffizienten in eine waveletähnliche Teilbandrepräsentation (vgl. Abbildung 5.1) eröffnet die Möglichkeit, zur Codierung der LCT-Koeffizienten die leistungsfähige Referenz-Implementierung des SPIHT-Codecs von Said und Pearlman [SP96a] zu nutzen¹. Für die LCT von Bilddaten wurden die in Kapitel 4 entwickelten Algorithmen zur effizienten Realisierung dieser Transformation mit Klappoperatoren und schnellen trigonometrischen Blocktransformationen in den SPIHT-Codec eingebunden. Die Verarbeitungsstufen des Transformationscodecs sind in Abbildung 5.5 angegeben.

Abbildung 5.7 zeigt am Beispiel des Testbildes „Barbara“ die rekonstruierten Bilder nach der Kompression bei einer Bitrate von 0.125 bpp unter Verwendung der 8×8 DCT, der 8×16 LCT-2, der 8×32 LCT-4 und der 3-level DWT (9/7-Wavelet). Wie zu erkennen ist, werden

¹In einer Reihe von Publikationen zu überlappenden Transformationen (z.B. in [QT00, LTT01, WWJ⁺08]) wird für die Codierung der Transformationskoeffizienten ebenfalls der Referenzcodec von Said und Pearlman [SP96a] genutzt. Daher existieren bei der Verwendung identischer Testbilder übereinstimmende Rahmenbedingungen für einen fairen Leistungsvergleich der LCT mit den publizierten Transformationen auf Grundlage der in diesen Publikationen angegebenen Kompressionsergebnisse.

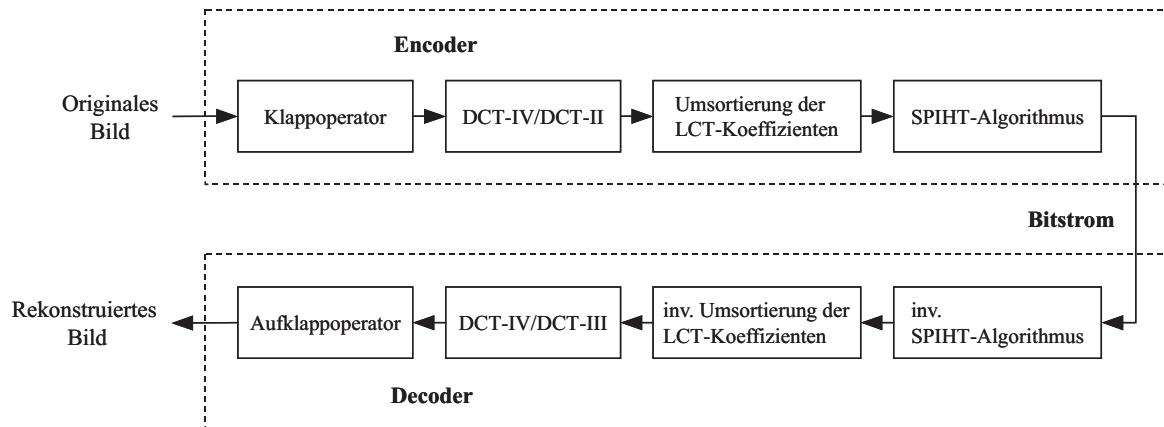


Abbildung 5.5: Blockschaftbild des SPIHT-Codec-I mit überlappender Kosinustransformation. Die überlappende Kosinustransformation wird mit Klapp- bzw. Aufklappoperatoren sowie trigonometrischen Blocktransformationen realisiert.

die Blockeffekte durch die Verwendung von überlappenden Basisfunktionen zwar reduziert, man erhält jedoch keinen gleichmäßigen Grauwertverlauf in den tieffrequenten Bildanteilen. Um z.B. mit der Vierfachüberlappung einen linearen Grauwertverlauf über mehrere Intervalle zu erzielen, müssen die Wichtungskoeffizienten der benachbarten Basisfunktionen ψ_j^0 eine sukzessive lineare Abstufung aufweisen. Eine solche lineare Abstufung ist aber nach der starken Quantisierung von ursprünglich fein abgestuften Transformationskoeffizienten nicht mehr gegeben. Wie die Kompressionsergebnisse in Abbildung 5.7 verdeutlichen, beschränkt sich dieses Problem nicht nur auf die überlappenden lokalen trigonometrischen Basen. Auch bei der strukturmäßig äquivalenten 3-level Wavelet-Transformation, die zu der gleichen Anzahl an DC-Koeffizienten führt wie die verwendeten Transformationen mit überlappenden lokalen trigonometrischen Basen, wird in dem gezeigten Kompressionsbeispiel keine Rekonstruktion der tieffrequenten Signalkomponenten ohne schnelle Grauwertwechsel erzielt.

5.2.3 Kompressionsstrategie SPIHT-Codec-II

Das im vorangegangenen Abschnitt aufgezeigte Problem der mangelhaften Darstellung tieffrequenter Bildinhalte bei niedrigen Bitraten löst sich bei der Wavelet-Transformation durch die Realisierung zusätzlicher Zerlegungsstufen. Werden in dem betrachteten Kompressionsbeispiel die 64×64 DC-Koeffizienten mit drei Zerlegungsstufen weiterverarbeitet, so erhält man in Bezug auf das Gesamtbild eine 6-level DWT. Wie in Abbildung 5.8 zu erkennen ist, werden die tieffrequenten Bildkomponenten nun sehr ausgewogen und ohne schnelle Grauwertwechsel dargestellt. Der gleiche Effekt lässt sich auch bei den analysierten mehrfach überlappenden Kosinustransformationen durch die Weiterverarbeitung der DC-Koeffizienten erzielen. In Analogie zu den Untersuchungsergebnissen in [TN99, KLX⁺06] hat sich auch bei der LCT die Zerlegung der DC-Koeffizienten mit der Wavelet-Transformation als sehr leistungsfähig erwiesen [MM03a, MM03b]. Abbildung 5.6 zeigt die Struktur des hieraus resultierenden Transformationscodes unter Verwendung von Klapp- und Aufklappoperatoren sowie der Nutzung schneller trigonometrischer Blocktransformationen.

Die beschriebene Verarbeitungsstrategie erzeugt zur effizienteren Darstellung von allmählichen Grauwertübergängen feiner abgestufte DC-Wichtungskoeffizienten der trigonometrischen Basisfunktionen ψ_j^0 . Betrachtet man im Codec-II die nacheinander ausgeführten Transforma-

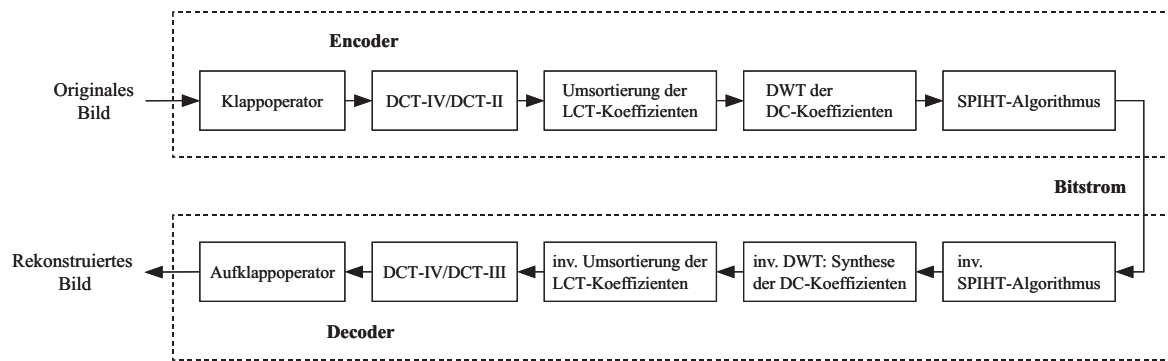


Abbildung 5.6: Blockschaltbild des SPIHT-Codec-II mit überlappender Kosinustransformation und mit Wavelet-Transformation der DC-Koeffizienten. Die überlappende Kosinustransformation wird mit Klapp- bzw. Aufklappoperatoren sowie trigonometrischen Blocktransformationen realisiert.

tionen in ihrer Gesamtheit, so werden im Vergleich zum Codec-I die tieffrequenten Bildkomponenten durch das Prinzip der Mehrfachauflösung mit Basisfunktionen mit längerem Träger reproduziert. Bei den in Abbildung 5.8 dargestellten Kompressionsbeispielen wurden die 64×64 DC-Koeffizienten der trigonometrischen Transformationen mit einer 3-level Wavelet-Transformation weiterverarbeitet. Wie anhand der Rekonstruktionsergebnisse zu erkennen ist, erhält man in Analogie zur 6-level DWT mit dem 9/7-Wavelet schließlich auch bei der vierfach überlappenden LCT-4 einen gleichmäßigen Grauwertverlauf in den tieffrequenten Bildbereichen. Blockeffekte, wie sie auch beim SPIHT-Codec-II unter Verwendung der DCT entstehen, sind nicht sichtbar und es treten auch keine schnellen Grauwertwechsel wie bei der zweifach überlappenden LCT-2 (Malvar-Basis) auf.

Die bisher betrachteten Kompressionsbeispiele zeigen bereits deutlich das Potential der untersuchten vierfach überlappenden lokalen trigonometrischen Basis im Vergleich zur Wavelet-Transformation. Obwohl die Verarbeitungsstrategien der SPIHT-Codierung unter Nutzung des Wavelet-Baumes an die Struktur der Wavelet-Transformation angepasst sind und die Eigenschaften der Wavelet-Transformation ausnutzen, werden bei dem in Abbildung 5.8 gezeigten Bildbeispiel „Barbara“ durch die Verwendung der vierfach überlappenden trigonometrischen Basis gegenüber dem 9/7-Wavelet signifikant bessere subjektive und objektive Rekonstruktionsergebnisse erzielt. Insbesondere die harmonisch strukturierten Texturen der Tischdecke, des Kopftuches, der Hose und des Korbsessels werden mit den Basisfunktionen der LCT-4 besser dargestellt als mit dem 9/7-Wavelet. Sichtbare Ringing-Artefakte treten bei der niedrigen Bitrate von 0.125 bpp sowohl bei der 8×32 LCT-4 als auch bei der 6-level DWT mit dem 9/7-Wavelet auf. An einigen Kanten sind sie bei der DWT ausgeprägter, an anderen Kanten bei der LCT-4. Weitere Untersuchungsergebnisse zur Beurteilung der Kompressionsleistung der vorgestellten Transformationscodecs werden in Kapitel 6 präsentiert.

8×8 DCT, PSNR = 23.96 dB 8×16 LCT-2, PSNR = 25.07 dB 8×32 LCT-4, PSNR = 25.36 dB

3-level DWT, PSNR = 24.16 dB



Abbildung 5.7: SPIHT-Codec-I: Kompression des Bildes „Barbara“ bei einer Bitrate von 0.125 bpp; Vergleich von 8×8 DCT, 8×16 LCT-2, 8×32 LCT-4 und 3-level DWT (9/7-Wavelet). Bei allen Kompressionsbeispielen besteht das DC-Teilband aus 64×64 Transformationskoeffizienten.



Abbildung 5.8: SPIHT-Codec-II: Kompression des Bildes „Barbara“ bei einer Bitrate von 0.125 bpp; Vergleich von 8×8 DCT, 8×16 LCT-2, 8×32 LCT-4 und 6-level DWT (9/7-Wavelet). Bei den trigonometrischen Transformationen wird das 64×64 DC-Teilband mit der 3-level DWT (9/7-Wavelet) weiterverarbeitet, so dass hieraus resultierend bei allen Kompressionsbeispielen das DC-Teilband aus 8×8 Transformationskoeffizienten besteht.

5.3 LCT-basierte L-CEB-Codierung

Mit den beiden von Tu und Tran [TT02b, TT02a] eingeführten L-CEB¹ und E-CEB² Codierungsstrategien stehen zwei sehr leistungsfähige Verfahren zur kontextbasierten Entropiecodierung von blockweise angeordneten Transformationskoeffizienten, die das Ergebnis überlappender oder nicht überlappender trigonometrischer Transformationen sind, zur Verfügung. Der L-CEB-Codec wurde im Gegensatz zum E-CEB-Codec, der bei einer höheren algorithmischen Komplexität nur geringfügig bessere Kompressionsergebnisse liefert [TT02b]³, speziell für die schnelle und speichereffiziente Codierung von quantisierten Transformationskoeffizienten entwickelt. Mit dem Ziel der effizienten und ressourcenschonenden Kompression von Bilddaten auf Grundlage der schnellen überlappenden Kosinustransformation wird im Folgenden die L-CEB-Codierung der quantisierten Transformationskoeffizienten untersucht. Details zur Funktionsweise der L-CEB-Codierung von blockweise angeordneten Transformationskoeffizienten sind dem Anhang F.2 zu entnehmen.

5.3.1 Kompressionsstrategie L-CEB-Codec-I

Das Blockdiagramm in Abbildung 5.9 stellt dar, wie Bilddaten auf Grundlage der L-CEB-Codierung unter Verwendung von überlappenden lokalen trigonometrischen Basen komprimiert bzw. rekonstruiert werden. Zunächst erfolgt die Verarbeitung des Bildes mit Klappoperatoren. Hierbei kommen die in Kapitel 4 entwickelten Algorithmen zur Anwendung. Danach wird das geklappte Bildsignal mit den schnellen trigonometrischen Blocktransformationen DCT-IV/II transformiert. Anschließend werden die Transformationskoeffizienten der in Abschnitt 1.3 beschriebenen gleichmäßigen Quantisierung mit Totzone unterworfen und die quantisierten Transformationskoeffizienten der L-CEB-Codierung zugeführt. Für die Analyse der Leistungsfähigkeit der im Rahmen dieser Arbeit untersuchten überlappenden Kosinustransformationen in der Bilddatenkompression wurde von den Autoren Tu/Tran [TT02b, TT02a] die Referenzimplementierung der Codec-Engine des L-CEB-Verfahrens zur Verfügung gestellt.

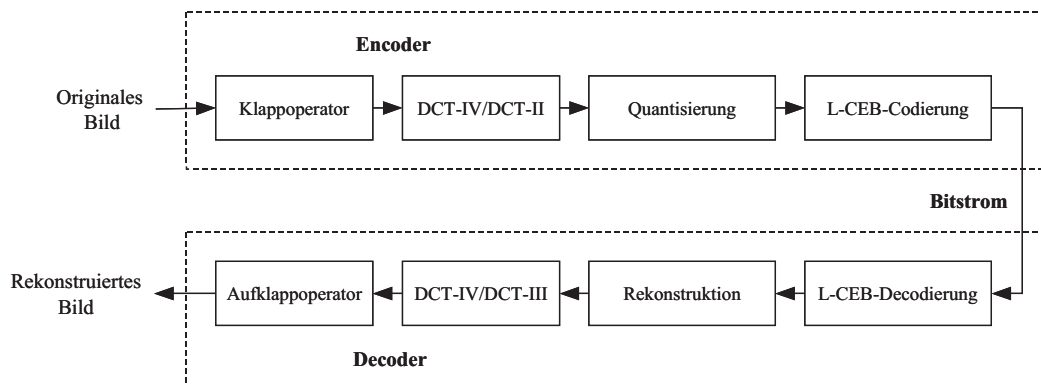


Abbildung 5.9: Blockschaltbild des L-CEB-Codec-I mit überlappender Kosinustransformation. Die überlappende Kosinustransformation wird mit Klapp- bzw. Aufklappoperatoren sowie trigonometrischen Blocktransformationen realisiert.

¹L-CEB: local context-based entropy coding of block transform coefficients

²E-CEB: embedded context-based entropy coding of block transform coefficients

³Die in [TT02b] für die L-CEB- und E-CEB-Codierung angegebenen objektiven Rekonstruktionsergebnisse (PSNR) unterscheiden sich im Mittel um ca. 0.2 dB.

Abbildung 5.10 zeigt die Rekonstruktionsergebnisse des Bildes „Barbara“ nach der Kompression bei einer Bitrate von 0.125 bpp unter Verwendung der DCT, der LCT-2, der LCT-4 und der DWT (9/7-Wavelet). Bei diesen Kompressionsbeispielen wird sowohl im Vergleich zu der in Abschnitt 5.2.2 vorgestellten Kompressionsstrategie SPIHT-Codec-I als auch im Vergleich zu der in Abschnitt 5.2.3 vorgestellten Kompressionsstrategie SPIHT-Codec-II eine signifikante Verbesserung der objektiven Rekonstruktionsqualität erzielt. So ist z.B. bei der vierfach überlappenden Kosinustransformation durch die Nutzung des L-CEB-Codec-I im Vergleich zum SPIHT-Codec-I eine Erhöhung des PSNR um 1.63 dB und im Vergleich zum SPIHT-Codec-II eine Erhöhung des PSNR um 0.56 dB zu verzeichnen (vgl. Abbildungen 5.7, 5.8 und 5.10). Die höhere objektive Rekonstruktionsqualität geht insbesondere mit einer deutlich sichtbaren Verbesserung der Darstellung der harmonisch strukturierten Texturen einher. Bemerkenswert ist auch die Leistungsfähigkeit des L-CEB-Codec-I bei der Codierung von Wavelet-Transformationskoeffizienten. Hierzu werden die Wavelet-Koeffizienten einer 3-level DWT gemäß Abbildung 5.1 in eine 8×8 Blockstruktur umsortiert. Obwohl die L-CEB-Codierung von Wavelet-Transformationskoeffizienten nicht optimal an die Eigenschaften einer Wavelet-Transformation angepasst ist, erzielt der L-CEB-Codec-I unter Verwendung der 3-level DWT (9/7-Wavelet) mit einem PSNR von 24.83 dB fast die objektive Rekonstruktionsqualität von 24.86 dB des SPIHT-Codec-II unter Verwendung der 6-level DWT (vgl. Abbildung 5.8 und 5.10).

Hinsichtlich der subjektiven Rekonstruktionsqualität ist bei den betrachteten Kompressionsbeispielen jedoch zu erkennen, dass in Analogie zum SPIHT-Codec-I auch beim L-CEB-Codec-I keine gleichmäßig fein abgestufte Darstellung der tieffrequenten Signalanteile erfolgt.

5.3.2 Kompressionsstrategie L-CEB-Codec-II

Wie bereits in Abschnitt 5.2.3 zur SPIHT-Codierung beschrieben wurde, lässt sich das aufgezeigte Problem der mangelhaften Darstellung tieffrequenter Bildkomponenten, das beim L-CEB-Codec-I ebenfalls auftritt, durch die weitere Zerlegung der DC-Transformationskoeffizienten beheben. Dieser Lösungsansatz kommt daher auch bei der in Abbildung 5.12 vorgestellten Kompressionsstrategie L-CEB-Codec-II zur Anwendung. Hierbei werden die zu komprimierenden Bilder zunächst mit einer geeigneten überlappenden Kosinustransformation unter Verwendung von Klappoperatoren und schnellen trigonometrischen Blocktransformationen verarbeitet. Danach werden die DC-Transformationskoeffizienten und AC-Transformationskoeffizienten voneinander separiert. Zur weiteren Zerlegung der DC-Transformationskoeffizienten hat sich in den Untersuchungen wieder die dreistufige Wavelet-Transformation mit dem 9/7-Wavelet als leistungsfähig erwiesen. Anschließend werden die Wavelet-Transformationskoeffizienten, wie in Abbildung 5.1 dargestellt, in eine 8×8 Blockstruktur umsortiert, quantisiert und der L-CEB-Codierung zugeführt. Parallel zu diesen Verarbeitungsschritten erfolgt die Quantisierung und Codierung der AC-Transformationskoeffizienten.

Abbildung 5.11 zeigt für die DCT, die LCT-2, die LCT-4 und die DWT (9/7-Wavelet) die Kompression des Bildes „Barbara“ unter Verwendung des L-CEB-Codec-II. Es ist im Vergleich zu den Rekonstruktionsergebnissen in Abbildung 5.10 zu erkennen, dass durch die Weiterverarbeitung der DC-Koeffizienten die tieffrequenten Bildanteile nun wesentlich besser dargestellt werden. Diesbezüglich liefert neben der DWT schließlich auch die vierfach überlappende LCT-4 ein rekonstruiertes Bild, das frei von sichtbaren Kompressionsartefakten in Form von schnellen Grauwertwechseln ist. Es treten keine sichtbaren blockartigen Kompressionsartefakte mehr auf.

Der Qualitätsabstand zwischen der Kompressionsstrategie I und II der L-CEB-Codierung fällt bei den bisher gezeigten Kompressionsbeispielen nicht so hoch aus wie bei der SPIHT-Codie-

8×8 DCT, PSNR = 24.84 dB 8×16 LCT-2, PSNR = 26.23 dB 8×32 LCT-4, PSNR = 26.99 dB

3-level DWT, PSNR = 24.83 dB



Abbildung 5.10: L-CEB-Codec-I: Kompression des Bildes „Barbara“ bei einer Bitrate von 0.125 bpp; Vergleich von 8×8 DCT, 8×16 LCT-2, 8×32 LCT-4 und 3-level DWT (9/7-Wavelet). Bei allen Kompressionsbeispielen besteht das DC-Teilband aus 64×64 Transformationskoeffizienten.



Abbildung 5.11: L-CEB-Codec-II: Kompression des Bildes „Barbara“ bei einer Bitrate von 0.125 bpp; Vergleich von 8×8 DCT, 8×16 LCT-2, 8×32 LCT-4 und 6-level DWT (9/7-Wavelet). Bei den trigonometrischen Transformationen wird das 64×64 DC-Teilband mit der 3-level DWT (9/7-Wavelet) weiterverarbeitet, so dass hieraus resultierend bei allen Kompressionsbeispielen das DC-Teilband aus 8×8 Transformationskoeffizienten besteht.

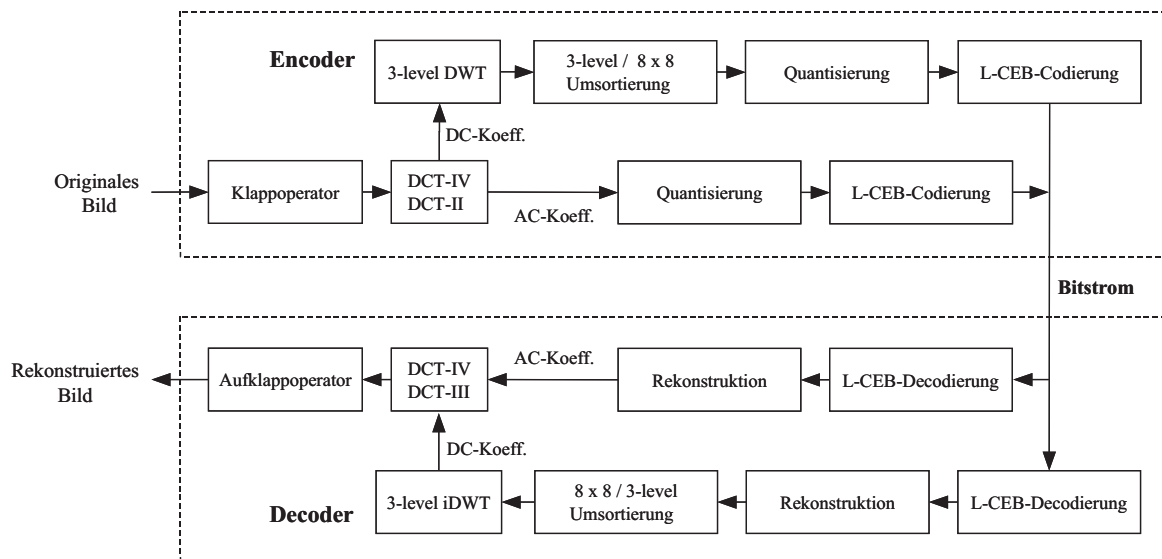


Abbildung 5.12: Blockschaltbild des L-CEB-Codec-II mit überlappender Kosinustransformation und mit Wavelet-Transformation der DC-Koeffizienten. Die überlappende Kosinustransformation wird mit Klapp- bzw. Aufklappoperatoren sowie trigonometrischen Blocktransformationen realisiert.

rung. Das liegt u.a. daran, dass der L-CEB-Codec-I bei diesen Kompressionsbeispielen bereits sehr effizient arbeitet, während die SPIHT-Codierung, deren Codiereffizienz entscheidend von der geschickten Gruppierung insignifikanter Koeffizienten zu Nullbäumen abhängt, ihre Leistungsfähigkeit grundsätzlich erst ab einer gewissen Baumordnung bzw. Baumtiefe und somit erst ab einer höheren Anzahl von Wavelet-Zerlegungsstufen entfaltet. In Abhängigkeit von den Bildinhalten sind jedoch auch bei der L-CEB-Codierung durch die Verwendung der Kompressionsstrategie II im Vergleich zur Kompressionsstrategie I deutliche Qualitätsverbesserungen zu erzielen, wie es aus Abbildung 5.13 am Beispiel des Testbildes „Zelda“ hervorgeht. Insbesondere bei sehr niedrigen Bitraten kommt die Leistungsfähigkeit der Kompressionsstrategie L-CEB-Codec-II zum Tragen. Bezüglich der Präsentation und Auswertung weiterer Kompressionsergebnisse sei auf Kapitel 6 verwiesen.

5.4 Erhöhung der Blockgröße und LCT-basierte AGU-Codierung

Die Untersuchungen in den vorangegangenen Abschnitten konzentrierten sich auf die Codierung von Transformationskoeffizientenblöcken der Größe 8×8 . Diese Koeffizientenblöcke können bzgl. der im Rahmen dieser Arbeit untersuchten trigonometrischen Transformationen das Resultat einer 8×8 DCT, 8×16 LCT-2, 8×24 LCT-3, 8×32 LCT-4 oder einer 8×48 LCT-6 sein. Bei allen aufgezählten Transformationen werden die diskreten Signale durch Überlagerung von 8 Basisfunktionen je Teilintervall reproduziert, wobei die Länge der Träger der Basisfunktionen vom verwendeten Überlappungsfaktor abhängt. In diesem Abschnitt soll mit dem Ziel der Steigerung der Kompressionsleistung die Möglichkeit untersucht werden, zur besseren Dekorrelation von Bildsignalen die Blockgröße zu erhöhen. Es sei jedoch darauf hingewiesen, dass die Transformationen für größere Blöcke mit einem erhöhten Berechnungsaufwand einhergehen (vgl. Tabelle 4.4). Des Weiteren verschlechtert sich die Lokalität

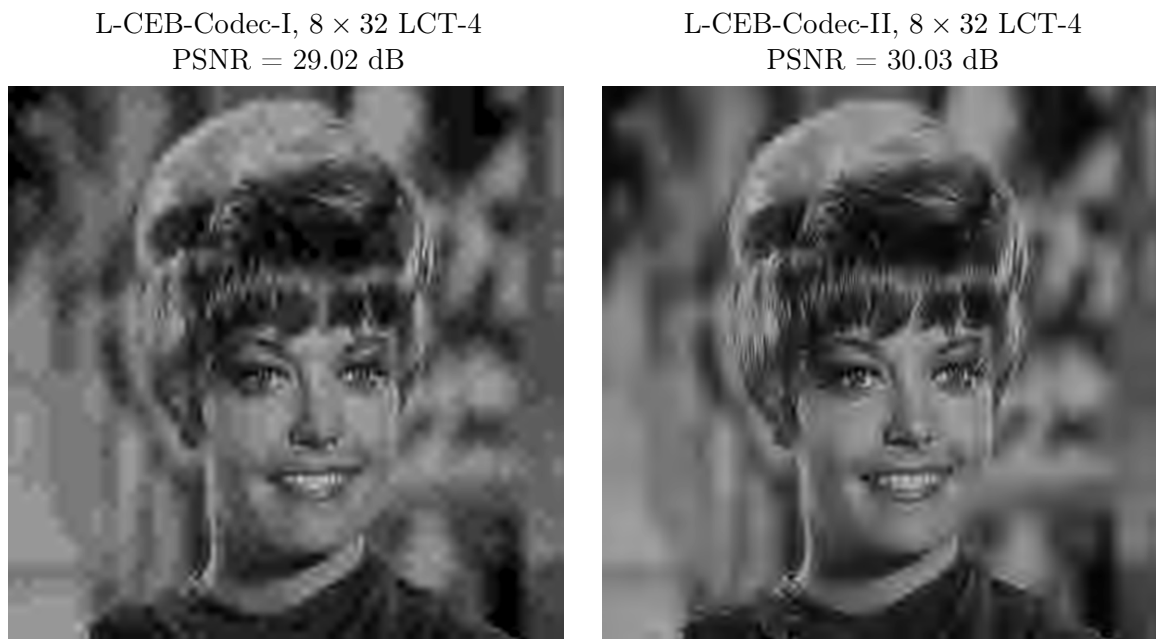


Abbildung 5.13: Vergleich der Rekonstruktionsqualität zwischen den Kompressionsstrategien L-CEB-Codec-I und L-CEB-Codec-II am Beispiel des Testbildes „Zelda“ bei einer Bitrate von 0.035 bpp.

der Basisfunktionen im Ortsbereich, so dass in der Bilddatenkompression insbesondere bei größeren Überlappungsfaktoren deutlich sichtbare Ringing-Artefakte zu erwarten sind. Der in Abschnitt 5.2 verwendete SPIHT-Algorithmus lässt sich ohne tiefgreifende Modifikationen zur Codierung von Koeffizientenblöcken der Größe 16×16 oder 32×32 nutzen und es kommt die in Abbildung 5.1 dargestellte Umsortierung der Transformationskoeffizienten zur Anwendung. Für die in Abschnitt 5.3 verwendete L-CEB-Codierung sind hingegen umfangreichere Untersuchungen erforderlich, um ein leistungsfähiges Kontextmodell für größere Koeffizientenblöcke herzuleiten. Die Entwicklung von neuen effizienten Verfahren zur kontextbasierten Entropiecodierung ist jedoch nicht Untersuchungsgegenstand dieser Arbeit. Da mit dem in [PELA05] vorgestellten AGU-Codec¹ bereits ein leistungsfähiges Verfahren zur kontextbasierten arithmetischen Codierung von größeren Koeffizientenblöcken existiert, wird im Folgenden die AGU-Codierung unter Verwendung von überlappenden trigonometrischen Basen untersucht und ihre Leistungsfähigkeit im Vergleich zur SPIHT-Codierung analysiert. Mit der Einführung des AGU-Codex in [PELA05] sollte gezeigt werden, dass die besseren Rekonstruktionsergebnisse neuer waveletbasierter Kompressionsverfahren im Vergleich zum JPEG-Standard nicht die Schlussfolgerung zulassen, dass die DWT in der Bilddatenkompression leistungsfähiger sei als die DCT. So geht schließlich aus den in [PELA05, Pon08b] dokumentierten Kompressionsergebnissen hervor, dass der AGU-Codec auf Grundlage der DCT im Vergleich zu den waveletbasierten Verfahren JPEG2000 und SPIHT bei fast allen Kompressionsbeispielen eine bessere Rekonstruktionsqualität liefert. Die hohe Leistungsfähigkeit der AGU-Codierung wird in Bezug auf die JPEG-Codierung durch folgende Modifikationen erzielt:

¹Der AGU-Codec ist zu Forschungszwecken als ausführbare Programmdatei auf den Webseiten von Egiazarian und Ponomarenko [Egi05, Pon08b] erhältlich.

1. Statt einer Blockgröße von 8×8 wird eine Blockgröße von 32×32 verwendet. Es können jedoch auch Koeffizientenblöcke der Größe 16×16 oder 64×64 sehr effizient codiert werden.
2. Die quantisierten Transformationskoeffizienten werden auf Bitebene codiert. Bei der Codierung werden benachbarte Koeffizienten innerhalb des Blocks und Koeffizienten der benachbarten Blöcke berücksichtigt.
3. Zur Reduktion von Blockartefakten erfolgt eine Nachbearbeitung¹ des rekonstruierten Bildes.

Die AGU-Codierung von Koeffizientenblöcken geringerer Blockgröße ist auch möglich, jedoch nicht so effizient. Man erhält z.B. bei der Codierung von Koeffizientenblöcken der Größe 8×8 für die im Rahmen dieser Arbeit untersuchten Testbilder keine besseren Kompressionsergebnisse als bei der L-CEB-Codierung und der AGU-Codierungsprozess ist mit einer höheren Programmlaufzeit verbunden. Die AGU-Codierung von Koeffizientenblöcken geringerer Blockgröße findet daher im Folgenden keine weitere Berücksichtigung.

Für die Untersuchung der Leistungsfähigkeit von überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression unter Verwendung der AGU-Codierung wurde von Ponomarenko der Quelltext der DCT-basierten Referenzimplementierung des AGU-Codecs zur Verfügung gestellt. Da die von der LCT und der DCT erzeugten Koeffizientenblöcke eine äquivalente Teilbandstruktur aufweisen, lässt sich die Kosinustransformation im AGU-Codec durch die überlappende Kosinustransformationen ersetzen, ohne dass eine Anpassung der Codierungsstrategie erforderlich ist. Zur Quantisierung der Transformationskoeffizienten kommt die in Abschnitt 1.3 beschriebene gleichmäßige Quantisierung mit Totzone² zur Anwendung. Im Anschluss werden die quantisierten Transformationskoeffizienten binarisiert und auf Bitebene codiert. Für eine nähere Beschreibung der im AGU-Codec realisierten kontextbasierten arithmetischen Codierung sei auf den Anhang F.3 verwiesen.

In Analogie zur SPIHT-Codierung und zur L-CEB-Codierung wurden auch bei der AGU-Codierung zwei Kompressionsstrategien³ untersucht, die sich in der Verarbeitung der DC-Koeffizienten unterscheiden. Die Realisierungen des AGU-Codec-I und des AGU-Codec-II erfolgen prinzipiell nach den Blockschaltbildern, die bereits bei der CEB-Codierung Anwendung finden (vgl. Abbildungen 5.9 und 5.12), wobei die CEB-Codierungsstufe durch die AGU-Codierungsstufe ersetzt wird. Aufgrund der höheren Blockgrößen werden im AGU-Codec-II die DC-Koeffizienten mit einer niedrigeren Anzahl an Zerlegungsstufen weiterverarbeitet, so dass im Rahmen dieser Arbeit für den Vergleich der Kompressionsleistung zwischen L-CEB-Codec-II, SPIHT-Codec-II und AGU-Codec-II das DC-Teilband immer aus 8×8 Transformationskoeffizienten besteht. Bei einer Blockgröße von 16×16 kommt daher die 2-level DWT (9/7-Wavelet) zur Anwendung. Bei einer Blockgröße von 32×32 erfolgt die Zerlegung der DC-Koeffizienten mit der 1-level DWT (9/7-Wavelet).

¹Im Hinblick auf einen fairen Leistungsvergleich verschiedener Kompressionsverfahren ist die Nachbearbeitung des rekonstruierten Bildes bei nur einem Transformationscodec kritisch zu bewerten. Da der im AGU-Codec verwendete DCT-Filter auch bei anderen Transformationscodern zu einer Verbesserung der subjektiven und objektiven Rekonstruktionsqualität führt, sollten beim Vergleich der Kompressionsleistung entweder alle Codecs eine solche Nachbearbeitung nutzen – oder keiner.

²In [PELA05] kommt hingegen die klassische Midtread-Quantisierung zum Einsatz, da sie in Verbindung mit dem Nachbearbeitungsschritt des Deblockings zu geringfügig besseren Kompressionsergebnissen führen kann. Verzichtet man jedoch auf diesen rechenaufwendigen Schritt der Nachbearbeitung, so erweist sich im Allgemeinen die Quantisierung mit Totzone als leistungsfähiger.

³Dieser Abschnitt konzentriert sich auf die AGU-Kompressionsstrategie II, da mit dieser im Vergleich zur Kompressionsstrategie I ebenso wie bei der SPIHT- und der L-CEB-Kompression eine bessere Rekonstruktion der tieffrequenten Bildinhalte erzielt wird.

In Abbildung 5.14 sind die Rekonstruktionsergebnisse des Bildes „Barbara“ nach der Kompression mit dem AGU-Codec-II sowie dem SPIHT-Codec-II unter Verwendung der 32×32 DCT und der 16×32 LCT-2 bei einer Bitrate von 0.125 bpp dargestellt. Um festzustellen, wie sich die Erhöhung der Blockgröße auf die Kompressionsleistung auswirkt und dabei möglichst einheitliche Rahmenbedingungen bzgl. der Quantisierung und Entropiecodierung gelten sollen, wird im Folgenden zunächst auf die Rekonstruktionsergebnisse der SPIHT-Codierung eingegangen. Der Vergleich der Rekonstruktionsergebnisse aus Abbildung 5.8 mit 5.14 zeigt, dass bei dem Bildbeispiel „Barbara“ die Erhöhung der Blockgröße der DCT von 8×8 auf 32×32 im SPIHT-Codec-II zu einer signifikanten Steigerung der objektiven und subjektiven Rekonstruktionsqualität führt. Das PSNR erhöht sich um 1.68 dB auf 26.25 dB. Auch bei der Zweifachüberlappung bewirkt der Wechsel von der 8×16 LCT-2 auf die 16×32 LCT-2 unter Verwendung der SPIHT-II-Codierung eine deutliche Verbesserung der subjektiven und der objektiven Rekonstruktionsqualität. Das PSNR erhöht sich bei dem betrachteten Kompressionsbeispiel um 0.94 dB auf 26.71 dB. Im Vergleich zur 8×16 LCT-2 werden durch die Verwendung der 16×32 LCT-2 die Kompressionsartefakte in Form von sichtbaren schnellen Grauwertwechsel in den tieffrequenten Bildbereichen reduziert und die harmonischen Texturen besser dargestellt. Aufgrund der längeren Träger der Basisfunktionen der 16×32 LCT-2 sind in der Umgebung von Kanten jedoch ausgeprägtere Ringing-Artefakte zu erkennen. Äußerst bemerkenswert sind diese Kompressionsergebnisse im Vergleich zur klassischen SPIHT-Codierung, die auf der 6-level DWT (9/7-Wavelet) beruht. Mit ihr wird bei diesem Kompressionsbeispiel nur ein PSNR von 24.85 dB erzielt und man erhält eine deutlich schlechtere Darstellung der harmonisch strukturierten Texturen (vgl. Abbildungen 5.8 und 5.14).

Die Rekonstruktionsergebnisse der AGU-Codierung¹ unter Verwendung der 32×32 DCT bzw. der 16×32 LCT-2 zeigen, dass sich durch eine an die Blockstruktur angepasste Codierungsstrategie die PSNR-Werte im Vergleich zur SPIHT-Codierung verbessern (vgl. Abbildung 5.14). Visuell unterscheiden sich jedoch die rekonstruierten Bilder, die man mit der AGU- und der SPIHT-Codierung unter Verwendung der gleichen trigonometrischen Transformation erhält, nur geringfügig. Weitere Kompressionsergebnisse werden in Kapitel 6 vorgestellt und ausgewertet.

Abschließend sei darauf hingewiesen, dass die Implementierung der AGU-Codierung mehr Rechenzeit benötigt als die Implementierung der SPIHT-Codierung und der L-CEB-Codierung.

5.5 Reduktion von Kompressionsartefakten

Leistungsfähige verlustbehaftete Bilddatenkompressionsverfahren sind neben einer schnellen Verarbeitungsgeschwindigkeit dadurch gekennzeichnet, dass die aus einer möglichst geringen Datenmenge rekonstruierten Bilder eine von Kompressionsartefakten ungestörte Wahrnehmung der wesentlichen Bildinhalte gewährleisten. Werden jedoch bestimmte Kompressionsverhältnisse überschritten, wie es in der Praxis nicht unüblich ist, so äußern sich die starken Datenverluste bei allen untersuchten Kompressionsstrategien durch sichtbare Artefakte. Welche Typen von Kompressionsartefakten in den rekonstruierten Bildern auftreten und wie umfangreich die jeweiligen Störungen ausgeprägt sind, hängt in der Transformationscodierung wesentlich mit den für die Signalrepräsentation verwendeten Basisfunktionen zusammen. Zu den bekanntesten Artefakten zählen Blocking, Blurring (Unschärfe) und Ringing. Eine ausführliche Übersicht über diese und weitere Arten von Kompressionsartefakten ist in [YW98, SK98, Pia08] zu finden. Kommt es in den rekonstruierten Bildern zu sichtbaren Kom-

¹Es sei darauf hingewiesen, dass für einen fairen Leistungsvergleich die im AGU-Codec implementierte Nachbearbeitungsstufe deaktiviert wurde.



Abbildung 5.14: AGU-Codec-II und SPIHT-Codec-II im Vergleich: Kompression des Bildes „Barbara“ bei einer Bitrate von 0.125 bpp unter Verwendung der 32×32 DCT (links) und der 16×32 LCT-2 (rechts).

pressionsartefakten, so werden häufig auf der Empfängerseite Artefaktreduktionsverfahren eingesetzt, um nachträglich die Sichtbarkeit der Störungen zu reduzieren. Trotz der Existenz von sehr wirksamen Nachbearbeitungsverfahren bleibt für die Transformationscodierung weiterhin primär das Ziel bestehen, durch die Entwicklung und den Einsatz von geeigneten Transformationen bereits das Aufkommen von wahrnehmbaren Kompressionsartefakten weitestgehend zu vermeiden. So lassen sich z.B. Blockartefakte, wie sie bei den DCT-basierten Kompressionsverfahren auftreten können, u.a. durch die Verwendung der Wavelet-Transformation oder der mehrfach überlappenden Kosinustransformation vollständig verhindern.

Blurring- bzw. Unschärfe-Artefakte äußern sich durch Detailverluste in Bildbereichen mit hoher örtlicher Aktivität [Pia08]. Diese Störungen resultieren aus einer zu starken Quantisierung bzw. der Auslöschung von hochfrequenten Transformationskoeffizienten, die für die Repräsentation von feinen Strukturen notwendig sind. Wie die bisherigen Rekonstruktionsergebnisse zeigen, lassen sich Bildbereiche, die durch feine harmonische Texturen geprägt sind, deutlich besser mit überlappenden lokalen trigonometrischen Basen rekonstruieren als mit Wavelets (vgl. Abbildungen 5.8 und 5.11). Kompressionsfehler, die sich durch Überschwinger bzw. Doppelkonturen an Kanten bemerkbar machen, werden als Ringing-Artefakte bezeichnet (vgl. Abschnitt 1.1.3). Sie sind ebenfalls das Ergebnis einer zu starken Quantisierung von hochfrequenten Transformationskoeffizienten.

In Abbildung 5.15 ist das rekonstruierte Testbild „Barbara“ nach der Kompression mit dem AGU-Codec-II bei einer Bitrate von 0.125 bpp unter Verwendung der 32×64 LCT-2 dargestellt. Mit einem PSNR von 27.53 dB wird im Vergleich zu allen bisher gezeigten Kompressionsbeispielen des Bildes „Barbara“ zwar die höchste objektive Rekonstruktionsqualität erzielt, aufgrund der sehr groben Quantisierung und der sehr langen Träger der Basisfunktionen der 32×64 LCT-2 treten jedoch in einem Großteil des rekonstruierten Bildes deutlich sichtbare Ringing-Artefakte in der Umgebung von Kanten hervor. Insbesondere ist auch eine Fortsetzung der harmonisch strukturierten Texturen auf die benachbarten Bildbereiche



Abbildung 5.15: AGU-Codec-II: Kompression des Bildes „Barbara“ bei einer Bitrate von 0.125 bpp; Vergleich von 32×64 LCT-2 (links) und 32×64 LCT-2 mit DCT-basierter Postfilterung (rechts).

zu beobachten, z.B. vom Kopftuch zu den Haaren, vom Kopftuch zum Kinn, vom Kopftuch zum Oberkörper, von den Hosenbeinen zum Oberkörper und von der Tischdecke zum vorderen Tischbein. Diese sehr deutlich ausgeprägten Rekonstruktionsfehler lassen sich senderseitig z.B. durch die Auswahl von Basisfunktionen mit kürzerem Träger oder durch eine feinere Quantisierung der hochfrequenten Transformationskoeffizienten vermindern. Treten ungeachtet dessen auf der Empfangsseite die genannten Kompressionsartefakte auf, so können sie mit Hilfe von Nachbearbeitungsverfahren reduziert werden.

Zur Verringerung von Kompressionsartefakten existiert eine Vielzahl von Ansätzen, die sich in der Komplexität, der Funktionsweise und der Wirksamkeit unterscheiden [Pia08]. Um den Rahmen dieser Arbeit nicht zu sprengen, beschränken sich die Untersuchungen auf die DCT-basierte Filterung [EHKA99, Yar98], die bereits im AGU-Codec [PELA05] Anwendung findet. Der DCT-basierte Filter dient dort ursprünglich zur Reduktion von Blockartefakten, indem er Quantisierungsrauschen aus dem rekonstruierten Bild entfernt. Hierzu werden die Bildpunkte, die sich in einem pixelweise über das Bild gleitenden 8×8 Fenster befinden, mit einer schnellen DCT transformiert. Anschließend werden die Transformationskoeffizienten, deren Beträge unter einer festgelegten Schwelle liegen, auf Null gesetzt (Hard-Tresholding) und es erfolgt die Rücktransformation¹. Aufgrund dieses Nachbearbeitungsschrittes erhöht sich bei den in [PELA05, Pon08b] dokumentierten Kompressionsergebnissen das PSNR um 0.5 – 1 dB.

Das Prinzip des Entfernens von Quantisierungsrauschen durch eine DCT-basierte Filterung lässt sich auch für andere Transformationen und für die Reduktion anderer Typen von Kompressionsartefakten verwenden (begründet Fußnote 1 auf Seite 140). Dieser Nachbearbeitungsschritt wurde daher für alle untersuchten Transformationsverfahren optional auswählbar in die Rekonstruktionsalgorithmen eingebunden und steht somit den SPIHT-, L-CEB- und AGU-Kompressionsstrategien I und II zur Verfügung. Die verwendeten Tresholding-Parameter sind in der Tabelle F.1 (Anhang F.4) zusammengefasst. Abbildung 5.15 veranschaulicht die Leistungsfähigkeit der Nachbearbeitung anhand der 32×64 LCT-2. Die Anwendung der DCT-basierten Filterung beseitigt die zuvor für dieses Kompressionsbeispiel genannten sichtbaren Kompressionsartefakte fast vollständig und führt zu einer Verbesserung des PSNR auf 27.83 dB. Des Weiteren ist zu erkennen, dass die fein strukturierten Texturen ohne sichtbare Verluste erhalten bleiben und nicht der Filterung zum Opfer fallen. Auf weitere Rekonstruktionsergebnisse wird in Kapitel 6 eingegangen. Abschließend sei darauf hingewiesen, dass die DCT-basierte Filterung des Bildes zur Verbesserung der Rekonstruktionsqualität mit einem vielfach höheren Berechnungsaufwand als die Signalrekonstruktion ohne Nachbearbeitung einhergeht.

5.6 Matlab-Tools zur Kompression von Bildern

Zur Veranschaulichung und Verifikation der Leistungsfähigkeit von überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression wurden zwei Matlab-Toolboxen entwickelt und implementiert, mit denen die im Anhang H dargestellten Testbilder komprimiert werden können. Beide Toolboxen umfassen die im Rahmen dieser Arbeit untersuchten zweifach und mehrfach überlappenden Kosinustransformationen

- 8×16 LCT-2, 16×32 LCT-2 und 32×64 LCT-2, jeweils mit dem Malvar-Fenster,
- 8×24 LCT-3 und 16×48 LCT-3, jeweils mit dem Fenster 3,
- 8×32 LCT-4, 16×64 LCT-4,

¹Für eine detaillierte Beschreibung der Funktionsweise der DCT-basierten Filterung zur Reduktion von Kompressionsartefakten sei auf [EHKA99, Yar98, PELA05] verwiesen.



Abbildung 5.16: Toolbox zur Kompression von Einzelbildern, Vergleich verschiedener Transformations- und Codierungsstrategien.

- 8×48 LCT-6, 16×96 LCT-6.

Darüber hinaus beinhalten beide Toolboxes zum Vergleich der Leistungsfähigkeit mit anderen etablierten Basisdarstellungen von Bildsignalen die Transformationen

- 8×8 DCT, 16×16 DCT, 32×32 DCT,
- 8×16 TDLT¹ [TT02b],
- Wavelet-Transformation (9/7-Wavelet),
- 8×16 LOT², Typ II [Mal92],
- 8×32 GenLOT³ ($L = 4$), 8×48 GenLOT ($L = 6$) [SN96, TN],
- 8×32 ELT⁴ ($L = 4$) [Mal92].

Auf die Kompression von Bildsignalen unter Verwendung der zuletzt genannten Transformationen wird in den Abschnitten 6.7 und 6.8 näher eingegangen. Alle überlappenden Kosinustransformationen verwenden zur Verarbeitung von Signalen mit endlichem Träger die in Abschnitt 3.5 neu eingeführten Basen auf dem Intervall, die eine Konstante mit den Basisfunktionen der niedrigsten Frequenz auflösen bzw. rekonstruieren. Zur recheneffizienten Realisierung der überlappenden trigonometrischen Transformationen werden die schnellen trigonometrischen Blocktransformationen DCT-II, DCT-III und DCT-IV sowie die in Kapitel 3 hergeleiteten Klapp- und Aufklappoperatoren genutzt. Hierbei kommen die in Kapitel 4 entwickelten schnellen Algorithmen dieser Operatoren zur Anwendung.

Die in Abbildung 5.16 dargestellte Toolbox basiert auf den L-CEB- und AGU-Codieralgorithmen. In Abhängigkeit von der Blockgröße werden die quantisierten Transformationskoeffizienten entweder nach dem L-CEB-Verfahren ($B = 8$) oder nach dem AGU-Verfahren ($B = 16$ und $B = 32$) codiert und es lässt sich hierbei zwischen dem Codec-I und dem Codec-II umschalten. Wie in den vorangegangenen Abschnitten beschrieben wurde, beinhaltet die Kompressionsstrategie II eine Weiterverarbeitung der DC-Koeffizienten mit der Wavelet-Transformation. Die Kompressionsrate wird über die Wahl des Quantisierungsfaktors gesteuert. Nach der Kompression des Bildes erfolgt zur subjektiven Beurteilung der Rekonstruktionsqualität wahlweise die Anzeige des rekonstruierten Bildes oder die Visualisierung des Rekonstruktionsfehlers. Des Weiteren werden zur Bewertung der Kompressionsleistung die Bitrate und als objektives Qualitätsmaß das PSNR ausgegeben. Um einen schnellen Vergleich der Rekonstruktionsqualität verschiedener Verarbeitungsstrategien bei vorgegebenen Bitraten zu ermöglichen, wurden für alle Transformations- und Codierverfahren die Quantisierungsfaktoren zur Kompression der untersuchten Bilder bei Bitraten von 0.0625 bpp, 0.1 bpp, 0.125 bpp, 0.25 bpp, 0.5 bpp, 0.75 bpp und 1.0 bpp bestimmt und eingebunden. Zur Verbesserung der subjektiven und objektiven Rekonstruktionsqualität kann das in Abschnitt 5.5 vorgestellte Postprocessing ausgewählt werden.

Die zweite Toolbox umfasst im Wesentlichen den Funktionsumfang der ersten Toolbox. Im Unterschied zur ersten Toolbox werden die Transformationskoeffizienten jedoch nach dem SPIHT-Verfahren quantisiert und codiert. Das Kompressionsverhältnis wird hierbei ausschließlich über die Bitrate eingestellt und nicht über einen Quantisierungsfaktor.

¹TDLT: time-domain lapped transform

²LOT: lapped orthogonal transform

³GenLOT: generalized linear-phase lapped orthogonal transform

⁴ELT: extended lapped transform

Kapitel 6

Kompressionsergebnisse – Analyse und Auswertung

In diesem Kapitel wird die Leistungsfähigkeit der vorgestellten Kompressionsstrategien anhand üblicher Testbilder analysiert und mit der Kompressionsleistung anderer publizierter Verfahren verglichen. Die Untersuchungen konzentrieren sich im Wesentlichen auf den Vergleich

- *zwischen lokalen trigonometrischen Basen auf der reellen Achse unter Verwendung der Signalfortsetzung und lokalen trigonometrischen Basen auf dem Intervall unter Verwendung der neu eingeführten Randbasisfunktionen,*
- *von Kosinustransformation, zweifach überlappender Kosinustransformation und mehrfach überlappender Kosinustransformation bei verschiedener Blockgröße der Transformationskoeffizienten und verschiedenen Codierungsstrategien,*
- *von überlappender Kosinustransformation mit anderen leistungsfähigen blocküberlappenden Transformationen,*
- *von LCT-basierten Kompressionsstrategien mit waveletbasierten State-of-the-Art Kompressionsverfahren.*

Zur Beurteilung der Leistungsfähigkeit der in Kapitel 5 vorgeschlagenen Kompressionsstrategien auf Grundlage von überlappenden lokalen trigonometrischen Basen wurden umfangreiche Untersuchungen durchgeführt. Insgesamt wurden 19 Testbilder mit allen vorgestellten Kompressionsalgorithmen bei verschiedenen Bitraten codiert. Für den Vergleich mit anderen Kompressionsansätzen wurden leistungsfähige State-of-the-Art Codierungsstrategien und Transformationen ausgewählt. Um einen fairen Leistungsvergleich durchzuführen, wurden Kompressionsverfahren, die durch Beste-Basen-Algorithmen eine signalangepasste Zerlegung des Bildes vornehmen und so eine Leistungssteigerung erzielen, nicht betrachtet. Hierzu zählt z.B. der in [PLEA08] eingeführte ADCTC-Codec¹, der die Blockgröße der DCT an die Bildinhalte anpasst. Dieser Codec liefert dadurch fast durchgehend bessere Kompressionsergebnisse als JPEG2000, SPIHT und die im Rahmen dieser Arbeit vorgestellten Kompressionsstrategien.

¹ADCTC: Advanced DCT based image Coder (Der Codec ist als ausführbare Programmdatei auf der Webseite von Ponomarenko [Pon08a] erhältlich.)

Die wesentlichen Ergebnisse zur Beurteilung der Leistungsfähigkeit der in Kapitel 5 eingeführten Kompressionsstrategien sind in den Tabellen, Diagrammen und Abbildungen im Anhang H aufgeführt. Anhang H enthält zu Beginn eine Quellenangabe der verwendeten Testbilder und ist im Weiteren so aufgebaut, dass im Anhang H.1 zunächst das jeweilig untersuchte Testbild mit seinen Eckdaten vorgestellt wird. Angegeben sind die Bildgröße, die Entropie, die Varianz, der mittlere Grauwert, der minimale Grauwert und der maximale Grauwert. Es folgen für jedes Testbild sieben PSNR-Diagramme, in denen verschiedene Transformations- und Codierungsstrategien gegenübergestellt werden. Für ausgewählte Verarbeitungsstrategien sind die PSNR-Werte bei verschiedenen Bitraten in jeweils zwei Tabellen – mit und ohne Nachbearbeitungsfilter – pro Testbild angegeben. Die im Anhang H.2 dargestellten Tabellen dienen dem Vergleich der Leistungsfähigkeit ausgewählter Kompressionsverfahren und umfassen jeweils alle untersuchten Testbilder. Schließlich befinden sich im Anhang H.3 Beispiele von rekonstruierten Bildern, die eine subjektive Beurteilung der Rekonstruktionsqualität ausgewählter Kompressionsstrategien ermöglichen. Die nachfolgenden Abschnitte geben eine detaillierte Analyse und Auswertung der Leistungsfähigkeit einzelner Transformations- und Codierungsstrategien.

6.1 Kompressionsstrategien I und II im Vergleich

In Kapitel 5 wurden bereits wichtige Kompressionsergebnisse am Beispiel des Testbildes „Barbara“ vorgestellt, miteinander verglichen und diskutiert. Für einen vollständigen Vergleich sind im Anhang H.3.1 für alle untersuchten Transformations- und Codierungsstrategien die Kompressionsergebnisse für das Testbild „Barbara“ bei einer Bitrate von 0.1 bpp dokumentiert. Es ist zu erkennen, dass bei fast allen Kombinationen von Transformationen und Codieransätzen (SPIHT, L-CEB und AGU) die Kompressionsstrategie II die bessere subjektive und objektive Rekonstruktionsqualität liefert. Dieses Resultat lässt sich mit Hilfe der in Abschnitt 5.6 entwickelten Matlab-Toolboxen auch für die übrigen Testbilder feststellen. Durch die Weiterverarbeitung der DC-Transformationskoeffizienten mit der Wavelet-Transformation werden die sichtbaren Kompressionsartefakte in den tieffrequenten Bildbereichen deutlich reduziert (vgl. Abschnitte 5.2.3 und 5.3.2). Anhang H.3.2 zeigt am Beispiel der Testbilder „Lena“, „Zelda“ und „Clown“ weitere Rekonstruktionsergebnisse zur subjektiven und objektiven Beurteilung der Rekonstruktionsqualität. Als Transformationen kommen die 8×8 DCT, 8×16 LCT-2, 8×32 LCT-4 und die 9/7-DWT zur Anwendung und es wird die L-CEB-Codierungsstrategie genutzt. Bei allen dargestellten Kompressionsbeispielen liefert die Kompressionsstrategie II die besseren Rekonstruktionsergebnisse.

Die Ausführungen in den nachfolgenden Abschnitten konzentrieren sich auf die Kompressionsstrategie II der untersuchten Transformations- und Codiervverfahren.

6.2 Basen auf der reellen Achse mit Signalfortsetzung im Vergleich zu Basen auf dem Intervall

In der Einführung zu Kapitel 3 wurde erläutert, dass die Repräsentation von Signalen, die nur auf einem beschränkten Intervall definiert sind, mit überlappenden lokalen trigonometrischen Basen, die auf der reellen Achse definiert sind, zu Problemen an den Signalrändern führt. Die überlappende Kosinustransformation benötigt Signalwerte, die außerhalb des beschränkten Signalintervalls liegen und somit nicht definiert sind. Als mögliche Lösungsansätze wurden

zum einen die Fortsetzung des Signals auf die gesamte reelle Achse und zum anderen die Entwicklung von Basen auf dem Intervall vorgeschlagen. Abschnitt 3.1 kam zu dem Resultat, dass sich die Verfahren zur Signalfortsetzung nicht für die effiziente Darstellung mit überlappenden Kosinus-IV-Basen eignen, da u.a. die verlustbehaftete Kompression des periodisierten Signals zu störenden Randartefakten führen kann. Abbildung 6.1 zeigt am Beispiel der Vierfachüberlappung die Kompression des Testbildes „Roof“ unter Verwendung der Signalfortsetzung. Bei einer Bitrate von 0.125 bpp sind im Zusammenhang mit der Signalfortsetzung deutlich sichtbare Kompressionsartefakte an den Bildrändern zu erkennen.

In Abschnitt 3.5 wurden daher Fensterfunktionen zur Konstruktion von lokalen trigonometrischen Basen auf dem Intervall eingeführt. Die so konstruierten Basen erfüllen die beiden für die Bilddatenkompression grundlegenden Eigenschaften der Riesz-Stabilität sowie der Auflösung bzw. Rekonstruktion einer Konstanten mit den Basisfunktionen der niedrigsten Frequenz. Durch die neu entwickelten Kosinus-IV-Kosinus-II-Basen auf dem Intervall ist ein Fortsetzen des Signals nicht erforderlich und es entstehen daher keine sichtbaren Randartefakte wie bei der Signalfortsetzung. Dieses Ergebnis wird ebenfalls in Abbildung 6.1 anhand der Vierfachüberlappung dargestellt. Neben der Verbesserung der subjektiven Rekonstruktionsqualität erhöht sich bei dem Kompressionsbeispiel „Roof“ das PSNR um 0.42 dB.

Da sich das PSNR auf das gesamte Bild bezieht, stellt es kein günstiges objektives Maß zur Beurteilung der Leistungsfähigkeit der Randbasisfunktionen dar. Aus diesem Grunde wird im Folgenden mit dem Rand-Spitzen-Signal-Rausch-Verhältnis R_n -PSNR ein geeigneteres Maß eingeführt, das ausschließlich die Rekonstruktionsqualität der Bildränder beziffert. Für eine Breite des Bildrandes von jeweils n Zeilen und n Spalten ist das R_n -PSNR durch

$$R_n\text{-PSNR} := 10 \cdot \log_{10} \left(\frac{255^2}{\text{MSE}_n} \right) \text{ dB} \quad (6.1)$$

mit

$$\text{MSE}_n := \frac{1}{2n(M+N-2n)} \cdot \left(\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (g_{x,y} - \hat{g}_{x,y})^2 - \sum_{x=n}^{M-1-n} \sum_{y=n}^{N-1-n} (g_{x,y} - \hat{g}_{x,y})^2 \right) \quad (6.2)$$

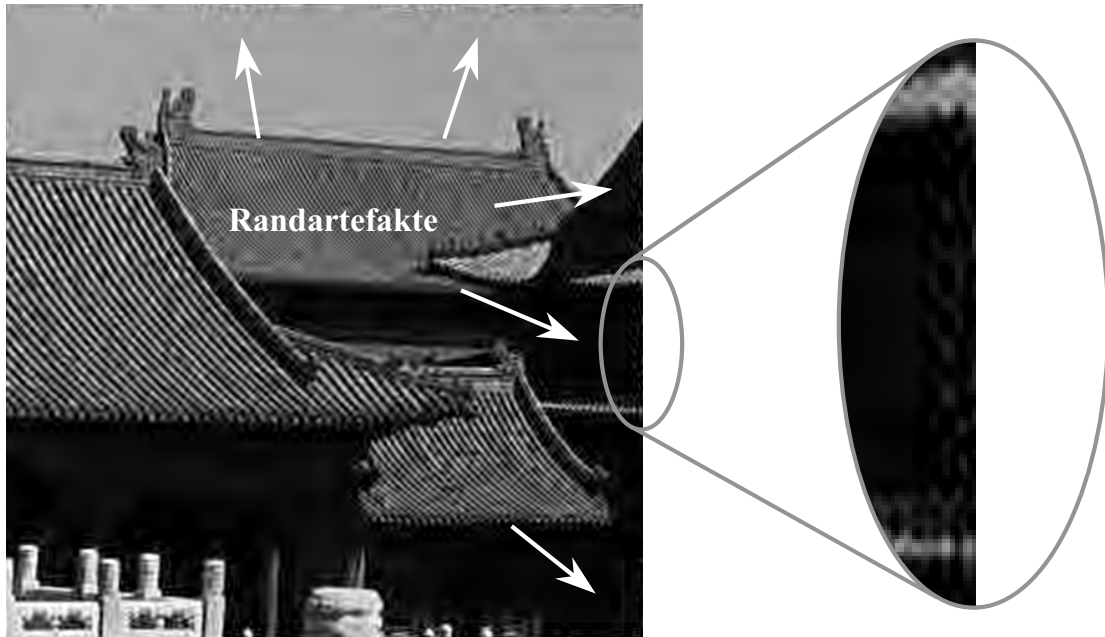
definiert. Betrachtet man in Abbildung 6.1 die Kompression des Bildes „Roof“ unter Verwendung von Basen auf der reellen Achse und periodischer Signalfortsetzung, so ist zu erkennen, dass durch die zusätzlichen Rekonstruktionsfehler, die aufgrund dieser Verarbeitungsstrategie hervorgerufen werden, das R_n -PSNR am Bildrand für kleine Parameter n einen deutlich schlechteren Wert aufweist als das PSNR über das gesamte Bild (z.B. R_8 -PSNR = 24.39 dB, während PSNR = 25.48 dB). Durch die Verwendung der neu eingeführten Randbasisfunktionen ist eine signifikante Steigerung des R_n -PSNR zu verzeichnen. So erhöht sich zum Beispiel das R_8 -PSNR um 3.23 dB auf 27.62 dB. Ein weiteres Kompressionsbeispiel, bei dem ebenfalls eine deutlich sichtbare Reduktion der Randartefakte durch die Nutzung von Randbasisfunktionen erzielt wird, ist in Abbildung H.329 (Anhang H.3.5) dargestellt.

Auch für die Zweifach-, Dreifach- und Sechsfachüberlappung lässt sich zeigen, dass mit den neu entwickelten Randbasisfunktionen im Vergleich zur Signalfortsetzung und Verwendung von Basisfunktionen auf der reellen Achse eine effizientere Repräsentation von Bildsignalen erzielt wird und keine sichtbaren Randartefakte entstehen. Die in Kapitel 3 entwickelten Kosinus-IV-Kosinus-II-Basen auf dem Intervall stellen somit im Hinblick auf die Zielsetzung der effizienten Anwendung von zweifach und mehrfach überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression ein wesentliches Resultat dieser Arbeit dar.

Verwendung von vierfach überlappenden Kosinus-IV-Basen auf der reellen Achse:

Das Signal wird durch Periodisierung fortgesetzt. Es entstehen sichtbare Randartefakte.

PSNR = 25.48 dB, R_8 -PSNR = 24.39 dB, R_{12} -PSNR = 24.91 dB, R_{16} -PSNR = 25.37 dB.

**Verwendung von vierfach überlappenden Kosinus-IV-Kosinus-II-Basen auf dem Intervall:**

Eine Signalfortsetzung ist nicht erforderlich. Es entstehen keine Randartefakte.

PSNR = 25.90 dB, R_8 -PSNR = 27.62 dB, R_{12} -PSNR = 27.21 dB, R_{16} -PSNR = 27.32 dB.

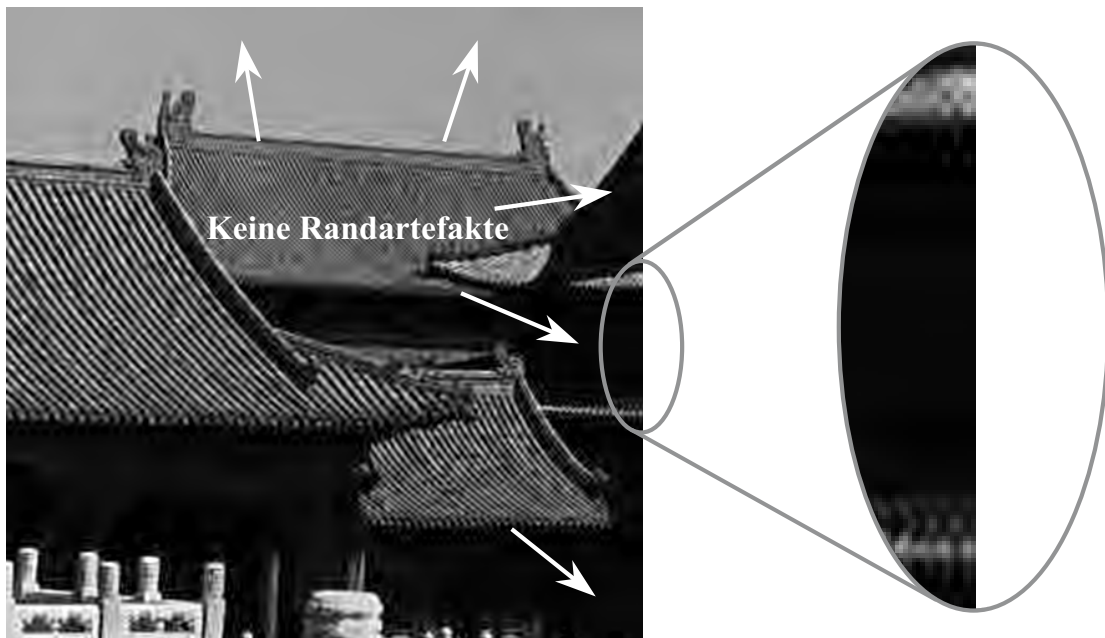


Abbildung 6.1: L-CEB-Codec-II: Kompression des Bildes „Roof“ bei einer Bitrate von 0.125 bpp; oben: Nutzung von Basisfunktionen, die auf der reellen Achse definiert sind; unten: Nutzung von Basisfunktionen, die auf dem Intervall definiert sind.

6.3 Kosinustransformation

Das im Anhang H.1 zu jedem Testbild angegebene Diagramm 1 vergleicht die Kompressionsstrategien

- L-CEB-Codec-II, 8×8 DCT,
- SPIHT-Codec-II, 8×8 DCT,
- AGU-Codec-II, 16×16 DCT,
- SPIHT-Codec-II, 16×16 DCT,
- AGU-Codec-II, 32×32 DCT und
- SPIHT-Codec-II, 32×32 DCT

miteinander. Anhand der SPIHT-Codierung ist zu erkennen, dass die Verwendung der 16×16 DCT im Vergleich zur 8×8 DCT bei fast allen untersuchten Testbildern zu signifikant besseren Kompressionsergebnissen führt. In Abhängigkeit vom Testbild und der verwendeten Bitrate erhöht sich das PSNR zumeist um ca. 0.3 – 1.7 dB (vgl. Tabelle H.39, Anhang H.2.1). Nur bei den Testbildern „Houses“ und „Peppers“ verschlechtert sich das PSNR geringfügig, wenn sie mit höheren Bitraten komprimiert werden. Eine weitere Erhöhung der Blockgröße auf 32×32 führt bei vielen Testbild-Bitraten-Kombinationen nochmals zu einer teilweise signifikanten Verbesserung der Rekonstruktionsqualität (vgl. Tabelle H.40). Einige Testbilder lassen sich jedoch bei höheren Bitraten geringfügig schlechter komprimieren, wenn anstatt der 16×16 DCT die 32×32 DCT Anwendung findet. Man kommt für die untersuchten Testbilder zu dem Resümee, dass im SPIHT-Codec-II die 32×32 DCT im Vergleich zur 16×16 DCT und zur 8×8 DCT im Allgemeinen die beste Rekonstruktionsqualität liefert.

Die L-CEB-Codierung führt bei der 8×8 DCT im Vergleich zur SPIHT-Codierung bei einem Großteil der untersuchten Testbild-Bitraten-Kombinationen zu einer Steigerung der Rekonstruktionsqualität. Die Verbesserung des PSNR liegt jedoch meistens nur im Bereich von ca. 0.1 – 0.3 dB (vgl. Tabelle H.41). Einen äquivalenten Leistungszuwachs erhält man auch für die DCT-Blockgrößen 16×16 und 32×32 beim Vergleich zwischen SPIHT- und AGU-Codierung (vgl. Tabellen H.42 und H.43). Die AGU- und die L-CEB-Codierung nutzen somit die Struktur der DCT-Koeffizienten effizienter aus. Da jedoch die Erhöhung der objektiven Bildqualität im Vergleich zur SPIHT-Codierung für die jeweils betrachtete Blockgröße überwiegend relativ gering ausfällt, sind in den rekonstruierten Bildern visuell nur geringfügige Unterschiede zu erkennen (vgl. Abbildung 5.8 und 5.11 bzgl. der 8×8 DCT sowie Abbildung 5.14 bzgl. der 32×32 DCT).

6.4 Zweifach überlappende Kosinustransformation

Im Diagramm 2, das für jedes untersuchte Testbild im Anhang H.1 angegeben ist, sind die PSNR-Kurven für die Transformationscodecs

- L-CEB-Codec-II, 8×16 LCT-2,
- SPIHT-Codec-II, 8×16 LCT-2,
- AGU-Codec-II, 16×32 LCT-2,
- SPIHT-Codec-II, 16×32 LCT-2,

- AGU-Codec-II, 32×64 LCT-2 und
- SPIHT-Codec-II, 32×64 LCT-2

dargestellt. In Analogie zu den Auswertungen für die DCT wird auch bei der Zweifachüberlappung zunächst die Codierung von verschiedenen großen Transformationskoeffizientenblöcken unter Verwendung einer einheitlichen Codierungsstrategie, der SPIHT-Codierung, untersucht. Die Nutzung der 16×32 LCT-2 führt im Vergleich zur 8×16 LCT-2 bei fast allen Testbild-Bitraten-Kombinationen zu einer Verbesserung der objektiven Rekonstruktionsqualität. Besonders deutlich fällt die Steigerung des PSNR mit bis zu 1.17 dB bei den texturreichen Testbildern „Barbara“, „Roof“, „Fingerprint“, „Wood Grain“ und „Water“ aus (vgl. Tabelle H.44, Anhang H.2.2). In Abhängigkeit von der Bitrate werden auch bei den Testbildern „Lena“, „Zelda“, „Lighthouse“ und „Clown“ signifikant bessere Kompressionsergebnisse erzielt. Die sichtbaren Kompressionsartefakte in Form von schnellen Grauwertwechsel, die bei der Zweifachüberlappung bei der Rekonstruktion von tieffrequenten Bildinhalten auftreten können, wirken bei der 16×32 LCT-2 weniger störend als bei der 8×16 LCT-2 (vgl. Abbildungen H.309 und H.319).

Eine weitere Erhöhung der Blockgröße wie bei der 32×64 LCT-2 führt im Vergleich zur 16×32 LCT-2 nur bei den reinen Texturbildern „Fingerprint“, „Wood Grain“ und „Water“ zu einer signifikanten Verbesserung der PSNR-Werte (vgl. Tabelle H.45). Bei den meisten Testbild-Bitraten-Kombinationen verschlechtert sich jedoch die Kompressionseffizienz und in den rekonstruierten Bildern kann es aufgrund der längeren Träger der Basisfunktionen zu deutlich sichtbaren Ringing-Artefakten kommen. Insgesamt ist bei der Zweifachüberlappung die 16×32 LCT-2 bzgl. der objektiven und subjektiven Rekonstruktionsqualität als am leistungsfähigsten einzuschätzen.

In Analogie zu den Resultaten für die DCT (vgl. Abschnitt 6.3) ist auch bei der zweifach überlappenden Kosinustransformation anhand der Kompressionsergebnisse für die AGU- bzw. L-CEB-Codierung im Vergleich zur SPIHT-Codierung zu erkennen, dass eine an die Blockstruktur der Transformationskoeffizienten angepasste Codierungsstrategie bei fast allen untersuchten Testbild-Bitraten-Kombinationen zu einer geringfügigen Verbesserung der objektiven Rekonstruktionsqualität von ca. 0.1 – 0.3 dB führt (vgl. Tabellen H.46 und H.47). Lediglich bei den texturreichen Testbildern „Barbara“ und „Roof“ fällt mit bis zu 0.67 dB die Erhöhung des PSNR etwas deutlicher aus (vgl. Tabelle H.46). Die höhere objektive Rekonstruktionsqualität, die durch die AGU- bzw. die L-CEB-Codierung im Vergleich zur SPIHT-Codierung erzielt wird, macht sich jedoch bei den meisten der untersuchten Testbilder visuell kaum bemerkbar. Eine Ausnahme bildet z.B. die L-CEB-II-Codierung des Testbildes „Barbara“ unter Verwendung der 8×16 LCT-2. Im Vergleich zur SPIHT-II-Codierung wird eine sichtbar bessere Rekonstruktion der Textur der Tischdecke und des Kopftuches erzielt (vgl. Abbildungen 5.8 und 5.11)

6.5 Mehrfach überlappende Kosinustransformation

Ab einem Überlappungsfaktor von 3 und aufgrund der Weiterverarbeitung der DC-Transformationskoeffizienten in der Kompressionsstrategie II gelingt es erstmals, dass bei der überlappenden Kosinustransformation unter Nutzung der üblicherweise verwendeten Blockgröße von 8×8 Transformationskoeffizienten in den rekonstruierten Bildern keine sichtbaren Blockartefakte mehr auftreten (vgl. Rekonstruktionsergebnisse für die 8×8 DCT, 8×16 LCT-2, 8×24 LCT-3, 8×32 LCT-4 und 8×48 LCT-6 im Anhang H.3.3 als auch Rekonstruktionsergebnisse im Anhang H.3.2).

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
L-CEB-Codec-II, 8×24 LCT-3/ $w^{(1)}$ <i>con.</i>	23.88	26.42	29.50	33.28	35.75	37.60
L-CEB-Codec-II, 8×24 LCT-3/ $w^{(1)}$ <i>lin.</i>	23.81	26.34	29.39	33.16	35.63	37.47
L-CEB-Codec-II, 8×24 LCT-3/ $w^{(2)}$ <i>con.</i>	23.58	26.01	29.03	32.74	35.19	37.08
L-CEB-Codec-II, 8×24 LCT-3/ $w^{(2)}$ <i>lin.</i>	23.51	25.96	28.93	32.62	35.07	36.94
L-CEB-Codec-II, 8×24 LCT-3/ $w^{(3)}$ <i>con.</i>	23.99	26.55	29.65	33.44	35.91	37.77
L-CEB-Codec-II, 8×24 LCT-3/ $w^{(3)}$ <i>lin.</i>	23.91	26.50	29.53	33.31	35.79	37.63

Tabelle 6.1: Dreifachüberlappung: Kompressionsergebnisse für das Testbild „Barbara“. Vergleich der verschiedenen dreifach überlappenden Fensterfunktionen aus Abschnitt 2.3.2 unter Verwendung der Randfenster aus Abschnitt 3.5.4 (*con.* : Basen können eine Konstante mit den Basisfunktionen der niedrigsten Frequenz rekonstruieren,) und Abschnitt 3.5.5 (*lin.* : Basen können eine Lineare mit den Basisfunktionen der niedrigsten Frequenz rekonstruieren).

In Abschnitt 2.3.2 wurden drei verschiedene dreifach überlappende Fensterfunktionen beschrieben. Die mit der Fensterfunktion $w^{(3)}$ erzeugten trigonometrischen Basen weisen im Vergleich zu den Basen, die mit den Fensterfunktionen $w^{(2)}$ und $w^{(1)}$ konstruiert werden, eine bessere Riesz-Stabilität auf. Wie in Tabelle 6.1 am Beispiel des Testbildes „Barbara“ gezeigt wird, liefert die Verwendung der Fensterfunktion $w^{(3)}$ im Vergleich zu den Fensterfunktionen $w^{(2)}$ und $w^{(1)}$ auch die besseren Kompressionsergebnisse.

In den Abschnitten 3.5.2 und 3.5.3 wurden für die Dreifachüberlappung zwei Lösungsansätze zur Konstruktion von Randbasisfunktionen vorgestellt. Die nach dem Ansatz 1 bestimmten Basen auf dem Intervall besitzen die Fähigkeit, eine Konstante mit den Basisfunktionen der niedrigsten Frequenz zu reproduzieren. Beim Ansatz 2 lassen sich darüber hinaus auch lineare Funktionen mit den Basisfunktionen der niedrigsten Frequenz reproduzieren. Die Basen auf dem Intervall nach Lösungsansatz 1 weisen jedoch die günstigeren Riesz-Schranken auf und die bessere Riesz-Stabilität spiegelt sich schließlich auch in den Kompressionsergebnissen wider. Tabelle 6.1 zeigt am Beispiel des Testbildes „Barbara“, dass die Verwendung der Randbasisfunktionen nach Ansatz 1 im Vergleich zu den Randbasisfunktionen nach Ansatz 2 die höheren PSNR-Werte liefert. Die Verbesserung der polynomialen Reproduktionseigenschaften der Randbasisfunktionen führt somit bei den untersuchten Testbildern zu keiner Steigerung der Kompressionsleistung. Alle im Anhang H angegebenen Kompressionsergebnisse zur Beurteilung der Leistungsfähigkeit der dreifach überlappenden Kosinustransformation wurden unter Verwendung der inneren Fensterfunktion $w^{(3)}$ und der Randfenster nach Konstruktionsansatz 1 bestimmt.

Für die Vierfachüberlappung wurden in den Abschnitten 3.5.4 und 3.5.5 ebenfalls zwei Lösungsansätze zur Konstruktion von Randbasisfunktionen vorgestellt. Wie in der Tabelle 6.2 am Beispiel des Testbildes „Barbara“ gezeigt wird, erhält man für beide Lösungsansätze fast übereinstimmende PSNR-Werte. Beide Verfahren eignen sich somit gleichermaßen zur Konstruktion von leistungsfähigen Randbasisfunktionen für den Einsatz in der Bilddatenkompression. Alle im Anhang H angegebenen Kompressionsergebnisse zur Beurteilung der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen wurden unter Verwendung von Konstruktionsansatz 1 gewonnen.

Da für die untersuchten trigonometrischen Transformationen die L-CEB- bzw. die AGU-Codierung im Vergleich zur SPIHT-Codierung bei den betrachteten Testbildern mehrheitlich

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
L-CEB-Codec-II, 8×24 LCT-4 <i>Ansatz 1</i>	24.50	27.05	30.26	34.14	36.63	38.49
L-CEB-Codec-II, 8×24 LCT-4 <i>Ansatz 2</i>	24.50	27.04	30.25	34.12	36.61	38.47

Tabelle 6.2: Vierfachüberlappung: Kompressionsergebnisse für das Testbild „Barbara“. Vergleich der Randbasisfunktionen nach Konstruktionsansatz 1 (vgl. Abschnitt 3.5.2) und Konstruktionsansatz 2 (vgl. Abschnitt 3.5.3).

zu besseren Kompressionsergebnissen führen (vgl. Tabellen H.1 – H.38, Anhang H.1), konzentrieren sich die nachfolgenden Auswertungen zur Beurteilung der Leistungsfähigkeit der Transformationen im Wesentlichen auf die AGU- bzw. die L-CEB-Codierung der Transformationskoeffizienten. Das im Anhang H.1 für jedes Testbild dokumentierte Diagramm 3 stellt die PSNR-Kurven der Transformationscodecs

- L-CEB-Codec-II, 8×24 LCT-3,
- AGU-Codec-II, 16×48 LCT-3,
- L-CEB-Codec-II, 8×32 LCT-4,
- AGU-Codec-II, 16×64 LCT-4,
- L-CEB-Codec-II, 8×48 LCT-6 und
- AGU-Codec-II, 16×96 LCT-6

gegenüber. Im Vergleich zur dreifach überlappenden 8×24 LCT-3 ist durch die Nutzung der vierfach überlappenden 8×32 LCT-4 bei allen untersuchten Testbildern eine signifikante Steigerung der objektiven Rekonstruktionsqualität zu verzeichnen. Das PSNR erhöht sich in Abhängigkeit vom Testbild und der gewählten Bitrate um bis zu 0.88 dB (vgl. Tabelle H.48, Anhang H.2.3). Für den Großteil der Testbilder liegt der PSNR-Zuwachs im Mittel bei ca. 0.5 dB. Bei dieser Steigerung der objektiven Bildqualität sind jedoch in den rekonstruierten Bildern visuell nur geringfügige qualitative Unterschiede zu erkennen. Harmonische Texturen stellen die dreifach und die vierfach überlappenden lokalen trigonometrischen Basen annähernd gleich gut dar (vgl. Abbildung H.316). Die rekonstruierten Bilder unterscheiden sich zumeist nur im Auftreten von Ringing-Artefakten. So können z.B. bei gleichem Testbild und gleicher Bitrate an einer Kante bei der 8×32 LCT-4 stärkere Ringing-Artefakte auftreten als bei der 8×24 LCT-3, während hingegen an einer anderen Kante die 8×24 LCT-3 stärkere Ringing-Artefakte verursacht als die 8×32 LCT-4. Tendenziell sind jedoch mit der Verbesserung der objektiven Rekonstruktionsqualität durch die Nutzung der 8×32 LCT-4 die Ringing-Artefakte weniger stark ausgeprägt als bei der 8×24 LCT-3 (vgl. Abbildung H.310: Ringing-Artefakte beim Übergang vom Hintergrund zum Hut, Abbildung H.314: Ringing-Artefakte beim Übergang vom Himmel zum Dach des Hauses am Leuchtturm).

Die Anwendung der sechsfach überlappenden 8×48 LCT-6 führt zu keiner weiteren Steigerung der Kompressionsleistung. Im Vergleich zur 8×32 LCT-4 erhält man bei allen untersuchten Testbildern schlechtere PSNR-Werte (vgl. Tabelle H.49).

Eine Erhöhung der Blockgröße bei der Dreifach- und der Vierfachüberlappung führt bei der Mehrheit der untersuchten Testbild-Bitraten-Kombinationen ebenfalls zu keiner Steigerung der Kompressionsleistung. Ausnahmen bilden u.a. die Testbilder „Roof“, „Fingerprint“, „Wood

Grain“ und „Water“, bei denen sich durch die Anwendung der 16×48 LCT-3 bzw. der 16×64 LCT-4 die PSNR-Werte im Vergleich zur 8×24 LCT-3 bzw. zur 8×32 LCT-4 zum Teil erheblich verbessern (vgl. Tabellen H.50 und H.51).

In der Summe ist bei der Mehrfachüberlappung die 8×32 LCT-4 als am leistungsfähigsten einzuschätzen. Eine weitere Erhöhung des Überlappungsfaktors oder die Verwendung von größeren Koeffizientenblöcken ist nicht empfehlenswert, da es bei niedrigen und mittleren Bitraten aufgrund der längeren Träger der Basisfunktionen zu ausgeprägteren sichtbaren Ringing-Artefakten in den rekonstruierten Bildern kommen kann.

6.6 Kosinus- und überlappende Kosinustransformation im Vergleich

In diesem Abschnitt wird die Leistungsfähigkeit der Kosinustransformation, der zweifach überlappenden Kosinustransformation und der mehrfach überlappenden Kosinustransformation unter zwei verschiedenen Rahmenbedingungen miteinander verglichen. Zunächst erfolgt die Analyse und Auswertung der Kompressionsergebnisse für die verschiedenen Transformationen bei gleicher Blockgröße der Transformationskoeffizienten von $B = 8$ und unterschiedlicher Länge der Träger der Basisfunktionen. Dieser Vergleich korreliert insbesondere mit einer der ursprünglichen Zielstellungen für die Untersuchung von überlappenden lokalen trigonometrischen Basen, und zwar zur Vermeidung von Blockartefakten und zur Verbesserung der Kompressionsleistung die konventionelle Kosinustransformation im JPEG-Verfahren durch leistungsfähige überlappende Kosinustransformationen zu ersetzen [Mal00, Bit00b]. Als gemeinsame Codierungsstrategie kommt für den Vergleich der verschiedenen Transformationen im Gegensatz zu den Untersuchungen in [Mal00, Bit00b] jedoch nicht mehr die ineffiziente JPEG-Codierung der quantisierten Transformationskoeffizienten zum Einsatz, sondern es wird die deutlich effizientere L-CEB-II-Codierung verwendet.

Anschließend wird unter Nutzung der SPIHT-II-Codierung die Leistungsfähigkeit der Transformationen bei gleicher Trägerlänge der trigonometrischen Basisfunktionen von $l = 32$ Stützstellen und unterschiedlicher Blockgröße der Transformationskoeffizienten untersucht.

Vergleich bei gleicher Blockgröße der Transformationskoeffizienten

Im Diagramm 4, das im Anhang H.1 für jedes der untersuchten Testbilder angegeben ist, werden die Transformationscodecs

- L-CEB-Codec-II, 8×8 DCT,
- L-CEB-Codec-II pp, 8×8 DCT,
- L-CEB-Codec-II, 8×16 LCT-2,
- L-CEB-Codec-II pp, 8×16 LCT-2,
- L-CEB-Codec-II, 8×32 LCT-4 und
- L-CEB-Codec-II pp, 8×32 LCT-4

gegenübergestellt. Bei allen untersuchten Testbildern führt die Substitution der 8×8 DCT im L-CEB-Codec-II durch die 8×16 LCT-2 zu einer signifikanten Steigerung der Kompressionsleistung. Wie die in Tabelle H.53 (Anhang H.2.4) dokumentierten Ergebnisse zeigen, erhöht sich in Abhängigkeit vom untersuchten Testbild und von der untersuchten Bitrate das PSNR

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	1.63	2.12	2.60	2.57	2.27	1.95
„Lena“	1.29	1.37	1.32	0.98	0.71	0.54
„Zelda“	2.27	2.18	1.62	0.94	0.80	0.81
„Boat“	0.68	0.63	0.69	0.64	0.65	0.56
„Goldhill“	0.67	0.63	0.65	0.57	0.53	0.47
„Lighthouse“	0.75	0.72	0.57	0.50	0.46	0.45
„Roof“	1.56	1.96	2.10	2.06	1.84	1.73
„Clown“	0.89	1.33	1.27	1.04	0.77	0.64
„Houses“	0.43	0.36	0.40	0.29	0.17	0.10
„Mandrill“	0.26	0.33	0.38	0.43	0.44	0.42
„Fingerprint“	1.28	1.83	2.08	2.23	2.30	2.40
„Peppers“	1.27	1.20	0.78	0.31	0.19	0.17
„F16“	0.88	0.78	0.76	0.63	0.44	0.32
„Bubble“	0.61	0.74	0.79	0.73	0.65	0.59
„Bark“	0.49	0.61	0.70	0.74	0.73	0.73
„Wood Grain“	0.22	0.35	0.72	0.87	0.78	0.65
„Sand“	0.31	0.46	0.62	0.61	0.61	0.60
„Water“	0.55	1.15	1.59	1.31	0.98	0.80
„Texture“	0.41	0.48	0.51	0.50	0.47	0.44

Tabelle 6.3: Leistungsfähigkeit der Kosinustransformation und der vierfach überlappenden Kosinustransformation im Vergleich:

$$\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II, } 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{L-CEB-Codec-II, } 8 \times 8 \text{ DCT}},$$

ΔPSNR in dB.

um bis zu 1.96 dB. Bei fast allen Testbild-Bitraten-Kombinationen wird eine Steigerung der objektiven Rekonstruktionsqualität von über 0.5 dB erzielt.

Durch die Anwendung der 8×32 LCT-4 ist im Vergleich zur 8×16 LCT-2 nochmals bei einem Großteil der Testbild-Bitraten-Kombinationen eine Steigerung der Kompressionsleistung zu verzeichnen. Am deutlichsten fällt die Erhöhung des PSNR um bis zu 1.03 dB beim Testbild „Zelda“ aus (vgl. Tabelle H.54). Bei den meisten Testbildern liegt die Verbesserung des PSNR jedoch unter 0.5 dB. Für höhere Bitraten verringert sich hingegen das PSNR bei einigen Testbildern geringfügig um bis zu 0.25 dB. Im Vergleich zur 8×8 DCT liefert die 8×32 LCT-4 bei allen untersuchten Testbildern und Bitraten signifikant höhere PSNR-Werte. Äußerst prägnant fällt die Steigerung des PSNR um bis zu 2.6 dB beim Testbild „Barbara“ aus (vgl. Tabelle 6.3).

Ein entscheidender Vorteil der vierfach überlappenden Kosinustransformation liegt in der Reduktion von Blockartefakten. Im Vergleich zur 8×8 DCT und zur 8×16 LCT-2 sind bei der 8×32 LCT-4 in den rekonstruierten Bildern keine Blockartefakte und keine Kompressionsartefakte in Form von schnellen Grauwertwechseln in der Umgebung von Blockgrenzen sichtbar (vgl. Rekonstruktionsergebnisse zu den Codecs-II im Anhang H.3.2 und H.3.3). Die Verwendung der 8×32 LCT-4 liefert im Vergleich zur 8×8 DCT und zur 8×16 LCT-2 sowohl die beste subjektive als auch die beste objektive Rekonstruktionsqualität. Durch das in Abschnitt 5.5 dargestellte Postprocessing-Verfahren zur Reduktion von Kompressionsartefakten lassen sich die Blockeffekte bei der 8×8 DCT und die Kompressionsartefakte in Form von schnellen Grauwertwechseln bei der 8×16 LCT-2 zwar deutlich verringern, dieser rechenaufwendige Nachbearbeitungsschritt führt jedoch auch bei der Vierfachüberlappung zu einer Verbesserung der objektiven und subjektiven Rekonstruktionsqualität, da Ringing-Artefakte

ebenfalls sehr wirksam reduziert werden (vgl. jeweils Diagramm 4 und Tabellen im Anhang H.1 sowie Rekonstruktionsbeispiele für die 8×32 LCT-4 im Anhang H.3.4).

Vergleich bei gleicher Länge der Träger der trigonometrischen Basisfunktionen

Die Untersuchungen in den vorangegangenen Abschnitten führten u.a. zu folgenden Resultaten:

- Beim Vergleich der Block-Kosinustransformation für verschiedene Blockgrößen werden die besten Kompressionsergebnisse mit der 32×32 DCT erzielt (vgl. Abschnitt 6.3).
- Bei der zweifach überlappenden Kosinustransformation erweist sich die 16×32 LCT-2 als am leistungsfähigsten (vgl. Abschnitt 6.4).
- Bei der Mehrfachüberlappung liefert die vierfach überlappende 8×32 LCT-4 die beste Rekonstruktionsqualität (vgl. Abschnitt 6.5).

Die Transformationen 32×32 DCT, 16×32 LCT-2 und 8×32 LCT-4 besitzen die Gemeinsamkeit, dass die Basisfunktionen mit 32 diskreten Abtastwerten jeweils die gleiche Anzahl an Stützstellen aufweisen. In diesem Abschnitt wird die Leistungsfähigkeit dieser drei Transformationen miteinander verglichen. Diagramm 5 stellt im Anhang H.1 für jedes untersuchte Testbild die PSNR-Kurven der Kompressionsstrategien

- SPIHT-Codec-II, 32×32 DCT,
- SPIHT-Codec-II, 16×32 LCT-2 und
- SPIHT-Codec-II, 8×32 LCT-4

gegenüber¹. Die Anwendung der 16×32 LCT-2 liefert im Vergleich zur 32×32 DCT bei fast allen untersuchten Testbildern bessere Kompressionsergebnisse. Besonders deutlich fällt die Erhöhung der objektiven Rekonstruktionsqualität mit bis zu 1.08 dB bei den Testbildern „Barbara“ und „Roof“ aus (vgl. Tabelle H.55). Bei den übrigen Testbildern bleibt hingegen die Verbesserung des PSNR überwiegend unter 0.5 dB. Nur bei den Testbildern „Wood Grain“ und „Water“ verschlechtert sich das PSNR bei niedrigen Bitraten geringfügig, wenn anstatt der 32×32 DCT die 16×32 LCT-2 genutzt wird.

Die Anwendung der 8×32 LCT-4 führt im Vergleich zur 32×32 DCT ebenfalls bei der Mehrheit der Testbild-Bitraten-Kombinationen zu einer Verbesserung der PSNR-Werte (vgl. Tabelle H.56). Nur die Testbilder „Fingerprint“, „Wood Grain“ und „Water“ lassen sich bzgl. der objektiven Rekonstruktionsqualität ausnahmslos effizienter durch die 32×32 DCT repräsentieren. Im Vergleich zur 16×32 LCT-2 liefert die 8×32 LCT-4 hingegen bei den meisten Testbildern mit einer Differenz von bis zu -0.3 dB geringfügig schlechtere PSNR-Werte (vgl. Tabelle H.57). Speziell die Testbilder „Roof“ und „Wood Grain“ können bzgl. der objektiven Rekonstruktionsqualität wesentlich effizienter durch die 16×32 LCT-2 repräsentiert werden. Lediglich bei den Testbildern „Lena“, „Zelda“ und „Peppers“ verbessert sich im Vergleich zur 16×32 LCT-2 die objektive Rekonstruktionsqualität geringfügig, wenn die 8×32 LCT-4 zur Anwendung kommt.

Im Allgemeinen korreliert die Bewertung der Rekonstruktionsqualität durch das PSNR sehr stark mit der subjektiven Beurteilung der Qualität des rekonstruierten Bildes [Str09]. Bei

¹Des Weiteren ist im Diagramm 5 für jedes untersuchte Testbild die PSNR-Kurve der waveletbasierten SPIHT-Codierung (SPIHT-Codec-II, 9/7-DWT) dargestellt. Zum Vergleich der Leistungsfähigkeit von trigonometrischen Basen mit Wavelets wird auf Abschnitt 6.8 verwiesen.

dem Vergleich der 32×32 DCT, der 16×32 LCT-2 und der 8×32 LCT-4 weicht jedoch das subjektive Empfinden der Rekonstruktionsqualität von der objektiven Bewertung durch das PSNR ab. Auch wenn die PSNR-Werte bei der Verwendung dieser drei Transformationen auf einem ähnlich hohen Niveau angesiedelt sind, so wird subjektiv beurteilt bei der Mehrheit der untersuchten Testbilder die beste visuelle Rekonstruktionsqualität mit der 8×32 LCT-4 erzielt. Im Vergleich zur 32×32 DCT treten bei Verwendung der 8×32 LCT-4 keine sichtbaren Blockartefakte auf (vgl. Abbildungen 5.8 und 5.14). Störende Kompressionsartefakte in Form von schnellen Grauwertwechseln, wie sie bei der 16×32 LCT-2 entstehen können, bleiben bei der 8×32 LCT-4 aus (vgl. Abbildungen 5.8 und 5.14).

6.7 Überlappende Kosinustransformation im Vergleich mit anderen blocküberlappenden Transformationen

In diesem Abschnitt wird die Leistungsfähigkeit von überlappenden lokalen trigonometrischen Basen mit der Leistungsfähigkeit von anderen blocküberlappenden Transformationsverfahren verglichen. Die Untersuchungen konzentrieren sich hierbei auf den Vergleich der vierfach überlappenden Kosinustransformation 8×32 LCT-4 mit den folgenden Transformationen:

- 8×16 TDLT (time-domain lapped transform) [LTT01, TT02b],
- 8×16 LOT, Typ II (lapped orthogonal transform) [Mal92],
- 8×32 GenLOT und 8×48 GenLOT (generalized linear-phase lapped orthogonal transform) [TN, SN96],
- 8×32 ELT (extended lapped transform) [Mal92].

Anhang G gibt eine Zusammenfassung der Transformationsmatrizen bzw. der Berechnungsalgorithmen sowie der Basisfunktionen dieser Transformationen. Für ein tiefergehendes Studium der Transformationsverfahren wird auf die oben angegebene Literatur verwiesen.

Um bei der Kompression von Bildsignalen sichtbare Randartefakte zu vermeiden, werden bei der LOT und der GenLOT die Bildsignale an den Rändern symmetrisch fortgesetzt. Da die Basisfunktionen dieser Transformationen gewisse Symmetrieeigenschaften aufweisen, entsteht durch die Signalfortsetzung kein zusätzlicher Codieraufwand. Die Basisfunktionen der ELT sind hingegen unsymmetrisch. Somit stellt für die ELT die symmetrische Signalfortsetzung in der Bilddatenkompression keine geeignete Lösung zur Vermeidung von Randartefakten dar (vgl. Abschnitt 3.1). Bei den Basisfunktionen der untersuchten ELT handelt es sich um Kosinus-IV-Funktionen, die mit einer vierfach überlappenden Fensterfunktion gewichtet werden. Sie lassen sich folglich der Klasse der überlappenden lokalen trigonometrischen Basen zuordnen. Für die ELT wurden daher Randbasisfunktionen nach dem in Abschnitt 3.5.5 neu eingeführten Verfahren entworfen. Sie sind im Anhang G.4 dargestellt. Bei der TDLT ist aufgrund der im Anhang G.3 angegebenen Verarbeitungsstrategie keine Randbehandlung erforderlich [TT02b].

Zur Quantisierung und Codierung der Transformationskoeffizienten wurden für alle Transformationen die Kompressionsstrategien L-CEB-Codec-I, L-CEB-Codec-II, SPIHT-Codec-I und SPIHT-Codec-II untersucht, wobei sich die nachfolgenden Auswertungen auf die Kompressionsstrategie L-CEB-Codec-II konzentrieren.

TDLT

Anhand des Rekonstruktionsergebnisses in Abbildung 6.2 ist zu erkennen, dass durch die Verwendung der 8×16 TDLT im Vergleich zur 8×8 DCT eine deutliche Reduktion von Blockartefakten erzielt wird. Lediglich in den tieffrequenten Bildbereichen, in denen sich der Grauwert allmählich ändert, sind bei der 8×16 TDLT äußerst geringfügige Verzerrungen in Form von schnellen Grauwertwechseln erkennbar¹. Die Rekonstruktionsqualität der 8×16 TDLT liegt bei diesem Kompressionsbeispiel über der Rekonstruktionsqualität der 8×16 LCT-2 (vgl. Abbildung 5.11). Im Vergleich zur 8×32 LCT-4 wird mit der TDLT eine etwas schlechtere Reduktion von Blockartefakten erzielt. Bei der 8×16 TDLT entstehen jedoch aufgrund der kürzeren Träger der Basisfunktionen weniger sichtbare Ringing-Artefakte als bei der 8×32 LCT-4.

Der objektive Vergleich der Rekonstruktionsergebnisse zeigt, dass bei den meisten Testbildern die PSNR-Werte mit einer Differenz von ca. ± 0.2 dB zwischen der Verwendung der 8×32 LCT-4 und der Verwendung der 8×16 TDLT nur geringfügig voneinander abweichen (vgl. Tabelle H.58, Anhang H.2.5). Lediglich bei den Testbildern „Barbara“, „Roof“ und „Fingerprint“ liefert die LCT-4 mit einem Zuwachs von bis zu 0.88 dB deutlich bessere PSNR-Werte.

LOT, Typ II

Die Verwendung der 8×16 LOT, Typ II führt zu ähnlichen sichtbaren Kompressionsartefakten wie die 8×16 LCT-2 (vgl. Abbildungen 6.2 und 5.11). Die Blockartefakte werden zwar reduziert, es entstehen jedoch in den tieffrequenten Bildteilen bei niedrigen Bitraten deutlich sichtbare Kompressionsartefakte in Form von schnellen Grauwertwechseln. Die 8×32 LCT-4 liefert im Vergleich zur 8×16 LOT, Typ II bei den untersuchten Testbildern sowohl in der subjektiven Beurteilung als auch in der objektiven Beurteilung (vgl. Tabelle H.59) eine bessere Rekonstruktionsqualität.

GenLOT

Abbildung 6.2 zeigt am Beispiel des Testbildes „Barbara“, dass mit der untersuchten vierfach überlappenden 8×32 GenLOT keine vollständige Reduktion der sichtbaren Blockartefakte erzielt wird. Die Ursache liegt in dem schnellen Wechsel der Funktionswerte der Basisfunktionen der niedrigsten Frequenz im Bereich der Blockgrenzen begründet. Bei der untersuchten sechsfach überlappenden 8×48 GenLOT treten hingegen keine sichtbaren Blockartefakte im rekonstruierten Bild auf (vgl. Abbildung 6.2). Visuell sind zwischen den rekonstruierten Bildern unter Verwendung der 8×32 LCT-4 und unter Verwendung der 8×48 GenLOT fast keine qualitativen Unterschiede zu erkennen. Tabelle H.60 zeigt, dass beide Transformationen bei den meisten Testbild-Bitraten-Kombinationen mit einer Differenz von ca. ± 0.15 dB fast die gleichen PSNR-Werte liefern. Lediglich die Testbilder „Barbara“, „Roof“ und „Fingerprint“ lassen sich mit einem PSNR-Zuwachs von bis zu 0.81 dB deutlich effizienter durch die 8×32 LCT-4 repräsentieren als durch die 8×48 GenLOT.

ELT

Bei der untersuchten vierfach überlappenden ELT hängt die Rekonstruktionsqualität in der verlustbehafteten Bilddatenkompression entscheidend von der Wahl des Parameters γ der Fensterfunktion ab. Für $\gamma = 1.0$ besitzt die Fensterfunktion zur Konstruktion der Basisfunktionen deutliche Sprünge an den Intervallrändern (vgl. Abbildung G.7) und es kann daher

¹Diese Kompressionsartefakte sind in der gedruckten Version dieser Arbeit kaum sichtbar.

8×16 TDLT, PSNR = 26.45 dB 8×16 LOT Typ II, PSNR = 26.21 dB 8×32 GenLOT, PSNR = 26.39 dB 8×48 GenLOT, PSNR = 26.76 dB

Abbildung 6.2: L-CEB-Codec-II: Kompression des Bildes „Barbara“ bei einer Bitrate von 0.125 bpp; Vergleich von 8×16 TDLT, 8×16 LOT Typ-II, 8×32 GenLOT und 8×48 GenLOT. Das 64×64 DC-Teilband wird bei dieser Kompressionsstrategie mit einer 3-level DWT weiterverarbeitet, so dass hieraus resultierend bei allen Kompressionsbeispielen das DC-Teilband aus 8×8 Transformationskoeffizienten besteht.

8×32 ELT, $\gamma = 1.0$, PSNR = 26.62 dB



8×32 ELT, $\gamma = 0.3$, PSNR = 26.93 dB



8×32 LCT-4, PSNR = 27.05 dB



DWT, JPEG2000, PSNR = 25.24 dB



Abbildung 6.3: L-CEB-Codec-II und JPEG2000: Kompression des Bildes „Barbara“ bei einer Bitrate von 0.125 bpp; Vergleich von DWT, 8×32 ELT und 8×32 LCT-4. Das 64×64 DC-Teilband der blocküberlappenden trigonometrischen Transformationen wird bei dieser Kompressionsstrategie mit einer 3-level DWT weiterverarbeitet, so dass hieraus resultierend das DC-Teilband aus 8×8 Transformationskoeffizienten besteht.

nach der Quantisierung der Transformationskoeffizienten zu sichtbaren blockähnlichen Artefakten in den rekonstruierten Bildern kommen (vgl. Abbildung 6.3). Wählt man hingegen den Parameter $\gamma = 0.3$, so reduziert sich die Höhe der Sprungstellen an den Intervallrändern (vgl. Abbildung G.7) und in den rekonstruierten Bildern treten fast keine sichtbaren blockähnlichen Artefakte mehr auf (vgl. Abbildung 6.3). In allen weiteren Untersuchungen kam daher ausschließlich der Parameter $\gamma = 0.3$ zur Anwendung. Sowohl in der subjektiven als auch in der objektiven Beurteilung der Rekonstruktionsqualität liefern die 8×32 LCT-4 und die 8×32 ELT annähernd gleiche Ergebnisse. Tabelle H.61 zeigt, dass sich die PSNR-Werte zwischen diesen beiden Transformationen bei den meisten Testbild-Bitraten-Kombinationen um weniger als 0.1 dB unterscheiden.

Zusammenfassend werden im Diagramm 6, das im Anhang H.1 für jedes der untersuchten Testbilder angegeben ist, die PSNR-Kurven der L-CEB-II-Codierung unter Verwendung der 8×32 LCT-4, der 8×16 TDLT, der 8×32 ELT und der 8×48 GenLOT miteinander verglichen. Bei Transformationskoeffizientenblöcken der Größe 8×8 handelt es sich um die leistungsfähigsten blocküberlappenden Transformationen, die im Rahmen dieser Arbeit untersucht wurden. Abschließend sei darauf hingewiesen, dass die nahezu vollständige Reduktion der sichtbaren Blockartefakte bzw. der blockähnlichen Artefakte bei diesen Transformationen erst durch die weitere Zerlegung der DC-Transformationskoeffizienten erreicht wird.

6.8 Überlappende Kosinustransformation im Vergleich zur Wavelet-Transformation

Da der ursprünglich von Said und Pearlman entwickelte SPIHT-Codec zur Kompression von Bilddaten [SP96b, PS08] auf der Wavelet-Transformation¹ basiert, ist bei der Untersuchung der LCT-basierten SPIHT-Codierungsstrategie die Analyse der Leistungsfähigkeit von lokalen trigonometrischen Basen im Vergleich zu Wavelets von besonderem Interesse. Im Diagramm 5 (Anhang H.1) wird daher für jedes Testbild neben den PSNR-Kurven für die 32×32 DCT, 16×32 LCT-2 und 8×32 LCT-4 auch die PSNR-Kurve für die SPIHT-Codierung unter Verwendung der DWT (9/7-Wavelet) angegeben.

Bemerkenswert ist bereits der Vergleich der 9/7-DWT mit der 32×32 DCT. Die Verwendung der 32×32 DCT führt gegenüber der 9/7-DWT bei den untersuchten Testbild-Bitraten-Kombinationen mehrheitlich zu besseren objektiven Rekonstruktionsergebnissen. Mit einer Erhöhung des PSNR um bis zu 2.43 dB fällt die Leistungssteigerung durch die 32×32 DCT insbesondere bei den Testbildern „Barbara“, „Lighthouse“, „Roof“, „Fingerprint“, „Wood Grain“ und „Water“ sehr deutlich aus (vgl. Tabelle H.62, Anhang H.2.6). Lediglich die Testbilder „Lena“, „Peppers“ und „F16“ lassen sich wesentlich besser durch die 9/7-DWT repräsentieren.

Betrachtet man schließlich die SPIHT-Codierung unter Verwendung der 16×32 LCT-2 oder der 8×32 LCT-4, so ergeben sich mit diesen überlappenden Kosinustransformationen bei fast allen Testbildern bessere objektive Rekonstruktionsergebnisse als mit der 9/7-DWT. Die 16×32 LCT-2 liefert im Vergleich zur 9/7-DWT bei 98 der insgesamt 114 dokumentierten Testbild-Bitraten-Kombinationen höhere PSNR-Werte (vgl. Tabelle H.63). Die 8×32 LCT-4 führt bei 102 der insgesamt 114 dokumentierten Testbild-Bitraten-Kombinationen zu besseren PSNR-Werten (vgl. Tabelle 6.4). Am signifikantesten fallen die Leistungssteigerungen durch die 16×32 LCT-2 und die 8×32 LCT-4 mit bis zu 2.84 dB bzw. 2.52 dB bei den Testbildern „Barbara“, „Lighthouse“, „Roof“, „Fingerprint“, „Wood Grain“ und „Water“ aus. Bei den wenigen Testbild-Bitraten-Kombinationen, bei denen die 9/7-DWT die besseren objektiven

¹Bei der verlustbehafteten Kompression kommt das 9/7-Wavelet zur Anwendung.

Kompressionsergebnisse liefert, stellen die 16×32 LCT-2 und 8×32 LCT-4 die Bilder mit einer PSNR-Differenz von bis zu -0.65 dB bzw. bis zu -0.36 dB jedoch nur geringfügig schlechter als die 9/7-DWT dar (vgl. Tabellen H.63 und 6.4).

In Bezug auf die mit dem SPIHT-Algorithmus erzielten Kompressionsergebnisse ist besonders hervorzuheben, dass die deutliche Verbesserung der objektiven Rekonstruktionsqualität durch die Verwendung der 16×32 LCT-2 oder der 8×32 LCT-4 im Vergleich zu dem für die Bilddatenkompression äußerst leistungsfähigen 9/7-Wavelet mit einer Codierungsstrategie erreicht wird, die ursprünglich für die Wavelet-Transformation entwickelt wurde, an die Eigenschaften der Wavelet-Transformation angepasst ist und mit zu den effizientesten Wavelet-Codierungsstrategien gehört. Das Resultat, dass die 16×32 LCT-2 und die 8×32 LCT-4 bereits in einer wavelet-freundlichen Codierungsumgebung bessere PSNR-Werte als die DWT mit dem 9/7-Wavelet liefert, bringt sehr deutlich das Potential der untersuchten lokalen trigonometrischen Basen bzgl. der objektiven Rekonstruktionsqualität zum Ausdruck.

Wie die Untersuchungsergebnisse der vorangegangenen Abschnitte zeigen, lässt sich die Effizienz der LCT-basierten Bildkomprimierung steigern, wenn anstatt des SPIHT-Algorithmus an die Blockstruktur angepasste Verarbeitungsstrategien wie die AGU- und die L-CEB-Codierung verwendet werden (vgl. Tabellen H.47 und H.52). Die Tabellen H.65 und 6.5 ermöglichen Aussagen über die Kompressionsleistung des AGU-Codec-II unter Verwendung der 16×32 LCT-2 und des L-CEB-Codec-II unter Verwendung der 8×32 LCT-4, wobei für den Leistungsvergleich mit der waveletbasierten Bildkomprimierung der JPEG2000-Codec [Tau10] gewählt wurde. Obwohl mit dem JPEG2000-Verfahren bei mehreren Testbildern deutlich bessere Rekonstruktionsergebnisse als mit dem SPIHT-Codec zu verzeichnen sind (vgl. Tabelle H.66), kommt auch dieser Wavelet-Codec nicht an die Leistungsfähigkeit der untersuchten LCT-basierten Kompressionsstrategien heran. Der AGU-Codec-II erzielt unter Verwendung der 16×32 LCT-2 im Vergleich zu JPEG2000 bei 103 der insgesamt 114 untersuchten Testbild-Bitraten-Kombinationen höhere PSNR-Werte (vgl. Tabelle H.65). Der L-CEB-Codec-II führt unter Verwendung der 8×32 LCT-4 bei 100 der insgesamt 114 untersuchten Testbild-Bitraten-Kombinationen zu besseren PSNR-Werten (vgl. Tabelle 6.5). Mit den LCT-basierten Kompressionsstrategien werden im Vergleich zu JPEG2000 PSNR-Gewinne von bis zu 2.26 dB bzw. 1.97 dB erzielt (vgl. Tabellen H.65 und 6.5).

Zum Abschluss der Betrachtungen zur objektiven Beurteilung der Rekonstruktionsqualität sei auf Diagramm 7 hingewiesen, das im Anhang H.2.5 für jedes Testbild angegeben ist. Es fasst die PSNR-Kurven der leistungsfähigsten untersuchten Kompressionsstrategien, die auf trigonometrischen Funktionen basieren, zusammen:

- AGU-Codec-II, 32×32 DCT,
- AGU-Codec-II, 16×32 LCT-2,
- L-CEB-Codec-II, 8×32 LCT-4

und vergleicht sie mit den Rekonstruktionsergebnissen von

- JPEG [IJG11],
- JPEG2000 [Tau10] und
- SPIHT [SP96a].

Die dargestellten PSNR-Kurven verdeutlichen nochmals die Leistungsfähigkeit der LCT-basierten Kompressionsstrategien und zeigen, dass mit ihnen nicht nur gegenüber dem herkömmlichen DCT-basierten JPEG-Verfahren erhebliche Codiergewinne erzielt werden können, sondern auch gegenüber waveletbasierten State-of-the-Art Kompressionsverfahren.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.83	1.58	1.92	2.10	1.91	1.86
„Lena“	0.03	0.09	0.15	0.14	0.10	0.05
„Zelda“	-0.05	0.11	0.16	0.18	0.17	0.37
„Boat“	0.01	0.09	0.24	0.43	0.44	0.30
„Goldhill“	0.03	0.09	0.16	0.21	0.18	0.20
„Lighthouse“	0.68	1.09	0.76	0.49	0.40	0.34
„Roof“	0.92	1.37	1.68	2.16	2.37	2.52
„Clown“	0.11	0.52	0.46	0.61	0.67	0.55
„Houses“	0.09	0.22	0.45	0.55	0.25	0.11
„Mandrill“	0.06	0.06	0.19	0.23	0.18	0.26
„Fingerprint“	0.30	0.63	0.86	0.93	1.36	1.68
„Peppers“	0.01	-0.17	-0.26	-0.36	-0.19	-0.33
„F16“	0.19	-0.04	0.08	0.01	-0.07	-0.15
„Bubble“	-0.05	0.03	0.09	0.11	0.12	-0.06
„Bark“	0.00	0.11	0.13	0.29	0.42	0.53
„Wood Grain“	0.25	0.61	1.35	1.87	1.86	1.40
„Sand“	0.03	0.08	0.26	0.27	0.26	0.21
„Water“	0.31	0.81	1.33	1.47	1.18	0.93
„Texture“	0.01	0.06	0.10	0.13	0.23	0.21

Tabelle 6.4: $\Delta\text{PSNR} = \text{PSNR}_{\text{SPIHT-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{SPIHT-Codec-II}, \text{DWT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	1.41	1.81	1.90	1.97	1.75	1.34
„Lena“	0.42	0.25	0.23	0.08	0.04	-0.06
„Zelda“	0.50	0.30	0.29	0.14	0.08	0.08
„Boat“	0.50	0.32	0.30	0.41	0.27	0.19
„Goldhill“	0.23	0.23	0.25	0.16	0.26	0.17
„Lighthouse“	0.83	0.80	0.39	0.25	0.21	0.39
„Roof“	1.45	1.55	1.62	1.88	1.80	1.32
„Clown“	0.45	0.68	0.40	0.49	0.41	0.35
„Houses“	0.21	0.16	0.27	0.19	-0.13	-0.55
„Mandrill“	0.17	0.26	0.29	0.34	0.33	0.31
„Fingerprint“	0.49	0.69	0.75	0.66	1.14	1.13
„Peppers“	0.30	-0.12	-0.32	-0.45	-0.39	-0.33
„F16“	0.50	0.08	0.03	-0.03	-0.33	-0.43
„Bubble“	0.43	0.30	0.44	0.28	0.28	0.13
„Bark“	0.27	0.27	0.44	0.34	0.70	0.68
„Wood Grain“	-0.01	-0.01	0.59	0.89	0.12	-0.08
„Sand“	0.31	0.46	0.45	0.37	0.49	0.50
„Water“	0.61	0.74	1.04	0.99	0.41	0.24
„Texture“	0.23	0.24	0.36	0.25	0.49	0.24

Tabelle 6.5: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{JPEG2000}, \text{DWT}}$, ΔPSNR in dB.

Für die subjektive Beurteilung der Rekonstruktionsqualität der LCT-basierten Bilddatenkompression im Vergleich zu JPEG2000 befinden sich in Abbildung 6.3 und im Anhang H.3.3 Beispiele von rekonstruierten Bildern. Es ist zu erkennen, dass mit dem L-CEB-Codec-II auf Basis der 8×32 LCT-4 alle gezeigten Bilder detailreicher rekonstruiert werden als mit JPEG2000. Die Verwendung der 8×32 LCT-4 führt jedoch teilweise zu geringfügig ausgeprägteren Ringing-Artefakten. Anhang H.3.4 zeigt anhand mehrerer Testbilder, dass sich diese Kompressionsartefakte durch eine Nachbearbeitung der rekonstruierten Bilder mittels DCT-basierter Filterung deutlich reduzieren lassen. Die hierbei zumeist erzielten PSNR-Gewinne können aus den Tabellen im Anhang H.1 bestimmt werden.

Die Ergebnisse der im Rahmen dieser Arbeit durchgeführten Leistungsanalysen zeigen, dass bzgl. der untersuchten lokalen trigonometrischen Basen die geringsten Rekonstruktionsfehler mit der 16×32 LCT-2 und der 8×32 LCT-4 erzielt werden. Die 8×32 LCT-4 führt gegenüber der 16×32 LCT-2 zu weniger störend empfundenen Kompressionsartefakten und liefert somit subjektiv bewertet die beste Rekonstruktionsqualität. Sind in den zu komprimierenden Testbildern keine Bildbestandteile auszumachen, die sich mit überlappenden lokalen trigonometrischen Basen kompakter darstellen lassen als mit dem 9/7-Wavelet, so erhält man bei den untersuchten Kompressionsstrategien sowohl mit der 16×32 LCT-2 als auch mit der 8×32 LCT-4 im Vergleich zur 9/7-DWT mehrheitlich gleichwertige oder geringfügig bessere Rekonstruktionsergebnisse. Sobald in Bildern jedoch harmonisch strukturierte Texturen auftreten, wird mit den LCT-basierten Verfahren eine erhebliche Steigerung der Rekonstruktionsqualität erzielt. Bemerkenswert ist, dass sich die Verbesserung der Kompressionsleistung durch die Verwendung von überlappenden lokalen trigonometrischen Basen im Vergleich zum 9/7-Wavelet bereits in einer ursprünglich für die Wavelet-Transformation entwickelten Codierungsumgebung, dem SPIHT-Codec, nachweisen lässt.

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

Die vorliegende Arbeit setzte sich zum Ziel, die Anwendung mehrfach überlappender lokaler trigonometrischer Basisfunktionen in der Bilddatenkompression zu untersuchen und einen Beitrag zur Entwicklung effizienter Kompressionsverfahren auf Grundlage dieser Basisfunktionen zu leisten.

Ausgehend von der Frage, wie Signale, die nur auf einem beschränkten Intervall $[0, L]$ definiert sind, durch überlappende lokale trigonometrische Basisfunktionen derart repräsentiert werden können, dass sich auch an den Signalrändern eine gute Signaldarstellung erzielen lässt, wurde als Lösungsansatz die Anwendung von Basen auf dem Intervall vorgeschlagen. Hierzu wurde ein neues System von Basisfunktionen eingeführt, das aus bestehenden Kosinus-IV-Basisfunktionen sowie aus neu entwickelten Kosinus-IV- und Kosinus-II-Randbasisfunktionen gebildet wird, und es wurden neue Randfenster eingeführt, mit denen sich geeignete Randbasisfunktionen konstruieren lassen. Die unter Verwendung dieser Randbasisfunktionen gebildeten Kosinus-IV-Kosinus-II-Basen auf dem Intervall ermöglichen eine effiziente Darstellung von endlich langen Signalen auch an den Signalrändern und erfüllen für die Anwendung in der Bilddatenkompression wichtige Eigenschaften:

- Die untersuchten Systeme von Kosinus-IV-Kosinus-II-Basisfunktionen auf dem Intervall sind stabil.
- Die Auflösung bzw. Rekonstruktion von Konstanten ausschließlich mit Basisfunktionen der niedrigsten Frequenz ist auch an den Signalrändern erfüllt.
- Die Anwendung der vorgeschlagenen Basen auf dem Intervall ruft keine sichtbaren Randartefakte hervor.
- Es wird kein zusätzlicher Codieraufwand für die gute Darstellung der Signalränder benötigt.

Als ein wichtiges Werkzeug zur Untersuchung von überlappenden lokalen trigonometrischen Basen auf dem Intervall und zur Entwicklung von schnellen Transformationsalgorithmen wurden Klapp- und Aufklappoperatoren vorgestellt und für das vorgeschlagene System von Kosinus-IV-Kosinus-II-Basisfunktionen auf dem Intervall hergeleitet. Die Nützlichkeit dieser Operatoren beruht darauf, dass sie die Abbildung zwischen dem System von überlappenden trigonometrischen Basisfunktionen auf dem Intervall und einem System von nicht überlappenden Orthonormalbasisfunktionen beschreiben. Daher lassen sich bestimmte Eigenschaften von

überlappenden trigonometrischen Basen auf dem Intervall mit bestimmten Eigenschaften der Klapp- und Aufklappoperatoren identifizieren. Mit Hilfe dieser Operatoren wurde die Riesz-Stabilität der untersuchten Basen auf dem Intervall nachgewiesen.

Ein weiterer Schwerpunkt dieser Arbeit lag auf der Entwicklung effizienter Algorithmen zur Realisierung der überlappenden Kosinustransformation. Um die neu vorgeschlagenen Systeme von überlappenden Kosinus-IV-Kosinus-II-Basisfunktionen auf dem Intervall in der Bilddatenkompression anwenden zu können, wurde zunächst die Darstellung mit diesen Basisfunktionen diskretisiert. Mittels numerischer Quadratur konnte gezeigt werden, dass sich die Transformationskoeffizienten eines diskreten Signals bzgl. überlappender Kosinus-IV-Kosinus-II-Basisfunktionen aus der DCT-IV bzw. DCT-II des diskreten geklappten Signals ergeben. Die Anwendung der Klapp- und Aufklappoperatoren ermöglicht daher die Nutzung bestehender Algorithmen zur schnellen Realisierung von trigonometrischen Blocktransformationen.

Um auch die Bestimmung des geklappten Signals bzw. die Rekonstruktion aus dem geklappten Signal mit einem möglichst geringen Berechnungsaufwand realisieren zu können, wurden effiziente Algorithmen für Klapp- und Aufklappoperatoren eingeführt. Da die vom Klappoperator verwendeten mehrfach überlappenden dualen Fenster längere Träger aufweisen als die vom Aufklappoperator benutzten Fenster, wurden für Klappoperatoren schnelle Algorithmen vorgeschlagen, die auf dem Lösen von linearen Gleichungssystemen mit Bandstruktur unter Verwendung von LU-Faktorisierungsverfahren beruhen. Es konnte gezeigt werden, dass durch die Anwendung der entwickelten schnellen Algorithmen der Berechnungsaufwand zur Realisierung der überlappenden Kosinustransformation signifikant reduziert wird.

Um die Leistungsfähigkeit der mehrfach überlappenden lokalen trigonometrischen Basen in der Bilddatenkompression beurteilen zu können, wurden verschiedene Ansätze zur Quantisierung und Codierung der Transformationskoeffizienten untersucht. Die Analyse der zunächst vorgeschlagenen LCT-basierten Kompressionsstrategie SPIHT-Codec-I führte zu dem Resultat, dass die Verwendung der mehrfach überlappenden Kosinustransformationen 8×24 LCT-3, 8×32 LCT-4 und 8×48 LCT-6 eine bessere Reduktion von Blockartefakten bewirkt als die zweifach überlappende 8×16 LCT-2, aber die Darstellung tieffrequenter Bildinhalte sichtbare Artefakte in Form von schnellen Grauwertwechseln aufweisen kann. Es ließ sich jedoch zeigen, dass solche Kompressionsartefakte auch bei der SPIHT-Codierung unter Verwendung der 3-level 9/7-DWT, die zur gleichen Anzahl an DC-Koeffizienten wie die oben genannten trigonometrischen Transformationen führt, im rekonstruierten Bild auftreten. Da sich bei der DWT dieses Problem durch die weitere Zerlegung der DC-Koeffizienten lösen lässt, wurde dieser Ansatz auch bei der LCT verfolgt. Die Analyse der daraufhin vorgeschlagenen LCT-basierten Kompressionsstrategie SPIHT-Codec-II führte zu dem Resultat, dass bei Verwendung der mehrfach überlappenden trigonometrischen Basen ebenfalls keine sichtbaren Blockartefakte mehr auftreten.

Mit Hilfe des SPIHT-basierten Codierungsansatzes konnten entscheidende Erkenntnisse über die Leistungsfähigkeit von mehrfach überlappenden lokalen trigonometrischen Basen im Vergleich zu Wavelets gewonnen werden. Die Kompression von insgesamt 19 Testbildern mit verschiedenartigen Bildinhalten zeigte, dass die Verwendung der 8×32 LCT-4 im Vergleich zur 9/7-DWT bei 102 der insgesamt 114 untersuchten Testbild-Bitraten-Kombinationen eine Verbesserung der objektiven Rekonstruktionsqualität bewirkt. Insbesondere bei der Kompression von Bildern, die wiederkehrende Muster aufweisen, ließen sich signifikante Steigerungen der PSNR-Werte um bis zu 2.52 dB erzielen. Das Resultat, dass bereits das Ersetzen der als sehr leistungsfähig geltenden 9/7-DWT durch die 8×32 LCT-4 in einer Codierungsumgebung, die ursprünglich für die Wavelet-Transformation entwickelt wurde, zu einer Erhöhung der Rekonstruktionsqualität führt, ist ein deutlicher Indikator für das Potential der 8×32 LCT-4.

Schließlich wurden mit der CEB- und der AGU-Codierung Verfahren untersucht, die an die

Eigenschaften trigonometrischer Transformationen angepasst sind. Es konnte festgestellt werden, dass diese Codieransätze im Vergleich zur SPIHT-Codierung eine effizientere Codierung der LCT-Koeffizienten ermöglichen. Eine Verbesserung der Darstellung tieffrequenter Bildinhalte wurde auch bei diesen Codieransätzen durch die weitere Zerlegung der DC-Koeffizienten in den Kompressionsstrategien-II erreicht.

Ausführliche Untersuchungen zur Leistungsfähigkeit der mehrfach überlappenden lokalen trigonometrischen Basen führten zu dem Resultat, dass sich insbesondere die vierfach überlappenden Basisfunktionen für die Entwicklung sehr effizienter Kompressionsverfahren eignen, die auch dem Vergleich mit leistungsfähigen waveletbasierten Kompressionsverfahren wie z.B. JPEG2000 standhalten.

Im Hinblick auf das Ziel dieser Arbeit, einen Beitrag zur Entwicklung leistungsfähiger Kompressionsverfahren zu leisten, stellen die neu eingeführten Basen auf dem Intervall sowie die entwickelten schnellen Transformationsalgorithmen wesentliche Resultate dieser Arbeit dar. Die Untersuchungsergebnisse dieser Arbeit zeigen, dass sich auf Grundlage von mehrfach überlappenden lokalen trigonometrischen Basen sehr leistungsfähige Kompressionsverfahren entwickeln lassen.

7.2 Ausblick

Die vorliegenden Untersuchungen zum Einsatz mehrfach überlappender lokaler trigonometrischer Basen in der Bilddatenkompression konzentrierten sich auf Systeme, die unter Verwendung von Kosinus-IV-Basisfunktionen gebildet werden. Die in [BC03] vorgestellten mehrfach überlappenden Kosinus-II-Sinus-II-Basisfunktionen fanden hierbei keine Berücksichtigung. Wie nachfolgend kurz erläutert wird, weisen auch diese Basisfunktionen Eigenschaften auf, die eine gute Eignung dieser Basisfunktionen für die Anwendung in der Bilddatenkompression erwarten lassen.

Bei Kosinus-II-Sinus-II-Basen handelt es sich um Systeme $\{\psi_j^k : j \in \mathbb{Z}, k \in \mathbb{N}_0\}$, die aus einer Mischung von Kosinus-II-Basisfunktionen

$$\psi_{2j}^k(t) = \begin{cases} w(t - 2j), & \text{falls } k = 0, \\ w(t - 2j)\sqrt{2} \cos(k(t - 2j)\pi), & \text{falls } k \geq 1 \end{cases} \quad (7.1)$$

und Sinus-II-Basisfunktionen

$$\psi_{2j+1}^k(t) = w(t - 2j - 1)\sqrt{2} \sin((k + 1)(t - 2j - 1)\pi), \quad \text{falls } k \geq 0 \quad (7.2)$$

bestehen. Zur Darstellung von Signalen durch diese Basisfunktionen wird die Einteilung der reellen Achse in Intervalle $[2j, 2j + 1]$ und $(2j + 1, 2j + 2)$, $j \in \mathbb{Z}$, betrachtet, wobei den Intervallen $[2j, 2j + 1]$ Kosinus-II-Basisfunktionen und den Intervallen $(2j + 1, 2j + 2)$ Sinus-II-Basisfunktionen zugeordnet werden. Die Signalrekonstruktion erfolgt nach

$$f = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{N}_0} (a_{j,k} \psi_{2j}^k + b_{j,k} \psi_{2j+1}^k). \quad (7.3)$$

In [BC03] werden zur Konstruktion von mehrfach überlappenden Kosinus-II-Sinus-II-Basisfunktionen B-Splines als Wichtungsfenster vorgeschlagen. B-Splines M_m der Ordnung m (vgl. Gleichungen (2.70) und (2.71)) besitzen die Eigenschaft, Polynome bis zu einer Ordnung von $m - 1$ zu reproduzieren, d.h., dass jedes Polynom $p \in \prod_{m-1}$ durch

$$p(t) = \sum_{j \in \mathbb{Z}} c_j M_m(t - j) \quad (7.4)$$

dargestellt werden kann. Da sich lokal glatte Bildbereiche gut durch stückweise Polynome approximieren lassen (vgl. Abschnitt 1.1.3), ist im Hinblick auf die Anwendung von B-Splines in der Bilddatenkompression die polynomreproduzierende Eigenschaft dieser Funktionen von besonderem Interesse. Wählt man zur Konstruktion von Kosinus-II-Sinus-II-Basisfunktionen als Wichtungsfenster

$$w(t) := M_m \left(\frac{1}{2}t - \frac{1}{4} \right), \quad (7.5)$$

so sind nach Gleichung (7.1) die Kosinus-II-Basisfunktionen ψ_{2j}^0 B-Splines. Demzufolge lassen sich die polynomialen Anteile von Signalen bis zu einer Ordnung von $m - 1$ bereits allein durch die Kosinus-II-Basisfunktionen der niedrigsten Frequenz darstellen

$$p = \sum_{j \in \mathbb{Z}} a_{j,0} \psi_{2j}^0. \quad (7.6)$$

Somit werden im Vergleich zu den in Abschnitt 2.3.3 beschriebenen Kosinus-IV-Basisfunktionen nur halb so viele Basisfunktionen bzw. Transformationskoeffizienten der niedrigsten Frequenz zur Darstellung der Polynome benötigt. Des Weiteren lassen sich auch mit Polynomen modulierte Harmonische sparsam durch wenige Basisfunktionen rekonstruieren [BC03]. Im Rahmen einer zukünftiger Arbeit könnte daher untersucht werden, inwieweit sich unter Ausnutzung dieser Eigenschaften leistungsfähige Verfahren zur Bilddatenkompression entwickeln lassen.

Literaturverzeichnis

- [AACI93] AHARONY, G., A. AVERBUCH, R. R. COIFMAN und M. ISRAELI: *Local cosine transform – a method for the reduction of the blocking effect in JPEG*. In: LAINE, A. F. (Herausgeber): *Mathematical Imaging: Wavelet Applications in Signal and Image Processing*, SPIE Proceedings Series vol. 2034, Seiten 205–217, San Diego, CA, 1993.
- [AACI94] AHARONY, G., A. AVERBUCH, R. R. COIFMAN und M. ISRAELI: *Local cosine transform – a method for the reduction of the blocking effect in JPEG*. *Journal of Mathematical Imaging and Vision*, 3:7–38, 1994.
- [AAN88] ARAI, Y., T. AGUI und M. NAKAJIMA: *A fast DCT-SQ Scheme for Images*. *Transactions of the IEICE*, E71(11):1095, Nov. 1988.
- [ABMD92] ANTONINI, M., M. BARLAUD, P. MATHIEU und I. DAUBECHIES: *Image coding using wavelet transform*. *IEEE Transactions on Image Processing*, Seiten 205–220, 1992.
- [ANR74] AHMED, N., T. NATARAJAN und K. R. RAO: *Discrete Cosine Transform*. *IEEE Transactions on Computers*, C-23:90–93, Jan. 1974.
- [Aus94] AUSCHER, P.: *Remarks on local Fourier bases*. In: BENEDETTO, J. J. und M. W. FRAZIER (Herausgeber): *Wavelets: Mathematics and Applications*, Seiten 203–218. CRC Press, Boca Raton, FL, 1994.
- [AWW92] AUSCHER, P., G. WEISS und M. V. WICKERHAUSER: *Local Sine and Cosine Bases of Coifman and Meyer and the Construction of Smooth Wavelets*. In: CHUI, C. K. (Herausgeber): *Wavelets: A Tutorial in Theory and Applications*, Seiten 237–256. Academic Press, Boston, 1992.
- [Bar46] BARI, N. K.: *Sur les bases dans l'espace de Hilbert*. *Dokl. Akad. Nauk. SSSR*, 54:379–382, 1946.
- [Bar51] BARI, N. K.: *Biorthogonal systems and bases in Hilbert spaces*. *Moskov. Gos. Univ. Uchenye Zapiski Matematika*, 148:69–107, 1951.
- [Bar94] BARNARD, H. J.: *Image and Video Coding Using a Wavelet Decomposition*. Dissertation, Technische Universität Delft, Netherlands, 1994.
- [BC99] BITTNER, K. und C. K. CHUI: *From local cosine bases to global harmonics*. *Applied and Computational Harmonic Analysis*, 6:382–399, 1999.
- [BC03] BITTNER, K. und C. K. CHUI: *Spline Modulation of Sinusoids for Signal Representation*. *Journal of Fourier Analysis and Applications*, 9:597–622, 2003.

- [BC04] BITTNER, K. und C. K. CHUI: *Formulation of Localized Cosine Bases that Preserve Polynomial Modulated Sinusoids*. Journal of Fourier Analysis and Applications, 10:475–496, 2004.
- [BCP97] BITTNER, K., C. K. CHUI und J. PRESTIN: *Multivariate cosine wavelets*. In: NÜRNBERGER, G., J. W. SCHMIDT und G. WALZ (Herausgeber): *Multivariate Approximation and Splines*, ISNM 125, Seiten 29–44. Birkhäuser, Basel, 1997.
- [BD11] BITTNER, K. und E. DAUTBEGOVIC: *Adaptiv wavelet-based method for simulation of electronic circuits*. In: *Scientific Computing in Electrical Engineering 2010*, Mathematics in Industry. Springer, Berlin Heidelberg, 2011. accepted.
- [BEL07] BERGH, J., F. EKSTEDT und M. LINDBERG: *Wavelets mit Anwendungen in Signal- und Bildbearbeitung*. Springer-Verlag Berlin, 2007.
- [Bit98a] BITTNER, K.: *Bivariate local trigonometric bases on triangular partitions*. In: *Wavelets and Multiscale Methods*, Proceedings of the International Wavelet Conference, Tanger, Seiten 96–99, 1998.
- [Bit98b] BITTNER, K.: *On the reproduction of linear functions by local trigonometric bases*. In: CHUI, C. K. und L. L. SCHUMAKER (Herausgeber): *Approximation Theory IX*, Band II, Computational Aspects of Innovations in Applied Mathematics, Seiten 1–8. Vanderbilt University Press, Nashville, 1998.
- [Bit99a] BITTNER, K.: *Biorthogonal Wilson Bases*. In: UNSER, M. A., A. ALDROUBI und A. F. LAINE (Herausgeber): *Wavelet Applications in Signal and Image Processing VII*, Band 3813 der Reihe *Proceedings of the SPIE's 44th Annual meeting*, Seiten 410–421, Denver, CO, USA, 1999.
- [Bit99b] BITTNER, K.: *Error estimates and reproduction of polynomials for biorthogonal local trigonometric bases*. Applied and Computational Harmonic Analysis, 6:75–102, 1999.
- [Bit00a] BITTNER, K.: *Biorthogonal local trigonometric bases*. In: ANASTASSIOU, G. (Herausgeber): *Handbook on Analytic-Computational Methods*. CRC Press, 2000.
- [Bit00b] BITTNER, K.: *Verallgemeinerte Klappoperatoren und Biorthogonale Wilson-Basen*. Dissertation, Technische Universität München, Jan. 2000. Shaker Verlag Aachen.
- [Bit02a] BITTNER, K.: *Linear Approximation and Reproduction of Polynomials by Wilson Bases*. Journal of Fourier Analysis and Applications, 8:79–99, 2002.
- [Bit02b] BITTNER, K.: *Wilson Bases on the Interval*. In: FEICHTINGER, H. und T. STROHMER (Herausgeber): *Advances in Gabor Analysis*. Birkhäuser, Boston, 2002.
- [Bra99] BRADLEY, A. P.: *A wavelet visible difference predictor*. IEEE Transactions on Image Processing, 8(5):717–730, 1999.
- [Bri96] BRISLAWN, C. M.: *Classification of nonexpansive symmetric extension transforms for multirate filter banks*. Applied and Computational Harmonic Analysis, 3:337–357, 1996.

- [BSMM08] BRONSTEIN, I. N., K. A. SEMENDJAJEW, G. MUSIOL und H. MÜHLIG: *Taschenbuch der Mathematik*. Harri Deutsch, 2008.
- [BT97a] BASZENSKI, G. und M. TASCHE: *Fast polynomial multiplication and convolution related to the discrete cosine transform*. Linear Algebra and its Applications, 252:1–25, 1997.
- [BT97b] BASZENSKI, G. und M. TASCHE: *Programmpaket zur Berechnung diskreter trigonometrischer Transformationen*, 1997.
<http://www.iuk.fh-dortmund.de/~baszenski/>.
- [BV98] BERNARDINI, R. und M. VETTERLI: *Discrete- and continuous-time local cosine bases with multiple overlapping*. IEEE Transactions on Signal Processing, 46(12):3166–3180, 1998.
- [CG90] CRAMER, S. und R. GLUTH: *Computationally efficient real-valued filterbanks based on a modified O^2DFT* . Proceedings of European Signal Processing Conference, Seiten 585–588, Sep. 1990.
- [CH90] CHAN, S. C. und K. L. HO: *Direct methods for computing discrete sinusoidal transforms*. Proc. Inst. Elect. Eng., 137(6):433–442, 1990.
- [Chu92] CHUI, C. K.: *An Introduction to Wavelets*. Academic Press, Boston, 1992.
- [CM91] COIFMAN, R. R. und Y. MEYER: *Remarques sur l'analyse de Fourier à fenêtre*. C. R. Acad. Sci. Paris, 312:259–261, 1991.
- [CS96] CHUI, C. K. und X. SHI: *On the convergence of localized cosine series*. Annals of Numerical Mathematics, 3:51–61, 1996.
- [CS97a] CHUI, C. K. und X. SHI: *Characterization of biorthogonal cosine wavelets*. Journal of Fourier Analysis and Applications, 3:559–575, 1997.
- [CS97b] CHUI, C. K. und X. SHI: *A study of biorthogonal sinusoidal wavelets*. In: A. LEMEAUTE, C. RABUT und L. L. SCHUMAKER (Herausgeber): *Surface Fitting and Multiresolution Methods*, Innovations in Applied Mathematics, Seiten 51–66. Vanderbilt University Press, Nashville, 1997.
- [CSF77] CHEN, W. H., C. H. SMITH und S. C. FRALICK: *A fast computational algorithm for the discrete cosine transform*. IEEE Transactions on Communications, 25:1004–1009, Sep. 1977.
- [Dau92] DAUBECHIES, I.: *Ten Lectures on Wavelets*. In: *CBMS-NSF Regional Conference Series in Applied Mathematics*, Band 61, 1992.
- [DJJ91] DAUBECHIES, I., S. JAFFARD und J. L. JOURNÉ: *A simple Wilson orthonormal basis with exponential decay*. SIAM Journal on Mathematical Analysis, 22:554–573, 1991.
- [DMP91] DUHAMEL, P., Y. MAHIEUX und J. P. PETIT: *A fast algorithm for the implementation of filter banks based on time domain aliasing cancelation*. IEEE International Conference on Acoustics, Speech and Signal Processing, Seiten 2209–2212, Mai 1991.

- [DN98] DAVIS, G. M. und A. NOSRATINIA: *Wavelet-based image coding: An overview*. Applied and Computational Control, Signals, and Circuits, 1:205–269, 1998.
- [DS98] DAUBECHIES, I. und W. SWELDENS: *Factoring Wavelet Transforms Into Lifting Steps*. Journal of Fourier Analysis and Applications, 4(3):247–269, 1998.
- [Egi05] EGI AZARIAN, K.: *AGU - DCT Based High Quality Image Compression*, 2005. <http://www.cs.tut.fi/~karen/agucoder.htm>.
- [EHKA99] EGI AZARIAN, K. O., M. HELSINGIUS, P. KUOSMANEN und J. T. ASTOLA: *Removal of blocking and ringing artifacts using transform domain denoising*. In: *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems*, Band 4, Seiten 139–142, Orlando, Florida, 30. Mai - 02. Juni 1999.
- [EJP⁺01] EGNER, S., J. JOHNSON, D. PADUA, M. PÜSCHEL und J. XIONG: *Automatic Derivation and Implementation of Signal Processing Algorithms*. ACM SIGSAM Bulletin Communications in Computer Algebra, 35(2):1–19, 2001.
- [EP01] EGNER, S. und M. PÜSCHEL: *Automatic Generation of Fast Discrete Signal Transforms*. IEEE Transactions on Signal Processing, 49(9):1992–2002, Sep. 2001.
- [Fei90] FEIG, E.: *A fast scaled-DCT algorithm*. In: MOORHEAD, R. J. und K. S. PENNINGTON (Herausgeber): *Image Processing Algorithms and Techniques*, Band 1244 der Reihe *Proceedings of SPIE*, Seiten 2–13, 1990.
- [Fey01] FEY, G.: *Set Partitioning in Hierarchical Trees: eine FPGA-Implementierung*. Diplomarbeit, Martin-Luther-Universität Halle-Wittenberg, Sept. 2001.
- [FJ98] FRIGO, M. und S. G. JOHNSON: *FFTW: An adaptive software architecture for the FFT*. In: *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Band 3, Seiten 1381–1384. IEEE, 1998.
- [FJ99] FRIGO, M. und S. G. JOHNSON: *A fast Fourier transform compiler*. In: *Proceedings 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation*, Band 34, Seiten 169–180. ACM, Mai 1999.
- [FJ05] FRIGO, M. und S. G. JOHNSON: *The Design and Implementation of FFTW3*. Proceedings of the IEEE, 93(2):216–231, 2005.
- [For09] FORSTER, O.: *Analysis 3, Integralrechnung im \mathbb{R}^n mit Anwendungen*. Vieweg + Teubner, 2009.
- [FS94] FANG, X. und E. SÉRÉ: *Adapted multiple folding local trigonometric transforms and wavelet packets*. Applied and Computational Harmonic Analysis, 1:169–179, 1994.
- [FSV03] FISCHER, A., W. SCHIROTZKE und K. VETTERS: *Lineare Algebra: Eine Einführung für Ingenieure und Naturwissenschaftler*. B. G. Teubner Verlag, Wiesbaden, 2003.
- [FW92] FEIG, E. und S. WINOGRAD: *On the multiplicative complexity of discrete cosine transform*. IEEE Transactions on Information Theory, 38:1387–1391, 1992.
- [Gab46] GABOR, D.: *Theory of communication*. Journal of Institute of Electrical Engineers, 93:429–457, 1946.

- [Gel50] GELFAND, I. M.: *Eigenfunction expansions for equations with periodic coefficients*. Dokl. Akad. Nauk. SSR, 73(6):1117–1120, 1950.
- [GR10] GANTZ, J. und D. REINSEL: *The Digital Universe Decade - Are You Ready?* IDC, Mai, 2010. <http://www.emc.com/collateral/analyst-reports/idc-digital-universe-are-you-ready.pdf>.
- [GW93] GONZALEZ, R. C. und R. E. WOODS: *Digital Image Processing*. Addison-Wesley, Reading, 1993.
- [H2603] Joint Video Team of ITU-T and ISO/IEC JTC 1: *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H. 264 / ISO/IEC 14496-10 AVC)*, 2003.
- [Hac08] HACHENBERGER, D.: *Mathematik für Informatiker*. Pearson Studium, 2008.
- [Heu06] HEUSER, H.: *Lehrbuch der Analysis, Teil 1*. B.G. Teubner Verlag, Wiesbaden, 2006.
- [Hou87] HOU, H. S.: *A fast Recursive Algorithm for Computing the Discrete Cosine Transform*. IEEE Transactions on Acoustics, Speech, and Signal Processing, 35, 1987.
- [Huf52] HUFFMAN, D. A.: *A Method for the Construction of Minimum-Redundancy Codes*. In: *Proceedings of the IRE 40*, Band 9, Seiten 1098–1101, Sept. 1952.
- [IJG11] IJG: *Independent JPEG Group: Free JPEG-Software, Release 8c of 16-Jan-2011*, 2011. <http://www.ijg.org/files/jpegsrc.v8c.tar.gz>.
- [JLS94] JAWERTH, B., Y. LIU und W. SWELDENS: *Signal compression with smooth local trigonometric bases*. Optical Engineering, 33:2125–2135, 1994.
- [JPE00] ISO/IEC JTC1/SC29 WG 1: *JPEG2000 Part 1 Final Committee Draft Version 1.0*, März 2000.
- [JS95] JAWERTH, B. und W. SWELDENS: *Biorthogonal smooth local trigonometric bases*. Journal of Fourier Analysis and Applications, 2:109–133, 1995.
- [Kan05] KANZOW, C.: *Numerik linearer Gleichungssysteme: Direkte und iterative Verfahren*. Springer, Berlin, 2005.
- [KLX⁺06] KWAN, C., B. LI, R. XU, X. LI, T. D. TRAN und T. NGUYEN: *A complete image compression scheme based on overlapped block transform with post-processing*. EURASIP Journal on Applied Signal Processing, Article ID 10968, 2006:1–15, 2006.
- [Kot33] KOTELNIKOV, V. A.: *On the transmission capacity of 'ether' and wire in electro-communications*. Izd. Red. Upr. Svyazi RKKK, 1933.
- [KSW08] KIENCKE, U., M. SCHWARZ und T. WEICKERT: *Signalverarbeitung: Zeit-Frequenz-Analyse und Schätzverfahren*. Oldenbourg Wissenschaftsverlag, München, 2008.
- [KV89] KARLSSON, G. und M. VETTERLI: *Extension of finite length signals for subband coding*. Signal Processing, 17:161–168, Juni 1989.

- [Lae90] LAENG, E.: *Une base orthonormale de $L^2(\mathbb{R})$ dont les éléments sont bien localisés dans l'espace de phase et leurs supports adaptés à toute partition symétrique de l'espace des fréquences*. C. R. Acad. Sci. Paris, 311:677–680, 1990.
- [Lan84] LANGDON, G. G.: *An Introduction to Arithmetic Coding*. IBM Journal of Research and Development, 28(2):135–149, März 1984.
- [LCC01] LIAN, C. J., K. F. CHEN und H. H. CHEN L. G. CHEN: *Lifting based discrete wavelet transform architecture for JPEG2000*. In: *Proceedings of the 2001 IEEE International Symposium on Circuits and Systems*, Band 2, Seiten 445–448, Sydney, NSW, Australia, 06. - 09. Mai 2001.
- [Lee84] LEE, B.: *A new algorithm to compute the discrete cosine transform*. IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-32:1243–1245, Juni 1984.
- [LLM89] LOEFFLER, C. A., A. LIGTENBERG und G. S. MOSCHYTZ: *Practical fast 1-D DCT algorithms with 11 multiplications*. IEEE International Conference on Acoustics, Speech and Signal Processing, Seiten 988–991, Mai 1989.
- [LMR98] LOUIS, A., P. MAASS und A. RIEDER: *Wavelets: Theorie und Anwendungen*. B. G. Teubner, 1998.
- [LTT01] LIANG, J., C. TU und T. D. TRAN: *Fast Lapped Transforms via Time Domain Pre- and Post-Processing*. In: *IEEE International Conference on Information and Communication Systems (ICICS)*, Singapore, Okt. 2001.
- [LWY06] LIAN, Q. F., Y. G. WANG und D. Y. YAN: *Optimized Local Trigonometric Bases with Nonuniform Partitions*. Acta Mathematica Sinica, 22(4):1069–1084, 2006.
- [Mae88] MAESS, G.: *Vorlesungen über numerische Mathematik 1, Lineare Algebra*. Birkhäuser Verlag Basel-Boston, 1988.
- [Mak80] MAKHOUL, J.: *A Fast Cosine Transform in One and Two Dimensions*. IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-28(1):27–34, 1980.
- [Mal89] MALLAT, S.: *A Theory for Multiresolution Signal Decomposition: the Wavelet Representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7):674–693, 1989.
- [Mal90] MALVAR, H. S.: *Lapped transforms for efficient transform/subband coding*. IEEE Transactions on Acoustics, Speech, and Signal Processing, 38:969–978, 1990.
- [Mal92] MALVAR, H. S.: *Signal Processing with Lapped Transforms*. Artech House, Inc., Norwood, MA, 1992.
- [Mal98] MALLAT, S.: *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.
- [Mal00] MALI, A.: *Untersuchungen zum Einsatz lokaler trigonometrischer Basen in der Bilddatenkompression*. Diplomarbeit, Universität Rostock, Feb. 2000.

- [Mat96] MATVIYENKO, G.: *Optimized local trigonometric bases*. Applied and Computational Harmonic Analysis, 3:301–323, 1996.
- [MBHW01] MARPE, D., G. BLÄTTERMANN, G. HEISING und T. WIEGAND: *Video compression using context-based adaptive arithmetic coding*. IEEE International Conference on Image Processing, Thessaloniki, Greece, 3:558–561, Okt. 2001.
- [Mer10] MERTINS, A.: *Signaltheorie: Grundlagen der Signalbeschreibung, Filterbänke, Wavelets, Zeit-Frequenz-Analyse, Parameter- und Signalschätzung*. Vieweg & Teubner, Wiesbaden, 2010.
- [Mey90] MEYER, YVES: *Ondelettes et opérateurs. I*. Actualités Mathématiques. Hermann Press, Paris, 1990.
- [Mey02] MEYER, F. G.: *Image Compression With Adaptive Local Cosines: A Comparative Study*. IEEE Transactions Image Processing, 11(6):616–629, Juni 2002.
- [MM00] MALI, A. und E. MÜLLER: *Local Trigonometric Bases for Image Data Compression*. Frequenz, Journal of Telecommunications, 54:225–235, 9-10/2000.
- [MM03a] MALI, A. und E. MÜLLER: *Still image compression with four-overlapping local cosine bases*. Picture Coding Symposium, PCS '03, Saint-Malo, France, 2003.
- [MM03b] MALI, A. und E. MÜLLER: *Still Image Compression with Multiple Overlapping Local Cosine Bases*. In: *Proceedings of IEEE 2003 International Conference on Image Processing (ICIP 2003)*, Band 3, Seiten 233–236, Barcelona, Spain, Sep. 2003.
- [MNW04] MÜLLGES, G. E., K. NIEDERDRENK und R. WODICKA: *Numerik-Algorithmen: Verfahren, Beispiele, Anwendungen*. Springer, Berlin, 2004.
- [MPG93] ISO/IEC 11172-2: *Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 2: Video*, 1993.
- [MPG96] ISO/IEC 13818-2, ITU-T Rec. H. 262. 0: *Information technology - Generic coding of moving pictures and associated audio information – Part 2: Video*, 1996.
- [MPG04] ISO/IEC 14496-2: *Information technology - Coding of audio-visual objects – Part 2: Visual*, Third edition, 2004.
- [MPM01a] MALI, A., T. PALFNER und E. MÜLLER: *Complete Removal of Boundary Effects in Image Compression with Local Trigonometric Bases*. In: TESCHER, A. G. (Herausgeber): *Application of Digital image Processing XXIV*, Band 4472 der Reihe *Proceedings of SPIE*, Seiten 452–462, San Diego, CA, USA, 2001.
- [MPM01b] MALI, A., T. PALFNER und E. MÜLLER: *Removal of Boundary Effects in Image Compression Caused by Transform Using Non-Symmetric Local Trigonometric Bases*. In: SCHMALZ, M. S. (Herausgeber): *Mathematics of Data / Image Coding, Compression, and Encryption IV, with Applications*, Band 4475 der Reihe *Proceedings of SPIE*, Seiten 94–105, San Diego, CA, USA, 2001.
- [MS04] MEINEL, C. und H. SACK: *WWW: Kommunikation, Internetworking, Web-Technologien; mit 106 Tabellen*. Springer, Berlin [u.a.], 2004.

- [Neu96] NEUNDORF, W.: *Behandlung großer Matrizen auf dem PC*. Preprint No. M11/96, TU Ilmenau, Juni 1996.
- [Neu03] NEUBAUER, A.: *Irreguläre Abtastung - Signaltheorie und Signalverarbeitung*. Springer-Verlag Berlin, 2003.
- [OB96] ODEGARD, J. und C. S. BURRUS: *Discrete Finite Variation: A New Measure Of Smoothness For The Design Of Wavelet Basis*. In: *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seiten 1467–1470, 1996.
- [Ode96] ODEGARD, J.E.: *Moments, smoothness and optimization of wavelet systems*. Dissertation, Rice University, Houston, Texas, 1996.
- [Ohm95] OHM, J. R.: *Digitale Bildcodierung, Repräsentation, Kompression und Übertragung von Bildsignalen*. Springer-Verlag Berlin Heidelberg New York, 1995.
- [Ohm02] OHM, J. R.: *Bildsignalverarbeitung für Multimediasysteme*. TU Berlin, Institut für Nachrichtentechnik und Theoretische Elektrotechnik, 2002.
- [PELA05] PONOMARENKO, N. N., K. O. EGIAZARIAN, V. V. LUKIN und J. T. ASTOLA: *DCT based high quality image compression*. In: *Proceedings of 14th Scandinavian Conference on Image Analysis*, Seiten 1177–1185, Joensuu, Finland, Juni 2005.
- [Pia08] PIASTOWSKI, P. R. S.: *Objektive Bewertung der Wirksamkeit von Deblocking-Verfahren*. Dissertation, Technische Universität Dortmund, 2008.
- [PLEA07] PONOMARENKO, N. N., V. V. LUKIN, K. O. EGIAZARIAN und J. T. ASTOLA: *High Quality DCT based image compression using partition schemes*. IEEE Signal Processing Letters, 14:105–108, 2007.
- [PLEA08] PONOMARENKO, N. N., V. V. LUKIN, K. O. EGIAZARIAN und J. T. ASTOLA: *ADCTC: A new high quality DCT based coder for lossy image compression*. CD ROM Proceedings of LNLA, International Workshop on Local and Non-Local Approximation in Image Processing, Switzerland, Aug. 2008.
- [PM93] PENNEBAKER, W. B. und J. L. MITCHELL: *Still Image Compression Standard*. Van Nostrand Reinhold, New York, 1993.
- [PM99a] PALFNER, T. und E. MÜLLER: *Effects of Symmetric Periodic Extension for Multiwavelet Filter Banks on Image Coding*. In: *Wavelets and Filter Bank*, Band 1 der Reihe *Proceedings of the 1999 International Conference on Image Processing (ICIP '99)*, Seiten 628–632, Kobe, Japan, 1999.
- [PM99b] PALFNER, T. und E. MÜLLER: *Generalized symmetric periodic extension for multiwavelet filter banks*. In: UNSER, M. A., A. ALDROUBI und A. F. LAINE (Herausgeber): *Wavelet Applications in Signal and Image Processing VII*, Band 3813 der Reihe *Proceedings of SPIE*, Seiten 174–185, Denver, CO, USA, 1999.
- [PMJ⁺05] PÜSCHEL, M, J. M. F. MOURA, J. JOHNSON, D. PADUA, M. VELOSO, B. W. SINGER, J. XIONG, F. FRANCHETTI, A. GAČIĆ, Y. VORONENKO, K. CHEN, R. W. JOHNSON und N. RIZZOLO: *SPIRAL: Code Generation for DSP Transforms*. Proceedings of the IEEE, special issue on "Program Generation, Optimization, and Adaptation", 93(2), 2005.

- [Pon08a] PONOMARENKO, N. N.: *ADCTC: a new high quality DCT based coder for lossy image compression*, 2008. <http://www.ponomarenko.info/adct.htm>.
- [Pon08b] PONOMARENKO, N. N.: *AGU and AGU-MHV page - High Quality DCT based image coder*, 2008. <http://www.ponomarenko.info/agu.htm>.
- [Pri06] PRIMBS, M.: *Stabile biorthogonale Spline-Waveletbasen auf dem Intervall*. Dissertation, Universität Duisburg-Essen, März 2006.
- [PS08] PEARLMAN, W. A. und A. SAID: *Set Partition Coding*. Now Publishers, 2008.
- [PT02] PLONKA, G. und M. TASCHE: *Split-Radix Algorithms for Discrete Trigonometric Transforms*, 2002. preprint.
- [QT00] QUEIROZ, R. und T. D. TRAN: *A Fast Lapped Transform for Image Coding*. IEEE Transactions on Image Processing, 4:828–832, 2000.
- [RH96] RAO, K. R. und J. J. HWANG: *Techniques Standards for Image Video Audio Coding*. Prentice Hall, PTR, Upper Saddle River, New Jersey, 1996.
- [Ric06] RICHTER, H.: *Standardübergreifende Konzepte für blockbasierte Videodecodierung*. Dissertation, Universität Rostock, 2006.
- [RL79] RISSANEN, J. J. und G. G. LANGDON: *Arithmetic Coding*. IBM Journal of Research and Development, 23(2):149–162, März 1979.
- [RS80] REED, M. und B. SIMON: *Methods of Modern Mathematical Physics I, Functional Analysis*. Academic Press, 1980.
- [RY90] RAO, K. R. und P. YIP: *Discrete cosine transform : Algorithms, Advantages, Applications*. Academic Press, New York, 1990.
- [Sch00] SCHWARZ, H.: *Untersuchungen zur objektbasierten Videocodierung mit einer 3D-Wavelet-Transformation*. Dissertation, Forschungsberichte aus den Ingenieurwissenschaften, Band 72, Mensch & Buch Verlag, Berlin, 2000.
- [Sch03] SCHWANDT, H.: *Parallele Numerik - Eine Einführung*. Vieweg & Teubner, Wiesbaden, 2003.
- [Sch06] SCHEIBLER, K.: *Analyse und Vergleich kontextbasierter Bildkompressionsverfahren*. Diplomarbeit, Martin-Luther-Universität Halle-Wittenberg, März 2006.
- [Sha49] SHANNON, C. E.: *Communication in the Presence of Noise*. Proceedings of the IRE, 37(1):10–21, Jan. 1949.
- [Sha93] SHAPIRO, J. M.: *Embedded image coding using zerotrees of wavelet coefficients*. IEEE Transactions on Signal Processing, 41:3445–3463, Dez. 1993.
- [SK98] SHEN, M. Y. und C.C.J. KUO: *Review of Postprocessing Techniques for Compression Artifact Removal*. Journal of Visual Communication and Image Representation, 9(1):2–14, März 1998.
- [SN96] STRANG, G. und T. NGUYEN: *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

- [SP96a] SAID, A. und W. A. PEARLMAN: *C++ Source Code - SPIHT Image Compression Programs*, 1996. <http://www.cipr.rpi.edu/research/SPIHT/spiht3.html>.
- [SP96b] SAID, A. und W. A. PEARLMAN: *A new, fast, and efficient image codec based on set partitioning in hierarchical trees*. IEEE Transactions on Circuits and Systems for Video Technology, 6:243–250, Juni 1996.
- [SRWW87] SULLIVAN, D. J., J. J. REHR, J. W. WILKINS und K. G. WILSON: *Phase space Wannier functions in electronic structure calculations*. Preprint, Cornell University, 1987.
- [ST91] STEIDL, G. und M. TASCHKE: *A polynomial approach to fast algorithms for discrete Fourier-cosine and Fourier-sine transforms*. Mathematics of Computation, 56(193):281–296, 1991.
- [Ste92] STEIDL, G.: *Fast radix-p discrete cosine transform*. Applicable Algebra in Engineering, Communication and Computing, 3:39–46, 1992.
- [Str98] STRUTZ, T.: *Untersuchungen zur skalierbaren Kompression von Bildsequenzen bei niedrigen Bitraten unter Verwendung der dyadischen Wavelet-Transformation*. Dissertation, Universität Rostock, 1998. Shaker Verlag Aachen.
- [Str09] STRUTZ, T.: *Bilddatenkompression - Grundlagen, Codierung, Wavelets, JPEG, MPEG, H. 264*. VIEWEG Verlag, Wiesbaden, 2009.
- [Swe94] SWELDENS, W.: *The Construction and Application of Wavelets in Numerical Analysis*. Dissertation, Universität Leuven, Belgium, 1994.
- [SWS03] SCHÄFER, R., T. WIEGAND und H. SCHWARZ: *The emerging H. 264/AVC standard*. EBU Technical Review, Jan. 2003.
- [Tau99] TAUBMAN, D. S.: *High Performance Scalable Image Compression with EBCOT*. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Band 3, Seiten 344–348, 1999.
- [Tau00] TAUBMAN, D. S.: *High performance scalable image compression with EBCOT*. IEEE Transactions on Image Processing, 9(7):1158–1170, 2000.
- [Tau10] TAUBMAN, D.: *Kakadu Software, JPEG2000 developer toolkit, Version 6.4*, Juli 2010. <http://www.kakadusoftware.com/>.
- [Ten11] TENORTH, S.: *Adaptive Waveletmethoden zur Approximation von Bildern*. Dissertation, Georg-August-Universität Göttingen, 2011.
- [TM01] TAUBMAN, D. S. und M. MARCELLIN: *Jpeg2000: Image Compression Fundamentals, Standards and Practice*. Kluwer International Series in Engineering & Computer Science, 2001.
- [TN] TRAUTMANN, S. und T. Q. NGUYEN: *GenLOT Design Center und Transform-Based Image Coder*. <http://saigon.ece.wisc.edu/~waveweb/QMF/software.html>.
- [TN98] TRAN, T. D. und T. Q. NGUYEN: *Block transforms in progressive image coding*. In: TOPIWALA, P. N. (Herausgeber): *Wavelet Image and Video Compression*, Seiten 303–315. Kluwer Academic, Norwell, 1998.

- [TN99] TRAN, T. D. und T. Q. NGUYEN: *A progressive transmission image coder using linear phase filter banks as block transforms*. IEEE Transactions on Image Processing, 8:1493–1507, Nov. 1999.
- [TN00] TRAN, T. D. und T. Q. NGUYEN: *Lapped transforms for image compression*. In: RAO, K. R. und P. YIP (Herausgeber): *The Handbook on Transforms and Data Compression*, Seiten 197–265. CRC Press, 2000.
- [TOWS02] TAUBMAN, D. S., E. ORDENTLICH, M. WEINBERGER und G. SEROUSSI: *Embedded Block Coding in JPEG2000*. Signal Processing - Image Communication, 17:49–72, 2002.
- [TT02a] TU, C. und T. D. TRAN: *Context-Based Entropy Coding Of Block Transform Coefficients*. IEEE Transactions on Image Processing, Seiten 1271–1283, Nov. 2002.
- [TT02b] TU, C. und T. D. TRAN: *On Context-Based Entropy Coding Of Block Transform Coefficients*. IEEE International Conference on Image Processing, Seiten 669–672, Sept. 2002.
- [VN84] VETTERLI, M. und H. J. NUSSBAUMER: *Simple FFT and DCT algorithms with reduced number of operations*. Signal Processing, 6:267–278, Juli 1984.
- [Wac06] WACHTLER, K.: *Informatik II - Algorithmen und Datenstrukturen*. 2006. http://www.wachtler.de/informatik_2.ps.gz.
- [Wal91] WALLACE, G. K.: *The JPEG Still Picture Compression Standard*. Comm. ACM, 34, Seiten 30–44, 1991.
- [Wal92] WALLACE, G. K.: *The JPEG Still Picture Compression Standard*. IEEE Transactions on Consumer Electronics, CE-38, (1):18–34, Feb. 1992.
- [Wan84] WANG, Z.: *Fast algorithms for the discrete W transform and for the discrete cosine transform*. IEEE Transactions Acoustics, Speech, and Signal Processing, 32:803–816, Aug. 1984.
- [Wei64] WEIL, A.: *Sur certains groupes d'opérateurs unitaires*. Acta Math., 111:143–211, 1964.
- [Wer07] WERNER, D.: *Funktionalanalysis*. Springer Berlin Heidelberg, 2007.
- [Wer08] WERNER, M.: *Information und Codierung: Grundlagen und Anwendungen*. Vieweg+Teubner Wiesbaden, 2008.
- [Wer09] WERNER, M.: *Nachrichtentechnik: Eine Einführung für alle Studiengänge*. Vieweg+Teubner Wiesbaden, 2009.
- [Wic93] WICKERHAUSER, M. V.: *Smooth localized orthonormal bases*. C. R. Acad. Sci. Paris, 316:423–427, 1993.
- [Wic96] WICKERHAUSER, M. V.: *Adaptive Wavelet-Analysis, Theorie und Software*. Friedr. Vieweg and Sohn Verlagsgesellschaft mbH, Braunschweig / Wiesbaden, 1996.

- [Wil87] WILSON, K. G.: *Generalized Wannier functions*. Preprint, Cornell University, 1987.
- [WNC87] WITTEN, I. H., R. M. NEAL und J. G. CLEARY: *Arithmetic coding for data compression*. Communications of the ACM, 30(6):520–540, 1987.
- [WS05] WENDLAND, W. L. und O. STEINBACH: *Analysis: Integral- und Differentialrechnung, gewöhnliche Differentialgleichungen, komplexe Funktionentheorie*. Vieweg+Teubner, Wiesbaden, 2005.
- [WWJ⁺08] WANG, L., J. WU, L. JIAO, L. ZHANG und G. SHI: *Lossy to lossless image compression based on reversible integer DCT*. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Seiten 1037–1040, 2008.
- [XGO96] XIONG, Z., O. G. GULERYUZ und M. T. ORCHARD: *A DCT-based embedded image coder*. IEEE Signal Processing Letters, 3:289–290, Nov. 1996.
- [Xia97] XIAOLIN, WU: *High-Order Context Modeling and Embedded Conditional Entropy Coding of Wavelet Coefficients for Image Compression*. In: *Proceedings of 31st Asilomar Conference on Signals, Systems and Computers*, Seiten 1378–1382, Nov. 1997.
- [XROZ99] XIONG, Z., K. RAMCHANDRAN, M. T. ORCHARD und Y. Q. ZHANG: *A Comparative Study of DCT- and Wavelet-Based Image Coding*. IEEE Transactions on Circuits and Systems for Video Technology, 9(5), Aug. 1999.
- [XS95] XIA, X.-G. und B. W. SUTER: *A family of two-dimensional nonseparable Malvar wavelets*. Applied and Computational Harmonic Analysis, 2(3):243–256, 1995.
- [Yar98] YAROSLAVSKY, L. P.: *Local adaptive filtering in transform domain for image restoration, enhancement, and target location*. In: E. WENGER, L. I. DIMITROV (Herausgeber): *Sixth International Workshop on Digital Image Processing and Computer Graphics: Applications in Humanities and Natural Sciences*, Band 3346 der Reihe *Proceedings of SPIE*, Seiten 2–17, Vienna, Austria, 1998.
- [YR84] YIP, P. und K. R. RAO: *Fast decimation-in-time-algorithms for the discrete sine and cosine transforms*. Circuits, Systems and Signal Processing, 3(4):387–408, 1984.
- [YW98] YUEN, M. und H. R. WU: *A survey of hybrid MC/DPCM/DCT video coding distortions*. Signal Processing, 70:247–278, 1998.
- [Zak67] ZAK, J.: *Finite translations in solid state physics*. Phys. Rev. Lett., 19:1385–1397, 1967.

Anhang A

Klapp- und Aufklappoperatoren

Aufklappoperator für vierfach überlappende Kosinus-IV-Kosinus-II-Basen, $t \in [0, \frac{1}{2})$

$$\begin{pmatrix} \mathcal{U}_w f_{\mathcal{I}}(t) \\ \mathcal{U}_w f_{\mathcal{I}}(2-t) \\ \mathcal{U}_w f_{\mathcal{I}}(2+t) \\ \mathcal{U}_w f_{\mathcal{I}}(4-t) \\ \mathcal{U}_w f_{\mathcal{I}}(4+t) \\ \vdots \\ \vdots \\ \vdots \\ \mathcal{U}_w f_{\mathcal{I}}(2N-4-t) \\ \mathcal{U}_w f_{\mathcal{I}}(2N-4+t) \\ \mathcal{U}_w f_{\mathcal{I}}(2N-2-t) \\ \mathcal{U}_w f_{\mathcal{I}}(2N-2+t) \\ \mathcal{U}_w f_{\mathcal{I}}(2N-t) \end{pmatrix} = \begin{pmatrix} \begin{array}{cc} w_{\mathcal{I}_0}(t) & w_{\mathcal{I}_1}(-1+t) \\ -w_{\mathcal{I}_0}(2-t) & w_{\mathcal{I}_1}(1-t) \\ -w_{\mathcal{I}_0}(2+t) & -w_{\mathcal{I}_1}(1+t) \end{array} & \begin{array}{cc} 0 & 0 \\ w(-t) & -w(-1-t) \\ w(t) & w(-1+t) \end{array} \\ 0 & \begin{array}{cc} -w(2-t) & w(1-t) \\ -w(2+t) & w(1+t) \end{array} \\ \vdots & \vdots \\ \mathcal{U}_w f_{\mathcal{I}}(2N-4-t) & \begin{array}{cc} 0 & -w(2-t) \\ 0 & -w(2+t) \end{array} \\ \mathcal{U}_w f_{\mathcal{I}}(2N-4+t) & \begin{array}{cc} 0 & -w(2-t) \\ 0 & -w(2+t) \end{array} \\ \mathcal{U}_w f_{\mathcal{I}}(2N-2-t) & \begin{array}{cc} -w(2-t) & w(1-t) \\ -w(2+t) & w(1+t) \end{array} \\ \mathcal{U}_w f_{\mathcal{I}}(2N-2+t) & \begin{array}{cc} -w(2-t) & w(1-t) \\ -w(2+t) & w(1+t) \end{array} \\ \mathcal{U}_w f_{\mathcal{I}}(2N-t) & \begin{array}{cc} -w(2-t) & w(1-t) \\ -w(2+t) & w(1+t) \end{array} \end{pmatrix} \begin{pmatrix} f_{\mathcal{I}}(t) \\ f_{\mathcal{I}}(2-t) \\ f_{\mathcal{I}}(2+t) \\ f_{\mathcal{I}}(4-t) \\ f_{\mathcal{I}}(4+t) \\ \vdots \\ \vdots \\ \vdots \\ f_{\mathcal{I}}(2N-4-t) \\ f_{\mathcal{I}}(2N-4+t) \\ f_{\mathcal{I}}(2N-2-t) \\ f_{\mathcal{I}}(2N-2+t) \\ f_{\mathcal{I}}(2N-t) \end{pmatrix}$$

Abbildung A.9: Rekonstruktion von Signalen, die auf ein Intervall $[0, 2N]$ begrenzt sind; Darstellung des Aufklappoperators für vierfach überlappende Kosinus-IV-Kosinus-II-Basen als Matrix-Vektor-Produkt, $t \in [0, \frac{1}{2})$. Die inneren Fenster sind Translate einer Fensterfunktion $w(t)$.

Aufklappoperator für vierfach überlappende Kosinus-IV-Kosinus-II-Basen, $t \in [\frac{1}{2}, 1)$

$$\begin{pmatrix} \mathcal{U}_w f_{\tau}(t) \\ \mathcal{U}_w f_{\tau}(2-t) \\ \mathcal{U}_w f_{\tau}(2+t) \\ \mathcal{U}_w f_{\tau}(4-t) \\ \mathcal{U}_w f_{\tau}(4+t) \\ \vdots \\ \vdots \\ \vdots \\ \mathcal{U}_w f_{\tau}(2N-4-t) \\ \mathcal{U}_w f_{\tau}(2N-4+t) \\ \mathcal{U}_w f_{\tau}(2N-2-t) \\ \mathcal{U}_w f_{\tau}(2N-2+t) \\ \mathcal{U}_w f_{\tau}(2N-t) \end{pmatrix} = \begin{pmatrix} w_{l_0}(t) & -w(-2+t) & 0 & w_{l_1}(-1+t) & \vdots & \vdots & \vdots & 0 & -w(3-t) & -w(2-t) & w(1-t) & w(-t) & 0 & -w_{r_1}(-2+t) & 0 \\ -w_{l_0}(2-t) & w(-t) & 0 & w_{l_1}(1-t) & \vdots & \vdots & \vdots & 0 & 0 & -w(1+t) & w(t) & w(-2+t) & 0 & -w_{r_1}(-t) & 0 \\ 0 & w(t) & w(-1+t) & -w_{l_1}(1+t) & \vdots & \vdots & \vdots & 0 & 0 & 0 & w(-1+t) & -w(2+t) & 0 & w_{r_1}(t) & 0 \\ 0 & -w(2-t) & w(1-t) & -w_{l_1}(3-t) & \vdots & \vdots & \vdots & 0 & 0 & -w(3-t) & -w(2-t) & w(1-t) & 0 & -w_{r_1}(-1+t) & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{U}_w f_{\tau}(2N-4-t) & 0 & 0 & 0 & \vdots & \vdots & \vdots & 0 & -w(3-t) & -w(2-t) & w(1-t) & w(-t) & 0 & -w_{r_1}(-2+t) & 0 \\ \mathcal{U}_w f_{\tau}(2N-4+t) & 0 & 0 & 0 & \vdots & \vdots & \vdots & 0 & 0 & -w(1+t) & w(t) & w(-2+t) & 0 & -w_{r_1}(-t) & 0 \\ \mathcal{U}_w f_{\tau}(2N-2-t) & 0 & 0 & 0 & \vdots & \vdots & \vdots & 0 & 0 & 0 & w(-1+t) & -w(2+t) & 0 & w_{r_1}(t) & 0 \\ \mathcal{U}_w f_{\tau}(2N-2+t) & 0 & 0 & 0 & \vdots & \vdots & \vdots & 0 & 0 & 0 & w(-1+t) & -w(2+t) & 0 & -w_{r_1}(-1+t) & 0 \\ \mathcal{U}_w f_{\tau}(2N-t) & 0 & 0 & 0 & \vdots & \vdots & \vdots & 0 & 0 & 0 & w(-1+t) & -w(2+t) & 0 & -w_{r_1}(-1+t) & 0 \end{pmatrix} \begin{pmatrix} f_{\tau}(t) \\ f_{\tau}(2-t) \\ f_{\tau}(2+t) \\ f_{\tau}(4-t) \\ f_{\tau}(4+t) \\ \vdots \\ \vdots \\ \vdots \\ f_{\tau}(2N-4-t) \\ f_{\tau}(2N-4+t) \\ f_{\tau}(2N-2-t) \\ f_{\tau}(2N-2+t) \\ f_{\tau}(2N-t) \end{pmatrix}$$

Abbildung A.10: Rekonstruktion von Signalen, die auf ein Intervall $[0, 2N]$ begrenzt sind; Darstellung des Aufklappoperators für vierfach überlappende Kosinus-IV-Kosinus-II-Basen als Matrix-Vektor-Produkt, $t \in [\frac{1}{2}, 1)$. Die inneren Fenster sind Translate einer Fensterfunktion $w(t)$.

Anhang B

Basisfunktionen

In Abschnitt B.1 sind ausgewählte überlappende trigonometrische Basisfunktionen zur Analyse und Reproduktion von Signalen, die auf der reellen Achse \mathbb{R} definiert sind, dargestellt. Bei orthonormalen trigonometrischen Transformationen stimmen die Basisfunktionen auf der Analyseseite mit den Basisfunktionen auf der Rekonstruktionsseite überein. Da die mehrfach überlappenden dualen Basisfunktionen der biorthogonalen trigonometrischen Transformationen eine unendliche Trägerlänge aufweisen, ist von diesen Basisfunktionen nur ein Ausschnitt dargestellt.

Abschnitt B.2 zeigt ausgewählte Basisfunktionen zur Konstruktion von Signalen, die auf einem endlich langen Intervall definiert sind.

Unter Berücksichtigung der Normierungsfaktoren sind in allen Abbildungen jeweils die kontinuierlichen und die für eine Blocklänge von $B = 8$ diskretisierten Basisfunktionen dargestellt. Die diskreten Basisfunktionen ergeben sich aus der Abtastung der kontinuierlichen Basisfunktionen an den Stellen $t = \frac{1}{2B} + \frac{n}{B}$, $n \in \mathbb{Z}$.

B.1 Basisfunktionen auf der reellen Achse

Zweifachüberlappung, Synthese- und Analysebasisfunktionen

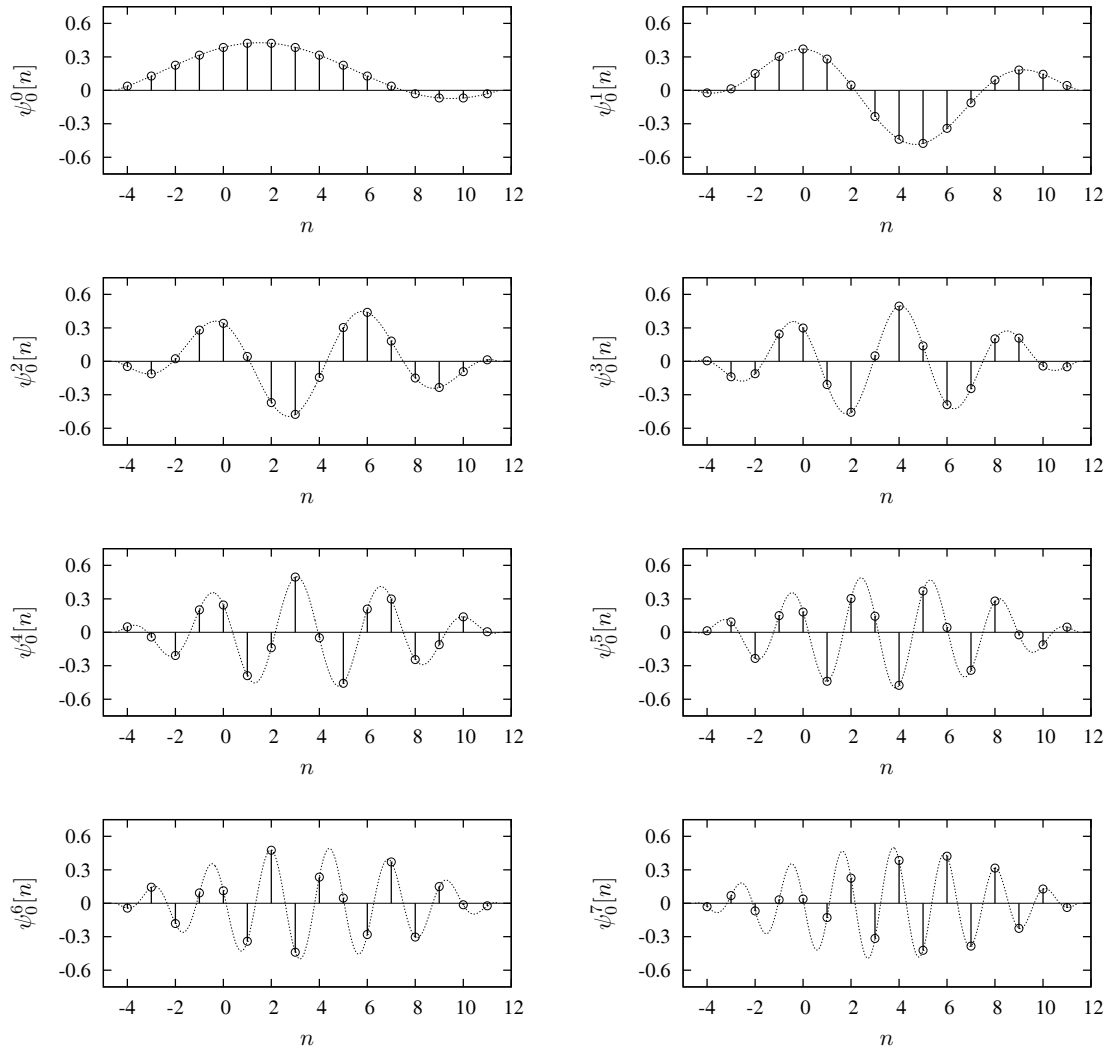


Abbildung B.1: Zweifach überlappende orthonormale Malvar-Basisfunktionen zur Signalrekonstruktion und Signalanalyse.

Dreifachüberlappung

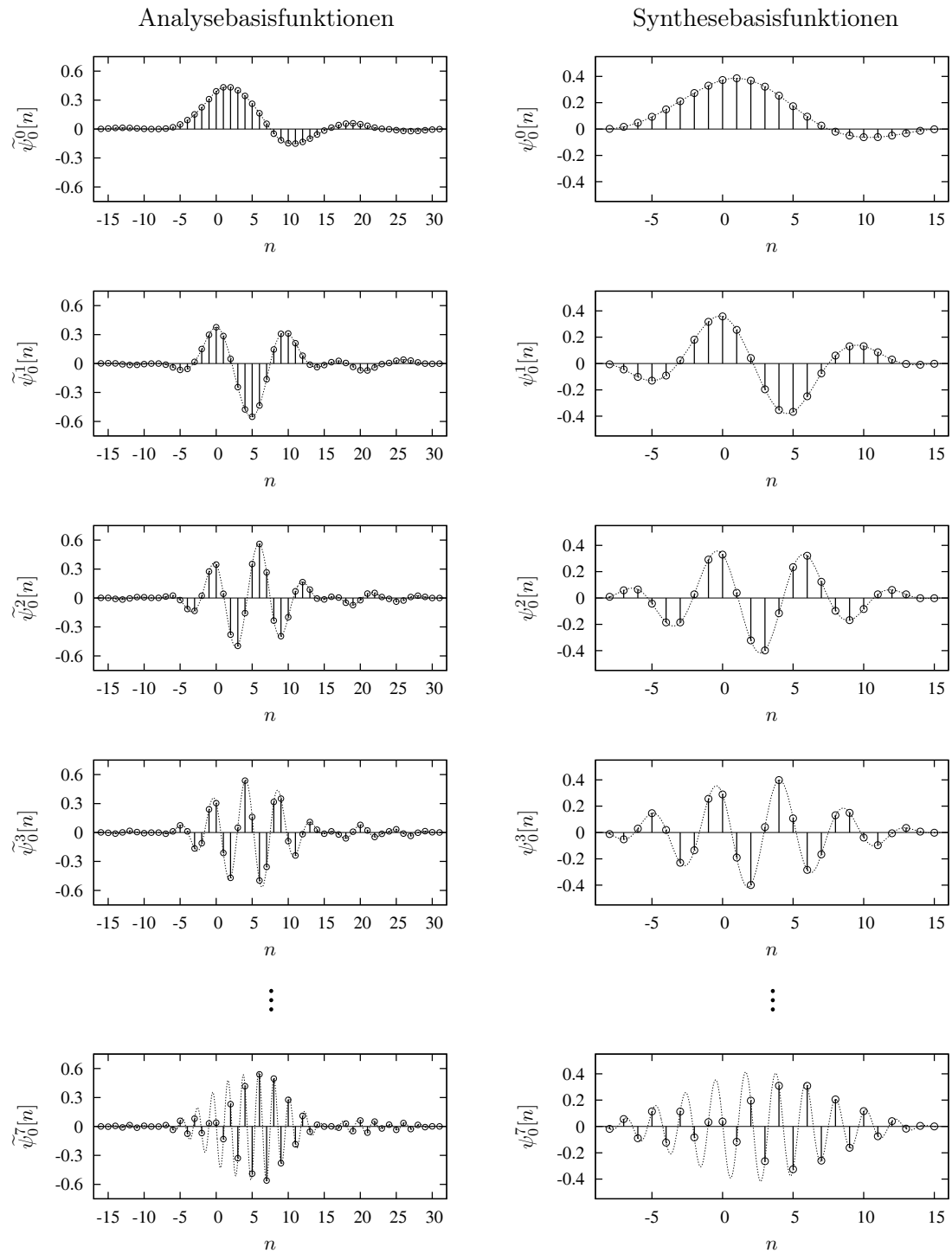


Abbildung B.2: Basisfunktionen zur Signalanalyse (links) und Signalrekonstruktion (rechts) (Verwendung von Fenster 3).

Vierfachüberlappung

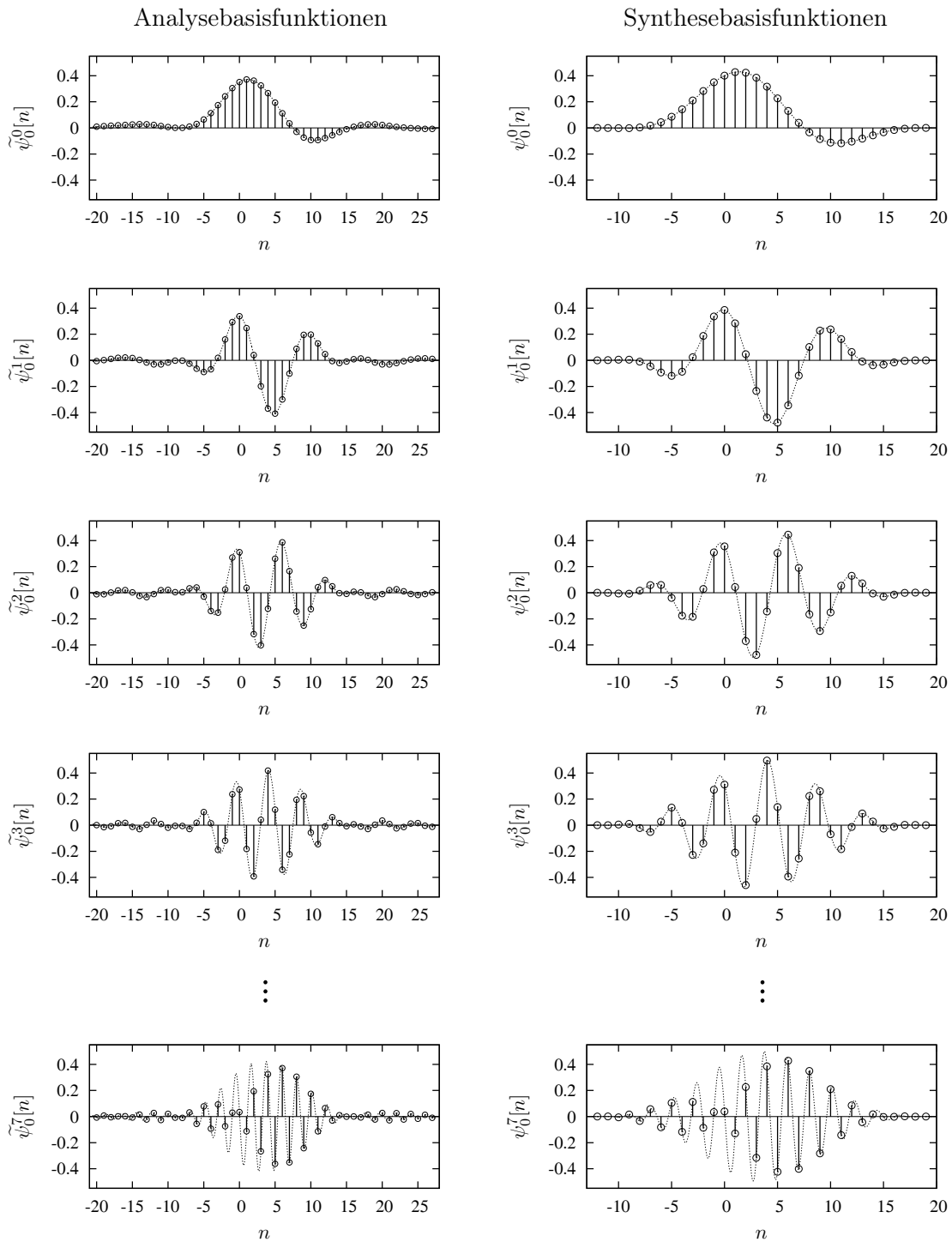


Abbildung B.3: Basisfunktionen zur Signalanalyse (links) und Signalrekonstruktion (rechts).

Sechsfachüberlappung

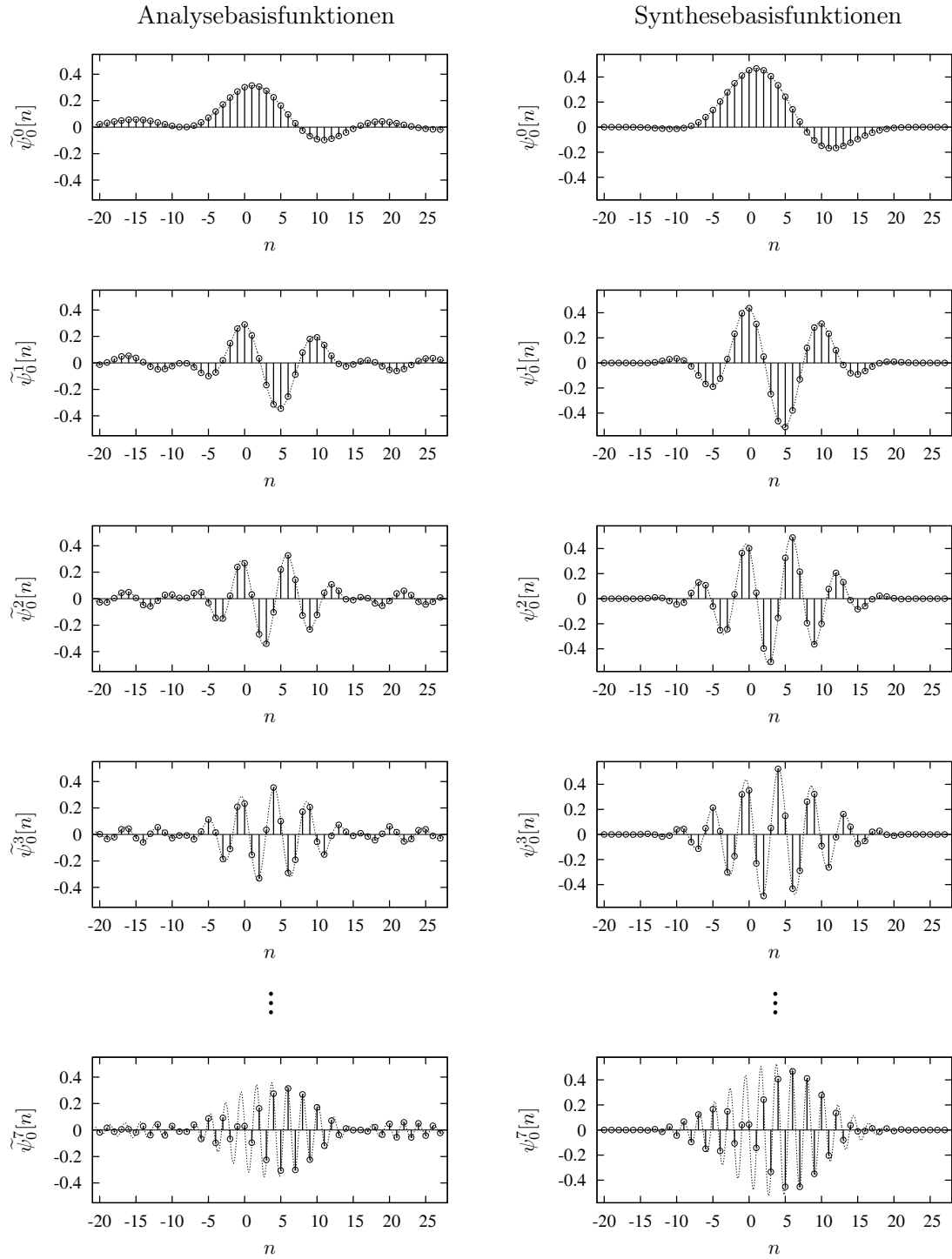


Abbildung B.4: Basisfunktionen zur Signalanalyse (links) und Signalrekonstruktion (rechts).

B.2 Basisfunktionen auf dem Intervall

Zweifachüberlappung

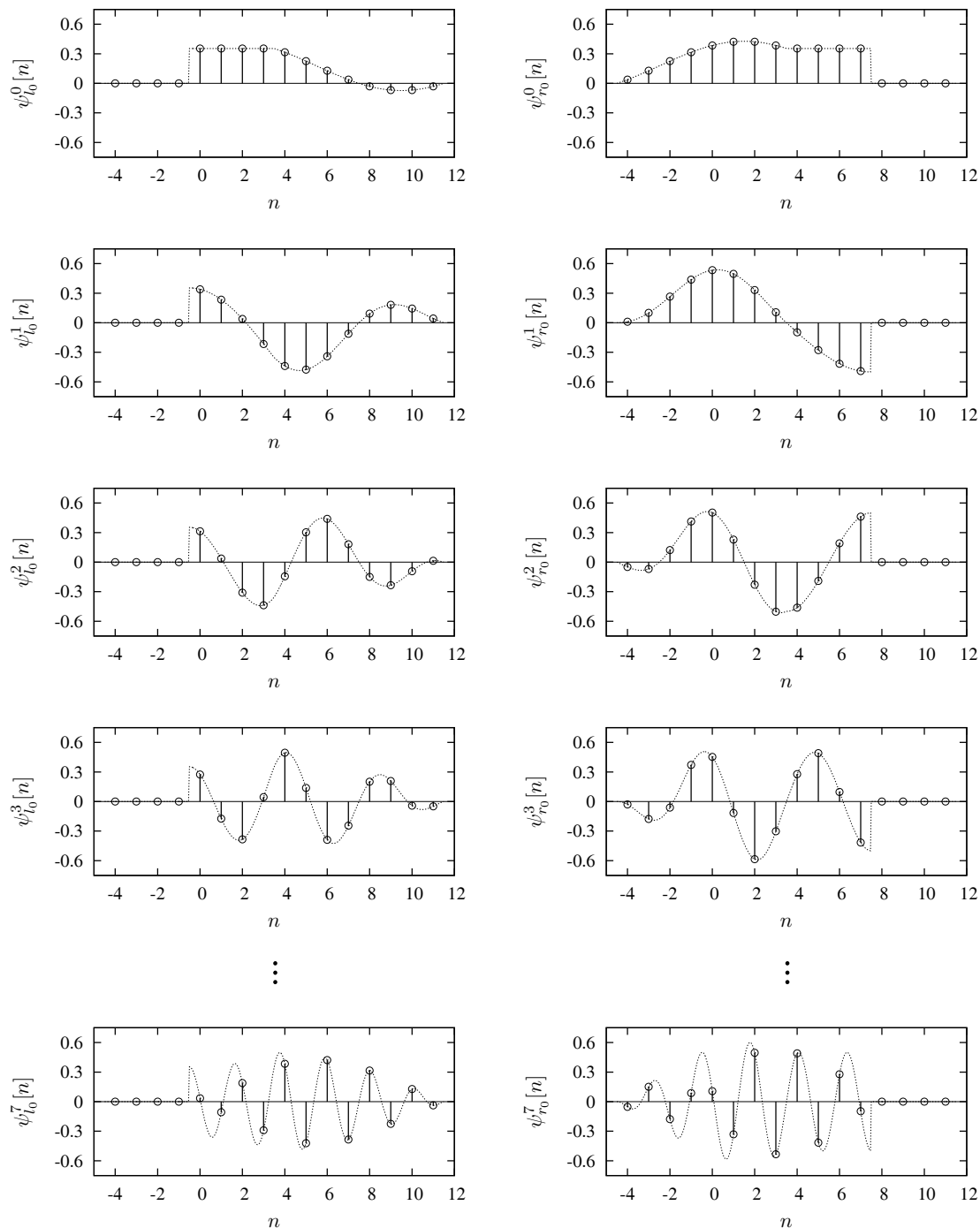


Abbildung B.5: Randbasisfunktionen ψ_{l_0} zur Signalrekonstruktion am linken Signalrand und ψ_{r_0} zur Signalrekonstruktion am rechten Signalrand.

Dreifachüberlappung

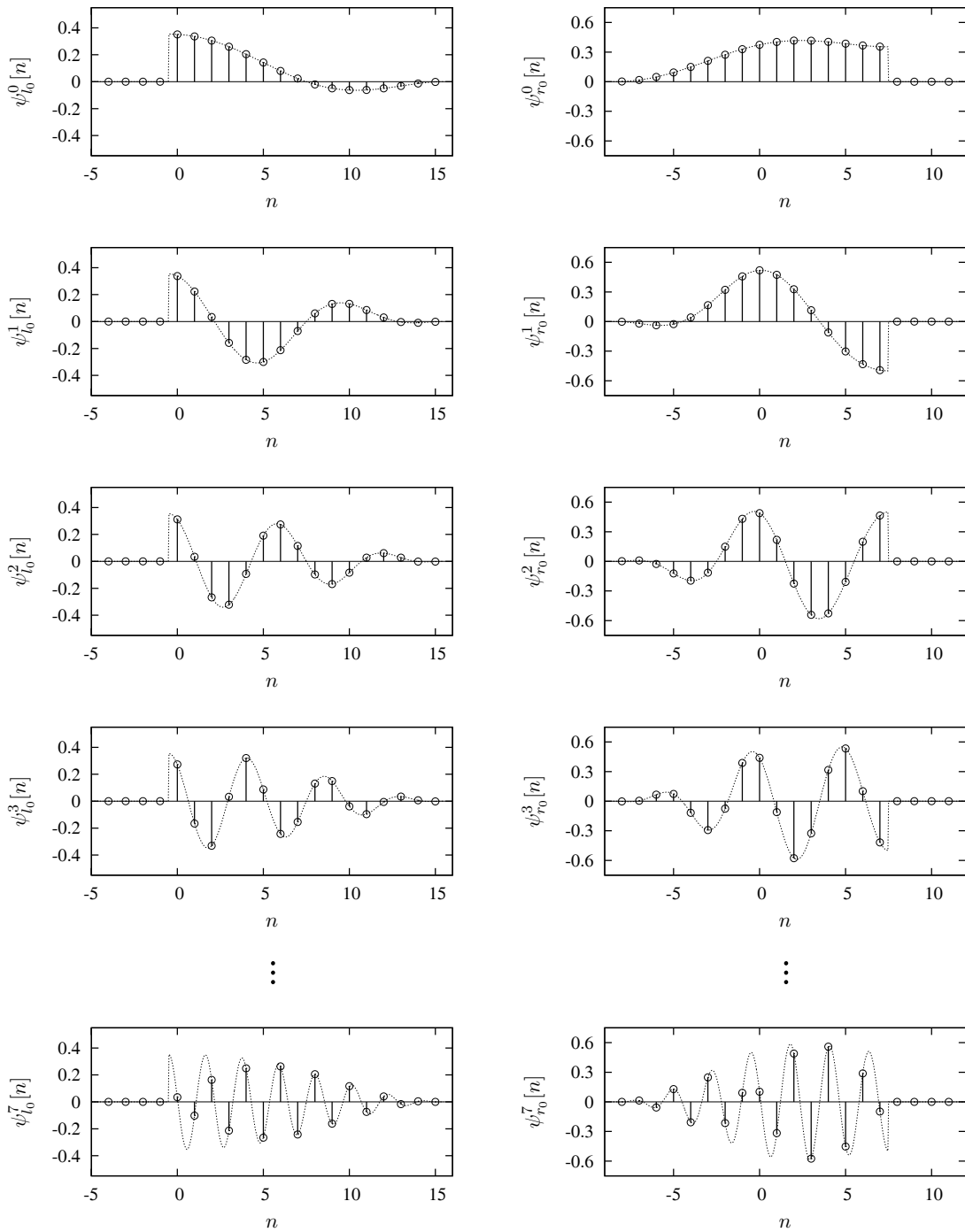


Abbildung B.6: Randbasisfunktionen ψ_{l_0} zur Signalrekonstruktion am linken Signalrand und ψ_{r_0} zur Signalrekonstruktion am rechten Signalrand.

Vierfachüberlappung

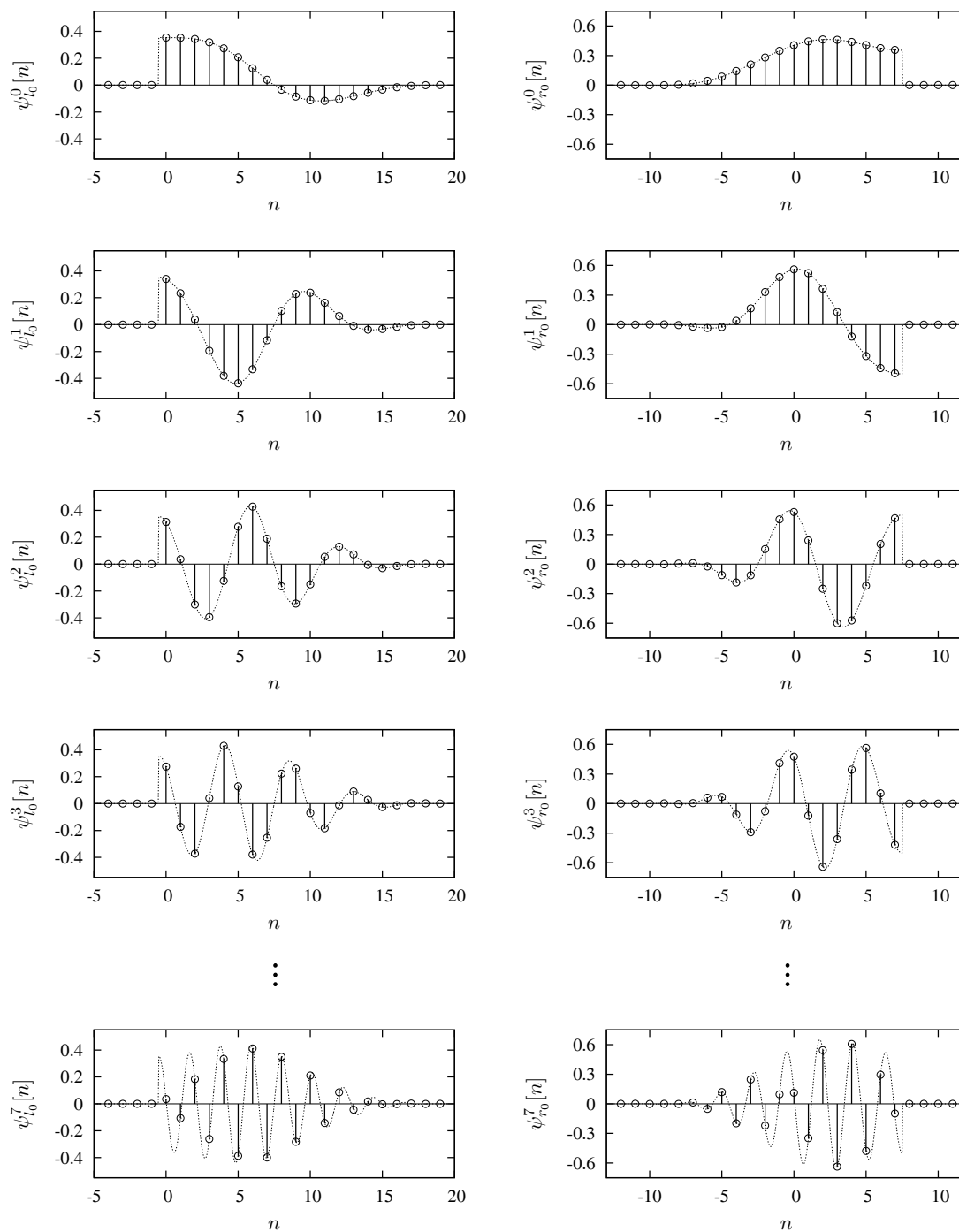


Abbildung B.7: Randbasisfunktionen ψ_{l_0} zur Signalrekonstruktion am linken Signalrand und ψ_{r_0} zur Signalrekonstruktion am rechten Signalrand.

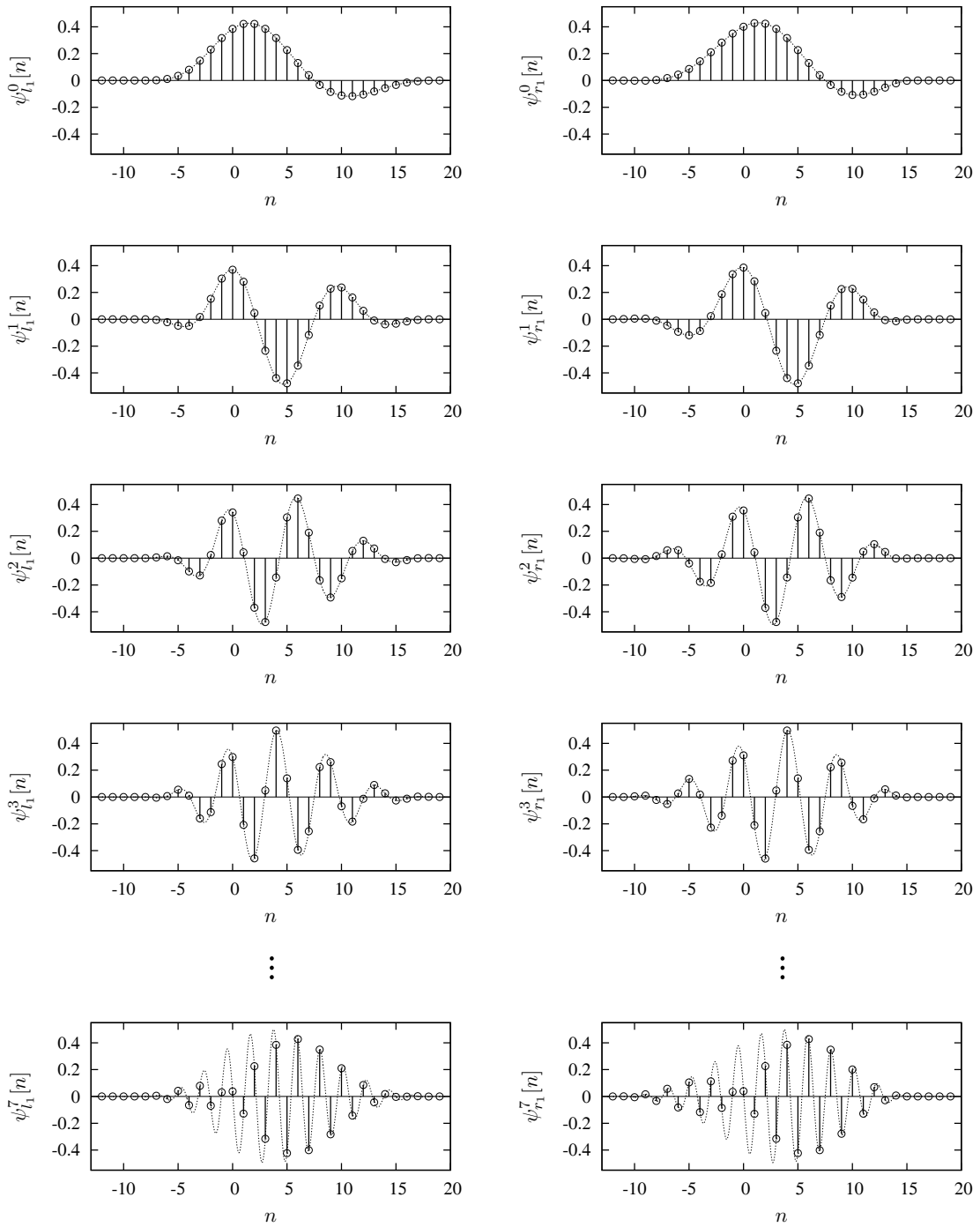


Abbildung B.8: Randbasisfunktionen ψ_{l_1} zur Signalrekonstruktion am linken Signalrand und ψ_{r_1} zur Signalrekonstruktion am rechten Signalrand.

Sechsfachüberlappung

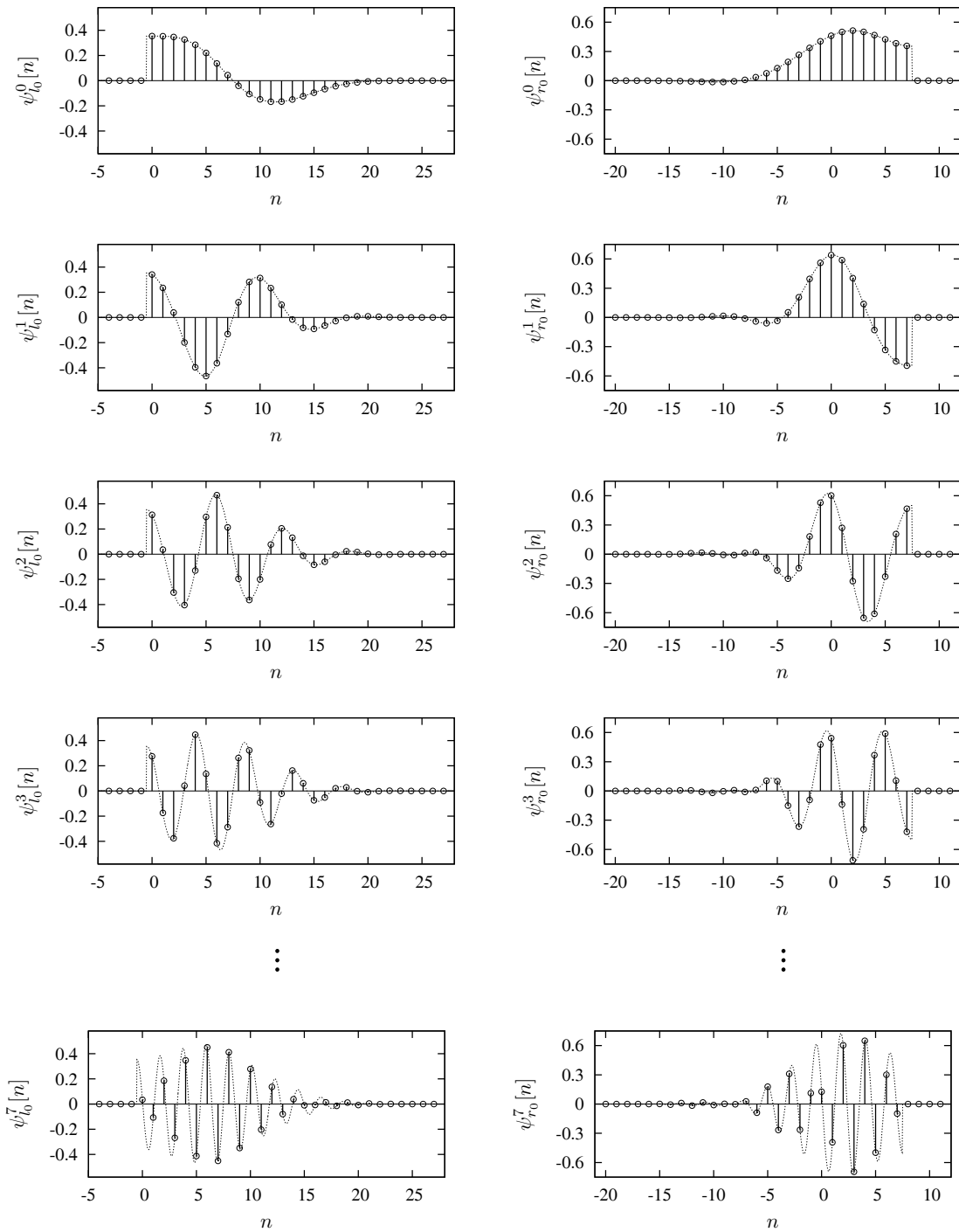


Abbildung B.9: Randbasisfunktionen ψ_{l_0} zur Signalrekonstruktion am linken Signalrand und ψ_{r_0} zur Signalrekonstruktion am rechten Signalrand.

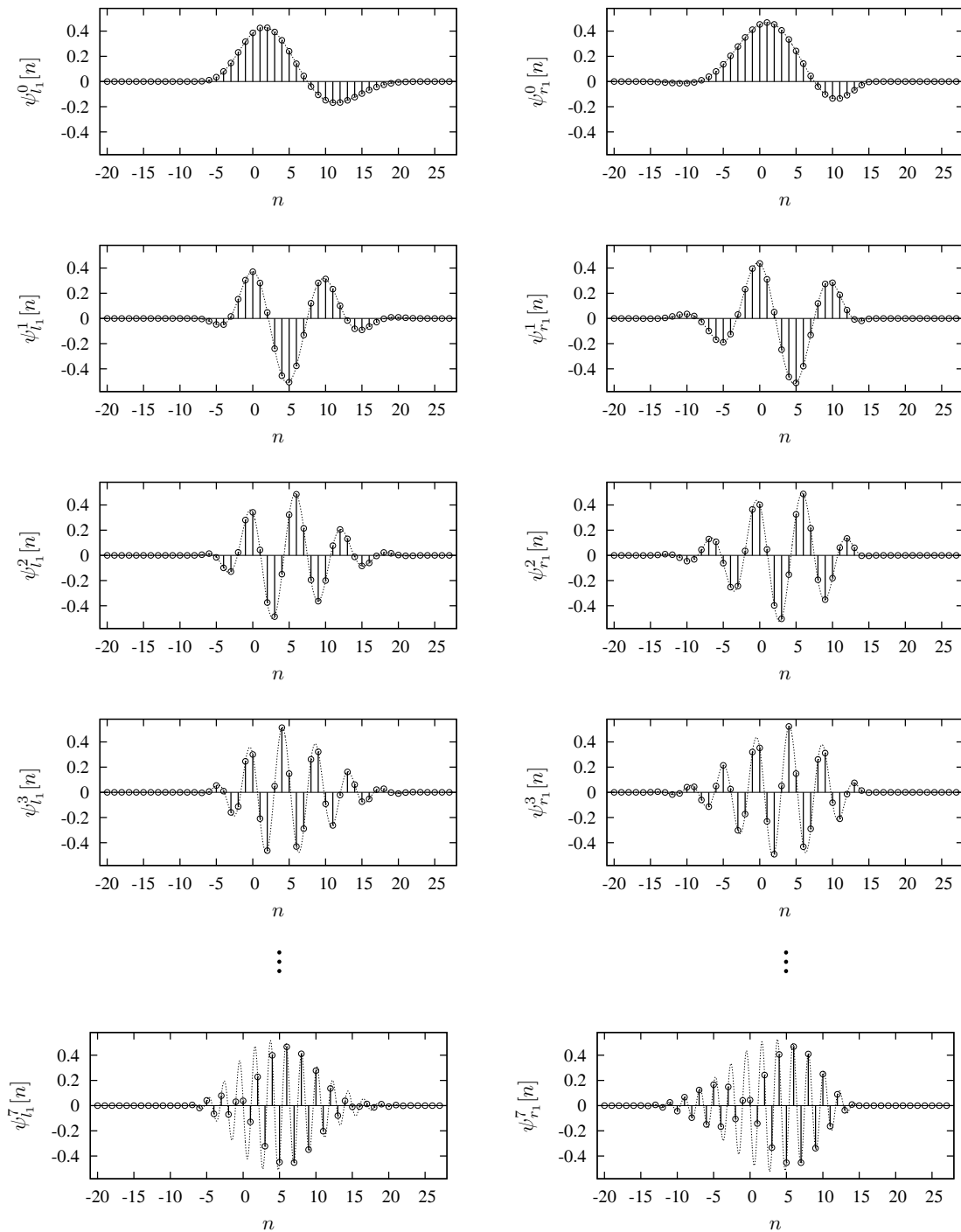


Abbildung B.10: Randbasisfunktionen ψ_{l_1} zur Signalrekonstruktion am linken Signalrand und ψ_{r_1} zur Signalrekonstruktion am rechten Signalrand. Auf eine Darstellung der Basisfunktionen ψ_{l_2} und ψ_{r_2} wird verzichtet, da sie sich nur sehr geringfügig von den inneren Basisfunktionen (vgl. Abbildung B.4) unterscheiden und diese Unterschiede in dem hier betrachteten Darstellungsbereich kaum sichtbar sind.

Anhang C

Amplitudengänge

Filterbankrealisierung und Übertragungsfunktionen

Die im Rahmen dieser Arbeit betrachteten trigonometrischen Transformationen lassen sich als Teilbandsysteme interpretieren (vgl. Abbildung C.1). Bei den Teilbandfiltern handelt es sich um frequenzverschobene Versionen eines Tiefpassprototyps, der sich aus dem Wichtungsfenster ergibt. Die Frequenzverschiebung wird durch die Multiplikation der Fensterfunktion mit trigonometrischen Funktionen verschiedener Frequenz erzielt. Man bezeichnet ein solches System von Teilbandfiltern auch als kosinus-modulierte Filterbank.

Durch die Betrachtung der Transformation als Filterbank wird sichtbar, welche Frequenzbänder des Signals die einzelnen Transformationskoeffizienten repräsentieren. Des Weiteren können mit Hilfe der Übertragungsfunktionen Aussagen zur Leistungsfähigkeit der Transformation in der Bilddatenkompression abgeleitet werden. Weisen die Amplitudengänge der Übertragungsfunktionen deutliche Nebenmaxima auf, so deutet dies auf eine ungenügende Frequenztrennung der Koeffizienten untereinander hin [Ohm95]. Um die Korrelation der Transformationskoeffizienten zu reduzieren, sind die Nebenmaxima im Spektralbereich zu verringern. Ein günstigerer Verlauf des Spektrums kann durch die Verwendung längerer Filter bzw. größerer Überlappungsfaktoren erzielt werden.

Zur Betrachtung der Kosinustransformation und der überlappenden Kosinustransformation von Signalen als Filterbankrealisierung werden die an den Stellen $t = \frac{1}{2B} + \frac{m}{B}$, $m \in \mathbb{Z}$ abgetasteten trigonometrischen Basisfunktionen als Impulsantworten der Tief-, Band- und Hochpassfilter aufgefasst. Nachfolgend sind für eine Blockgröße von $B = 8$ die Amplitudengänge der Filterbankrealisierungen für die untersuchten Überlappungsfaktoren bzw. Fensterfunktionen dargestellt.

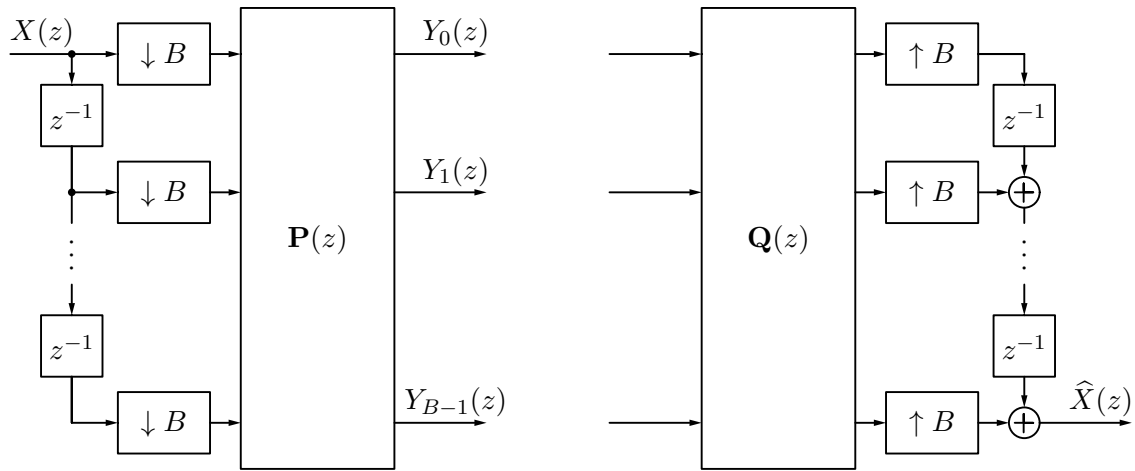


Abbildung C.1: Filterbankrealisierung von trigonometrischen Transformationen.

Diskrete Kosinus-Transformation und zweifach überlappende Malvar-Basis

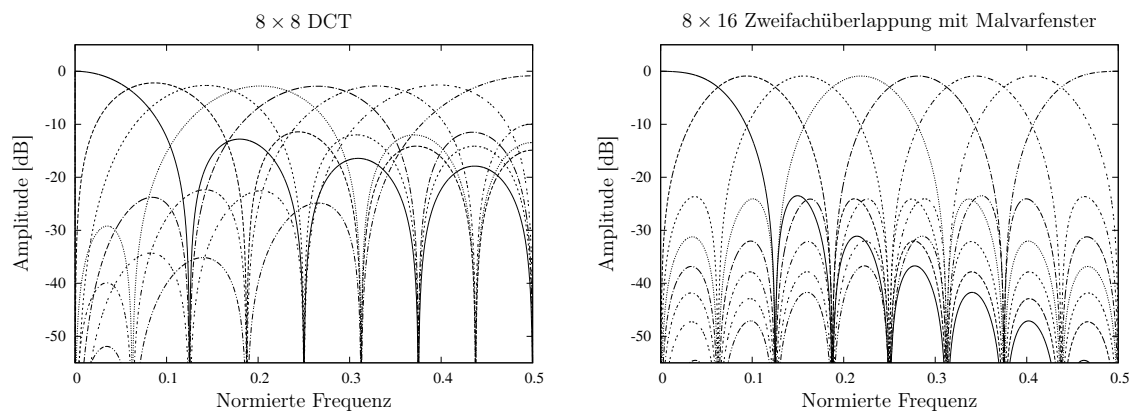


Abbildung C.2: Amplitudengänge der Filterbankrealisierungen zur Transformation von Signalen mit trigonometrischen Basen für die Kosinus-Transformation (links) und die Malvar-Basis (rechts).

Dreifachüberlappung

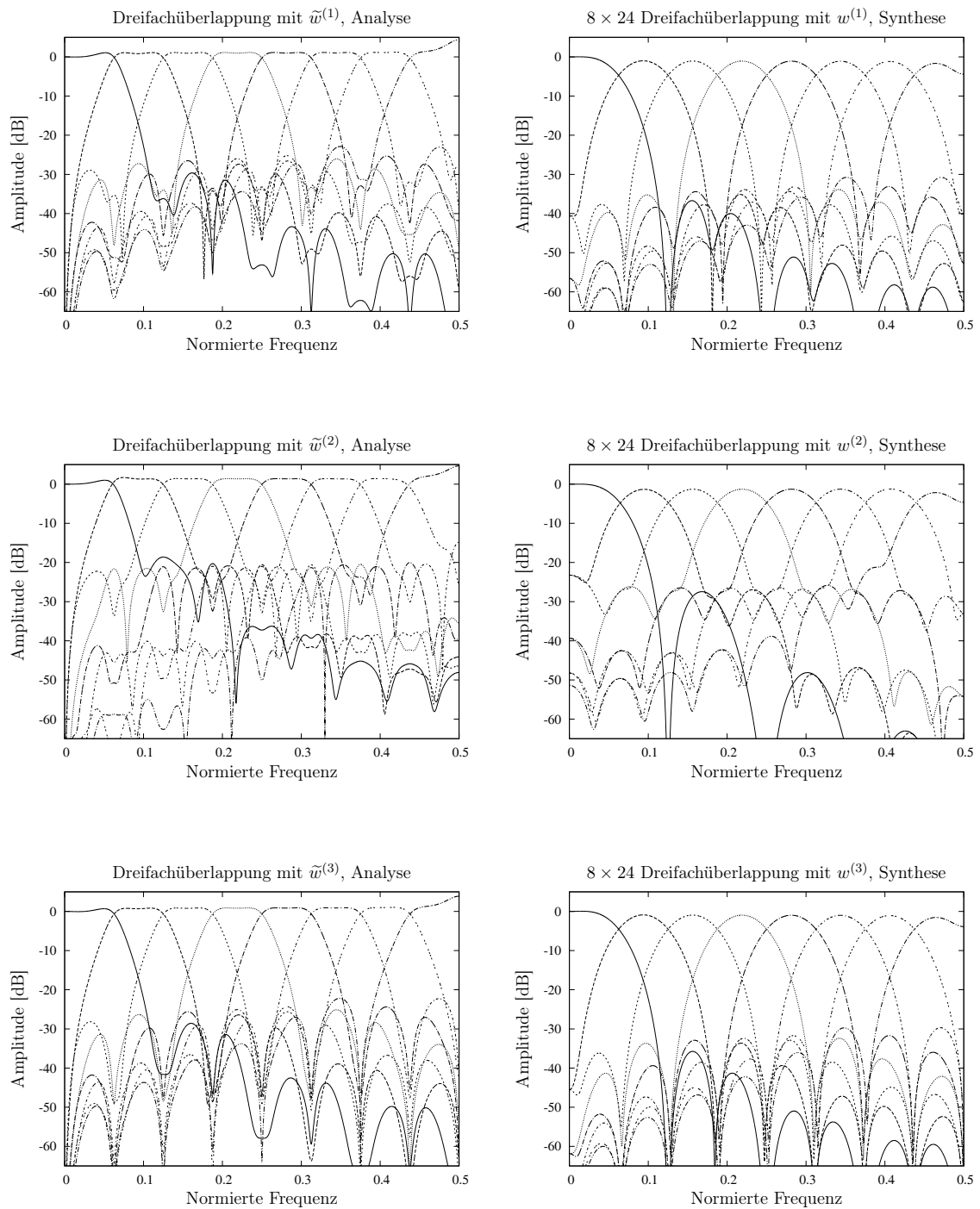


Abbildung C.3: Amplitudengänge der Filterbankrealisierungen zur Transformation von Signalen mit dreifach überlappenden lokalen trigonometrischen Basen für die Signalanalyse (links) und die Signalrekonstruktion (rechts).

Vierfachüberlappung

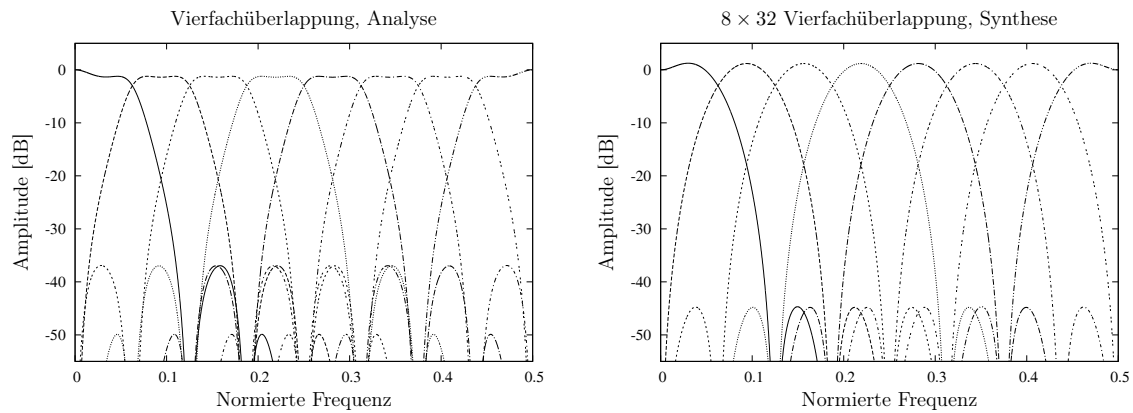


Abbildung C.4: Amplitudengänge der Filterbankrealisierungen zur Transformation von Signalen mit vierfach überlappenden lokalen trigonometrischen Basen für die Signalanalyse (links) und die Signalrekonstruktion (rechts).

Sechsfachüberlappung

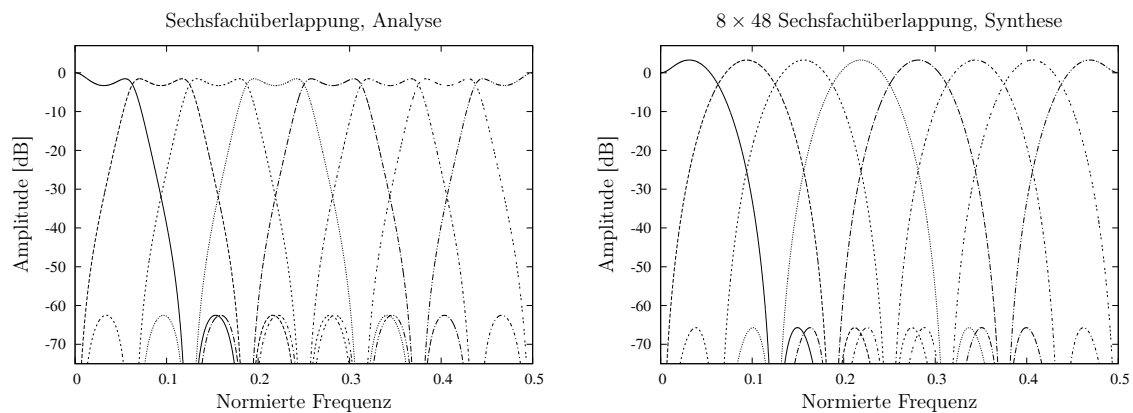


Abbildung C.5: Amplitudengänge der Filterbankrealisierungen zur Transformation von Signalen mit sechsfach überlappenden lokalen trigonometrischen Basen für die Signalanalyse (links) und die Signalrekonstruktion (rechts).

Anhang D

Duale Randfenster

Dreifachüberlappung

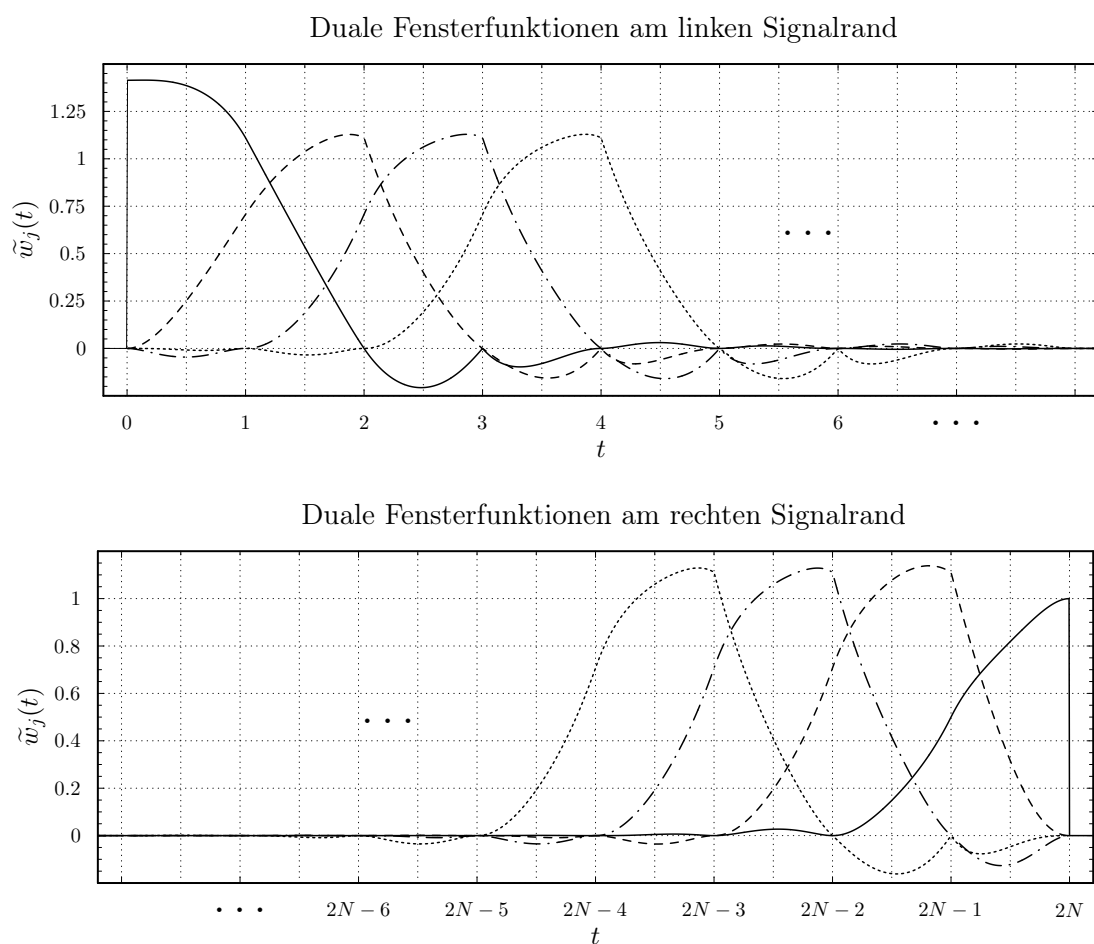


Abbildung D.1: Fensterfunktionen zur Konstruktion von dualen Basisfunktionen für die Analyse von Signalen, die auf einem Intervall $[0, 2N]$ definiert sind. Eine Konstante kann mit den dualen Basisfunktionen $\tilde{\psi}_j^0$ aufgelöst werden.

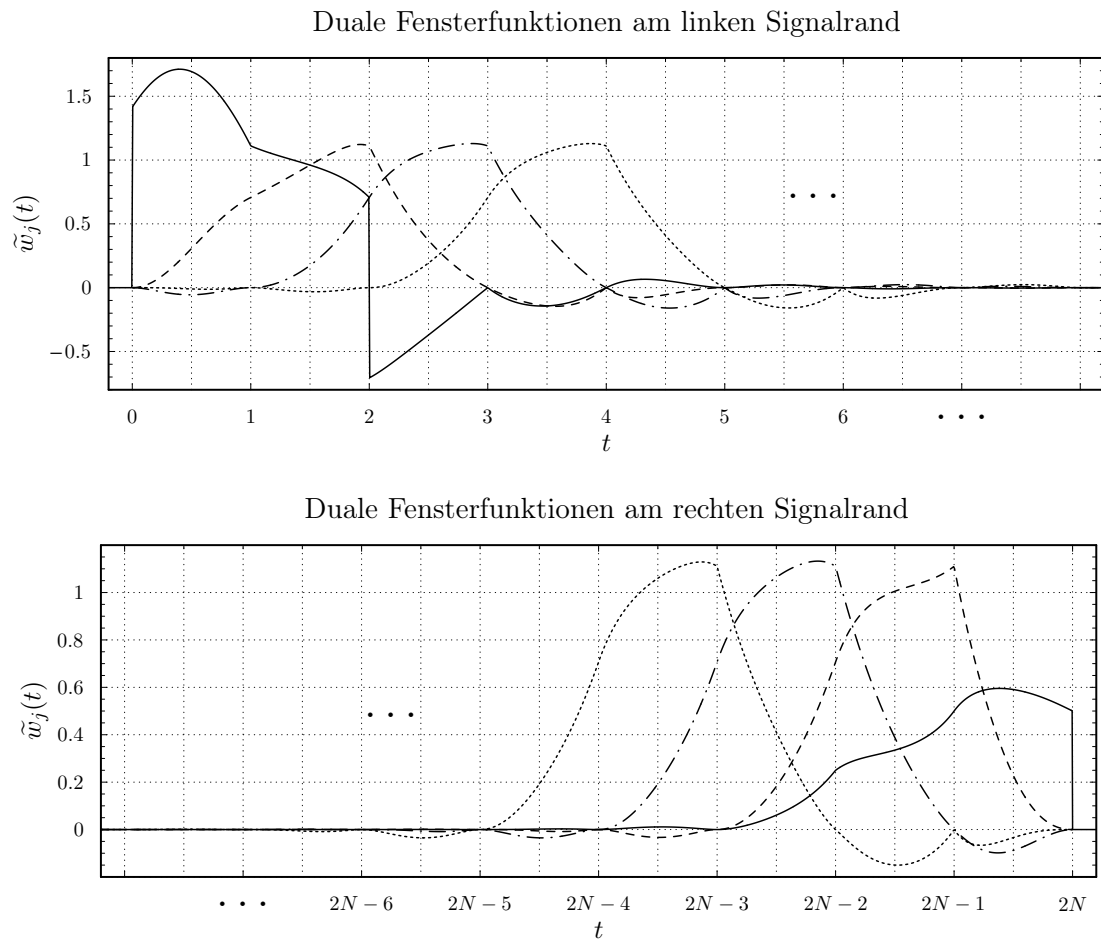


Abbildung D.2: Fensterfunktionen zur Konstruktion von dualen Basisfunktionen für die Analyse von Signalen, die auf einem Intervall $[0, 2N]$ definiert sind. Eine Lineare kann mit den dualen Basisfunktionen $\tilde{\psi}_j^0$ aufgelöst werden.

Vierfachüberlappung

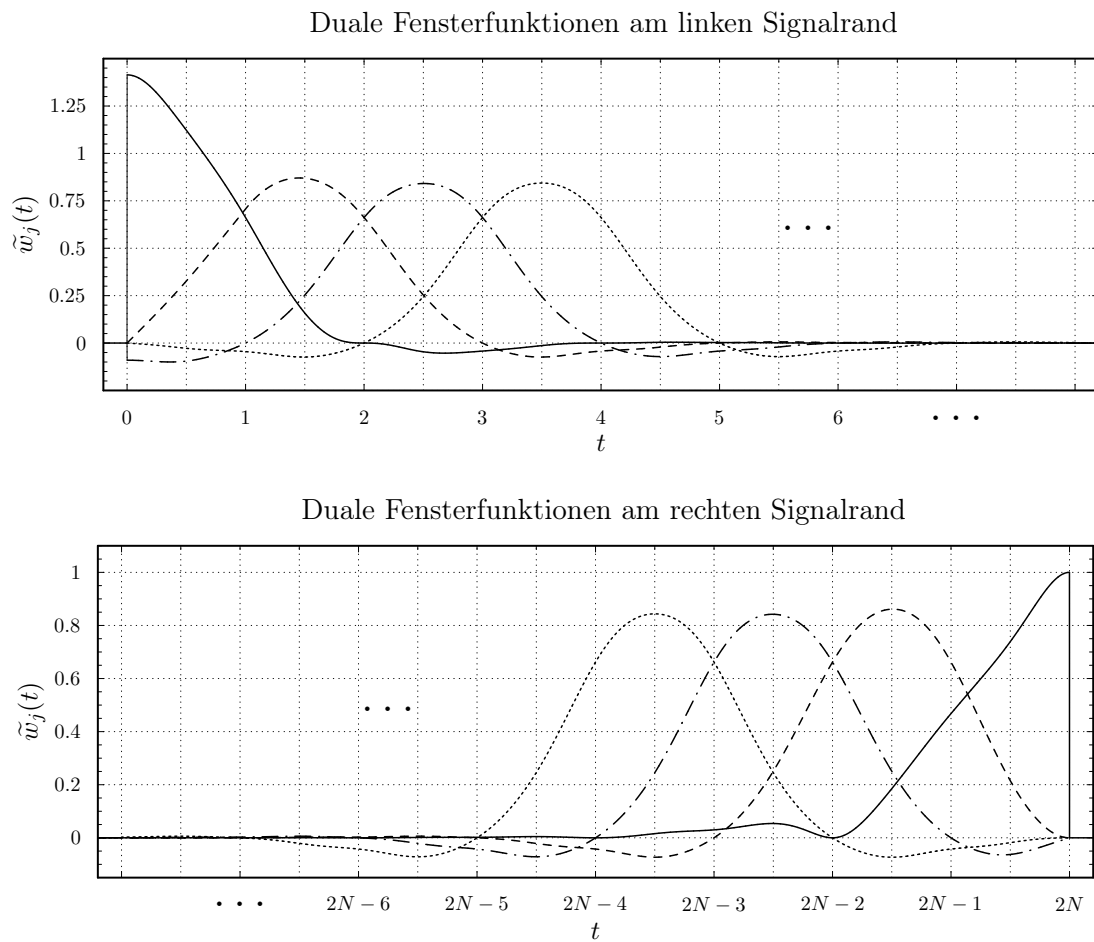


Abbildung D.3: Fensterfunktionen zur Konstruktion von dualen Basisfunktionen für die Analyse von Signalen, die auf einem Intervall $[0, 2N]$ definiert sind, nach Ansatz 1. Eine Konstante kann mit den dualen Basisfunktionen $\tilde{\psi}_j^0$ aufgelöst werden.

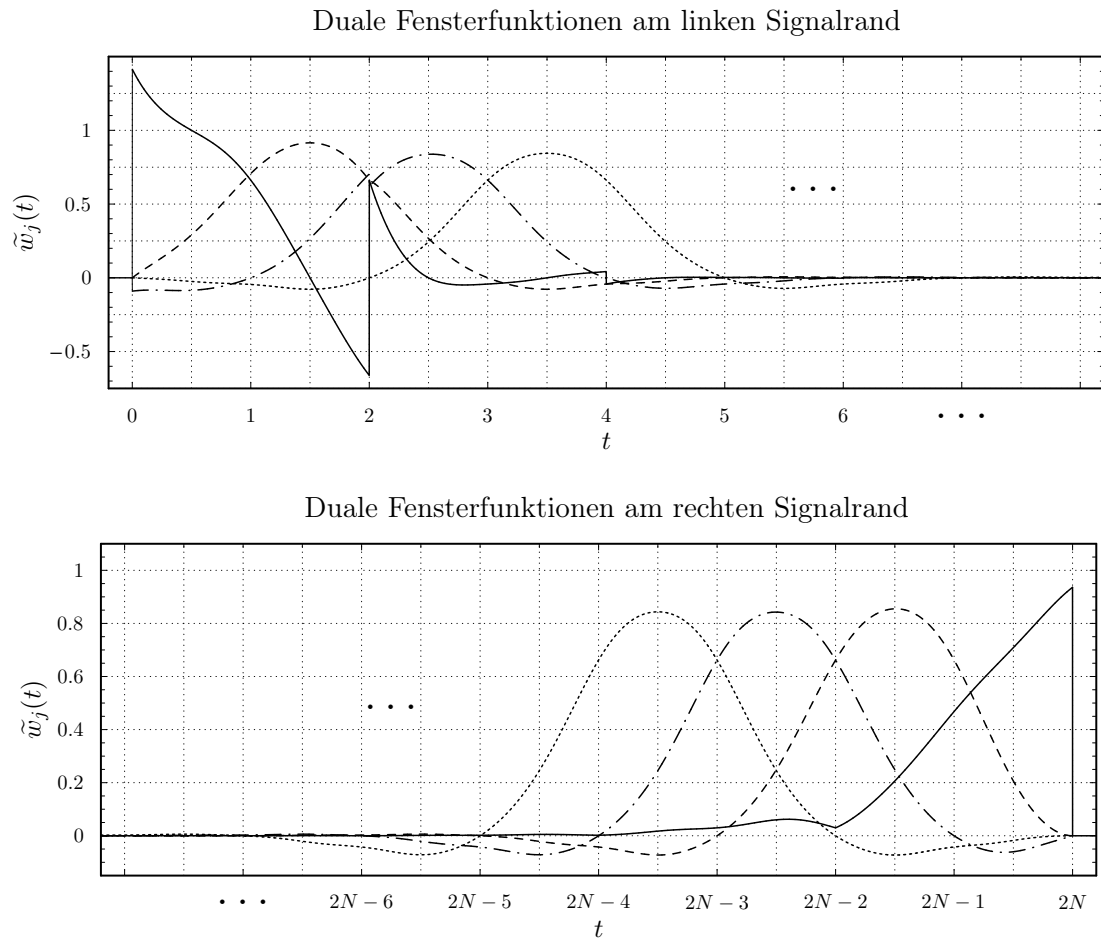


Abbildung D.4: Fensterfunktionen zur Konstruktion von dualen Basisfunktionen für die Analyse von Signalen, die auf einem Intervall $[0, 2N]$ definiert sind, nach Ansatz 2. Eine Konstante kann mit den dualen Basisfunktionen $\tilde{\psi}_j^0$ aufgelöst werden.

Sechsfachüberlappung

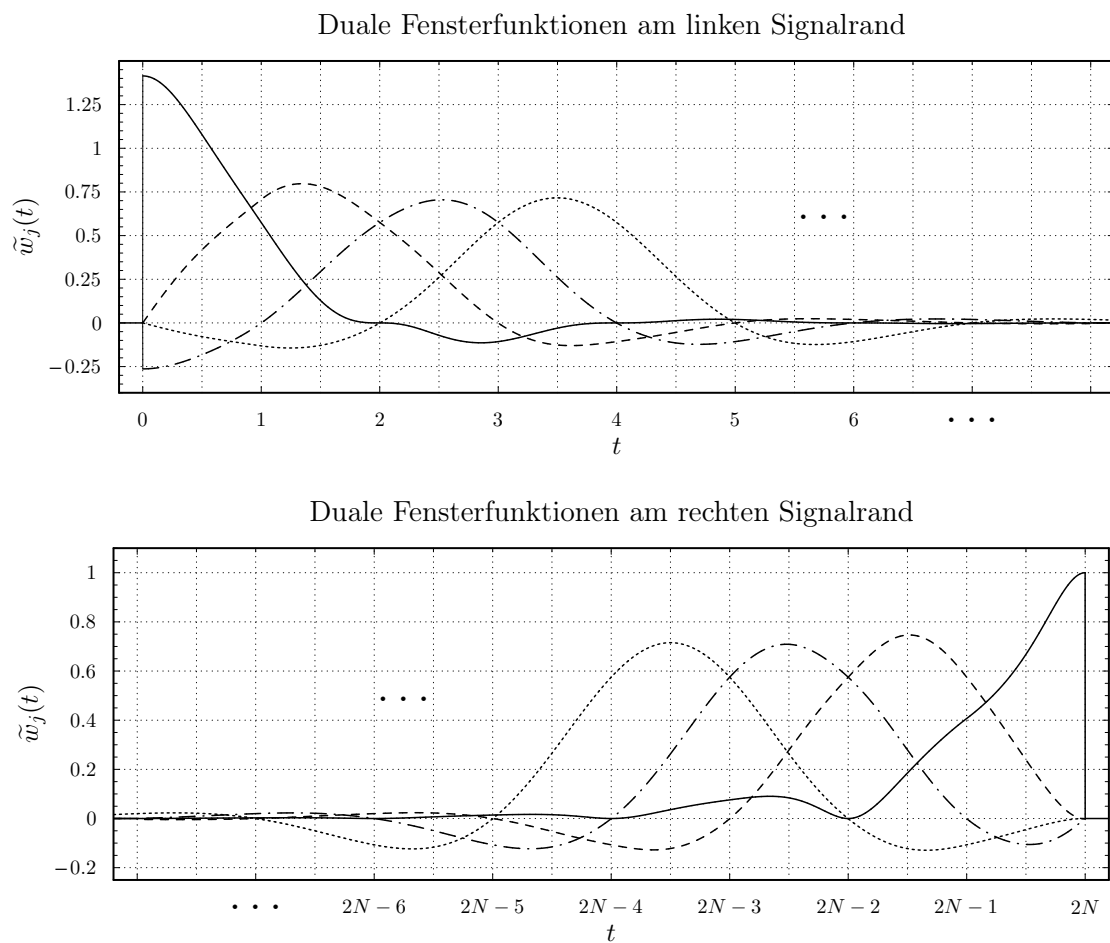


Abbildung D.5: Fensterfunktionen zur Konstruktion von dualen Basisfunktionen für die Analyse von Signalen, die auf einem Intervall $[0, 2N]$ definiert sind. Eine Konstante kann mit den dualen Basisfunktionen $\tilde{\psi}_j^0$ aufgelöst werden.

Anhang E

Algorithmen

Die nachfolgend eingeführten Algorithmen sind an die im Rahmen dieser Arbeit vorgestellten Fensterfunktionen angepasst. Für allgemeinere Ansätze können die in den Abschnitten 4.2 und 4.3 entwickelten allgemeinen Algorithmen verwendet werden (vgl. Abbildungen 4.2, 4.5 und 4.6).

E.1 Algorithmen für die Zweifachüberlappung

Aufklappoperator

Vektoren zur kompakten Speicherung der Abbildungsmatrizen

Da Nullkoeffizienten in den äußeren Nebendiagonalen in Abhängigkeit vom Parameter m auftreten, sind bei der Realisierung der Algorithmen zwei Fälle zu unterscheiden.

Für $m = 0, 1, \dots, \frac{B}{2} - 1$:

$$\vec{a}_0^{(m)} = \begin{pmatrix} a_{0,0}^{(m)} \\ a_{0,1}^{(m)} \\ a_{0,2}^{(m)} \end{pmatrix}, \quad \vec{a}_1^{(m)} = \begin{pmatrix} a_{1,0}^{(m)} \end{pmatrix}. \quad (\text{E.1})$$

Für $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$:

$$\vec{a}_0^{(m)} = \begin{pmatrix} a_{0,0}^{(m)} \\ a_{0,1}^{(m)} \end{pmatrix}, \quad \vec{a}_1^{(m)} = \begin{pmatrix} a_{1,0}^{(m)} \\ a_{1,1}^{(m)} \end{pmatrix}. \quad (\text{E.2})$$

Die Speicherstruktur ist in den Abbildungen E.3 und E.4 veranschaulicht.

Initialisierung der Koeffizientenvektoren

Für $m = 0, 1, \dots, \frac{B}{2} - 1$:

$$\begin{aligned} a_{0,0}^{(m)} &= w_{l_0}[m], & a_{0,2}^{(m)} &= w_{r_0}[B - m - 1], \\ a_{0,1}^{(m)} &= w[m], & a_{1,0}^{(m)} &= w[-m - 1]. \end{aligned} \quad (\text{E.3})$$

Für $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$:

$$\begin{aligned} a_{0,0}^{(m)} &= w[m], & a_{1,0}^{(m)} &= w[-B + m], \\ a_{0,1}^{(m)} &= w_{r_0}[B - m - 1], & a_{1,1}^{(m)} &= w_{r_0}[-B + m]. \end{aligned} \quad (\text{E.4})$$

Für $m = 0, 1, \dots, \frac{B}{2} - 1$

(1) *Aufklappoperator - linker Rand*

$$f_0^{(m)} = a_{0,0}^{(m)} f_{\mathcal{T} 0}^{(m)}$$

(2) *Aufklappoperator - innere Fenster*

For $n = 1$ **To** $N - 1$

$$\begin{aligned} f_{2n-1}^{(m)} &= a_{0,1}^{(m)} f_{\mathcal{T} 2n-1}^{(m)} + a_{1,0}^{(m)} f_{\mathcal{T} 2n}^{(m)} \\ f_{2n}^{(m)} &= -a_{1,0}^{(m)} f_{\mathcal{T} 2n-1}^{(m)} + a_{0,1}^{(m)} f_{\mathcal{T} 2n}^{(m)} \end{aligned}$$

End

(3) *Aufklappoperator - rechter Rand*

$$f_{2N-1}^{(m)} = a_{0,2}^{(m)} f_{\mathcal{T} 2N-1}^{(m)}$$

Für $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$

(1 & 2) *Aufklappoperator - linker Rand und innere Fenster*

For $n = 0$ **To** $N - 2$

$$\begin{aligned} f_{2n}^{(m)} &= a_{0,0}^{(m)} f_{\mathcal{T} 2n}^{(m)} + a_{1,0}^{(m)} f_{\mathcal{T} 2n+1}^{(m)} \\ f_{2n+1}^{(m)} &= -a_{1,0}^{(m)} f_{\mathcal{T} 2n}^{(m)} + a_{0,0}^{(m)} f_{\mathcal{T} 2n+1}^{(m)} \end{aligned}$$

End

(3) *Aufklappoperator - rechter Rand*

$$\begin{aligned} f_{2N-2}^{(m)} &= a_{0,0}^{(m)} f_{\mathcal{T} 2N-2}^{(m)} + a_{1,1}^{(m)} f_{\mathcal{T} 2N-1}^{(m)} \\ f_{2N-1}^{(m)} &= -a_{1,0}^{(m)} f_{\mathcal{T} 2N-2}^{(m)} + a_{0,1}^{(m)} f_{\mathcal{T} 2N-1}^{(m)} \end{aligned}$$

Abbildung E.1: Algorithmus zur effizienten Realisierung des Aufklappoperators für zweifach überlappende Fenster, $m = 0, 1, \dots, B - 1$; $B = \text{Blocklänge}$, $L = 2NB = \text{Signallänge}$, $O_{\mathcal{U}} = 3 - 2\frac{B}{L}$ arithmetische Operationen pro Signalwert.

Klappoperator

Vektoren zur kompakten Speicherung der Koeffizientenmatrizen

Da Nullkoeffizienten in den äußeren Nebendiagonalen in Abhängigkeit vom Parameter m auftreten, sind bei der Realisierung der Algorithmen zwei Fälle zu unterscheiden.

Für $m = 0, 1, \dots, \frac{B}{2} - 1$:

$$\vec{a}_0^{(m)} = \begin{pmatrix} a_{0,0}^{(m)} \\ a_{0,1}^{(m)} \\ a_{0,2}^{(m)} \end{pmatrix}, \quad \vec{a}_1^{(m)} = \begin{pmatrix} a_{1,0}^{(m)} \end{pmatrix}. \quad (\text{E.5})$$

Für $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$:

$$\vec{a}_0^{(m)} = \begin{pmatrix} a_{0,0}^{(m)} \\ a_{0,1}^{(m)} \end{pmatrix}, \quad \vec{a}_1^{(m)} = \begin{pmatrix} a_{1,0}^{(m)} \\ a_{1,1}^{(m)} \end{pmatrix}. \quad (\text{E.6})$$

Die Speicherstruktur ist in den Abbildungen E.5 und E.6 veranschaulicht.

Initialisierung der Koeffizientenvektoren

Für $m = 0, 1, \dots, \frac{B}{2} - 1$:

$$\begin{aligned} a_{0,0}^{(m)} &= \tilde{w}_{l_0}[m], & a_{0,2}^{(m)} &= \tilde{w}_{r_0}[B - m - 1], \\ a_{0,1}^{(m)} &= \tilde{w}[m], & a_{1,0}^{(m)} &= \tilde{w}[-m - 1]. \end{aligned} \quad (\text{E.7})$$

Für $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$:

$$\begin{aligned} a_{0,0}^{(m)} &= \tilde{w}[m], & a_{1,0}^{(m)} &= \tilde{w}[-B + m], \\ a_{0,1}^{(m)} &= \tilde{w}_{r_0}[B - m - 1], & a_{1,1}^{(m)} &= \tilde{w}_{r_0}[-B + m]. \end{aligned} \quad (\text{E.8})$$

```

Für  $m = 0, 1, \dots, \frac{B}{2} - 1$ 

(1) Klappoperator - linker Rand
 $f_{\mathcal{T}0}^{(m)} = a_{0,0}^{(m)} f_0^{(m)}$ 

(2) Klappoperator - innere Fenster
For  $n = 1$  To  $N - 1$ 
 $f_{\mathcal{T}2n-1}^{(m)} = a_{0,1}^{(m)} f_{2n-1}^{(m)} - a_{1,0}^{(m)} f_{2n}^{(m)}$ 
 $f_{\mathcal{T}2n}^{(m)} = a_{1,0}^{(m)} f_{2n-1}^{(m)} + a_{0,1}^{(m)} f_{2n}^{(m)}$ 
End

(3) Klappoperator - rechter Rand
 $f_{\mathcal{T}2N-1}^{(m)} = a_{0,2}^{(m)} f_{2N-1}^{(m)}$ 

Für  $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$ 

(1 & 2) Klappoperator - linker Rand und innere Fenster
For  $n = 0$  To  $N - 2$ 
 $f_{\mathcal{T}2n}^{(m)} = a_{0,0}^{(m)} f_{2n}^{(m)} - a_{1,0}^{(m)} f_{2n+1}^{(m)}$ 
 $f_{\mathcal{T}2n+1}^{(m)} = a_{1,0}^{(m)} f_{2n}^{(m)} + a_{0,0}^{(m)} f_{2n+1}^{(m)}$ 
End

(3) Klappoperator - rechter Rand
 $f_{\mathcal{T}2N-2}^{(m)} = a_{0,0}^{(m)} f_{2N-2}^{(m)} - a_{1,0}^{(m)} f_{2N-1}^{(m)}$ 
 $f_{\mathcal{T}2N-1}^{(m)} = a_{1,0}^{(m)} f_{2N-2}^{(m)} + a_{0,0}^{(m)} f_{2N-1}^{(m)}$ 

```

Abbildung E.2: Algorithmus zur effizienten Realisierung des Klappoperators für zweifach überlappende Fenster, $m = 0, 1, \dots, B - 1$; $B = \text{Blocklänge}$, $L = 2NB = \text{Signallänge}$, $O_{\mathcal{U}} = 3 - 2\frac{B}{L}$ arithmetische Operationen pro Signalwert.

[illegible]

Abbildung E.3: Indizierung und Zuordnung der Matrixkoeffizienten zur effizienten Realisierung des Aufklappoperators als Matrix-Vektor-Produkt für die Zweifachüberlappung, $m = 0, 1, \dots, \frac{B}{2} - 1$; B = Blockgröße, $L = 2NB$ = Signallänge.

[illegible]

Abbildung E.5: Indizierung und Zuordnung der Matrixkoeffizienten zur effizienten Realisierung des Klappoperators als Matrix-Vektor-Produkt für die Zweifachüberlappung, $m = 0, 1, \dots, \frac{B}{2} - 1$; B = Blockgröße, $L = 2NB$ = Signallänge.

[illegible]

Abbildung E.6: Indizierung und Zuordnung der Matrixkoeffizienten zur effizienten Realisierung des Klappoperators als Matrix-Vektor-Produkt für die Zweifachüberlappung, $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$; B = Blockgröße, $L = 2NB$ = Signallänge.

E.2 Algorithmen für die Dreifachüberlappung

Aufklappoperator

Vektoren zur kompakten Speicherung der Abbildungsmatrizen

$$\vec{a}_{-1}^{(m)} = \begin{pmatrix} a_{-1,0}^{(m)} \\ a_{-1,1}^{(m)} \\ a_{-1,2}^{(m)} \end{pmatrix}, \quad \vec{a}_0^{(m)} = \begin{pmatrix} a_{0,0}^{(m)} \\ a_{0,1}^{(m)} \\ a_{0,2}^{(m)} \\ a_{0,3}^{(m)} \end{pmatrix}, \quad \vec{a}_1^{(m)} = \begin{pmatrix} a_{1,0}^{(m)} \\ a_{1,1}^{(m)} \\ a_{1,2}^{(m)} \\ a_{1,3}^{(m)} \end{pmatrix}. \quad (\text{E.9})$$

Die Speicherstruktur ist in Abbildung E.9 veranschaulicht.

Initialisierung der Koeffizientenvektoren

$$\begin{aligned} a_{-1,0}^{(m)} &= -w[2B - m - 1], & a_{0,0}^{(m)} &= w_{l_0}[m], & a_{1,0}^{(m)} &= w_{l_1}[-B + m], \\ a_{-1,1}^{(m)} &= -w[B + m], & a_{0,1}^{(m)} &= w[B - m - 1], & a_{1,1}^{(m)} &= w[-m - 1], \\ a_{-1,2}^{(m)} &= -w_{r_1}[2B - m - 1], & a_{0,2}^{(m)} &= w[m], & a_{1,2}^{(m)} &= w[-B + m], \\ & & a_{0,3}^{(m)} &= w_{r_0}[B - m - 1], & a_{1,3}^{(m)} &= w_{r_0}[-B + m]. \end{aligned} \quad (\text{E.10})$$

(1) *Aufklappoperator - linker Rand*

$$f_0^{(m)} = a_{0,0}^{(m)} f_{\mathcal{T}0}^{(m)} + a_{1,0}^{(m)} f_{\mathcal{T}1}^{(m)}$$

(2) *Aufklappoperator - innere Fenster*

For $n = 1$ **To** $N - 2$

$$\begin{aligned} f_{2n-1}^{(m)} &= a_{-1,0}^{(m)} f_{\mathcal{T}2n-2}^{(m)} + a_{0,1}^{(m)} f_{\mathcal{T}2n-1}^{(m)} + a_{1,1}^{(m)} f_{\mathcal{T}2n}^{(m)} \\ f_{2n}^{(m)} &= a_{-1,1}^{(m)} f_{\mathcal{T}2n-1}^{(m)} + a_{0,2}^{(m)} f_{\mathcal{T}2n}^{(m)} + a_{1,2}^{(m)} f_{\mathcal{T}2n+1}^{(m)} \end{aligned}$$

End

$$f_{2N-3}^{(m)} = a_{-1,0}^{(m)} f_{\mathcal{T}2N-4}^{(m)} + a_{0,1}^{(m)} f_{\mathcal{T}2N-3}^{(m)} + a_{1,1}^{(m)} f_{\mathcal{T}2N-2}^{(m)}$$

(3) *Aufklappoperator - rechter Rand*

$$\begin{aligned} f_{2N-2}^{(m)} &= a_{-1,1}^{(m)} f_{\mathcal{T}2N-3}^{(m)} + a_{0,2}^{(m)} f_{\mathcal{T}2N-2}^{(m)} + a_{1,3}^{(m)} f_{\mathcal{T}2N-1}^{(m)} \\ f_{2N-1}^{(m)} &= a_{-1,2}^{(m)} f_{\mathcal{T}2N-2}^{(m)} + a_{0,3}^{(m)} f_{\mathcal{T}2N-1}^{(m)} \end{aligned}$$

Abbildung E.7: Algorithmus zur effizienten Realisierung des Aufklappoperators für dreifach überlappende Fenster, $m = 0, 1, \dots, B - 1$; $B = \text{Blocklänge}$, $L = 2NB = \text{Signallänge}$, $O_{\mathcal{U}} = 5 - 4\frac{B}{L}$ arithmetische Operationen pro Signalwert.

Klappoperator

Vektoren zur kompakten Speicherung der Koeffizientenmatrizen

$$\vec{a}_{-1}^{(m)} = \begin{pmatrix} a_{-1,0}^{(m)} \\ a_{-1,1}^{(m)} \\ \vdots \\ a_{-1,2N-2}^{(m)} \end{pmatrix}, \quad \vec{a}_0^{(m)} = \begin{pmatrix} a_{0,0}^{(m)} \\ a_{0,1}^{(m)} \\ \vdots \\ a_{0,2N-1}^{(m)} \end{pmatrix}, \quad \vec{a}_1^{(m)} = \begin{pmatrix} a_{1,0}^{(m)} \\ a_{1,1}^{(m)} \\ a_{1,2}^{(m)} \\ a_{1,3}^{(m)} \end{pmatrix}. \quad (\text{E.11})$$

Die Speicherstruktur ist in Abbildung E.10 veranschaulicht.

Initialisierung der Matrixkoeffizienten bzw. der Vektorkoeffizienten

$$\begin{aligned} a_{-1,2N-4}^{(m)} &= -w[2B - m - 1], & a_{0,0}^{(m)} &= w_{l_0}[m], & a_{1,0}^{(m)} &= w_{l_1}[-B + m], \\ a_{-1,2N-3}^{(m)} &= -w[B + m], & a_{0,2N-3}^{(m)} &= w[B - m - 1], & a_{1,1}^{(m)} &= w[-m - 1], \\ a_{-1,2N-2}^{(m)} &= -w_{r_1}[2B - m - 1], & a_{0,2N-2}^{(m)} &= w[m], & a_{1,2}^{(m)} &= w[-B + m], \\ & & a_{0,2N-1}^{(m)} &= w_{r_0}[B - m - 1], & a_{1,3}^{(m)} &= w_{r_0}[-B + m]. \end{aligned} \quad (\text{E.12})$$

Eine Initialisierung der übrigen Koeffizienten ist für den in Abbildung E.8 dargestellten Algorithmus nicht erforderlich. Sie werden mit den Werten der LU-Zerlegung beschrieben.

LU-Koeffizienten	Klappoperation
<p>(1) <i>LU-Zerlegung</i></p> $a_{-1,0}^{(m)} = a_{-1,2N-4}^{(m)} / a_{0,0}^{(m)}$ $a_{0,1}^{(m)} = a_{0,2N-3}^{(m)} - a_{-1,0}^{(m)} a_{1,0}^{(m)}$ <p>For $n = 1$ To $N - 2$</p> $a_{-1,2n-1}^{(m)} = a_{-1,2N-3}^{(m)} / a_{0,2n-1}^{(m)}$ $a_{0,2n}^{(m)} = a_{0,2N-2}^{(m)} - a_{-1,2n-1}^{(m)} a_{1,1}^{(m)}$ $a_{-1,2n}^{(m)} = a_{-1,2N-4}^{(m)} / a_{0,2n}^{(m)}$ $a_{0,2n+1}^{(m)} = a_{0,2N-3}^{(m)} - a_{-1,2n}^{(m)} a_{1,2}^{(m)}$ <p>End</p> $a_{-1,2N-3}^{(m)} = a_{-1,2N-3}^{(m)} / a_{0,2N-3}^{(m)}$ $a_{0,2N-2}^{(m)} = a_{0,2N-2}^{(m)} - a_{-1,2N-3}^{(m)} a_{1,1}^{(m)}$ $a_{-1,2N-2}^{(m)} = a_{-1,2N-2}^{(m)} / a_{0,2N-2}^{(m)}$ $a_{0,2N-1}^{(m)} = a_{0,2N-1}^{(m)} - a_{-1,2N-2}^{(m)} a_{1,3}^{(m)}$	<p>(2) <i>Vorwärtseinsetzen</i></p> <p>For $n = 1$ To $2N - 1$</p> $f_n^{(m)} = f_n^{(m)} - a_{-1,n-1}^{(m)} f_{n-1}^{(m)}$ <p>End</p> <p>(3) <i>Rückwärtseinsetzen</i></p> $f_{2N-1}^{(m)} = f_{2N-1}^{(m)} / a_{0,2N-1}^{(m)}$ $f_{2N-2}^{(m)} = \frac{f_{2N-2}^{(m)} - a_{1,3}^{(m)} f_{2N-1}^{(m)}}{a_{0,2N-2}^{(m)}}$ <p>For $n = N - 2$ To (-1)</p> $f_{2n+1}^{(m)} = \frac{f_{2n+1}^{(m)} - a_{1,1}^{(m)} f_{2n+2}^{(m)}}{a_{0,2n+1}^{(m)}}$ $f_{2n}^{(m)} = \frac{f_{2n}^{(m)} - a_{1,2}^{(m)} f_{2n+1}^{(m)}}{a_{0,2n}^{(m)}}$ <p>End</p> $f_1^{(m)} = (f_1^{(m)} - a_{1,1}^{(m)} f_2^{(m)}) / a_{0,1}^{(m)}$ $f_0^{(m)} = (f_0^{(m)} - a_{1,0}^{(m)} f_1^{(m)}) / a_{0,0}^{(m)}$

Abbildung E.8: Algorithmen zur effizienten Realisierung der LU-Zerlegung und der Klappoperation durch Vorwärts- und Rückwärtseinsetzen für dreifach überlappende Fenster, $m = 0, 1, \dots, B - 1$; B = Blocklänge, $L = 2NB$ = Signallänge, $O_{LU} = 3 - 3\frac{B}{L}$ und $O_{\mathcal{T}} = 5 - 4\frac{B}{L}$ arithmetische Operationen pro Signalwert. Das geklappte Signal $f_{\mathcal{T}}$ wird Inplace in f gespeichert.

[illegible]

Abbildung E.9: Indizierung und Zuordnung der Matrixkoeffizienten zur effizienten Realisierung des Aufklappoperators als Matrix-Vektor-Produkt für die Dreifachüberlappung, $m = 0, 1, \dots, B - 1$; B = Blockgröße, $L = 2NB$ = Signallänge.

E.3 Algorithmen für die Vierfachüberlappung

Aufklappoperator

Vektoren zur kompakten Speicherung der Abbildungsmatrizen

Da Nullkoeffizienten in den äußeren Nebendiagonalen in Abhängigkeit vom Parameter m auftreten, sind bei der Realisierung der Algorithmen zwei Fälle zu unterscheiden.

Für $m = 0, 1, \dots, \frac{B}{2} - 1$:

$$\vec{a}_{-1}^{(m)} = \begin{pmatrix} a_{-1,0}^{(m)} \end{pmatrix}, \quad \vec{a}_0^{(m)} = \begin{pmatrix} a_{0,0}^{(m)} \\ a_{0,1}^{(m)} \\ a_{0,2}^{(m)} \end{pmatrix}, \quad \vec{a}_1^{(m)} = \begin{pmatrix} a_{1,0}^{(m)} \\ a_{1,1}^{(m)} \\ a_{1,2}^{(m)} \\ a_{1,3}^{(m)} \end{pmatrix}, \quad \vec{a}_2^{(m)} = \begin{pmatrix} a_{2,0}^{(m)} \\ a_{2,1}^{(m)} \end{pmatrix}. \quad (\text{E.13})$$

Für $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$:

$$\vec{a}_{-1}^{(m)} = \begin{pmatrix} a_{-1,0}^{(m)} \end{pmatrix}, \quad \vec{a}_0^{(m)} = \begin{pmatrix} a_{0,0}^{(m)} \\ a_{0,1}^{(m)} \\ a_{0,2}^{(m)} \\ a_{0,3}^{(m)} \end{pmatrix}, \quad \vec{a}_1^{(m)} = \begin{pmatrix} a_{1,0}^{(m)} \\ a_{1,1}^{(m)} \\ a_{1,2}^{(m)} \\ a_{1,3}^{(m)} \end{pmatrix}, \quad \vec{a}_2^{(m)} = \begin{pmatrix} a_{2,0}^{(m)} \end{pmatrix}. \quad (\text{E.14})$$

Die Speicherstruktur ist in den Abbildungen E.14 und E.15 veranschaulicht.

Initialisierung der Koeffizientenvektoren

Für $m = 0, 1, \dots, B - 1$:

$$\begin{aligned} a_{-1,0}^{(m)} &= -w_{r_1}[2B - m - 1], & a_{0,0}^{(m)} &= w_{l_0}[m], \\ a_{1,0}^{(m)} &= w_{l_1}[-B + m], & a_{1,1}^{(m)} &= w[-m - 1], \\ a_{1,2}^{(m)} &= w[-B + m], & a_{1,3}^{(m)} &= w_{r_0}[-B + m]. \end{aligned} \quad (\text{E.15})$$

Für $m = 0, 1, \dots, \frac{B}{2} - 1$:

$$\begin{aligned} a_{0,1}^{(m)} &= w[B - m - 1], & a_{2,0}^{(m)} &= -w[-B - m - 1], \\ a_{0,2}^{(m)} &= w_{r_0}[B - m - 1], & a_{2,1}^{(m)} &= w_{r_0}[-B - m - 1]. \end{aligned} \quad (\text{E.16})$$

Für $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$:

$$\begin{aligned} a_{0,1}^{(m)} &= w_{l_1}[B - m - 1], & a_{2,0}^{(m)} &= -w[-2B + m], \\ a_{0,2}^{(m)} &= w[m], & a_{0,3}^{(m)} &= w_{r_0}[B - m - 1]. \end{aligned} \quad (\text{E.17})$$

Für $m = 0, 1, \dots, \frac{B}{2} - 1$

(1) *Aufklappoperator - linker Rand*

$$f_0^{(m)} = a_{0,0}^{(m)} f_{\mathcal{T}0}^{(m)} + a_{1,0}^{(m)} f_{\mathcal{T}1}^{(m)}$$

(2) *Aufklappoperator - innere Fenster*

For $n = 1$ **To** $N - 2$

$$\begin{aligned} f_{2n-1}^{(m)} &= -a_{1,2}^{(m)} f_{\mathcal{T}2n-2}^{(m)} + a_{0,1}^{(m)} f_{\mathcal{T}2n-1}^{(m)} + a_{1,1}^{(m)} f_{\mathcal{T}2n}^{(m)} + a_{2,0}^{(m)} f_{\mathcal{T}2n+1}^{(m)} \\ f_{2n}^{(m)} &= a_{2,0}^{(m)} f_{\mathcal{T}2n-2}^{(m)} - a_{1,1}^{(m)} f_{\mathcal{T}2n-1}^{(m)} + a_{0,1}^{(m)} f_{\mathcal{T}2n}^{(m)} + a_{1,2}^{(m)} f_{\mathcal{T}2n+1}^{(m)} \end{aligned}$$

End

(3) *Aufklappoperator - rechter Rand*

$$\begin{aligned} f_{2N-3}^{(m)} &= -a_{1,2}^{(m)} f_{\mathcal{T}2N-4}^{(m)} + a_{0,1}^{(m)} f_{\mathcal{T}2N-3}^{(m)} + a_{1,1}^{(m)} f_{\mathcal{T}2N-2}^{(m)} + a_{2,1}^{(m)} f_{\mathcal{T}2N-1}^{(m)} \\ f_{2N-2}^{(m)} &= a_{2,0}^{(m)} f_{\mathcal{T}2N-4}^{(m)} - a_{1,1}^{(m)} f_{\mathcal{T}2N-3}^{(m)} + a_{0,1}^{(m)} f_{\mathcal{T}2N-2}^{(m)} + a_{1,3}^{(m)} f_{\mathcal{T}2N-1}^{(m)} \\ f_{2N-1}^{(m)} &= a_{-1,0}^{(m)} f_{\mathcal{T}2N-2}^{(m)} + a_{0,2}^{(m)} f_{\mathcal{T}2N-1}^{(m)} \end{aligned}$$

Für $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$

(1) *Aufklappoperator - linker Rand*

$$\begin{aligned} f_0^{(m)} &= a_{0,0}^{(m)} f_{\mathcal{T}0}^{(m)} + a_{1,0}^{(m)} f_{\mathcal{T}1}^{(m)} + a_{2,0}^{(m)} f_{\mathcal{T}2}^{(m)} \\ f_1^{(m)} &= -a_{1,2}^{(m)} f_{\mathcal{T}0}^{(m)} + a_{0,1}^{(m)} f_{\mathcal{T}1}^{(m)} + a_{1,1}^{(m)} f_{\mathcal{T}2}^{(m)} \end{aligned}$$

(2) *Aufklappoperator - innere Fenster*

For $n = 1$ **To** $N - 2$

$$\begin{aligned} f_{2n}^{(m)} &= -a_{1,1}^{(m)} f_{\mathcal{T}2n-1}^{(m)} + a_{0,2}^{(m)} f_{\mathcal{T}2n}^{(m)} + a_{1,2}^{(m)} f_{\mathcal{T}2n+1}^{(m)} + a_{2,0}^{(m)} f_{\mathcal{T}2n+2}^{(m)} \\ f_{2n+1}^{(m)} &= a_{2,0}^{(m)} f_{\mathcal{T}2n-1}^{(m)} - a_{1,2}^{(m)} f_{\mathcal{T}2n}^{(m)} + a_{0,2}^{(m)} f_{\mathcal{T}2n+1}^{(m)} + a_{1,1}^{(m)} f_{\mathcal{T}2n+2}^{(m)} \end{aligned}$$

End

(3) *Aufklappoperator - rechter Rand*

$$\begin{aligned} f_{2N-2}^{(m)} &= -a_{1,1}^{(m)} f_{\mathcal{T}2N-3}^{(m)} + a_{0,2}^{(m)} f_{\mathcal{T}2N-2}^{(m)} + a_{1,3}^{(m)} f_{\mathcal{T}2N-1}^{(m)} \\ f_{2N-1}^{(m)} &= a_{2,0}^{(m)} f_{\mathcal{T}2N-3}^{(m)} + a_{-1,0}^{(m)} f_{\mathcal{T}2N-2}^{(m)} + a_{0,3}^{(m)} f_{\mathcal{T}2N-1}^{(m)} \end{aligned}$$

Abbildung E.11: Algorithmus zur effizienten Realisierung des Aufklappoperators für vierfach überlappende Fenster, $m = 0, 1, \dots, B - 1$; B = Blocklänge, $L = 2NB$ = Signallänge, $O_U = 7 - 8 \frac{B}{L}$ arithmetische Operationen pro Signalwert.

Klappoperator

Vektoren zur kompakten Speicherung der Koeffizientenmatrizen

Da Nullkoeffizienten in den äußeren Nebendiagonalen in Abhängigkeit vom Parameter m auftreten, sind bei der Realisierung der Algorithmen zwei Fälle zu unterscheiden.

Für $m = 0, 1, \dots, \frac{B}{2} - 1$:

$$\begin{aligned}\vec{a}_{-2}^{(m)} &= (a_{-2,0}^{(m)}), \\ \vec{a}_{-1}^{(m)} &= (a_{-1,0}^{(m)} \quad a_{-1,1}^{(m)} \quad \dots \quad a_{-1,2N-3}^{(m)} \quad a_{-1,2N-2}^{(m)})^T, \\ \vec{a}_0^{(m)} &= (a_{0,0}^{(m)} \quad a_{0,1}^{(m)} \quad \dots \quad a_{0,2N-2}^{(m)} \quad a_{0,2N-1}^{(m)})^T, \\ \vec{a}_1^{(m)} &= (a_{1,0}^{(m)} \quad a_{1,1}^{(m)} \quad a_{1,2}^{(m)} \quad a_{1,3}^{(m)} \quad a_{1,3}^{(m)} \quad \dots \quad a_{1,1}^{(m)} \quad a_{1,N}^{(m)})^T, \\ \vec{a}_2^{(m)} &= (a_{2,0}^{(m)} \quad a_{2,1}^{(m)})^T.\end{aligned}\tag{E.18}$$

Für $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$:

$$\begin{aligned}\vec{a}_{-2}^{(m)} &= (a_{-2,0}^{(m)}), \\ \vec{a}_{-1}^{(m)} &= (a_{-1,0}^{(m)} \quad a_{-1,1}^{(m)} \quad \dots \quad a_{-1,2N-3}^{(m)} \quad a_{-1,2N-2}^{(m)})^T, \\ \vec{a}_0^{(m)} &= (a_{0,0}^{(m)} \quad a_{0,1}^{(m)} \quad \dots \quad a_{0,2N-2}^{(m)} \quad a_{0,2N-1}^{(m)})^T, \\ \vec{a}_1^{(m)} &= (a_{1,0}^{(m)} \quad a_{1,2}^{(m)} \quad a_{1,1}^{(m)} \quad a_{1,3}^{(m)} \quad a_{1,1}^{(m)} \quad \dots \quad a_{1,1}^{(m)} \quad a_{1,N}^{(m)} \quad a_{1,N+1}^{(m)})^T, \\ \vec{a}_2^{(m)} &= (a_{2,0}^{(m)}).\end{aligned}\tag{E.19}$$

Die Speicherstruktur ist in den Abbildungen E.16 und E.17 veranschaulicht.

Initialisierung der Koeffizientenvektoren

Für $m = 0, 1, \dots, B - 1$:

$$\begin{aligned}a_{-1,2N-3}^{(m)} &= -w[B + m], & a_{0,0}^{(m)} &= w_{l_0}[m], & a_{1,0}^{(m)} &= w_{l_1}[-B + m], \\ a_{-1,2N-2}^{(m)} &= -w_{r_1}[2B - m - 1], & a_{0,2N-2}^{(m)} &= w[m], & a_{0,2N-1}^{(m)} &= w_{r_0}[B - m - 1].\end{aligned}\tag{E.20}$$

Für $m = 0, 1, \dots, \frac{B}{2} - 1$:

$$\begin{aligned}a_{-2,0}^{(m)} &= -w[2B + m], & a_{-1,2N-4}^{(m)} &= -w[2B - m - 1], \\ a_{0,2N-3}^{(m)} &= w[B - m - 1], & a_{1,N-1}^{(m)} &= w[-B + m], \\ a_{1,N}^{(m)} &= w_{r_0}[-B + m], & a_{2,0}^{(m)} &= -w[-B - m - 1], \\ a_{2,1}^{(m)} &= w_{r_0}[-B - m - 1], & a_{1,1}^{(m)} &= w[-m - 1].\end{aligned}\tag{E.21}$$

Für $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$:

$$\begin{aligned}a_{-2,0}^{(m)} &= -w[3B - m - 1], & a_{-1,0}^{(m)} &= -w[2B - m - 1], \\ a_{0,1}^{(m)} &= w_{l_1}[B - m - 1], & a_{1,2}^{(m)} &= w[-m - 1], \\ a_{1,N+1}^{(m)} &= w_{r_0}[-B + m], & a_{2,0}^{(m)} &= -w[-2B + m], \\ a_{1,1}^{(m)} &= w[-B + m].\end{aligned}\tag{E.22}$$

Eine Initialisierung der übrigen Koeffizienten ist für den in Abbildung E.12 dargestellten Algorithmus nicht erforderlich. Diese Koeffizienten werden mit den Werten der LU-Zerlegung beschrieben.

$m = 0, 1, \dots, \frac{B}{2} - 1$	$m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$
<p>(1) <i>LU Zerlegung</i></p> <p>(1a) $a_{-2,0}^{(m)} = a_{-2,0}^{(m)} / a_{-1,2N-4}^{(m)}$ $a_{-1,2N-3}^{(m)} = a_{-1,2N-3}^{(m)} - a_{-2,0}^{(m)} a_{0,2N-3}^{(m)}$ $a_{0,2N-2}^{(m)} = a_{0,2N-2}^{(m)} - a_{-2,0}^{(m)} a_{1,1}^{(m)}$ $a_{1,N}^{(m)} = a_{1,N}^{(m)} - a_{-2,0}^{(m)} a_{2,1}^{(m)}$ $a_{1,N-1}^{(m)} = a_{1,N-1}^{(m)} - a_{-2,0}^{(m)} a_{2,0}^{(m)}$</p> <p>(1b) $a_{-1,0}^{(m)} = a_{-1,2N-4}^{(m)} / a_{0,0}^{(m)}$ $a_{0,1}^{(m)} = a_{0,2N-3}^{(m)} - a_{-1,0}^{(m)} a_{1,0}^{(m)}$ For $n = 1$ To $N - 2$ $a_{-1,2n-1}^{(m)} = a_{-1,2N-3}^{(m)} / a_{0,2n-1}^{(m)}$ $a_{0,2n}^{(m)} = a_{0,2N-2}^{(m)} - a_{-1,2n-1}^{(m)} a_{1,1}^{(m)}$ $a_{1,n+1}^{(m)} = a_{1,N-1}^{(m)} - a_{-1,2n-1}^{(m)} a_{2,0}^{(m)}$ $a_{-1,2n}^{(m)} = a_{-1,2N-4}^{(m)} / a_{0,2n}^{(m)}$ $a_{0,2n+1}^{(m)} = a_{0,2N-3}^{(m)} - a_{-1,2n}^{(m)} a_{1,n+1}^{(m)}$ End $a_{-1,2N-3}^{(m)} = a_{-1,2N-3}^{(m)} / a_{0,2N-3}^{(m)}$ $a_{0,2N-2}^{(m)} = a_{0,2N-2}^{(m)} - a_{-1,2N-3}^{(m)} a_{1,1}^{(m)}$ $a_{1,N}^{(m)} = a_{1,N}^{(m)} - a_{-1,2N-3}^{(m)} a_{2,1}^{(m)}$ $a_{-1,2N-2}^{(m)} = a_{-1,2N-2}^{(m)} / a_{0,2N-2}^{(m)}$ $a_{0,2N-1}^{(m)} = a_{0,2N-1}^{(m)} - a_{-1,2N-2}^{(m)} a_{1,N}^{(m)}$</p>	<p>(1) <i>LU Zerlegung</i></p> <p>(1a) $a_{-2,0}^{(m)} = a_{-2,0}^{(m)} / a_{-1,2N-3}^{(m)}$ $a_{-1,2N-2}^{(m)} = a_{-1,2N-2}^{(m)} - a_{-2,0}^{(m)} a_{0,2N-2}^{(m)}$ $a_{0,2N-1}^{(m)} = a_{0,2N-1}^{(m)} - a_{-2,0}^{(m)} a_{1,N+1}^{(m)}$ $a_{-1,2N-4}^{(m)} = a_{-1,0}^{(m)} - a_{-2,0}^{(m)} a_{0,2N-2}^{(m)}$ $a_{0,2N-3}^{(m)} = a_{0,2N-2}^{(m)} - a_{-2,0}^{(m)} a_{1,1}^{(m)}$ $a_{1,N}^{(m)} = a_{1,2}^{(m)} - a_{-2,0}^{(m)} a_{2,0}^{(m)}$ <p>(1b) $a_{-1,0}^{(m)} = a_{-1,0}^{(m)} / a_{0,0}^{(m)}$ $a_{0,1}^{(m)} = a_{0,1}^{(m)} - a_{-1,0}^{(m)} a_{1,0}^{(m)}$ $a_{1,2}^{(m)} = a_{1,2}^{(m)} - a_{-1,0}^{(m)} a_{2,0}^{(m)}$ For $n = 1$ To $N - 2$ $a_{-1,2n-1}^{(m)} = a_{-1,2N-3}^{(m)} / a_{0,2n-1}^{(m)}$ $a_{0,2n}^{(m)} = a_{0,2N-2}^{(m)} - a_{-1,2n-1}^{(m)} a_{1,n+1}^{(m)}$ $a_{-1,2n}^{(m)} = a_{-1,2N-4}^{(m)} / a_{0,2n}^{(m)}$ $a_{0,2n+1}^{(m)} = a_{0,2N-3}^{(m)} - a_{-1,2n}^{(m)} a_{1,1}^{(m)}$ $a_{1,n+2}^{(m)} = a_{1,N}^{(m)} - a_{-1,2n}^{(m)} a_{2,0}^{(m)}$ End $a_{-1,2N-3}^{(m)} = a_{-1,2N-3}^{(m)} / a_{0,2N-3}^{(m)}$ $a_{0,2N-2}^{(m)} = a_{0,2N-2}^{(m)} - a_{-1,2N-3}^{(m)} a_{1,N}^{(m)}$ $a_{-1,2N-2}^{(m)} = a_{-1,2N-2}^{(m)} / a_{0,2N-2}^{(m)}$ $a_{0,2N-1}^{(m)} = a_{0,2N-1}^{(m)} - a_{-1,2N-2}^{(m)} a_{1,N+1}^{(m)}$</p> </p>

Abbildung E.12: Algorithmus zur effizienten Realisierung der LU-Zerlegung für vierfach überlappende Fenster, $m = 0, 1, \dots, B - 1$; $B = \text{Blocklänge}$, $L = 2NB = \text{Signallänge}$, $O_{LU} = 4 + 5\frac{B}{L}$ arithmetische Operationen pro Signalwert.

$m = 0, 1, \dots, \frac{B}{2} - 1$ (2) <i>Vorwärtseinsetzen</i> (2a) For $n = 1$ To $N - 1$ $f_{2n}^{(m)} = f_{2n}^{(m)} - a_{-2,0}^{(m)} f_{2n-1}^{(m)}$ End (2b) For $n = 1$ To $2N - 1$ $f_n^{(m)} = f_n^{(m)} - a_{-1,n-1}^{(m)} f_{n-1}^{(m)}$ End (3) <i>Rückwärtseinsetzen</i> $f_{2N-1}^{(m)} = \frac{f_{2N-1}^{(m)}}{a_{0,2N-1}^{(m)}}$ $f_{2N-2}^{(m)} = \frac{f_{2N-2}^{(m)} - a_{1,N}^{(m)} f_{2N-1}^{(m)}}{a_{0,2N-2}^{(m)}}$ $f_{2N-3}^{(m)} = \frac{f_{2N-3}^{(m)} - a_{2,1}^{(m)} f_{2N-1}^{(m)} - a_{1,1}^{(m)} f_{2N-2}^{(m)}}{a_{0,2N-3}^{(m)}}$ For $n = N - 2$ To $(-1) 1$ $f_{2n}^{(m)} = \frac{f_{2n}^{(m)} - a_{1,n+1}^{(m)} f_{2n+1}^{(m)}}{a_{0,2n}^{(m)}}$ $f_{2n-1}^{(m)} = \frac{f_{2n-1}^{(m)} - a_{2,0}^{(m)} f_{2n+1}^{(m)} - a_{1,1}^{(m)} f_{2n}^{(m)}}{a_{0,2n-1}^{(m)}}$ End $f_0^{(m)} = \frac{f_0^{(m)} - a_{1,0}^{(m)} f_1^{(m)}}{a_{0,0}^{(m)}}$	$m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$ (2) <i>Vorwärtseinsetzen</i> (2a) For $n = 1$ To $N - 1$ $f_{2n+1}^{(m)} = f_{2n+1}^{(m)} - a_{-2,0}^{(m)} f_{2n}^{(m)}$ End (2b) For $n = 1$ To $2N - 1$ $f_n^{(m)} = f_n^{(m)} - a_{-1,n-1}^{(m)} f_{n-1}^{(m)}$ End (3) <i>Rückwärtseinsetzen</i> $f_{2N-1}^{(m)} = \frac{f_{2N-1}^{(m)}}{a_{0,2N-1}^{(m)}}$ $f_{2N-2}^{(m)} = \frac{f_{2N-2}^{(m)} - a_{1,N+1}^{(m)} f_{2N-1}^{(m)}}{a_{0,2N-2}^{(m)}}$ For $n = N - 2$ To $(-1) 1$ $f_{2n+1}^{(m)} = \frac{f_{2n+1}^{(m)} - a_{1,n+2}^{(m)} f_{2n+2}^{(m)}}{a_{0,2n+1}^{(m)}}$ $f_{2n}^{(m)} = \frac{f_{2n}^{(m)} - a_{2,0}^{(m)} f_{2n+2}^{(m)} - a_{1,1}^{(m)} f_{2n+1}^{(m)}}{a_{0,2n}^{(m)}}$ End $f_1^{(m)} = \frac{f_1^{(m)} - a_{1,2}^{(m)} f_2^{(m)}}{a_{0,1}^{(m)}}$ $f_0^{(m)} = \frac{f_0^{(m)} - a_{2,0}^{(m)} f_2^{(m)} - a_{1,0}^{(m)} f_1^{(m)}}{a_{0,0}^{(m)}}$
---	--

Abbildung E.13: Algorithmus zur effizienten Realisierung des Klappoperators für vierfach überlappende Fenster, $m = 0, 1, \dots, B - 1$; $B = \text{Blocklänge}$, $L = 2NB = \text{Signallänge}$, $O_{\mathcal{T}} = 7 - 8 \frac{B}{L}$ arithmetische Operationen pro Signalwert. Das geklappte Signal $f_{\mathcal{T}}$ wird Inplace in f gespeichert.

[illegible]

Abbildung E.15: Indizierung und Zuordnung der Matrixkoeffizienten zur effizienten Realisierung des Aufklappoperators als Matrix-Vektor-Produkt für die Vierfachüberlappung, $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$; B = Blockgröße, $L = 2NB$ = Signallänge.

$$\begin{array}{c}
\begin{array}{c} a_{0,0}^{(m)} \\ w_{10}[m] \end{array} \quad \begin{array}{c} a_{1,0}^{(m)} \\ w_{11}[-B+m] \end{array} \quad \begin{array}{c} a_{2,0}^{(m)} \\ -w[-2B+m] \end{array} \\
\begin{array}{c} a_{-1,0}^{(m)} \\ -w[2B-m-1] \end{array} \quad \begin{array}{c} a_{0,1}^{(m)} \\ w_{11}[B-m-1] \end{array} \quad \begin{array}{c} a_{1,2}^{(m)} \\ w[-m-1] \end{array} \\
\begin{array}{c} a_{-1,1}^{(m)} \\ -w[B+m] \end{array} \quad \begin{array}{c} a_{0,2}^{(m)} \\ w[m] \end{array} \quad \begin{array}{c} a_{1,1}^{(m)} \\ w[-B+m] \end{array} \quad \begin{array}{c} a_{2,0}^{(m)} \\ -w[-2B+m] \end{array} \\
\begin{array}{c} a_{-2,0}^{(m)} \\ -w[3B-m-1] \end{array} \quad \begin{array}{c} a_{-1,2}^{(m)} \\ w[B-m-1] \end{array} \quad \begin{array}{c} a_{0,3}^{(m)} \\ w[B-m-1] \end{array} \quad \begin{array}{c} a_{1,3}^{(m)} \\ w[-m-1] \end{array} \\
\begin{array}{c} a_{-1,3}^{(m)} \\ -w[B+m] \end{array} \quad \begin{array}{c} a_{0,4}^{(m)} \\ w[m] \end{array} \quad \begin{array}{c} a_{1,1}^{(m)} \\ w[-B+m] \end{array} \quad \begin{array}{c} a_{2,0}^{(m)} \\ -w[-2B+m] \end{array} \\
\begin{array}{c} a_{-2,0}^{(m)} \\ -w[3B-m-1] \end{array} \quad \begin{array}{c} a_{-1,4}^{(m)} \\ w[2B-m-1] \end{array} \quad \begin{array}{c} a_{0,5}^{(m)} \\ w[B-m-1] \end{array} \quad \begin{array}{c} a_{1,4}^{(m)} \\ w[-m-1] \end{array} \\
\cdot \quad \cdot \quad \cdot
\end{array}$$

Abbildung E.17: Indizierung und Zuordnung der Matrixkoeffizienten zur effizienten Realisierung des Klappoperators mittels LU-Zerlegung für die Vierfachüberlappung, $m = \frac{B}{2}, \frac{B}{2} + 1, \dots, B - 1$; B = Blockgröße, $L = 2NB$ = Signallänge.

Anhang F

Codierungsalgorithmen

F.1 Grundprinzip der SPIHT-Codierung

Dieser Abschnitt gibt einen Überblick über die Funktionsweise der SPIHT-Codierung, indem die hierfür wesentlichen Informationen aus der zugrunde liegenden Literatur [SP96b, PS08, Fey01, Sch06, Sch00] zusammengefasst werden.

Zur Komprimierung der Transformationskoeffizienten mit dem SPIHT-Algorithmus werden zunächst vier Mengen definiert [SP96b]:

- \mathcal{H} = Menge der Koordinaten der Wurzeln der Baumstruktur (Transformationskoeffizienten des DC-Teilbands),
- \mathcal{O} = Menge der direkten Nachkommen eines Knotens (i, j) ,
- \mathcal{D} = Menge aller Nachkommen eines Knotens (i, j) ,
- \mathcal{L} = Menge aller Nachkommen ohne die direkten Nachkommen eines Knotens (i, j) :

$$\mathcal{L}(i, j) = \mathcal{D}(i, j) \setminus \mathcal{O}(i, j). \quad (\text{F.1})$$

Für den Wavelet-Baum sind die verschiedenen Mengentypen bezogen auf ein Element (i, j) in Abbildung F.1 dargestellt. Mit Hilfe der Baumstruktur werden nun die Transformationskoeffizienten in jeder Bitschicht k in signifikante und insignifikante Koeffizienten partitioniert. Dazu wird die Signifikanzfunktion

$$S_k(\mathcal{I}) := \begin{cases} 1, & \max_{(i,j) \in \mathcal{I}} \{|c_{i,j}|\} \geq 2^k, \\ 0, & \text{sonst} \end{cases} \quad (\text{F.2})$$

definiert, wobei \mathcal{I} entweder eine einzelne Koordinate (i, j) oder eine Koordinatenmenge $\{(i, j)\}$ beschreibt. Ein Transformationskoeffizient ist in der Quantisierungsschicht k signifikant ($S_k = 1$), wenn sein Betrag größer oder gleich der Schwelle 2^k ist. Ansonsten ist er insignifikant ($S_k = 0$). Eine Menge ist in der Quantisierungsschicht k signifikant, wenn mindestens ein Transformationskoeffizient dieser Menge signifikant ist. Die Signifikanzinformationen werden jeweils durch ein Bit codiert.

Der SPIHT-Algorithmus ist dadurch charakterisiert, dass die Codierung und die Decodierung nach dem gleichen Prinzip erfolgen. Alle Entscheidungen werden vollständig synchron durchgeführt. Der Codieralgorithmus erzeugt für jede Entscheidung ein Bit und gibt dieses aus. Der

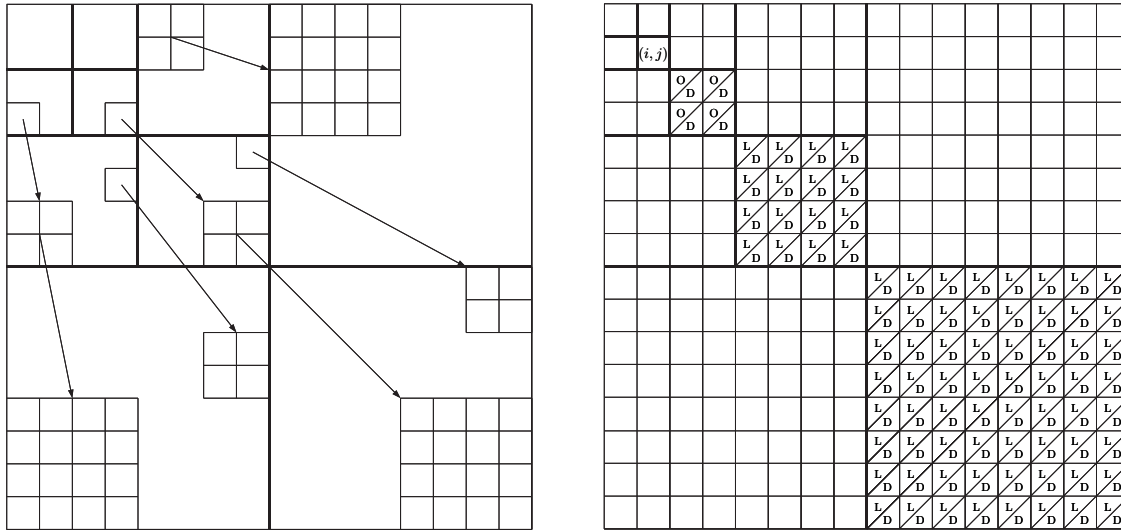


Abbildung F.1: Mengentypen bei der SPIHT-Codierung am Beispiel der Wavelet-Baumstruktur [SP96b, Sch00, Fey01].

Decodieralgorithmus liest dieses Bit ein und vollzieht diese Entscheidung nach. Der aktuelle Verarbeitungszustand des Codecs wird währenddessen in drei geordneten Koordinatenlisten gespeichert. Die Listen LIP^1 und LIS^2 bilden die Grundlage für die Codierung der Signifikanzinformationen der Koeffizienten und Teilmengen. Die Liste LSP^3 dient zur Bestimmung und Codierung der Verfeinerungsinformationen. Die SPIHT-Codierung von Transformationskoeffizienten kann in die vier Stufen Initialisierung, Sortierung, Verfeinerung und Aktualisierung der Bitschicht unterteilt werden. Der Pseudocode [SP96b, Fey01] zur Realisierung der vier Stufen ist in Abbildung F.2 dargestellt.

Initialisierung

In dieser Stufe wird die für die Codierung höchstwertige Bitschicht bestimmt. Der Parameter k_0 , der die erste Schwelle beschreibt, ist so zu wählen, dass für den betragsmäßig größten Transformationskoeffizienten $c_{i,j}$ die Bedingung $2^{k_0} \leq \max |c_{i,j}| < 2^{k_0+1}$ erfüllt wird. Hieraus ergibt sich für den Parameter k_0 die Bestimmungsgleichung

$$k_0 = \lfloor \log_2(\max |c_{i,j}|) \rfloor. \quad (F.3)$$

Dieser Parameter ist für die Decodierung des Datenstroms auf der Empfängerseite notwendig und muss daher übertragen werden. Weiterhin erfolgt in dieser Stufe die Initialisierung der Listen. Hierbei werden die Koordinaten aller Transformationskoeffizienten der Menge \mathcal{H} in die Liste LIP eingefügt. Die Koordinaten der Transformationskoeffizienten, die Nachkommen besitzen, werden auch in die Liste LIS als Koeffizient vom Typ A eingetragen. Die Liste LSP signifikanter Pixel bleibt leer. Demzufolge sind zu Verarbeitungsbeginn alle Transformationskoeffizienten als insignifikant gekennzeichnet.

¹LIP: list of insignificant pixels

²LIS: list of insignificant sets

³LSP: list of significant pixels

- (0) **Initialisierung**
Ausgabe von $k = \lfloor \log_2(\max |c_{i,j}|) \rfloor$
LSP = leere Menge
Die Koordinaten $(i, j) \in \mathcal{H}$ in LIP einfügen
Die Koordinaten $(i, j) \in \mathcal{H}$ mit direkten Nachkommen in LIS als
Eintrag vom Typ A einfügen
- (1) **Sortierung**
(2) Für alle $(i, j) \in \text{LIP}$
(3) **Ausgabe** von $S_k(i, j)$
(4) Falls $S_k(i, j) = 1$, Verschieben von (i, j) in LSP und
Ausgabe des Vorzeichens von $c_{i,j}$
(5) Für alle $(i, j) \in \text{LIS}$
(6) Falls (i, j) Eintrag vom Typ A ist,
(7) **Ausgabe** von $S_k(\mathcal{D}(i, j))$
(8) Falls $S_k(\mathcal{D}(i, j)) = 1$
(9) Für alle $(e, f) \in \mathcal{O}(i, j)$
(10) **Ausgabe** $S_k(e, f)$
(11) Falls $S_k(e, f) = 1$, Hinzufügen von (e, f) zu LSP
und **Ausgabe** des Vorzeichens von $c_{e,f}$
(12) sonst Anhängen von (e, f) an LIP
(13) Falls $\mathcal{L}(i, j) \neq \text{leere Menge}$
(14) (i, j) an das Ende von LIS als Eintrag vom Typ B
verschieben, Weiter bei (5)
(15) sonst (i, j) aus LIS entfernen
(16) Falls (i, j) Eintrag vom Typ B ist,
(17) **Ausgabe** von $S_k(\mathcal{L}(i, j))$
(18) Falls $S_k(\mathcal{L}(i, j)) = 1$
(19) Hänge alle $(e, f) \in \mathcal{O}(i, j)$ an LIS
als Eintrag vom Typ A an
(20) Entferne (i, j) aus LIS
- (21) **Verfeinerung**
(22) Für alle $(i, j) \in \text{LSP}$ außer denen, die in der letzten Sortierstufe (d.h. mit
gleichem k) betrachtet wurden, Ausgabe des k -ten Bits von $|c_{i,j}|$
- (23) **Bitschichtaktualisierung**
(24) Vermindere k um 1, gehe zur Zeile (1)

Abbildung F.2: Pseudocode zur Realisierung des SPIHT-Algorithmus [SP96b, Fey01].

Sortierung

In dieser Stufe wird festgestellt, ob in der aktuellen Bitschicht k die bisher als insignifikant gekennzeichneten Koeffizienten und Koeffizientenmengen signifikant werden. Die Zeilen (2)–(4) des in Abbildung F.2 dargestellten Pseudocodes realisieren für jedes Element (i, j) aus der LIP-Liste insignifikanter Koeffizienten die Bestimmung und Ausgabe der Signifikanzinformation $S_k(i, j)$. Elemente, die in dieser Quantisierungsschicht erstmals signifikant werden, sind aus der Liste LIP an das Ende der LSP-Liste mit signifikanten Koeffizienten zu verschieben und es ist das Vorzeichen des Koeffizienten auszugeben.

In den Zeilen (5)–(20) werden über die Liste LIS die Koeffizientenmengen abgearbeitet. Hierbei ist zwischen den Listenelementen $(i, j) \in \text{LIS}$ vom Typ A und vom Typ B zu unterscheiden. Beim Typ A werden alle Nachkommen $\mathcal{D}(i, j)$ des Listenelements (i, j) betrachtet. Der Typ B beschreibt hingegen die Menge $\mathcal{L}(i, j)$ aller Nachkommen außer den direkten Nachkommen. Sind bei einem Listenelement vom Typ A alle seine Nachkommen insignifikant, so wird die Signifikanzinformation $S_k(\mathcal{D}(i, j)) = 0$ ausgegeben und mit dem nächsten Listenelement fortgefahren. Ansonsten wird die Signifikanzinformation $S_k(\mathcal{D}(i, j)) = 1$ ausgegeben. Darüberhinaus wird in diesem Fall auch die Signifikanzinformation eines jeden direkten Nachkommens $(e, f) \in \mathcal{O}(i, j)$ ermittelt und ausgegeben. Bei Signifikanz erfolgt die Ausgabe des Vorzeichens und die Einsortierung des betrachteten Nachkommen in die LSP-Liste signifikanter Elemente. Anderenfalls wird er an die LIP-Liste insignifikanter Elemente angehängt. Sind außer den direkten Nachkommen noch weitere Nachkommen vorhanden, so wird das Element (i, j) nicht gelöscht, sondern an das Ende der Liste LIS verschoben und als Typ B gekennzeichnet. Die Bearbeitung der weiteren Nachkommen ist somit sichergestellt.

Die Elemente $(i, j) \in \text{LIS}$ vom Typ B werden in den Zeilen (16)–(20) des Pseudocodes verarbeitet. Da die direkten Nachkommen eines Elements dieser Liste bereits betrachtet wurden, erfolgt nun die Bestimmung und Ausgabe der Signifikanzinformation der Menge $\mathcal{L}(i, j)$ aller Nachkommen außer den direkten Nachkommen. Ist in der Menge $\mathcal{L}(i, j)$ ein Element signifikant, so werden die Elemente $(e, f) \in \mathcal{O}(i, j)$ der direkten Nachkommen vom Element (i, j) an die Liste LIS angehängt und als Typ A gekennzeichnet. In diesem Fall wird das Element (i, j) aus der LIS-Liste gelöscht.

Verfeinerung

In dieser Stufe werden weitere Verfeinerungsinformationen für signifikante Koeffizienten aus vorangegangenen Bitschichten übertragen. Die Vorzeichen dieser Koeffizienten sind bereits bekannt und erfordern keine erneute Übertragung. Koeffizienten, die erst in der aktuellen Bitschicht signifikant wurden, sind nicht zu verarbeiten.

Bitschichtaktualisierung

In der Stufe der Bitschichtaktualisierung wird der Parameter k um 1 reduziert und schließlich der Codierprozeß mit der Sortierungsstufe fortgesetzt (Zeile (24) in Abbildung F.2).

Der durch die SPIHT-Codierung erzeugte Datenstrom kann durch Entropiecodierung, z.B. mit einem Arithmetic-Coder [WNC87], weiter komprimiert werden. Der Codierungsprozess lässt sich beim Erreichen einer gewünschte Zielbitrate abbrechen. Anderenfalls wird durch das beschriebene Aufsplitten der Mengen solange im Suchbaum hinabgestiegen, bis eine Bitschicht erreicht ist, bei der alle Koeffizienten in der LSP-Liste für signifikante Koeffizienten stehen. In diesem Fall kann das ursprüngliche Signal fehlerfrei rekonstruiert werden.

Rekonstruktion

Der Decoder arbeitet nach dem gleichen Prinzip wie der Encoder. Zunächst sind die Ausgabebefehle des in Abbildung F.2 dargestellten Algorithmus durch Einlesebefehle auszutauschen. Wird nun in der Bitschicht k ein Koeffizient durch das Einlesen der Signifikanzinformation in der Zeile (4) oder in der Zeile (11) signifikant, so ist er in Abhängigkeit vom Vorzeichenbit auf $\pm \frac{3}{2}2^k$ zu setzen. In der Verfeinerungsstufe wird der Koeffizient (i, j) in Abhängigkeit vom eingelesenen Bit um $2^{(k-1)}$ erhöht oder reduziert.

F.2 Grundprinzip der L-CEB-Codierung

Um einen Überblick über die Funktionsweise der L-CEB-Codierung zu geben, werden in diesem Abschnitt die hierfür wesentlichen Informationen aus der zugrunde liegenden Literatur [TT02a, TT02b] zusammengefasst.

Die L-CEB-Codierung von quantisierten Transformationskoeffizienten beinhaltet die folgenden Verarbeitungstufen:

1. Bestimmung der Signifikanz: quantisierte Koeffizienten werden getestet, ob sie signifikant (ungleich Null) oder nicht signifikant (gleich Null) sind,
2. Codierung der Amplitude: Ist ein Koeffizient signifikant, so wird seine Amplitude binarisiert,
3. Codierung der Vorzeichen: Ist ein Koeffizient signifikant, so wird sein Vorzeichen codiert,
4. Entropiecodierung: adaptive binäre arithmetische Codierung der Amplituden und kontextbasierte adaptive binäre arithmetische Codierung der Signifikanzinformationen.

Für die Realisierung der L-CEB-Codierung erfolgt zunächst, wie in Abbildung F.3 dargestellt, die Durchnummerierung aller Transformationskoeffizienten $c_{x,y}^{i,j}$ nach einer Reihenfolge, die dem Zig-Zag Scan im JPEG-Verfahren entspricht:

$$c_{x,y}^{i,j} \rightarrow c_{x,y}^l, \quad 0 \leq l < 64. \quad (\text{F.4})$$

Die Koordinate (i, j) bzw. der Parameter l beschreibt die Position des betrachteten Koeffizienten innerhalb des Koeffizientenblocks (x, y) . Weiterhin wird eine Koeffizientenmenge $\mathcal{D}_{x,y}^l$

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Abbildung F.3: Zigzag-Scan [TT02a].

- (1) Suche nach dem Zig-Zag Index l des letzten signifikanten Koeffizienten
- (2) Codiere die Signifikanzinformation $S(c_{x,y}^0)$
- (3) Falls $S(c_{x,y}^0) = 1$, dann codiere die Amplitude und das Vorzeichen von $c_{x,y}^0$
- (4) Codiere $S(\mathcal{D}_{x,y}^1)$
- (5) Falls $S(\mathcal{D}_{x,y}^1) = 1$, dann für $i = 1$ bis l
 - (a) Codiere $S(c_{x,y}^i)$
 - (b) Falls $S(c_{x,y}^i) = 1$, dann
 - (i) Codiere Amplitude und Vorzeichen von $c_{x,y}^i$
 - (ii) Falls $i < 63$, dann codiere die Signifikanzinformation $S(\mathcal{D}_{x,y}^{i+1})$, welche für $i < l$ 1 ist und sonst 0

Abbildung F.4: Pseudocode zur Implementierung der L-CEB-Codierung der quantisierten Transformationskoeffizienten $c_{x,y}^l$ eines Koeffizientenblocks (x, y) nach [TT02a].

eingeführt, die durch

$$\mathcal{D}_{x,y}^l := \{c_{x,y}^l, c_{x,y}^{l+1}, \dots, c_{x,y}^{63}\} \quad (\text{F.5})$$

definiert ist. Der Pseudocode zur Realisierung der ersten drei Verarbeitungsstufen ist in Abbildung F.4 dargestellt. Er beinhaltet u.a. die folgenden Arbeitsschritte:

Codierung der DC-Koeffizienten

Um die Korrelation zwischen den DC-Koeffizienten der benachbarten Blöcke aufzulösen, wird der DC-Koeffizient $c_{x,y}^{0,0}$ eines Blocks (x, y) aus den DC-Koeffizienten des linken und des oberen benachbarten Blocks durch Mittelwertbildung $(c_{x-1,y}^{0,0} + c_{x,y-1}^{0,0})/2$ prädiziert und der Prädiktionsfehler codiert.

Bestimmung der Signifikanz

Die Größen $S(c_{x,y}^l)$ und $S(\mathcal{D}_{x,y}^l)$ stellen die Signifikanzinformation von $c_{x,y}^l$ und $\mathcal{D}_{x,y}^l$ dar, wobei [TT02a]

$$S(c_{x,y}^l) := \begin{cases} 1, & \text{falls } c_{x,y}^l \text{ signifikant,} \\ 0, & \text{sonst,} \end{cases} \quad S(\mathcal{D}_{x,y}^l) := \begin{cases} 1, & \text{falls } \mathcal{D}_{x,y}^l \text{ signifikant,} \\ 0, & \text{sonst.} \end{cases} \quad (\text{F.6})$$

Ein quantisierter Transformationskoeffizient wird als signifikant bezeichnet, wenn sein Betrag einen Wert größer als Null annimmt ($|c_{x,y}^l| > 0$). Die Menge $\mathcal{D}_{x,y}^l$ gilt als insignifikant, wenn alle Elemente dieser Menge insignifikant sind. Anderenfalls gilt sie als signifikant. Ist $\mathcal{D}_{x,y}^l$ insignifikant, so sind für $k > l$ auch alle $\mathcal{D}_{x,y}^k$ insignifikant. Die Bestimmung der Signifikanzinformation beginnt mit $\mathcal{D}_{x,y}^l$, $l = 1$. Da grundsätzlich die Bestimmung der Signifikanzinformation des DC-Koeffizienten $c_{x,y}^0$ erfolgt, ist die Bestimmung von $S(\mathcal{D}_{x,y}^0)$ nicht notwendig. Ist $\mathcal{D}_{x,y}^l$ signifikant, so wird diese Menge in $c_{x,y}^l$ und $\mathcal{D}_{x,y}^{l+1}$ geteilt. Diese Zerlegung wird solange wiederholt, bis sich keine signifikanten Koeffizienten mehr in der Menge $\mathcal{D}_{x,y}^{l+1}$ befinden. Für den Fall $S(\mathcal{D}_{x,y}^l) = 1$ und $S(c_{x,y}^l) = 0$ ist die Codierung und Übertragung von $S(\mathcal{D}_{x,y}^{l+1})$ nicht

erforderlich, da in diesem Fall immer $S(\mathcal{D}_{x,y}^{l+1}) = 1$ gilt. Ebenso ist die Codierung von $S(c_{x,y}^{63})$ nicht erforderlich, da aufgrund der Gleichheit von $S(c_{x,y}^{63}) = S(\mathcal{D}_{x,y}^{63})$ die Information schon mit $S(\mathcal{D}_{x,y}^{63})$ übertragen wird.

Codierung der Beträge

Ist ein Koeffizient $c_{x,y}^l$ signifikant, so wird sein Betrag codiert und übertragen. Hierzu wird der Betrag n in eine binäre Darstellung überführt, die sich aus $n - 1$ Nullen gefolgt durch eine Eins ergibt.

Codierung der Vorzeichen

Zur Übertragung der Vorzeichen der signifikanten Koeffizienten $c_{x,y}^l$ gelten die Zuordnungen $- \rightarrow „1“$ und $+ \rightarrow „0“$ als vereinbart.

Die bisherigen Arbeitsschritte zur L-CEB-Codierung sollen anhand eines einfachen Beispiels verdeutlicht werden, welches [TT02a] entnommen wurde. Es sei ein 8×8 Block von quantisierten Transformationskoeffizienten gegeben [Wal92]:

$$c_{x,y} = \begin{bmatrix} 3 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (\text{F.7})$$

Das L-CEB-Verfahren bildet diesen Block auf den Bitstrom

$$\begin{array}{cccccccc} S(c_{x,y}^0) & 3 & + & S(\mathcal{D}_{x,y}^1) & S(c_{x,y}^1) & S(c_{x,y}^2) & 2 & - & S(\mathcal{D}_{x,y}^3) \\ \mathbf{1} & \mathbf{001} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{01} & \mathbf{1} & \mathbf{1} \\ \\ S(c_{x,y}^3) & 1 & - & S(\mathcal{D}_{x,y}^4) & S(c_{x,y}^4) & 1 & - & S(\mathcal{D}_{x,y}^5) & S(c_{x,y}^5) \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \\ 1 & - & S(\mathcal{D}_{x,y}^6) & S(c_{x,y}^6) & S(c_{x,y}^7) & S(c_{x,y}^8) & 1 & - & S(\mathcal{D}_{x,y}^9) \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} \end{array} \quad (\text{F.8})$$

ab. Im Anschluss an die bisher dargestellten Verarbeitungsstufen werden zur Reduktion der Datenmenge die Signifikanz- und Amplitudeninformationen mittels binärer arithmetischer Codierung unter Nutzung verschiedener adaptiver Wahrscheinlichkeitsmodelle komprimiert. Auf eine Entropiecodierung der Vorzeichenbits wird im L-CEB-Codec hingegen verzichtet.

Die arithmetische Codierung der Signifikanzinformationen erfolgt kontextabhängig, indem die Signifikanzinformationen benachbarter Koeffizienten bzw. benachbarter Mengen bei der Auswahl des jeweils zu verwendenden Wahrscheinlichkeitsmodells Berücksichtigung finden.

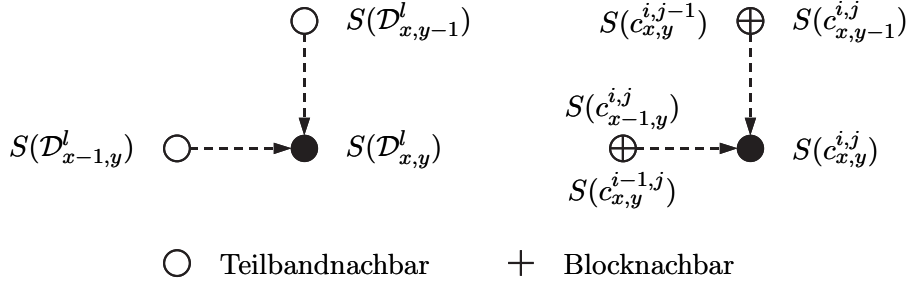


Abbildung F.5: Kontextmodellierung [TT02a].

Hierfür werden gemäß Abbildung F.5 mit den Blocknachbarn und den Teilbandnachbarn zwei verschiedene Typen von benachbarten Elementen unterschieden. Die kontextbasierte Auswahl des Wahrscheinlichkeitsmodells zur arithmetischen Codierung der Signifikanzinformation der Menge $\mathcal{D}_{x,y}^l$ erfolgt in Abhängigkeit von den Signifikanzinformationen des linken Teilbandnachbarn $\mathcal{D}_{x-1,y}^l$ und des oberen Teilbandnachbarn $\mathcal{D}_{x,y-1}^l$ und wird durch Gleichung [TT02a]

$$S(\mathcal{D}_{x,y}^l) : S(\mathcal{D}_{x-1,y}^l) + S(\mathcal{D}_{x,y-1}^l) + \begin{cases} 0, & \text{falls } l = 1, \\ 3, & \text{falls } l = 2, \\ 6, & \text{falls } 2 < l < 15, \\ 9, & \text{falls } l > 14 \end{cases} \quad (\text{F.9})$$

beschrieben. Bei der Auswahl des Wahrscheinlichkeitsmodells zur Codierung der Signifikanzinformation von $c_{x,y}^{i,j}$ fließen hingegen neben den Signifikanzinformationen des linken Teilbandnachbarn $c_{x-1,y}^{i,j}$ und des oberen Teilbandnachbarn $c_{x,y-1}^{i,j}$ in Abhängigkeit von der Teilbandzugehörigkeit von $c_{x,y}^{i,j}$ auch die Signifikanzinformationen des linken Blocknachbarn $c_{x,y}^{i-1,j}$ und des oberen Blocknachbarn $c_{x,y}^{i,j-1}$ ein. Hierzu werden entsprechend Abbildung F.6 die Teilbandtypen

- 1 × DC Teilband
- 1 × PV Teilband (principal vertical subband)

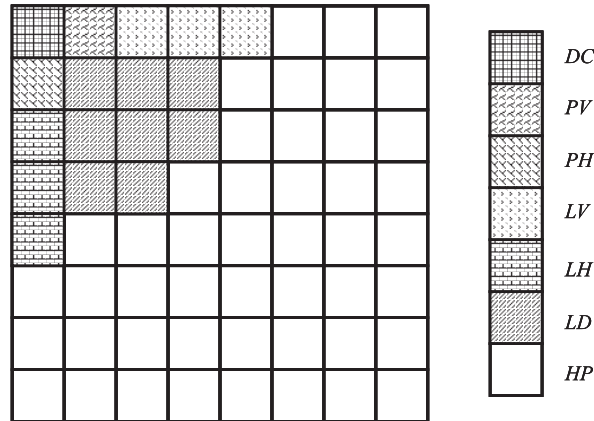


Abbildung F.6: Teilbandklassifikation [TT02a].

- 1× PH Teilband (principal horizontal subband)
- 3× LV Teilband (low-frequency vertical subband)
- 3× LH Teilband (low-frequency horizontal subband)
- 8× LD Teilband (low-frequency diagonal subband)
- 47× HP Teilband (high-pass subband)

klassifiziert. Die kontextbasierte Auswahl des adaptiven Wahrscheinlichkeitsmodells erfolgt schließlich nach Gleichung [TT02a]

$$S(c_{x,y}^{i,j}) : S(c_{x-1,y}^{i,j}) + S(c_{x,y-1}^{i,j}) + \begin{cases} 12, & \text{falls } c_{x,y}^{i,j} \in DC, \\ 15, & \text{falls } c_{x,y}^{i,j} \in PV, \\ 18, & \text{falls } c_{x,y}^{i,j} \in PH, \\ 21 + S(c_{x,y}^{i-1,j}), & \text{falls } c_{x,y}^{i,j} \in LV, \\ 25 + S(c_{x,y}^{i,j-1}), & \text{falls } c_{x,y}^{i,j} \in LH, \\ 29 + S(c_{x,y}^{i-1,j}) + S(c_{x,y}^{i,j-1}), & \text{falls } c_{x,y}^{i,j} \in LD, \\ 34 + S(c_{x,y}^{i-1,j}) + S(c_{x,y}^{i,j-1}), & \text{falls } c_{x,y}^{i,j} \in HP. \end{cases} \quad (\text{F.10})$$

Der L-CEB-Coder verwendet somit insgesamt 38 adaptive Wahrscheinlichkeitsmodelle zur Codierung der Signifikanzinformationen. Es kommen 4 weitere adaptive Wahrscheinlichkeitsmodelle zur Codierung der Amplitudeninformationen hinzu: eines für die DC-Koeffizienten, eines für die PV- und PH-Koeffizienten, eines für die LV-, LH-, und LD-Koeffizienten sowie eines für die HP-Koeffizienten. Bei der Codierung der Amplitudeninformationen werden im Gegensatz zur Codierung der Signifikanzinformationen keine benachbarten Elemente bei der Auswahl des Wahrscheinlichkeitsmodells berücksichtigt.

F.3 Grundprinzip der AGU-Codierung

Bei der AGU-Codierung [PELA05] werden die quantisierten Transformationskoeffizienten auf Bitebene codiert. Eine Bitebene umfasst jeweils die Menge aller gleichwertigen Bits. Die Codierung beginnt mit der höchstwertigen Bitschicht k_0 und endet mit der Verarbeitung der niedrigstwertigen Bitschicht. Die im AGU-Codec realisierte kontextbasierte adaptive arithmetische Codierung verwendet insgesamt 15 Wahrscheinlichkeitsmodelle. Die Auswahl des Wahrscheinlichkeitsmodells zur Codierung eines Symbols erfolgt in Abhängigkeit vom

- eigenen Zustand in der aktuellen Bitschicht,
- eigenen Zustand in den bereits codierten höheren Bitschichten,
- Zustand seiner Block- und Teilbandnachbarn in der aktuellen Bitschicht,
- Zustand seiner Block- und Teilbandnachbarn in den bereits codierten höheren Bitschichten.

Abbildung F.7 zeigt das hierzu verwendete Flussdiagramm zur Auswahl des Wahrscheinlichkeitsmodells ($PM1 - PM15$) im Detail, wobei nach [PELA05] folgende Entscheidungsregeln ($C1 - C14$) Anwendung finden:

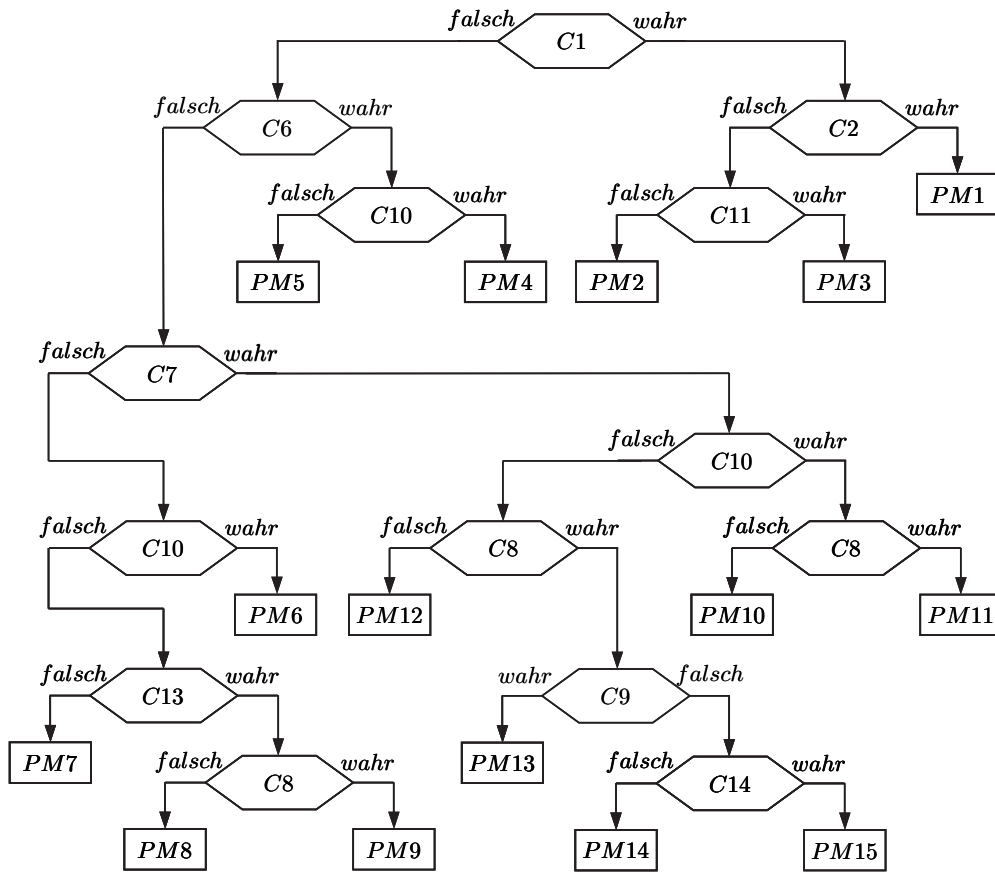


Abbildung F.7: Flussdiagramm zur Auswahl des Wahrscheinlichkeitsmodells bei der adaptiven arithmetischen Codierung im AGU-Codec [PELA05].

Notation: $P_{x,y}^k(i,j)$ beschreibt für den Koeffizienten mit der Koordinate (i,j) , der im Koeffizientenblock (x,y) liegt, den Bitwert der Bitebene k .

- Die Bedingung $C1$ ist *wahr*, wenn für den betrachteten Koeffizienten wenigstens ein Bit in einer der höheren und bereits codierten Bitebenen gleich 1 ist.

$$C1(k, x, y, i, j) = \text{wahr}, \quad \text{wenn } 1 \in \{P_{x,y}^{k+1}(i, j), \dots, P_{x,y}^n(i, j)\}.$$

- Die Bedingung $C2$ ist *wahr*, wenn für den betrachteten Koeffizienten wenigstens ein Bit in einer der höheren und bereits codierten Bitebene gleich 1 ist. Im Unterschied zu $C1$ wird jedoch die vorangegangene Bitebene nicht mit einbezogen.

$$C2(k, x, y, i, j) = \text{wahr}, \quad \text{wenn } 1 \in \{P_{x,y}^{k+2}(i, j), \dots, P_{x,y}^n(i, j)\}.$$

- Die Bedingung $C3$ ist *wahr*, wenn für den betrachteten Koeffizienten wenigstens ein Bit in der aktuellen oder in einer der höheren und bereits codierten Bitebene gleich 1 ist.

$$C3(k, x, y, i, j) = \text{wahr}, \quad \text{wenn } 1 \in \{P_{x,y}^k(i, j), \dots, P_{x,y}^n(i, j)\}.$$

Die Bedingung $C3$ kommt in dem in Abbildung F.7 dargestellten Entscheidungsbaum nicht direkt zu Anwendung, sondern wird für die Klassifikationsregeln $C8$, $C9$ und $C10$ benötigt.

- Die Bedingung $C4$ ist *wahr*, wenn für den betrachteten Koeffizienten das Bit der vorangegangene Bitebene gleich 1 ist.

$$C4(k, x, y, i, j) = \text{wahr}, \quad \text{wenn } P_{x,y}^{k+1}(i, j) = 1.$$

- Die Bedingung $C5$ ist *wahr*, wenn für den betrachteten Koeffizienten das Bit der aktuellen Bitebene gleich 1 ist.

$$C5(k, x, y, i, j) = \text{wahr}, \quad \text{wenn } P_{x,y}^k(i, j) = 1.$$

Die Bedingung $C5$ kommt in dem in Abbildung F.7 dargestellten Entscheidungsbaum nicht direkt zu Anwendung, sondern wird für die Klassifikationsregeln $C7$ und $C12$ benötigt.

- Die Bedingung $C6$ ist *wahr*, wenn wenigstens von einem Nachbarn des betrachteten Koeffizienten ein Bit in einer der höheren und bereits codierten Bitebenen gleich 1 ist.

$$C6(k, x, y, i, j) = \text{wahr}, \text{ wenn } \text{wahr} \in \{C1(k, x, y, i-1, j-1), \quad C1(k, x, y, i-1, j), \\ C1(k, x, y, i-1, j+1), \quad C1(k, x, y, i, j-1), \\ C1(k, x, y, i, j+1), \quad C1(k, x, y, i+1, j-1), \\ C1(k, x, y, i+1, j), \quad C1(k, x, y, i+1, j+1)\}.$$

- Die Bedingung $C7$ ist *wahr*, wenn wenigstens von einem Nachbarn des betrachteten Koeffizienten in der aktuellen Bitebene das bereits codierte Bit gleich 1 ist.

$$C7(k, x, y, i, j) = \text{wahr}, \text{ wenn } \text{wahr} \in \{C5(k, x, y, i-1, j-1), \quad C5(k, x, y, i-1, j), \\ C5(k, x, y, i-1, j+1), \quad C5(k, x, y, i, j-1)\}.$$

- Die Bedingung $C8$ ist *wahr*, wenn wenigstens von einem Nachbarn, der vom betrachteten Koeffizienten 2 Reihen bzw. 2 Spalten entfernt ist, ein bereits codiertes Bit in der aktuellen oder in einer höheren Bitebene gleich 1 ist.

$$C8(k, x, y, i, j) = \text{wahr}, \text{ wenn } \text{wahr} \in \{C3(k, x, y, i-2, j-2), \quad C3(k, x, y, i-2, j-1), \\ C3(k, x, y, i-2, j), \quad C3(k, x, y, i-2, j+1), \\ C3(k, x, y, i-1, j-2), \quad C3(k, x, y, i-1, j+2), \\ C3(k, x, y, i, j-2), \quad C3(k, x, y, i, j+2), \\ C3(k, x, y, i+1, j-2), \quad C3(k, x, y, i+1, j+2), \\ C3(k, x, y, i+2, j-2), \quad C3(k, x, y, i+2, j-1), \\ C3(k, x, y, i+2, j), \quad C3(k, x, y, i+2, j+1), \\ C3(k, x, y, i+2, j+2)\}.$$

- Die Bedingung $C9$ ist *wahr*, wenn wenigstens von einem Nachbarn, der vom betrachteten Koeffizienten 3 Reihen bzw. 3 Spalten entfernt ist, ein bereits codiertes Bit in der aktuellen oder in einer höheren Bitebene gleich 1 ist.

$$C9(k, x, y, i, j) = \text{wahr}, \text{ wenn } \text{wahr} \in \{C3(k, x, y, i-3, j-3), C3(k, x, y, i-3, j-2), \\ C3(k, x, y, i-3, j-1), C3(k, x, y, i-3, j), \\ C3(k, x, y, i-3, j+1), C3(k, x, y, i-3, j+2), \\ C3(k, x, y, i-3, j+3), C3(k, x, y, i-2, j-3), \\ C3(k, x, y, i-2, j+3), C3(k, x, y, i-1, j-3), \\ C3(k, x, y, i-1, j+3), C3(k, x, y, i, j-3), \\ C3(k, x, y, i, j+3), C3(k, x, y, i+1, j-3), \\ C3(k, x, y, i+1, j+3), C3(k, x, y, i+2, j-3), \\ C3(k, x, y, i+2, j+3), C3(k, x, y, i+3, j-3), \\ C3(k, x, y, i+3, j-2), C3(k, x, y, i+3, j-1), \\ C3(k, x, y, i+3, j), C3(k, x, y, i+3, j+1), \\ C3(k, x, y, i+3, j+2), C3(k, x, y, i+3, j+3)\}.$$

- Die Bedingung $C10$ ist *wahr*, wenn von wenigstens einem Blocknachbarn des betrachteten Koeffizienten ein Bit in der aktuellen oder in einer höheren Bitebene gleich 1 ist.

$$C10(k, x, y, i, j) = \text{wahr}, \text{ wenn } \text{wahr} \in \{C3(k, x-1, y-1, i, j), C3(k, x-1, y, i, j), \\ C3(k, x-1, y+1, i, j), C3(k, x, y-1, i, j), \\ C3(k, x, y+1, i, j), C3(k, x+1, y-1, i, j), \\ C3(k, x+1, y, i, j), C3(k, x+1, y+1, i, j)\}.$$

- Bis auf $C14$ basieren die restlichen Entscheidungsregeln ebenfalls auf bereits beschriebenen Entscheidungsregeln und lauten:

$$C11(k, x, y, i, j) = (C2(k, x, y, i, j) = \text{falsch}) \quad \text{und} \quad (C6(k+1, x, y, i, j) = \text{falsch}),$$

$$C12(k, x, y, i, j) = \begin{cases} 1, & \text{wenn } C5(k, x, y, i, j) = \text{wahr}, \\ 0, & \text{wenn } C5(k, x, y, i, j) = \text{falsch}, \end{cases}$$

$$C13(k, x, y, i, j) = \text{wahr}, \quad \text{wenn } 1 = C12(k, x, y, i-1, j-1) + C12(k, x, y, i-1, j) \\ + C12(k, x, y, i-1, j+1) + C12(k, x, y, i, j-1),$$

$$C14(k, x, y, i, j) = \text{wahr}, \quad \text{wenn } k = 1.$$

Für ein vertiefendes Studium der Entscheidungsregeln in der AGU-Codierung wird auf die zugrunde liegende Literatur [PELA05] verwiesen.

F.4 Filterparameter

In der Tabelle F.1 sind für alle untersuchten Transformationen und Codierungsstrategien die Thresholding-Parameter η zusammengefasst, die bei der in Abschnitt 5.5 zur Reduktion von Kompressionsartefakten betrachteten DCT-basierten Filterung des rekonstruierten Bildes zur Anwendung kommen. Hierbei werden alle Transformationskoeffizienten, die betragsmäßig unter der Schwelle $\eta \cdot Q$ (Q = Quantisierungsfaktor) liegen, auf Null gesetzt (Hard-Thresholding). Die Thresholding-Parameter wurden durch Testreihen unter Einbeziehung aller im Rahmen dieser Arbeit untersuchten Testbilder bestimmt. Sie liefern im Mittel für die jeweils betrachtete Kompressionsstrategie die höchsten PSNR-Werte. Kleinere Thresholding-Parameter reduzieren die Kompressionsartefakte nur unzureichend. Höhere Thresholding-Parameter führen in den rekonstruierten Bildern zu sichtbaren Detailverlusten.

	8×8		16×16		32×32	
	SPIHT	L-CEB	SPIHT	AGU	SPIHT	AGU
DCT	0.8	0.65	0.8	0.65	0.85	0.7
LCT-2	0.75	0.65	0.75	0.65	0.8	0.7
LCT-3	0.65	0.55	0.65	0.60	/	/
LCT-4	0.85	0.75	0.9	0.8	/	/
LCT-6	1.25	1.1	1.3	1.15	/	/
9/7-DWT	0.75	0.7	0.75	0.7	0.75	0.7
TDLT	0.5	0.45	/	/	/	/
LOT	0.85	0.7	/	/	/	/
GenLOT, L=4	0.8	0.65	/	/	/	/
GenLOT, L=6	0.75	0.65	/	/	/	/
ELT	0.75	0.65	/	/	/	/

Tabelle F.1: Thresholding-Parameter η in Abhängigkeit von den untersuchten Transformationen, Codierungsstrategien und Blockgrößen der Transformationskoeffizienten (bei der 9/7-DWT bezieht sich die Blockgröße auf die nach Abbildung 5.1 umsortierten Wavelet-Transformationskoeffizienten).

Anhang G

Weitere blocküberlappende Transformationen

Bezugnehmend auf Abschnitt 6.7 geben die nachfolgenden Abschnitte eine Zusammenfassung der Transformationsmatrizen bzw. Berechnungsalgorithmen von blocküberlappenden Transformationen, die beim Vergleich der Leistungsfähigkeit zwischen überlappender Kosinustransformation und bestehenden blocküberlappenden Transformationen verwendet wurden. Für ein tiefergehendes Studium der folgenden Transformationen wird auf die angegebene Literatur verwiesen.

G.1 Lapped Orthogonal Transform (LOT)

ψ^0	ψ^1	ψ^2	ψ^3	ψ^4	ψ^5	ψ^6	ψ^7
-0.068420	-0.075503	0.023102	-0.027515	0.037884	-0.000301	0.046898	0.048906
-0.031091	-0.091223	0.144443	0.078235	0.068420	0.142010	-0.092077	-0.019513
0.037884	-0.011525	0.149525	0.243855	-0.225549	-0.123838	0.023102	-0.016291
0.128004	0.153167	-0.092077	-0.030737	-0.031091	-0.102447	0.149525	0.115272
0.225549	0.337226	-0.369862	-0.384998	0.384644	0.380232	-0.340867	-0.212817
0.315669	0.427260	-0.340867	-0.146310	-0.128004	-0.366555	0.438837	0.294076
0.384644	0.369008	0.046898	0.412158	-0.421973	-0.044465	-0.369862	-0.396222
0.421973	0.173048	0.438837	0.305300	0.315669	0.416036	0.144443	0.441487
0.421973	-0.173048	0.438837	-0.305300	0.315669	-0.416036	0.144443	-0.441487
0.384644	-0.369008	0.046898	-0.412158	-0.421973	0.044465	-0.369862	0.396222
0.315669	-0.427260	-0.340867	0.146310	-0.128004	0.366555	0.438837	-0.294076
0.225549	-0.337226	-0.369862	0.384998	0.384644	-0.380232	-0.340867	0.212817
0.128004	-0.153167	-0.092077	0.030737	-0.031091	0.102447	0.149525	-0.115272
0.037884	0.011525	0.149525	-0.243855	-0.225549	0.123838	0.023102	0.016291
-0.031091	0.091223	0.144443	-0.078235	0.068420	-0.142010	-0.092077	0.019513
-0.068420	0.075503	0.023102	0.027515	0.037884	0.000301	0.046898	-0.048906

Tabelle G.1: Transformationsmatrix der LOT, Typ-II [Mal92], $B = 8$.

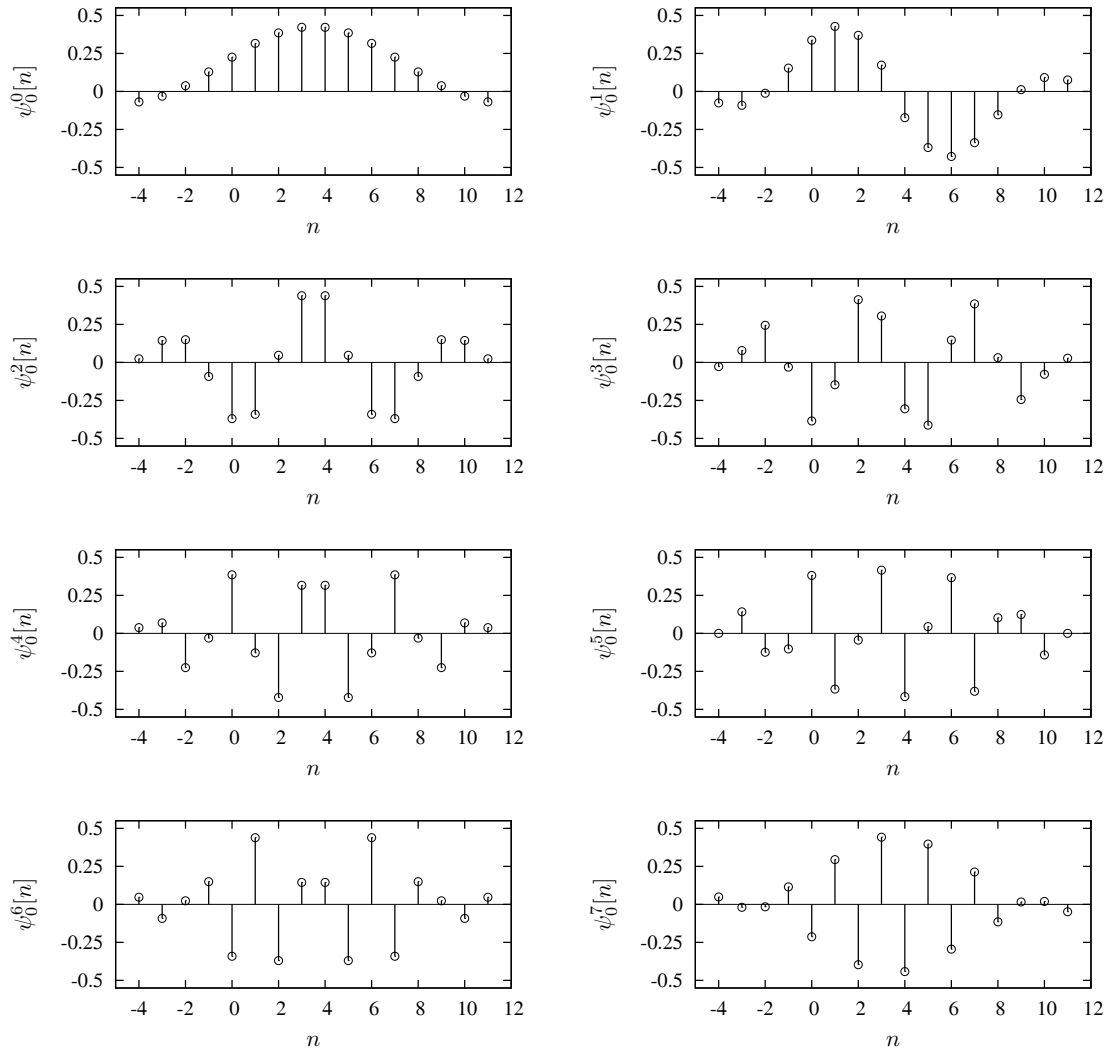


Abbildung G.1: Zweifach überlappende Basisfunktionen der Typ-II LOT zur Signalrekonstruktion und Signalanalyse, $B = 8$.

G.2 Generalized Linear-Phase Lapped Orthogonal Transform (Genlot)

Die Transformationsmatrizen der vierfach überlappenden 8×32 GenLOT und der sechsfach überlappenden 8×48 GenLOT wurden mit dem GenLOT-Design-Center von Trautmann und Nguyen [TN, SN96] entworfen.

ψ^0	ψ^1	ψ^2	ψ^3	ψ^4	ψ^5	ψ^6	ψ^7
0.004569	0.004668	0.002954	-0.002813	0.000614	0.000459	0.000371	0.000451
0.008428	-0.000269	-0.005460	-0.009602	0.001337	-0.003172	0.001265	-0.001380
0.007578	-0.005842	-0.011079	-0.011611	0.001988	-0.003491	-0.000355	-0.001764
-0.010807	-0.006944	-0.001821	0.008817	-0.002344	0.001508	-0.001787	-0.000755
-0.034613	-0.009307	0.007977	0.034712	-0.006784	0.003893	-0.000181	0.001039
-0.032056	-0.015216	0.001125	0.028746	-0.006571	0.006439	-0.005246	-0.000449
-0.015177	-0.030303	-0.027436	0.001732	-0.000603	-0.006991	-0.001530	-0.003953
-0.000715	-0.034634	-0.041583	-0.015759	-0.003310	-0.012109	-0.000679	-0.012105
-0.015325	-0.057511	-0.000088	-0.083582	0.051058	-0.014151	0.054535	0.046913
-0.017844	-0.087582	0.123542	0.045817	0.093407	0.138809	-0.097095	-0.024432
0.018796	-0.050777	0.135569	0.224676	-0.211225	-0.136731	0.012469	-0.028788
0.123682	0.110677	-0.111339	-0.031813	-0.045252	-0.107894	0.150400	0.112575
0.275292	0.331794	-0.407183	-0.380484	0.343594	0.378398	-0.318247	-0.219608
0.359236	0.452072	-0.252738	-0.130985	-0.175083	-0.343116	0.447053	0.314095
0.378147	0.370724	0.137314	0.420213	-0.402741	-0.041647	-0.379924	-0.387697
0.365024	0.140576	0.450245	0.313566	0.361914	0.432748	0.138951	0.431670
0.365024	-0.140576	0.450245	-0.313566	0.361914	-0.432748	0.138951	-0.431670
0.378147	-0.370724	0.137314	-0.420213	-0.402741	0.041647	-0.379924	0.387697
0.359236	-0.452072	-0.252738	0.130985	-0.175083	0.343116	0.447053	-0.314095
0.275292	-0.331794	-0.407183	0.380484	0.343594	-0.378398	-0.318247	0.219608
0.123682	-0.110677	-0.111339	0.031813	-0.045252	0.107894	0.150400	-0.112575
0.018796	0.050777	0.135569	-0.224676	-0.211225	0.136731	0.012469	0.028788
-0.017844	0.087582	0.123542	-0.045817	0.093407	-0.138809	-0.097095	0.024432
-0.015325	0.057511	-0.000088	0.083582	0.051058	0.014151	0.054535	-0.046913
-0.000715	0.034634	-0.041583	0.015759	-0.003310	0.012109	-0.000679	0.012105
-0.015177	0.030303	-0.027436	-0.001732	-0.000603	0.006991	-0.001530	0.003953
-0.032056	0.015216	0.001125	-0.028746	-0.006571	-0.006439	-0.005246	0.000449
-0.034613	0.009307	0.007977	-0.034712	-0.006784	-0.003893	-0.000181	-0.001039
-0.010807	0.006944	-0.001821	-0.008817	-0.002344	-0.001508	-0.001787	0.000755
0.007578	0.005842	-0.011079	0.011611	0.001988	0.003491	-0.000355	0.001764
0.008428	0.000269	-0.005460	0.009602	0.001337	0.003172	0.001265	0.001380
0.004569	-0.004668	0.002954	0.002813	0.000614	-0.000459	0.000371	-0.000451

Tabelle G.2: Transformationsmatrix der vierfach überlappenden GenLOT, $B = 8$.

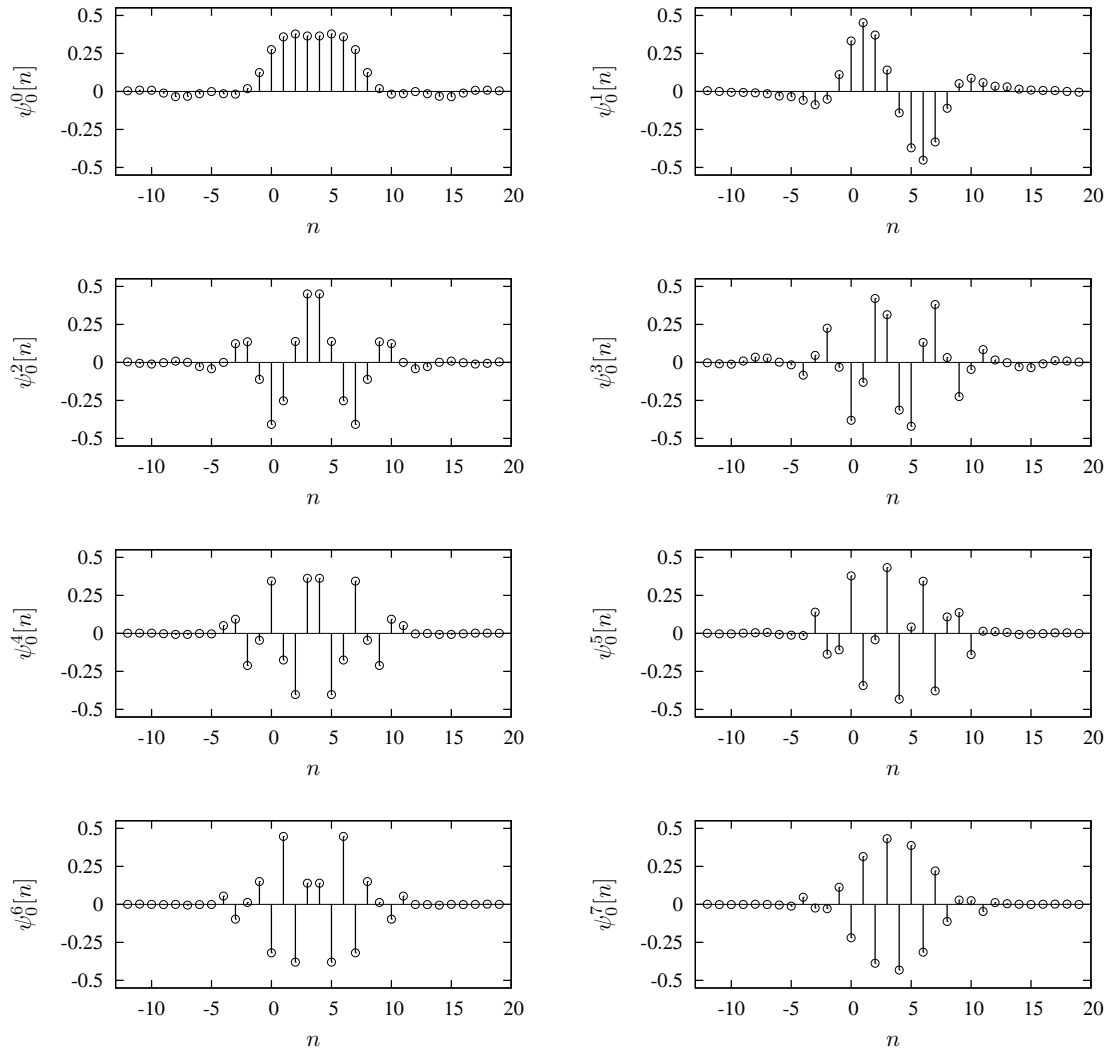


Abbildung G.2: Vierfach überlappende Basisfunktionen der GenLOT zur Signalrekonstruktion und Signalanalyse, $B = 8$.

ψ^0	ψ^1	ψ^2	ψ^3	ψ^4	ψ^5	ψ^6	ψ^7
-0.000118	-0.000410	-0.000131	0.000280	-0.000668	-0.000189	0.000077	-0.000449
-0.000306	0.000919	-0.000497	0.000534	0.000221	0.001167	-0.001578	-0.000621
0.000673	0.000222	-0.000035	-0.000608	0.000554	0.000677	-0.000391	0.000192
0.001694	0.000424	0.001729	-0.002770	0.002266	-0.000694	0.001528	0.002235
-0.001188	0.002063	0.000405	0.000390	0.002107	0.000083	-0.000466	0.001348
0.004208	0.000098	0.004457	-0.006895	0.004981	-0.002696	0.004994	0.005701
0.000276	0.002517	-0.000303	-0.000559	0.003279	0.002239	-0.001808	0.001589
0.006223	0.007830	0.005300	-0.010040	0.012354	0.003876	-0.002441	0.006948
0.006524	0.009742	0.005859	-0.005495	0.007344	0.001685	-0.001255	0.001537
0.019226	-0.019227	0.016073	-0.023461	-0.014268	0.000029	0.011475	0.007985
-0.001022	-0.029539	-0.012558	0.013070	-0.030851	0.005008	-0.006205	-0.019785
-0.020274	-0.007404	-0.018943	0.010590	0.015727	0.004041	-0.010854	-0.005714
-0.031899	0.017255	0.010403	-0.006368	0.027442	-0.023496	0.024341	0.042593
-0.046064	-0.013231	-0.044865	0.026293	0.033343	-0.000492	-0.028221	-0.032872
-0.047269	-0.064086	-0.038658	0.053035	-0.067402	-0.006004	0.002102	-0.013705
-0.041386	-0.109322	0.010115	-0.043771	-0.016187	0.012619	-0.016760	0.000660
-0.019907	-0.115283	0.025025	-0.093958	0.015169	-0.012933	0.091885	0.058537
0.010740	-0.121231	0.153068	0.076742	0.162195	0.141510	-0.114433	-0.051556
0.078843	-0.052799	0.197549	0.218068	-0.255700	-0.160059	0.039911	-0.072807
0.161214	0.130294	-0.080245	-0.082939	-0.024410	-0.125011	0.141475	0.171451
0.244006	0.339528	-0.379461	-0.352555	0.330941	0.394835	-0.335943	-0.223915
0.316915	0.430968	-0.339172	-0.108349	-0.143571	-0.335281	0.418920	0.272543
0.370886	0.347875	0.085414	0.436423	-0.397755	-0.028575	-0.375661	-0.346268
0.402218	0.110560	0.399470	0.312840	0.332890	0.410627	0.159306	0.459604
0.402218	-0.110560	0.399470	-0.312840	0.332890	-0.410627	0.159306	-0.459604
0.370886	-0.347875	0.085414	-0.436423	-0.397755	0.028575	-0.375661	0.346268
0.316915	-0.430968	-0.339172	0.108349	-0.143571	0.335281	0.418920	-0.272543
0.244006	-0.339528	-0.379461	0.352555	0.330941	-0.394835	-0.335943	0.223915
0.161214	-0.130294	-0.080245	0.082939	-0.024410	0.125011	0.141475	-0.171451
0.078843	0.052799	0.197549	-0.218068	-0.255700	0.160059	0.039911	0.072807
0.010740	0.121231	0.153068	-0.076742	0.162195	-0.141510	-0.114433	0.051556
-0.019907	0.115283	0.025025	0.093958	0.015169	0.012933	0.091885	-0.058537
-0.041386	0.109322	0.010115	0.043771	-0.016187	-0.012619	-0.016760	-0.000660
-0.047269	0.064086	-0.038658	-0.053035	-0.067402	0.006004	0.002102	0.013705
-0.046064	0.013231	-0.044865	-0.026293	0.033343	0.000492	-0.028221	0.032872
-0.031899	-0.017255	0.010403	0.006368	0.027442	0.023496	0.024341	-0.042593
-0.020274	0.007404	-0.018943	-0.010590	0.015727	-0.004041	-0.010854	0.005714
-0.001022	0.029539	-0.012558	-0.013070	-0.030851	-0.005008	-0.006205	0.019785
0.019226	0.019227	0.016073	0.023461	-0.014268	-0.000029	0.011475	-0.007985
0.006524	-0.009742	0.005859	0.005495	0.007344	-0.001685	-0.001255	-0.001537
0.006223	-0.007830	0.005300	0.010040	0.012354	-0.003876	-0.002441	-0.006948
0.000276	-0.002517	-0.000303	0.000559	0.003279	-0.002239	-0.001808	-0.001589
0.004208	-0.000098	0.004457	0.006895	0.004981	0.002696	0.004994	-0.005701
-0.001188	-0.002063	0.000405	-0.000390	0.002107	-0.000083	-0.000466	-0.001348
0.001694	-0.000424	0.001729	0.002770	0.002266	0.000694	0.001528	-0.002235
0.000673	-0.000222	-0.000035	0.000608	0.000554	-0.000677	-0.000391	-0.000192
-0.000306	-0.000919	-0.000497	-0.000534	0.000221	-0.001167	-0.001578	0.000621
-0.000118	0.000410	-0.000131	-0.000280	-0.000668	0.000189	0.000077	0.000449

Tabelle G.3: Transformationsmatrix der sechsfach überlappenden GenLOT, $B = 8$.

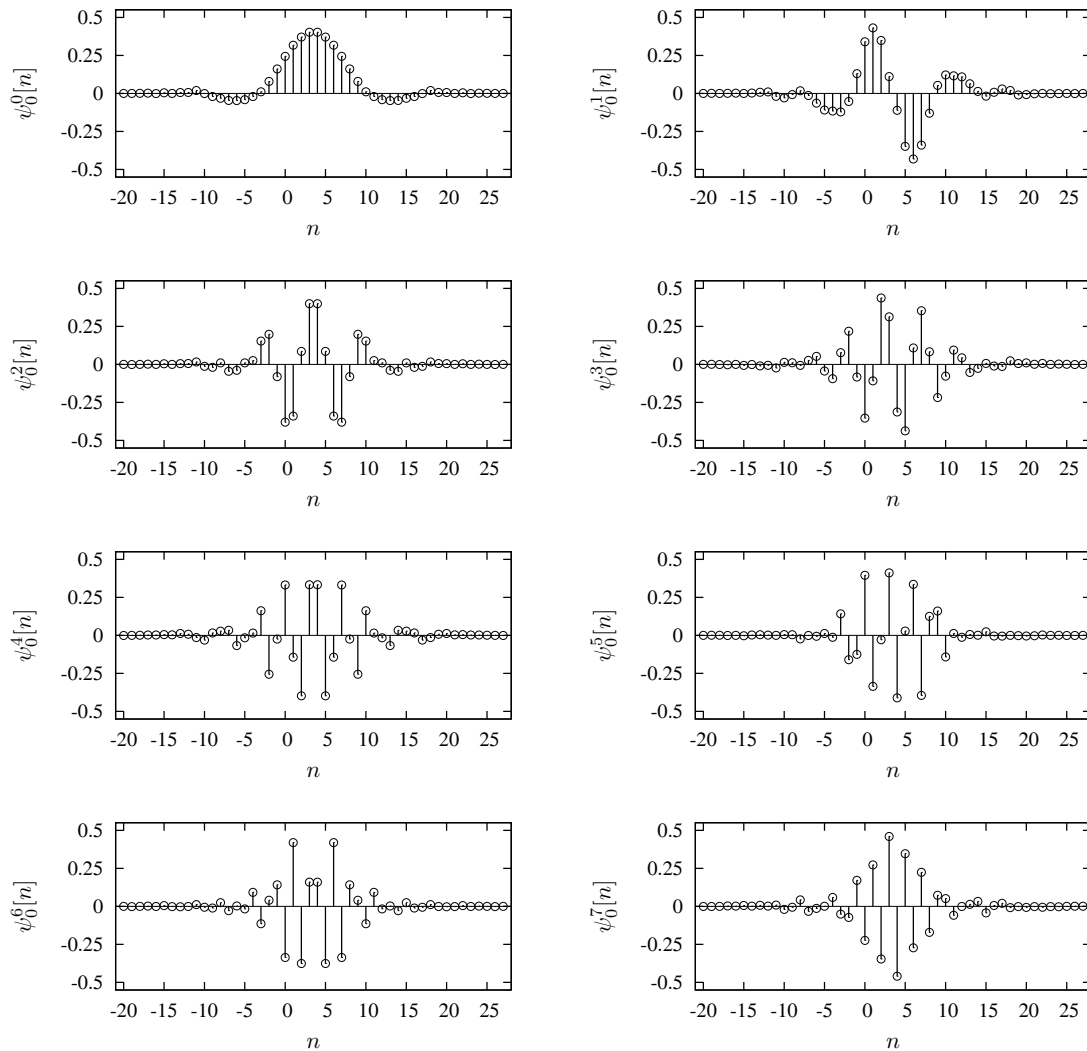


Abbildung G.3: Sechsfach überlappende Basisfunktionen der GenLOT zur Signalrekonstruktion und Signalanalyse, $B = 8$.

G.3 Time-Domain Lapped Transform (TDLT)

Die TDLT basiert auf der DCT und filtert zur Reduktion von Blockartefakten auf der Analyseseite das Bildsignal in der Umgebung der Blockgrenzen. Auf der SyntheseSeite erfolgt nach der inversen Kosinuustransformation eine Nachfilterung des rekonstruierten Signals.

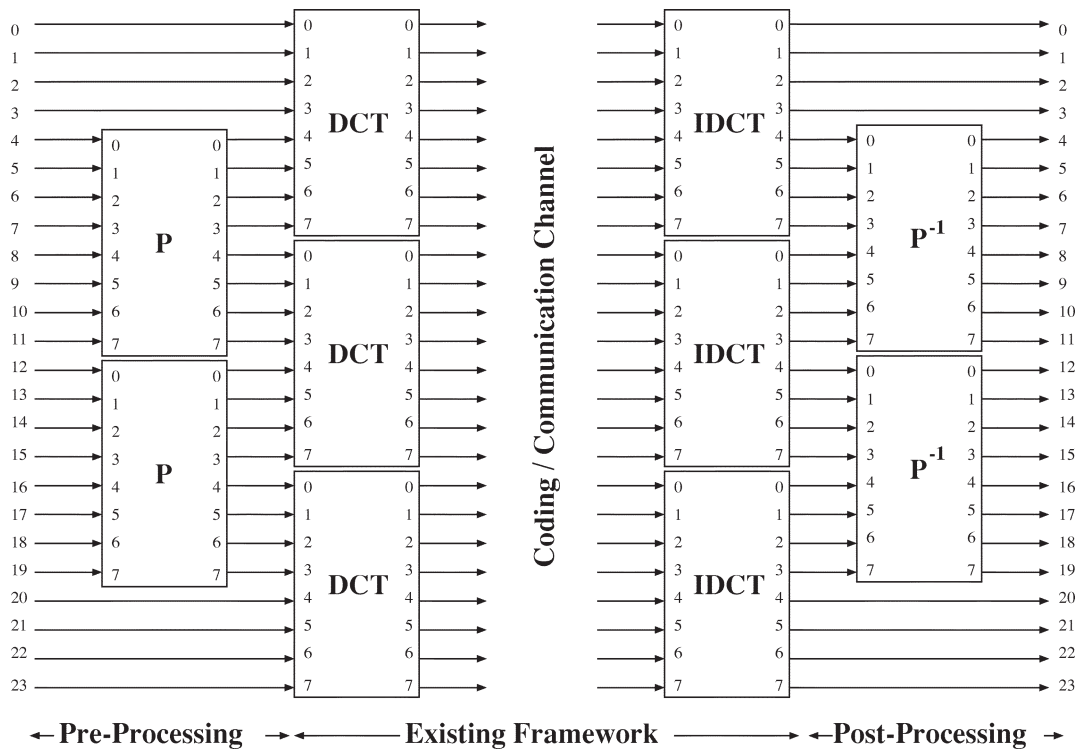


Abbildung G.4: Pre- und Postfilterung an den Blockgrenzen, entnommen aus [TT02a].

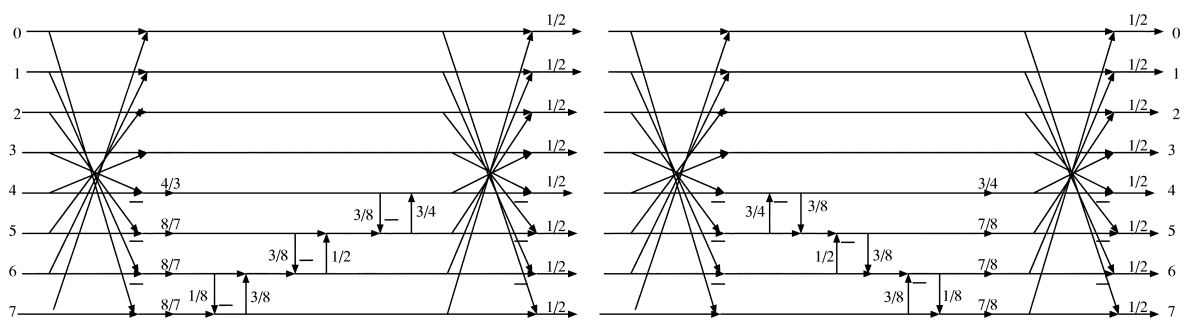


Abbildung G.5: Prefilter (links) und Postfilter (rechts), entnommen aus [TT02a].

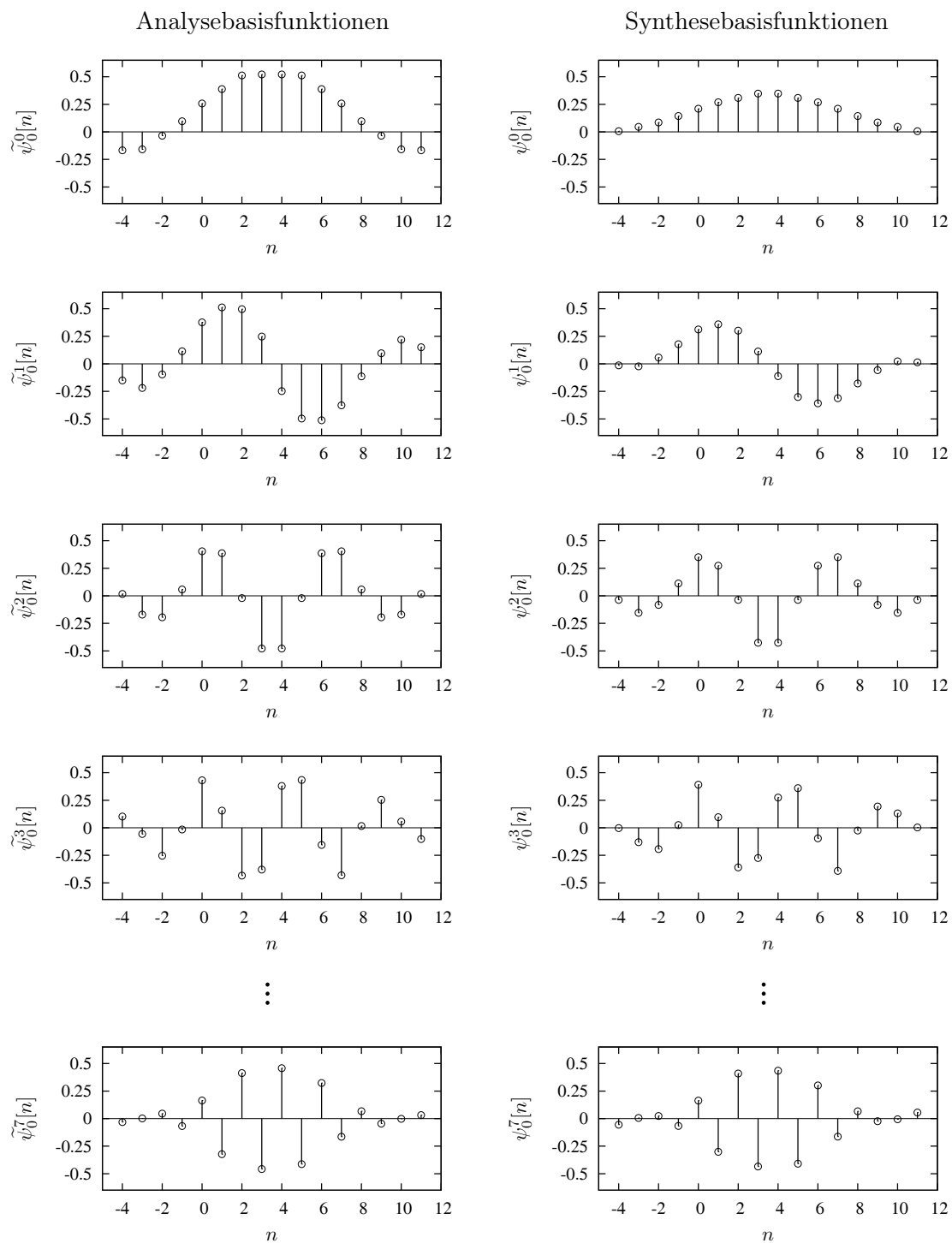


Abbildung G.6: Zweifach überlappende Basisfunktionen der TDLT zur Signalanalyse (links) und zur Signalrekonstruktion (rechts).

G.4 Extended Lapped Transform (ELT)

Bei den Basisfunktionen der untersuchten vierfach überlappenden ELT handelt es sich um Kosinus-IV-Funktionen, die mit einer vierfach überlappenden Fensterfunktion gewichtet werden. Diese Fensterfunktion, deren Verlauf über einen Parameter γ beeinflusst werden kann, ist in [Mal92] beschrieben. Zur Transformation von Signalen, die auf einem endlich langen Intervall definiert sind, lassen sich die im Rahmen dieser Arbeit entwickelten Klapp- und Aufklappoperatoren nutzen. Die hierbei verwendeten Randfenster wurden nach dem in Abschnitt 3.5.5 entwickelten Verfahren entworfen.

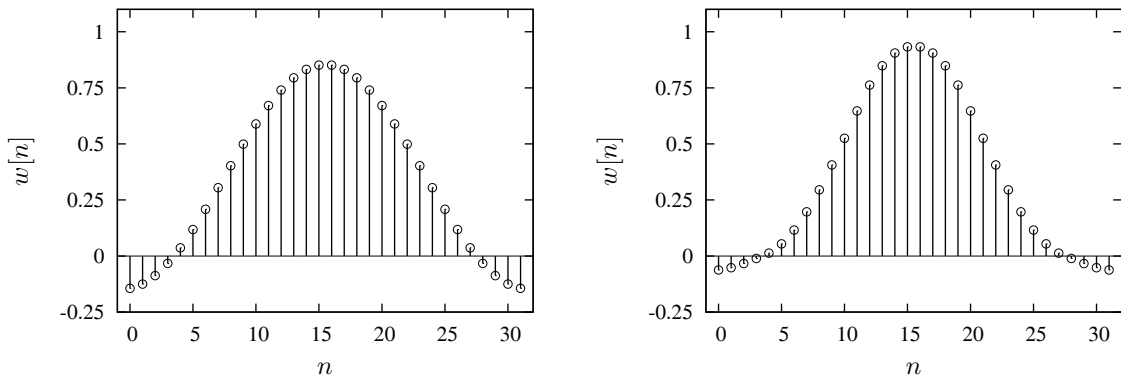


Abbildung G.7: Fenster der vierfach überlappenden ELT für $\gamma = 1.0$ (links) und $\gamma = 0.3$ (rechts), $B=8$.

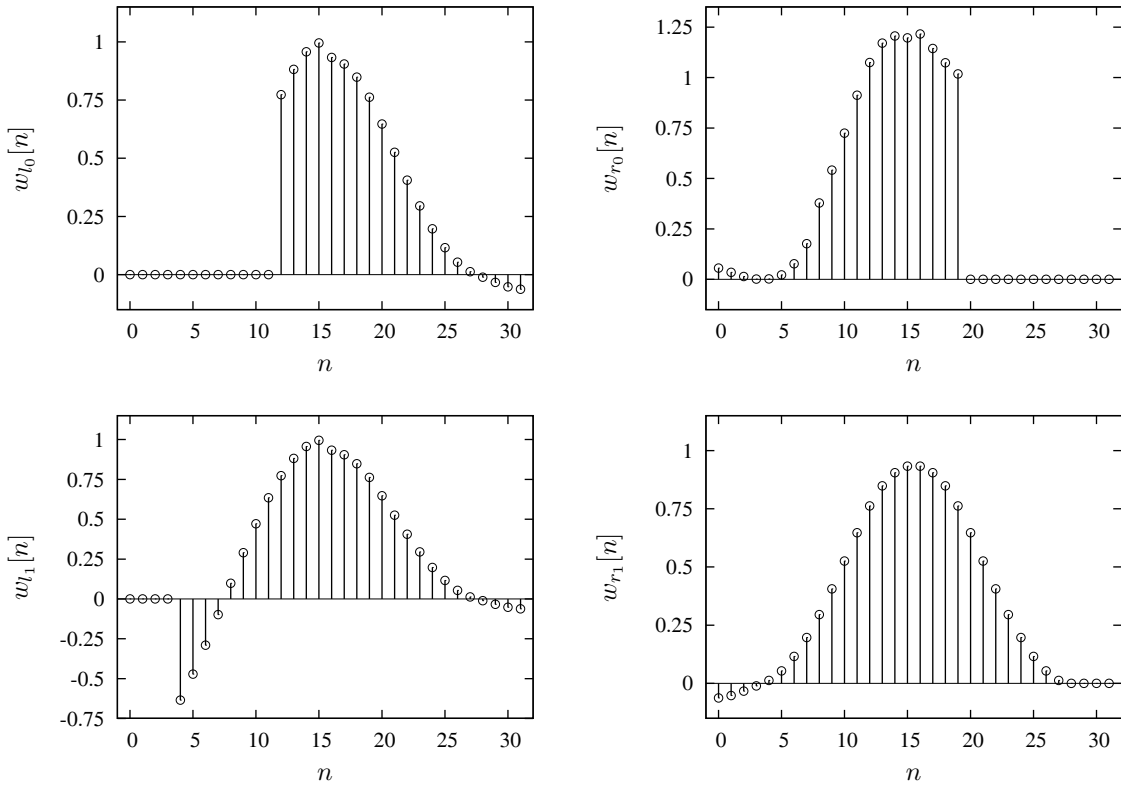


Abbildung G.8: Randfenster der vierfach überlappenden ELT für $\gamma = 0.3$, $B = 8$.

w_{l_0}	w_{l_1}	w	w_{r_1}	w_{r_0}
0.00000000	0.00000000	-0.06239773	-0.06239773	0.05598122
0.00000000	0.00000000	-0.05198987	-0.05198987	0.03465934
0.00000000	0.00000000	-0.03340858	-0.03340858	0.01371504
0.00000000	0.00000000	-0.01092266	-0.01092266	0.00151407
0.00000000	-0.63439328	0.01286053	0.01286053	0.00178269
0.00000000	-0.47139674	0.05395889	0.05395889	0.02215145
0.00000000	-0.29028468	0.11584531	0.11584531	0.07722893
0.00000000	-0.09801714	0.19717354	0.19717354	0.17689771
0.00000000	0.09801714	0.29519068	0.29519068	0.37868422
0.00000000	0.29028468	0.40612999	0.40612999	0.54119481
0.00000000	0.47139674	0.52535563	0.52535563	0.72468827
0.00000000	0.63439328	0.64725381	0.64725381	0.91246150
0.77301045	0.77301045	0.76208779	0.76208779	1.07434790
0.88192126	0.88192126	0.84851268	0.84851268	1.17045893
0.95694034	0.95694034	0.90495047	0.90495047	1.20590579
0.99518473	0.99518473	0.93278700	0.93278700	1.19662219
0.93278700	0.93278700	0.93278700	0.93278700	1.21555048
0.90495047	0.90495047	0.90495047	0.90495047	1.14448517
0.84851268	0.84851268	0.84851268	0.84851268	1.07302354
0.76208779	0.76208779	0.76208779	0.76208779	1.01809996
0.64725381	0.64725381	0.64725381	0.64725381	0.00000000
0.52535563	0.52535563	0.52535563	0.52535563	0.00000000
0.40612999	0.40612999	0.40612999	0.40612999	0.00000000
0.29519068	0.29519068	0.29519068	0.29519068	0.00000000
0.19717354	0.19717354	0.19717354	0.19717354	0.00000000
0.11584531	0.11584531	0.11584531	0.11584531	0.00000000
0.05395889	0.05395889	0.05395889	0.05395889	0.00000000
0.01286053	0.01286053	0.01286053	0.01286053	0.00000000
-0.01092266	-0.01092266	-0.01092266	0.00000000	0.00000000
-0.03340858	-0.03340858	-0.03340858	0.00000000	0.00000000
-0.05198987	-0.05198987	-0.05198987	0.00000000	0.00000000
-0.06239773	-0.06239773	-0.06239773	0.00000000	0.00000000

Tabelle G.4: Randfenster und inneres Fenster der ELT für $\gamma = 0.3$, $B = 8$.

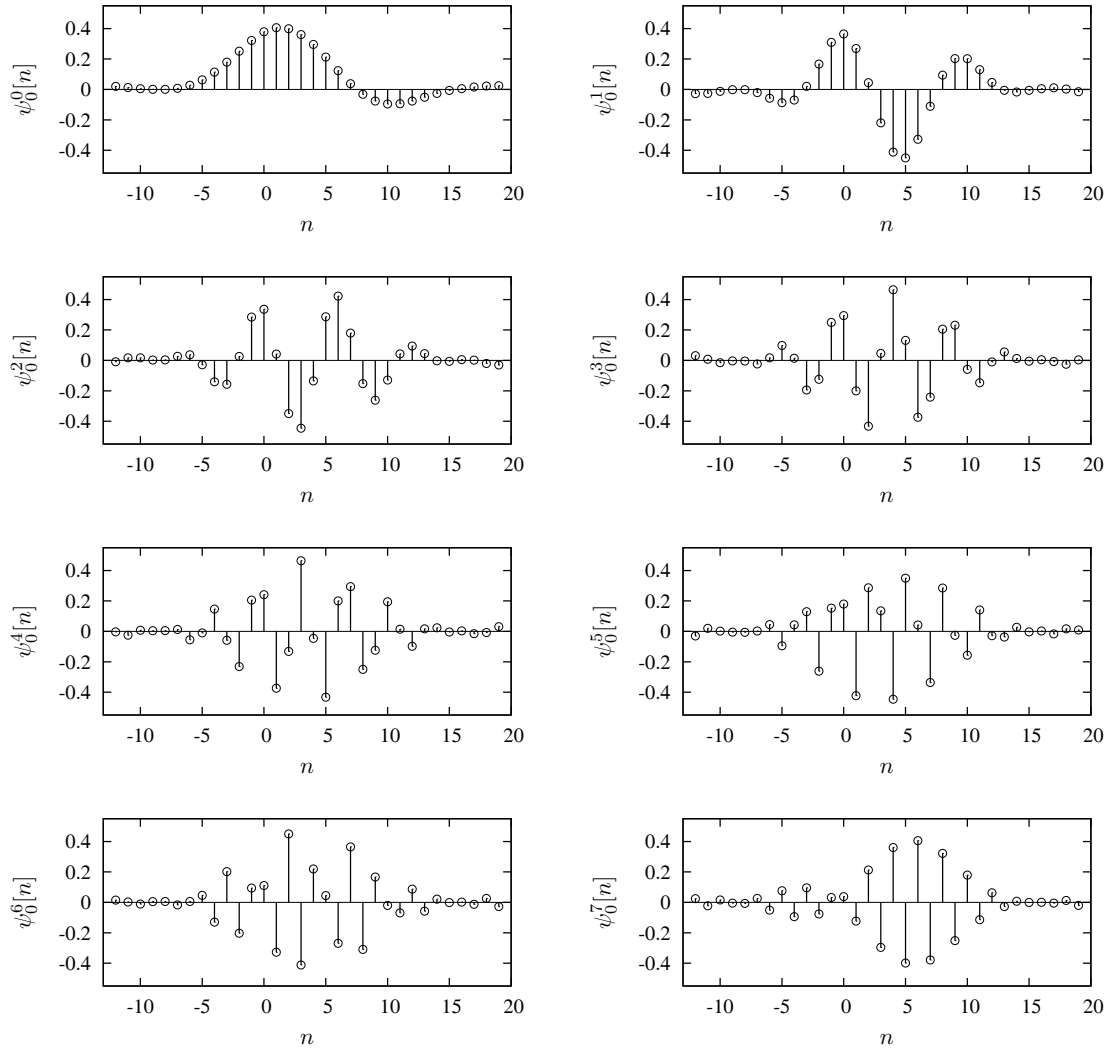


Abbildung G.9: Vierfach überlappende Basisfunktionen der ELT zur Signalrekonstruktion und Signalanalyse, $\gamma = 0.3$, $B = 8$.

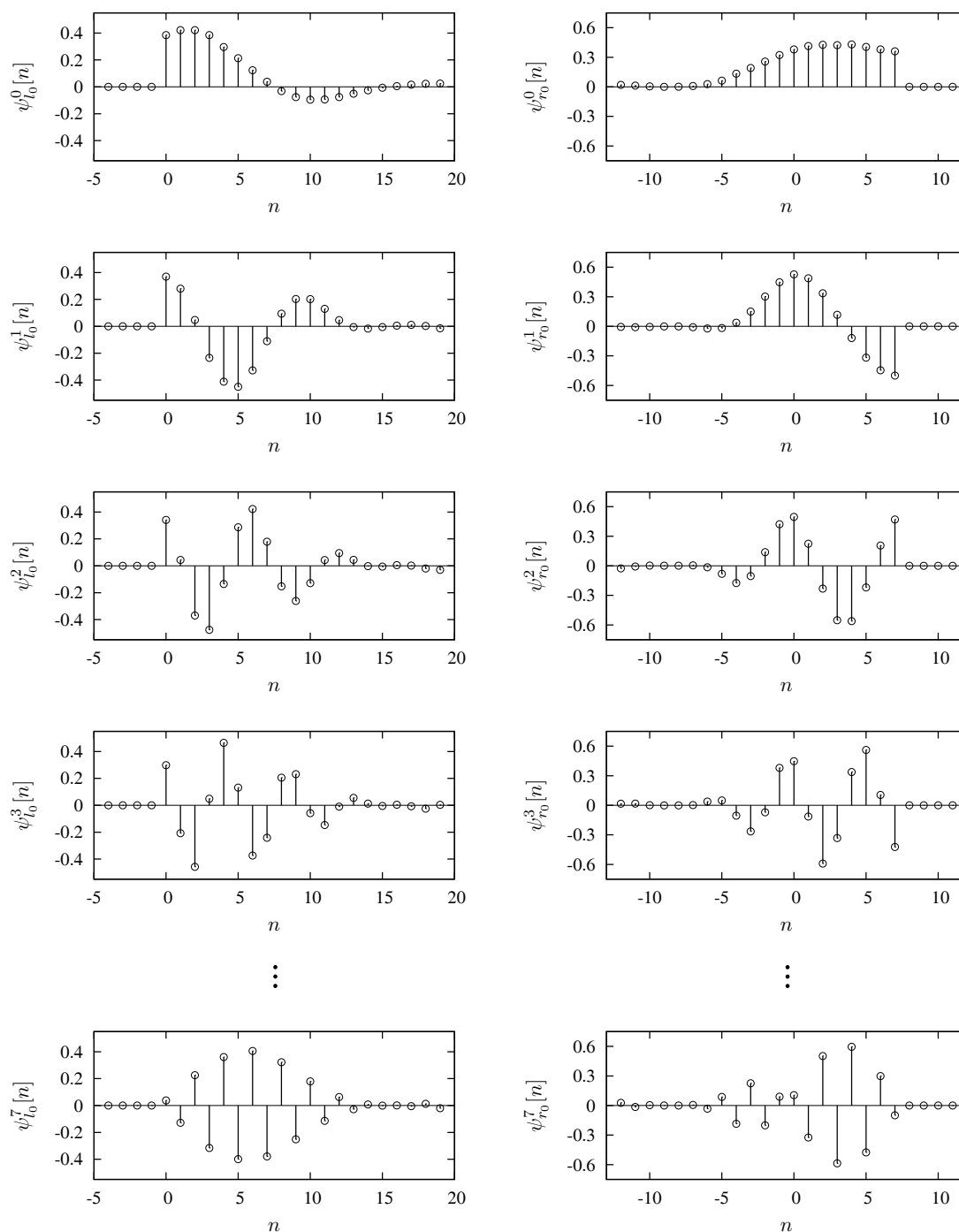


Abbildung G.10: Randbasisfunktionen ψ_{l_0} zur Signalrekonstruktion am linken Signalrand und ψ_{r_0} zur Signalrekonstruktion am rechten Signalrand.

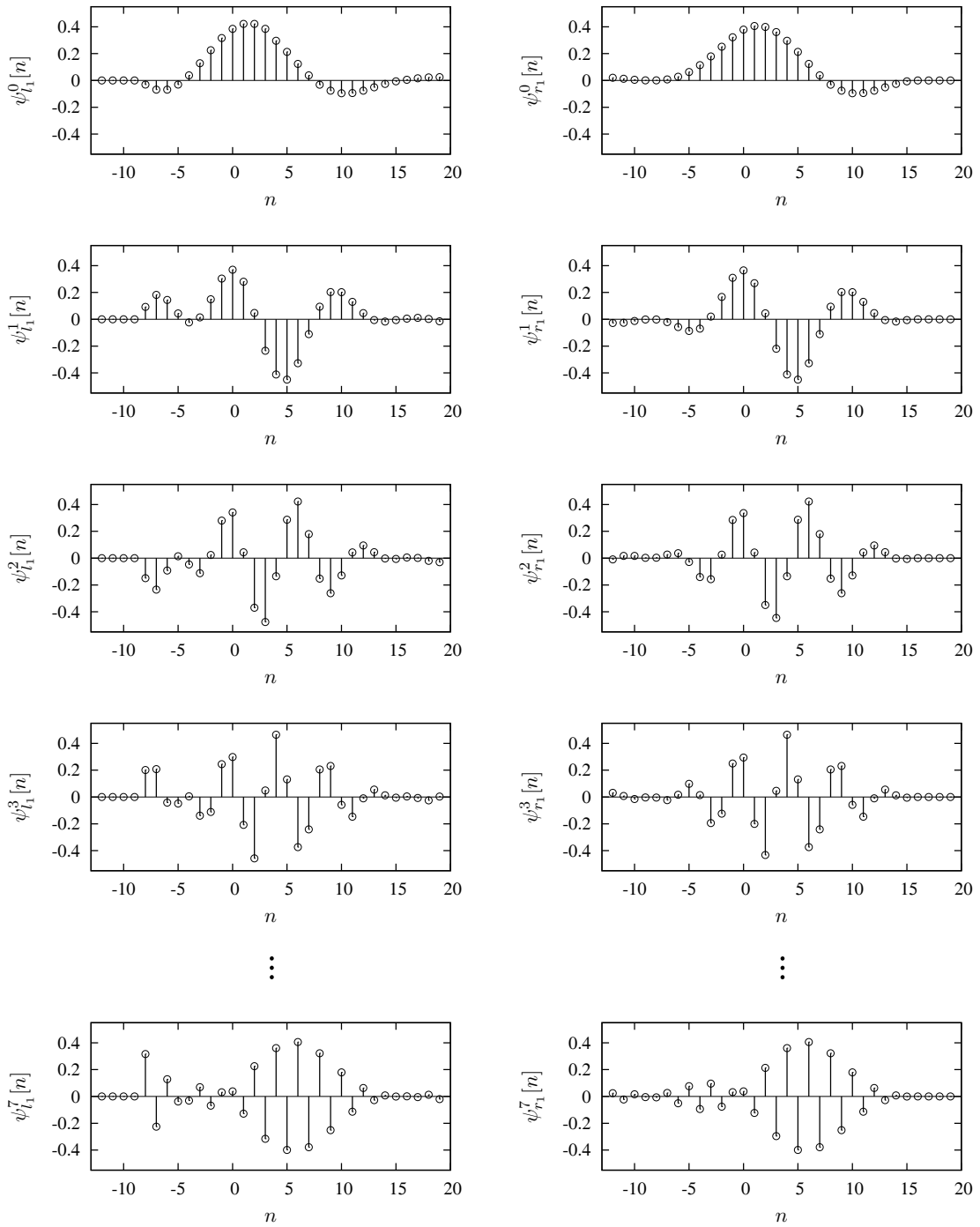


Abbildung G.11: Randbasisfunktionen ψ_{l_1} zur Signalrekonstruktion am linken Signalrand und ψ_{r_1} zur Signalrekonstruktion am rechten Signalrand.

Anhang H

Testbilder und Kompressionsergebnisse

Für die Untersuchung der Leistungsfähigkeit der entwickelten Kompressionsstrategien wurden die Testbilder den folgenden Datenbanken entnommen:

1. „Boat“, „Barbara“, „Goldhill“, „Mandrill“, „Peppers“, „Zelda“
Testbilddatenbank von Chengjie Tu
(<http://thanglong.ece.jhu.edu/~cjtu/link.html>)
2. „Wood Grain“, „Roof“, „Clown“, „Fingerprint“, „Texture“
Testbilddatenbank von Francois G. Meyer
(<http://ece-www.colorado.edu/~fmeyer/Software/lct.html>)
3. „F16“, „Houses“, „Lighthouse“
Testbilddatenbank von Peter Meewald
(<http://www.ganesh.org/~pmeerw/images/>)
4. „Bark“, „Bubble“, „Sand“, „Water“
Testbilddatenbank der University of Southern California (USC) / Institute for Robotics and Intelligent Systems School of Engineering (IRIS)
(<http://iris.usc.edu/Image-database/Industrial.html>)
5. „Lena“
Testbilddatenbank von Mow-Song
(<http://pesona.mmu.edu.my/~msng/EZW.html>)

H.1 Testbilder, PSNR-Diagramme und PSNR-Tabellen

Testbild „Barbara“



Abbildung H.1: Testbild „Barbara“ (512×512 , $H = 7.6321$, $\sigma^2 = 2982.0063$, $\bar{g} = 117.3928$, $g_{min} = 12$, $g_{max} = 246$).

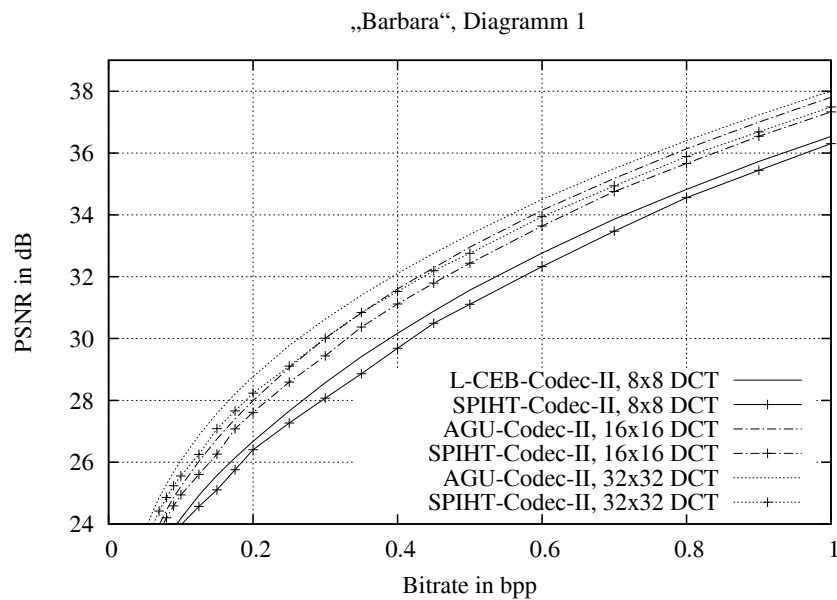


Abbildung H.2: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

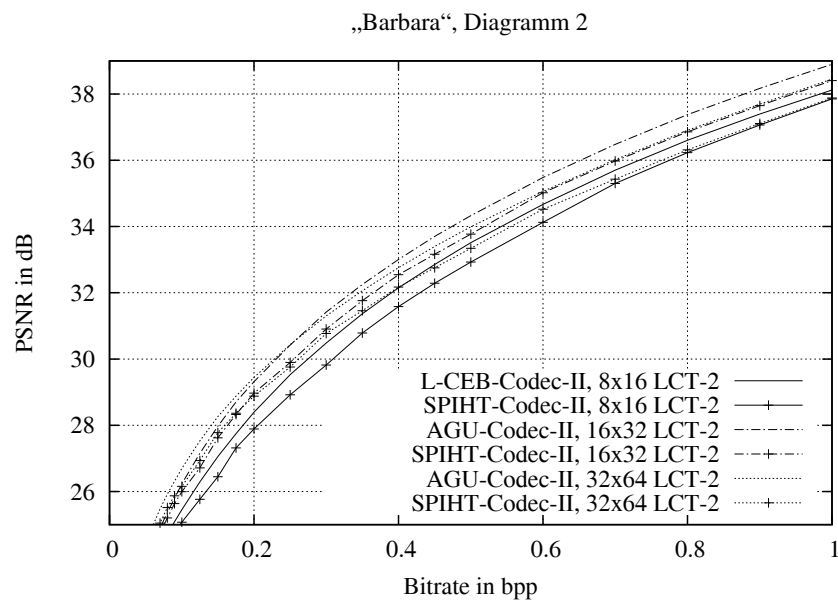


Abbildung H.3: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

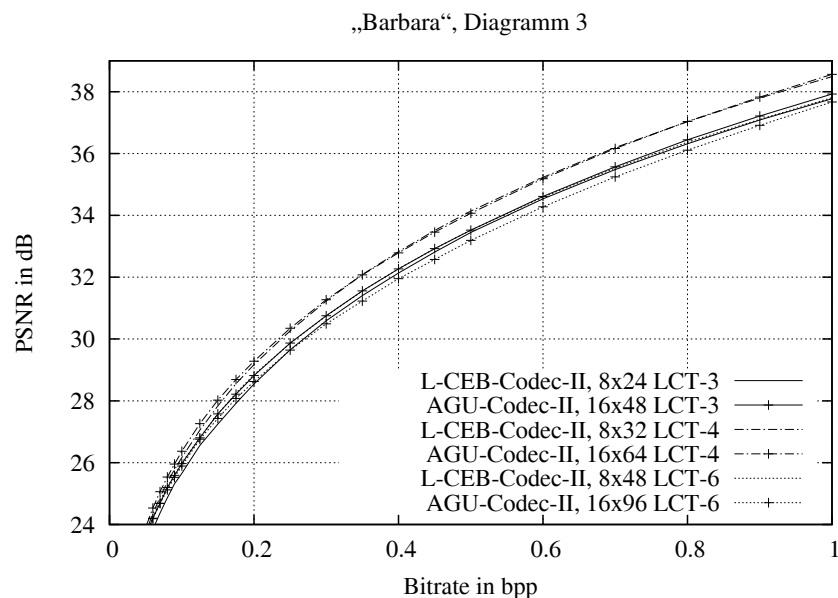


Abbildung H.4: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

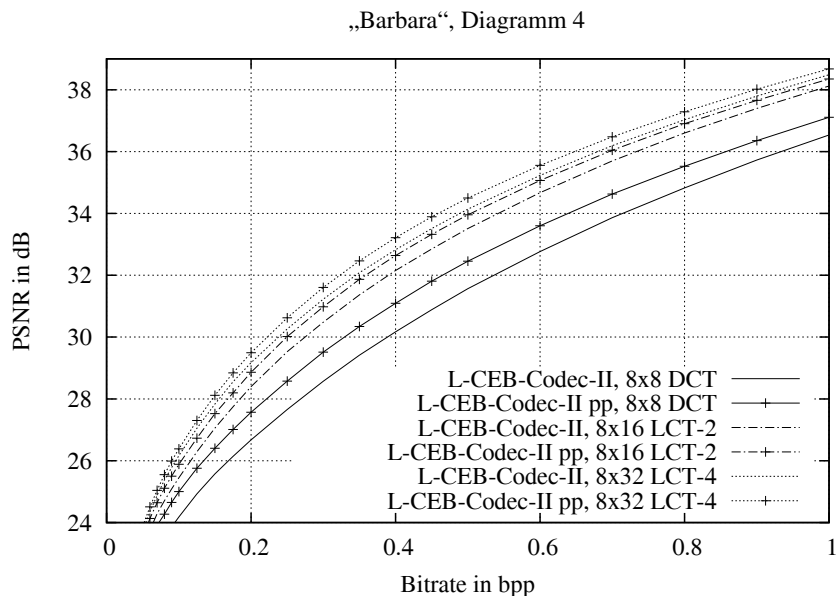


Abbildung H.5: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

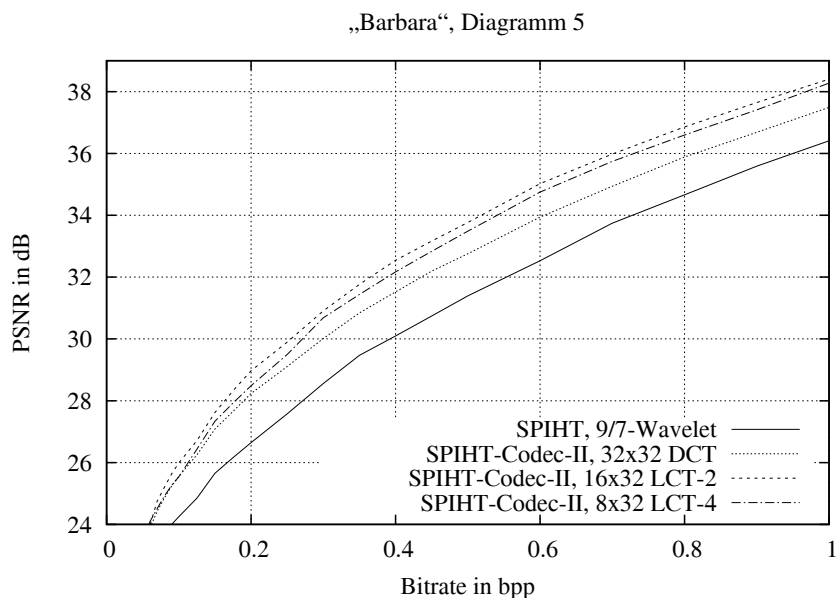


Abbildung H.6: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

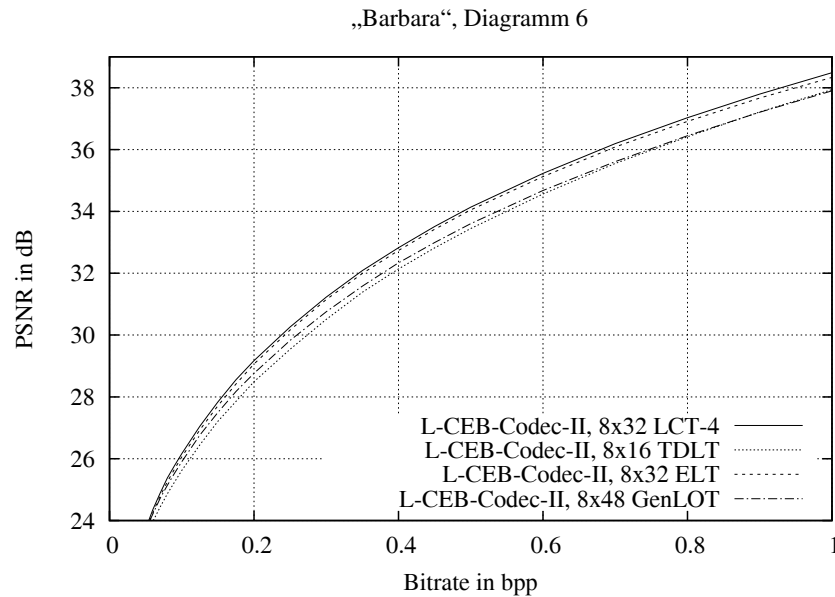


Abbildung H.7: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

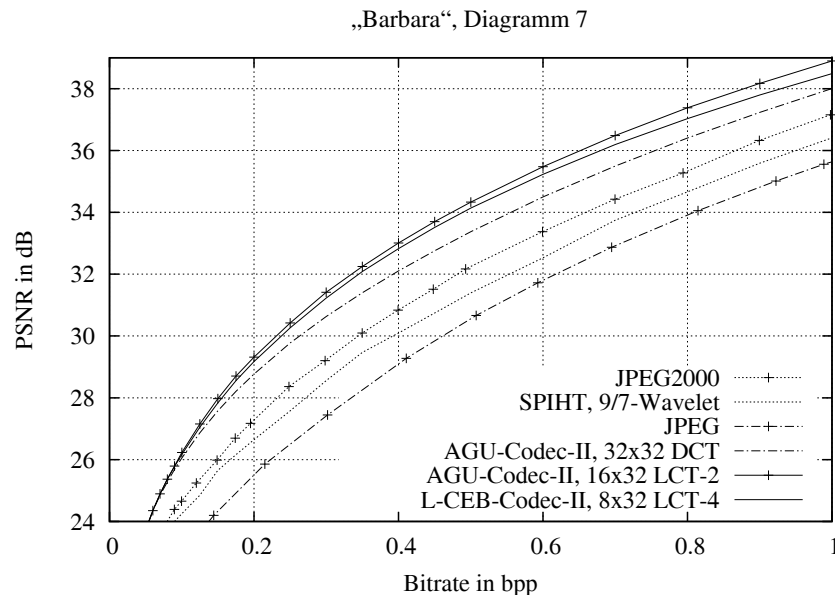


Abbildung H.8: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	22.87	24.93	27.66	31.57	34.36	36.54
SPIHT-Codec-II, 8×8 DCT	22.66	24.57	27.27	31.11	34.00	36.31
AGU-Codec-II, 16×16 DCT	23.60	26.03	29.06	32.96	35.67	37.80
SPIHT-Codec-II, 16×16 DCT	23.21	25.60	28.59	32.43	35.20	37.33
AGU-Codec-II, 32×32 DCT	24.49	26.87	29.77	33.37	35.96	38.01
SPIHT-Codec-II, 32×32 DCT	24.08	26.25	29.11	32.76	35.39	37.49
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	23.90	26.29	29.53	33.51	36.17	38.12
SPIHT-Codec-II, 8×16 LCT-2	23.44	25.77	28.92	32.93	35.77	37.85
AGU-Codec-II, 16×32 LCT-2	24.51	27.15	30.42	34.33	36.95	38.90
SPIHT-Codec-II, 16×32 LCT-2	24.15	26.71	29.90	33.77	36.47	38.40
AGU-Codec-II, 32×64 LCT-2	25.09	27.53	30.44	33.99	36.47	38.45
SPIHT-Codec-II, 32×64 LCT-2	24.70	26.93	29.76	33.34	35.92	37.88
L-CEB-Codec-II, 8×16 TDLT	24.10	26.45	29.54	33.45	36.00	37.94
SPIHT-Codec-II, 8×16 TDLT	23.59	26.01	29.03	32.96	35.68	37.73
L-CEB-Codec-II, 8×16 LOT Typ II	23.81	26.21	29.32	33.21	35.72	37.65
SPIHT-Codec-II, 8×16 LOT Typ II	23.35	25.69	28.79	32.70	35.40	37.42
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	23.99	26.55	29.65	33.44	35.91	37.77
SPIHT-Codec-II, 8×24 LCT-3	23.71	26.11	29.10	32.92	35.42	37.59
AGU-Codec-II, 16×48 LCT-3	24.30	26.81	29.87	33.52	36.02	37.93
SPIHT-Codec-II, 16×48 LCT-3	24.16	26.61	29.53	33.10	35.48	37.61
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	24.50	27.05	30.26	34.14	36.63	38.49
SPIHT-Codec-II, 8×32 LCT-4	24.18	26.43	29.50	33.49	36.16	38.27
AGU-Codec-II, 16×64 LCT-4	24.66	27.26	30.35	34.07	36.60	38.56
SPIHT-Codec-II, 16×64 LCT-4	24.61	27.05	29.91	33.60	36.14	38.15
L-CEB-Codec-II, 8×32 GenLOT	24.06	26.39	29.51	33.45	36.07	38.05
SPIHT-Codec-II, 8×32 GenLOT	23.63	25.84	28.90	32.87	35.69	37.76
L-CEB-Codec-II, 8×32 ELT	24.41	26.93	30.17	34.05	36.51	38.34
SPIHT-Codec-II, 8×32 ELT	23.88	26.30	29.48	33.41	36.10	38.04
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	24.35	26.81	29.87	33.52	35.96	37.81
SPIHT-Codec-II, 8×48 LCT-6	24.01	26.42	29.39	33.07	35.48	37.57
AGU-Codec-II, 16×96 LCT-6	24.30	26.76	29.64	33.19	35.68	37.66
SPIHT-Codec-II, 16×96 LCT-6	24.21	26.59	29.34	32.81	35.20	37.27
L-CEB-Codec-II, 8×48 GenLOT	24.35	26.76	29.83	33.60	36.05	37.90
SPIHT-Codec-II, 8×48 GenLOT	23.95	26.24	29.25	33.08	35.71	37.67
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	23.21	24.89	27.61	31.27	33.96	36.04
AGU-Codec-II, 9/7-Wavelet	23.25	25.15	27.80	31.54	34.26	36.44
SPIHT, 9/7-Wavelet	23.35	24.85	27.58	31.39	34.25	36.41
JPEG2000	23.09	25.24	28.36	32.17	34.88	37.15

Tabelle H.1: Kompressionsergebnisse für das Testbild „Barbara“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	23.60	25.75	28.58	32.45	35.09	37.11
SPIHT-Codec-II pp, 8×8 DCT	23.20	25.17	27.93	31.74	34.68	36.71
AGU-Codec-II pp, 16×16 DCT	24.06	26.51	29.61	33.51	36.13	38.13
SPIHT-Codec-II pp, 16×16 DCT	23.61	25.96	28.99	32.85	35.48	37.59
AGU-Codec-II pp, 32×32 DCT	24.75	27.23	30.26	33.94	36.43	38.38
SPIHT-Codec-II pp, 32×32 DCT	24.22	26.45	29.42	33.15	35.68	37.78
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	24.27	26.73	30.02	33.96	36.48	38.35
SPIHT-Codec-II pp, 8×16 LCT-2	23.65	26.05	29.23	33.22	35.95	37.99
AGU-Codec-II pp, 16×32 LCT-2	24.72	27.40	30.78	34.67	37.21	39.07
SPIHT-Codec-II pp, 16×32 LCT-2	24.25	26.84	30.12	34.01	36.62	38.57
AGU-Codec-II pp, 32×64 LCT-2	25.25	27.83	30.90	34.51	36.91	38.79
SPIHT-Codec-II pp, 32×64 LCT-2	24.90	27.09	30.06	33.72	36.17	38.18
L-CEB-Codec-II pp, 8×16 TDLT	24.23	26.73	29.91	33.81	36.29	38.14
SPIHT-Codec-II pp, 8×16 TDLT	23.67	26.18	29.24	33.15	35.91	37.82
L-CEB-Codec-II pp, 8×16 LOT Typ II	24.23	26.71	29.92	33.86	36.28	38.09
SPIHT-Codec-II pp, 8×16 LOT Typ II	23.65	26.04	29.21	33.18	35.75	37.74
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	24.31	26.98	30.19	33.99	36.36	38.14
SPIHT-Codec-II pp, 8×24 LCT-3	23.87	26.38	29.44	33.24	35.75	37.77
AGU-Codec-II pp, 16×48 LCT-3	24.57	27.21	30.43	34.16	36.58	38.40
SPIHT-Codec-II pp, 16×48 LCT-3	24.36	26.91	29.92	33.49	35.94	37.87
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	24.66	27.30	30.63	34.50	36.91	38.67
SPIHT-Codec-II pp, 8×32 LCT-4	24.34	26.53	29.69	33.73	36.32	38.31
AGU-Codec-II pp, 16×64 LCT-4	24.83	27.54	30.79	34.56	37.02	38.87
SPIHT-Codec-II pp, 16×64 LCT-4	24.77	27.14	30.20	34.01	36.42	38.27
L-CEB-Codec-II pp, 8×32 GenLOT	24.36	26.81	30.00	33.92	36.42	38.30
SPIHT-Codec-II pp, 8×32 GenLOT	23.79	26.10	29.23	33.20	35.91	37.92
L-CEB-Codec-II pp, 8×32 ELT	24.58	27.21	30.55	34.46	36.86	38.60
SPIHT-Codec-II pp, 8×32 ELT	23.92	26.44	29.71	33.68	36.31	38.22
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	24.60	27.19	30.42	34.13	36.47	38.23
SPIHT-Codec-II pp, 8×48 LCT-6	24.16	26.68	29.73	33.40	35.87	37.78
AGU-Codec-II pp, 16×96 LCT-6	24.59	27.24	30.34	33.97	36.37	38.24
SPIHT-Codec-II pp, 16×96 LCT-6	24.45	26.96	29.83	33.28	35.78	37.60
L-CEB-Codec-II pp, 8×48 GenLOT	24.55	27.08	30.31	34.12	36.48	38.27
SPIHT-Codec-II pp, 8×48 GenLOT	23.99	26.41	29.55	33.44	35.98	37.93
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	23.38	25.23	28.14	31.88	34.50	36.49
AGU-Codec-II pp, 9/7-Wavelet	23.34	25.47	28.24	32.09	34.72	36.83
SPIHT pp, 9/7-Wavelet	23.43	25.01	27.85	31.69	34.45	36.61

Tabelle H.2: Kompressionsergebnisse für das Testbild „Barbara“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Lena“



Abbildung H.9: Testbild „Lena“ (512×512 , $H = 7.4455$, $\sigma^2 = 2289.9848$, $\bar{g} = 124.0505$, $g_{min} = 25$, $g_{max} = 245$).

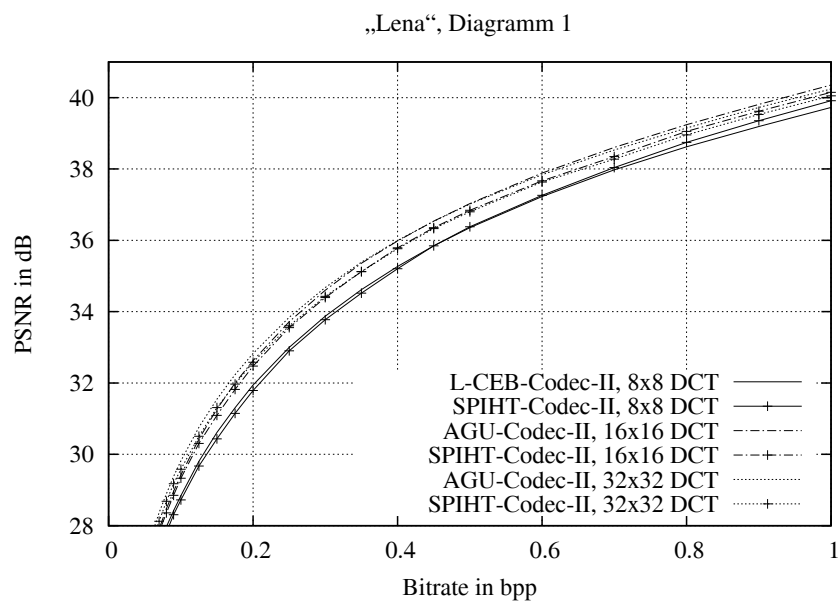


Abbildung H.10: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

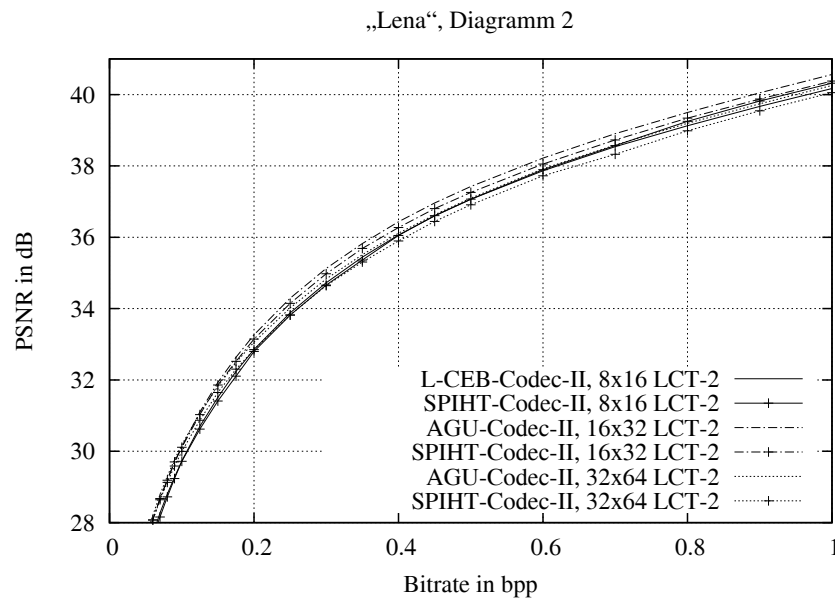


Abbildung H.11: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

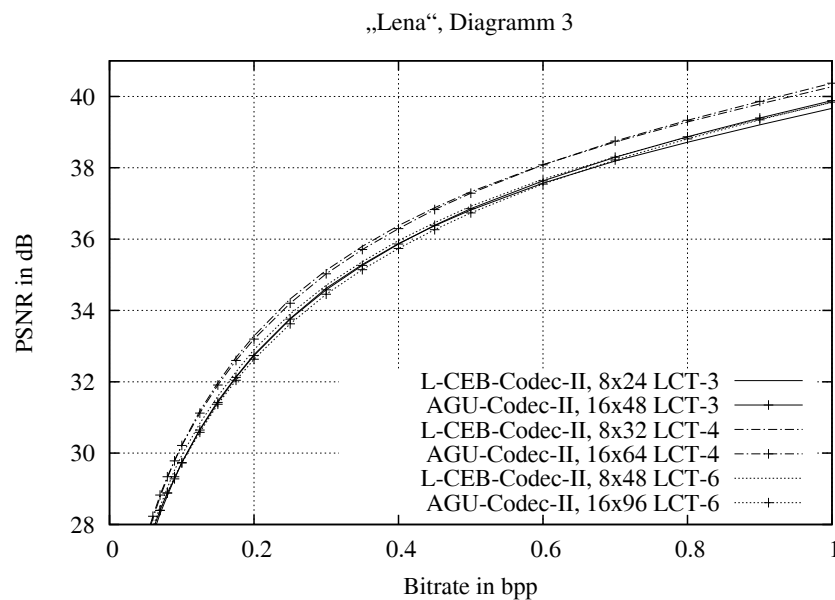


Abbildung H.12: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

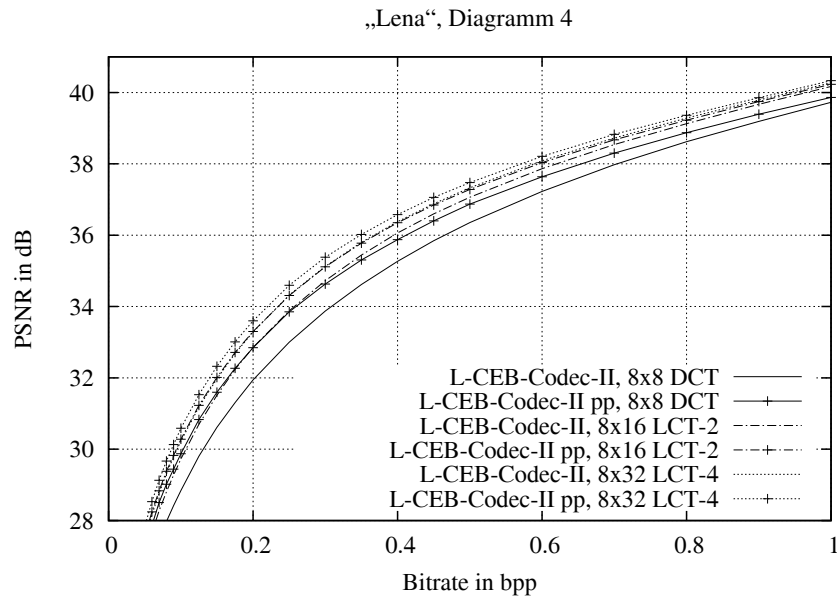


Abbildung H.13: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

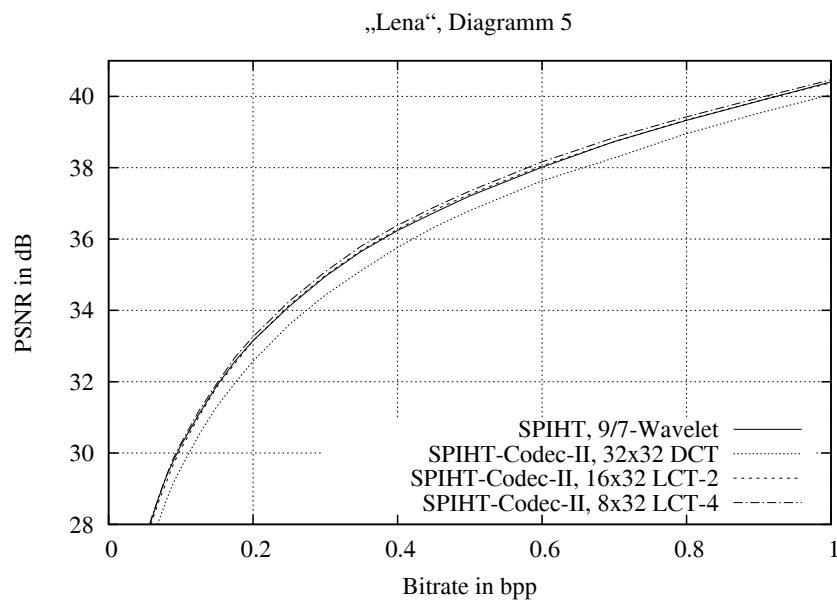


Abbildung H.14: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

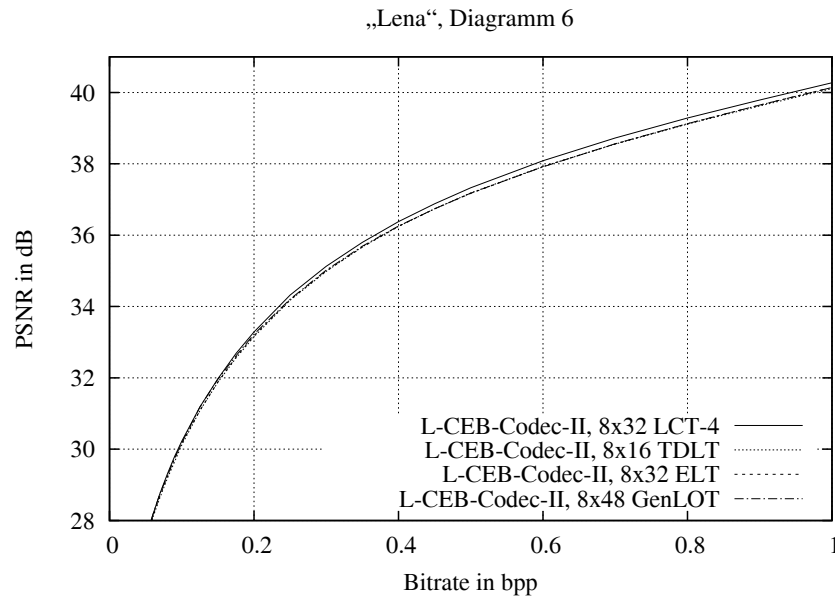


Abbildung H.15: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

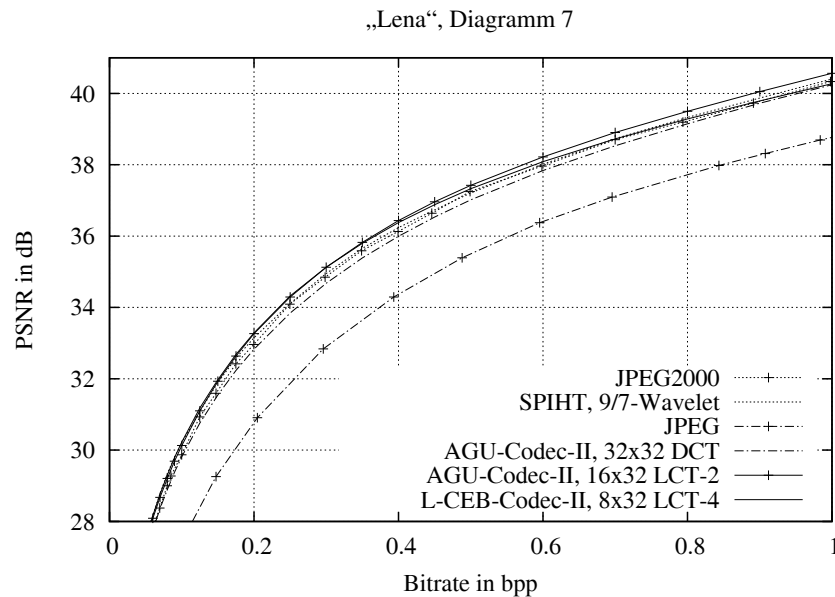


Abbildung H.16: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	27.04	29.81	33.00	36.35	38.30	39.73
SPIHT-Codec-II, 8×8 DCT	26.91	29.67	32.90	36.38	38.36	39.91
AGU-Codec-II, 16×16 DCT	27.46	30.43	33.72	37.03	38.93	40.36
SPIHT-Codec-II, 16×16 DCT	27.38	30.31	33.55	36.84	38.75	40.15
AGU-Codec-II, 32×32 DCT	27.92	30.76	33.84	37.02	38.85	40.24
SPIHT-Codec-II, 32×32 DCT	27.71	30.51	33.60	36.80	38.65	40.05
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	27.84	30.70	33.88	37.06	38.84	40.17
SPIHT-Codec-II, 8×16 LCT-2	27.74	30.62	33.82	37.07	38.96	40.32
AGU-Codec-II, 16×32 LCT-2	28.26	31.10	34.29	37.43	39.20	40.56
SPIHT-Codec-II, 16×32 LCT-2	28.22	31.03	34.15	37.26	39.05	40.38
AGU-Codec-II, 32×64 LCT-2	28.35	31.07	34.02	37.10	38.89	40.25
SPIHT-Codec-II, 32×64 LCT-2	28.23	30.87	33.82	36.91	38.69	40.06
L-CEB-Codec-II, 8×16 TDLT	28.23	31.08	34.18	37.18	38.83	40.11
SPIHT-Codec-II, 8×16 TDLT	28.26	31.10	34.14	37.21	38.98	40.28
L-CEB-Codec-II, 8×16 LOT Typ II	27.71	30.56	33.60	36.68	38.52	39.88
SPIHT-Codec-II, 8×16 LOT Typ II	27.61	30.48	33.56	36.73	38.63	40.05
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	27.88	30.67	33.78	36.81	38.46	39.66
SPIHT-Codec-II, 8×24 LCT-3	28.06	30.76	33.79	36.89	38.67	39.72
AGU-Codec-II, 16×48 LCT-3	27.98	30.65	33.76	36.85	38.60	39.88
SPIHT-Codec-II, 16×48 LCT-3	28.10	30.66	33.67	36.69	38.54	39.61
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	28.33	31.18	34.32	37.33	39.01	40.27
SPIHT-Codec-II, 8×32 LCT-4	28.39	31.18	34.26	37.35	39.14	40.46
AGU-Codec-II, 16×64 LCT-4	28.39	31.12	34.20	37.29	39.04	40.37
SPIHT-Codec-II, 16×64 LCT-4	28.44	31.09	34.08	37.15	38.93	40.26
L-CEB-Codec-II, 8×32 GenLOT	27.97	30.85	34.01	37.11	38.87	40.19
SPIHT-Codec-II, 8×32 GenLOT	27.96	30.78	33.94	37.12	39.00	40.35
L-CEB-Codec-II, 8×32 ELT	28.26	31.09	34.19	37.17	38.85	40.14
SPIHT-Codec-II, 8×32 ELT	28.28	31.09	34.16	37.20	38.99	40.30
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	28.07	30.84	33.90	36.93	38.59	39.83
SPIHT-Codec-II, 8×48 LCT-6	28.23	30.95	33.95	37.01	38.76	39.98
AGU-Codec-II, 16×96 LCT-6	27.95	30.58	33.63	36.74	38.53	39.84
SPIHT-Codec-II, 16×96 LCT-6	28.10	30.63	33.55	36.59	38.43	39.62
L-CEB-Codec-II, 8×48 GenLOT	28.32	31.18	34.20	37.18	38.85	40.14
SPIHT-Codec-II, 8×48 GenLOT	28.34	31.13	34.19	37.21	38.99	40.31
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	28.26	30.97	33.92	36.99	38.76	40.07
AGU-Codec-II, 9/7-Wavelet	28.15	30.89	33.91	37.06	38.95	40.34
SPIHT, 9/7-Wavelet	28.36	31.09	34.11	37.21	39.04	40.41
JPEG2000	27.91	30.93	34.09	37.25	38.97	40.33

Tabelle H.3: Kompressionsergebnisse für das Testbild „Lena“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	28.02	30.83	33.85	36.87	38.59	39.86
SPIHT-Codec-II pp, 8×8 DCT	27.76	30.49	33.56	36.72	38.56	39.93
AGU-Codec-II pp, 16×16 DCT	28.22	31.10	34.23	37.30	39.07	40.40
SPIHT-Codec-II pp, 16×16 DCT	27.97	30.84	33.93	37.01	38.86	40.12
AGU-Codec-II pp, 32×32 DCT	28.44	31.28	34.29	37.30	39.02	40.34
SPIHT-Codec-II pp, 32×32 DCT	28.08	30.88	33.91	36.96	38.78	40.06
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	28.39	31.23	34.31	37.28	38.96	40.22
SPIHT-Codec-II pp, 8×16 LCT-2	28.12	30.97	34.08	37.17	38.95	40.24
AGU-Codec-II pp, 16×32 LCT-2	28.65	31.51	34.60	37.59	39.28	40.58
SPIHT-Codec-II pp, 16×32 LCT-2	28.49	31.26	34.32	37.33	39.04	40.31
AGU-Codec-II pp, 32×64 LCT-2	28.73	31.47	34.43	37.36	39.07	40.36
SPIHT-Codec-II pp, 32×64 LCT-2	28.46	31.14	34.05	37.04	38.79	40.06
L-CEB-Codec-II pp, 8×16 TDLT	28.48	31.36	34.43	37.33	38.90	40.12
SPIHT-Codec-II pp, 8×16 TDLT	28.41	31.26	34.27	37.25	38.96	40.18
L-CEB-Codec-II pp, 8×16 LOT Typ II	28.35	31.24	34.26	37.12	38.78	40.04
SPIHT-Codec-II pp, 8×16 LOT Typ II	28.10	31.01	34.05	37.01	38.78	40.07
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	28.38	31.18	34.23	37.13	38.72	39.90
SPIHT-Codec-II pp, 8×24 LCT-3	28.37	31.06	34.10	37.10	38.75	39.83
AGU-Codec-II pp, 16×48 LCT-3	28.46	31.20	34.26	37.21	38.87	40.10
SPIHT-Codec-II pp, 16×48 LCT-3	28.45	31.00	34.07	36.98	38.66	39.75
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	28.71	31.53	34.60	37.47	39.10	40.33
SPIHT-Codec-II pp, 8×32 LCT-4	28.59	31.39	34.42	37.41	39.10	40.39
AGU-Codec-II pp, 16×64 LCT-4	28.73	31.52	34.54	37.51	39.19	40.45
SPIHT-Codec-II pp, 16×64 LCT-4	28.65	31.35	34.33	37.29	38.96	40.25
L-CEB-Codec-II pp, 8×32 GenLOT	28.50	31.36	34.41	37.34	38.99	40.24
SPIHT-Codec-II pp, 8×32 GenLOT	28.33	31.15	34.22	37.24	39.00	40.28
L-CEB-Codec-II pp, 8×32 ELT	28.67	31.49	34.56	37.43	39.02	40.25
SPIHT-Codec-II pp, 8×32 ELT	28.50	31.33	34.38	37.34	39.02	40.26
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	28.54	31.34	34.35	37.25	38.84	40.05
SPIHT-Codec-II pp, 8×48 LCT-6	28.55	31.26	34.26	37.14	38.85	40.09
AGU-Codec-II pp, 16×96 LCT-6	28.46	31.20	34.20	37.17	38.85	40.10
SPIHT-Codec-II pp, 16×96 LCT-6	28.50	31.02	34.03	36.96	38.62	39.80
L-CEB-Codec-II pp, 8×48 GenLOT	28.70	31.61	34.59	37.46	39.03	40.25
SPIHT-Codec-II pp, 8×48 GenLOT	28.58	31.38	34.44	37.36	39.02	40.28
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	28.48	31.23	34.18	37.13	38.82	40.05
AGU-Codec-II pp, 9/7-Wavelet	28.32	31.10	34.13	37.18	38.97	40.28
SPIHT pp, 9/7-Wavelet	28.48	31.24	34.24	37.24	38.98	40.27

Tabelle H.4: Kompressionsergebnisse für das Testbild „Lena“, Vergleich der Codecs mit Post-processing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Zelda“



Abbildung H.17: Testbild „Zelda“ (512×512 , $H = 7.2668$, $\sigma^2 = 1642.3238$, $\bar{g} = 91.1696$, $g_{min} = 0$, $g_{max} = 187$).

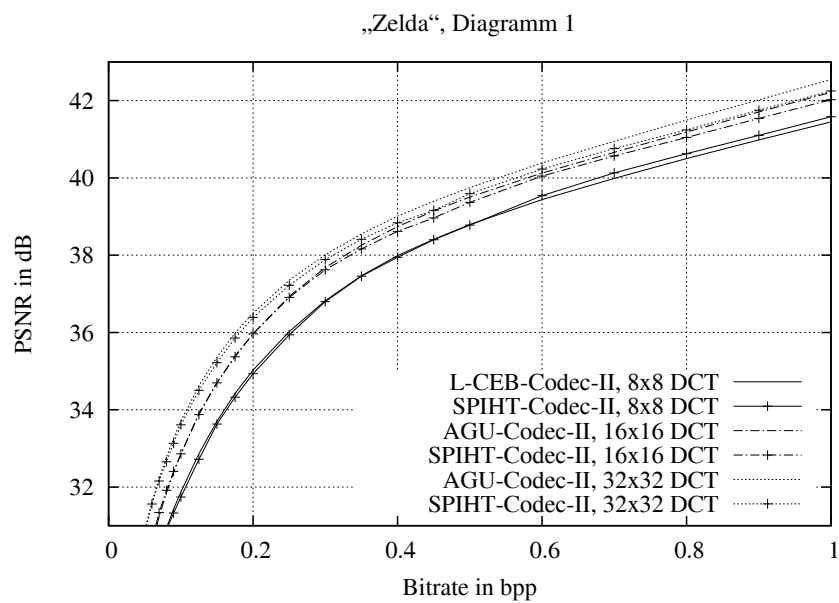


Abbildung H.18: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

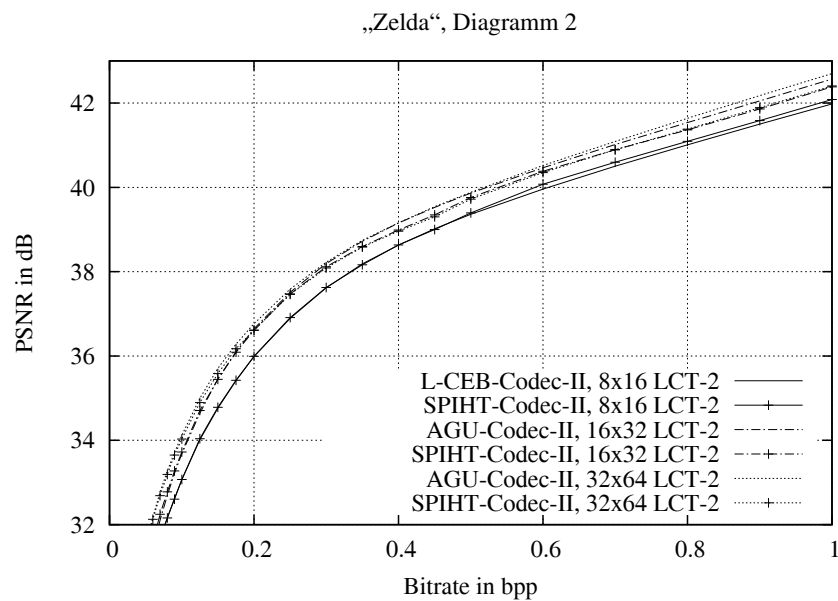


Abbildung H.19: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

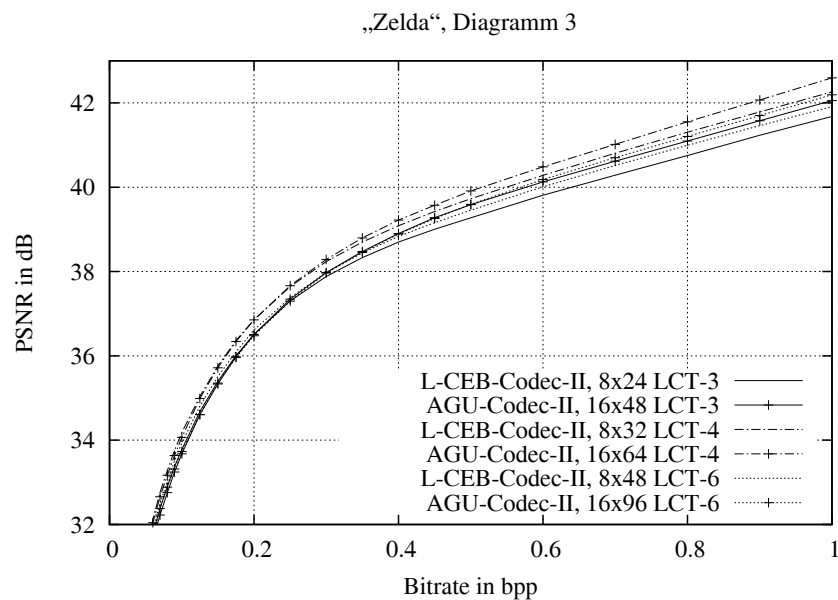


Abbildung H.20: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

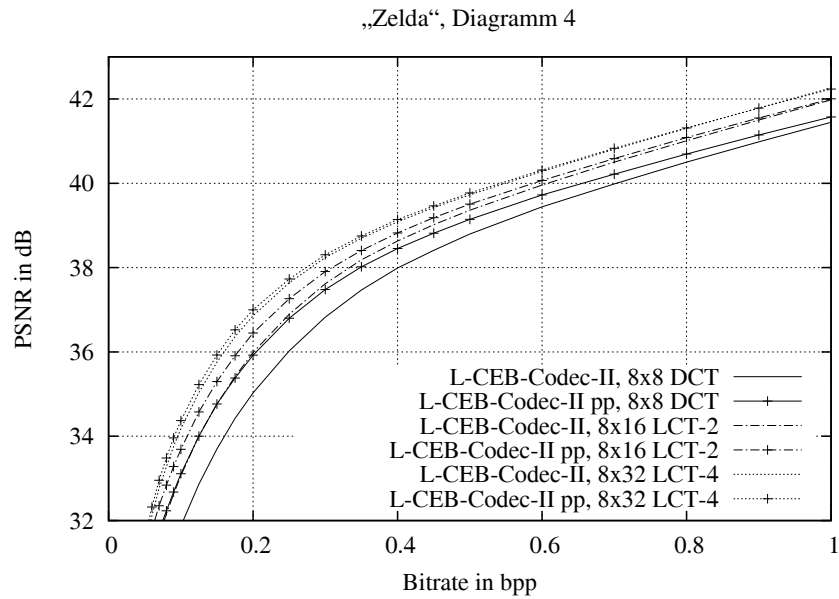


Abbildung H.21: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

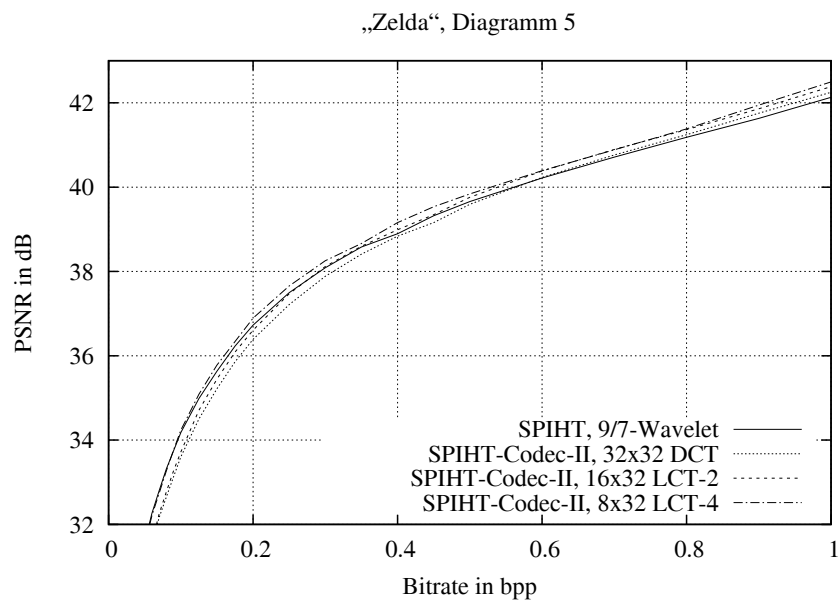


Abbildung H.22: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

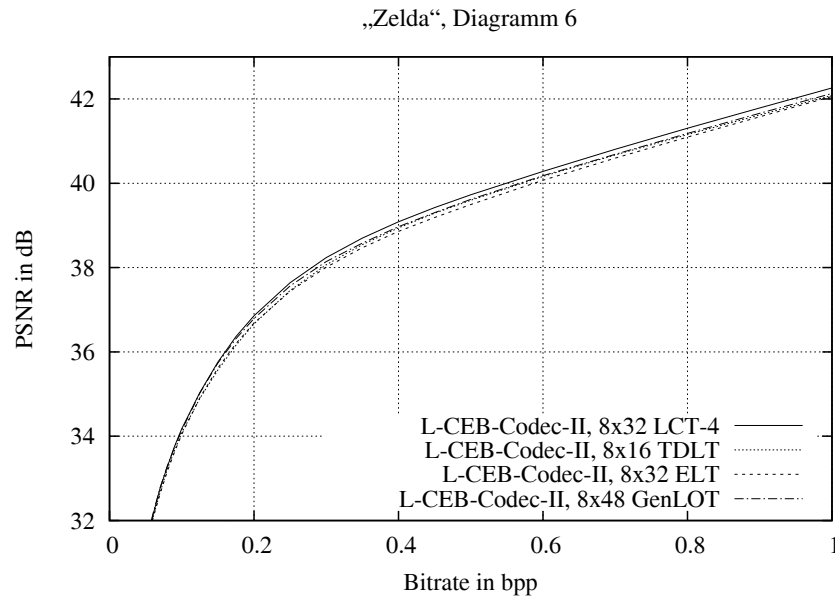


Abbildung H.23: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

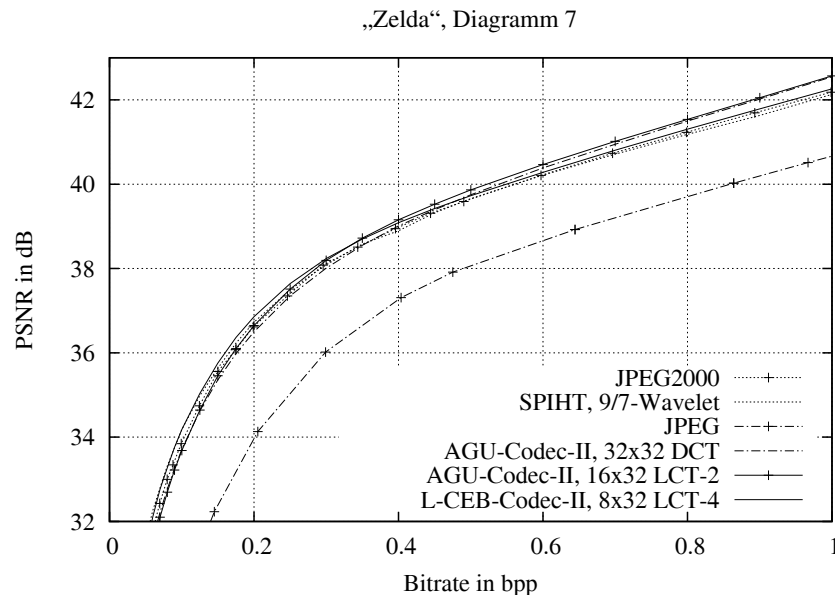


Abbildung H.24: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	30.01	32.86	36.02	38.79	40.25	41.45
SPIHT-Codec-II, 8×8 DCT	29.95	32.72	35.94	38.78	40.38	41.58
AGU-Codec-II, 16×16 DCT	30.71	33.90	36.94	39.50	40.93	42.21
SPIHT-Codec-II, 16×16 DCT	30.80	33.87	36.91	39.37	40.81	42.03
AGU-Codec-II, 32×32 DCT	31.77	34.62	37.35	39.75	41.22	42.55
SPIHT-Codec-II, 32×32 DCT	31.71	34.50	37.22	39.59	41.00	42.25
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	31.26	34.01	36.89	39.36	40.75	41.98
SPIHT-Codec-II, 8×16 LCT-2	31.29	34.04	36.91	39.39	40.85	42.08
AGU-Codec-II, 16×32 LCT-2	31.61	34.64	37.51	39.87	41.28	42.57
SPIHT-Codec-II, 16×32 LCT-2	31.74	34.71	37.47	39.76	41.13	42.39
AGU-Codec-II, 32×64 LCT-2	32.34	35.00	37.59	39.89	41.36	42.70
SPIHT-Codec-II, 32×64 LCT-2	32.30	34.89	37.46	39.72	41.14	42.41
L-CEB-Codec-II, 8×16 TDLT	32.20	34.90	37.46	39.60	40.91	42.09
SPIHT-Codec-II, 8×16 TDLT	32.14	34.84	37.44	39.64	41.00	42.10
L-CEB-Codec-II, 8×16 LOT Typ II	30.91	33.68	36.60	39.04	40.47	41.68
SPIHT-Codec-II, 8×16 LOT Typ II	30.95	33.67	36.62	39.06	40.57	41.80
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	31.95	34.68	37.30	39.27	40.52	41.68
SPIHT-Codec-II, 8×24 LCT-3	31.87	34.59	37.25	39.47	40.71	41.94
AGU-Codec-II, 16×48 LCT-3	31.80	34.61	37.35	39.59	40.86	42.06
SPIHT-Codec-II, 16×48 LCT-3	31.88	34.62	37.27	39.50	40.69	41.97
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	32.28	35.04	37.64	39.73	41.05	42.26
SPIHT-Codec-II, 8×32 LCT-4	32.34	35.09	37.66	39.84	41.12	42.50
AGU-Codec-II, 16×64 LCT-4	32.21	34.99	37.66	39.91	41.28	42.59
SPIHT-Codec-II, 16×64 LCT-4	32.35	35.01	37.63	39.83	41.14	42.51
L-CEB-Codec-II, 8×32 GenLOT	31.65	34.33	37.14	39.48	40.85	42.06
SPIHT-Codec-II, 8×32 GenLOT	31.74	34.35	37.14	39.55	40.92	42.17
L-CEB-Codec-II, 8×32 ELT	32.20	34.91	37.44	39.50	40.84	42.06
SPIHT-Codec-II, 8×32 ELT	32.28	34.95	37.47	39.61	40.93	42.16
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	32.08	34.80	37.37	39.46	40.76	41.91
SPIHT-Codec-II, 8×48 LCT-6	32.14	34.83	37.39	39.59	40.97	42.08
AGU-Codec-II, 16×96 LCT-6	31.92	34.60	37.30	39.59	40.95	42.19
SPIHT-Codec-II, 16×96 LCT-6	32.08	34.69	37.30	39.53	40.88	42.07
L-CEB-Codec-II, 8×48 GenLOT	32.31	35.02	37.56	39.62	40.93	42.13
SPIHT-Codec-II, 8×48 GenLOT	32.42	35.04	37.60	39.73	41.01	42.26
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	32.26	34.83	37.33	39.41	40.67	41.81
AGU-Codec-II, 9/7-Wavelet	32.21	34.81	37.36	39.61	40.96	42.15
SPIHT, 9/7-Wavelet	32.39	34.98	37.50	39.66	40.95	42.13
JPEG2000	31.78	34.74	37.35	39.59	40.97	42.18

Tabelle H.5: Kompressionsergebnisse für das Testbild „Zelda“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	31.34	34.00	36.80	39.15	40.46	41.57
SPIHT-Codec-II pp, 8×8 DCT	31.12	33.73	36.62	39.05	40.42	41.55
AGU-Codec-II pp, 16×16 DCT	31.49	34.54	37.34	39.62	40.96	42.19
SPIHT-Codec-II pp, 16×16 DCT	31.45	34.40	37.23	39.50	40.79	41.96
AGU-Codec-II pp, 32×32 DCT	32.18	35.00	37.61	39.83	41.24	42.53
SPIHT-Codec-II pp, 32×32 DCT	32.05	34.81	37.38	39.65	40.97	42.13
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	31.94	34.58	37.26	39.51	40.84	42.00
SPIHT-Codec-II pp, 8×16 LCT-2	31.77	34.49	37.21	39.49	40.78	41.97
AGU-Codec-II pp, 16×32 LCT-2	31.89	34.93	37.66	39.87	41.25	42.48
SPIHT-Codec-II pp, 16×32 LCT-2	31.98	34.89	37.52	39.75	41.05	42.24
AGU-Codec-II pp, 32×64 LCT-2	32.52	35.23	37.75	39.94	41.36	42.64
SPIHT-Codec-II pp, 32×64 LCT-2	32.44	35.04	37.53	39.73	41.07	42.25
L-CEB-Codec-II pp, 8×16 TDLT	32.32	35.06	37.56	39.62	40.88	42.00
SPIHT-Codec-II pp, 8×16 TDLT	32.15	34.95	37.48	39.57	40.87	41.89
L-CEB-Codec-II pp, 8×16 LOT Typ II	31.79	34.46	37.20	39.38	40.69	41.84
SPIHT-Codec-II pp, 8×16 LOT Typ II	31.61	34.32	37.15	39.32	40.64	41.80
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	32.27	35.01	37.52	39.47	40.71	41.82
SPIHT-Codec-II pp, 8×24 LCT-3	32.06	34.80	37.37	39.48	40.76	41.88
AGU-Codec-II pp, 16×48 LCT-3	32.05	34.89	37.56	39.70	40.96	42.12
SPIHT-Codec-II pp, 16×48 LCT-3	31.99	34.80	37.39	39.53	40.73	41.92
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	32.48	35.22	37.73	39.77	41.06	42.24
SPIHT-Codec-II pp, 8×32 LCT-4	32.45	35.18	37.67	39.77	41.03	42.31
AGU-Codec-II pp, 16×64 LCT-4	32.34	35.19	37.78	39.93	41.26	42.51
SPIHT-Codec-II pp, 16×64 LCT-4	32.45	35.12	37.66	39.80	41.05	42.36
L-CEB-Codec-II pp, 8×32 GenLOT	32.15	34.85	37.43	39.59	40.92	42.07
SPIHT-Codec-II pp, 8×32 GenLOT	32.17	34.71	37.31	39.59	40.85	42.05
L-CEB-Codec-II pp, 8×32 ELT	32.47	35.22	37.69	39.68	40.97	42.12
SPIHT-Codec-II pp, 8×32 ELT	32.49	35.16	37.61	39.64	40.91	42.10
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	32.35	35.06	37.56	39.61	40.89	42.01
SPIHT-Codec-II pp, 8×48 LCT-6	32.24	34.96	37.48	39.59	40.95	42.00
AGU-Codec-II pp, 16×96 LCT-6	32.20	34.93	37.56	39.73	41.04	42.24
SPIHT-Codec-II pp, 16×96 LCT-6	32.18	34.87	37.45	39.57	40.87	42.02
L-CEB-Codec-II pp, 8×48 GenLOT	32.49	35.27	37.75	39.74	41.03	42.17
SPIHT-Codec-II pp, 8×48 GenLOT	32.57	35.21	37.71	39.75	40.97	42.18
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	32.35	34.97	37.45	39.44	40.67	41.75
AGU-Codec-II pp, 9/7-Wavelet	32.28	34.89	37.44	39.59	40.90	42.06
SPIHT pp, 9/7-Wavelet	32.44	35.04	37.52	39.60	40.80	41.92

Tabelle H.6: Kompressionsergebnisse für das Testbild „Zelda“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Boat“



Abbildung H.25: Testbild „Boat“ (512×512 , $H = 7.1238$, $\sigma^2 = 2737.6465$, $\bar{g} = 136.1268$, $g_{min} = 0$, $g_{max} = 239$).

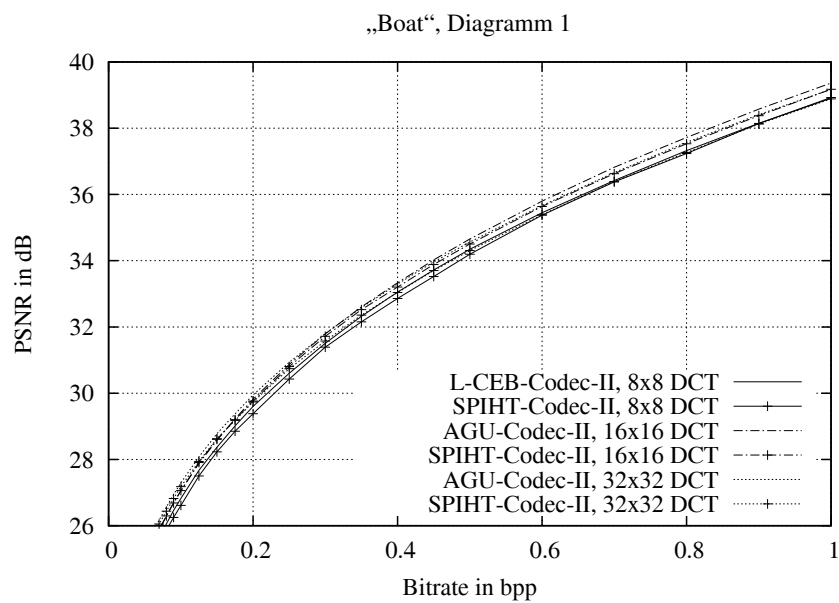


Abbildung H.26: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

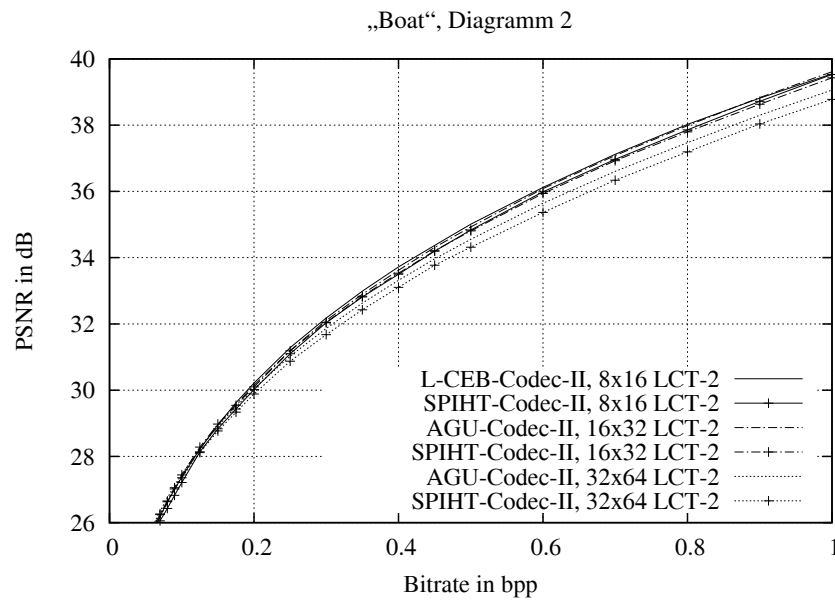


Abbildung H.27: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

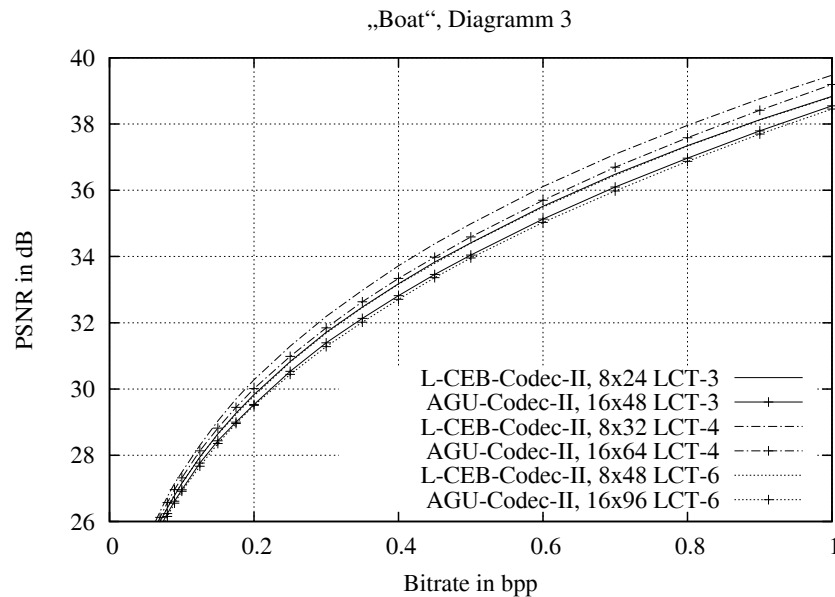


Abbildung H.28: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

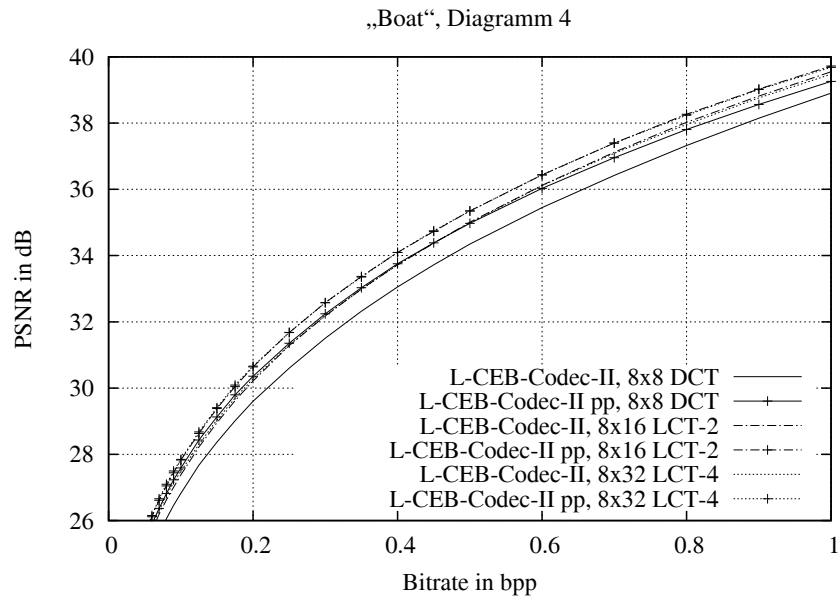


Abbildung H.29: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

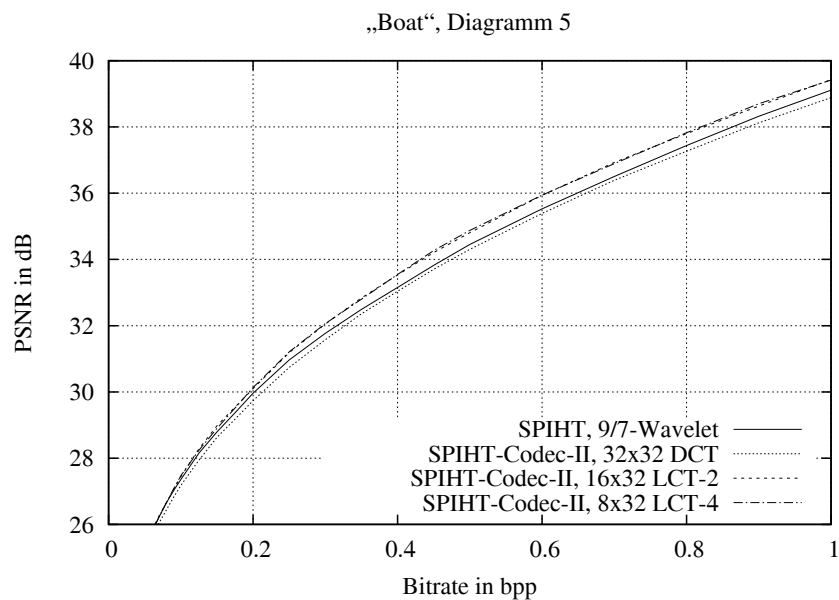


Abbildung H.30: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

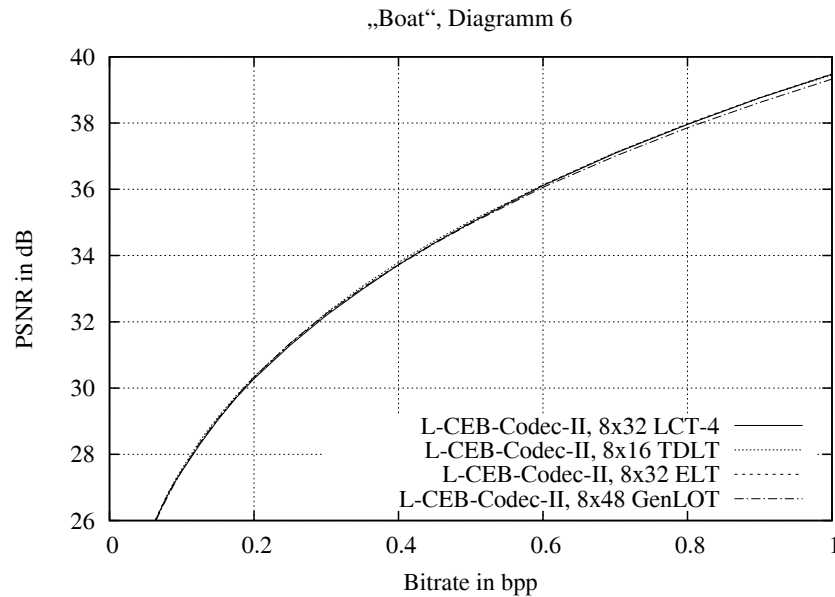


Abbildung H.31: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

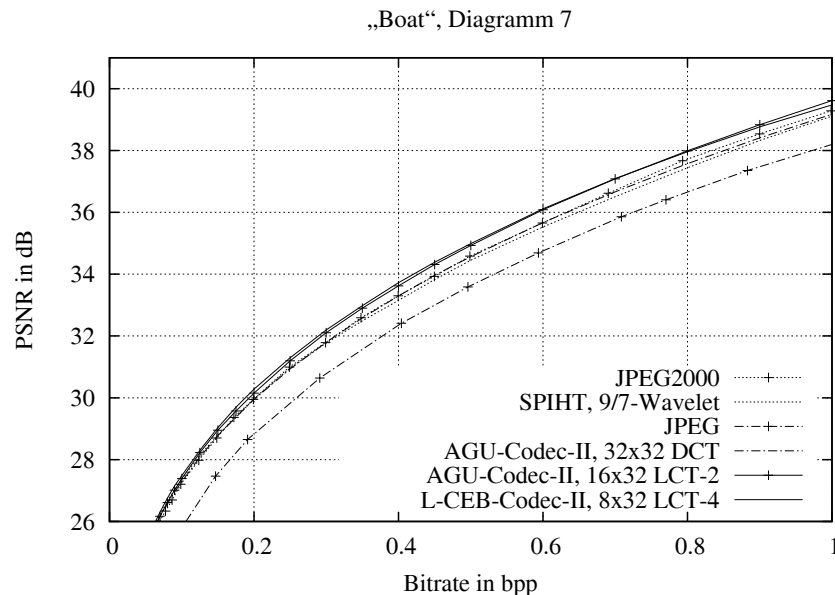


Abbildung H.32: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	25.27	27.67	30.60	34.35	36.89	38.91
SPIHT-Codec-II, 8×8 DCT	25.18	27.50	30.43	34.19	36.81	38.93
AGU-Codec-II, 16×16 DCT	25.47	27.88	30.87	34.66	37.28	39.36
SPIHT-Codec-II, 16×16 DCT	25.58	27.91	30.81	34.51	37.07	39.18
AGU-Codec-II, 32×32 DCT	25.84	28.08	30.94	34.56	37.14	39.17
SPIHT-Codec-II, 32×32 DCT	25.76	27.94	30.74	34.31	36.82	38.89
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	25.82	28.22	31.28	35.01	37.58	39.55
SPIHT-Codec-II, 8×16 LCT-2	25.74	28.14	31.10	34.83	37.40	39.53
AGU-Codec-II, 16×32 LCT-2	25.82	28.22	31.20	34.93	37.54	39.61
SPIHT-Codec-II, 16×32 LCT-2	25.95	28.28	31.19	34.81	37.37	39.43
AGU-Codec-II, 32×64 LCT-2	26.01	28.23	31.03	34.56	37.05	39.06
SPIHT-Codec-II, 32×64 LCT-2	25.95	28.12	30.87	34.31	36.76	38.78
L-CEB-Codec-II, 8×16 TDLT	25.96	28.41	31.34	35.05	37.55	39.45
SPIHT-Codec-II, 8×16 TDLT	25.85	28.19	31.17	34.90	37.45	39.51
L-CEB-Codec-II, 8×16 LOT Typ II	25.77	28.13	31.09	34.75	37.28	39.22
SPIHT-Codec-II, 8×16 LOT Typ II	25.71	28.07	30.92	34.61	37.13	39.26
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	25.67	27.94	30.83	34.41	36.92	38.83
SPIHT-Codec-II, 8×24 LCT-3	25.57	27.83	30.72	34.33	36.88	38.77
AGU-Codec-II, 16×48 LCT-3	25.46	27.76	30.53	34.04	36.55	38.55
SPIHT-Codec-II, 16×48 LCT-3	25.55	27.78	30.53	34.04	36.45	38.41
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	25.95	28.30	31.29	34.99	37.54	39.47
SPIHT-Codec-II, 8×32 LCT-4	25.90	28.24	31.21	34.88	37.40	39.41
AGU-Codec-II, 16×64 LCT-4	25.78	28.13	30.98	34.59	37.15	39.19
SPIHT-Codec-II, 16×64 LCT-4	25.89	28.20	30.98	34.51	36.99	39.00
L-CEB-Codec-II, 8×32 GenLOT	25.89	28.32	31.30	35.01	37.58	39.51
SPIHT-Codec-II, 8×32 GenLOT	25.85	28.24	31.14	34.87	37.41	39.52
L-CEB-Codec-II, 8×32 ELT	25.97	28.34	31.34	35.02	37.56	39.49
SPIHT-Codec-II, 8×32 ELT	25.96	28.29	31.20	34.86	37.37	39.46
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	25.67	27.95	30.81	34.41	36.90	38.82
SPIHT-Codec-II, 8×48 LCT-6	25.59	27.84	30.70	34.29	36.84	38.72
AGU-Codec-II, 16×96 LCT-6	25.42	27.67	30.44	33.96	36.43	38.46
SPIHT-Codec-II, 16×96 LCT-6	25.55	27.74	30.48	33.92	36.34	38.32
L-CEB-Codec-II, 8×48 GenLOT	25.93	28.35	31.36	34.96	37.44	39.33
SPIHT-Codec-II, 8×48 GenLOT	25.92	28.33	31.24	34.86	37.30	39.33
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	25.84	28.04	30.79	34.20	36.72	38.73
AGU-Codec-II, 9/7-Wavelet	25.70	27.92	30.72	34.22	36.83	38.92
SPIHT, 9/7-Wavelet	25.89	28.15	30.97	34.45	36.96	39.11
JPEG2000	25.45	27.98	30.99	34.58	37.27	39.28

Tabelle H.7: Kompressionsergebnisse für das Testbild „Boat“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	26.02	28.43	31.35	34.98	37.39	39.26
SPIHT-Codec-II pp, 8×8 DCT	25.80	28.09	30.99	34.68	37.16	39.21
AGU-Codec-II pp, 16×16 DCT	25.96	28.38	31.40	35.12	37.66	39.62
SPIHT-Codec-II pp, 16×16 DCT	25.98	28.31	31.21	34.88	37.31	39.40
AGU-Codec-II pp, 32×32 DCT	26.21	28.52	31.44	35.04	37.56	39.51
SPIHT-Codec-II pp, 32×32 DCT	26.05	28.28	31.14	34.70	37.07	39.17
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	26.25	28.63	31.67	35.34	37.83	39.72
SPIHT-Codec-II pp, 8×16 LCT-2	26.00	28.42	31.37	35.06	37.53	39.55
AGU-Codec-II pp, 16×32 LCT-2	26.08	28.56	31.56	35.28	37.84	39.83
SPIHT-Codec-II pp, 16×32 LCT-2	26.15	28.51	31.46	35.09	37.54	39.49
AGU-Codec-II pp, 32×64 LCT-2	26.31	28.62	31.51	35.05	37.51	39.45
SPIHT-Codec-II pp, 32×64 LCT-2	26.14	28.38	31.24	34.73	37.03	39.13
L-CEB-Codec-II pp, 8×16 TDLT	26.11	28.63	31.60	35.30	37.74	39.57
SPIHT-Codec-II pp, 8×16 TDLT	25.92	28.29	31.30	35.03	37.50	39.54
L-CEB-Codec-II pp, 8×16 LOT Typ II	26.21	28.62	31.60	35.26	37.69	39.55
SPIHT-Codec-II pp, 8×16 LOT Typ II	26.02	28.44	31.28	34.97	37.40	39.48
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	26.10	28.43	31.36	34.94	37.37	39.24
SPIHT-Codec-II pp, 8×24 LCT-3	25.85	28.11	31.04	34.63	37.08	38.96
AGU-Codec-II pp, 16×48 LCT-3	25.87	28.27	31.13	34.68	37.15	39.09
SPIHT-Codec-II pp, 16×48 LCT-3	25.84	28.10	30.92	34.41	36.91	38.71
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	26.29	28.68	31.68	35.35	37.83	39.68
SPIHT-Codec-II pp, 8×32 LCT-4	26.07	28.45	31.42	35.05	37.57	39.46
AGU-Codec-II pp, 16×64 LCT-4	26.08	28.51	31.42	35.06	37.58	39.54
SPIHT-Codec-II pp, 16×64 LCT-4	26.06	28.44	31.25	34.76	37.28	39.15
L-CEB-Codec-II pp, 8×32 GenLOT	26.26	28.71	31.71	35.38	37.86	39.71
SPIHT-Codec-II pp, 8×32 GenLOT	26.12	28.53	31.40	35.11	37.57	39.55
L-CEB-Codec-II pp, 8×32 ELT	26.31	28.70	31.74	35.40	37.87	39.73
SPIHT-Codec-II pp, 8×32 ELT	26.14	28.54	31.45	35.11	37.54	39.50
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	26.12	28.48	31.41	34.98	37.42	39.26
SPIHT-Codec-II pp, 8×48 LCT-6	25.89	28.17	31.07	34.64	37.05	38.96
AGU-Codec-II pp, 16×96 LCT-6	25.86	28.21	31.10	34.64	37.09	39.05
SPIHT-Codec-II pp, 16×96 LCT-6	25.86	28.11	30.92	34.36	36.89	38.68
L-CEB-Codec-II pp, 8×48 GenLOT	26.24	28.72	31.77	35.39	37.80	39.61
SPIHT-Codec-II pp, 8×48 GenLOT	26.09	28.58	31.51	35.15	37.52	39.40
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	26.06	28.34	31.14	34.58	37.05	38.96
AGU-Codec-II pp, 9/7-Wavelet	25.84	28.13	30.98	34.52	37.11	39.12
SPIHT pp, 9/7-Wavelet	26.00	28.34	31.15	34.72	37.11	39.14

Tabelle H.8: Kompressionsergebnisse für das Testbild „Boat“, Vergleich der Codecs mit Post-processing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Goldhill“



Abbildung H.33: Testbild „Goldhill“ (512×512 , $H = 7.4778$, $\sigma^2 = 2423.2778$, $\bar{g} = 112.2034$, $g_{min} = 16$, $g_{max} = 235$).

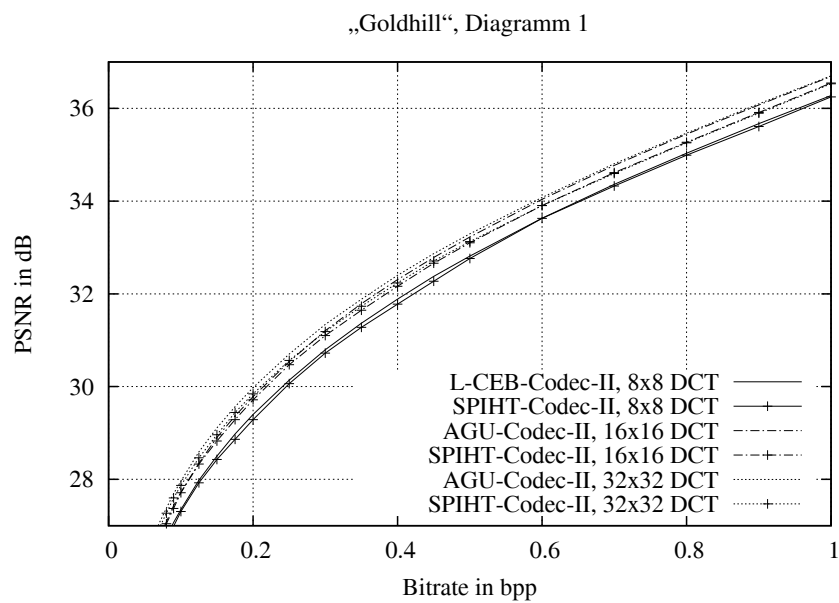


Abbildung H.34: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

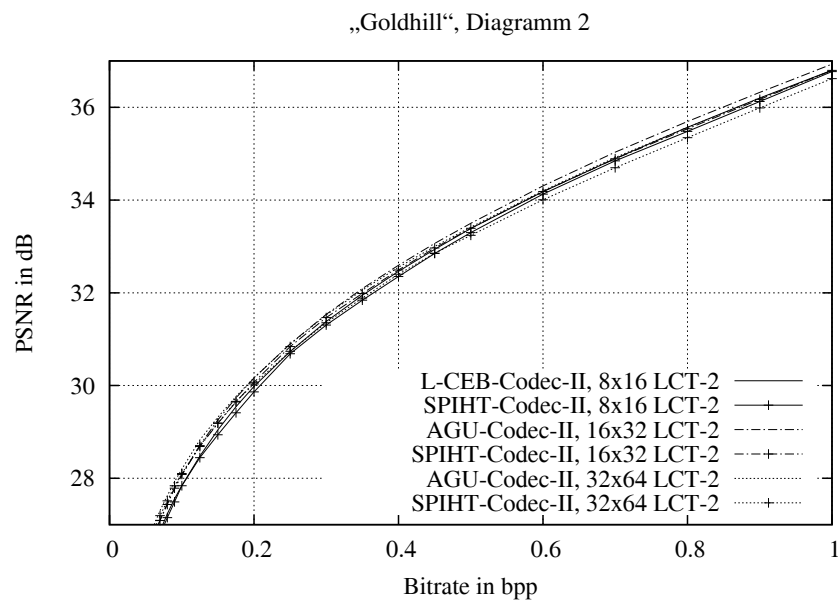


Abbildung H.35: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

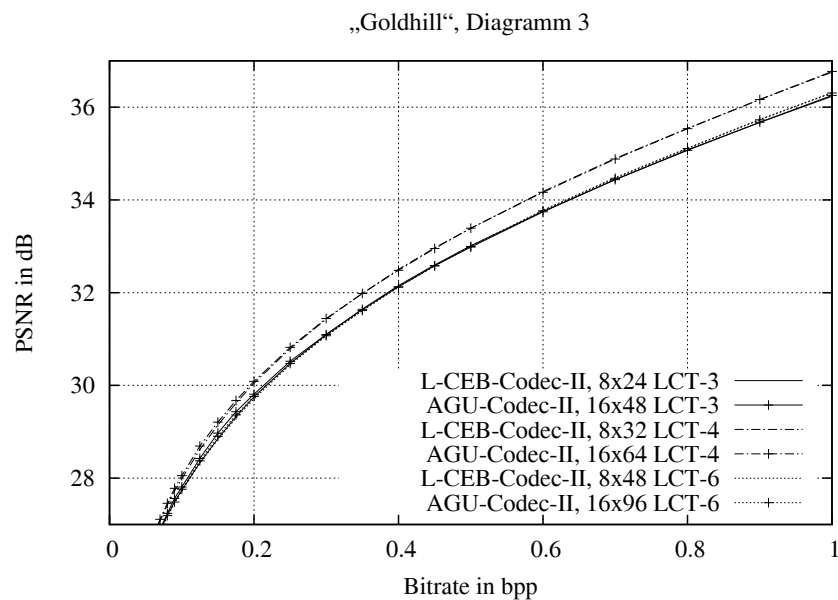


Abbildung H.36: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

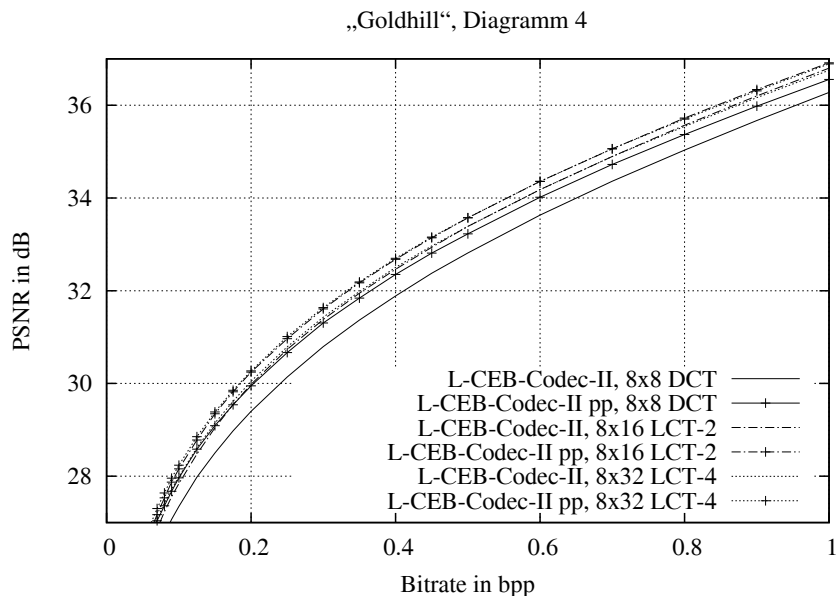


Abbildung H.37: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

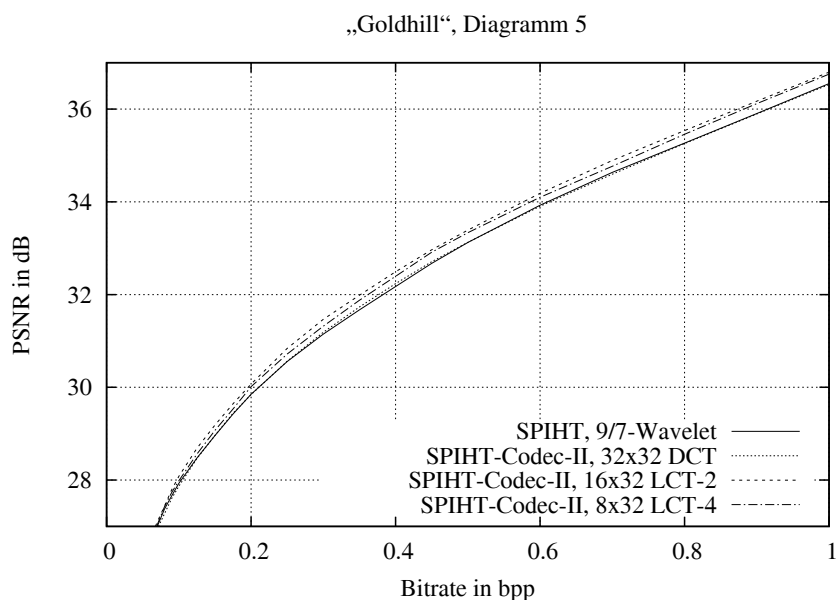


Abbildung H.38: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

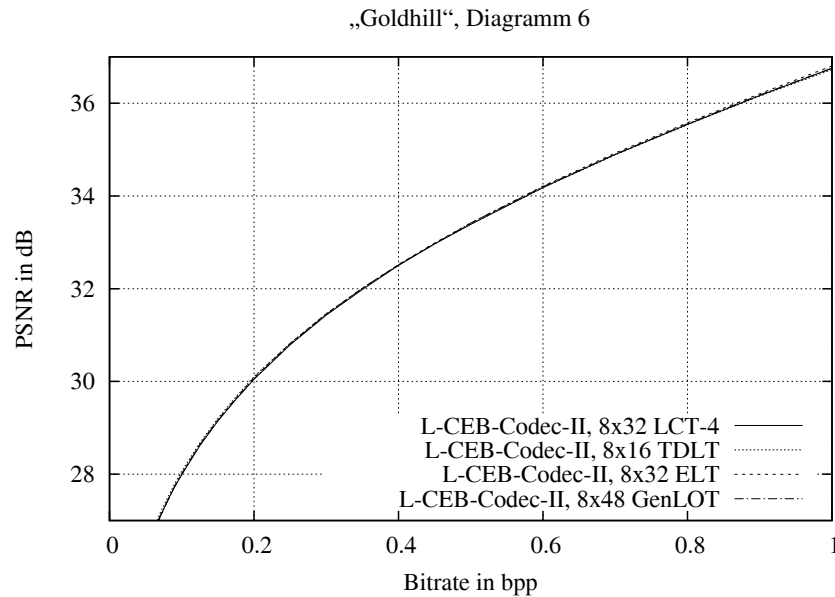


Abbildung H.39: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

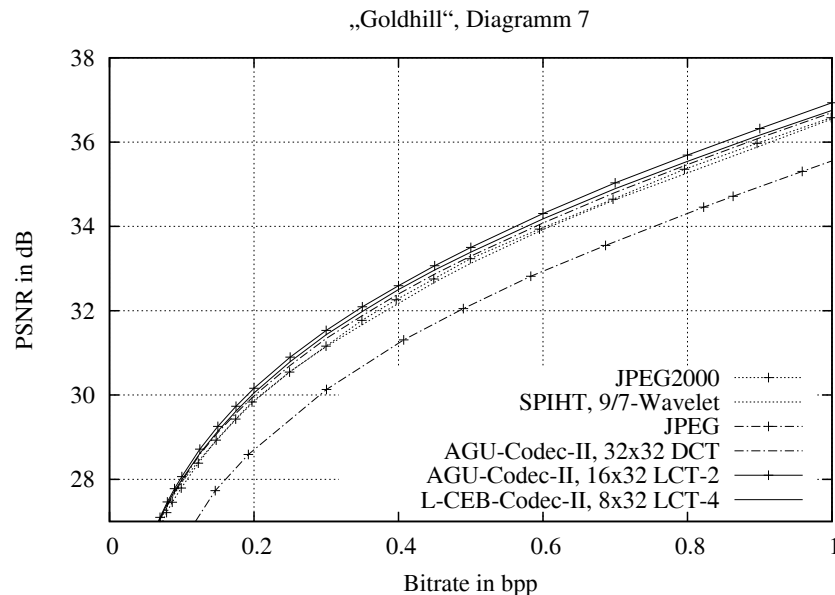


Abbildung H.40: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	26.11	27.98	30.14	32.82	34.70	36.28
SPIHT-Codec-II, 8×8 DCT	26.00	27.92	30.06	32.76	34.68	36.25
AGU-Codec-II, 16×16 DCT	26.42	28.35	30.54	33.23	35.12	36.69
SPIHT-Codec-II, 16×16 DCT	26.38	28.33	30.47	33.10	34.94	36.54
AGU-Codec-II, 32×32 DCT	26.77	28.59	30.70	33.30	35.14	36.70
SPIHT-Codec-II, 32×32 DCT	26.66	28.47	30.56	33.13	34.93	36.53
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	26.56	28.48	30.73	33.39	35.23	36.80
SPIHT-Codec-II, 8×16 LCT-2	26.49	28.44	30.69	33.30	35.16	36.77
AGU-Codec-II, 16×32 LCT-2	26.80	28.71	30.90	33.50	35.37	36.93
SPIHT-Codec-II, 16×32 LCT-2	26.82	28.68	30.84	33.39	35.21	36.80
AGU-Codec-II, 32×64 LCT-2	27.04	28.83	30.89	33.42	35.25	36.79
SPIHT-Codec-II, 32×64 LCT-2	26.93	28.70	30.74	33.24	35.02	36.62
L-CEB-Codec-II, 8×16 TDLT	26.87	28.67	30.82	33.40	35.21	36.72
SPIHT-Codec-II, 8×16 TDLT	26.80	28.62	30.68	33.27	35.14	36.62
L-CEB-Codec-II, 8×16 LOT Typ II	26.55	28.44	30.61	33.25	35.10	36.66
SPIHT-Codec-II, 8×16 LOT Typ II	26.46	28.36	30.56	33.17	35.03	36.63
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	26.60	28.33	30.47	33.01	34.75	36.23
SPIHT-Codec-II, 8×24 LCT-3	26.65	28.22	30.41	32.98	34.63	36.30
AGU-Codec-II, 16×48 LCT-3	26.62	28.42	30.52	32.98	34.76	36.25
SPIHT-Codec-II, 16×48 LCT-3	26.72	28.31	30.45	32.96	34.57	36.19
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	26.78	28.61	30.79	33.39	35.23	36.75
SPIHT-Codec-II, 8×32 LCT-4	26.74	28.56	30.72	33.33	35.12	36.75
AGU-Codec-II, 16×64 LCT-4	26.84	28.69	30.82	33.39	35.22	36.77
SPIHT-Codec-II, 16×64 LCT-4	26.87	28.63	30.76	33.29	35.09	36.67
L-CEB-Codec-II, 8×32 GenLOT	26.70	28.54	30.74	33.39	35.23	36.80
SPIHT-Codec-II, 8×32 GenLOT	26.59	28.49	30.69	33.30	35.15	36.76
L-CEB-Codec-II, 8×32 ELT	26.80	28.64	30.82	33.42	35.26	36.80
SPIHT-Codec-II, 8×32 ELT	26.71	28.58	30.78	33.33	35.16	36.75
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	26.58	28.32	30.44	33.01	34.80	36.31
SPIHT-Codec-II, 8×48 LCT-6	26.66	28.28	30.37	32.97	34.74	36.31
AGU-Codec-II, 16×96 LCT-6	26.56	28.37	30.47	33.00	34.79	36.31
SPIHT-Codec-II, 16×96 LCT-6	26.70	28.35	30.42	32.94	34.66	36.22
L-CEB-Codec-II, 8×48 GenLOT	26.83	28.64	30.80	33.42	35.23	36.76
SPIHT-Codec-II, 8×48 GenLOT	26.77	28.59	30.76	33.33	35.15	36.73
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	26.75	28.48	30.47	33.02	34.83	36.34
AGU-Codec-II, 9/7-Wavelet	26.65	28.37	30.44	33.05	34.92	36.49
SPIHT, 9/7-Wavelet	26.71	28.47	30.56	33.12	34.94	36.55
JPEG2000	26.55	28.38	30.54	33.23	34.97	36.58

Tabelle H.9: Kompressionsergebnisse für das Testbild „Goldhill“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	26.75	28.58	30.66	33.23	35.05	36.55
SPIHT-Codec-II pp, 8×8 DCT	26.50	28.41	30.44	33.02	34.90	36.39
AGU-Codec-II pp, 16×16 DCT	26.84	28.72	30.87	33.49	35.33	36.86
SPIHT-Codec-II pp, 16×16 DCT	26.69	28.62	30.73	33.25	35.07	36.61
AGU-Codec-II pp, 32×32 DCT	27.04	28.88	30.98	33.53	35.36	36.89
SPIHT-Codec-II pp, 32×32 DCT	26.84	28.66	30.77	33.26	35.05	36.59
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	26.88	28.78	30.96	33.57	35.39	36.91
SPIHT-Codec-II pp, 8×16 LCT-2	26.69	28.62	30.83	33.36	35.21	36.78
AGU-Codec-II pp, 16×32 LCT-2	27.00	28.88	31.07	33.67	35.51	37.04
SPIHT-Codec-II pp, 16×32 LCT-2	26.91	28.79	30.95	33.46	35.27	36.81
AGU-Codec-II pp, 32×64 LCT-2	27.18	29.00	31.10	33.63	35.44	36.97
SPIHT-Codec-II pp, 32×64 LCT-2	26.99	28.81	30.88	33.35	35.11	36.67
L-CEB-Codec-II pp, 8×16 TDLT	26.95	28.80	30.96	33.53	35.33	36.82
SPIHT-Codec-II pp, 8×16 TDLT	26.85	28.64	30.73	33.31	35.16	36.62
L-CEB-Codec-II pp, 8×16 LOT Typ II	26.90	28.77	30.92	33.52	35.33	36.85
SPIHT-Codec-II pp, 8×16 LOT Typ II	26.70	28.61	30.79	33.28	35.16	36.67
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	26.86	28.67	30.79	33.31	35.05	36.50
SPIHT-Codec-II pp, 8×24 LCT-3	26.80	28.42	30.57	33.12	34.78	36.41
AGU-Codec-II pp, 16×48 LCT-3	26.86	28.68	30.80	33.30	35.06	36.55
SPIHT-Codec-II pp, 16×48 LCT-3	26.88	28.48	30.61	33.12	34.75	36.33
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	27.00	28.85	31.01	33.58	35.39	36.89
SPIHT-Codec-II pp, 8×32 LCT-4	26.84	28.66	30.84	33.42	35.16	36.77
AGU-Codec-II pp, 16×64 LCT-4	27.00	28.87	31.02	33.59	35.40	36.93
SPIHT-Codec-II pp, 16×64 LCT-4	26.96	28.74	30.87	33.41	35.16	36.72
L-CEB-Codec-II pp, 8×32 GenLOT	26.97	28.83	30.98	33.59	35.40	36.93
SPIHT-Codec-II pp, 8×32 GenLOT	26.78	28.68	30.86	33.36	35.22	36.76
L-CEB-Codec-II pp, 8×32 ELT	27.02	28.88	31.04	33.61	35.43	36.94
SPIHT-Codec-II pp, 8×32 ELT	26.83	28.72	30.92	33.39	35.22	36.77
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	26.90	28.69	30.79	33.33	35.11	36.58
SPIHT-Codec-II pp, 8×48 LCT-6	26.87	28.49	30.57	33.14	34.89	36.43
AGU-Codec-II pp, 16×96 LCT-6	26.80	28.67	30.79	33.32	35.10	36.61
SPIHT-Codec-II pp, 16×96 LCT-6	26.89	28.53	30.61	33.14	34.85	36.38
L-CEB-Codec-II pp, 8×48 GenLOT	27.05	28.88	31.03	33.62	35.42	36.92
SPIHT-Codec-II pp, 8×48 GenLOT	26.88	28.73	30.92	33.41	35.24	36.78
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	26.89	28.64	30.65	33.19	34.98	36.46
AGU-Codec-II pp, 9/7-Wavelet	26.74	28.48	30.54	33.18	35.02	36.57
SPIHT pp, 9/7-Wavelet	26.75	28.53	30.66	33.18	34.97	36.53

Tabelle H.10: Kompressionsergebnisse für das Testbild „Goldhill“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Lighthouse“



Abbildung H.41: Testbild „Lighthouse“ (512×512 , $H = 7.4486$, $\sigma^2 = 2881.7162$, $\bar{g} = 115.3908$, $g_{\min} = 2$, $g_{\max} = 255$).

„Lighthouse“, Diagramm 1

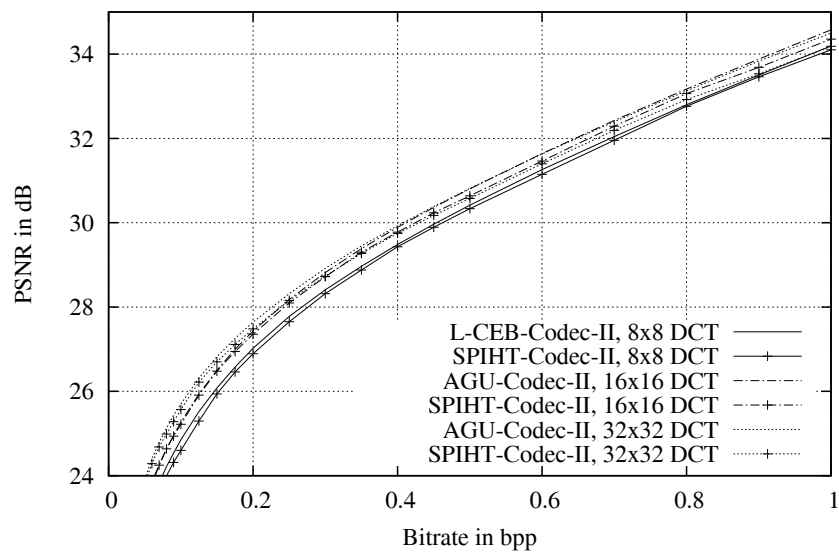


Abbildung H.42: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

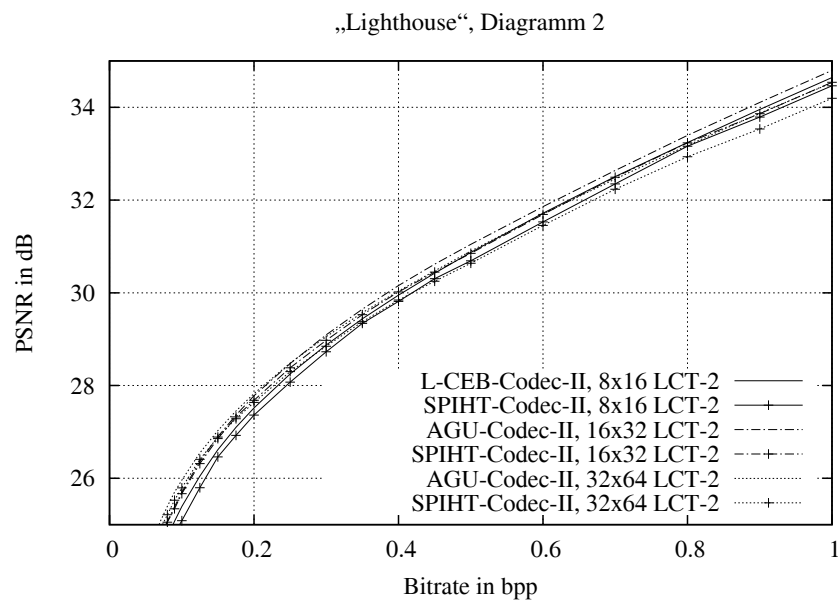


Abbildung H.43: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

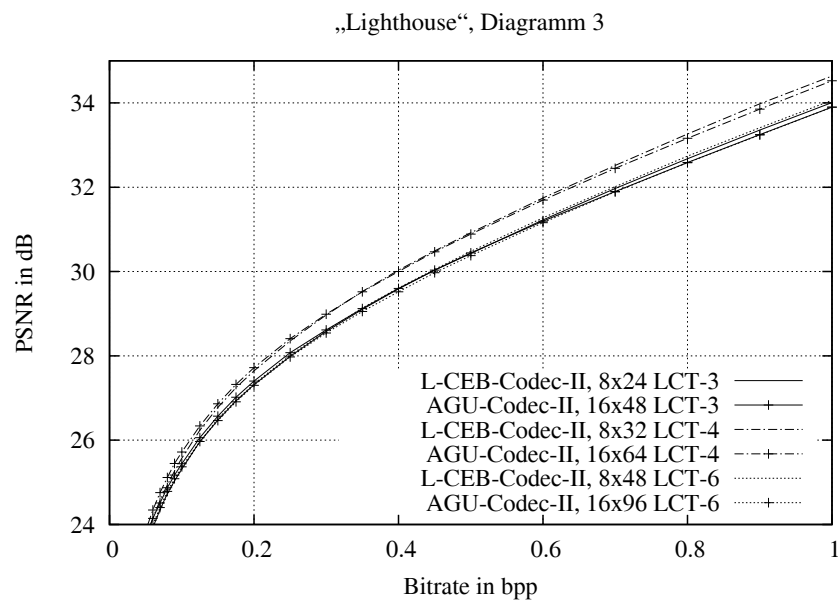


Abbildung H.44: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

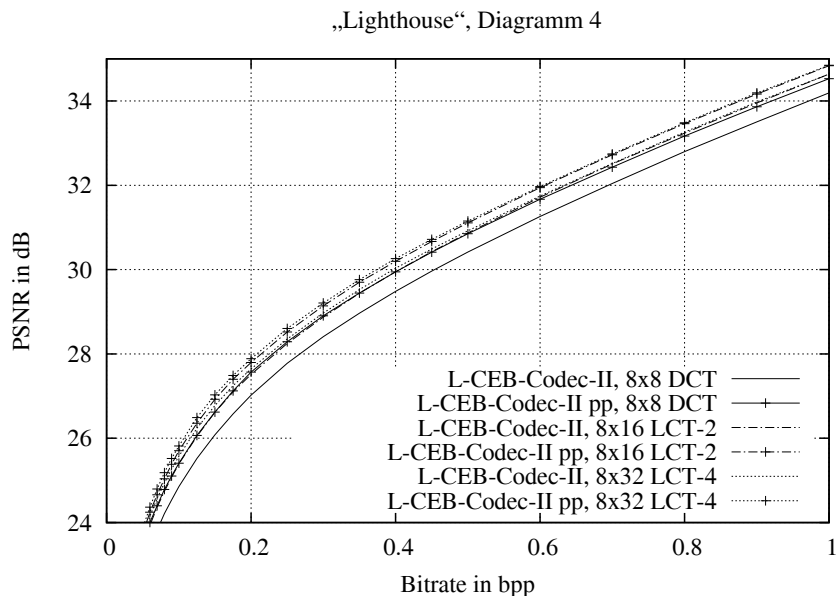


Abbildung H.45: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

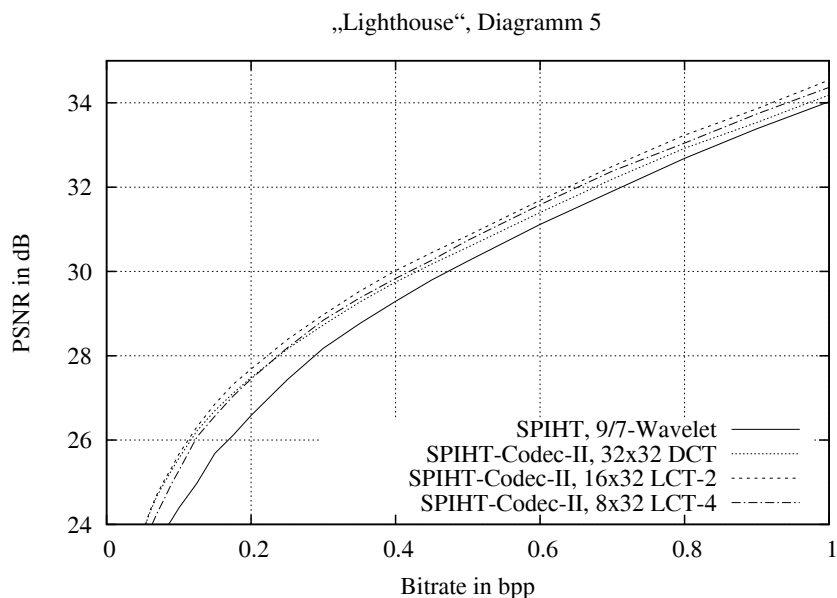


Abbildung H.46: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

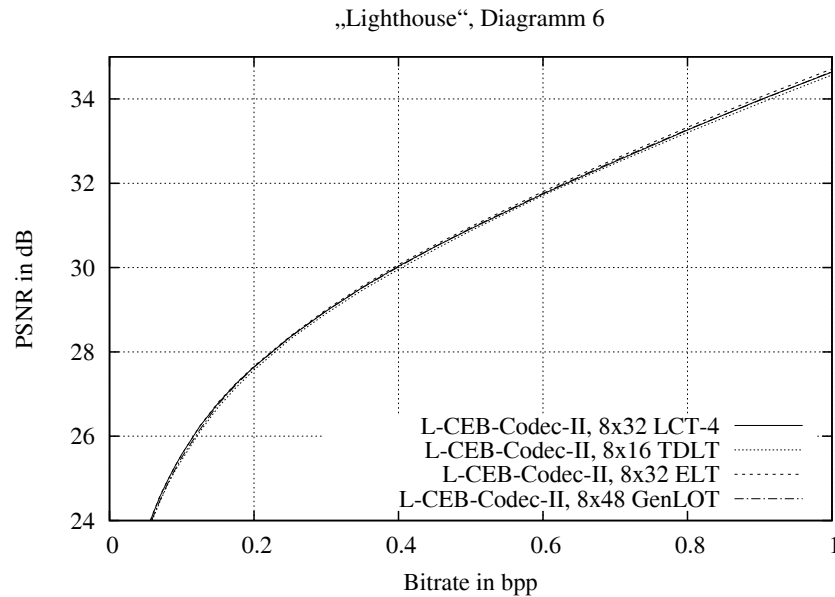


Abbildung H.47: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

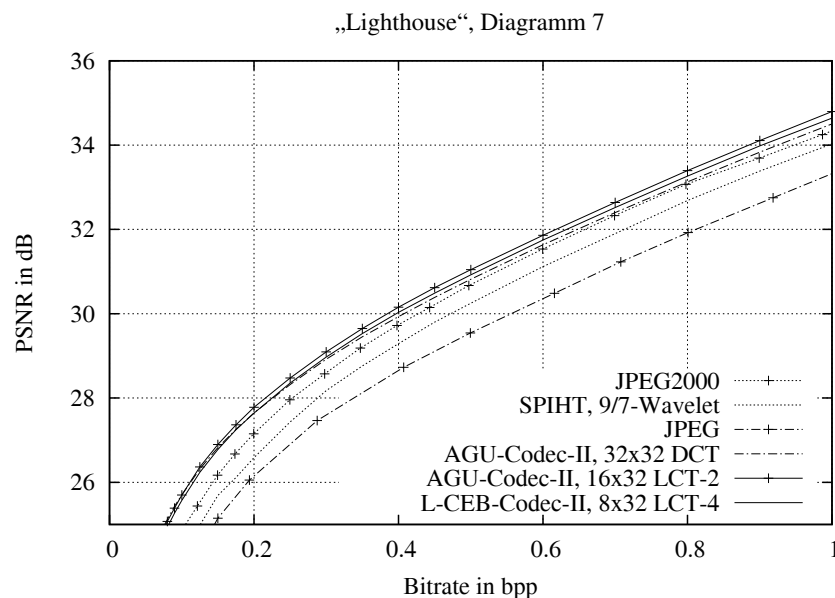


Abbildung H.48: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	23.52	25.52	27.78	30.42	32.43	34.19
SPIHT-Codec-II, 8×8 DCT	23.29	25.30	27.65	30.33	32.37	34.10
AGU-Codec-II, 16×16 DCT	23.90	25.93	28.18	30.81	32.81	34.57
SPIHT-Codec-II, 16×16 DCT	23.97	25.91	28.10	30.64	32.68	34.36
AGU-Codec-II, 32×32 DCT	24.51	26.32	28.32	30.81	32.77	34.50
SPIHT-Codec-II, 32×32 DCT	24.38	26.22	28.16	30.58	32.56	34.18
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	24.05	26.05	28.25	30.87	32.87	34.64
SPIHT-Codec-II, 8×16 LCT-2	23.83	25.80	28.07	30.69	32.75	34.46
AGU-Codec-II, 16×32 LCT-2	24.40	26.36	28.47	31.05	33.02	34.80
SPIHT-Codec-II, 16×32 LCT-2	24.43	26.32	28.38	30.85	32.89	34.54
AGU-Codec-II, 32×64 LCT-2	24.78	26.55	28.48	30.90	32.81	34.53
SPIHT-Codec-II, 32×64 LCT-2	24.64	26.41	28.30	30.64	32.59	34.19
L-CEB-Codec-II, 8×16 TDLT	24.16	26.11	28.30	30.87	32.84	34.57
SPIHT-Codec-II, 8×16 TDLT	23.82	25.87	28.16	30.74	32.66	34.48
L-CEB-Codec-II, 8×16 LOT Typ II	24.02	26.01	28.20	30.81	32.79	34.53
SPIHT-Codec-II, 8×16 LOT Typ II	23.82	25.76	28.02	30.63	32.68	34.38
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	24.03	25.97	27.99	30.43	32.33	34.01
SPIHT-Codec-II, 8×24 LCT-3	23.79	25.78	27.93	30.28	32.22	33.84
AGU-Codec-II, 16×48 LCT-3	24.25	26.06	28.08	30.44	32.25	33.90
SPIHT-Codec-II, 16×48 LCT-3	24.21	26.04	28.08	30.31	32.15	33.76
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	24.27	26.24	28.35	30.92	32.89	34.64
SPIHT-Codec-II, 8×32 LCT-4	23.99	26.07	28.19	30.74	32.71	34.37
AGU-Codec-II, 16×64 LCT-4	24.45	26.34	28.41	30.89	32.81	34.53
SPIHT-Codec-II, 16×64 LCT-4	24.50	26.33	28.31	30.74	32.63	34.27
L-CEB-Codec-II, 8×32 GenLOT	24.16	26.15	28.28	30.92	32.93	34.69
SPIHT-Codec-II, 8×32 GenLOT	23.96	25.88	28.13	30.72	32.80	34.51
L-CEB-Codec-II, 8×32 ELT	24.28	26.24	28.37	30.97	32.96	34.72
SPIHT-Codec-II, 8×32 ELT	24.08	26.02	28.21	30.76	32.83	34.50
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	24.04	25.95	28.01	30.48	32.39	34.06
SPIHT-Codec-II, 8×48 LCT-6	23.77	25.78	27.91	30.33	32.21	33.91
AGU-Codec-II, 16×96 LCT-6	24.10	25.97	27.98	30.38	32.24	33.90
SPIHT-Codec-II, 16×96 LCT-6	24.12	25.96	27.95	30.27	32.08	33.75
L-CEB-Codec-II, 8×48 GenLOT	24.19	26.16	28.36	30.94	32.91	34.65
SPIHT-Codec-II, 8×48 GenLOT	24.02	25.95	28.19	30.73	32.79	34.47
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	23.29	24.93	27.28	30.06	32.07	33.84
AGU-Codec-II, 9/7-Wavelet	23.26	24.92	27.25	30.13	32.16	33.98
SPIHT, 9/7-Wavelet	23.31	24.98	27.43	30.25	32.31	34.03
JPEG2000	23.44	25.44	27.96	30.67	32.68	34.25

Tabelle H.11: Kompressionsergebnisse für das Testbild „Lighthouse“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	24.10	26.07	28.29	30.85	32.81	34.53
SPIHT-Codec-II pp, 8×8 DCT	23.68	25.72	28.02	30.63	32.58	34.30
AGU-Codec-II pp, 16×16 DCT	24.27	26.28	28.52	31.09	33.07	34.83
SPIHT-Codec-II pp, 16×16 DCT	24.29	26.16	28.32	30.82	32.81	34.50
AGU-Codec-II pp, 32×32 DCT	24.79	26.60	28.62	31.10	33.05	34.79
SPIHT-Codec-II pp, 32×32 DCT	24.60	26.39	28.35	30.74	32.71	34.34
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	24.35	26.36	28.52	31.11	33.10	34.84
SPIHT-Codec-II pp, 8×16 LCT-2	24.03	26.01	28.24	30.82	32.84	34.55
AGU-Codec-II pp, 16×32 LCT-2	24.59	26.58	28.70	31.25	33.23	34.99
SPIHT-Codec-II pp, 16×32 LCT-2	24.57	26.46	28.50	30.96	32.98	34.63
AGU-Codec-II pp, 32×64 LCT-2	25.00	26.80	28.76	31.19	33.11	34.82
SPIHT-Codec-II pp, 32×64 LCT-2	24.81	26.60	28.48	30.80	32.77	34.34
L-CEB-Codec-II pp, 8×16 TDLT	24.30	26.28	28.49	31.07	33.03	34.74
SPIHT-Codec-II pp, 8×16 TDLT	23.87	25.95	28.25	30.82	32.74	34.52
L-CEB-Codec-II pp, 8×16 LOT Typ II	24.33	26.34	28.52	31.10	33.06	34.79
SPIHT-Codec-II pp, 8×16 LOT Typ II	23.98	25.97	28.23	30.80	32.77	34.53
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	24.30	26.30	28.35	30.80	32.70	34.37
SPIHT-Codec-II pp, 8×24 LCT-3	23.94	25.98	28.13	30.51	32.40	34.03
AGU-Codec-II pp, 16×48 LCT-3	24.47	26.38	28.46	30.84	32.66	34.32
SPIHT-Codec-II pp, 16×48 LCT-3	24.35	26.27	28.31	30.59	32.37	34.01
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	24.50	26.49	28.60	31.15	33.12	34.85
SPIHT-Codec-II pp, 8×32 LCT-4	24.14	26.18	28.33	30.85	32.81	34.46
AGU-Codec-II pp, 16×64 LCT-4	24.64	26.60	28.67	31.16	33.08	34.81
SPIHT-Codec-II pp, 16×64 LCT-4	24.63	26.51	28.49	30.89	32.78	34.41
L-CEB-Codec-II pp, 8×32 GenLOT	24.41	26.41	28.55	31.16	33.16	34.88
SPIHT-Codec-II pp, 8×32 GenLOT	24.05	26.03	28.28	30.85	32.86	34.60
L-CEB-Codec-II pp, 8×32 ELT	24.49	26.49	28.62	31.19	33.17	34.92
SPIHT-Codec-II pp, 8×32 ELT	24.15	26.17	28.34	30.87	32.90	34.59
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	24.33	26.32	28.40	30.87	32.77	34.44
SPIHT-Codec-II pp, 8×48 LCT-6	23.97	26.01	28.13	30.57	32.42	34.12
AGU-Codec-II pp, 16×96 LCT-6	24.40	26.38	28.41	30.82	32.68	34.36
SPIHT-Codec-II pp, 16×96 LCT-6	24.34	26.24	28.23	30.58	32.36	34.05
L-CEB-Codec-II pp, 8×48 GenLOT	24.42	26.43	28.61	31.18	33.15	34.89
SPIHT-Codec-II pp, 8×48 GenLOT	24.20	26.09	28.32	30.86	32.89	34.59
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	23.51	25.26	27.61	30.31	32.27	34.02
AGU-Codec-II pp, 9/7-Wavelet	23.43	25.14	27.50	30.32	32.31	34.11
SPIHT pp, 9/7-Wavelet	23.46	25.19	27.61	30.36	32.40	34.10

Tabelle H.12: Kompressionsergebnisse für das Testbild „Lighthouse“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Roof“



Abbildung H.49: Testbild „Roof“ (512×512 , $H = 7.0855$, $\sigma^2 = 4676.1827$, $\bar{g} = 80.6938$, $g_{min} = 0$, $g_{max} = 252$).

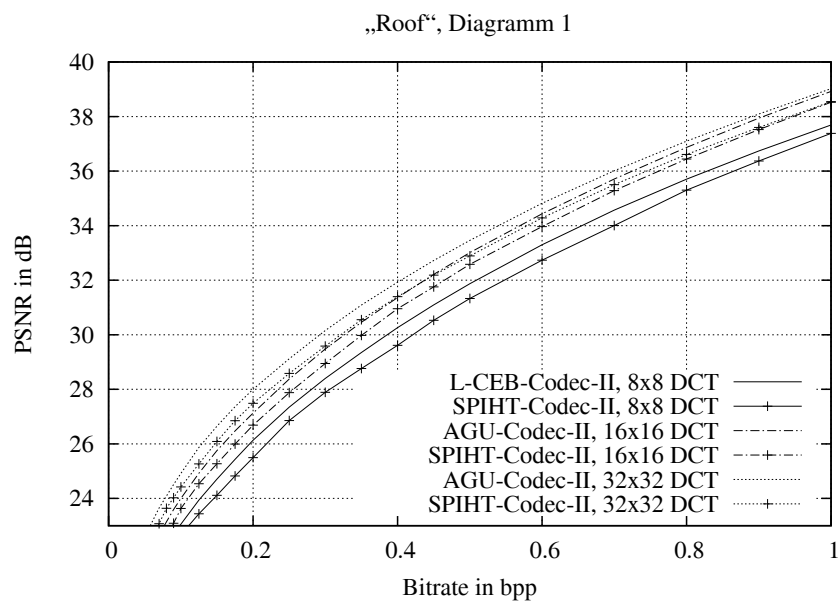


Abbildung H.50: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

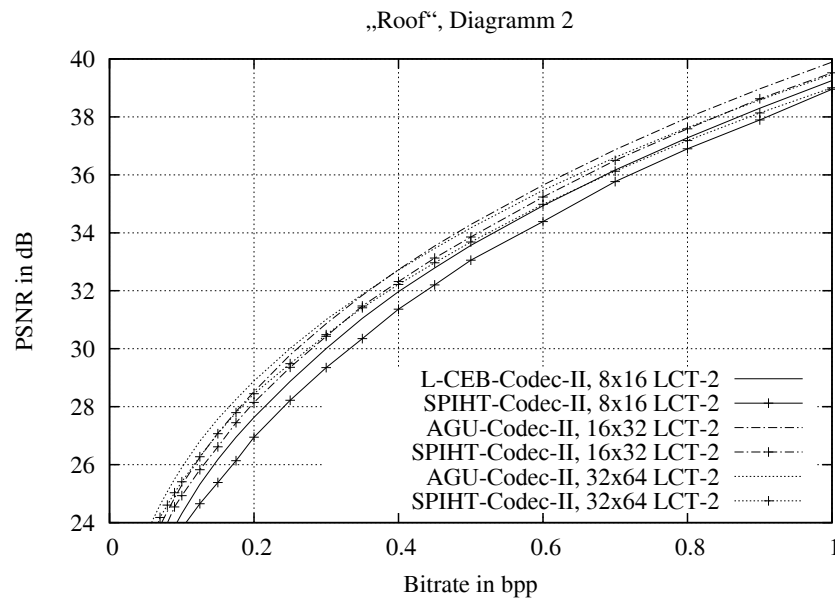


Abbildung H.51: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

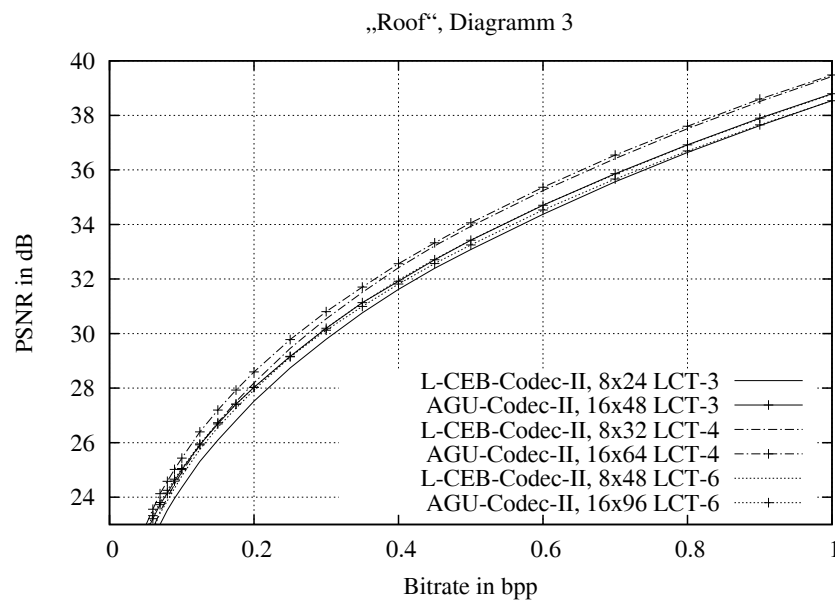


Abbildung H.52: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

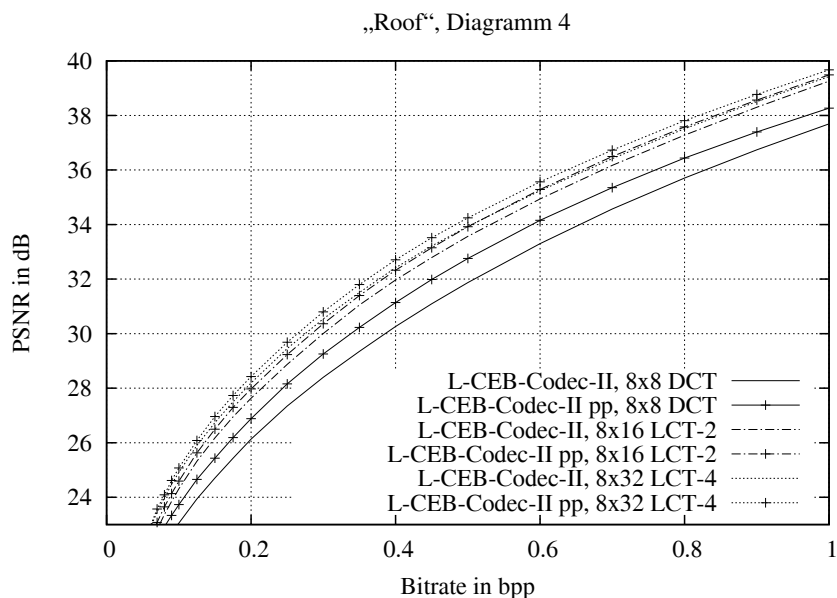


Abbildung H.53: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

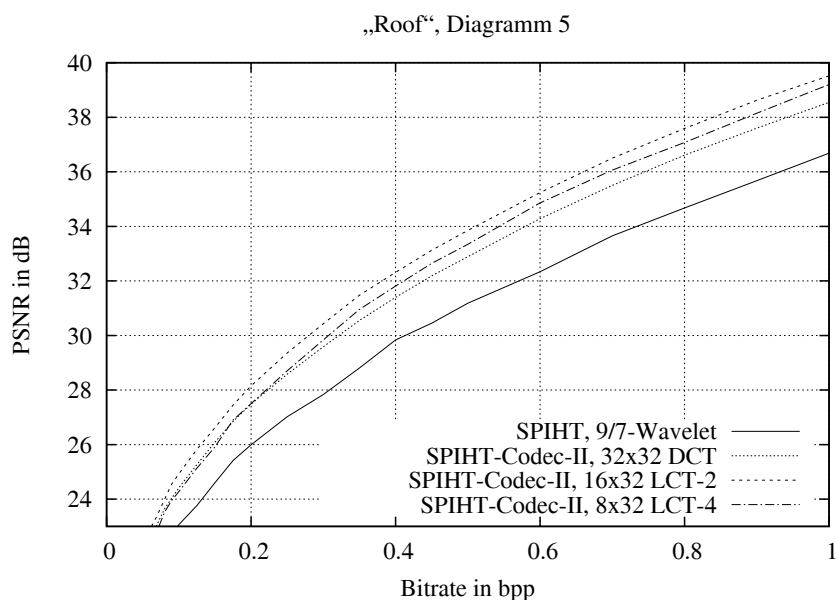


Abbildung H.54: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

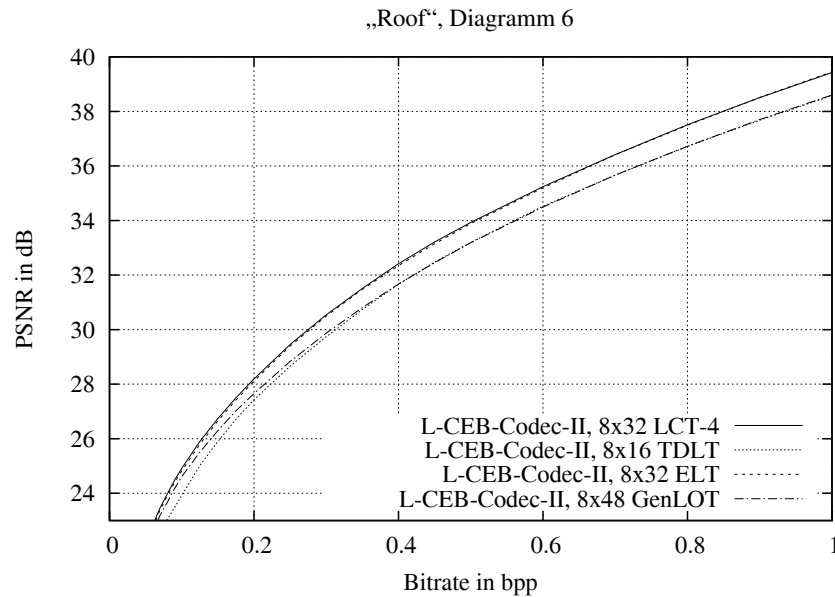


Abbildung H.55: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

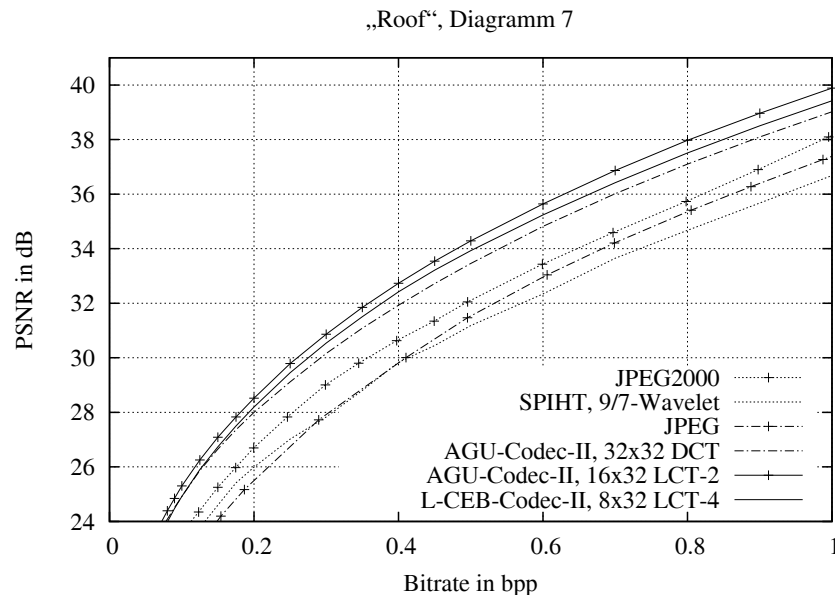


Abbildung H.56: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	21.45	23.94	27.35	31.87	35.14	37.69
SPIHT-Codec-II, 8×8 DCT	20.78	23.43	26.85	31.33	34.64	37.38
AGU-Codec-II, 16×16 DCT	22.26	24.92	28.38	33.01	36.30	38.92
SPIHT-Codec-II, 16×16 DCT	21.81	24.54	27.88	32.58	35.88	38.53
AGU-Codec-II, 32×32 DCT	23.25	25.87	29.12	33.46	36.57	39.02
SPIHT-Codec-II, 32×32 DCT	22.67	25.26	28.58	32.89	36.05	38.54
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	22.45	25.32	28.87	33.57	36.73	39.26
SPIHT-Codec-II, 8×16 LCT-2	21.90	24.65	28.22	33.05	36.34	38.95
AGU-Codec-II, 16×32 LCT-2	23.49	26.26	29.79	34.29	37.44	39.89
SPIHT-Codec-II, 16×32 LCT-2	23.04	25.82	29.36	33.85	37.09	39.52
AGU-Codec-II, 32×64 LCT-2	24.25	26.81	30.01	34.17	37.14	39.47
SPIHT-Codec-II, 32×64 LCT-2	23.67	26.28	29.49	33.68	36.70	39.00
L-CEB-Codec-II, 8×16 TDLT	22.13	25.02	28.66	33.20	36.20	38.58
SPIHT-Codec-II, 8×16 TDLT	21.85	24.44	28.05	32.78	35.90	38.39
L-CEB-Codec-II, 8×16 LOT Typ II	22.36	25.22	28.68	33.25	36.33	38.79
SPIHT-Codec-II, 8×16 LOT Typ II	21.75	24.56	28.05	32.83	35.98	38.51
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	22.58	25.31	28.74	33.08	36.12	38.54
SPIHT-Codec-II, 8×24 LCT-3	22.17	24.56	28.01	32.62	35.65	38.34
AGU-Codec-II, 16×48 LCT-3	23.35	25.92	29.17	33.42	36.40	38.79
SPIHT-Codec-II, 16×48 LCT-3	23.03	25.59	28.89	33.13	36.01	38.58
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	23.01	25.90	29.45	33.93	36.98	39.42
SPIHT-Codec-II, 8×32 LCT-4	22.43	25.14	28.70	33.34	36.59	39.20
AGU-Codec-II, 16×64 LCT-4	23.71	26.39	29.78	34.06	37.09	39.48
SPIHT-Codec-II, 16×64 LCT-4	23.39	25.99	29.41	33.72	36.78	39.20
L-CEB-Codec-II, 8×32 GenLOT	22.30	25.20	28.69	33.22	36.47	39.01
SPIHT-Codec-II, 8×32 GenLOT	21.78	24.49	28.02	32.70	36.06	38.66
L-CEB-Codec-II, 8×32 ELT	22.95	25.80	29.40	33.88	36.95	39.45
SPIHT-Codec-II, 8×32 ELT	22.33	25.07	28.68	33.34	36.58	39.12
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	22.94	25.75	29.14	33.43	36.39	38.78
SPIHT-Codec-II, 8×48 LCT-6	22.60	25.11	28.47	33.02	35.94	38.53
AGU-Codec-II, 16×96 LCT-6	23.45	25.95	29.15	33.25	36.19	38.53
SPIHT-Codec-II, 16×96 LCT-6	23.31	25.76	28.90	33.03	35.87	38.31
L-CEB-Codec-II, 8×48 GenLOT	22.74	25.51	28.83	33.19	36.21	38.61
SPIHT-Codec-II, 8×48 GenLOT	22.19	24.85	28.27	32.77	35.88	38.30
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	21.67	23.97	27.04	31.06	33.93	36.49
AGU-Codec-II, 9/7-Wavelet	21.65	24.03	27.07	31.10	33.97	36.67
SPIHT, 9/7-Wavelet	21.51	23.77	27.02	31.18	34.22	36.68
JPEG2000	21.56	24.35	27.83	32.05	35.18	38.10

Tabelle H.13: Kompressionsergebnisse für das Testbild „Roof“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	22.02	24.65	28.16	32.76	35.89	38.27
SPIHT-Codec-II pp, 8×8 DCT	21.26	24.01	27.45	31.90	35.33	37.79
AGU-Codec-II pp, 16×16 DCT	22.54	25.32	28.91	33.57	36.78	39.27
SPIHT-Codec-II pp, 16×16 DCT	21.99	24.88	28.26	32.97	36.15	38.81
AGU-Codec-II pp, 32×32 DCT	23.43	26.18	29.54	33.97	37.08	39.45
SPIHT-Codec-II pp, 32×32 DCT	22.71	25.47	28.87	33.24	36.35	38.92
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	22.65	25.63	29.23	33.92	37.04	39.49
SPIHT-Codec-II pp, 8×16 LCT-2	21.98	24.88	28.45	33.27	36.49	39.08
AGU-Codec-II pp, 16×32 LCT-2	23.56	26.39	30.00	34.58	37.71	40.10
SPIHT-Codec-II pp, 16×32 LCT-2	23.06	25.91	29.49	34.04	37.23	39.72
AGU-Codec-II pp, 32×64 LCT-2	24.35	27.01	30.32	34.63	37.61	39.89
SPIHT-Codec-II pp, 32×64 LCT-2	23.65	26.38	29.69	33.98	36.96	39.41
L-CEB-Codec-II pp, 8×16 TDLT	22.30	25.28	29.00	33.60	36.59	38.86
SPIHT-Codec-II pp, 8×16 TDLT	21.99	24.60	28.25	32.98	36.20	38.50
L-CEB-Codec-II pp, 8×16 LOT Typ II	22.56	25.59	29.14	33.84	36.87	39.21
SPIHT-Codec-II pp, 8×16 LOT Typ II	21.85	24.83	28.37	33.22	36.29	38.79
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	22.78	25.67	29.18	33.60	36.63	38.98
SPIHT-Codec-II pp, 8×24 LCT-3	22.32	24.69	28.27	32.98	36.01	38.56
AGU-Codec-II pp, 16×48 LCT-3	23.49	26.20	29.61	33.98	37.00	39.35
SPIHT-Codec-II pp, 16×48 LCT-3	23.16	25.82	29.13	33.45	36.49	38.87
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	23.08	26.08	29.69	34.25	37.27	39.67
SPIHT-Codec-II pp, 8×32 LCT-4	22.35	25.23	28.83	33.54	36.76	39.38
AGU-Codec-II pp, 16×64 LCT-4	23.77	26.54	30.07	34.48	37.53	39.88
SPIHT-Codec-II pp, 16×64 LCT-4	23.33	26.05	29.59	34.04	37.07	39.34
L-CEB-Codec-II pp, 8×32 GenLOT	22.55	25.57	29.17	33.69	36.89	39.32
SPIHT-Codec-II pp, 8×32 GenLOT	21.92	24.78	28.33	33.02	36.29	38.86
L-CEB-Codec-II pp, 8×32 ELT	23.04	26.01	29.66	34.23	37.28	39.71
SPIHT-Codec-II pp, 8×32 ELT	22.30	25.20	28.86	33.57	36.76	39.30
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	23.04	26.00	29.50	33.91	36.89	39.25
SPIHT-Codec-II pp, 8×48 LCT-6	22.70	25.15	28.66	33.35	36.30	38.75
AGU-Codec-II pp, 16×96 LCT-6	23.61	26.28	29.63	33.91	36.88	39.20
SPIHT-Codec-II pp, 16×96 LCT-6	23.49	26.05	29.20	33.39	36.44	38.68
L-CEB-Codec-II pp, 8×48 GenLOT	22.90	25.84	29.29	33.76	36.77	39.09
SPIHT-Codec-II pp, 8×48 GenLOT	22.25	25.09	28.58	33.19	36.18	38.64
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	21.75	24.31	27.48	31.71	34.60	37.09
AGU-Codec-II pp, 9/7-Wavelet	21.72	24.33	27.44	31.66	34.57	37.22
SPIHT pp, 9/7-Wavelet	21.45	23.95	27.26	31.52	34.80	36.94

Tabelle H.14: Kompressionsergebnisse für das Testbild „Roof“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Clown“



Abbildung H.57: Testbild „Clown“ (512×512 , $H = 7.7389$, $\sigma^2 = 4228.9209$, $\bar{g} = 99.783$, $g_{\min} = 3$, $g_{\max} = 252$).

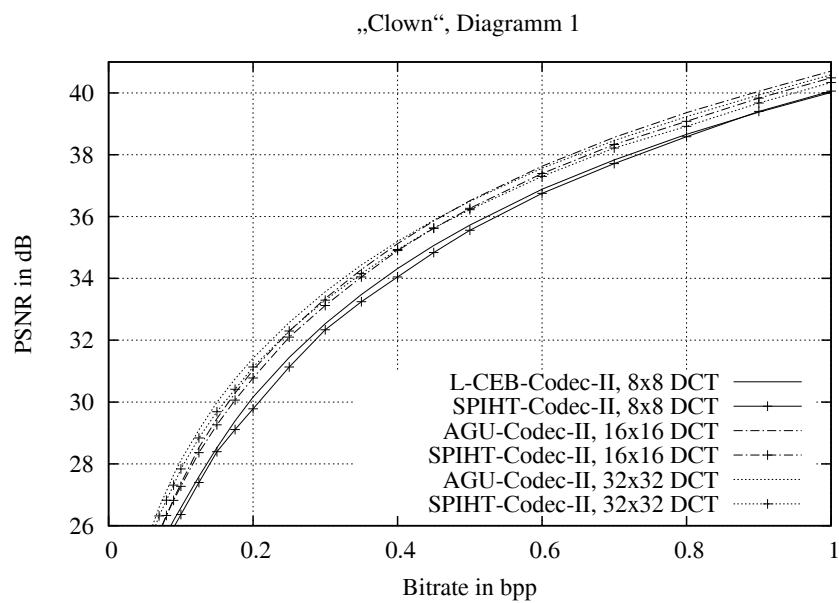


Abbildung H.58: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

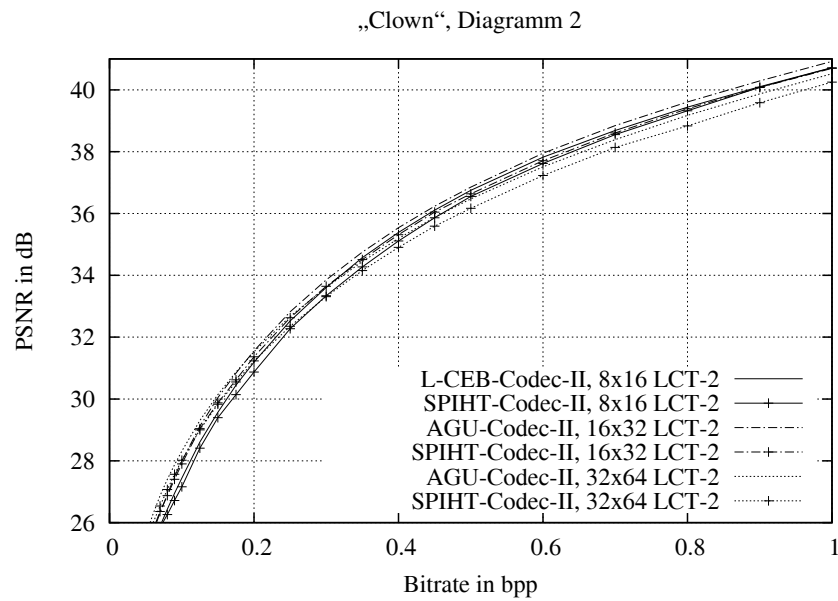


Abbildung H.59: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

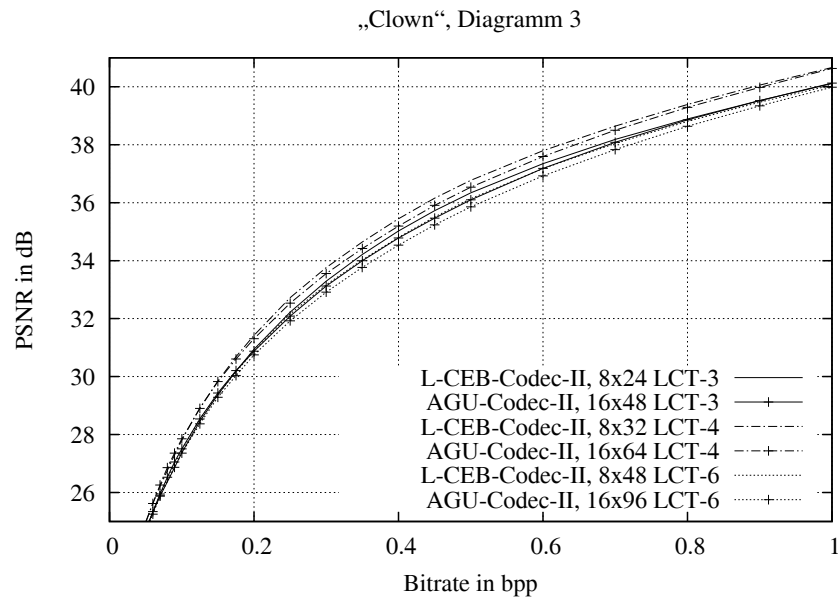


Abbildung H.60: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

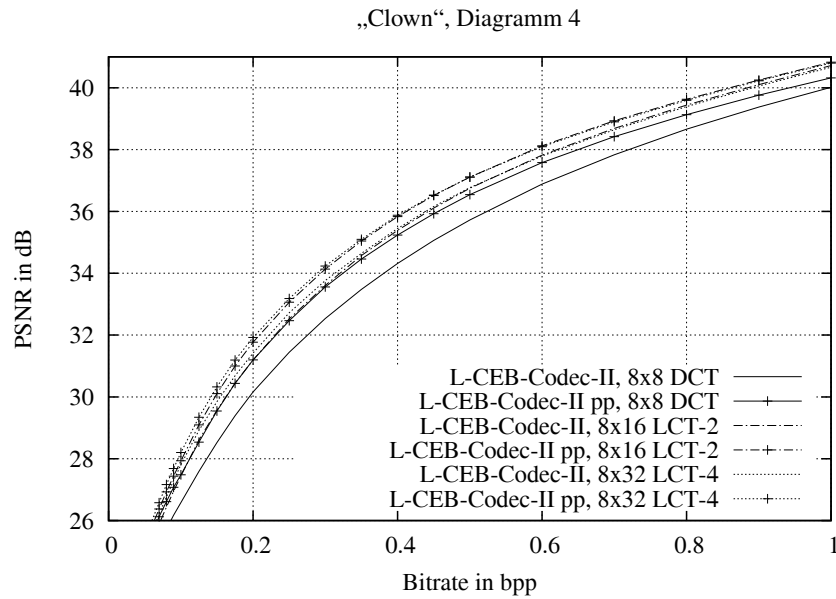


Abbildung H.61: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

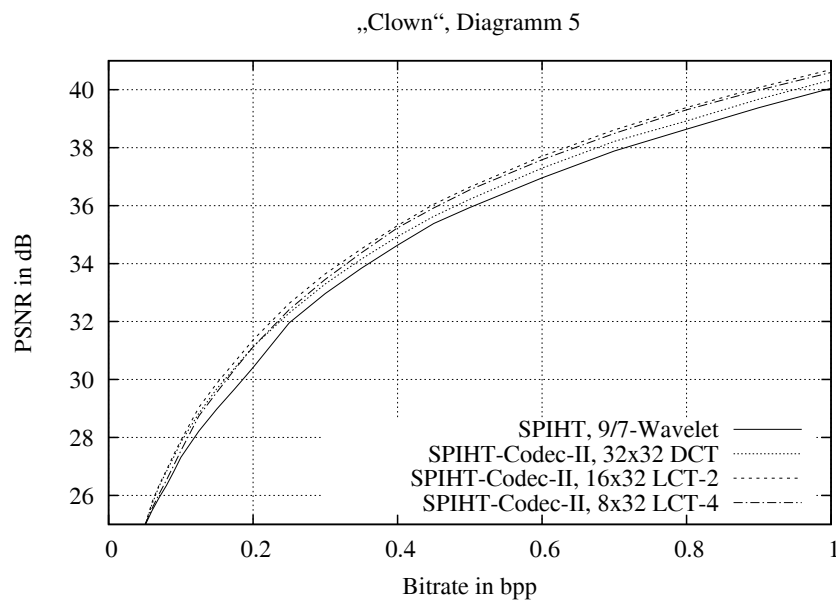


Abbildung H.62: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

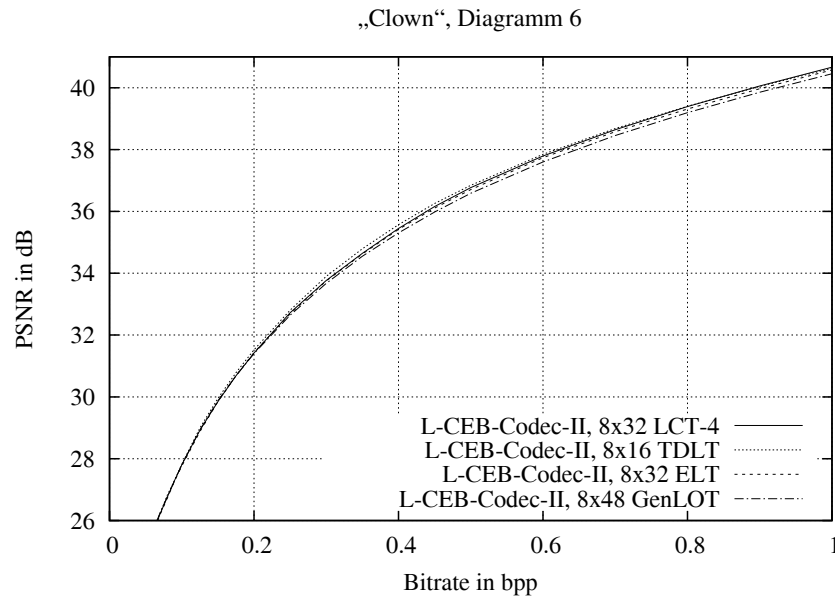


Abbildung H.63: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

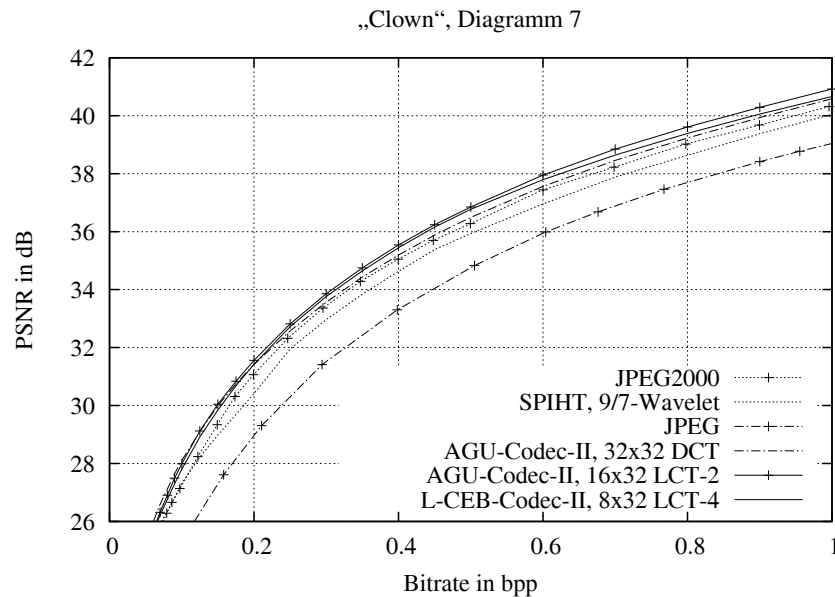


Abbildung H.64: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	24.87	27.59	31.45	35.73	38.26	40.02
SPIHT-Codec-II, 8×8 DCT	24.81	27.40	31.13	35.56	38.18	40.06
AGU-Codec-II, 16×16 DCT	25.32	28.53	32.29	36.52	38.98	40.70
SPIHT-Codec-II, 16×16 DCT	25.39	28.36	32.10	36.27	38.70	40.49
AGU-Codec-II, 32×32 DCT	26.10	29.10	32.58	36.49	38.85	40.58
SPIHT-Codec-II, 32×32 DCT	25.84	28.84	32.30	36.22	38.55	40.33
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	25.47	28.61	32.52	36.75	39.07	40.72
SPIHT-Codec-II, 8×16 LCT-2	25.45	28.41	32.27	36.55	38.89	40.71
AGU-Codec-II, 16×32 LCT-2	25.83	29.12	32.82	36.85	39.24	40.92
SPIHT-Codec-II, 16×32 LCT-2	25.90	29.04	32.63	36.64	38.95	40.70
AGU-Codec-II, 32×64 LCT-2	26.42	29.30	32.67	36.47	38.79	40.52
SPIHT-Codec-II, 32×64 LCT-2	26.13	29.00	32.35	36.17	38.48	40.25
L-CEB-Codec-II, 8×16 TDLT	25.84	28.99	32.81	36.85	39.05	40.63
SPIHT-Codec-II, 8×16 TDLT	25.59	28.57	32.46	36.69	39.01	40.66
L-CEB-Codec-II, 8×16 LOT Typ II	25.50	28.55	32.28	36.30	38.62	40.30
SPIHT-Codec-II, 8×16 LOT Typ II	25.45	28.36	32.05	36.17	38.55	40.32
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	25.46	28.46	32.22	36.34	38.55	40.11
SPIHT-Codec-II, 8×24 LCT-3	25.15	28.13	31.89	36.05	38.51	39.97
AGU-Codec-II, 16×48 LCT-3	25.49	28.54	32.08	36.10	38.50	40.13
SPIHT-Codec-II, 16×48 LCT-3	25.44	28.43	31.91	35.89	38.35	39.86
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	25.76	28.92	32.72	36.77	39.03	40.66
SPIHT-Codec-II, 8×32 LCT-4	25.67	28.74	32.41	36.55	38.94	40.60
AGU-Codec-II, 16×64 LCT-4	25.81	28.90	32.54	36.53	38.91	40.63
SPIHT-Codec-II, 16×64 LCT-4	25.81	28.79	32.33	36.29	38.69	40.40
L-CEB-Codec-II, 8×32 GenLOT	25.65	28.80	32.59	36.75	39.07	40.70
SPIHT-Codec-II, 8×32 GenLOT	25.66	28.63	32.39	36.59	38.90	40.69
L-CEB-Codec-II, 8×32 ELT	25.75	28.87	32.69	36.71	38.95	40.58
SPIHT-Codec-II, 8×32 ELT	25.70	28.71	32.46	36.49	38.76	40.53
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	25.48	28.54	32.16	36.14	38.43	40.06
SPIHT-Codec-II, 8×48 LCT-6	25.40	28.27	31.83	35.86	38.36	39.97
AGU-Codec-II, 16×96 LCT-6	25.43	28.37	31.92	35.85	38.25	39.99
SPIHT-Codec-II, 16×96 LCT-6	25.41	28.24	31.77	35.64	38.12	39.78
L-CEB-Codec-II, 8×48 GenLOT	25.82	28.93	32.64	36.59	38.83	40.45
SPIHT-Codec-II, 8×48 GenLOT	25.83	28.84	32.49	36.43	38.69	40.43
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	25.62	28.15	31.78	35.69	37.97	39.65
AGU-Codec-II, 9/7-Wavelet	25.49	28.07	31.69	35.76	38.11	39.89
SPIHT, 9/7-Wavelet	25.56	28.22	31.95	35.94	38.27	40.05
JPEG2000	25.31	28.24	32.32	36.28	38.62	40.31

Tabelle H.15: Kompressionsergebnisse für das Testbild „Clown“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	25.74	28.54	32.47	36.54	38.80	40.32
SPIHT-Codec-II pp, 8×8 DCT	25.50	28.13	31.98	36.28	38.59	40.19
AGU-Codec-II pp, 16×16 DCT	25.91	29.19	33.00	37.01	39.26	40.84
SPIHT-Codec-II pp, 16×16 DCT	25.89	28.85	32.68	36.58	38.97	40.52
AGU-Codec-II pp, 32×32 DCT	26.55	29.66	33.18	36.99	39.20	40.80
SPIHT-Codec-II pp, 32×32 DCT	26.20	29.25	32.69	36.52	38.86	40.43
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	25.95	29.09	33.06	37.11	39.29	40.83
SPIHT-Codec-II pp, 8×16 LCT-2	25.78	28.73	32.67	36.75	39.04	40.68
AGU-Codec-II pp, 16×32 LCT-2	26.15	29.49	33.26	37.20	39.44	41.03
SPIHT-Codec-II pp, 16×32 LCT-2	26.13	29.31	32.89	36.80	39.12	40.67
AGU-Codec-II pp, 32×64 LCT-2	26.72	29.74	33.22	36.97	39.15	40.78
SPIHT-Codec-II pp, 32×64 LCT-2	26.35	29.31	32.68	36.43	38.79	40.34
L-CEB-Codec-II pp, 8×16 TDLT	26.04	29.26	33.16	37.12	39.22	40.69
SPIHT-Codec-II pp, 8×16 TDLT	25.70	28.73	32.67	36.91	39.06	40.56
L-CEB-Codec-II pp, 8×16 LOT Typ II	26.03	29.15	33.05	36.93	39.06	40.59
SPIHT-Codec-II pp, 8×16 LOT Typ II	25.89	28.75	32.64	36.55	38.87	40.43
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	25.93	29.02	32.85	36.85	38.96	40.44
SPIHT-Codec-II pp, 8×24 LCT-3	25.50	28.50	32.33	36.43	38.71	40.17
AGU-Codec-II pp, 16×48 LCT-3	25.92	29.08	32.76	36.72	38.98	40.51
SPIHT-Codec-II pp, 16×48 LCT-3	25.73	28.78	32.37	36.33	38.58	40.13
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	26.13	29.34	33.18	37.11	39.26	40.80
SPIHT-Codec-II pp, 8×32 LCT-4	25.88	29.02	32.70	36.73	38.99	40.60
AGU-Codec-II pp, 16×64 LCT-4	26.12	29.35	33.08	36.99	39.23	40.84
SPIHT-Codec-II pp, 16×64 LCT-4	26.02	29.09	32.67	36.55	38.81	40.47
L-CEB-Codec-II pp, 8×32 GenLOT	26.09	29.29	33.17	37.15	39.32	40.83
SPIHT-Codec-II pp, 8×32 GenLOT	25.99	28.90	32.80	36.83	39.08	40.68
L-CEB-Codec-II pp, 8×32 ELT	26.13	29.32	33.21	37.13	39.25	40.77
SPIHT-Codec-II pp, 8×32 ELT	25.98	28.96	32.83	36.72	39.00	40.55
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	25.96	29.12	32.87	36.74	38.89	40.42
SPIHT-Codec-II pp, 8×48 LCT-6	25.71	28.68	32.31	36.28	38.57	40.18
AGU-Codec-II pp, 16×96 LCT-6	25.88	28.99	32.68	36.56	38.80	40.42
SPIHT-Codec-II pp, 16×96 LCT-6	25.73	28.67	32.28	36.15	38.41	40.08
L-CEB-Codec-II pp, 8×48 GenLOT	26.20	29.38	33.23	37.09	39.20	40.70
SPIHT-Codec-II pp, 8×48 GenLOT	26.07	29.06	32.88	36.72	38.96	40.49
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	25.87	28.52	32.28	36.14	38.31	39.91
AGU-Codec-II pp, 9/7-Wavelet	25.71	28.36	32.13	36.11	38.38	40.07
SPIHT pp, 9/7-Wavelet	25.71	28.44	32.24	36.14	38.52	40.05

Tabelle H.16: Kompressionsergebnisse für das Testbild „Clown“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Houses“



Abbildung H.65: Testbild „Houses“ (512×512 , $H = 7.6548$, $\sigma^2 = 4079.792$, $\bar{g} = 127.8214$, $g_{min} = 1$, $g_{max} = 255$).

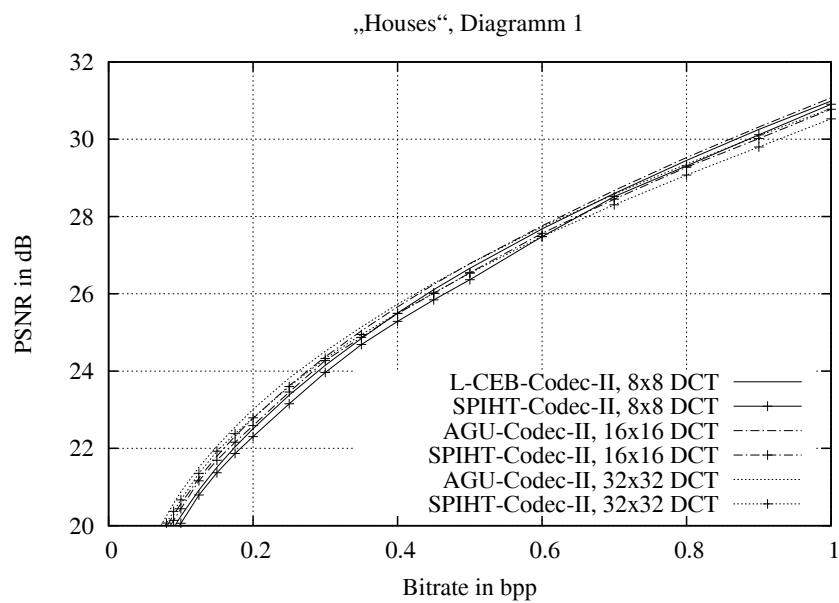


Abbildung H.66: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

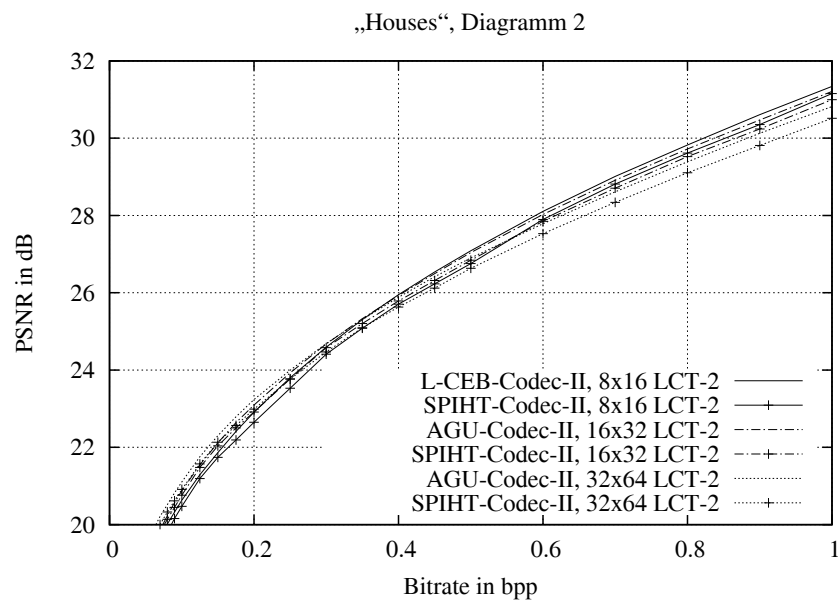


Abbildung H.67: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

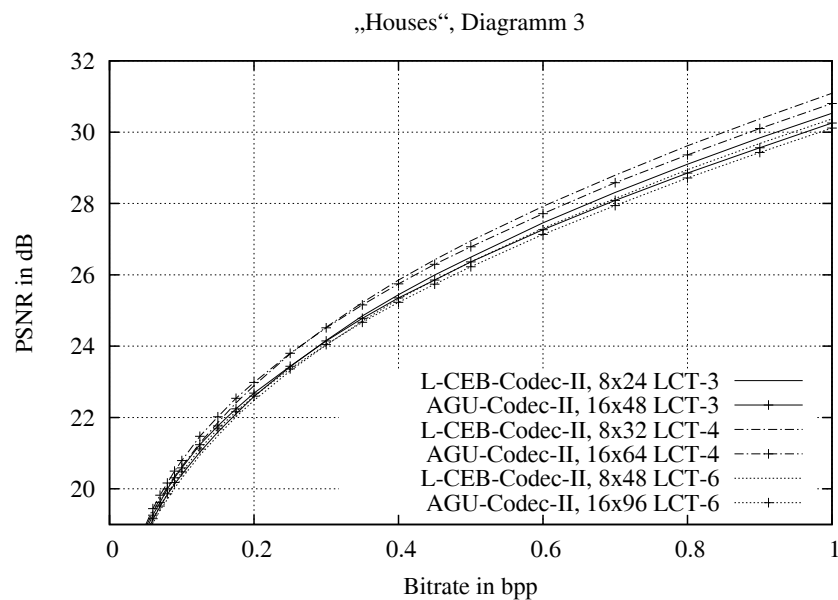


Abbildung H.68: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

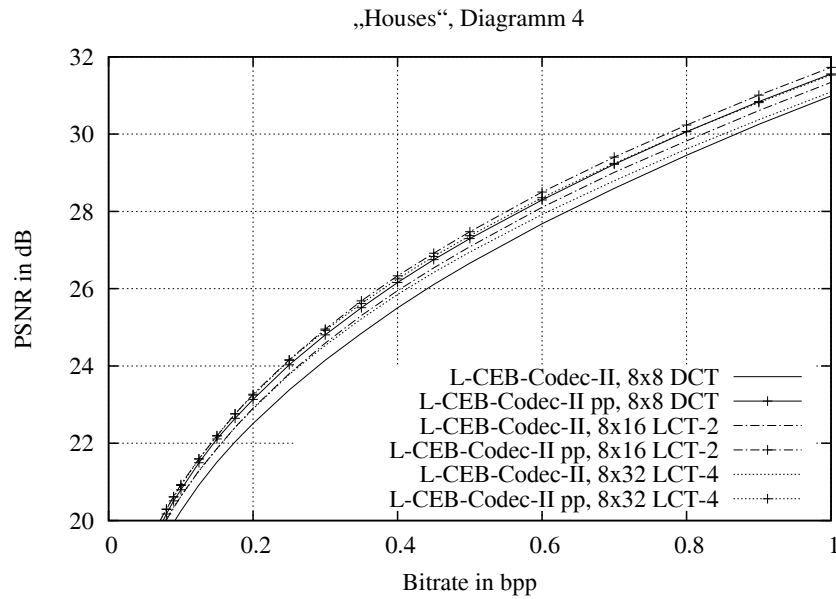


Abbildung H.69: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

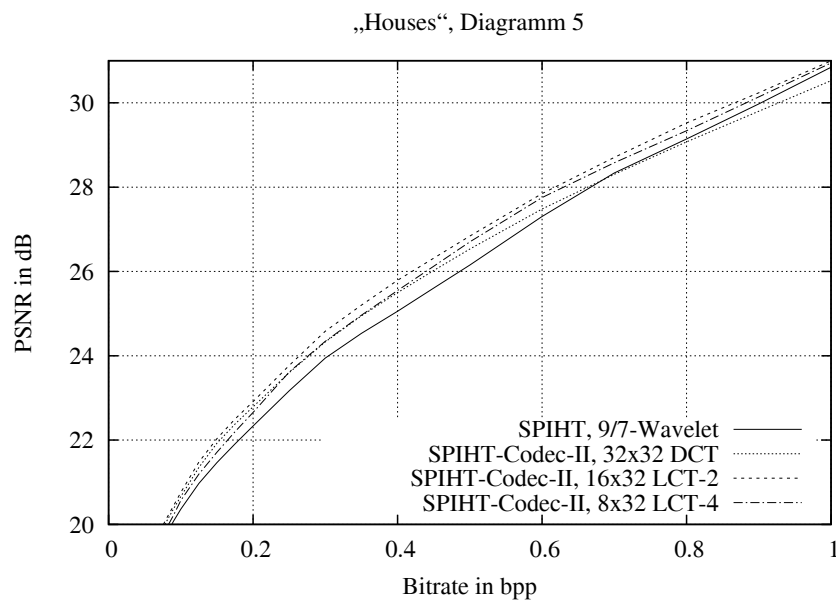


Abbildung H.70: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

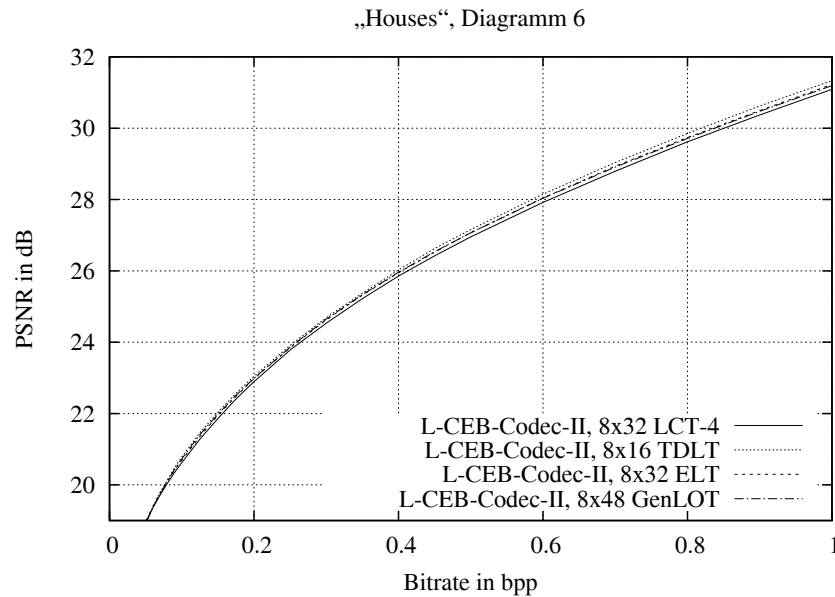


Abbildung H.71: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

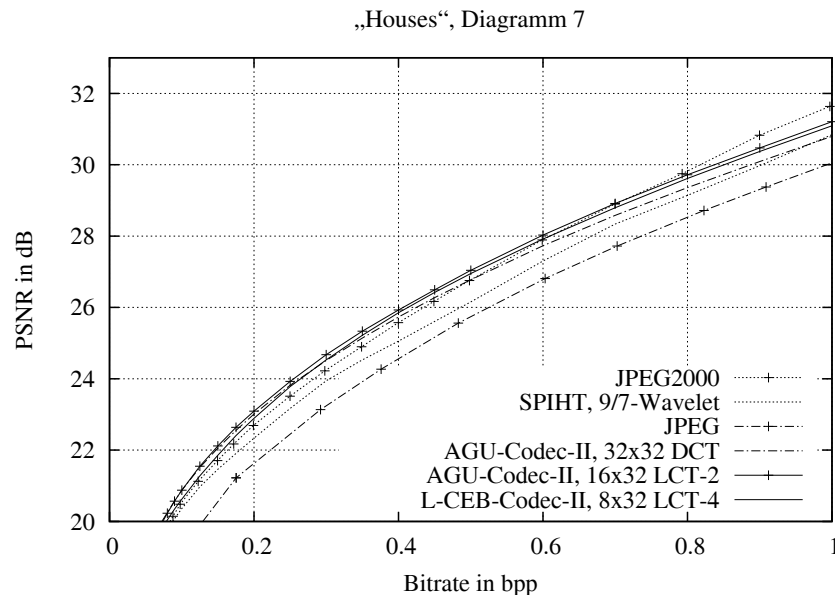


Abbildung H.72: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	19.01	20.92	23.38	26.66	29.03	30.99
SPIHT-Codec-II, 8×8 DCT	18.89	20.80	23.15	26.37	28.90	30.90
AGU-Codec-II, 16×16 DCT	19.33	21.23	23.62	26.78	29.11	31.07
SPIHT-Codec-II, 16×16 DCT	19.25	21.16	23.46	26.55	28.85	30.78
AGU-Codec-II, 32×32 DCT	19.63	21.51	23.82	26.78	28.98	30.80
SPIHT-Codec-II, 32×32 DCT	19.47	21.35	23.60	26.53	28.69	30.53
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	19.38	21.29	23.80	27.09	29.43	31.34
SPIHT-Codec-II, 8×16 LCT-2	19.23	21.19	23.53	26.76	29.19	31.15
AGU-Codec-II, 16×32 LCT-2	19.59	21.55	23.93	27.04	29.33	31.21
SPIHT-Codec-II, 16×32 LCT-2	19.53	21.48	23.77	26.84	29.11	31.00
AGU-Codec-II, 32×64 LCT-2	19.93	21.76	23.99	26.90	29.01	30.82
SPIHT-Codec-II, 32×64 LCT-2	19.76	21.57	23.76	26.63	28.73	30.51
L-CEB-Codec-II, 8×16 TDLT	19.46	21.46	23.91	27.16	29.46	31.33
SPIHT-Codec-II, 8×16 TDLT	19.40	21.22	23.57	26.89	29.36	31.17
L-CEB-Codec-II, 8×16 LOT Typ II	19.35	21.26	23.72	27.00	29.32	31.21
SPIHT-Codec-II, 8×16 LOT Typ II	19.21	21.16	23.46	26.66	29.13	31.07
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	19.23	21.03	23.42	26.50	28.71	30.53
SPIHT-Codec-II, 8×24 LCT-3	19.22	20.90	23.26	26.35	28.48	30.43
AGU-Codec-II, 16×48 LCT-3	19.36	21.24	23.44	26.36	28.49	30.26
SPIHT-Codec-II, 16×48 LCT-3	19.35	21.09	23.40	26.31	28.32	30.12
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	19.44	21.28	23.78	26.95	29.20	31.09
SPIHT-Codec-II, 8×32 LCT-4	19.26	21.19	23.62	26.70	28.97	30.95
AGU-Codec-II, 16×64 LCT-4	19.55	21.47	23.80	26.79	28.98	30.80
SPIHT-Codec-II, 16×64 LCT-4	19.47	21.43	23.78	26.65	28.78	30.64
L-CEB-Codec-II, 8×32 GenLOT	19.43	21.38	23.88	27.13	29.45	31.35
SPIHT-Codec-II, 8×32 GenLOT	19.29	21.28	23.58	26.78	29.23	31.19
L-CEB-Codec-II, 8×32 ELT	19.48	21.36	23.85	27.07	29.34	31.22
SPIHT-Codec-II, 8×32 ELT	19.34	21.27	23.58	26.74	29.11	31.00
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	19.13	20.94	23.31	26.36	28.54	30.37
SPIHT-Codec-II, 8×48 LCT-6	19.07	20.87	23.18	26.23	28.34	30.27
AGU-Codec-II, 16×96 LCT-6	19.25	21.12	23.36	26.23	28.34	30.12
SPIHT-Codec-II, 16×96 LCT-6	19.26	21.04	23.34	26.17	28.19	29.99
L-CEB-Codec-II, 8×48 GenLOT	19.46	21.41	23.86	27.07	29.32	31.19
SPIHT-Codec-II, 8×48 GenLOT	19.35	21.33	23.61	26.74	29.11	31.02
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	19.24	20.95	23.11	26.10	28.49	30.57
AGU-Codec-II, 9/7-Wavelet	19.13	20.86	22.98	25.99	28.53	30.70
SPIHT, 9/7-Wavelet	19.17	20.97	23.17	26.15	28.72	30.84
JPEG2000	19.23	21.12	23.51	26.76	29.33	31.64

Tabelle H.17: Kompressionsergebnisse für das Testbild „Houses“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	19.53	21.50	24.03	27.30	29.65	31.56
SPIHT-Codec-II pp, 8×8 DCT	19.28	21.16	23.60	26.83	29.32	31.30
AGU-Codec-II pp, 16×16 DCT	19.66	21.61	24.05	27.27	29.61	31.54
SPIHT-Codec-II pp, 16×16 DCT	19.53	21.51	23.76	26.88	29.16	31.11
AGU-Codec-II pp, 32×32 DCT	19.86	21.84	24.24	27.28	29.52	31.34
SPIHT-Codec-II pp, 32×32 DCT	19.66	21.64	23.90	26.88	29.01	30.90
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	19.64	21.60	24.16	27.48	29.82	31.73
SPIHT-Codec-II pp, 8×16 LCT-2	19.39	21.41	23.73	26.99	29.41	31.39
AGU-Codec-II pp, 16×32 LCT-2	19.75	21.76	24.23	27.43	29.75	31.64
SPIHT-Codec-II pp, 16×32 LCT-2	19.64	21.66	23.97	27.10	29.35	31.29
AGU-Codec-II pp, 32×64 LCT-2	20.06	22.01	24.39	27.42	29.58	31.39
SPIHT-Codec-II pp, 32×64 LCT-2	19.84	21.78	24.04	26.99	29.05	30.92
L-CEB-Codec-II pp, 8×16 TDLT	19.57	21.61	24.13	27.44	29.76	31.63
SPIHT-Codec-II pp, 8×16 TDLT	19.48	21.27	23.69	27.04	29.58	31.30
L-CEB-Codec-II pp, 8×16 LOT Typ II	19.62	21.60	24.13	27.45	29.80	31.68
SPIHT-Codec-II pp, 8×16 LOT Typ II	19.39	21.28	23.71	26.95	29.43	31.37
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	19.50	21.39	23.90	27.05	29.31	31.13
SPIHT-Codec-II pp, 8×24 LCT-3	19.41	21.07	23.53	26.66	28.88	30.75
AGU-Codec-II pp, 16×48 LCT-3	19.58	21.56	23.88	26.97	29.18	30.99
SPIHT-Codec-II pp, 16×48 LCT-3	19.54	21.28	23.68	26.69	28.83	30.50
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	19.68	21.59	24.15	27.37	29.65	31.53
SPIHT-Codec-II pp, 8×32 LCT-4	19.37	21.38	23.84	26.98	29.22	31.27
AGU-Codec-II pp, 16×64 LCT-4	19.70	21.72	24.15	27.28	29.52	31.37
SPIHT-Codec-II pp, 16×64 LCT-4	19.55	21.62	24.04	27.03	29.12	31.10
L-CEB-Codec-II pp, 8×32 GenLOT	19.69	21.69	24.24	27.53	29.88	31.76
SPIHT-Codec-II pp, 8×32 GenLOT	19.44	21.53	23.78	27.03	29.49	31.45
L-CEB-Codec-II pp, 8×32 ELT	19.69	21.63	24.20	27.46	29.77	31.65
SPIHT-Codec-II pp, 8×32 ELT	19.45	21.47	23.77	26.99	29.35	31.28
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	19.46	21.39	23.83	26.99	29.22	31.05
SPIHT-Codec-II pp, 8×48 LCT-6	19.25	21.10	23.53	26.59	28.81	30.61
AGU-Codec-II pp, 16×96 LCT-6	19.49	21.49	23.89	26.92	29.11	30.90
SPIHT-Codec-II pp, 16×96 LCT-6	19.38	21.27	23.71	26.61	28.77	30.41
L-CEB-Codec-II pp, 8×48 GenLOT	19.70	21.69	24.22	27.50	29.79	31.66
SPIHT-Codec-II pp, 8×48 GenLOT	19.45	21.54	23.82	27.01	29.38	31.32
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	19.40	21.16	23.45	26.54	28.95	31.00
AGU-Codec-II pp, 9/7-Wavelet	19.23	21.02	23.19	26.32	28.92	31.09
SPIHT pp, 9/7-Wavelet	19.23	21.11	23.37	26.38	28.96	31.07

Tabelle H.18: Kompressionsergebnisse für das Testbild „Houses“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Mandrill“

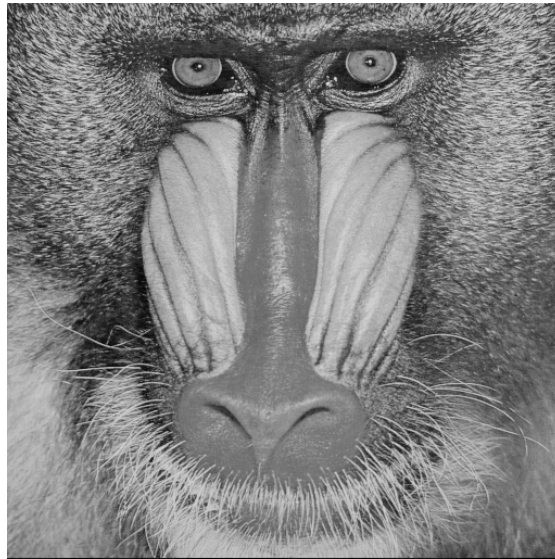


Abbildung H.73: Testbild „Mandrill“ (512×512 , $H = 7.3579$, $\sigma^2 = 1789.3828$, $\bar{g} = 129.147$, $g_{min} = 0$, $g_{max} = 230$).

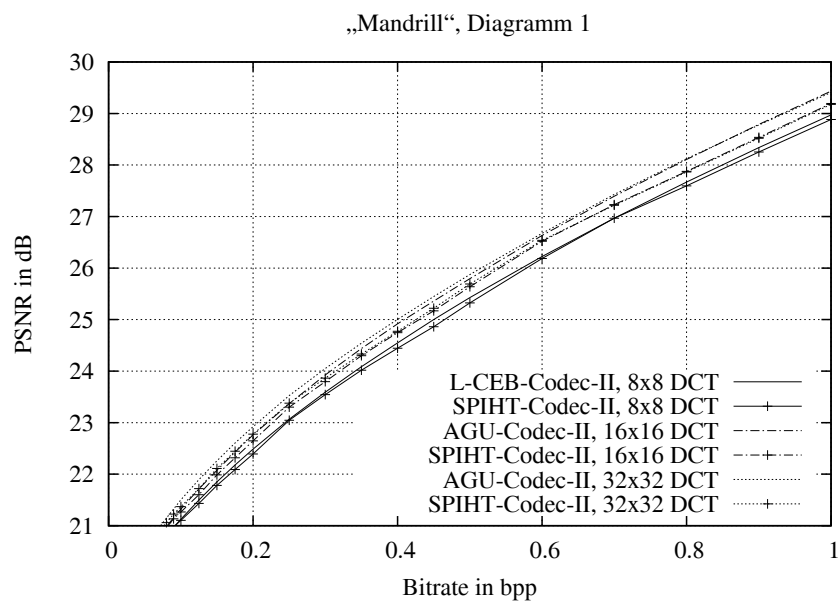


Abbildung H.74: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

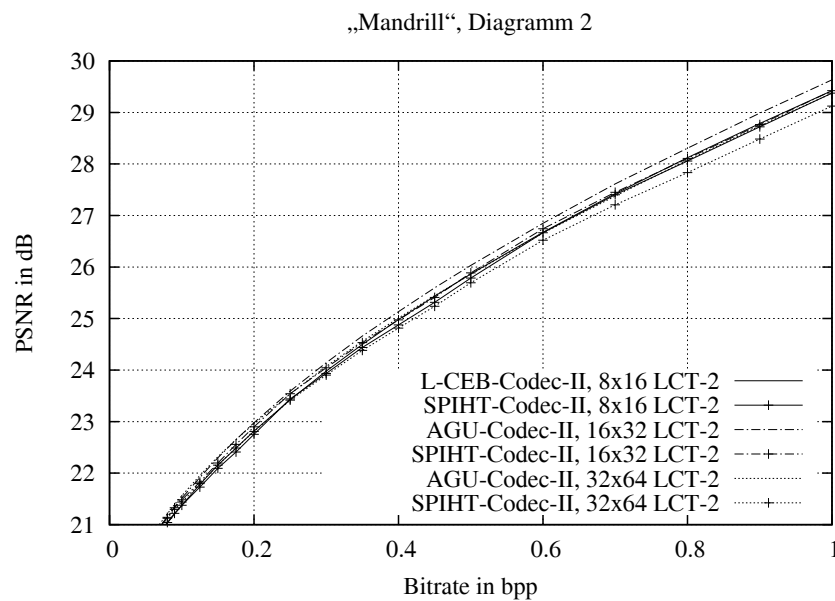


Abbildung H.75: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

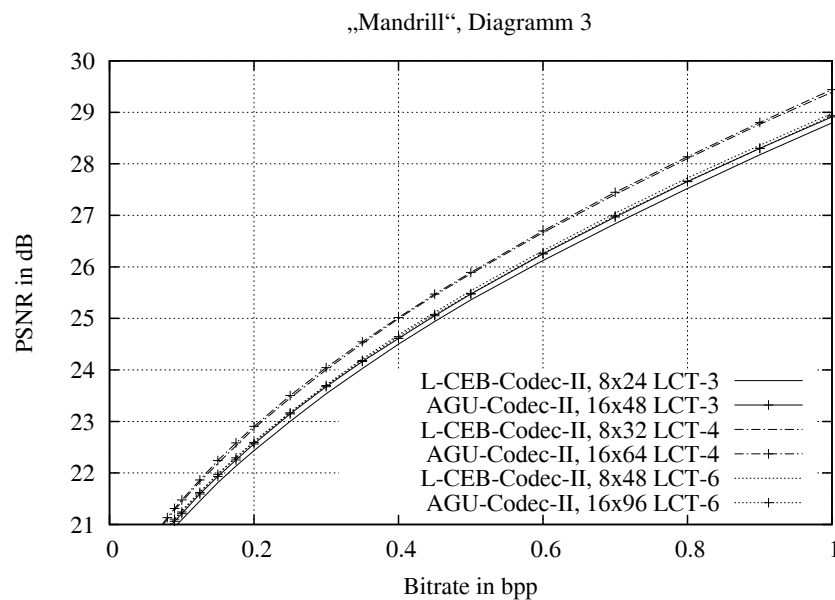


Abbildung H.76: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

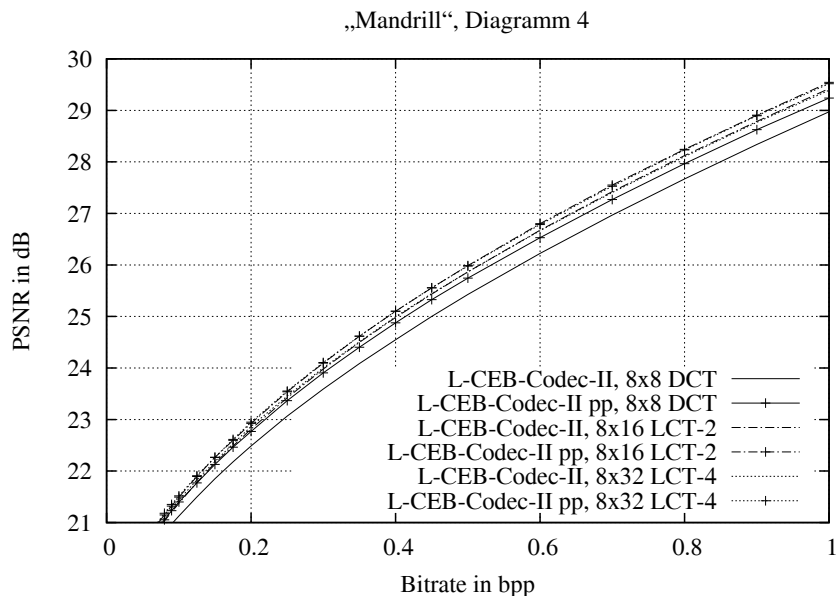


Abbildung H.77: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

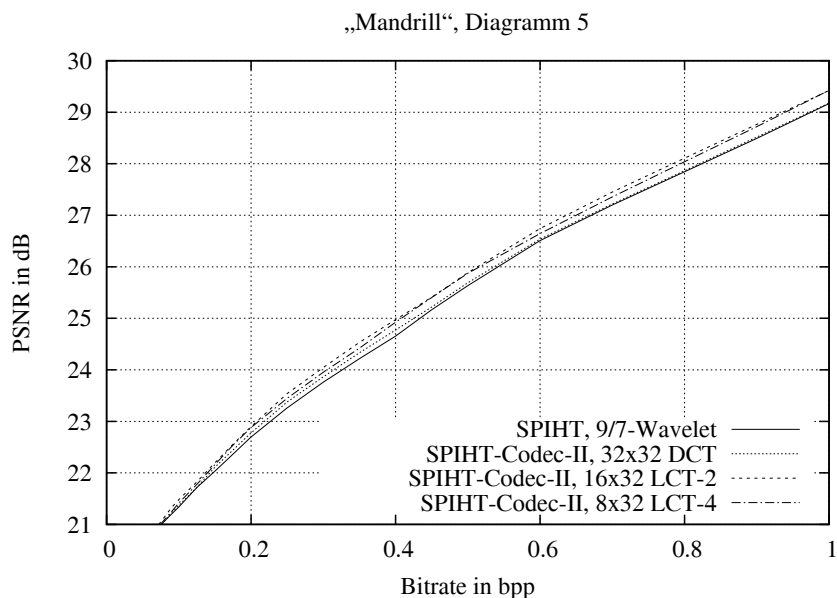


Abbildung H.78: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

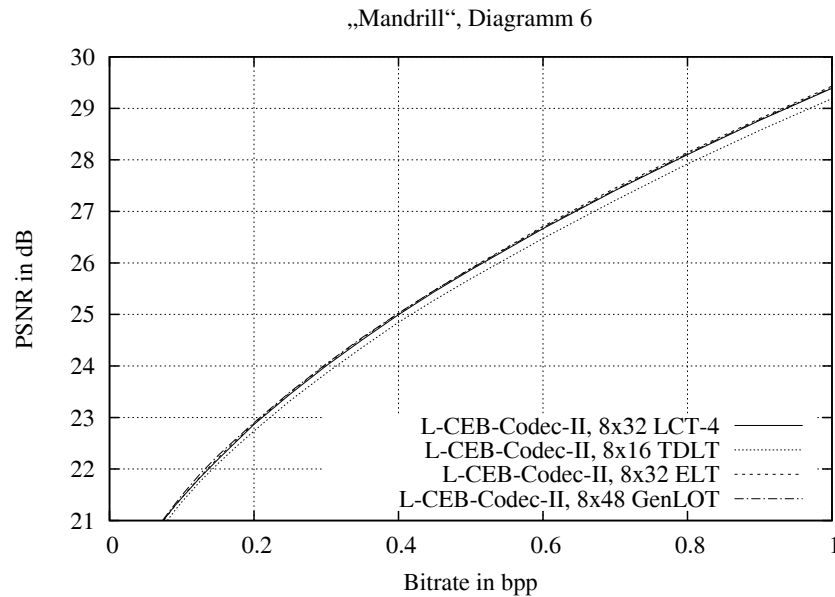


Abbildung H.79: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

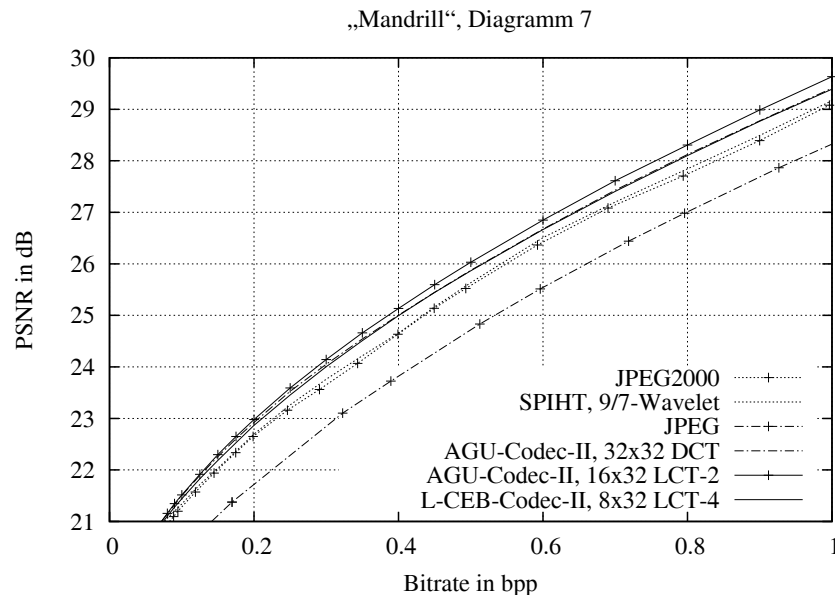


Abbildung H.80: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	20.51	21.50	23.07	25.43	27.32	28.97
SPIHT-Codec-II, 8×8 DCT	20.52	21.43	23.05	25.32	27.28	28.88
AGU-Codec-II, 16×16 DCT	20.65	21.71	23.37	25.79	27.76	29.43
SPIHT-Codec-II, 16×16 DCT	20.63	21.60	23.30	25.64	27.55	29.20
AGU-Codec-II, 32×32 DCT	20.82	21.89	23.53	25.88	27.78	29.41
SPIHT-Codec-II, 32×32 DCT	20.75	21.72	23.38	25.69	27.54	29.18
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	20.71	21.78	23.42	25.87	27.78	29.42
SPIHT-Codec-II, 8×16 LCT-2	20.71	21.73	23.44	25.79	27.73	29.37
AGU-Codec-II, 16×32 LCT-2	20.81	21.91	23.59	26.03	27.96	29.63
SPIHT-Codec-II, 16×32 LCT-2	20.79	21.83	23.54	25.89	27.77	29.42
AGU-Codec-II, 32×64 LCT-2	20.86	21.95	23.54	25.86	27.73	29.37
SPIHT-Codec-II, 32×64 LCT-2	20.80	21.79	23.42	25.69	27.51	29.12
L-CEB-Codec-II, 8×16 TDLT	20.65	21.75	23.33	25.69	27.57	29.19
SPIHT-Codec-II, 8×16 TDLT	20.60	21.77	23.26	25.60	27.64	29.12
L-CEB-Codec-II, 8×16 LOT Typ II	20.69	21.78	23.40	25.81	27.71	29.36
SPIHT-Codec-II, 8×16 LOT Typ II	20.71	21.69	23.42	25.74	27.66	29.31
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	20.48	21.46	23.00	25.36	27.19	28.80
SPIHT-Codec-II, 8×24 LCT-3	20.52	21.55	23.10	25.44	27.28	28.88
AGU-Codec-II, 16×48 LCT-3	20.59	21.59	23.15	25.47	27.32	28.91
SPIHT-Codec-II, 16×48 LCT-3	20.55	21.50	23.10	25.40	27.24	28.78
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	20.77	21.83	23.45	25.86	27.76	29.39
SPIHT-Codec-II, 8×32 LCT-4	20.80	21.78	23.45	25.87	27.70	29.43
AGU-Codec-II, 16×64 LCT-4	20.79	21.86	23.50	25.89	27.80	29.44
SPIHT-Codec-II, 16×64 LCT-4	20.80	21.80	23.43	25.81	27.61	29.32
L-CEB-Codec-II, 8×32 GenLOT	20.72	21.81	23.42	25.85	27.77	29.40
SPIHT-Codec-II, 8×32 GenLOT	20.74	21.71	23.43	25.75	27.68	29.34
L-CEB-Codec-II, 8×32 ELT	20.78	21.85	23.49	25.90	27.81	29.43
SPIHT-Codec-II, 8×32 ELT	20.80	21.79	23.50	25.85	27.75	29.40
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	20.64	21.63	23.18	25.53	27.39	28.98
SPIHT-Codec-II, 8×48 LCT-6	20.68	21.69	23.20	25.56	27.41	29.04
AGU-Codec-II, 16×96 LCT-6	20.63	21.62	23.17	25.48	27.33	28.93
SPIHT-Codec-II, 16×96 LCT-6	20.66	21.60	23.11	25.40	27.21	28.83
L-CEB-Codec-II, 8×48 GenLOT	20.79	21.90	23.50	25.89	27.78	29.40
SPIHT-Codec-II, 8×48 GenLOT	20.81	21.81	23.52	25.84	27.73	29.37
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	20.71	21.68	23.18	25.51	27.42	29.05
AGU-Codec-II, 9/7-Wavelet	20.72	21.71	23.25	25.65	27.56	29.22
SPIHT, 9/7-Wavelet	20.74	21.72	23.26	25.64	27.52	29.17
JPEG2000	20.60	21.57	23.16	25.52	27.43	29.08

Tabelle H.19: Kompressionsergebnisse für das Testbild „Mandrill“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	20.74	21.77	23.36	25.74	27.62	29.24
SPIHT-Codec-II pp, 8×8 DCT	20.72	21.58	23.14	25.50	27.49	29.06
AGU-Codec-II pp, 16×16 DCT	20.78	21.86	23.54	25.98	27.93	29.60
SPIHT-Codec-II pp, 16×16 DCT	20.77	21.68	23.46	25.70	27.68	29.29
AGU-Codec-II pp, 32×32 DCT	20.89	21.97	23.62	26.02	27.94	29.57
SPIHT-Codec-II pp, 32×32 DCT	20.83	21.73	23.48	25.71	27.65	29.25
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	20.82	21.89	23.54	25.99	27.90	29.54
SPIHT-Codec-II pp, 8×16 LCT-2	20.73	21.76	23.37	25.80	27.80	29.41
AGU-Codec-II pp, 16×32 LCT-2	20.87	21.96	23.65	26.13	28.07	29.75
SPIHT-Codec-II pp, 16×32 LCT-2	20.78	21.82	23.60	25.88	27.83	29.46
AGU-Codec-II pp, 32×64 LCT-2	20.90	22.00	23.62	26.00	27.89	29.54
SPIHT-Codec-II pp, 32×64 LCT-2	20.84	21.79	23.50	25.71	27.61	29.20
L-CEB-Codec-II pp, 8×16 TDLT	20.71	21.82	23.43	25.82	27.71	29.32
SPIHT-Codec-II pp, 8×16 TDLT	20.58	21.80	23.26	25.65	27.67	29.17
L-CEB-Codec-II pp, 8×16 LOT Typ II	20.81	21.88	23.52	25.94	27.86	29.51
SPIHT-Codec-II pp, 8×16 LOT Typ II	20.81	21.72	23.30	25.74	27.76	29.37
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	20.64	21.66	23.22	25.59	27.43	29.03
SPIHT-Codec-II pp, 8×24 LCT-3	20.56	21.67	23.16	25.56	27.36	28.88
AGU-Codec-II pp, 16×48 LCT-3	20.71	21.72	23.31	25.67	27.55	29.14
SPIHT-Codec-II pp, 16×48 LCT-3	20.57	21.55	23.15	25.52	27.33	28.85
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	20.84	21.91	23.55	25.98	27.89	29.52
SPIHT-Codec-II pp, 8×32 LCT-4	20.83	21.76	23.50	25.85	27.75	29.44
AGU-Codec-II pp, 16×64 LCT-4	20.85	21.93	23.58	26.02	27.94	29.60
SPIHT-Codec-II pp, 16×64 LCT-4	20.83	21.77	23.50	25.81	27.69	29.38
L-CEB-Codec-II pp, 8×32 GenLOT	20.82	21.92	23.54	25.97	27.90	29.54
SPIHT-Codec-II pp, 8×32 GenLOT	20.81	21.74	23.31	25.76	27.76	29.38
L-CEB-Codec-II pp, 8×32 ELT	20.85	21.93	23.58	26.01	27.92	29.55
SPIHT-Codec-II pp, 8×32 ELT	20.85	21.79	23.57	25.85	27.82	29.43
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	20.77	21.78	23.35	25.75	27.62	29.20
SPIHT-Codec-II pp, 8×48 LCT-6	20.71	21.69	23.26	25.69	27.50	29.15
AGU-Codec-II pp, 16×96 LCT-6	20.75	21.75	23.36	25.73	27.61	29.22
SPIHT-Codec-II pp, 16×96 LCT-6	20.69	21.62	23.20	25.56	27.36	28.92
L-CEB-Codec-II pp, 8×48 GenLOT	20.87	21.95	23.58	26.02	27.93	29.55
SPIHT-Codec-II pp, 8×48 GenLOT	20.87	21.81	23.59	25.85	27.81	29.43
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	20.74	21.73	23.25	25.62	27.53	29.17
AGU-Codec-II pp, 9/7-Wavelet	20.71	21.73	23.30	25.72	27.65	29.32
SPIHT pp, 9/7-Wavelet	20.75	21.67	23.33	25.62	27.59	29.20

Tabelle H.20: Kompressionsergebnisse für das Testbild „Mandrill“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Fingerprint“



Abbildung H.81: Testbild „Fingerprint“ (512×512 , $H = 7.0309$, $\sigma^2 = 2904.4883$, $\bar{g} = 98.9244$, $g_{min} = 5$, $g_{max} = 255$).

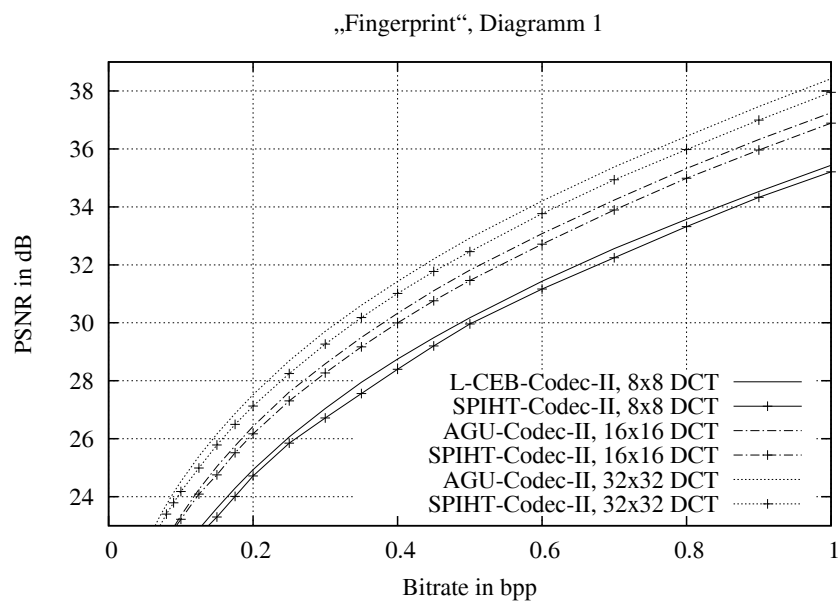


Abbildung H.82: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

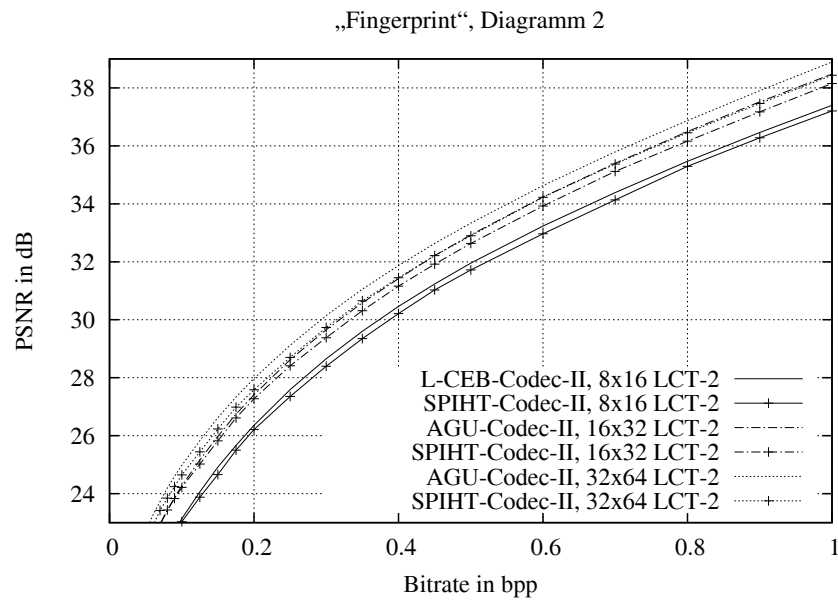


Abbildung H.83: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

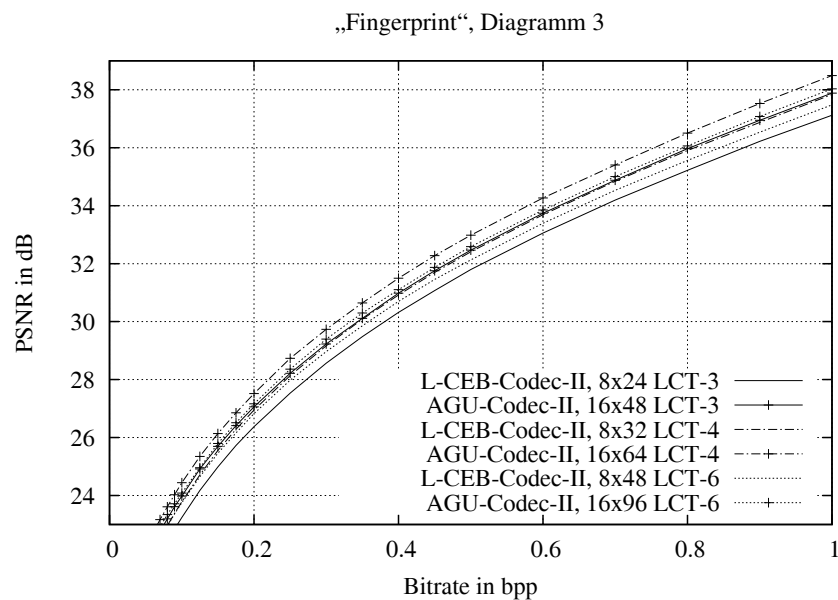


Abbildung H.84: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

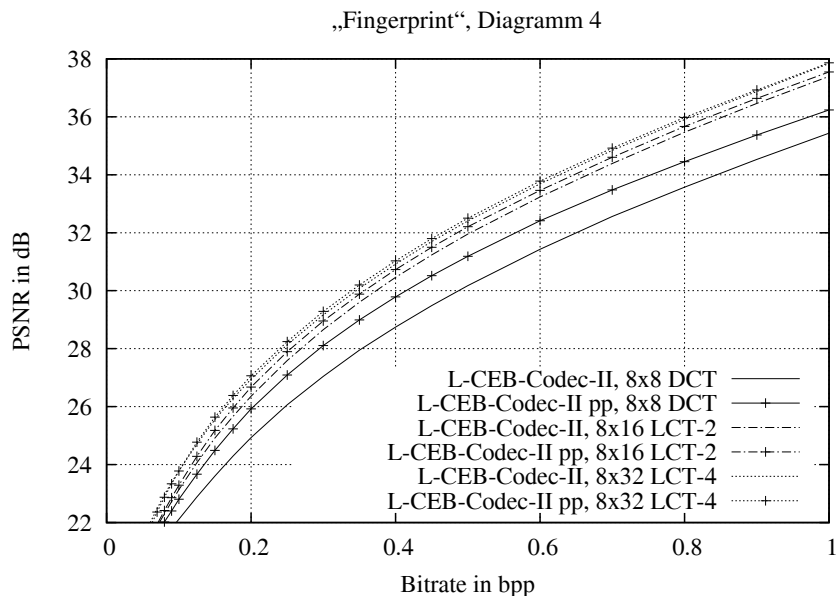


Abbildung H.85: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

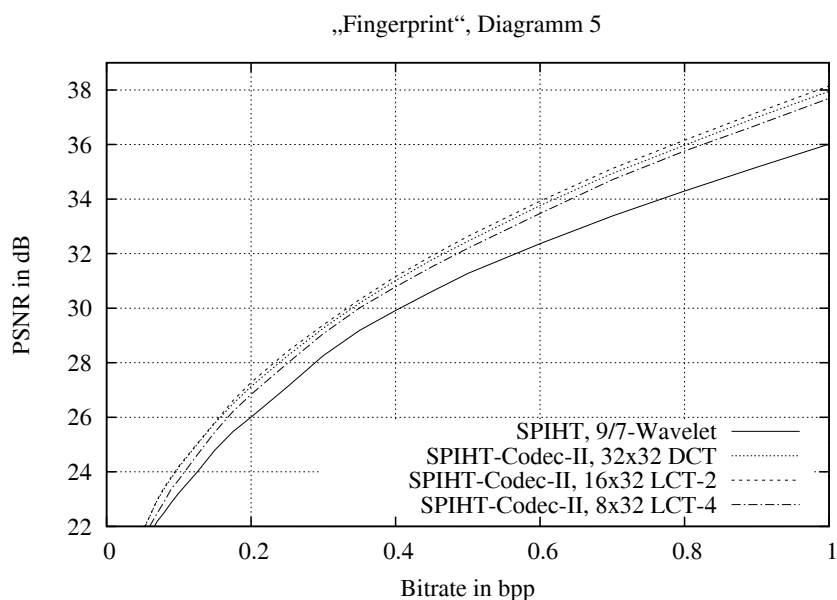


Abbildung H.86: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

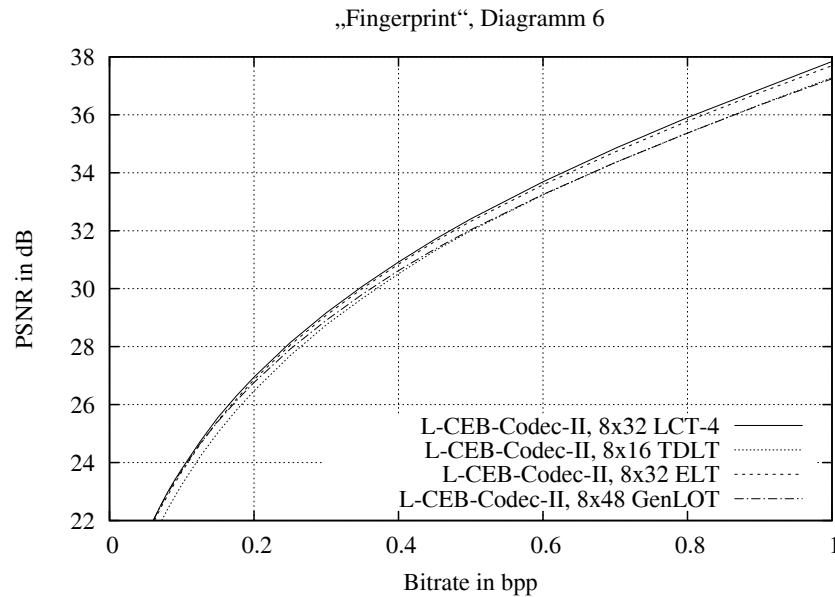


Abbildung H.87: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

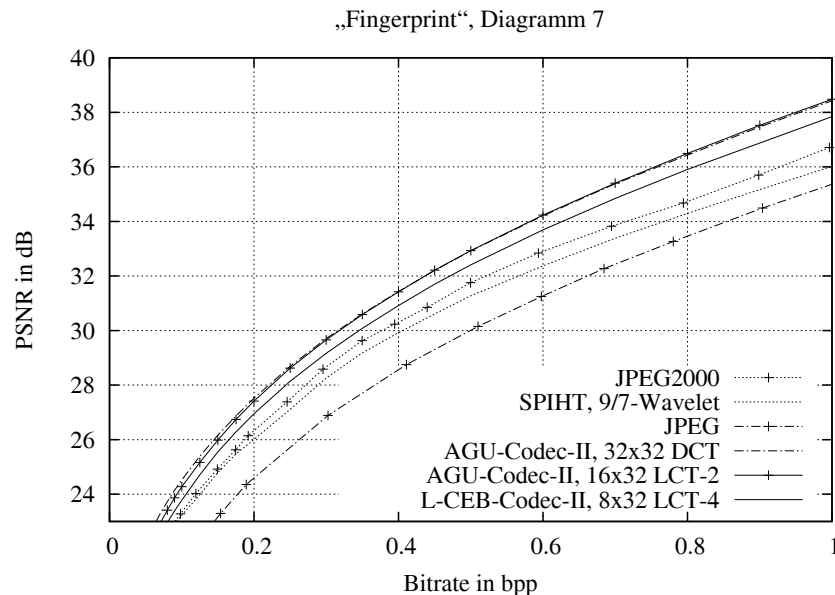


Abbildung H.88: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	20.85	22.88	26.06	30.18	33.08	35.44
SPIHT-Codec-II, 8×8 DCT	20.73	22.66	25.85	29.96	32.80	35.21
AGU-Codec-II, 16×16 DCT	21.71	24.25	27.61	31.82	34.79	37.25
SPIHT-Codec-II, 16×16 DCT	21.57	24.09	27.30	31.46	34.43	36.89
AGU-Codec-II, 32×32 DCT	22.90	25.39	28.69	32.93	35.91	38.42
SPIHT-Codec-II, 32×32 DCT	22.49	24.99	28.25	32.46	35.46	37.95
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	21.53	24.06	27.59	31.96	34.94	37.40
SPIHT-Codec-II, 8×16 LCT-2	21.38	23.88	27.36	31.72	34.73	37.21
AGU-Codec-II, 16×32 LCT-2	22.54	25.17	28.62	32.93	35.96	38.48
SPIHT-Codec-II, 16×32 LCT-2	22.51	25.02	28.41	32.63	35.64	38.15
AGU-Codec-II, 32×64 LCT-2	23.38	25.83	29.12	33.33	36.35	38.90
SPIHT-Codec-II, 32×64 LCT-2	23.02	25.44	28.70	32.90	35.90	38.44
L-CEB-Codec-II, 8×16 TDLT	21.49	24.19	27.69	32.00	34.87	37.28
SPIHT-Codec-II, 8×16 TDLT	21.47	24.15	27.60	31.87	34.79	37.10
L-CEB-Codec-II, 8×16 LOT Typ II	21.54	24.00	27.39	31.58	34.40	36.74
SPIHT-Codec-II, 8×16 LOT Typ II	21.43	23.84	27.19	31.41	34.24	36.59
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	21.58	24.17	27.53	31.80	34.71	37.12
SPIHT-Codec-II, 8×24 LCT-3	21.86	24.21	27.38	31.55	34.53	37.10
AGU-Codec-II, 16×48 LCT-3	22.41	24.92	28.21	32.47	35.44	37.88
SPIHT-Codec-II, 16×48 LCT-3	22.48	24.89	27.98	32.12	35.12	37.60
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	22.13	24.71	28.14	32.41	35.38	37.84
SPIHT-Codec-II, 8×32 LCT-4	22.16	24.61	27.97	32.20	35.24	37.69
AGU-Codec-II, 16×64 LCT-4	22.82	25.35	28.74	32.98	35.96	38.49
SPIHT-Codec-II, 16×64 LCT-4	22.86	25.20	28.47	32.69	35.70	38.16
L-CEB-Codec-II, 8×32 GenLOT	21.58	23.94	27.33	31.72	34.70	37.12
SPIHT-Codec-II, 8×32 GenLOT	21.43	23.73	27.05	31.40	34.39	36.82
L-CEB-Codec-II, 8×32 ELT	22.01	24.62	28.06	32.33	35.26	37.70
SPIHT-Codec-II, 8×32 ELT	21.95	24.47	27.90	32.13	35.15	37.52
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	22.13	24.65	27.98	32.13	35.06	37.48
SPIHT-Codec-II, 8×48 LCT-6	22.19	24.66	27.84	31.97	34.90	37.43
AGU-Codec-II, 16×96 LCT-6	22.59	24.96	28.36	32.59	35.53	38.03
SPIHT-Codec-II, 16×96 LCT-6	22.65	24.97	28.16	32.32	35.31	37.77
L-CEB-Codec-II, 8×48 GenLOT	22.09	24.64	27.91	32.03	34.88	37.24
SPIHT-Codec-II, 8×48 GenLOT	22.01	24.44	27.72	31.86	34.78	37.09
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	21.80	23.98	27.02	31.02	33.50	35.60
AGU-Codec-II, 9/7-Wavelet	21.79	24.06	27.23	31.32	33.89	36.15
SPIHT, 9/7-Wavelet	21.86	23.98	27.11	31.27	33.88	36.01
JPEG2000	21.64	24.02	27.39	31.75	34.24	36.71

Tabelle H.21: Kompressionsergebnisse für das Testbild „Fingerprint“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	21.30	23.67	27.09	31.19	33.99	36.24
SPIHT-Codec-II pp, 8×8 DCT	20.91	23.42	26.66	30.75	33.53	35.75
AGU-Codec-II pp, 16×16 DCT	21.76	24.60	28.11	32.30	35.20	37.60
SPIHT-Codec-II pp, 16×16 DCT	21.57	24.44	27.75	31.85	34.72	37.14
AGU-Codec-II pp, 32×32 DCT	22.75	25.49	28.90	33.14	36.10	38.58
SPIHT-Codec-II pp, 32×32 DCT	22.20	25.03	28.42	32.62	35.58	38.01
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	21.59	24.29	27.90	32.21	35.15	37.55
SPIHT-Codec-II pp, 8×16 LCT-2	21.33	24.04	27.57	31.87	34.81	37.26
AGU-Codec-II pp, 16×32 LCT-2	22.30	25.13	28.66	32.97	35.98	38.46
SPIHT-Codec-II pp, 16×32 LCT-2	22.23	24.95	28.42	32.63	35.62	38.06
AGU-Codec-II pp, 32×64 LCT-2	23.11	25.73	29.11	33.35	36.35	38.86
SPIHT-Codec-II pp, 32×64 LCT-2	22.62	25.30	28.65	32.86	35.88	38.34
L-CEB-Codec-II pp, 8×16 TDLT	21.31	24.18	27.80	32.15	35.03	37.42
SPIHT-Codec-II pp, 8×16 TDLT	21.42	23.98	27.58	31.92	34.88	37.15
L-CEB-Codec-II pp, 8×16 LOT Typ II	21.56	24.25	27.79	32.07	34.90	37.21
SPIHT-Codec-II pp, 8×16 LOT Typ II	21.35	24.02	27.53	31.77	34.57	36.93
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	21.61	24.34	27.80	32.05	34.94	37.30
SPIHT-Codec-II pp, 8×24 LCT-3	21.81	24.36	27.46	31.64	34.61	37.13
AGU-Codec-II pp, 16×48 LCT-3	22.20	24.88	28.30	32.58	35.55	37.99
SPIHT-Codec-II pp, 16×48 LCT-3	22.36	24.93	27.91	32.09	35.12	37.60
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	21.98	24.77	28.25	32.50	35.46	37.86
SPIHT-Codec-II pp, 8×32 LCT-4	21.85	24.54	28.00	32.21	35.25	37.64
AGU-Codec-II pp, 16×64 LCT-4	22.54	25.26	28.74	33.01	36.00	38.48
SPIHT-Codec-II pp, 16×64 LCT-4	22.38	25.01	28.41	32.65	35.67	38.07
L-CEB-Codec-II pp, 8×32 GenLOT	21.64	24.24	27.74	32.04	34.97	37.35
SPIHT-Codec-II pp, 8×32 GenLOT	21.37	23.93	27.36	31.62	34.53	36.95
L-CEB-Codec-II pp, 8×32 ELT	21.90	24.69	28.22	32.47	35.41	37.81
SPIHT-Codec-II pp, 8×32 ELT	21.75	24.48	27.98	32.21	35.21	37.54
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	21.93	24.68	28.12	32.29	35.21	37.59
SPIHT-Codec-II pp, 8×48 LCT-6	22.06	24.40	27.82	31.98	34.93	37.43
AGU-Codec-II pp, 16×96 LCT-6	22.33	24.92	28.42	32.69	35.64	38.11
SPIHT-Codec-II pp, 16×96 LCT-6	22.49	25.01	28.05	32.25	35.30	37.77
L-CEB-Codec-II pp, 8×48 GenLOT	21.97	24.71	28.16	32.33	35.18	37.53
SPIHT-Codec-II pp, 8×48 GenLOT	21.81	24.46	27.88	32.05	34.94	37.26
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	21.60	24.02	27.30	31.34	33.97	36.10
AGU-Codec-II pp, 9/7-Wavelet	21.59	24.12	27.50	31.61	34.32	36.63
SPIHT pp, 9/7-Wavelet	21.60	23.99	27.28	31.46	34.20	36.29

Tabelle H.22: Kompressionsergebnisse für das Testbild „Fingerprint“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Peppers“



Abbildung H.89: Testbild „Peppers“ (512×512 , $H = 7.5715$, $\sigma^2 = 3295.1212$, $\bar{g} = 104.2099$, $g_{min} = 0$, $g_{max} = 229$).

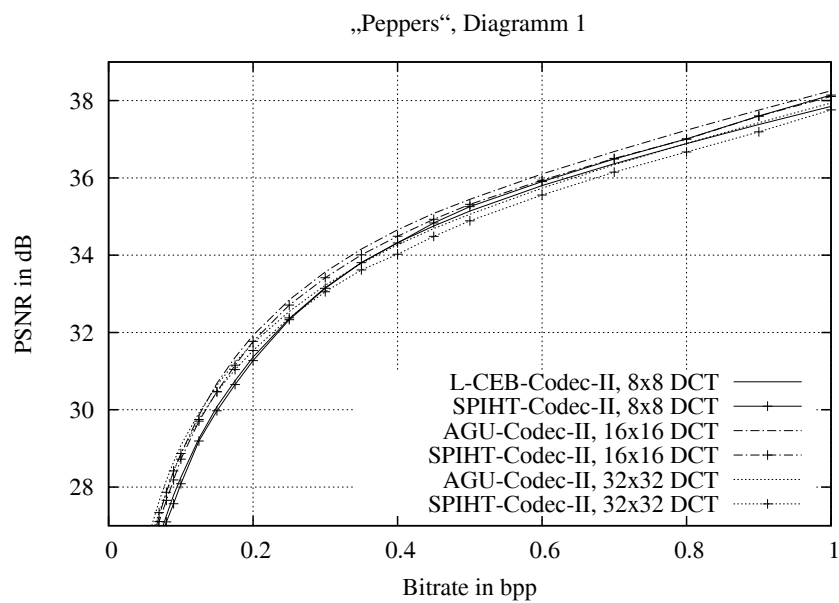


Abbildung H.90: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

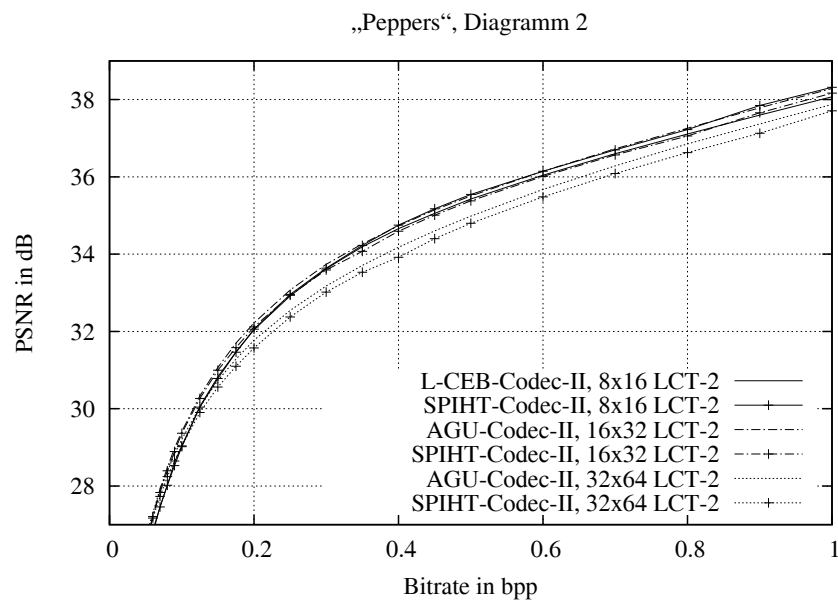


Abbildung H.91: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

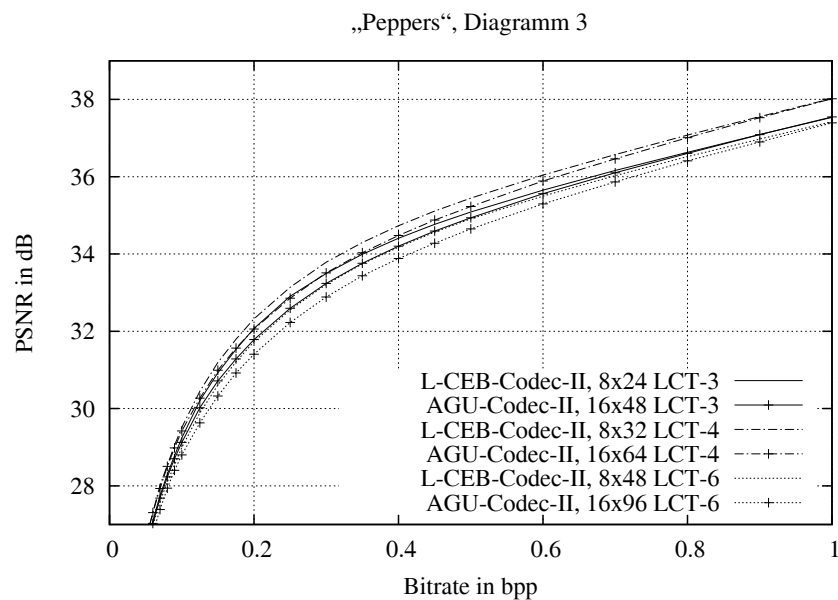


Abbildung H.92: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

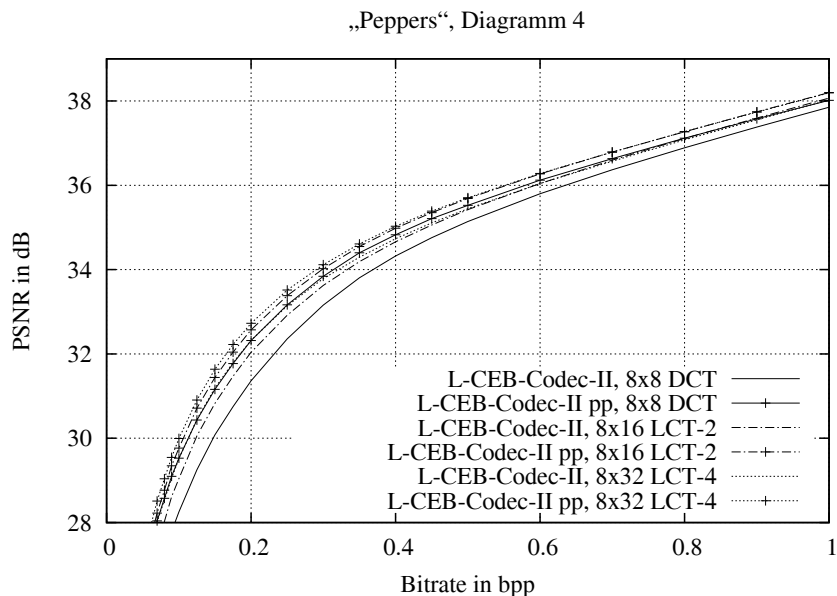


Abbildung H.93: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

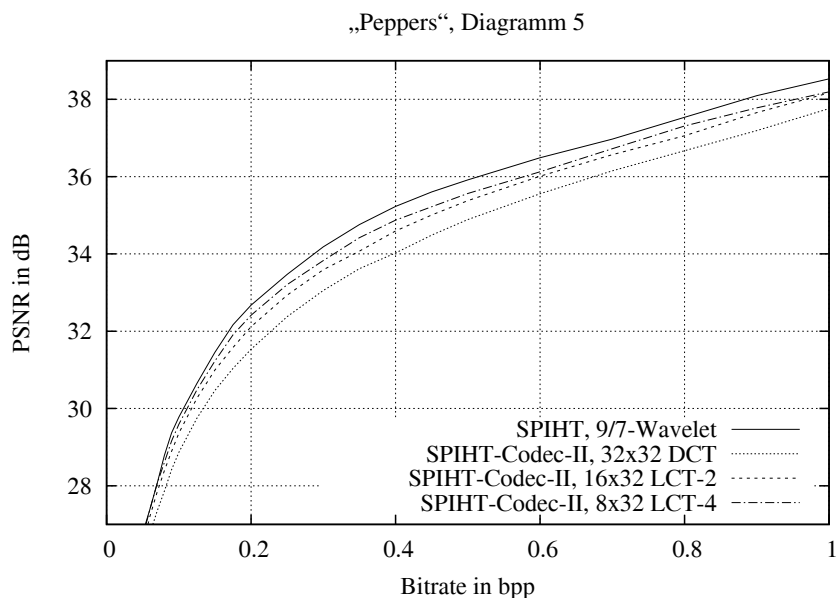


Abbildung H.94: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

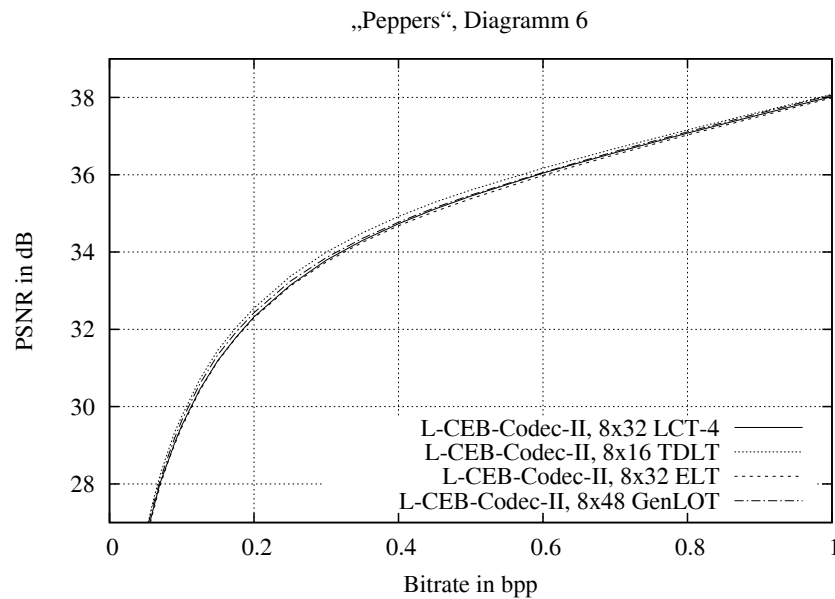


Abbildung H.95: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

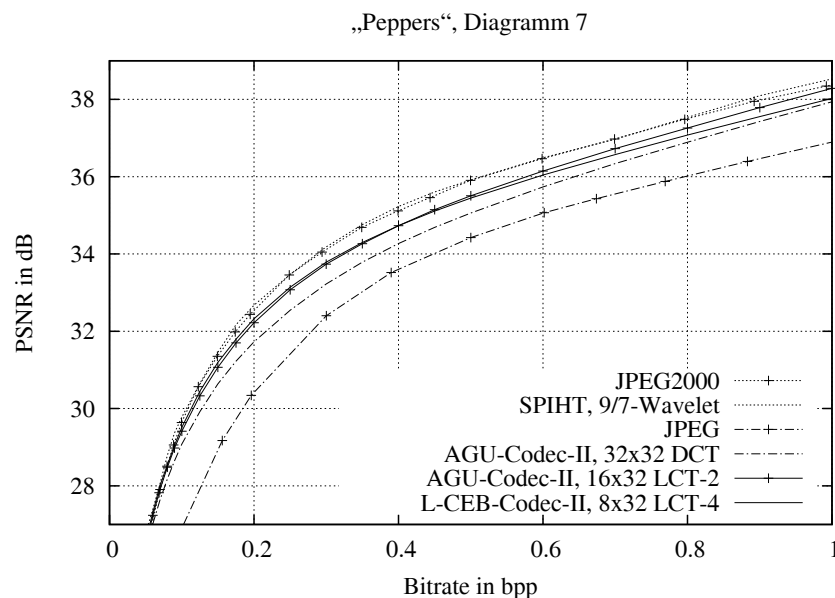


Abbildung H.96: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	26.23	29.25	32.36	35.14	36.64	37.85
SPIHT-Codec-II, 8×8 DCT	26.15	29.19	32.33	35.26	36.75	38.14
AGU-Codec-II, 16×16 DCT	26.76	29.86	32.85	35.45	36.96	38.26
SPIHT-Codec-II, 16×16 DCT	26.62	29.71	32.71	35.32	36.75	38.11
AGU-Codec-II, 32×32 DCT	27.17	29.92	32.55	35.06	36.62	37.94
SPIHT-Codec-II, 32×32 DCT	26.92	29.75	32.38	34.89	36.42	37.76
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	26.96	30.06	32.92	35.43	36.85	38.07
SPIHT-Codec-II, 8×16 LCT-2	27.00	30.04	32.96	35.54	36.95	38.32
AGU-Codec-II, 16×32 LCT-2	27.43	30.33	33.07	35.50	36.99	38.28
SPIHT-Codec-II, 16×32 LCT-2	27.39	30.26	32.93	35.38	36.80	38.17
AGU-Codec-II, 32×64 LCT-2	27.46	30.04	32.54	34.99	36.57	37.87
SPIHT-Codec-II, 32×64 LCT-2	27.31	29.90	32.37	34.80	36.36	37.71
L-CEB-Codec-II, 8×16 TDLT	27.73	30.71	33.37	35.60	36.92	38.08
SPIHT-Codec-II, 8×16 TDLT	27.73	30.72	33.37	35.71	37.04	38.34
L-CEB-Codec-II, 8×16 LOT Typ II	26.93	29.85	32.66	35.14	36.62	37.85
SPIHT-Codec-II, 8×16 LOT Typ II	26.88	29.81	32.70	35.29	36.73	38.12
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	27.28	30.20	32.91	35.09	36.39	37.53
SPIHT-Codec-II, 8×24 LCT-3	27.40	30.27	32.96	35.12	36.69	37.81
AGU-Codec-II, 16×48 LCT-3	27.19	30.02	32.59	34.94	36.35	37.55
SPIHT-Codec-II, 16×48 LCT-3	27.22	29.95	32.34	34.76	36.31	37.47
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	27.50	30.45	33.14	35.45	36.83	38.02
SPIHT-Codec-II, 8×32 LCT-4	27.57	30.48	33.21	35.56	37.06	38.20
AGU-Codec-II, 16×64 LCT-4	27.48	30.27	32.85	35.23	36.74	38.02
SPIHT-Codec-II, 16×64 LCT-4	27.47	30.24	32.75	35.11	36.58	37.88
L-CEB-Codec-II, 8×32 GenLOT	27.33	30.27	33.09	35.49	36.90	38.12
SPIHT-Codec-II, 8×32 GenLOT	27.29	30.25	33.13	35.60	36.99	38.35
L-CEB-Codec-II, 8×32 ELT	27.44	30.42	33.12	35.38	36.78	37.99
SPIHT-Codec-II, 8×32 ELT	27.50	30.44	33.17	35.51	36.87	38.22
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	27.11	29.89	32.56	34.91	36.28	37.42
SPIHT-Codec-II, 8×48 LCT-6	27.24	29.99	32.65	35.02	36.52	37.64
AGU-Codec-II, 16×96 LCT-6	26.94	29.63	32.23	34.65	36.14	37.39
SPIHT-Codec-II, 16×96 LCT-6	26.96	29.59	32.12	34.52	36.08	37.28
L-CEB-Codec-II, 8×48 GenLOT	27.58	30.59	33.24	35.47	36.85	38.06
SPIHT-Codec-II, 8×48 GenLOT	27.58	30.58	33.26	35.58	36.94	38.29
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	27.53	30.57	33.35	35.69	37.04	38.19
AGU-Codec-II, 9/7-Wavelet	27.34	30.34	33.20	35.76	37.20	38.46
SPIHT, 9/7-Wavelet	27.56	30.65	33.47	35.92	37.25	38.53
JPEG2000	27.20	30.57	33.46	35.90	37.22	38.35

Tabelle H.23: Kompressionsergebnisse für das Testbild „Peppers“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	27.59	30.43	33.16	35.53	36.88	38.02
SPIHT-Codec-II pp, 8×8 DCT	27.21	30.10	32.94	35.50	36.85	38.17
AGU-Codec-II pp, 16×16 DCT	27.69	30.64	33.37	35.70	37.12	38.36
SPIHT-Codec-II pp, 16×16 DCT	27.34	30.27	33.08	35.49	36.86	38.16
AGU-Codec-II pp, 32×32 DCT	27.85	30.52	33.05	35.41	36.88	38.14
SPIHT-Codec-II pp, 32×32 DCT	27.40	30.16	32.71	35.10	36.59	37.85
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	27.73	30.71	33.38	35.69	37.03	38.20
SPIHT-Codec-II pp, 8×16 LCT-2	27.57	30.49	33.25	35.67	37.00	38.32
AGU-Codec-II pp, 16×32 LCT-2	27.98	30.82	33.44	35.74	37.16	38.41
SPIHT-Codec-II pp, 16×32 LCT-2	27.75	30.57	33.16	35.50	36.91	38.20
AGU-Codec-II pp, 32×64 LCT-2	27.96	30.57	33.02	35.38	36.89	38.14
SPIHT-Codec-II pp, 32×64 LCT-2	27.64	30.24	32.67	35.03	36.56	37.87
L-CEB-Codec-II pp, 8×16 TDLT	28.05	31.01	33.62	35.77	37.05	38.17
SPIHT-Codec-II pp, 8×16 TDLT	27.93	30.90	33.51	35.76	37.05	38.34
L-CEB-Codec-II pp, 8×16 LOT Typ II	27.84	30.71	33.36	35.57	36.91	38.07
SPIHT-Codec-II pp, 8×16 LOT Typ II	27.57	30.47	33.21	35.55	36.87	38.20
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	27.84	30.77	33.39	35.48	36.76	37.87
SPIHT-Codec-II pp, 8×24 LCT-3	27.80	30.63	33.36	35.40	36.83	37.92
AGU-Codec-II pp, 16×48 LCT-3	27.83	30.64	33.16	35.40	36.76	37.91
SPIHT-Codec-II pp, 16×48 LCT-3	27.65	30.49	32.86	35.13	36.54	37.66
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	28.04	30.91	33.52	35.71	37.03	38.19
SPIHT-Codec-II pp, 8×32 LCT-4	27.92	30.79	33.45	35.69	37.12	38.23
AGU-Codec-II pp, 16×64 LCT-4	27.99	30.77	33.33	35.57	37.01	38.24
SPIHT-Codec-II pp, 16×64 LCT-4	27.81	30.61	33.09	35.32	36.80	37.99
L-CEB-Codec-II pp, 8×32 GenLOT	28.03	30.89	33.53	35.75	37.09	38.26
SPIHT-Codec-II pp, 8×32 GenLOT	27.81	30.71	33.42	35.73	37.04	38.38
L-CEB-Codec-II pp, 8×32 ELT	28.06	30.95	33.56	35.70	37.03	38.18
SPIHT-Codec-II pp, 8×32 ELT	27.90	30.79	33.46	35.68	37.00	38.27
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	27.80	30.58	33.18	35.39	36.69	37.81
SPIHT-Codec-II pp, 8×48 LCT-6	27.77	30.55	33.16	35.36	36.71	37.82
AGU-Codec-II pp, 16×96 LCT-6	27.66	30.39	32.95	35.22	36.64	37.83
SPIHT-Codec-II pp, 16×96 LCT-6	27.53	30.26	32.73	34.98	36.38	37.55
L-CEB-Codec-II pp, 8×48 GenLOT	28.15	31.09	33.66	35.78	37.10	38.24
SPIHT-Codec-II pp, 8×48 GenLOT	27.92	30.90	33.55	35.75	37.05	38.33
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	27.91	30.90	33.61	35.82	37.12	38.23
AGU-Codec-II pp, 9/7-Wavelet	27.63	30.63	33.41	35.84	37.23	38.46
SPIHT pp, 9/7-Wavelet	27.76	30.86	33.61	35.92	37.20	38.47

Tabelle H.24: Kompressionsergebnisse für das Testbild „Peppers“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „F16“



Abbildung H.97: Testbild „F16“ (512×512 , $H = 6.704$, $\sigma^2 = 2153.8785$, $\bar{g} = 178.673$, $g_{min} = 16$, $g_{max} = 231$).

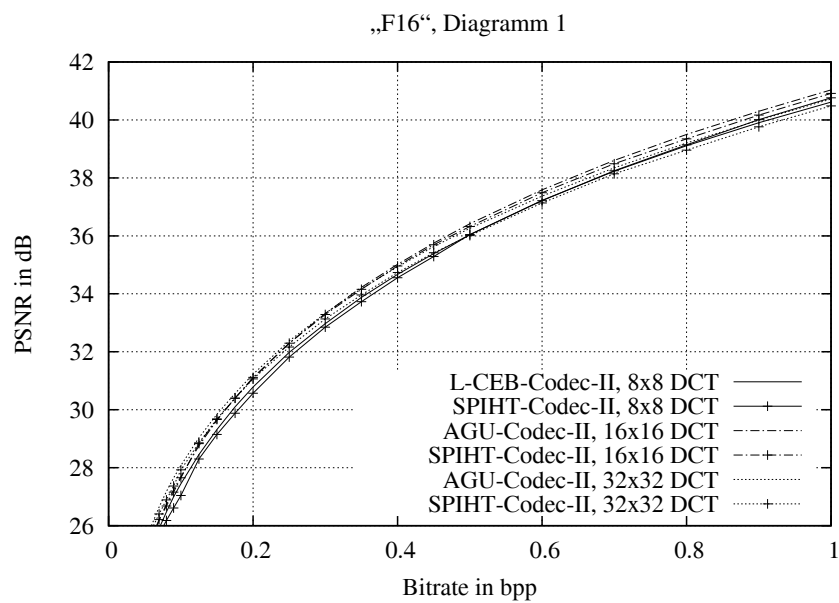


Abbildung H.98: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

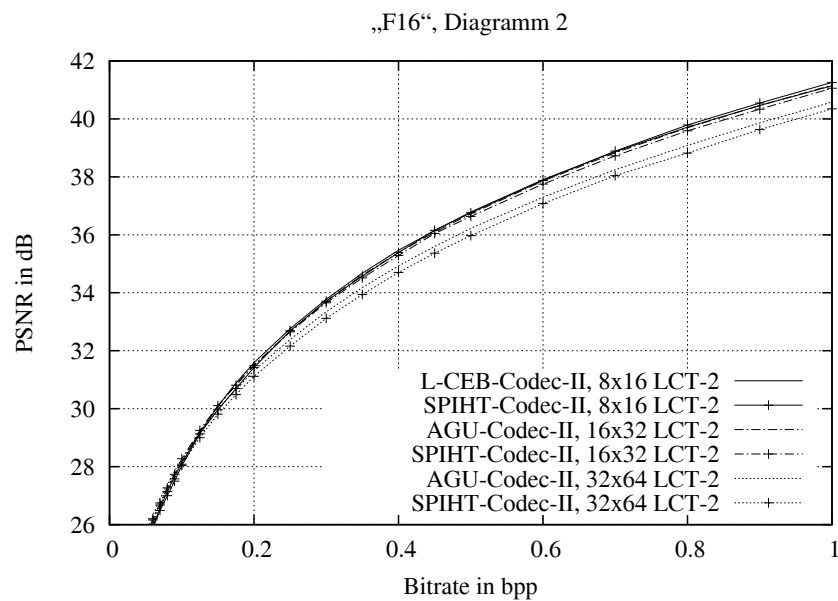


Abbildung H.99: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

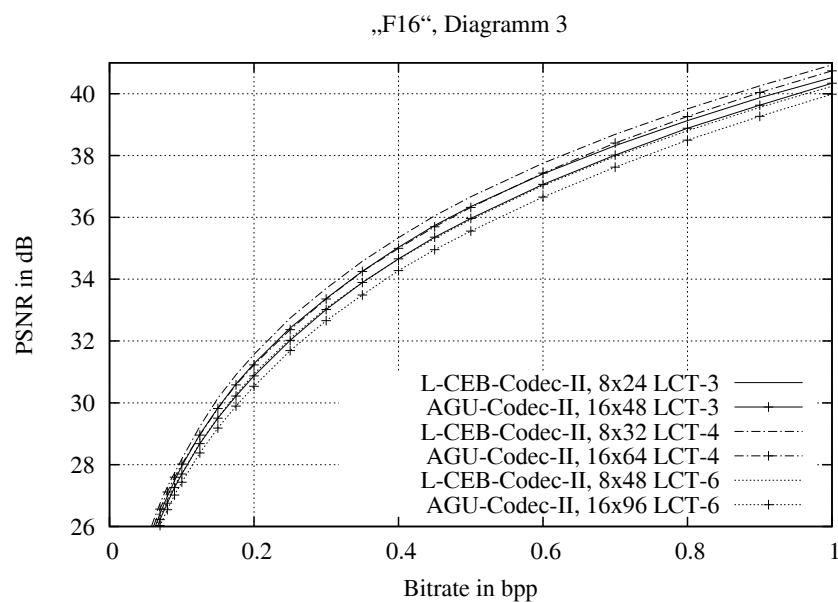


Abbildung H.100: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

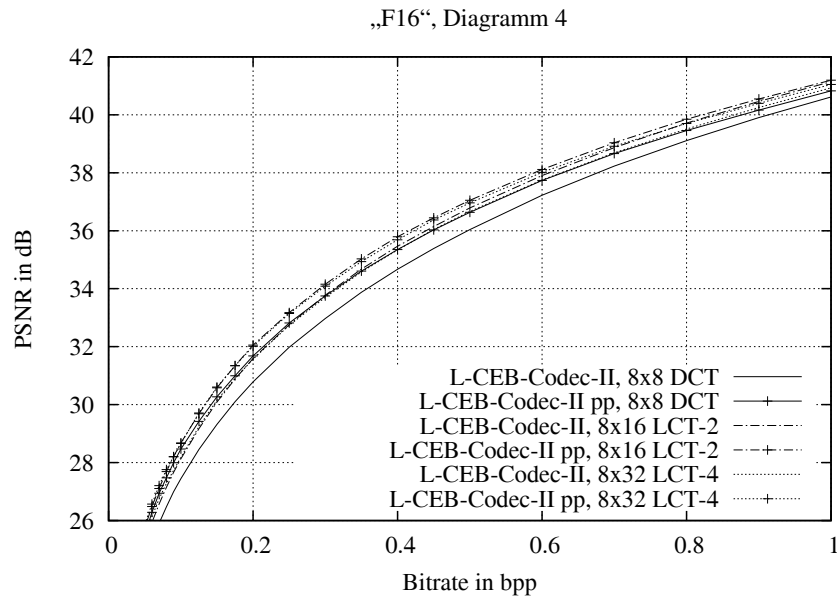


Abbildung H.101: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

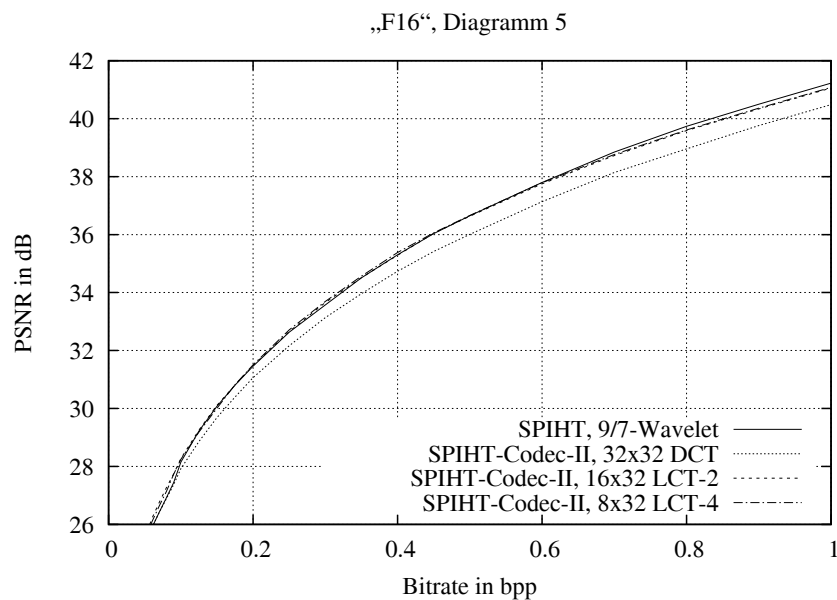


Abbildung H.102: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

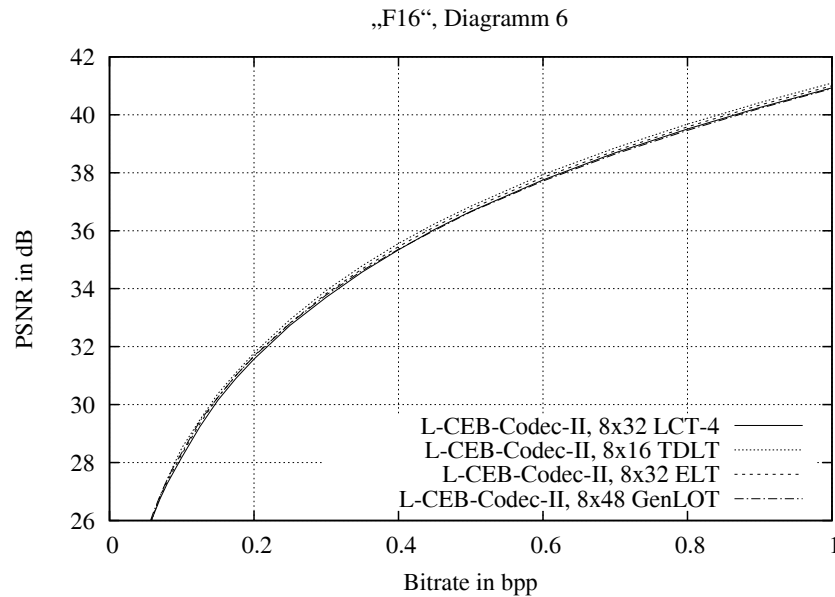


Abbildung H.103: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

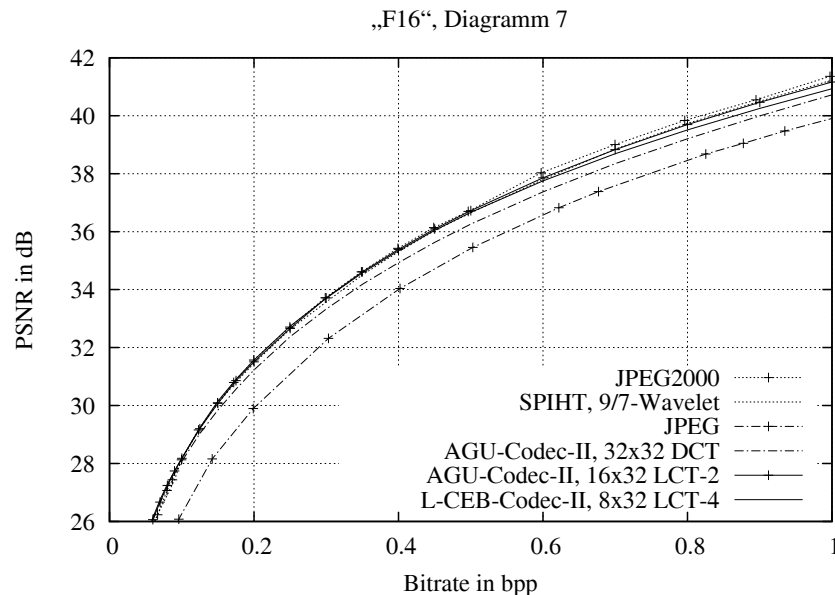


Abbildung H.104: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	25.42	28.46	31.98	36.04	38.68	40.61
SPIHT-Codec-II, 8×8 DCT	25.31	28.30	31.82	36.04	38.73	40.76
AGU-Codec-II, 16×16 DCT	25.70	28.76	32.29	36.42	39.06	41.04
SPIHT-Codec-II, 16×16 DCT	25.78	28.83	32.29	36.32	38.92	40.91
AGU-Codec-II, 32×32 DCT	26.20	29.03	32.38	36.26	38.79	40.72
SPIHT-Codec-II, 32×32 DCT	26.03	28.86	32.17	36.01	38.59	40.48
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	26.10	29.17	32.76	36.79	39.31	41.14
SPIHT-Codec-II, 8×16 LCT-2	26.05	29.13	32.69	36.76	39.30	41.26
AGU-Codec-II, 16×32 LCT-2	26.21	29.20	32.66	36.72	39.27	41.17
SPIHT-Codec-II, 16×32 LCT-2	26.36	29.26	32.65	36.64	39.12	41.06
AGU-Codec-II, 32×64 LCT-2	26.48	29.18	32.39	36.22	38.67	40.58
SPIHT-Codec-II, 32×64 LCT-2	26.29	29.00	32.16	35.97	38.47	40.35
L-CEB-Codec-II, 8×16 TDLT	26.29	29.48	32.94	36.85	39.28	41.09
SPIHT-Codec-II, 8×16 TDLT	26.05	29.17	32.84	36.80	39.34	41.17
L-CEB-Codec-II, 8×16 LOT Typ II	26.09	29.11	32.56	36.48	39.00	40.91
SPIHT-Codec-II, 8×16 LOT Typ II	25.91	29.00	32.50	36.48	39.04	41.03
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	25.96	28.98	32.42	36.35	38.73	40.52
SPIHT-Codec-II, 8×24 LCT-3	25.86	28.74	32.28	36.21	38.82	40.44
AGU-Codec-II, 16×48 LCT-3	25.76	28.68	32.02	35.96	38.46	40.34
SPIHT-Codec-II, 16×48 LCT-3	25.73	28.59	32.01	35.87	38.42	40.15
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	26.30	29.24	32.74	36.67	39.12	40.93
SPIHT-Codec-II, 8×32 LCT-4	26.23	29.18	32.71	36.66	39.21	41.08
AGU-Codec-II, 16×64 LCT-4	26.12	28.96	32.37	36.32	38.85	40.74
SPIHT-Codec-II, 16×64 LCT-4	26.23	29.07	32.38	36.25	38.76	40.63
L-CEB-Codec-II, 8×32 GenLOT	26.21	29.26	32.80	36.77	39.26	41.13
SPIHT-Codec-II, 8×32 GenLOT	26.11	29.18	32.73	36.75	39.28	41.24
L-CEB-Codec-II, 8×32 ELT	26.34	29.34	32.82	36.76	39.18	41.00
SPIHT-Codec-II, 8×32 ELT	26.26	29.27	32.76	36.73	39.20	41.11
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	25.90	28.69	32.08	35.93	38.41	40.25
SPIHT-Codec-II, 8×48 LCT-6	25.88	28.76	32.10	35.98	38.56	40.36
AGU-Codec-II, 16×96 LCT-6	25.60	28.38	31.69	35.56	38.08	39.98
SPIHT-Codec-II, 16×96 LCT-6	25.81	28.53	31.70	35.50	38.02	39.88
L-CEB-Codec-II, 8×48 GenLOT	26.42	29.43	32.81	36.65	39.06	40.90
SPIHT-Codec-II, 8×48 GenLOT	26.34	29.35	32.78	36.63	39.08	40.99
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	26.29	29.29	32.56	36.47	39.03	40.90
AGU-Codec-II, 9/7-Wavelet	25.87	28.94	32.49	36.46	39.11	41.13
SPIHT, 9/7-Wavelet	26.04	29.22	32.63	36.65	39.28	41.23
JPEG2000	25.80	29.16	32.71	36.70	39.45	41.36

Tabelle H.25: Kompressionsergebnisse für das Testbild „F16“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	26.43	29.42	32.82	36.63	39.06	40.82
SPIHT-Codec-II pp, 8×8 DCT	26.11	29.11	32.51	36.44	39.01	40.85
AGU-Codec-II pp, 16×16 DCT	26.44	29.44	32.87	36.83	39.32	41.17
SPIHT-Codec-II pp, 16×16 DCT	26.32	29.32	32.69	36.58	39.13	40.92
AGU-Codec-II pp, 32×32 DCT	26.74	29.58	32.88	36.70	39.12	40.96
SPIHT-Codec-II pp, 32×32 DCT	26.42	29.25	32.53	36.27	38.85	40.57
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	26.66	29.69	33.18	37.05	39.46	41.20
SPIHT-Codec-II pp, 8×16 LCT-2	26.44	29.48	32.94	36.90	39.38	41.17
AGU-Codec-II pp, 16×32 LCT-2	26.66	29.64	33.06	37.00	39.47	41.28
SPIHT-Codec-II pp, 16×32 LCT-2	26.64	29.52	32.87	36.77	39.26	41.02
AGU-Codec-II pp, 32×64 LCT-2	26.88	29.66	32.90	36.66	39.04	40.86
SPIHT-Codec-II pp, 32×64 LCT-2	26.56	29.32	32.45	36.20	38.74	40.44
L-CEB-Codec-II pp, 8×16 TDLT	26.55	29.77	33.22	37.04	39.39	41.09
SPIHT-Codec-II pp, 8×16 TDLT	26.18	29.35	33.03	36.95	39.36	41.05
L-CEB-Codec-II pp, 8×16 LOT Typ II	26.75	29.72	33.14	36.93	39.30	41.05
SPIHT-Codec-II pp, 8×16 LOT Typ II	26.38	29.45	32.91	36.75	39.22	41.04
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	26.51	29.56	32.95	36.76	39.07	40.77
SPIHT-Codec-II pp, 8×24 LCT-3	26.27	29.14	32.64	36.50	38.96	40.60
AGU-Codec-II pp, 16×48 LCT-3	26.33	29.30	32.63	36.52	38.91	40.70
SPIHT-Codec-II pp, 16×48 LCT-3	26.04	29.02	32.43	36.25	38.64	40.40
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	26.73	29.71	33.15	36.96	39.32	41.04
SPIHT-Codec-II pp, 8×32 LCT-4	26.52	29.48	32.94	36.81	39.24	41.06
AGU-Codec-II pp, 16×64 LCT-4	26.55	29.44	32.85	36.75	39.18	40.98
SPIHT-Codec-II pp, 16×64 LCT-4	26.53	29.37	32.67	36.48	39.00	40.72
L-CEB-Codec-II pp, 8×32 GenLOT	26.79	29.76	33.23	37.06	39.43	41.20
SPIHT-Codec-II pp, 8×32 GenLOT	26.49	29.51	33.01	36.92	39.38	41.19
L-CEB-Codec-II pp, 8×32 ELT	26.80	29.81	33.25	37.08	39.41	41.12
SPIHT-Codec-II pp, 8×32 ELT	26.56	29.58	33.02	36.90	39.35	41.07
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	26.52	29.37	32.69	36.48	38.84	40.59
SPIHT-Codec-II pp, 8×48 LCT-6	26.28	29.20	32.51	36.34	38.74	40.56
AGU-Codec-II pp, 16×96 LCT-6	26.23	29.08	32.40	36.24	38.67	40.47
SPIHT-Codec-II pp, 16×96 LCT-6	26.23	29.04	32.23	35.99	38.33	40.23
L-CEB-Codec-II pp, 8×48 GenLOT	26.86	29.86	33.27	37.04	39.34	41.07
SPIHT-Codec-II pp, 8×48 GenLOT	26.63	29.64	33.07	36.85	39.27	41.00
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	26.62	29.65	32.91	36.75	39.17	40.93
AGU-Codec-II pp, 9/7-Wavelet	26.10	29.24	32.77	36.70	39.22	41.11
SPIHT pp, 9/7-Wavelet	26.18	29.45	32.84	36.78	39.31	41.09

Tabelle H.26: Kompressionsergebnisse für das Testbild „F16“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Bubble“

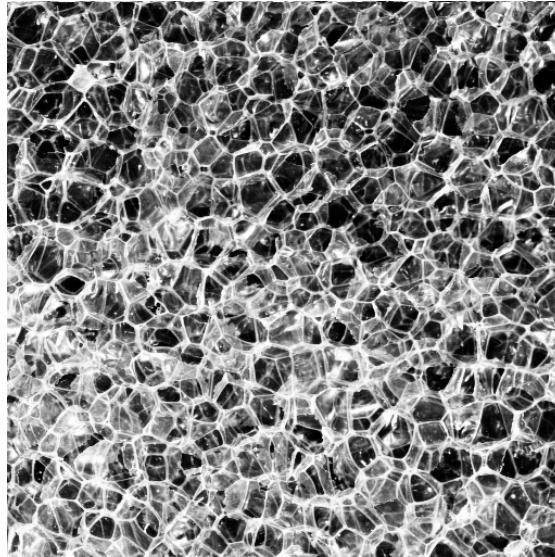


Abbildung H.105: Testbild „Bubble“ (512×512 , $H = 7.1715$, $\sigma^2 = 5477.0194$, $\bar{g} = 128.1747$, $g_{min} = 0$, $g_{max} = 255$).

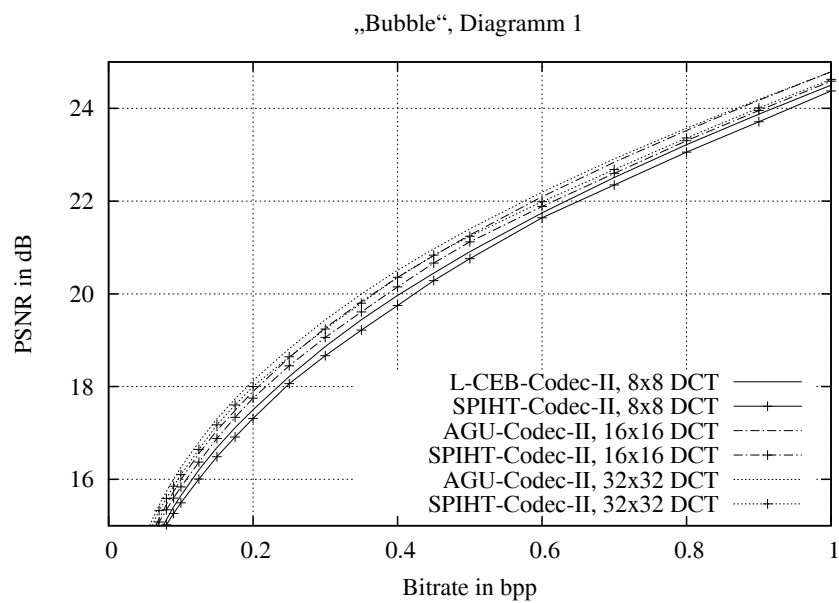


Abbildung H.106: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

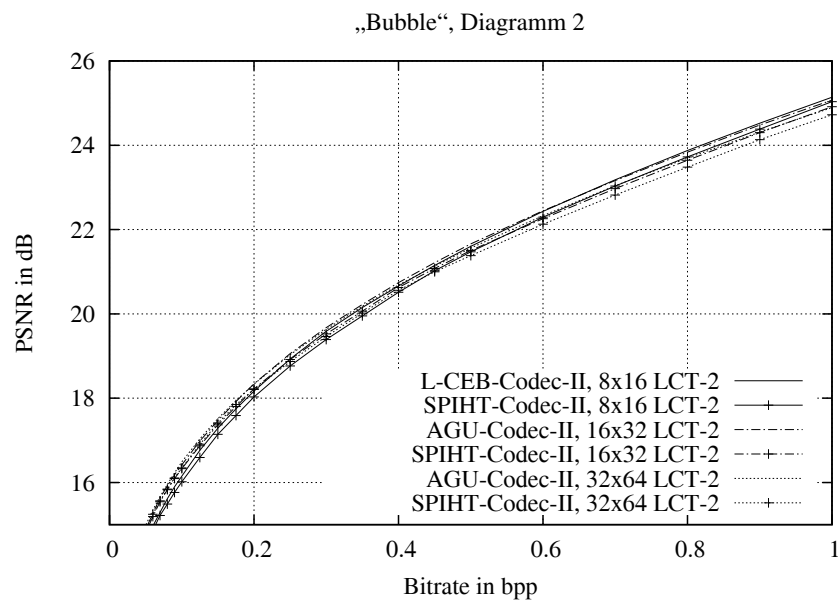


Abbildung H.107: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

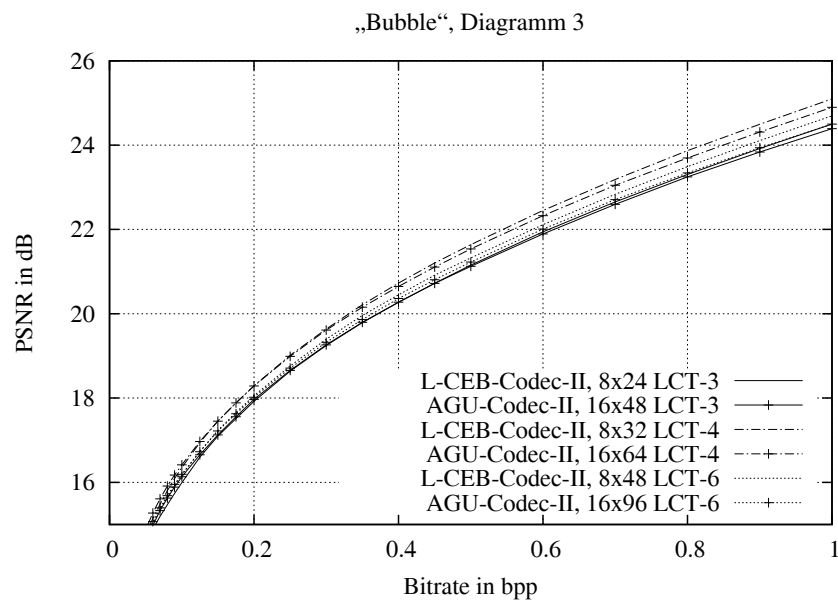


Abbildung H.108: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

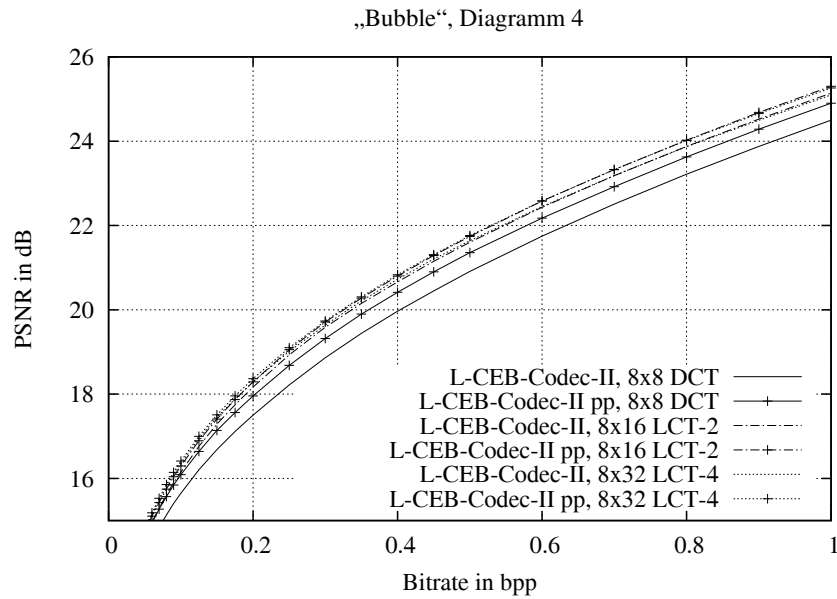


Abbildung H.109: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

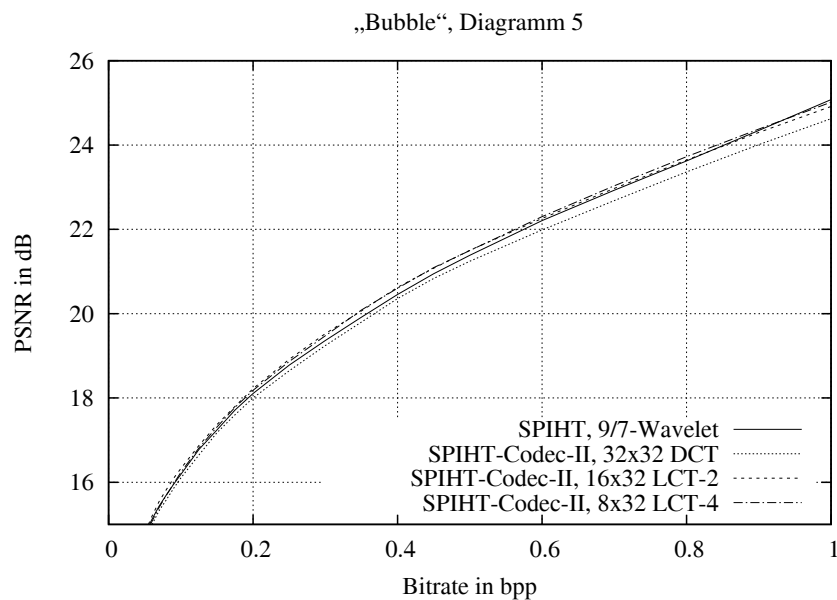


Abbildung H.110: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

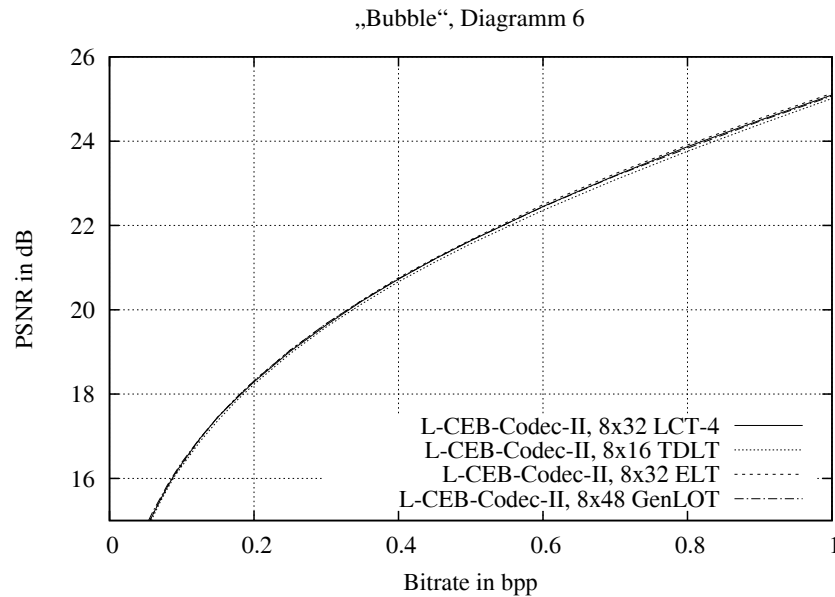


Abbildung H.111: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

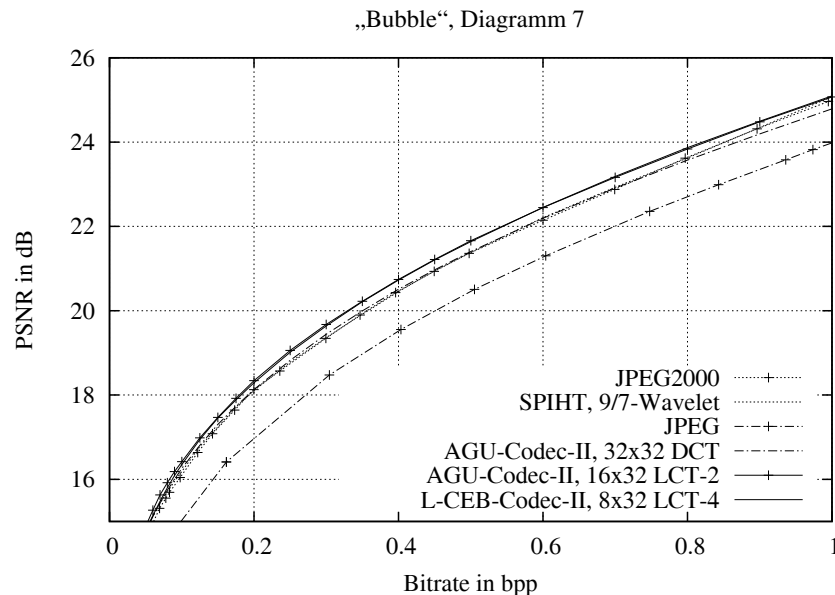


Abbildung H.112: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	14.62	16.20	18.22	20.91	22.87	24.50
SPIHT-Codec-II, 8×8 DCT	14.54	16.00	18.06	20.76	22.70	24.37
AGU-Codec-II, 16×16 DCT	14.90	16.56	18.62	21.27	23.19	24.79
SPIHT-Codec-II, 16×16 DCT	14.83	16.37	18.45	21.12	22.96	24.59
AGU-Codec-II, 32×32 DCT	15.20	16.81	18.82	21.40	23.26	24.79
SPIHT-Codec-II, 32×32 DCT	15.09	16.65	18.64	21.24	23.03	24.63
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	15.06	16.76	18.93	21.60	23.53	25.14
SPIHT-Codec-II, 8×16 LCT-2	14.98	16.59	18.76	21.47	23.38	25.03
AGU-Codec-II, 16×32 LCT-2	15.36	16.98	19.06	21.66	23.50	25.07
SPIHT-Codec-II, 16×32 LCT-2	15.29	16.87	18.92	21.50	23.32	24.91
AGU-Codec-II, 32×64 LCT-2	15.43	17.04	19.03	21.56	23.37	24.89
SPIHT-Codec-II, 32×64 LCT-2	15.34	16.91	18.87	21.38	23.17	24.72
L-CEB-Codec-II, 8×16 TDLT	15.16	16.87	18.96	21.57	23.42	25.01
SPIHT-Codec-II, 8×16 TDLT	15.02	16.74	18.78	21.40	23.29	24.85
L-CEB-Codec-II, 8×16 LOT Typ II	15.02	16.76	18.87	21.52	23.41	25.01
SPIHT-Codec-II, 8×16 LOT Typ II	14.95	16.57	18.72	21.40	23.28	24.93
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	14.95	16.60	18.64	21.15	22.97	24.50
SPIHT-Codec-II, 8×24 LCT-3	14.97	16.54	18.54	21.11	22.79	24.49
AGU-Codec-II, 16×48 LCT-3	15.07	16.65	18.65	21.12	22.92	24.39
SPIHT-Codec-II, 16×48 LCT-3	15.06	16.56	18.54	21.05	22.68	24.27
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	15.23	16.94	19.01	21.64	23.52	25.09
SPIHT-Codec-II, 8×32 LCT-4	15.15	16.81	18.87	21.50	23.39	25.02
AGU-Codec-II, 16×64 LCT-4	15.37	16.97	18.99	21.54	23.38	24.90
SPIHT-Codec-II, 16×64 LCT-4	15.31	16.90	18.85	21.37	23.22	24.74
L-CEB-Codec-II, 8×32 GenLOT	15.08	16.75	18.90	21.59	23.51	25.12
SPIHT-Codec-II, 8×32 GenLOT	15.01	16.57	18.74	21.46	23.35	25.03
L-CEB-Codec-II, 8×32 ELT	15.24	16.93	19.04	21.66	23.58	25.14
SPIHT-Codec-II, 8×32 ELT	15.17	16.78	18.91	21.54	23.41	25.02
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	15.12	16.73	18.76	21.32	23.16	24.69
SPIHT-Codec-II, 8×48 LCT-6	15.12	16.64	18.67	21.24	23.03	24.59
AGU-Codec-II, 16×96 LCT-6	15.15	16.73	18.71	21.23	23.02	24.50
SPIHT-Codec-II, 16×96 LCT-6	15.08	16.64	18.60	21.13	22.85	24.37
L-CEB-Codec-II, 8×48 GenLOT	15.30	16.95	19.04	21.64	23.51	25.07
SPIHT-Codec-II, 8×48 GenLOT	15.23	16.81	18.90	21.52	23.38	24.99
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	15.28	16.87	18.83	21.35	23.28	24.96
AGU-Codec-II, 9/7-Wavelet	15.20	16.82	18.77	21.31	23.25	24.99
SPIHT, 9/7-Wavelet	15.20	16.78	18.78	21.39	23.27	25.08
JPEG2000	14.80	16.64	18.57	21.36	23.24	24.96

Tabelle H.27: Kompressionsergebnisse für das Testbild „Bubble“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	15.03	16.64	18.68	21.36	23.28	24.90
SPIHT-Codec-II pp, 8×8 DCT	14.80	16.39	18.49	21.04	23.04	24.64
AGU-Codec-II pp, 16×16 DCT	15.14	16.78	18.85	21.50	23.42	25.03
SPIHT-Codec-II pp, 16×16 DCT	14.93	16.54	18.66	21.21	23.12	24.70
AGU-Codec-II pp, 32×32 DCT	15.27	16.89	18.93	21.56	23.43	24.99
SPIHT-Codec-II pp, 32×32 DCT	15.04	16.68	18.75	21.26	23.13	24.69
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	15.20	16.89	19.05	21.74	23.68	25.30
SPIHT-Codec-II pp, 8×16 LCT-2	14.98	16.65	18.86	21.46	23.45	25.06
AGU-Codec-II pp, 16×32 LCT-2	15.37	17.00	19.10	21.75	23.63	25.22
SPIHT-Codec-II pp, 16×32 LCT-2	15.18	16.85	18.95	21.47	23.36	24.94
AGU-Codec-II pp, 32×64 LCT-2	15.40	17.01	19.06	21.64	23.50	25.06
SPIHT-Codec-II pp, 32×64 LCT-2	15.20	16.84	18.90	21.35	23.22	24.76
L-CEB-Codec-II pp, 8×16 TDLT	15.13	16.88	19.02	21.69	23.59	25.20
SPIHT-Codec-II pp, 8×16 TDLT	14.97	16.75	18.74	21.44	23.38	24.94
L-CEB-Codec-II pp, 8×16 LOT Typ II	15.18	16.89	19.01	21.71	23.63	25.25
SPIHT-Codec-II pp, 8×16 LOT Typ II	14.96	16.63	18.86	21.41	23.42	25.00
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	15.08	16.76	18.82	21.40	23.24	24.79
SPIHT-Codec-II pp, 8×24 LCT-3	14.98	16.66	18.57	21.20	22.88	24.65
AGU-Codec-II pp, 16×48 LCT-3	15.15	16.72	18.76	21.30	23.15	24.67
SPIHT-Codec-II pp, 16×48 LCT-3	15.06	16.62	18.55	21.12	22.77	24.41
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	15.27	17.00	19.10	21.76	23.67	25.26
SPIHT-Codec-II pp, 8×32 LCT-4	15.20	16.76	18.89	21.57	23.42	25.03
AGU-Codec-II pp, 16×64 LCT-4	15.36	16.97	19.03	21.64	23.50	25.07
SPIHT-Codec-II pp, 16×64 LCT-4	15.33	16.82	18.86	21.43	23.25	24.78
L-CEB-Codec-II pp, 8×32 GenLOT	15.23	16.91	19.04	21.74	23.68	25.30
SPIHT-Codec-II pp, 8×32 GenLOT	15.03	16.67	18.86	21.44	23.43	25.05
L-CEB-Codec-II pp, 8×32 ELT	15.28	17.00	19.12	21.78	23.72	25.30
SPIHT-Codec-II pp, 8×32 ELT	15.10	16.78	18.97	21.51	23.47	25.06
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	15.20	16.84	18.91	21.52	23.40	24.97
SPIHT-Codec-II pp, 8×48 LCT-6	15.12	16.58	18.69	21.32	23.11	24.75
AGU-Codec-II pp, 16×96 LCT-6	15.19	16.78	18.80	21.39	23.23	24.76
SPIHT-Codec-II pp, 16×96 LCT-6	15.08	16.54	18.59	21.19	22.92	24.51
L-CEB-Codec-II pp, 8×48 GenLOT	15.34	17.01	19.14	21.80	23.70	25.29
SPIHT-Codec-II pp, 8×48 GenLOT	15.14	16.81	18.98	21.53	23.49	25.07
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	15.24	16.87	18.86	21.46	23.43	25.12
AGU-Codec-II pp, 9/7-Wavelet	15.15	16.79	18.79	21.39	23.37	25.12
SPIHT pp, 9/7-Wavelet	15.21	16.71	18.82	21.35	23.37	25.10

Tabelle H.28: Kompressionsergebnisse für das Testbild „Bubble“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Bark“



Abbildung H.113: Testbild „Bark“ (512×512 , $H = 7.1321$, $\sigma^2 = 5495.5292$, $\bar{g} = 127.6333$, $g_{min} = 0$, $g_{max} = 255$).

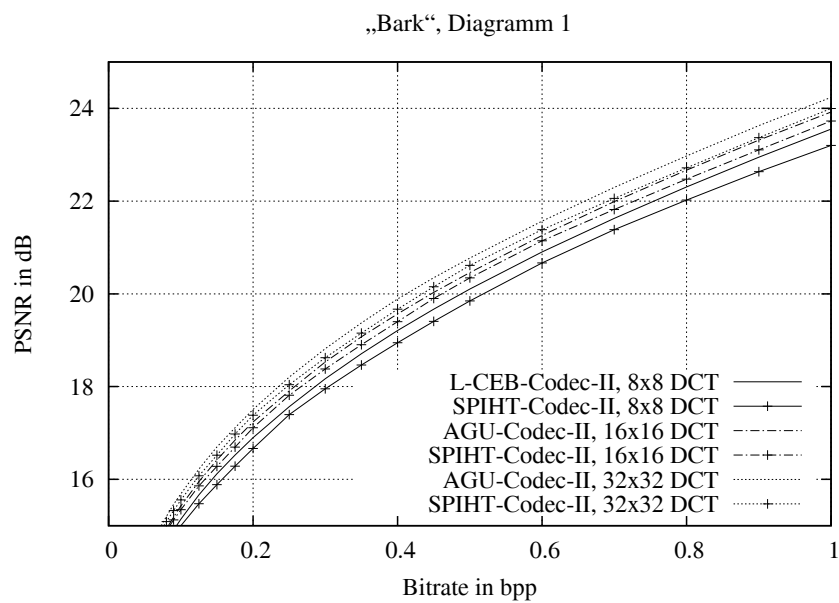


Abbildung H.114: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

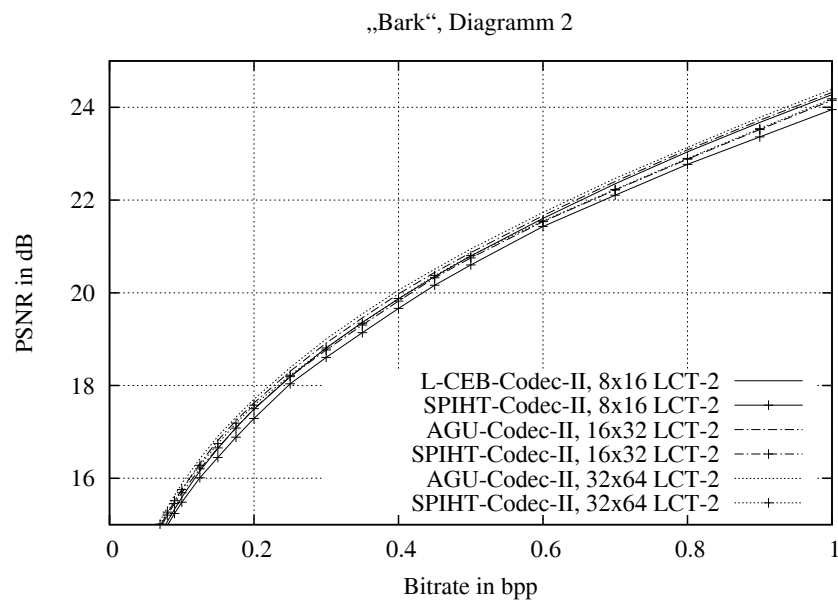


Abbildung H.115: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

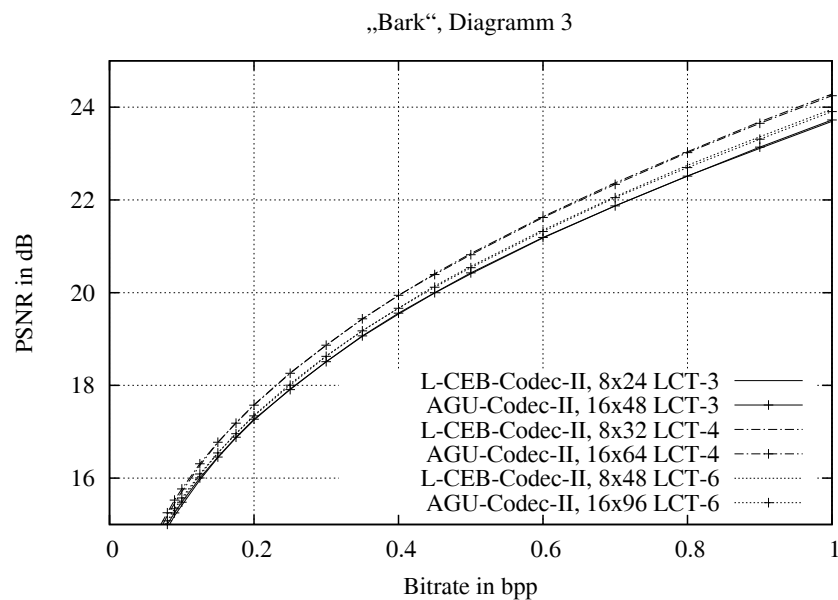


Abbildung H.116: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

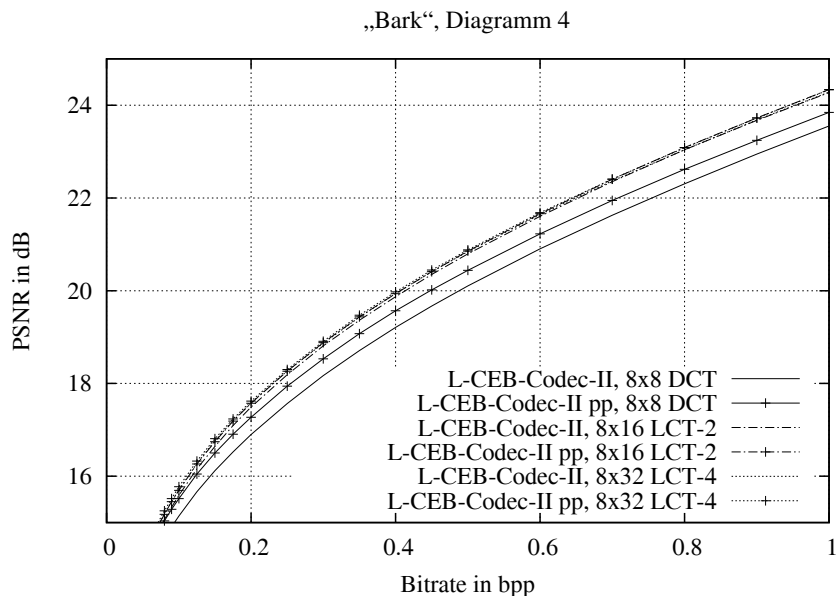


Abbildung H.117: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

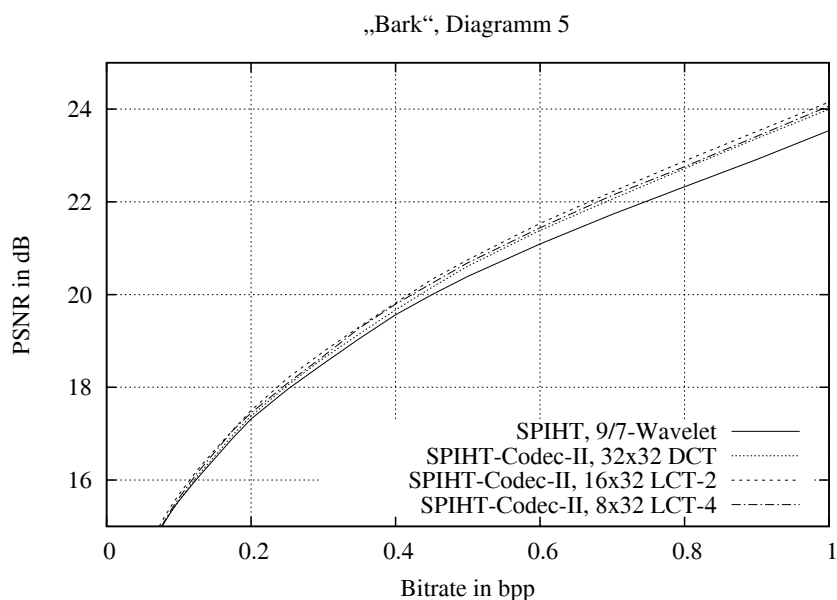


Abbildung H.118: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

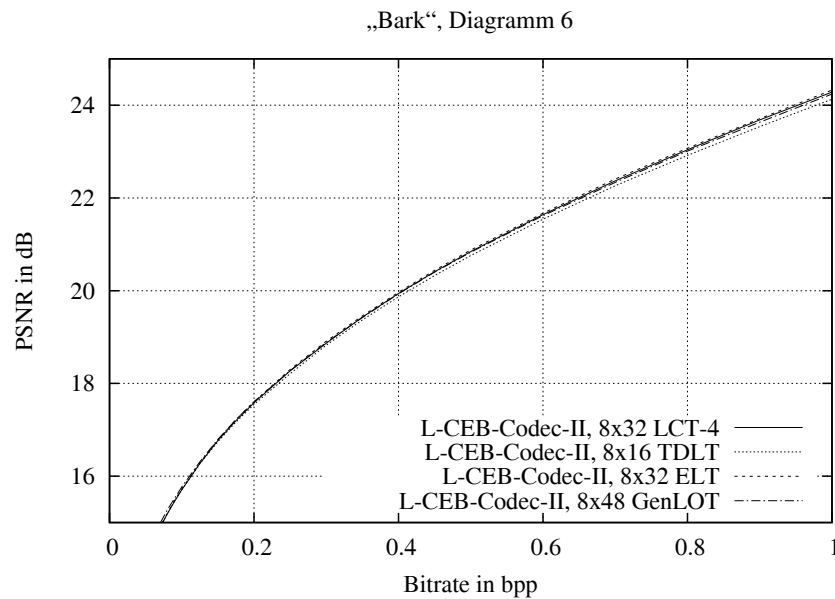


Abbildung H.119: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

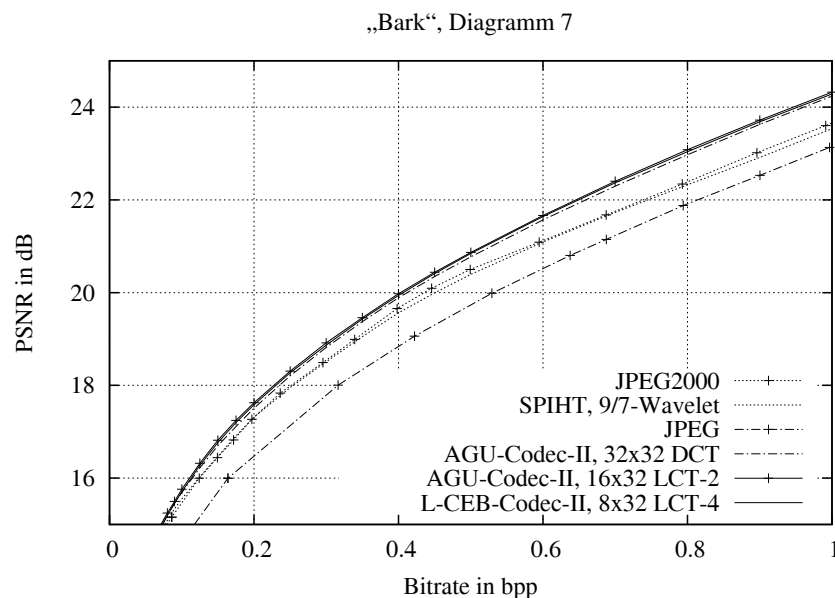


Abbildung H.120: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	14.17	15.66	17.57	20.10	21.98	23.55
SPIHT-Codec-II, 8×8 DCT	14.08	15.47	17.40	19.85	21.70	23.20
AGU-Codec-II, 16×16 DCT	14.42	15.95	17.90	20.46	22.33	23.92
SPIHT-Codec-II, 16×16 DCT	14.37	15.86	17.81	20.34	22.15	23.73
AGU-Codec-II, 32×32 DCT	14.69	16.22	18.21	20.77	22.64	24.24
SPIHT-Codec-II, 32×32 DCT	14.58	16.08	18.04	20.61	22.39	24.00
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	14.54	16.16	18.19	20.79	22.71	24.27
SPIHT-Codec-II, 8×16 LCT-2	14.46	16.01	18.04	20.60	22.45	23.95
AGU-Codec-II, 16×32 LCT-2	14.75	16.32	18.31	20.87	22.75	24.32
SPIHT-Codec-II, 16×32 LCT-2	14.68	16.21	18.19	20.75	22.54	24.15
AGU-Codec-II, 32×64 LCT-2	14.86	16.42	18.39	20.95	22.81	24.39
SPIHT-Codec-II, 32×64 LCT-2	14.78	16.28	18.22	20.80	22.56	24.18
L-CEB-Codec-II, 8×16 TDLT	14.63	16.27	18.21	20.76	22.59	24.13
SPIHT-Codec-II, 8×16 TDLT	14.51	16.11	18.02	20.52	22.41	23.83
L-CEB-Codec-II, 8×16 LOT Typ II	14.56	16.16	18.15	20.75	22.62	24.19
SPIHT-Codec-II, 8×16 LOT Typ II	14.44	16.00	18.00	20.55	22.39	23.88
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	14.42	15.96	17.92	20.41	22.19	23.69
SPIHT-Codec-II, 8×24 LCT-3	14.42	15.84	17.80	20.29	21.97	23.58
AGU-Codec-II, 16×48 LCT-3	14.51	16.00	17.91	20.42	22.19	23.72
SPIHT-Codec-II, 16×48 LCT-3	14.50	15.90	17.85	20.35	22.08	23.61
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	14.66	16.27	18.27	20.84	22.71	24.28
SPIHT-Codec-II, 8×32 LCT-4	14.61	16.14	18.08	20.68	22.45	24.07
AGU-Codec-II, 16×64 LCT-4	14.74	16.31	18.26	20.82	22.68	24.25
SPIHT-Codec-II, 16×64 LCT-4	14.72	16.20	18.16	20.71	22.53	24.14
L-CEB-Codec-II, 8×32 GenLOT	14.58	16.17	18.19	20.78	22.66	24.25
SPIHT-Codec-II, 8×32 GenLOT	14.48	16.00	18.02	20.56	22.41	23.92
L-CEB-Codec-II, 8×32 ELT	14.67	16.29	18.30	20.88	22.75	24.33
SPIHT-Codec-II, 8×32 ELT	14.63	16.14	18.14	20.69	22.51	24.02
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	14.50	16.06	18.03	20.57	22.41	23.96
SPIHT-Codec-II, 8×48 LCT-6	14.54	16.02	17.92	20.40	22.20	23.78
AGU-Codec-II, 16×96 LCT-6	14.55	16.09	18.01	20.54	22.38	23.91
SPIHT-Codec-II, 16×96 LCT-6	14.57	16.04	17.95	20.46	22.28	23.79
L-CEB-Codec-II, 8×48 GenLOT	14.75	16.31	18.28	20.83	22.68	24.24
SPIHT-Codec-II, 8×48 GenLOT	14.66	16.15	18.13	20.63	22.45	23.94
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	14.66	16.12	18.02	20.36	22.08	23.60
AGU-Codec-II, 9/7-Wavelet	14.62	16.05	17.98	20.41	22.07	23.61
SPIHT, 9/7-Wavelet	14.61	16.03	17.95	20.39	22.03	23.54
JPEG2000	14.39	16.00	17.83	20.50	22.01	23.60

Tabelle H.29: Kompressionsergebnisse für das Testbild „Bark“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	14.54	16.05	17.94	20.44	22.29	23.84
SPIHT-Codec-II pp, 8×8 DCT	14.30	15.83	17.58	20.07	21.96	23.41
AGU-Codec-II pp, 16×16 DCT	14.62	16.14	18.08	20.60	22.48	24.05
SPIHT-Codec-II pp, 16×16 DCT	14.44	16.01	17.99	20.36	22.26	23.76
AGU-Codec-II pp, 32×32 DCT	14.75	16.27	18.26	20.81	22.69	24.29
SPIHT-Codec-II pp, 32×32 DCT	14.53	16.11	18.12	20.53	22.44	23.95
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	14.66	16.26	18.26	20.86	22.76	24.34
SPIHT-Codec-II pp, 8×16 LCT-2	14.46	16.05	18.11	20.54	22.48	23.93
AGU-Codec-II pp, 16×32 LCT-2	14.75	16.32	18.29	20.86	22.77	24.34
SPIHT-Codec-II pp, 16×32 LCT-2	14.60	16.18	18.20	20.64	22.54	24.08
AGU-Codec-II pp, 32×64 LCT-2	14.82	16.39	18.36	20.92	22.80	24.39
SPIHT-Codec-II pp, 32×64 LCT-2	14.66	16.24	18.22	20.66	22.54	24.10
L-CEB-Codec-II pp, 8×16 TDLT	14.60	16.25	18.23	20.82	22.68	24.23
SPIHT-Codec-II pp, 8×16 TDLT	14.49	15.99	17.98	20.54	22.39	23.88
L-CEB-Codec-II pp, 8×16 LOT Typ II	14.67	16.26	18.24	20.84	22.71	24.29
SPIHT-Codec-II pp, 8×16 LOT Typ II	14.43	16.05	18.10	20.48	22.46	23.87
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	14.56	16.10	18.05	20.54	22.34	23.86
SPIHT-Codec-II pp, 8×24 LCT-3	14.45	15.82	17.81	20.34	21.96	23.66
AGU-Codec-II pp, 16×48 LCT-3	14.57	16.05	17.97	20.49	22.28	23.81
SPIHT-Codec-II pp, 16×48 LCT-3	14.51	15.83	17.82	20.37	22.04	23.67
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	14.73	16.33	18.30	20.88	22.75	24.33
SPIHT-Codec-II pp, 8×32 LCT-4	14.56	16.10	18.10	20.55	22.44	23.99
AGU-Codec-II pp, 16×64 LCT-4	14.72	16.29	18.25	20.81	22.69	24.27
SPIHT-Codec-II pp, 16×64 LCT-4	14.74	16.12	18.15	20.72	22.49	24.02
L-CEB-Codec-II pp, 8×32 GenLOT	14.71	16.28	18.27	20.84	22.73	24.33
SPIHT-Codec-II pp, 8×32 GenLOT	14.48	16.06	18.10	20.49	22.45	23.89
L-CEB-Codec-II pp, 8×32 ELT	14.74	16.34	18.34	20.92	22.78	24.37
SPIHT-Codec-II pp, 8×32 ELT	14.56	16.13	18.18	20.59	22.52	23.98
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	14.61	16.16	18.12	20.66	22.52	24.07
SPIHT-Codec-II pp, 8×48 LCT-6	14.56	15.97	17.91	20.45	22.16	23.84
AGU-Codec-II pp, 16×96 LCT-6	14.59	16.11	18.04	20.58	22.44	23.98
SPIHT-Codec-II pp, 16×96 LCT-6	14.58	15.92	17.91	20.48	22.20	23.84
L-CEB-Codec-II pp, 8×48 GenLOT	14.79	16.36	18.32	20.90	22.76	24.35
SPIHT-Codec-II pp, 8×48 GenLOT	14.56	16.14	18.19	20.57	22.51	23.94
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	14.64	16.12	18.04	20.42	22.21	23.75
AGU-Codec-II pp, 9/7-Wavelet	14.59	16.04	17.99	20.44	22.16	23.73
SPIHT pp, 9/7-Wavelet	14.50	15.99	18.01	20.35	22.11	23.56

Tabelle H.30: Kompressionsergebnisse für das Testbild „Bark“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Wood Grain“

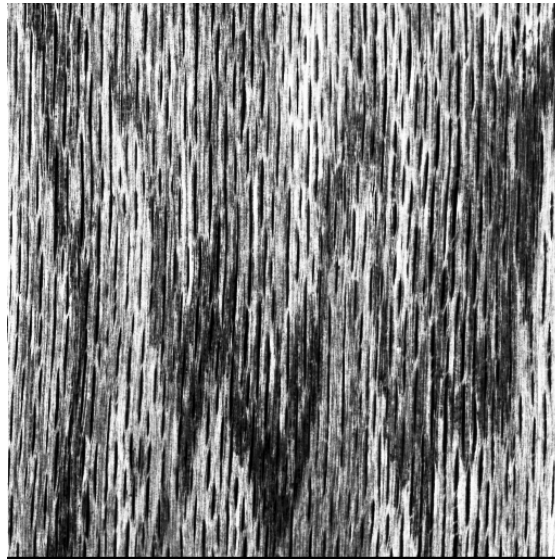


Abbildung H.121: Testbild „Wood Grain“ (512×512 , $H = 6.224$, $\sigma^2 = 5299.0572$, $\bar{g} = 124.9629$, $g_{min} = 0$, $g_{max} = 255$).

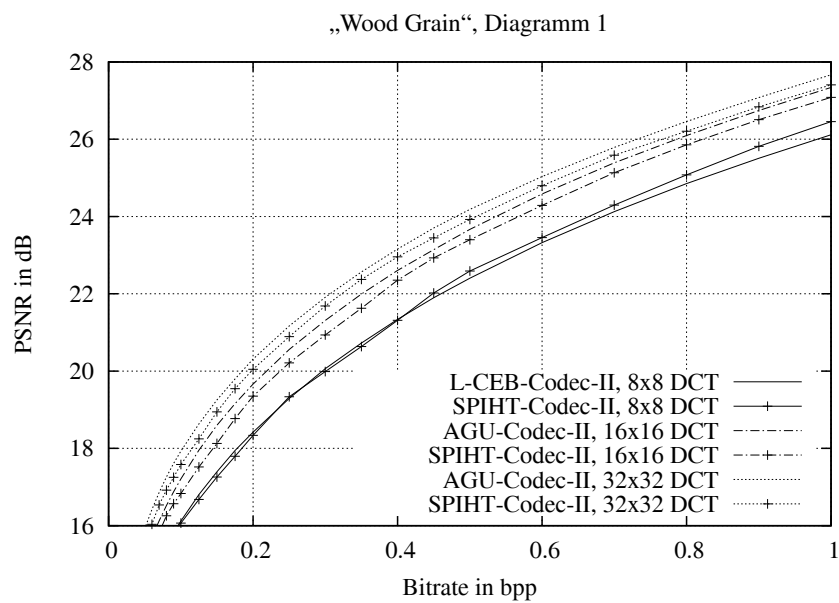


Abbildung H.122: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

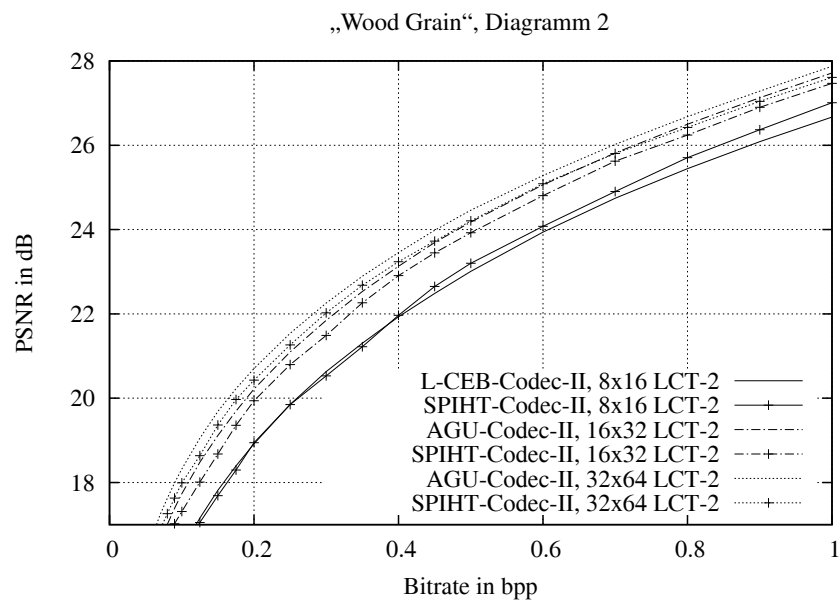


Abbildung H.123: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

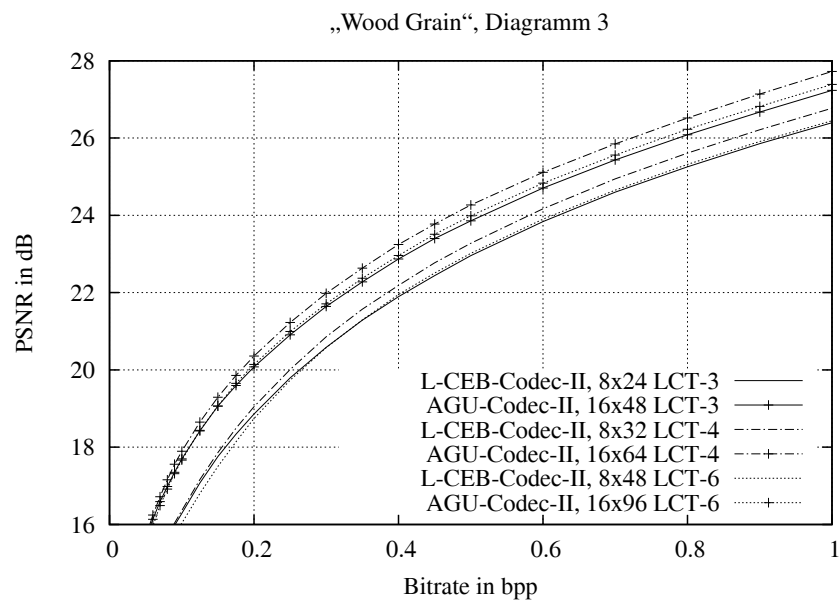


Abbildung H.124: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

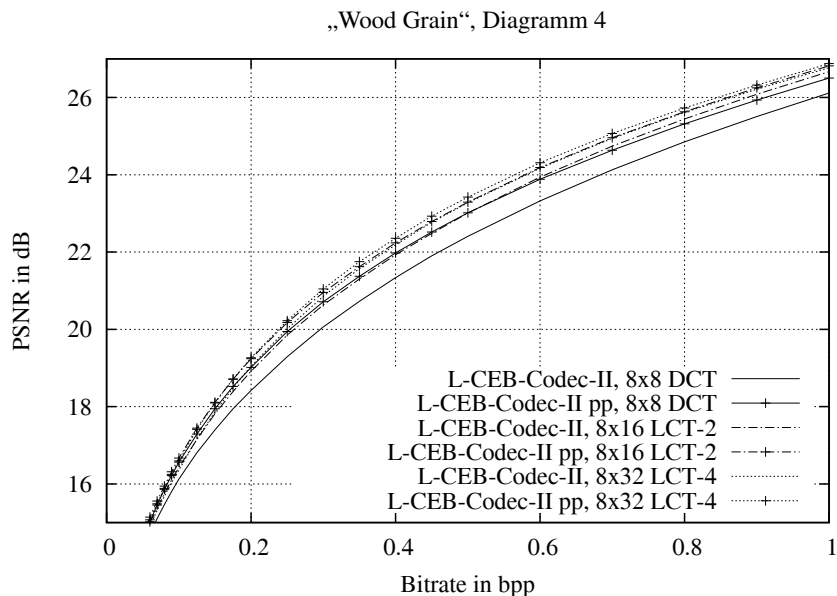


Abbildung H.125: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

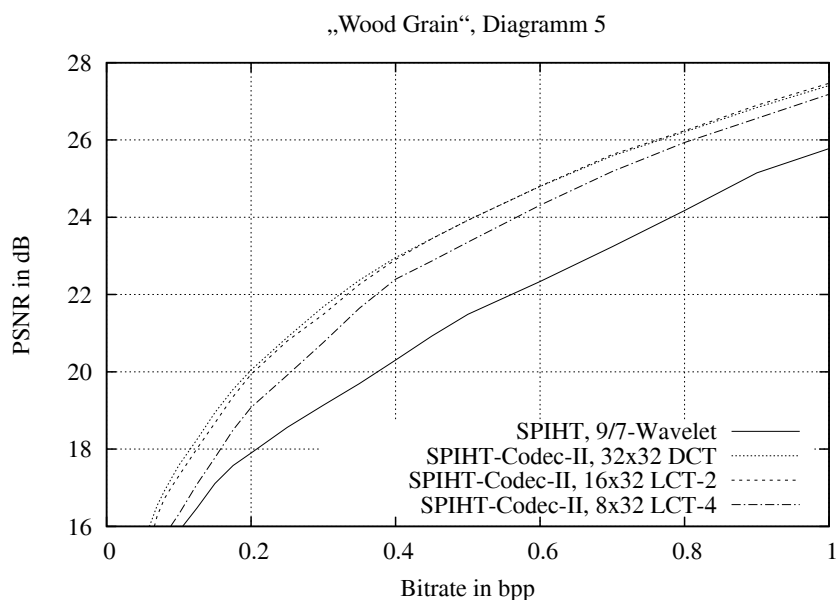


Abbildung H.126: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

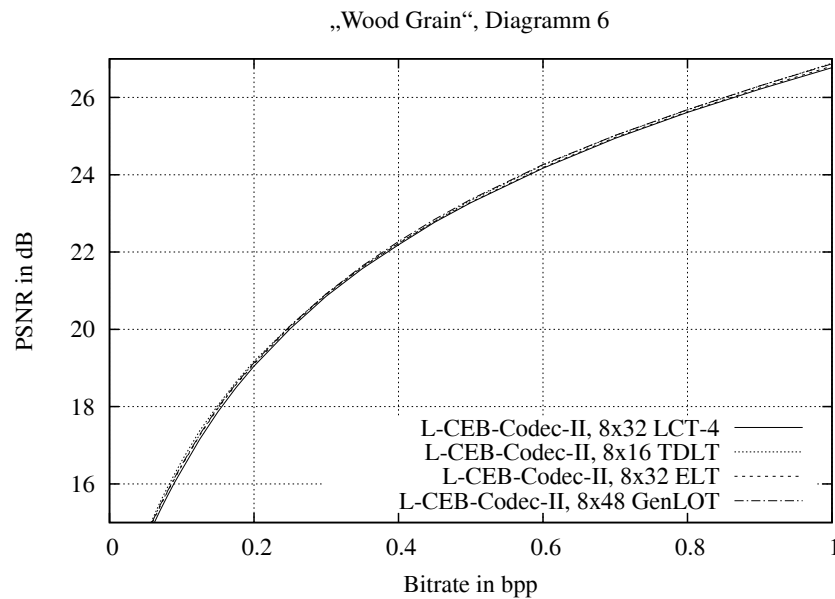


Abbildung H.127: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

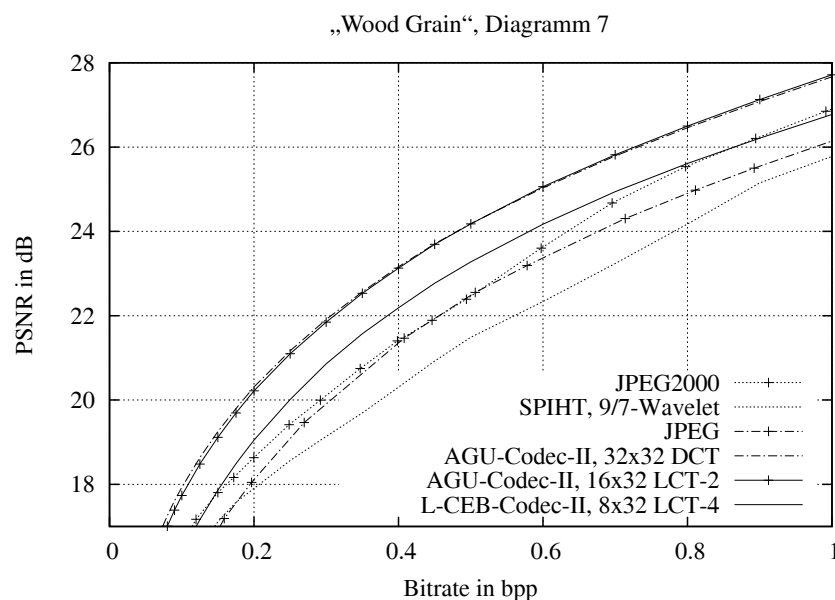


Abbildung H.128: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	14.81	16.81	19.29	22.41	24.50	26.12
SPIHT-Codec-II, 8×8 DCT	14.65	16.68	19.34	22.59	24.67	26.46
AGU-Codec-II, 16×16 DCT	15.80	17.95	20.56	23.67	25.75	27.34
SPIHT-Codec-II, 16×16 DCT	15.36	17.52	20.21	23.40	25.52	27.08
AGU-Codec-II, 32×32 DCT	16.54	18.64	21.17	24.19	26.14	27.67
SPIHT-Codec-II, 32×32 DCT	16.17	18.25	20.89	23.92	25.91	27.41
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	15.10	17.15	19.85	23.01	25.10	26.67
SPIHT-Codec-II, 8×16 LCT-2	14.99	17.05	19.85	23.20	25.34	27.01
AGU-Codec-II, 16×32 LCT-2	16.26	18.48	21.10	24.18	26.16	27.71
SPIHT-Codec-II, 16×32 LCT-2	15.75	18.01	20.80	23.92	25.94	27.47
AGU-Codec-II, 32×64 LCT-2	16.89	19.05	21.54	24.46	26.36	27.87
SPIHT-Codec-II, 32×64 LCT-2	16.50	18.64	21.26	24.21	26.10	27.61
L-CEB-Codec-II, 8×16 TDLT	15.20	17.38	20.09	23.34	25.35	26.89
SPIHT-Codec-II, 8×16 TDLT	14.95	17.33	20.24	23.56	25.57	27.19
L-CEB-Codec-II, 8×16 LOT Typ II	15.08	17.16	19.75	22.95	25.05	26.62
SPIHT-Codec-II, 8×16 LOT Typ II	14.94	17.02	19.75	23.13	25.25	26.94
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	14.95	17.08	19.79	22.96	24.94	26.40
SPIHT-Codec-II, 8×24 LCT-3	14.94	17.15	19.83	23.05	25.22	26.74
AGU-Codec-II, 16×48 LCT-3	16.26	18.42	20.91	23.86	25.75	27.23
SPIHT-Codec-II, 16×48 LCT-3	15.69	18.04	20.60	23.60	25.63	27.05
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	15.03	17.16	20.01	23.28	25.28	26.77
SPIHT-Codec-II, 8×32 LCT-4	15.05	17.09	19.91	23.36	25.54	27.18
AGU-Codec-II, 16×64 LCT-4	16.37	18.65	21.23	24.27	26.19	27.72
SPIHT-Codec-II, 16×64 LCT-4	16.04	18.18	20.83	23.97	25.97	27.49
L-CEB-Codec-II, 8×32 GenLOT	15.10	17.24	19.92	23.13	25.16	26.72
SPIHT-Codec-II, 8×32 GenLOT	14.93	17.11	19.92	23.27	25.38	27.05
L-CEB-Codec-II, 8×32 ELT	15.13	17.26	20.06	23.29	25.31	26.81
SPIHT-Codec-II, 8×32 ELT	15.02	17.15	20.05	23.44	25.56	27.16
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	14.65	16.80	19.73	23.02	24.99	26.45
SPIHT-Codec-II, 8×48 LCT-6	14.74	16.78	19.64	23.10	25.29	26.88
AGU-Codec-II, 16×96 LCT-6	16.09	18.44	20.99	23.99	25.90	27.39
SPIHT-Codec-II, 16×96 LCT-6	15.70	18.11	20.73	23.78	25.74	27.20
L-CEB-Codec-II, 8×48 GenLOT	15.15	17.28	20.09	23.36	25.37	26.87
SPIHT-Codec-II, 8×48 GenLOT	15.00	17.20	20.10	23.50	25.61	27.20
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	14.67	16.31	18.27	20.80	23.00	24.92
AGU-Codec-II, 9/7-Wavelet	14.64	16.43	18.46	21.12	23.55	25.55
SPIHT, 9/7-Wavelet	14.80	16.48	18.56	21.49	23.68	25.78
JPEG2000	15.04	17.17	19.42	22.39	25.16	26.85

Tabelle H.31: Kompressionsergebnisse für das Testbild „Wood Grain“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	15.13	17.29	19.94	23.01	24.98	26.50
SPIHT-Codec-II pp, 8×8 DCT	14.74	17.09	19.87	23.02	25.04	26.65
AGU-Codec-II pp, 16×16 DCT	16.01	18.25	20.87	23.96	25.97	27.51
SPIHT-Codec-II pp, 16×16 DCT	15.39	17.76	20.49	23.64	25.63	27.21
AGU-Codec-II pp, 32×32 DCT	16.54	18.73	21.30	24.33	26.25	27.75
SPIHT-Codec-II pp, 32×32 DCT	16.01	18.28	21.00	24.04	25.90	27.47
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	15.26	17.45	20.17	23.29	25.30	26.82
SPIHT-Codec-II pp, 8×16 LCT-2	14.98	17.23	20.07	23.37	25.43	27.08
AGU-Codec-II pp, 16×32 LCT-2	16.26	18.54	21.18	24.27	26.23	27.78
SPIHT-Codec-II pp, 16×32 LCT-2	15.65	18.04	20.85	23.98	25.92	27.49
AGU-Codec-II pp, 32×64 LCT-2	16.74	18.98	21.53	24.51	26.40	27.91
SPIHT-Codec-II pp, 32×64 LCT-2	16.20	18.53	21.25	24.22	26.04	27.63
L-CEB-Codec-II pp, 8×16 TDLT	15.21	17.45	20.18	23.42	25.43	26.97
SPIHT-Codec-II pp, 8×16 TDLT	14.93	17.39	20.23	23.56	25.60	27.19
L-CEB-Codec-II pp, 8×16 LOT Typ II	15.22	17.43	20.11	23.28	25.30	26.83
SPIHT-Codec-II pp, 8×16 LOT Typ II	14.90	17.23	20.03	23.36	25.38	27.08
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	15.09	17.29	20.05	23.20	25.17	26.62
SPIHT-Codec-II pp, 8×24 LCT-3	14.99	17.34	19.95	23.18	25.33	26.80
AGU-Codec-II pp, 16×48 LCT-3	16.15	18.42	21.00	23.99	25.91	27.39
SPIHT-Codec-II pp, 16×48 LCT-3	15.61	18.09	20.56	23.63	25.69	27.07
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	15.18	17.41	20.22	23.42	25.40	26.88
SPIHT-Codec-II pp, 8×32 LCT-4	15.01	17.21	20.04	23.44	25.55	27.21
AGU-Codec-II pp, 16×64 LCT-4	16.27	18.60	21.25	24.33	26.26	27.78
SPIHT-Codec-II pp, 16×64 LCT-4	15.75	18.09	20.81	23.98	26.00	27.50
L-CEB-Codec-II pp, 8×32 GenLOT	15.27	17.49	20.22	23.39	25.36	26.87
SPIHT-Codec-II pp, 8×32 GenLOT	14.91	17.29	20.15	23.45	25.47	27.13
L-CEB-Codec-II pp, 8×32 ELT	15.24	17.49	20.26	23.46	25.44	26.93
SPIHT-Codec-II pp, 8×32 ELT	14.97	17.28	20.20	23.54	25.60	27.23
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	14.89	17.09	19.98	23.19	25.18	26.62
SPIHT-Codec-II pp, 8×48 LCT-6	14.99	16.91	19.79	23.21	25.38	26.92
AGU-Codec-II pp, 16×96 LCT-6	15.99	18.39	21.05	24.10	26.02	27.51
SPIHT-Codec-II pp, 16×96 LCT-6	15.73	17.87	20.65	23.77	25.79	27.20
L-CEB-Codec-II pp, 8×48 GenLOT	15.28	17.49	20.30	23.52	25.49	26.98
SPIHT-Codec-II pp, 8×48 GenLOT	14.97	17.34	20.26	23.60	25.63	27.26
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	14.77	16.52	18.57	21.31	23.52	25.33
AGU-Codec-II pp, 9/7-Wavelet	14.70	16.62	18.67	21.58	24.02	25.87
SPIHT pp, 9/7-Wavelet	14.82	16.59	18.71	21.92	23.91	26.00

Tabelle H.32: Kompressionsergebnisse für das Testbild „Wood Grain“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Sand“

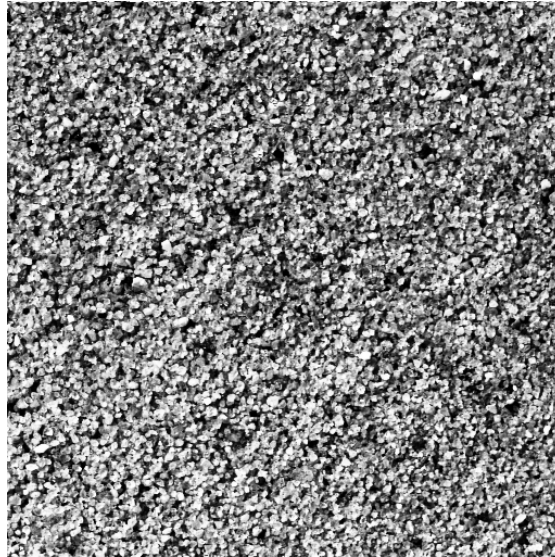


Abbildung H.129: Testbild „Sand“ (512×512 , $H = 6.5097$, $\sigma^2 = 5527.8657$, $\bar{g} = 129.4034$, $g_{min} = 0$, $g_{max} = 255$).

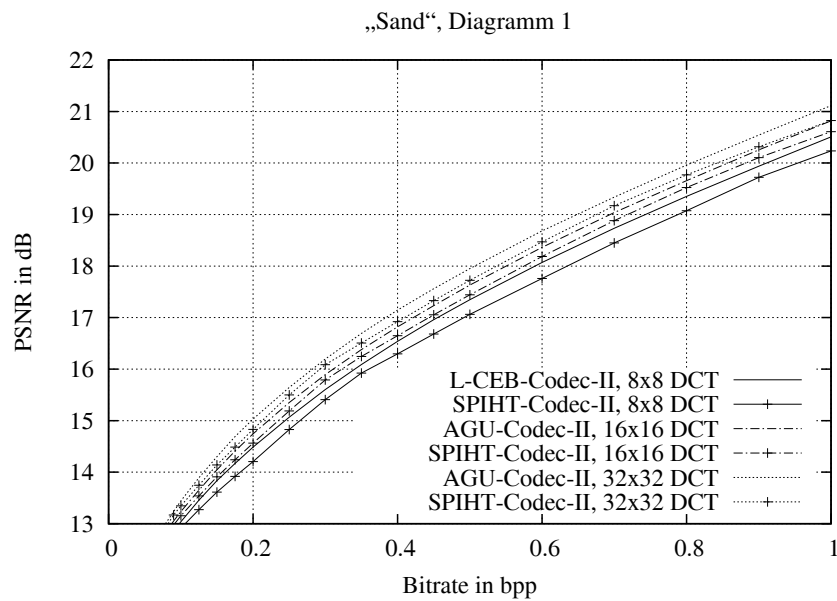


Abbildung H.130: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

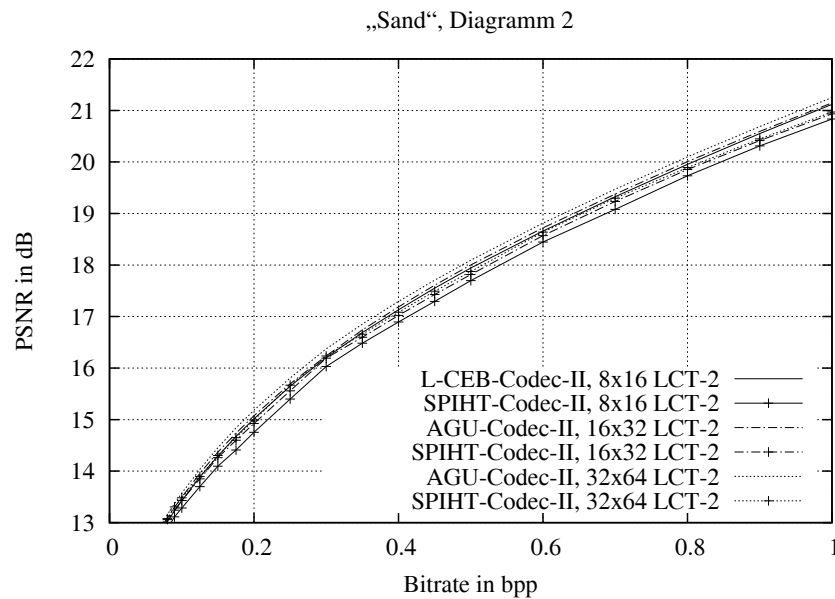


Abbildung H.131: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

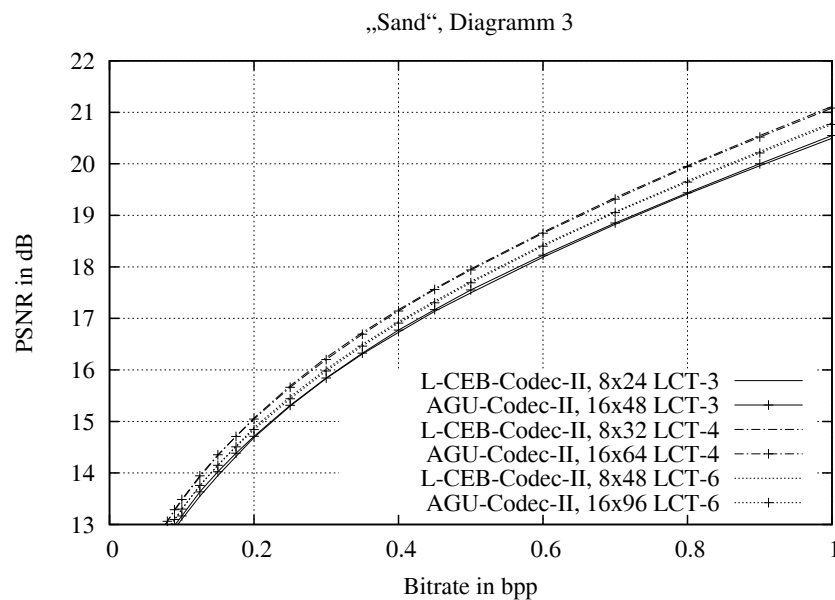


Abbildung H.132: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

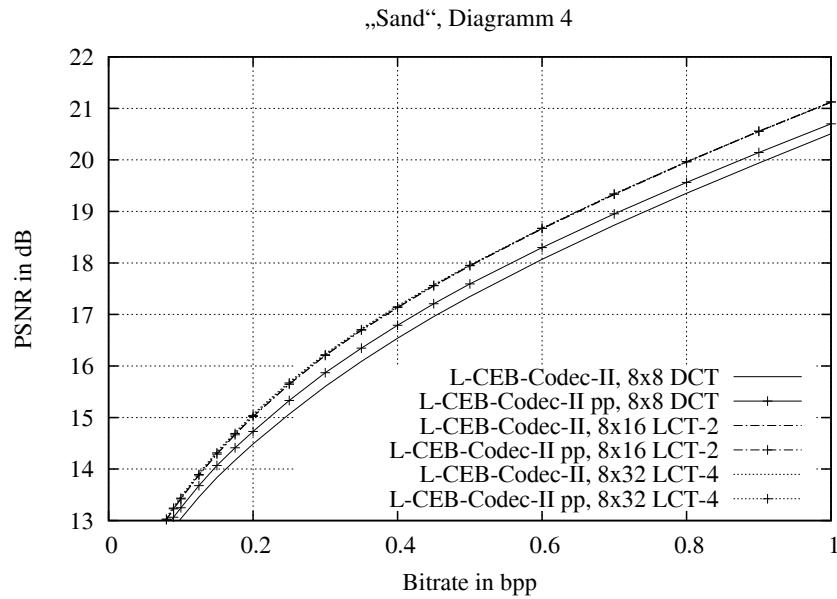


Abbildung H.133: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

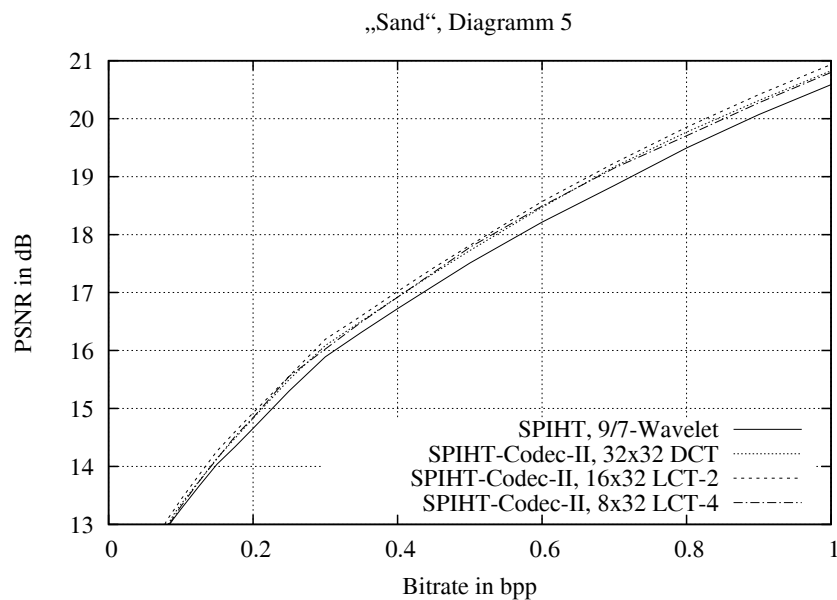


Abbildung H.134: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

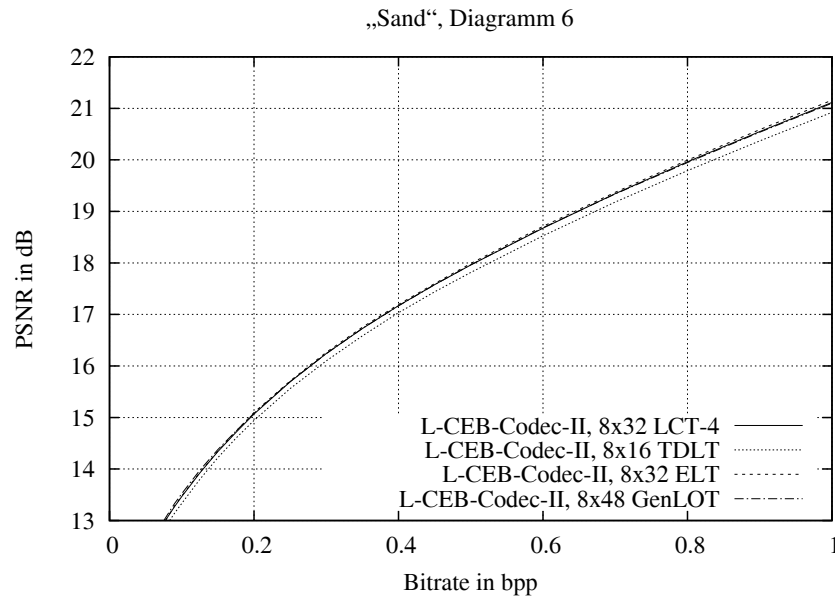


Abbildung H.135: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

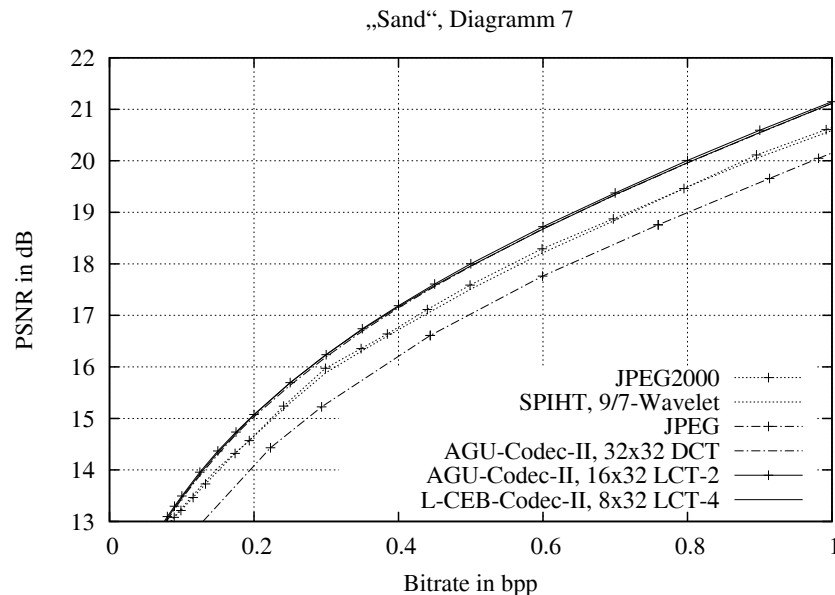


Abbildung H.136: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	12.35	13.46	15.07	17.35	19.05	20.51
SPIHT-Codec-II, 8×8 DCT	12.25	13.27	14.83	17.06	18.75	20.23
AGU-Codec-II, 16×16 DCT	12.49	13.65	15.35	17.63	19.35	20.82
SPIHT-Codec-II, 16×16 DCT	12.38	13.54	15.19	17.44	19.21	20.61
AGU-Codec-II, 32×32 DCT	12.62	13.89	15.65	17.95	19.65	21.11
SPIHT-Codec-II, 32×32 DCT	12.51	13.74	15.50	17.72	19.48	20.83
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	12.63	13.89	15.65	17.94	19.65	21.12
SPIHT-Codec-II, 8×16 LCT-2	12.48	13.70	15.40	17.70	19.40	20.83
AGU-Codec-II, 16×32 LCT-2	12.70	13.96	15.69	18.00	19.70	21.15
SPIHT-Codec-II, 16×32 LCT-2	12.58	13.85	15.56	17.82	19.56	20.93
AGU-Codec-II, 32×64 LCT-2	12.74	14.03	15.81	18.09	19.78	21.25
SPIHT-Codec-II, 32×64 LCT-2	12.59	13.89	15.67	17.88	19.62	20.97
L-CEB-Codec-II, 8×16 TDLT	12.52	13.80	15.56	17.82	19.49	20.92
SPIHT-Codec-II, 8×16 TDLT	12.40	13.67	15.28	17.62	19.24	20.72
L-CEB-Codec-II, 8×16 LOT Typ II	12.63	13.90	15.62	17.92	19.62	21.10
SPIHT-Codec-II, 8×16 LOT Typ II	12.50	13.70	15.40	17.66	19.38	20.80
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	12.34	13.55	15.30	17.50	19.12	20.49
SPIHT-Codec-II, 8×24 LCT-3	12.36	13.48	15.18	17.30	19.02	20.25
AGU-Codec-II, 16×48 LCT-3	12.38	13.63	15.31	17.55	19.15	20.55
SPIHT-Codec-II, 16×48 LCT-3	12.41	13.58	15.26	17.39	19.07	20.41
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	12.66	13.92	15.69	17.96	19.66	21.11
SPIHT-Codec-II, 8×32 LCT-4	12.58	13.74	15.56	17.78	19.43	20.80
AGU-Codec-II, 16×64 LCT-4	12.67	13.95	15.66	17.94	19.64	21.08
SPIHT-Codec-II, 16×64 LCT-4	12.61	13.84	15.60	17.83	19.47	20.87
L-CEB-Codec-II, 8×32 GenLOT	12.61	13.87	15.60	17.92	19.63	21.09
SPIHT-Codec-II, 8×32 GenLOT	12.47	13.68	15.37	17.65	19.37	20.80
L-CEB-Codec-II, 8×32 ELT	12.67	13.94	15.71	17.99	19.68	21.16
SPIHT-Codec-II, 8×32 ELT	12.52	13.76	15.47	17.75	19.44	20.87
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	12.48	13.75	15.48	17.72	19.38	20.80
SPIHT-Codec-II, 8×48 LCT-6	12.45	13.60	15.31	17.54	19.19	20.49
AGU-Codec-II, 16×96 LCT-6	12.50	13.76	15.44	17.69	19.35	20.77
SPIHT-Codec-II, 16×96 LCT-6	12.46	13.67	15.35	17.55	19.21	20.60
L-CEB-Codec-II, 8×48 GenLOT	12.70	13.98	15.70	17.96	19.65	21.10
SPIHT-Codec-II, 8×48 GenLOT	12.56	13.80	15.48	17.72	19.41	20.81
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	12.65	13.78	15.43	17.56	19.18	20.66
AGU-Codec-II, 9/7-Wavelet	12.60	13.73	15.38	17.58	19.14	20.63
SPIHT, 9/7-Wavelet	12.55	13.66	15.30	17.51	19.17	20.59
JPEG2000	12.35	13.46	15.24	17.59	19.17	20.61

Tabelle H.33: Kompressionsergebnisse für das Testbild „Sand“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	12.53	13.68	15.33	17.59	19.27	20.70
SPIHT-Codec-II pp, 8×8 DCT	12.29	13.54	14.98	17.30	18.84	20.39
AGU-Codec-II pp, 16×16 DCT	12.57	13.72	15.44	17.71	19.41	20.88
SPIHT-Codec-II pp, 16×16 DCT	12.35	13.66	15.18	17.51	19.14	20.67
AGU-Codec-II pp, 32×32 DCT	12.61	13.86	15.61	17.92	19.63	21.10
SPIHT-Codec-II pp, 32×32 DCT	12.42	13.77	15.37	17.70	19.31	20.84
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	12.63	13.88	15.65	17.95	19.65	21.13
SPIHT-Codec-II pp, 8×16 LCT-2	12.40	13.72	15.31	17.68	19.28	20.85
AGU-Codec-II pp, 16×32 LCT-2	12.66	13.87	15.64	17.96	19.66	21.12
SPIHT-Codec-II pp, 16×32 LCT-2	12.47	13.83	15.44	17.77	19.40	20.92
AGU-Codec-II pp, 32×64 LCT-2	12.66	13.92	15.71	18.02	19.71	21.19
SPIHT-Codec-II pp, 32×64 LCT-2	12.47	13.85	15.49	17.81	19.42	20.95
L-CEB-Codec-II pp, 8×16 TDLT	12.48	13.75	15.55	17.83	19.54	20.99
SPIHT-Codec-II pp, 8×16 TDLT	12.38	13.55	15.27	17.65	19.25	20.69
L-CEB-Codec-II pp, 8×16 LOT Typ II	12.63	13.87	15.62	17.92	19.63	21.12
SPIHT-Codec-II pp, 8×16 LOT Typ II	12.41	13.73	15.28	17.65	19.23	20.84
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	12.42	13.61	15.36	17.57	19.22	20.60
SPIHT-Codec-II pp, 8×24 LCT-3	12.37	13.41	15.22	17.21	19.04	20.22
AGU-Codec-II pp, 16×48 LCT-3	12.44	13.61	15.29	17.54	19.17	20.58
SPIHT-Codec-II pp, 16×48 LCT-3	12.42	13.48	15.26	17.26	19.06	20.35
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	12.65	13.89	15.67	17.95	19.65	21.12
SPIHT-Codec-II pp, 8×32 LCT-4	12.47	13.71	15.40	17.71	19.44	20.79
AGU-Codec-II pp, 16×64 LCT-4	12.62	13.85	15.59	17.89	19.59	21.05
SPIHT-Codec-II pp, 16×64 LCT-4	12.47	13.78	15.40	17.73	19.47	20.83
L-CEB-Codec-II pp, 8×32 GenLOT	12.64	13.88	15.62	17.93	19.64	21.11
SPIHT-Codec-II pp, 8×32 GenLOT	12.40	13.72	15.28	17.64	19.24	20.81
L-CEB-Codec-II pp, 8×32 ELT	12.65	13.90	15.68	17.98	19.68	21.16
SPIHT-Codec-II pp, 8×32 ELT	12.42	13.76	15.34	17.71	19.31	20.87
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	12.54	13.76	15.49	17.74	19.42	20.85
SPIHT-Codec-II pp, 8×48 LCT-6	12.47	13.50	15.33	17.41	19.20	20.44
AGU-Codec-II pp, 16×96 LCT-6	12.51	13.70	15.39	17.67	19.34	20.78
SPIHT-Codec-II pp, 16×96 LCT-6	12.47	13.56	15.34	17.38	19.20	20.51
L-CEB-Codec-II pp, 8×48 GenLOT	12.68	13.95	15.70	17.97	19.67	21.14
SPIHT-Codec-II pp, 8×48 GenLOT	12.46	13.81	15.38	17.72	19.31	20.85
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	12.58	13.71	15.38	17.54	19.21	20.70
AGU-Codec-II pp, 9/7-Wavelet	12.53	13.66	15.33	17.54	19.15	20.65
SPIHT pp, 9/7-Wavelet	12.42	13.62	15.16	17.48	19.07	20.65

Tabelle H.34: Kompressionsergebnisse für das Testbild „Sand“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Water“

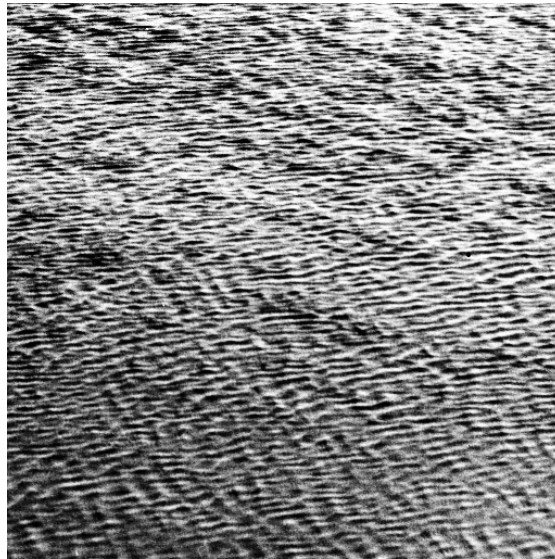


Abbildung H.137: Testbild „Water“ (512×512 , $H = 5.653$, $\sigma^2 = 5467.5326$, $\bar{g} = 130.0839$, $g_{min} = 0$, $g_{max} = 255$).

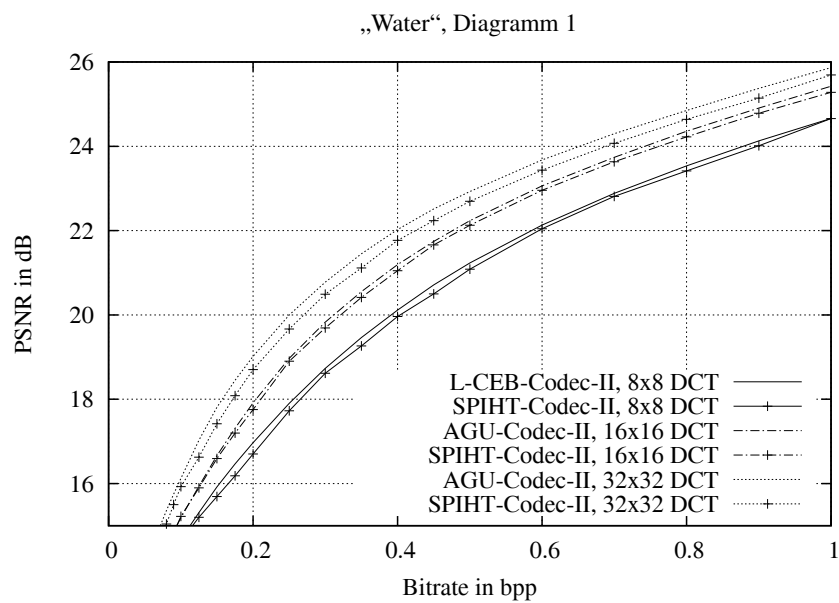


Abbildung H.138: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

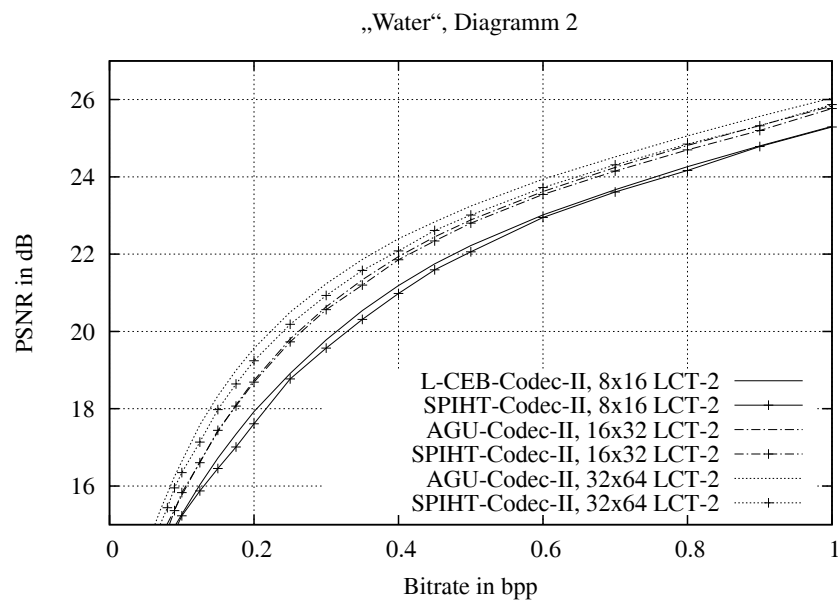


Abbildung H.139: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

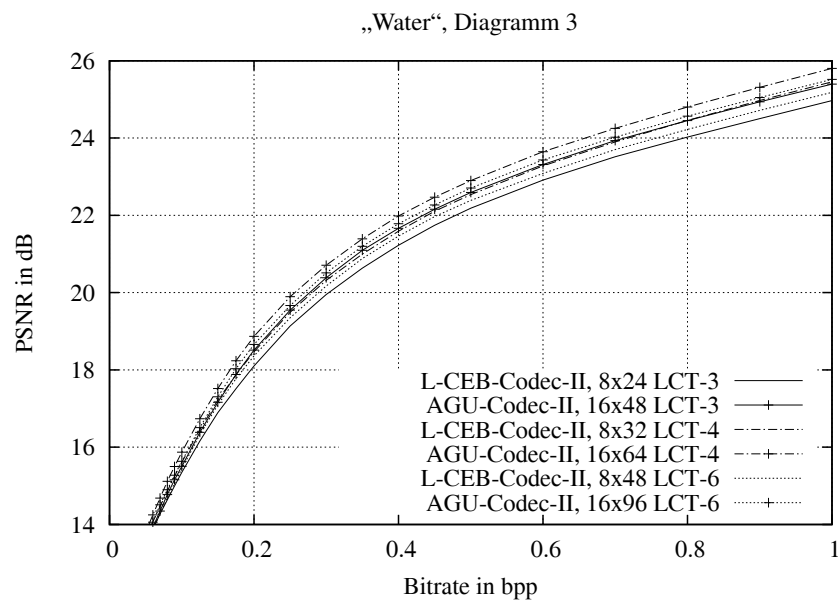


Abbildung H.140: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

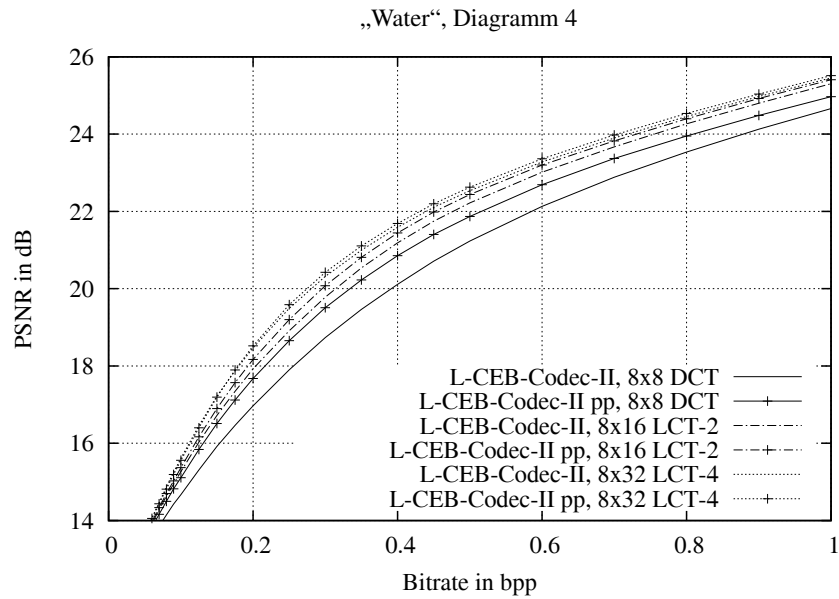


Abbildung H.141: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

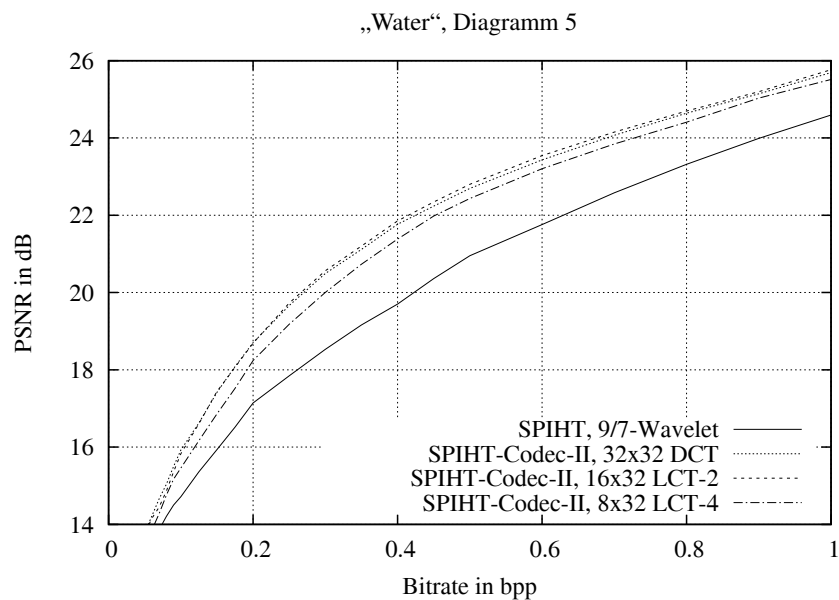


Abbildung H.142: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

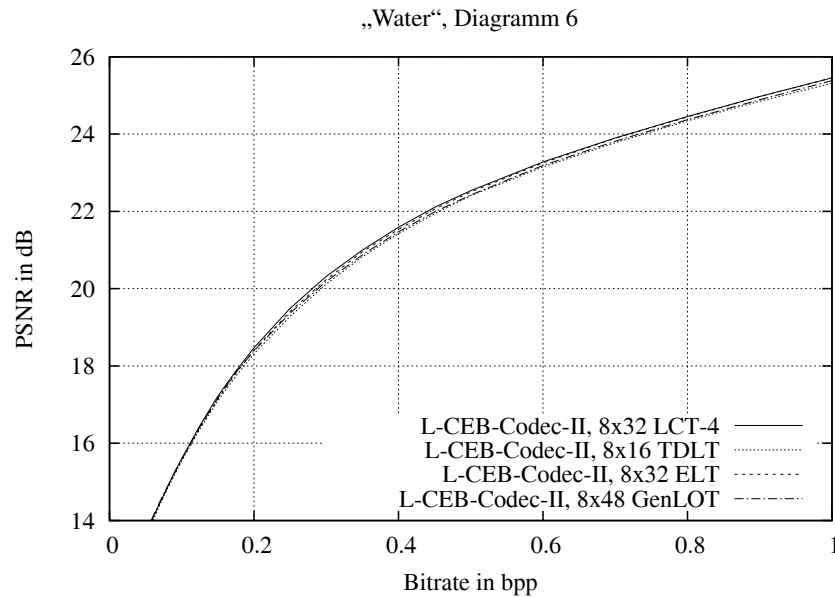


Abbildung H.143: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

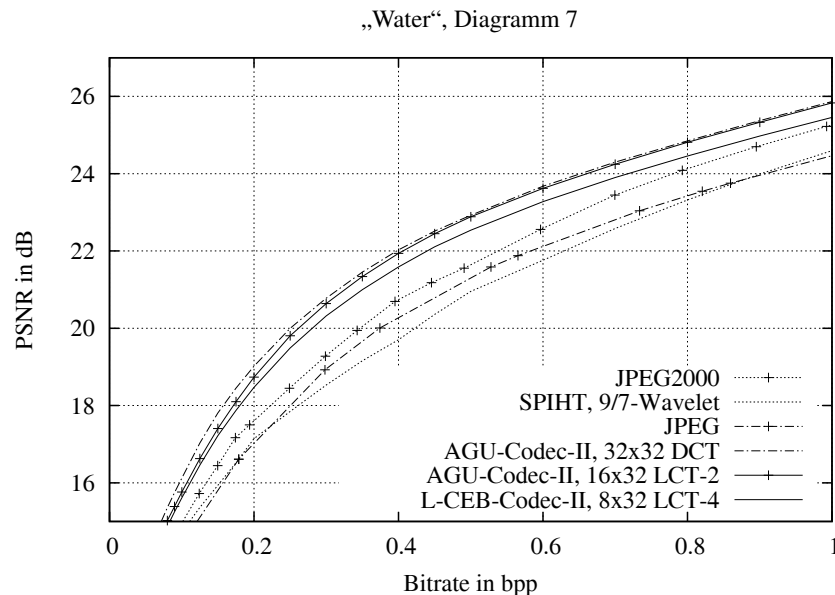


Abbildung H.144: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	13.62	15.31	17.90	21.23	23.21	24.66
SPIHT-Codec-II, 8×8 DCT	13.41	15.20	17.73	21.09	23.15	24.66
AGU-Codec-II, 16×16 DCT	13.90	15.94	18.97	22.24	24.06	25.43
SPIHT-Codec-II, 16×16 DCT	13.77	15.90	18.90	22.13	23.92	25.28
AGU-Codec-II, 32×32 DCT	14.62	17.03	20.00	22.93	24.58	25.87
SPIHT-Codec-II, 32×32 DCT	14.34	16.62	19.67	22.69	24.35	25.69
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	14.00	16.03	18.91	22.22	23.98	25.31
SPIHT-Codec-II, 8×16 LCT-2	13.74	15.87	18.77	22.06	23.88	25.29
AGU-Codec-II, 16×32 LCT-2	14.30	16.63	19.80	22.89	24.53	25.83
SPIHT-Codec-II, 16×32 LCT-2	14.23	16.60	19.73	22.80	24.41	25.77
AGU-Codec-II, 32×64 LCT-2	14.99	17.58	20.50	23.25	24.80	26.04
SPIHT-Codec-II, 32×64 LCT-2	14.63	17.14	20.18	23.01	24.59	25.87
L-CEB-Codec-II, 8×16 TDLT	14.10	16.37	19.29	22.41	24.07	25.32
SPIHT-Codec-II, 8×16 TDLT	13.91	16.00	18.92	22.32	24.07	25.39
L-CEB-Codec-II, 8×16 LOT Typ II	13.99	16.02	18.83	22.13	23.88	25.19
SPIHT-Codec-II, 8×16 LOT Typ II	13.73	15.84	18.70	21.98	23.79	25.19
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	13.97	16.16	19.14	22.19	23.77	24.97
SPIHT-Codec-II, 8×24 LCT-3	13.89	16.01	18.79	22.09	23.87	25.07
AGU-Codec-II, 16×48 LCT-3	14.03	16.38	19.55	22.59	24.20	25.40
SPIHT-Codec-II, 16×48 LCT-3	14.11	16.28	19.42	22.54	24.14	25.35
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	14.17	16.46	19.49	22.54	24.19	25.46
SPIHT-Codec-II, 8×32 LCT-4	13.94	16.17	19.18	22.43	24.14	25.52
AGU-Codec-II, 16×64 LCT-4	14.35	16.74	19.90	22.90	24.53	25.80
SPIHT-Codec-II, 16×64 LCT-4	14.25	16.80	19.87	22.84	24.38	25.70
L-CEB-Codec-II, 8×32 GenLOT	14.09	16.18	18.99	22.16	23.95	25.28
SPIHT-Codec-II, 8×32 GenLOT	13.81	16.01	18.83	22.00	23.84	25.26
L-CEB-Codec-II, 8×32 ELT	14.16	16.40	19.41	22.51	24.17	25.45
SPIHT-Codec-II, 8×32 ELT	13.92	16.24	19.23	22.42	24.10	25.49
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	13.97	16.32	19.35	22.39	23.96	25.18
SPIHT-Codec-II, 8×48 LCT-6	13.93	16.04	19.04	22.30	24.04	25.24
AGU-Codec-II, 16×96 LCT-6	14.17	16.51	19.67	22.71	24.30	25.52
SPIHT-Codec-II, 16×96 LCT-6	14.32	16.58	19.61	22.69	24.27	25.47
L-CEB-Codec-II, 8×48 GenLOT	14.20	16.43	19.37	22.43	24.10	25.38
SPIHT-Codec-II, 8×48 GenLOT	13.95	16.25	19.22	22.37	24.06	25.44
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	13.52	15.20	17.62	20.56	22.53	24.09
AGU-Codec-II, 9/7-Wavelet	13.63	15.50	17.94	20.93	22.98	24.57
SPIHT, 9/7-Wavelet	13.63	15.36	17.85	20.96	22.96	24.59
JPEG2000	13.56	15.72	18.45	21.55	23.78	25.22

Tabelle H.35: Kompressionsergebnisse für das Testbild „Water“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	13.92	15.84	18.65	21.87	23.66	24.97
SPIHT-Codec-II pp, 8×8 DCT	13.51	15.72	18.27	21.63	23.51	24.83
AGU-Codec-II pp, 16×16 DCT	14.02	16.14	19.29	22.51	24.23	25.54
SPIHT-Codec-II pp, 16×16 DCT	13.79	16.17	19.00	22.28	24.05	25.28
AGU-Codec-II pp, 32×32 DCT	14.52	16.97	20.09	23.03	24.64	25.90
SPIHT-Codec-II pp, 32×32 DCT	14.21	16.69	19.81	22.79	24.35	25.59
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	14.06	16.17	19.20	22.44	24.12	25.41
SPIHT-Codec-II pp, 8×16 LCT-2	13.70	16.04	18.90	22.18	23.96	25.26
AGU-Codec-II pp, 16×32 LCT-2	14.19	16.50	19.80	22.93	24.56	25.85
SPIHT-Codec-II pp, 16×32 LCT-2	14.08	16.58	19.76	22.83	24.38	25.65
AGU-Codec-II pp, 32×64 LCT-2	14.72	17.32	20.41	23.25	24.80	26.04
SPIHT-Codec-II pp, 32×64 LCT-2	14.39	17.00	20.15	23.02	24.52	25.88
L-CEB-Codec-II pp, 8×16 TDLT	13.99	16.27	19.34	22.51	24.16	25.41
SPIHT-Codec-II pp, 8×16 TDLT	13.87	15.83	18.92	22.36	24.12	25.41
L-CEB-Codec-II pp, 8×16 LOT Typ II	14.05	16.16	19.13	22.40	24.10	25.36
SPIHT-Codec-II pp, 8×16 LOT Typ II	13.69	16.02	18.79	22.12	23.95	25.20
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	13.98	16.18	19.30	22.38	23.96	25.16
SPIHT-Codec-II pp, 8×24 LCT-3	13.89	15.86	18.83	22.15	23.96	25.13
AGU-Codec-II pp, 16×48 LCT-3	13.97	16.23	19.50	22.66	24.29	25.49
SPIHT-Codec-II pp, 16×48 LCT-3	14.10	16.08	19.36	22.54	24.19	25.37
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	14.15	16.40	19.59	22.63	24.26	25.51
SPIHT-Codec-II pp, 8×32 LCT-4	13.79	16.17	19.26	22.48	24.12	25.54
AGU-Codec-II pp, 16×64 LCT-4	14.17	16.51	19.82	22.92	24.56	25.82
SPIHT-Codec-II pp, 16×64 LCT-4	14.01	16.65	19.84	22.83	24.31	25.71
L-CEB-Codec-II pp, 8×32 GenLOT	14.14	16.31	19.29	22.41	24.11	25.39
SPIHT-Codec-II pp, 8×32 GenLOT	13.76	16.16	18.92	22.14	23.94	25.24
L-CEB-Codec-II pp, 8×32 ELT	14.15	16.38	19.52	22.62	24.26	25.52
SPIHT-Codec-II pp, 8×32 ELT	13.82	16.29	19.16	22.42	24.11	25.42
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	13.98	16.23	19.43	22.51	24.10	25.32
SPIHT-Codec-II pp, 8×48 LCT-6	14.00	15.91	19.06	22.33	24.11	25.27
AGU-Codec-II pp, 16×96 LCT-6	14.04	16.27	19.59	22.77	24.38	25.60
SPIHT-Codec-II pp, 16×96 LCT-6	14.27	16.30	19.51	22.66	24.31	25.46
L-CEB-Codec-II pp, 8×48 GenLOT	14.18	16.40	19.52	22.58	24.22	25.48
SPIHT-Codec-II pp, 8×48 GenLOT	13.85	16.30	19.17	22.41	24.10	25.39
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	13.56	15.30	17.86	20.98	22.91	24.38
AGU-Codec-II pp, 9/7-Wavelet	13.65	15.59	18.11	21.30	23.29	24.77
SPIHT pp, 9/7-Wavelet	13.55	15.44	18.04	21.27	23.10	24.68

Tabelle H.36: Kompressionsergebnisse für das Testbild „Water“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

Testbild „Texture“

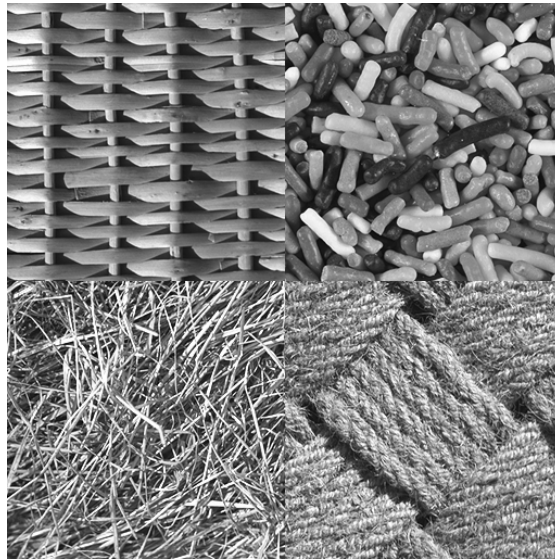


Abbildung H.145: Testbild „Texture“ (512×512 , $H = 7.7845$, $\sigma^2 = 3453.4771$, $\bar{g} = 125.8761$, $g_{min} = 3$, $g_{max} = 255$).

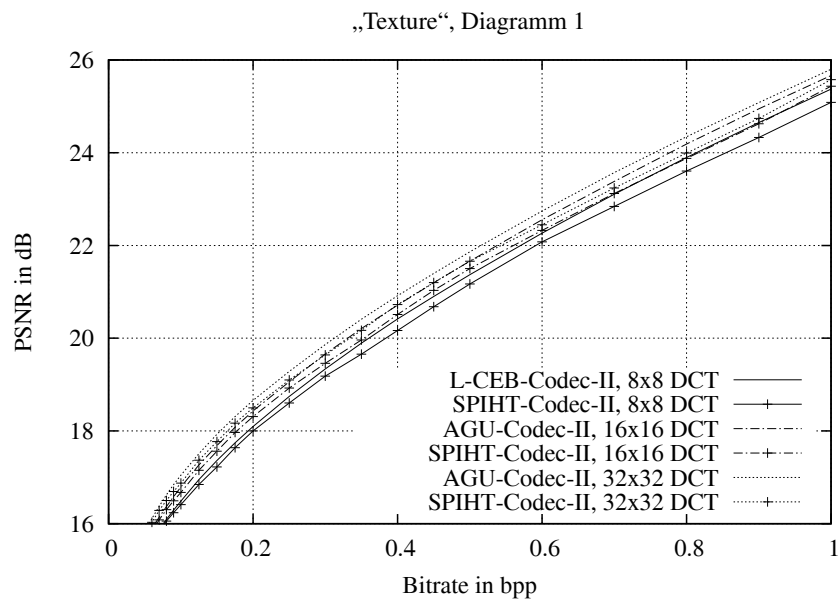


Abbildung H.146: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung der Kosinustransformation.

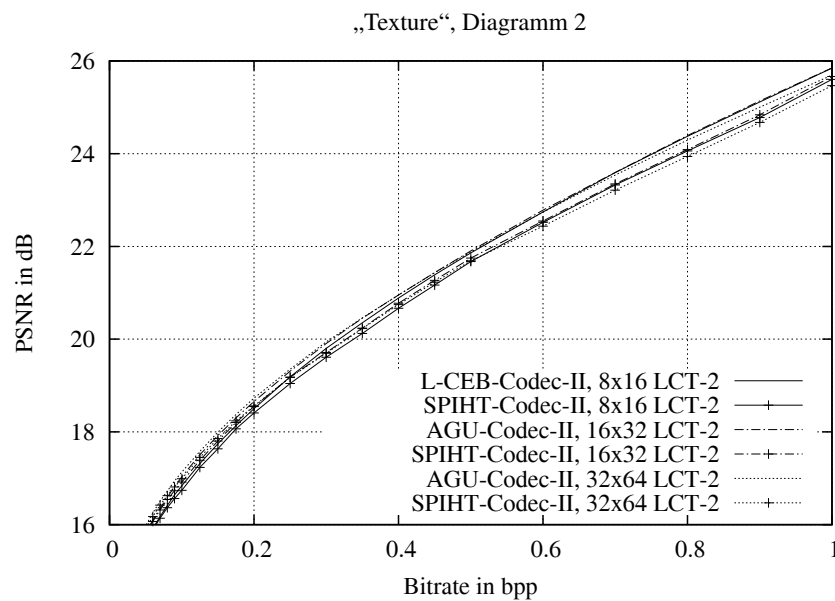


Abbildung H.147: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von zweifach überlappenden lokalen trigonometrischen Basen.

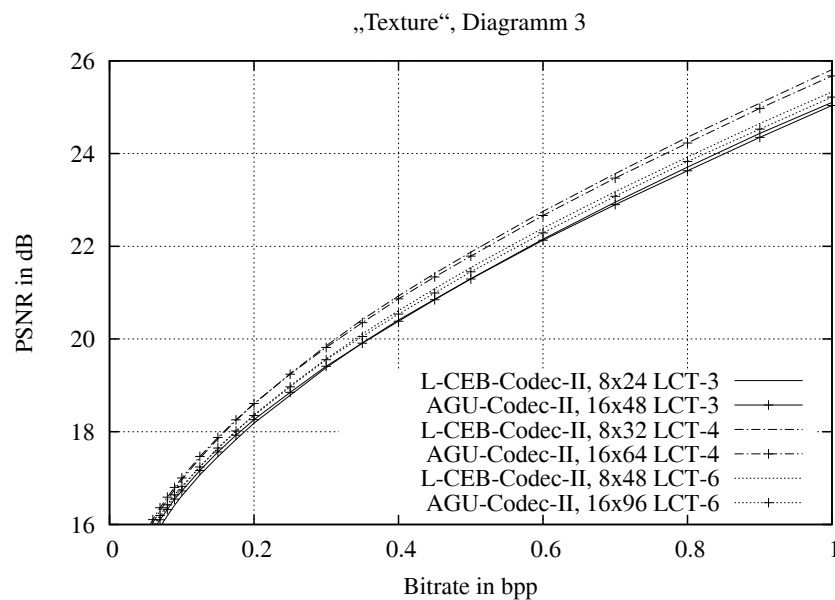


Abbildung H.148: Vergleich der Kompressionsleistung ausgewählter Codecs unter Verwendung von mehrfach überlappenden lokalen trigonometrischen Basen.

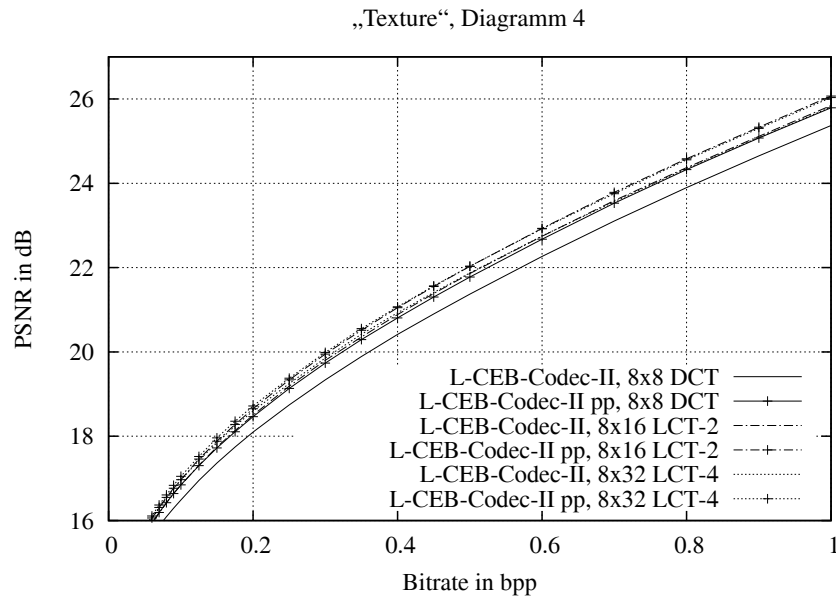


Abbildung H.149: L-CEB-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$, pp: mit Post Processing Filter.

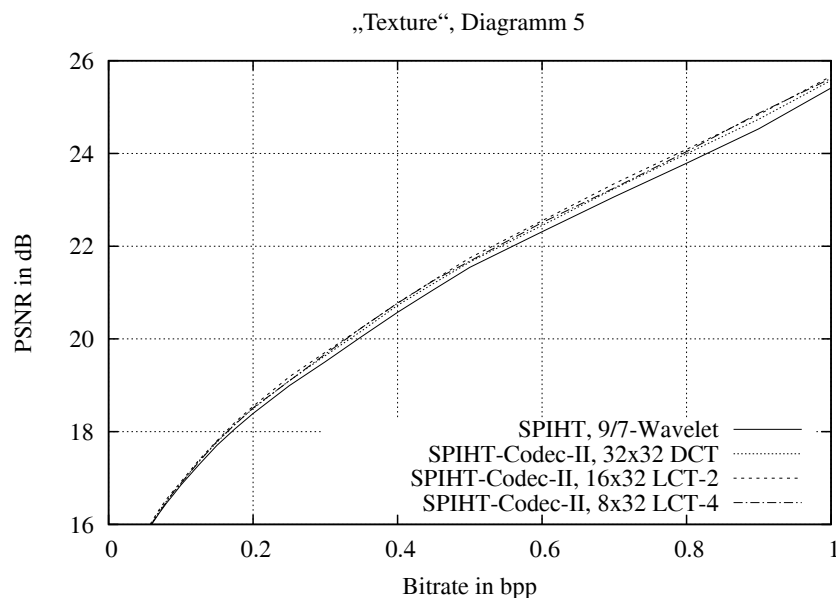


Abbildung H.150: SPIHT-Codec-II: Vergleich der Kompressionsleistung für verschiedene Überlappungsfaktoren bei gleicher Länge der trigonometrischen Basisfunktionen ($l = 32$) und Vergleich mit der DWT.

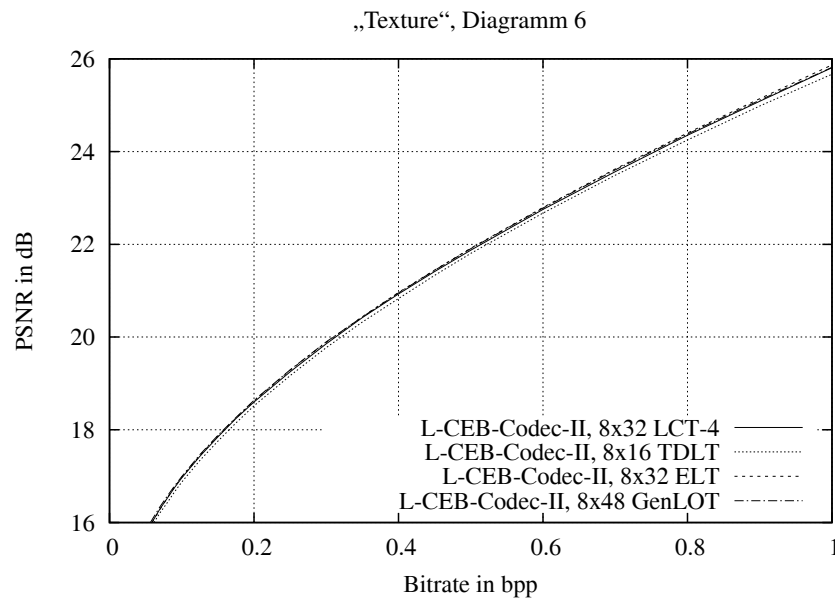


Abbildung H.151: L-CEB-Codec-II: Vergleich der Leistungsfähigkeit von vierfach überlappenden lokalen trigonometrischen Basen mit verschiedenen blocküberlappenden State-of-the-Art Transformationsverfahren bei gleicher Blockgröße B der Transformationskoeffizienten, $B = 8 \times 8$.

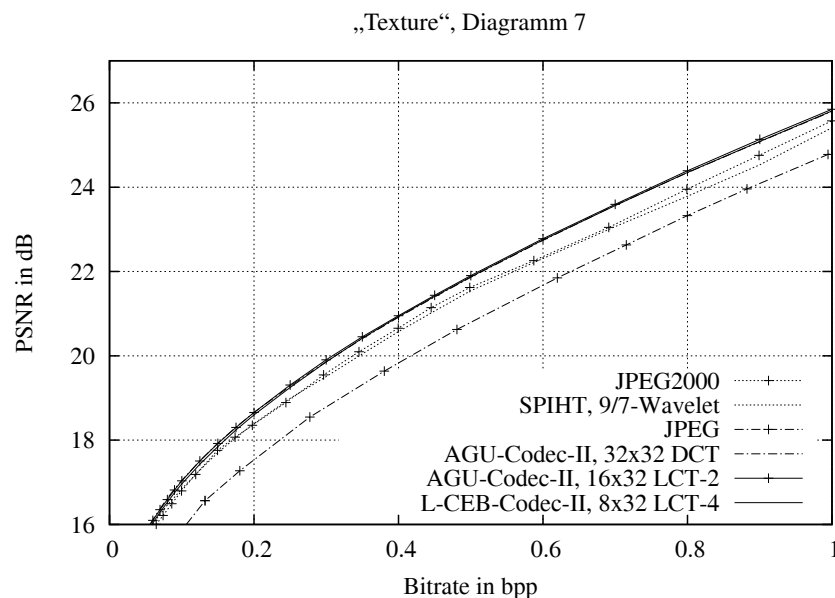


Abbildung H.152: Kompressionsleistung ausgewählter Codecs unter Verwendung der leistungsfähigsten trigonometrischen Basen im Vergleich mit State-of-the-Art Kompressionsverfahren.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II, 8×8 DCT	15.70	16.95	18.74	21.37	23.50	25.37
SPIHT-Codec-II, 8×8 DCT	15.70	16.84	18.60	21.17	23.25	25.09
AGU-Codec-II, 16×16 DCT	15.96	17.27	19.06	21.66	23.79	25.66
SPIHT-Codec-II, 16×16 DCT	15.91	17.15	18.92	21.50	23.50	25.43
AGU-Codec-II, 32×32 DCT	16.19	17.51	19.28	21.86	23.96	25.80
SPIHT-Codec-II, 32×32 DCT	16.09	17.37	19.10	21.66	23.61	25.58
Zweifachüberlappung						
L-CEB-Codec-II, 8×16 LCT-2	15.98	17.31	19.19	21.86	24.00	25.85
SPIHT-Codec-II, 8×16 LCT-2	15.95	17.23	19.05	21.67	23.70	25.60
AGU-Codec-II, 16×32 LCT-2	16.16	17.51	19.31	21.90	23.99	25.85
SPIHT-Codec-II, 16×32 LCT-2	16.13	17.39	19.18	21.75	23.72	25.66
AGU-Codec-II, 32×64 LCT-2	16.32	17.58	19.35	21.88	23.92	25.69
SPIHT-Codec-II, 32×64 LCT-2	16.24	17.45	19.18	21.69	23.58	25.47
L-CEB-Codec-II, 8×16 TDLT	16.02	17.34	19.17	21.80	23.87	25.67
SPIHT-Codec-II, 8×16 TDLT	15.99	17.27	19.07	21.52	23.63	25.37
L-CEB-Codec-II, 8×16 LOT Typ II	15.99	17.31	19.18	21.85	23.95	25.78
SPIHT-Codec-II, 8×16 LOT Typ II	15.96	17.25	19.06	21.66	23.69	25.57
Dreifachüberlappung						
L-CEB-Codec-II, 8×24 LCT-3	15.78	17.07	18.79	21.30	23.32	25.10
SPIHT-Codec-II, 8×24 LCT-3	15.85	17.16	18.76	21.22	23.10	24.98
AGU-Codec-II, 16×48 LCT-3	15.92	17.17	18.85	21.29	23.28	25.04
SPIHT-Codec-II, 16×48 LCT-3	15.94	17.19	18.78	21.20	23.07	24.87
Vierfachüberlappung						
L-CEB-Codec-II, 8×32 LCT-4	16.11	17.43	19.25	21.87	23.97	25.81
SPIHT-Codec-II, 8×32 LCT-4	16.08	17.35	19.09	21.68	23.64	25.62
AGU-Codec-II, 16×64 LCT-4	16.17	17.47	19.24	21.79	23.86	25.68
SPIHT-Codec-II, 16×64 LCT-4	16.14	17.40	19.08	21.61	23.58	25.48
L-CEB-Codec-II, 8×32 GenLOT	16.03	17.36	19.24	21.91	24.04	25.90
SPIHT-Codec-II, 8×32 GenLOT	16.02	17.28	19.10	21.71	23.75	25.67
L-CEB-Codec-II, 8×32 ELT	16.10	17.44	19.29	21.92	24.01	25.87
SPIHT-Codec-II, 8×32 ELT	16.08	17.38	19.17	21.74	23.73	25.62
Sechsfachüberlappung						
L-CEB-Codec-II, 8×48 LCT-6	15.98	17.24	18.99	21.54	23.55	25.33
SPIHT-Codec-II, 8×48 LCT-6	16.00	17.26	18.91	21.35	23.36	25.12
AGU-Codec-II, 16×96 LCT-6	16.01	17.24	18.97	21.45	23.46	25.22
SPIHT-Codec-II, 16×96 LCT-6	16.00	17.25	18.87	21.30	23.32	25.03
L-CEB-Codec-II, 8×48 GenLOT	16.15	17.46	19.30	21.91	24.00	25.81
SPIHT-Codec-II, 8×48 GenLOT	16.15	17.41	19.18	21.75	23.72	25.60
Wavelets						
L-CEB-Codec-II, 9/7-Wavelet	16.08	17.32	19.03	21.50	23.53	25.35
AGU-Codec-II, 9/7-Wavelet	16.10	17.35	19.03	21.52	23.52	25.37
SPIHT, 9/7-Wavelet	16.07	17.29	18.99	21.55	23.41	25.41
JPEG2000	15.88	17.19	18.89	21.62	23.48	25.57

Tabelle H.37: Kompressionsergebnisse für das Testbild „Texture“, Vergleich der Codecs bei ausgewählten Bitraten, PSNR in dB.

Transformationscodec	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
Keine Überlappung						
L-CEB-Codec-II pp, 8×8 DCT	16.01	17.30	19.13	21.78	23.94	25.79
SPIHT-Codec-II pp, 8×8 DCT	15.99	17.07	18.93	21.37	23.56	25.35
AGU-Codec-II pp, 16×16 DCT	16.13	17.46	19.27	21.91	24.07	25.95
SPIHT-Codec-II pp, 16×16 DCT	16.07	17.22	19.09	21.56	23.66	25.59
AGU-Codec-II pp, 32×32 DCT	16.26	17.57	19.39	22.02	24.16	26.04
SPIHT-Codec-II pp, 32×32 DCT	16.17	17.35	19.18	21.63	23.72	25.66
Zweifachüberlappung						
L-CEB-Codec-II pp, 8×16 LCT-2	16.12	17.46	19.34	22.03	24.19	26.07
SPIHT-Codec-II pp, 8×16 LCT-2	16.06	17.27	19.15	21.68	23.81	25.69
AGU-Codec-II pp, 16×32 LCT-2	16.22	17.56	19.40	22.03	24.17	26.06
SPIHT-Codec-II pp, 16×32 LCT-2	16.17	17.36	19.23	21.73	23.79	25.75
AGU-Codec-II pp, 32×64 LCT-2	16.31	17.61	19.43	22.02	24.11	25.94
SPIHT-Codec-II pp, 32×64 LCT-2	16.25	17.40	19.22	21.67	23.67	25.58
L-CEB-Codec-II pp, 8×16 TDLT	16.04	17.39	19.28	21.97	24.07	25.88
SPIHT-Codec-II pp, 8×16 TDLT	15.95	17.29	19.07	21.59	23.72	25.48
L-CEB-Codec-II pp, 8×16 LOT Typ II	16.12	17.45	19.32	22.04	24.20	26.07
SPIHT-Codec-II pp, 8×16 LOT Typ II	16.07	17.25	19.17	21.64	23.84	25.70
Dreifachüberlappung						
L-CEB-Codec-II pp, 8×24 LCT-3	15.95	17.26	19.03	21.60	23.66	25.47
SPIHT-Codec-II pp, 8×24 LCT-3	15.89	17.26	18.82	21.36	23.25	25.18
AGU-Codec-II pp, 16×48 LCT-3	16.04	17.29	19.04	21.55	23.59	25.40
SPIHT-Codec-II pp, 16×48 LCT-3	15.97	17.27	18.84	21.32	23.23	25.06
Vierfachüberlappung						
L-CEB-Codec-II pp, 8×32 LCT-4	16.18	17.52	19.38	22.04	24.16	26.03
SPIHT-Codec-II pp, 8×32 LCT-4	16.10	17.32	19.14	21.77	23.72	25.65
AGU-Codec-II pp, 16×64 LCT-4	16.21	17.51	19.33	21.93	24.06	25.92
SPIHT-Codec-II pp, 16×64 LCT-4	16.14	17.33	19.10	21.69	23.66	25.60
L-CEB-Codec-II pp, 8×32 GenLOT	16.16	17.49	19.38	22.08	24.25	26.13
SPIHT-Codec-II pp, 8×32 GenLOT	16.11	17.29	19.19	21.69	23.85	25.74
L-CEB-Codec-II pp, 8×32 ELT	16.17	17.53	19.41	22.07	24.21	26.09
SPIHT-Codec-II pp, 8×32 ELT	16.13	17.36	19.24	21.72	23.82	25.70
Sechsfachüberlappung						
L-CEB-Codec-II pp, 8×48 LCT-6	16.10	17.40	19.20	21.79	23.85	25.67
SPIHT-Codec-II pp, 8×48 LCT-6	16.00	17.35	18.95	21.48	23.48	25.31
AGU-Codec-II pp, 16×96 LCT-6	16.10	17.36	19.13	21.68	23.76	25.56
SPIHT-Codec-II pp, 16×96 LCT-6	15.99	17.33	18.89	21.42	23.44	25.22
L-CEB-Codec-II pp, 8×48 GenLOT	16.20	17.55	19.41	22.09	24.24	26.10
SPIHT-Codec-II pp, 8×48 GenLOT	16.19	17.37	19.26	21.76	23.86	25.74
Wavelets						
L-CEB-Codec-II pp, 9/7-Wavelet	16.09	17.35	19.10	21.62	23.72	25.57
AGU-Codec-II pp, 9/7-Wavelet	16.10	17.36	19.06	21.59	23.66	25.56
SPIHT pp, 9/7-Wavelet	16.06	17.24	19.03	21.52	23.50	25.49

Tabelle H.38: Kompressionsergebnisse für das Testbild „Texture“, Vergleich der Codecs mit Postprocessing Filter (pp) bei ausgewählten Bitraten, PSNR in dB.

H.2 Quervergleiche zur objektiven Beurteilung der Leistungsfähigkeit

H.2.1 Kosinustransformation

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.55	1.03	1.32	1.32	1.20	1.02
„Lena“	0.47	0.64	0.65	0.46	0.39	0.24
„Zelda“	0.85	1.15	0.97	0.59	0.43	0.45
„Boat“	0.40	0.41	0.38	0.32	0.26	0.25
„Goldhill“	0.38	0.41	0.41	0.34	0.26	0.29
„Lighthouse“	0.68	0.61	0.45	0.31	0.31	0.26
„Roof“	1.03	1.11	1.03	1.25	1.24	1.15
„Clown“	0.58	0.96	0.97	0.71	0.52	0.43
„Houses“	0.36	0.36	0.31	0.18	-0.05	-0.12
„Mandrill“	0.11	0.17	0.25	0.32	0.27	0.32
„Fingerprint“	0.84	1.43	1.45	1.50	1.63	1.68
„Peppers“	0.47	0.52	0.38	0.06	0.00	-0.03
„F16“	0.47	0.53	0.47	0.28	0.19	0.15
„Bubble“	0.29	0.37	0.39	0.36	0.26	0.22
„Bark“	0.29	0.39	0.41	0.49	0.45	0.53
„Wood Grain“	0.71	0.84	0.87	0.81	0.85	0.62
„Sand“	0.13	0.27	0.36	0.38	0.46	0.38
„Water“	0.36	0.70	1.17	1.04	0.77	0.62
„Texture“	0.21	0.31	0.32	0.33	0.25	0.34

Tabelle H.39: $\Delta\text{PSNR} = \text{PSNR}_{\text{SPIHT-Codec-II, } 16 \times 16 \text{ DCT}} - \text{PSNR}_{\text{SPIHT-Codec-II, } 8 \times 8 \text{ DCT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.87	0.65	0.52	0.33	0.19	0.16
„Lena“	0.33	0.20	0.05	-0.04	-0.10	-0.10
„Zelda“	0.91	0.63	0.31	0.22	0.19	0.22
„Boat“	0.18	0.03	-0.07	-0.20	-0.25	-0.29
„Goldhill“	0.28	0.14	0.09	0.03	-0.01	-0.01
„Lighthouse“	0.41	0.31	0.06	-0.06	-0.12	-0.18
„Roof“	0.86	0.72	0.70	0.31	0.17	0.01
„Clown“	0.45	0.48	0.20	-0.05	-0.15	-0.16
„Houses“	0.22	0.19	0.14	-0.02	-0.16	-0.25
„Mandrill“	0.12	0.12	0.08	0.05	-0.01	-0.02
„Fingerprint“	0.92	0.90	0.95	1.00	1.03	1.06
„Peppers“	0.30	0.04	-0.33	-0.43	-0.33	-0.35
„F16“	0.25	0.03	-0.12	-0.31	-0.33	-0.43
„Bubble“	0.26	0.28	0.19	0.12	0.07	0.04
„Bark“	0.21	0.22	0.23	0.27	0.24	0.27
„Wood Grain“	0.81	0.73	0.68	0.52	0.39	0.33
„Sand“	0.13	0.20	0.31	0.28	0.27	0.22
„Water“	0.57	0.72	0.77	0.56	0.43	0.41
„Texture“	0.18	0.22	0.18	0.16	0.11	0.15

Tabelle H.40: $\Delta\text{PSNR} = \text{PSNR}_{\text{SPIHT-Codec-II, } 32 \times 32 \text{ DCT}} - \text{PSNR}_{\text{SPIHT-Codec-II, } 16 \times 16 \text{ DCT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.21	0.36	0.39	0.46	0.36	0.23
„Lena“	0.13	0.14	0.10	-0.03	-0.06	-0.18
„Zelda“	0.06	0.14	0.08	0.01	-0.13	-0.13
„Boat“	0.09	0.17	0.17	0.16	0.08	-0.02
„Goldhill“	0.11	0.06	0.08	0.06	0.02	0.03
„Lighthouse“	0.23	0.22	0.13	0.09	0.06	0.09
„Roof“	0.67	0.51	0.50	0.54	0.50	0.31
„Clown“	0.06	0.19	0.32	0.17	0.08	-0.04
„Houses“	0.12	0.12	0.23	0.29	0.13	0.09
„Mandrill“	-0.01	0.07	0.02	0.11	0.04	0.09
„Fingerprint“	0.12	0.22	0.21	0.22	0.28	0.23
„Peppers“	0.08	0.06	0.03	-0.12	-0.11	-0.29
„F16“	0.11	0.16	0.16	0.00	-0.05	-0.15
„Bubble“	0.08	0.20	0.16	0.15	0.17	0.13
„Bark“	0.09	0.19	0.17	0.25	0.28	0.35
„Wood Grain“	0.16	0.13	-0.05	-0.18	-0.17	-0.34
„Sand“	0.10	0.19	0.24	0.29	0.30	0.28
„Water“	0.21	0.11	0.17	0.14	0.06	0.00
„Texture“	0.00	0.11	0.14	0.20	0.25	0.28

Tabelle H.41: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II, } 8 \times 8 \text{ DCT}} - \text{PSNR}_{\text{SPIHT-Codec-II, } 8 \times 8 \text{ DCT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.39	0.43	0.47	0.53	0.47	0.47
„Lena“	0.08	0.12	0.17	0.19	0.18	0.21
„Zelda“	-0.09	0.03	0.03	0.13	0.12	0.18
„Boat“	-0.11	-0.03	0.06	0.15	0.21	0.18
„Goldhill“	0.04	0.02	0.07	0.13	0.18	0.15
„Lighthouse“	-0.07	0.02	0.08	0.17	0.13	0.21
„Roof“	0.45	0.38	0.50	0.43	0.42	0.39
„Clown“	-0.07	0.17	0.19	0.25	0.28	0.21
„Houses“	0.08	0.07	0.16	0.23	0.26	0.29
„Mandrill“	0.02	0.11	0.07	0.15	0.21	0.23
„Fingerprint“	0.14	0.16	0.31	0.36	0.36	0.36
„Peppers“	0.14	0.15	0.14	0.13	0.21	0.15
„F16“	-0.08	-0.07	0.00	0.10	0.14	0.13
„Bubble“	0.07	0.19	0.17	0.15	0.23	0.20
„Bark“	0.05	0.09	0.09	0.12	0.18	0.19
„Wood Grain“	0.44	0.43	0.35	0.27	0.23	0.26
„Sand“	0.11	0.11	0.16	0.19	0.14	0.21
„Water“	0.13	0.04	0.07	0.11	0.14	0.15
„Texture“	0.05	0.12	0.14	0.16	0.29	0.23

Tabelle H.42: $\Delta\text{PSNR} = \text{PSNR}_{\text{AGU-Codec-II, } 16 \times 16 \text{ DCT}} - \text{PSNR}_{\text{SPIHT-Codec-II, } 16 \times 16 \text{ DCT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.41	0.62	0.66	0.61	0.57	0.52
„Lena“	0.21	0.25	0.24	0.22	0.20	0.19
„Zelda“	0.06	0.12	0.13	0.16	0.22	0.30
„Boat“	0.08	0.14	0.20	0.25	0.32	0.28
„Goldhill“	0.11	0.12	0.14	0.17	0.21	0.17
„Lighthouse“	0.13	0.10	0.16	0.23	0.21	0.32
„Roof“	0.58	0.61	0.54	0.57	0.52	0.48
„Clown“	0.26	0.26	0.28	0.27	0.30	0.25
„Houses“	0.16	0.16	0.22	0.25	0.29	0.27
„Mandrill“	0.07	0.17	0.15	0.19	0.24	0.23
„Fingerprint“	0.41	0.40	0.44	0.47	0.45	0.47
„Peppers“	0.25	0.17	0.17	0.17	0.20	0.18
„F16“	0.17	0.17	0.21	0.25	0.20	0.24
„Bubble“	0.11	0.16	0.18	0.16	0.23	0.16
„Bark“	0.11	0.14	0.17	0.16	0.25	0.24
„Wood Grain“	0.37	0.39	0.28	0.27	0.23	0.26
„Sand“	0.11	0.15	0.15	0.23	0.17	0.28
„Water“	0.28	0.41	0.33	0.24	0.23	0.18
„Texture“	0.10	0.14	0.18	0.20	0.35	0.22

Tabelle H.43: $\Delta\text{PSNR} = \text{PSNR}_{\text{AGU-Codec-II, } 32 \times 32 \text{ DCT}} - \text{PSNR}_{\text{SPIHT-Codec-II, } 32 \times 32 \text{ DCT}}$, ΔPSNR in dB.

H.2.2 Zweifach überlappende Kosinustransformation

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.71	0.94	0.98	0.84	0.70	0.55
„Lena“	0.48	0.41	0.33	0.19	0.09	0.06
„Zelda“	0.45	0.67	0.56	0.37	0.28	0.31
„Boat“	0.21	0.14	0.09	-0.02	-0.03	-0.10
„Goldhill“	0.33	0.24	0.15	0.09	0.05	0.03
„Lighthouse“	0.60	0.52	0.31	0.16	0.14	0.08
„Roof“	1.14	1.17	1.14	0.80	0.75	0.57
„Clown“	0.45	0.63	0.36	0.09	0.06	-0.01
„Houses“	0.30	0.29	0.24	0.08	-0.08	-0.15
„Mandrill“	0.08	0.10	0.10	0.10	0.04	0.05
„Fingerprint“	1.13	1.14	1.05	0.91	0.91	0.94
„Peppers“	0.39	0.22	-0.03	-0.16	-0.15	-0.15
„F16“	0.31	0.13	-0.04	-0.12	-0.18	-0.20
„Bubble“	0.31	0.28	0.16	0.03	-0.06	-0.12
„Bark“	0.22	0.20	0.15	0.15	0.09	0.20
„Wood Grain“	0.76	0.96	0.95	0.72	0.60	0.46
„Sand“	0.10	0.15	0.16	0.12	0.16	0.10
„Water“	0.49	0.73	0.96	0.74	0.53	0.48
„Texture“	0.18	0.16	0.13	0.08	0.02	0.06

Tabelle H.44: $\Delta\text{PSNR} = \text{PSNR}_{\text{SPIHT-Codec-II, } 16 \times 32 \text{ LCT-2}} - \text{PSNR}_{\text{SPIHT-Codec-II, } 8 \times 16 \text{ LCT-2}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.55	0.22	-0.14	-0.43	-0.55	-0.52
„Lena“	0.01	-0.16	-0.33	-0.35	-0.36	-0.32
„Zelda“	0.56	0.18	-0.01	-0.04	0.01	0.02
„Boat“	0.00	-0.16	-0.32	-0.50	-0.61	-0.65
„Goldhill“	0.11	0.02	-0.10	-0.15	-0.19	-0.18
„Lighthouse“	0.21	0.09	-0.08	-0.21	-0.30	-0.35
„Roof“	0.63	0.46	0.13	-0.17	-0.39	-0.52
„Clown“	0.23	-0.04	-0.28	-0.47	-0.47	-0.45
„Houses“	0.23	0.09	-0.01	-0.21	-0.38	-0.49
„Mandrill“	0.01	-0.04	-0.12	-0.20	-0.26	-0.30
„Fingerprint“	0.51	0.42	0.29	0.27	0.26	0.29
„Peppers“	-0.08	-0.36	-0.56	-0.58	-0.44	-0.46
„F16“	-0.07	-0.26	-0.49	-0.67	-0.65	-0.71
„Bubble“	0.05	0.04	-0.05	-0.12	-0.15	-0.19
„Bark“	0.10	0.07	0.03	0.05	0.02	0.03
„Wood Grain“	0.75	0.63	0.46	0.29	0.16	0.14
„Sand“	0.01	0.04	0.11	0.06	0.06	0.04
„Water“	0.40	0.54	0.45	0.21	0.18	0.10
„Texture“	0.11	0.06	0.00	-0.06	-0.14	-0.19

Tabelle H.45: $\Delta\text{PSNR} = \text{PSNR}_{\text{SPIHT-Codec-II, } 32 \times 64 \text{ LCT-2}} - \text{PSNR}_{\text{SPIHT-Codec-II, } 16 \times 32 \text{ LCT-2}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.46	0.52	0.61	0.58	0.40	0.27
„Lena“	0.10	0.08	0.06	-0.01	-0.12	-0.15
„Zelda“	-0.03	-0.03	-0.02	-0.03	-0.10	-0.10
„Boat“	0.08	0.08	0.18	0.18	0.18	0.02
„Goldhill“	0.07	0.04	0.04	0.09	0.07	0.03
„Lighthouse“	0.22	0.25	0.18	0.18	0.12	0.18
„Roof“	0.55	0.67	0.65	0.52	0.39	0.31
„Clown“	0.02	0.20	0.25	0.20	0.18	0.01
„Houses“	0.15	0.10	0.27	0.33	0.24	0.19
„Mandrill“	0.00	0.05	-0.02	0.08	0.05	0.05
„Fingerprint“	0.15	0.18	0.23	0.24	0.21	0.19
„Peppers“	-0.04	0.02	-0.04	-0.11	-0.10	-0.25
„F16“	0.05	0.04	0.07	0.03	0.01	-0.12
„Bubble“	0.08	0.17	0.17	0.13	0.15	0.11
„Bark“	0.08	0.15	0.15	0.19	0.26	0.32
„Wood Grain“	0.11	0.10	0.00	-0.19	-0.24	-0.34
„Sand“	0.15	0.19	0.25	0.24	0.25	0.29
„Water“	0.26	0.16	0.14	0.16	0.10	0.02
„Texture“	0.03	0.08	0.14	0.19	0.30	0.25

Tabelle H.46: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II, } 8 \times 16 \text{ LCT-2}} - \text{PSNR}_{\text{SPIHT-Codec-II, } 8 \times 16 \text{ LCT-2}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.36	0.44	0.52	0.56	0.48	0.50
„Lena“	0.04	0.07	0.14	0.17	0.15	0.18
„Zelda“	-0.13	-0.07	0.04	0.11	0.15	0.18
„Boat“	-0.13	-0.06	0.01	0.12	0.17	0.18
„Goldhill“	-0.02	0.03	0.06	0.11	0.16	0.13
„Lighthouse“	-0.03	0.04	0.09	0.20	0.13	0.26
„Roof“	0.45	0.44	0.43	0.44	0.35	0.37
„Clown“	-0.07	0.08	0.19	0.21	0.29	0.22
„Houses“	0.06	0.07	0.16	0.20	0.22	0.21
„Mandrill“	0.02	0.08	0.05	0.14	0.19	0.21
„Fingerprint“	0.03	0.15	0.21	0.30	0.32	0.33
„Peppers“	0.04	0.07	0.14	0.12	0.19	0.11
„F16“	-0.15	-0.06	0.01	0.08	0.15	0.11
„Bubble“	0.07	0.11	0.14	0.16	0.18	0.16
„Bark“	0.07	0.11	0.12	0.12	0.21	0.17
„Wood Grain“	0.51	0.47	0.30	0.26	0.22	0.24
„Sand“	0.12	0.11	0.13	0.18	0.14	0.22
„Water“	0.07	0.03	0.07	0.09	0.12	0.06
„Texture“	0.03	0.12	0.13	0.15	0.27	0.19

Tabelle H.47: $\Delta\text{PSNR} = \text{PSNR}_{\text{AGU-Codec-II, } 16 \times 32 \text{ LCT-2}} - \text{PSNR}_{\text{SPIHT-Codec-II, } 16 \times 32 \text{ LCT-2}}$, ΔPSNR in dB.

H.2.3 Mehrfach überlappende Kosinustransformation

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.51	0.50	0.61	0.70	0.72	0.72
„Lena“	0.45	0.51	0.54	0.52	0.55	0.61
„Zelda“	0.33	0.36	0.34	0.46	0.53	0.58
„Boat“	0.28	0.36	0.46	0.58	0.62	0.64
„Goldhill“	0.18	0.28	0.32	0.38	0.48	0.52
„Lighthouse“	0.24	0.27	0.36	0.49	0.56	0.63
„Roof“	0.43	0.59	0.71	0.85	0.86	0.88
„Clown“	0.30	0.46	0.50	0.43	0.48	0.55
„Houses“	0.21	0.25	0.36	0.45	0.49	0.56
„Mandrill“	0.29	0.37	0.45	0.50	0.57	0.59
„Fingerprint“	0.55	0.54	0.61	0.61	0.67	0.72
„Peppers“	0.22	0.25	0.23	0.36	0.44	0.49
„F16“	0.34	0.26	0.32	0.32	0.39	0.41
„Bubble“	0.28	0.34	0.37	0.49	0.55	0.59
„Bark“	0.24	0.31	0.35	0.43	0.52	0.59
„Wood Grain“	0.08	0.08	0.22	0.32	0.34	0.37
„Sand“	0.32	0.37	0.39	0.46	0.54	0.62
„Water“	0.20	0.30	0.35	0.35	0.42	0.49
„Texture“	0.33	0.36	0.46	0.57	0.65	0.71

Tabelle H.48: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 24 \text{ LCT-3}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	-0.15	-0.24	-0.39	-0.62	-0.67	-0.68
„Lena“	-0.26	-0.34	-0.42	-0.40	-0.42	-0.44
„Zelda“	-0.20	-0.24	-0.27	-0.27	-0.29	-0.35
„Boat“	-0.28	-0.35	-0.48	-0.58	-0.64	-0.65
„Goldhill“	-0.20	-0.29	-0.35	-0.38	-0.43	-0.44
„Lighthouse“	-0.23	-0.29	-0.34	-0.44	-0.50	-0.58
„Roof“	-0.07	-0.15	-0.31	-0.50	-0.59	-0.64
„Clown“	-0.28	-0.38	-0.56	-0.63	-0.60	-0.60
„Houses“	-0.31	-0.34	-0.47	-0.59	-0.66	-0.72
„Mandrill“	-0.13	-0.20	-0.27	-0.33	-0.37	-0.41
„Fingerprint“	0.00	-0.06	-0.16	-0.28	-0.32	-0.36
„Peppers“	-0.39	-0.56	-0.58	-0.54	-0.55	-0.60
„F16“	-0.40	-0.55	-0.66	-0.74	-0.71	-0.68
„Bubble“	-0.11	-0.21	-0.25	-0.32	-0.36	-0.40
„Bark“	-0.16	-0.21	-0.24	-0.27	-0.30	-0.32
„Wood Grain“	-0.38	-0.36	-0.28	-0.26	-0.29	-0.32
„Sand“	-0.18	-0.17	-0.21	-0.24	-0.28	-0.31
„Water“	-0.20	-0.14	-0.14	-0.15	-0.23	-0.28
„Texture“	-0.13	-0.19	-0.26	-0.33	-0.42	-0.48

Tabelle H.49: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 48 \text{ LCT-6}} - \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 32 \text{ LCT-4}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.31	0.26	0.22	0.08	0.11	0.16
„Lena“	0.10	-0.02	-0.02	0.04	0.14	0.22
„Zelda“	-0.15	-0.07	0.05	0.32	0.34	0.38
„Boat“	-0.21	-0.18	-0.30	-0.37	-0.37	-0.28
„Goldhill“	0.02	0.09	0.05	-0.03	0.01	0.02
„Lighthouse“	0.22	0.09	0.09	0.01	-0.08	-0.11
„Roof“	0.77	0.61	0.43	0.34	0.28	0.25
„Clown“	0.03	0.08	-0.14	-0.24	-0.05	0.02
„Houses“	0.13	0.21	0.02	-0.14	-0.22	-0.27
„Mandrill“	0.11	0.13	0.15	0.11	0.13	0.11
„Fingerprint“	0.83	0.75	0.68	0.67	0.73	0.76
„Peppers“	-0.09	-0.18	-0.32	-0.15	-0.04	0.02
„F16“	-0.20	-0.30	-0.40	-0.39	-0.27	-0.18
„Bubble“	0.12	0.05	0.01	-0.03	-0.05	-0.11
„Bark“	0.09	0.04	-0.01	0.01	0.00	0.03
„Wood Grain“	1.31	1.34	1.12	0.90	0.81	0.83
„Sand“	0.04	0.08	0.01	0.05	0.03	0.06
„Water“	0.06	0.22	0.41	0.40	0.43	0.43
„Texture“	0.14	0.10	0.06	-0.01	-0.04	-0.06

Tabelle H.50: $\Delta\text{PSNR} = \text{PSNR}_{\text{AGU-Codec-II, } 16 \times 48 \text{ LCT-3}} - \text{PSNR}_{\text{L-CEB-Codec-II, } 8 \times 24 \text{ LCT-3}}$,
 ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.16	0.21	0.09	-0.07	-0.03	0.07
„Lena“	0.06	-0.06	-0.12	-0.04	0.03	0.10
„Zelda“	-0.07	-0.05	0.02	0.18	0.23	0.33
„Boat“	-0.17	-0.17	-0.31	-0.40	-0.39	-0.28
„Goldhill“	0.06	0.08	0.03	0.00	-0.01	0.02
„Lighthouse“	0.18	0.10	0.06	-0.03	-0.08	-0.11
„Roof“	0.70	0.49	0.33	0.13	0.11	0.06
„Clown“	0.05	-0.02	-0.18	-0.24	-0.12	-0.03
„Houses“	0.11	0.19	0.02	-0.16	-0.22	-0.29
„Mandrill“	0.02	0.03	0.05	0.03	0.04	0.05
„Fingerprint“	0.69	0.64	0.60	0.57	0.58	0.65
„Peppers“	-0.02	-0.18	-0.29	-0.22	-0.09	0.00
„F16“	-0.18	-0.28	-0.37	-0.35	-0.27	-0.19
„Bubble“	0.14	0.03	-0.02	-0.10	-0.14	-0.19
„Bark“	0.08	0.04	-0.01	-0.02	-0.03	-0.03
„Wood Grain“	1.34	1.49	1.22	0.99	0.91	0.95
„Sand“	0.01	0.03	-0.03	-0.02	-0.02	-0.03
„Water“	0.18	0.28	0.41	0.36	0.34	0.34
„Texture“	0.06	0.04	-0.01	-0.08	-0.11	-0.13

Tabelle H.51: $\Delta\text{PSNR} = \text{PSNR}_{\text{AGU-Codec-II, } 16 \times 64 \text{ LCT-4}} - \text{PSNR}_{\text{L-CEB-Codec-II, } 8 \times 32 \text{ LCT-4}}$,
 ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.32	0.62	0.76	0.65	0.47	0.22
„Lena“	-0.06	0.00	0.06	-0.02	-0.13	-0.19
„Zelda“	-0.06	-0.05	-0.02	-0.11	-0.07	-0.24
„Boat“	0.05	0.06	0.08	0.11	0.14	0.06
„Goldhill“	0.04	0.05	0.07	0.06	0.11	0.00
„Lighthouse“	0.28	0.17	0.16	0.18	0.18	0.27
„Roof“	0.58	0.76	0.75	0.59	0.39	0.22
„Clown“	0.09	0.18	0.31	0.22	0.09	0.06
„Houses“	0.18	0.09	0.16	0.25	0.23	0.14
„Mandrill“	-0.03	0.05	0.00	-0.01	0.06	-0.04
„Fingerprint“	-0.03	0.10	0.17	0.21	0.14	0.15
„Peppers“	-0.07	-0.03	-0.07	-0.11	-0.23	-0.18
„F16“	0.07	0.06	0.03	0.01	-0.09	-0.15
„Bubble“	0.08	0.13	0.14	0.14	0.13	0.07
„Bark“	0.05	0.13	0.19	0.16	0.26	0.21
„Wood Grain“	-0.02	0.07	0.10	-0.08	-0.26	-0.41
„Sand“	0.08	0.18	0.13	0.18	0.23	0.31
„Water“	0.23	0.29	0.31	0.11	0.05	-0.06
„Texture“	0.03	0.08	0.16	0.19	0.33	0.19

Tabelle H.52: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{SPIHT-Codec-II}, 8 \times 32 \text{ LCT-4}}$,
 ΔPSNR in dB.

H.2.4 Kosinus- und überlappende Kosinustransformation im Vergleich

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	1.03	1.36	1.87	1.94	1.81	1.58
„Lena“	0.80	0.89	0.88	0.71	0.54	0.44
„Zelda“	1.25	1.15	0.87	0.57	0.50	0.53
„Boat“	0.55	0.55	0.68	0.66	0.69	0.64
„Goldhill“	0.45	0.50	0.59	0.57	0.53	0.52
„Lighthouse“	0.53	0.53	0.47	0.45	0.44	0.45
„Roof“	1.00	1.38	1.52	1.70	1.59	1.57
„Clown“	0.60	1.02	1.07	1.02	0.81	0.70
„Houses“	0.37	0.37	0.42	0.43	0.40	0.35
„Mandrill“	0.20	0.28	0.35	0.44	0.46	0.45
„Fingerprint“	0.68	1.18	1.53	1.78	1.86	1.96
„Peppers“	0.73	0.81	0.56	0.29	0.21	0.22
„F16“	0.68	0.71	0.78	0.75	0.63	0.53
„Bubble“	0.44	0.56	0.71	0.69	0.66	0.64
„Bark“	0.37	0.50	0.62	0.69	0.73	0.72
„Wood Grain“	0.29	0.34	0.56	0.60	0.60	0.55
„Sand“	0.28	0.43	0.58	0.59	0.60	0.61
„Water“	0.38	0.72	1.01	0.99	0.77	0.65
„Texture“	0.28	0.36	0.45	0.49	0.50	0.48

Tabelle H.53: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 16 \text{ LCT-2}} - \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 8 \text{ DCT}}$,
 ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.60	0.76	0.73	0.63	0.46	0.37
„Lena“	0.49	0.48	0.44	0.27	0.17	0.10
„Zelda“	1.02	1.03	0.75	0.37	0.30	0.28
„Boat“	0.13	0.08	0.01	-0.02	-0.04	-0.08
„Goldhill“	0.22	0.13	0.06	0.00	0.00	-0.05
„Lighthouse“	0.22	0.19	0.10	0.05	0.02	0.00
„Roof“	0.56	0.58	0.58	0.36	0.25	0.16
„Clown“	0.29	0.31	0.20	0.02	-0.04	-0.06
„Houses“	0.06	-0.01	-0.02	-0.14	-0.23	-0.25
„Mandrill“	0.06	0.05	0.03	-0.01	-0.02	-0.03
„Fingerprint“	0.60	0.65	0.55	0.45	0.44	0.44
„Peppers“	0.54	0.39	0.22	0.02	-0.02	-0.05
„F16“	0.20	0.07	-0.02	-0.12	-0.19	-0.21
„Bubble“	0.17	0.18	0.08	0.04	-0.01	-0.05
„Bark“	0.12	0.11	0.08	0.05	0.00	0.01
„Wood Grain“	-0.07	0.01	0.16	0.27	0.18	0.10
„Sand“	0.03	0.03	0.04	0.02	0.01	-0.01
„Water“	0.17	0.43	0.58	0.32	0.21	0.15
„Texture“	0.13	0.12	0.06	0.01	-0.03	-0.04

Tabelle H.54: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 16 \text{ LCT-2}}$,
 ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.07	0.46	0.79	1.01	1.08	0.91
„Lena“	0.51	0.52	0.55	0.46	0.40	0.33
„Zelda“	0.03	0.21	0.25	0.17	0.13	0.14
„Boat“	0.19	0.34	0.45	0.50	0.55	0.54
„Goldhill“	0.16	0.21	0.28	0.26	0.28	0.27
„Lighthouse“	0.05	0.10	0.22	0.27	0.33	0.36
„Roof“	0.37	0.56	0.78	0.96	1.04	0.98
„Clown“	0.06	0.20	0.33	0.42	0.40	0.37
„Houses“	0.06	0.13	0.17	0.31	0.42	0.47
„Mandrill“	0.04	0.11	0.16	0.20	0.23	0.24
„Fingerprint“	0.02	0.03	0.16	0.17	0.18	0.20
„Peppers“	0.47	0.51	0.55	0.49	0.38	0.41
„F16“	0.33	0.40	0.48	0.63	0.53	0.58
„Bubble“	0.20	0.22	0.28	0.26	0.29	0.28
„Bark“	0.10	0.13	0.15	0.14	0.15	0.15
„Wood Grain“	-0.42	-0.24	-0.09	0.00	0.03	0.06
„Sand“	0.07	0.11	0.06	0.10	0.08	0.10
„Water“	-0.11	-0.02	0.06	0.11	0.06	0.08
„Texture“	0.04	0.02	0.08	0.09	0.11	0.08

Tabelle H.55: $\Delta\text{PSNR} = \text{PSNR}_{\text{SPIHT-Codec-II}, 16 \times 32 \text{ LCT-2}} - \text{PSNR}_{\text{SPIHT-Codec-II}, 32 \times 32 \text{ DCT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.10	0.18	0.39	0.73	0.77	0.78
„Lena“	0.68	0.67	0.66	0.55	0.49	0.41
„Zelda“	0.63	0.59	0.44	0.25	0.12	0.25
„Boat“	0.14	0.30	0.47	0.57	0.58	0.52
„Goldhill“	0.08	0.09	0.16	0.20	0.19	0.22
„Lighthouse“	-0.39	-0.15	0.03	0.16	0.15	0.19
„Roof“	-0.24	-0.12	0.12	0.45	0.54	0.66
„Clown“	-0.17	-0.10	0.11	0.33	0.39	0.27
„Houses“	-0.21	-0.16	0.02	0.17	0.28	0.42
„Mandrill“	0.05	0.06	0.07	0.18	0.16	0.25
„Fingerprint“	-0.33	-0.38	-0.28	-0.26	-0.22	-0.26
„Peppers“	0.65	0.73	0.83	0.67	0.64	0.44
„F16“	0.20	0.32	0.54	0.65	0.62	0.60
„Bubble“	0.06	0.16	0.23	0.26	0.36	0.39
„Bark“	0.03	0.06	0.04	0.07	0.06	0.07
„Wood Grain“	-1.12	-1.16	-0.98	-0.56	-0.37	-0.23
„Sand“	0.07	0.00	0.06	0.06	-0.05	-0.03
„Water“	-0.40	-0.45	-0.49	-0.26	-0.21	-0.17
„Texture“	-0.01	-0.02	-0.01	0.02	0.03	0.04

Tabelle H.56: $\Delta\text{PSNR} = \text{PSNR}_{\text{SPIHT-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{SPIHT-Codec-II}, 32 \times 32 \text{ DCT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.03	-0.28	-0.40	-0.28	-0.31	-0.13
„Lena“	0.17	0.15	0.11	0.09	0.09	0.08
„Zelda“	0.60	0.38	0.19	0.08	-0.01	0.11
„Boat“	-0.05	-0.04	0.02	0.07	0.03	-0.02
„Goldhill“	-0.08	-0.12	-0.12	-0.06	-0.09	-0.05
„Lighthouse“	-0.44	-0.25	-0.19	-0.11	-0.18	-0.17
„Roof“	-0.61	-0.68	-0.66	-0.51	-0.50	-0.32
„Clown“	-0.23	-0.30	-0.22	-0.09	-0.01	-0.10
„Houses“	-0.27	-0.29	-0.15	-0.14	-0.14	-0.05
„Mandrill“	0.01	-0.05	-0.09	-0.02	-0.07	0.01
„Fingerprint“	-0.35	-0.41	-0.44	-0.43	-0.40	-0.46
„Peppers“	0.18	0.22	0.28	0.18	0.26	0.03
„F16“	-0.13	-0.08	0.06	0.02	0.09	0.02
„Bubble“	-0.14	-0.06	-0.05	0.00	0.07	0.11
„Bark“	-0.07	-0.07	-0.11	-0.07	-0.09	-0.08
„Wood Grain“	-0.70	-0.92	-0.89	-0.56	-0.40	-0.29
„Sand“	0.00	-0.11	0.00	-0.04	-0.13	-0.13
„Water“	-0.29	-0.43	-0.55	-0.37	-0.27	-0.25
„Texture“	-0.05	-0.04	-0.09	-0.07	-0.08	-0.04

Tabelle H.57: $\Delta\text{PSNR} = \text{PSNR}_{\text{SPIHT-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{SPIHT-Codec-II}, 16 \times 32 \text{ LCT-2}}$, ΔPSNR in dB.

H.2.5 Vergleich mit anderen blocküberlappenden Transformationen

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.40	0.60	0.72	0.69	0.63	0.55
„Lena“	0.10	0.10	0.14	0.15	0.18	0.16
„Zelda“	0.08	0.14	0.18	0.13	0.14	0.17
„Boat“	-0.01	-0.11	-0.05	-0.06	-0.01	0.02
„Goldhill“	-0.09	-0.06	-0.03	-0.01	0.02	0.03
„Lighthouse“	0.11	0.13	0.05	0.05	0.05	0.07
„Roof“	0.88	0.88	0.79	0.73	0.78	0.84
„Clown“	-0.08	-0.07	-0.09	-0.08	-0.02	0.03
„Houses“	-0.02	-0.18	-0.13	-0.21	-0.26	-0.24
„Mandrill“	0.12	0.08	0.12	0.17	0.19	0.20
„Fingerprint“	0.64	0.52	0.45	0.41	0.51	0.56
„Peppers“	-0.23	-0.26	-0.23	-0.15	-0.09	-0.06
„F16“	0.01	-0.24	-0.20	-0.18	-0.16	-0.16
„Bubble“	0.07	0.07	0.05	0.07	0.10	0.08
„Bark“	0.03	0.00	0.06	0.08	0.12	0.15
„Wood Grain“	-0.17	-0.22	-0.08	-0.06	-0.07	-0.12
„Sand“	0.14	0.12	0.13	0.14	0.17	0.19
„Water“	0.07	0.09	0.20	0.13	0.12	0.14
„Texture“	0.09	0.09	0.08	0.07	0.10	0.14

Tabelle H.58: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 16 \text{ TDLT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.69	0.84	0.94	0.93	0.91	0.84
„Lena“	0.62	0.62	0.72	0.65	0.49	0.39
„Zelda“	1.37	1.36	1.04	0.69	0.58	0.58
„Boat“	0.18	0.17	0.20	0.24	0.26	0.25
„Goldhill“	0.23	0.17	0.18	0.14	0.13	0.09
„Lighthouse“	0.25	0.23	0.15	0.11	0.10	0.11
„Roof“	0.65	0.68	0.77	0.68	0.65	0.63
„Clown“	0.26	0.37	0.44	0.47	0.41	0.36
„Houses“	0.09	0.02	0.06	-0.05	-0.12	-0.12
„Mandrill“	0.08	0.05	0.05	0.05	0.05	0.03
„Fingerprint“	0.59	0.71	0.75	0.83	0.98	1.10
„Peppers“	0.57	0.60	0.48	0.31	0.21	0.17
„F16“	0.21	0.13	0.18	0.19	0.12	0.02
„Bubble“	0.21	0.18	0.14	0.12	0.11	0.08
„Bark“	0.10	0.11	0.12	0.09	0.09	0.09
„Wood Grain“	-0.05	0.00	0.26	0.33	0.23	0.15
„Sand“	0.03	0.02	0.07	0.04	0.04	0.01
„Water“	0.18	0.44	0.66	0.41	0.31	0.27
„Texture“	0.12	0.12	0.07	0.02	0.02	0.03

Tabelle H.59: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 16 \text{ LOT-II}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.15	0.29	0.43	0.54	0.58	0.59
„Lena“	0.01	0.00	0.12	0.15	0.16	0.13
„Zelda“	-0.03	0.02	0.08	0.11	0.12	0.13
„Boat“	0.02	-0.05	-0.07	0.03	0.10	0.14
„Goldhill“	-0.05	-0.03	-0.01	-0.03	0.00	-0.01
„Lighthouse“	0.08	0.08	-0.01	-0.02	-0.02	-0.01
„Roof“	0.27	0.39	0.62	0.74	0.77	0.81
„Clown“	-0.06	-0.01	0.08	0.18	0.20	0.21
„Houses“	-0.02	-0.13	-0.08	-0.12	-0.12	-0.10
„Mandrill“	-0.02	-0.07	-0.05	-0.03	-0.02	-0.01
„Fingerprint“	0.04	0.07	0.23	0.38	0.50	0.60
„Peppers“	-0.08	-0.14	-0.10	-0.02	-0.02	-0.04
„F16“	-0.12	-0.19	-0.07	0.02	0.06	0.03
„Bubble“	-0.07	-0.01	-0.03	0.00	0.01	0.02
„Bark“	-0.09	-0.04	-0.01	0.01	0.03	0.04
„Wood Grain“	-0.12	-0.12	-0.08	-0.08	-0.09	-0.10
„Sand“	-0.04	-0.06	-0.01	0.00	0.01	0.01
„Water“	-0.03	0.03	0.12	0.11	0.09	0.08
„Texture“	-0.04	-0.03	-0.05	-0.04	-0.03	0.00

Tabelle H.60: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 48 \text{ GenLOT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.09	0.12	0.09	0.09	0.12	0.15
„Lena“	0.07	0.09	0.13	0.16	0.16	0.13
„Zelda“	0.08	0.13	0.20	0.23	0.21	0.20
„Boat“	-0.02	-0.04	-0.05	-0.03	-0.02	-0.02
„Goldhill“	-0.02	-0.03	-0.03	-0.03	-0.03	-0.05
„Lighthouse“	-0.01	0.00	-0.02	-0.05	-0.07	-0.08
„Roof“	0.06	0.10	0.05	0.05	0.03	-0.03
„Clown“	0.01	0.05	0.03	0.06	0.08	0.08
„Houses“	-0.04	-0.08	-0.07	-0.12	-0.14	-0.13
„Mandrill“	-0.01	-0.02	-0.04	-0.04	-0.05	-0.04
„Fingerprint“	0.12	0.09	0.08	0.08	0.12	0.14
„Peppers“	0.06	0.03	0.02	0.07	0.05	0.03
„F16“	-0.04	-0.10	-0.08	-0.09	-0.06	-0.07
„Bubble“	-0.01	0.01	-0.03	-0.02	-0.06	-0.05
„Bark“	-0.01	-0.02	-0.03	-0.04	-0.04	-0.05
„Wood Grain“	-0.10	-0.10	-0.05	-0.01	-0.03	-0.04
„Sand“	-0.01	-0.02	-0.02	-0.03	-0.02	-0.05
„Water“	0.01	0.06	0.08	0.03	0.02	0.01
„Texture“	0.01	-0.01	-0.04	-0.05	-0.04	-0.06

Tabelle H.61: $\Delta\text{PSNR} = \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 32 \text{ LCT-4}} - \text{PSNR}_{\text{L-CEB-Codec-II}, 8 \times 32 \text{ ELT}}$, ΔPSNR in dB.

H.2.6 Vergleich mit waveletbasierten Kompressionsverfahren

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.73	1.40	1.53	1.37	1.14	1.08
„Lena“	-0.65	-0.58	-0.51	-0.41	-0.39	-0.36
„Zelda“	-0.68	-0.48	-0.28	-0.07	0.05	0.12
„Boat“	-0.13	-0.21	-0.23	-0.14	-0.14	-0.22
„Goldhill“	-0.05	0.00	0.00	0.01	-0.01	-0.02
„Lighthouse“	1.07	1.24	0.73	0.33	0.25	0.15
„Roof“	1.16	1.49	1.56	1.71	1.83	1.86
„Clown“	0.28	0.62	0.35	0.28	0.28	0.28
„Houses“	0.30	0.38	0.43	0.38	-0.03	-0.31
„Mandrill“	0.01	0.00	0.12	0.05	0.02	0.01
„Fingerprint“	0.63	1.01	1.14	1.19	1.58	1.94
„Peppers“	-0.64	-0.90	-1.09	-1.03	-0.83	-0.77
„F16“	-0.01	-0.36	-0.46	-0.64	-0.69	-0.75
„Bubble“	-0.11	-0.13	-0.14	-0.15	-0.24	-0.45
„Bark“	-0.03	0.05	0.09	0.22	0.36	0.46
„Wood Grain“	1.37	1.77	2.33	2.43	2.23	1.63
„Sand“	-0.04	0.08	0.20	0.21	0.31	0.24
„Water“	0.71	1.26	1.82	1.73	1.39	1.10
„Texture“	0.02	0.08	0.11	0.11	0.20	0.17

Tabelle H.62: $\Delta\text{PSNR} = \text{PSNR}_{\text{SPIHT-Codec-II}, 32 \times 32 \text{ DCT}} - \text{PSNR}_{\text{SPIHT-Codec-II}, \text{DWT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	0.80	1.86	2.32	2.38	2.22	1.99
„Lena“	-0.14	-0.06	0.04	0.05	0.01	-0.03
„Zelda“	-0.65	-0.27	-0.03	0.10	0.18	0.26
„Boat“	0.06	0.13	0.22	0.36	0.41	0.32
„Goldhill“	0.11	0.21	0.28	0.27	0.27	0.25
„Lighthouse“	1.12	1.34	0.95	0.60	0.58	0.51
„Roof“	1.53	2.05	2.34	2.67	2.87	2.84
„Clown“	0.34	0.82	0.68	0.70	0.68	0.65
„Houses“	0.36	0.51	0.60	0.69	0.39	0.16
„Mandrill“	0.05	0.11	0.28	0.25	0.25	0.25
„Fingerprint“	0.65	1.04	1.30	1.36	1.76	2.14
„Peppers“	-0.17	-0.39	-0.54	-0.54	-0.45	-0.36
„F16“	0.32	0.04	0.02	-0.01	-0.16	-0.17
„Bubble“	0.09	0.09	0.14	0.11	0.05	-0.17
„Bark“	0.07	0.18	0.24	0.36	0.51	0.61
„Wood Grain“	0.95	1.53	2.24	2.43	2.26	1.69
„Sand“	0.03	0.19	0.26	0.31	0.39	0.34
„Water“	0.60	1.24	1.88	1.84	1.45	1.18
„Texture“	0.06	0.10	0.19	0.20	0.31	0.25

Tabelle H.63: $\Delta\text{PSNR} = \text{PSNR}_{\text{SPIHT-Codec-II}, 16 \times 32 \text{ LCT-2}} - \text{PSNR}_{\text{SPIHT-Codec-II}, \text{DWT}}$, ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	1.40	1.63	1.41	1.20	1.08	0.86
„Lena“	0.01	-0.17	-0.25	-0.23	-0.12	-0.09
„Zelda“	-0.01	-0.12	0.00	0.16	0.25	0.37
„Boat“	0.39	0.10	-0.05	-0.02	-0.13	-0.11
„Goldhill“	0.22	0.21	0.16	0.07	0.17	0.12
„Lighthouse“	1.07	0.88	0.36	0.14	0.09	0.25
„Roof“	1.69	1.52	1.29	1.41	1.39	0.92
„Clown“	0.79	0.86	0.26	0.21	0.23	0.27
„Houses“	0.40	0.39	0.31	0.02	-0.35	-0.84
„Mandrill“	0.22	0.32	0.37	0.36	0.35	0.33
„Fingerprint“	1.26	1.37	1.30	1.18	1.67	1.71
„Peppers“	-0.03	-0.65	-0.91	-0.84	-0.60	-0.41
„F16“	0.40	-0.13	-0.33	-0.44	-0.66	-0.64
„Bubble“	0.40	0.17	0.25	0.04	0.02	-0.17
„Bark“	0.30	0.22	0.38	0.27	0.63	0.64
„Wood Grain“	1.50	1.47	1.75	1.80	0.98	0.82
„Sand“	0.27	0.43	0.41	0.36	0.48	0.50
„Water“	1.06	1.31	1.55	1.38	0.80	0.65
„Texture“	0.31	0.32	0.39	0.24	0.48	0.23

Tabelle H.64: $\Delta\text{PSNR} = \text{PSNR}_{\text{AGU-Codec-II, } 32 \times 32 \text{ DCT}} - \text{PSNR}_{\text{JPEG2000, DWT}}$,
 ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	1.42	1.91	2.06	2.16	2.07	1.75
„Lena“	0.35	0.17	0.20	0.18	0.23	0.23
„Zelda“	-0.17	-0.10	0.16	0.28	0.31	0.39
„Boat“	0.37	0.24	0.21	0.35	0.27	0.33
„Goldhill“	0.25	0.33	0.36	0.27	0.40	0.35
„Lighthouse“	0.96	0.92	0.51	0.38	0.34	0.55
„Roof“	1.93	1.91	1.96	2.24	2.26	1.79
„Clown“	0.52	0.88	0.50	0.57	0.62	0.61
„Houses“	0.36	0.43	0.42	0.28	0.00	-0.43
„Mandrill“	0.21	0.34	0.43	0.51	0.53	0.55
„Fingerprint“	0.90	1.15	1.23	1.18	1.72	1.77
„Peppers“	0.23	-0.24	-0.39	-0.40	-0.23	-0.07
„F16“	0.41	0.04	-0.05	0.02	-0.18	-0.19
„Bubble“	0.56	0.34	0.49	0.30	0.26	0.11
„Bark“	0.36	0.32	0.48	0.37	0.74	0.72
„Wood Grain“	1.22	1.31	1.68	1.79	1.00	0.86
„Sand“	0.35	0.50	0.45	0.41	0.53	0.54
„Water“	0.74	0.91	1.35	1.34	0.75	0.61
„Texture“	0.28	0.32	0.42	0.28	0.51	0.28

Tabelle H.65: $\Delta\text{PSNR} = \text{PSNR}_{\text{AGU-Codec-II, } 16 \times 32 \text{ LCT-2}} - \text{PSNR}_{\text{JPEG2000, DWT}}$,
 ΔPSNR in dB.

Testbild	Bitrate in bpp					
	0.0625	0.125	0.25	0.5	0.75	1
„Barbara“	-0.26	0.39	0.78	0.78	0.63	0.74
„Lena“	-0.45	-0.16	-0.02	0.04	-0.07	-0.08
„Zelda“	-0.61	-0.24	-0.15	-0.07	0.02	0.05
„Boat“	-0.44	-0.17	0.02	0.13	0.31	0.17
„Goldhill“	-0.16	-0.09	-0.02	0.11	0.03	0.03
„Lighthouse“	0.13	0.46	0.53	0.42	0.37	0.22
„Roof“	0.05	0.58	0.81	0.87	0.96	1.42
„Clown“	-0.25	0.02	0.37	0.34	0.35	0.26
„Houses“	0.06	0.15	0.34	0.61	0.61	0.80
„Mandrill“	-0.14	-0.15	-0.10	-0.12	-0.09	-0.09
„Fingerprint“	-0.22	0.04	0.28	0.48	0.36	0.70
„Peppers“	-0.36	-0.08	-0.01	-0.02	-0.03	-0.18
„F16“	-0.24	-0.06	0.08	0.05	0.17	0.13
„Bubble“	-0.40	-0.14	-0.21	-0.03	-0.03	-0.12
„Bark“	-0.22	-0.03	-0.12	0.11	-0.02	0.06
„Wood Grain“	0.24	0.69	0.86	0.90	1.48	1.07
„Sand“	-0.20	-0.20	-0.06	0.08	0.00	0.02
„Water“	-0.07	0.36	0.60	0.59	0.82	0.63
„Texture“	-0.19	-0.10	-0.10	0.07	0.07	0.16

Tabelle H.66: $\Delta\text{PSNR} = \text{PSNR}_{\text{JPEG2000, DWT}} - \text{PSNR}_{\text{SPIHT, DWT}}$,
 ΔPSNR in dB.

H.3 Testergebnisse zur subjektiven Beurteilung der Rekonstruktionsqualität

H.3.1 Vergleich aller untersuchten Kompressionsstrategien am Beispiel des Testbildes „Barbara“ bei 0.1 bpp

Kosinustransformation



Abbildung H.153: JPEG Coder [IJG11] mit arithmetischer Codierung und skalarer Quantisierung (keine wahrnehmungsangepasste Quantisierung) , 8×8 DCT, Bitrate = 0.11 bpp, PSNR = 23.19 dB.



Abbildung H.154: L-CEB-Codec-I, 8×8 DCT, PSNR = 24.10 dB.

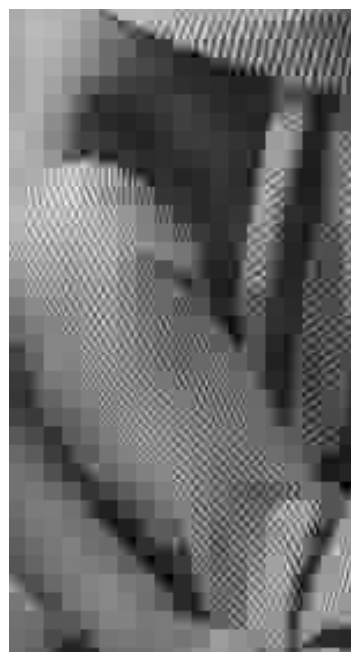


Abbildung H.155: L-CEB-Codec-II, 8×8 DCT & 3-level DWT, PSNR = 24.18 dB.



Abbildung H.156: L-CEB-Codec-I pp, 8×8 DCT, PSNR = 24.95 dB.



Abbildung H.157: L-CEB-Codec-II pp, 8×8 DCT & 3-level DWT, PSNR = 25.00 dB.



Abbildung H.158: SPIHT-Codec-I, 8×8 DCT, PSNR = 23.23 dB.



Abbildung H.159: SPIHT-Codec-II, 8×8 DCT & 3-level DWT, PSNR = 23.97 dB.



Abbildung H.160: SPIHT-Codec-I pp, 8×8 DCT, PSNR = 23.92 dB.



Abbildung H.161: SPIHT-Codec-II pp, 8×8 DCT & 3-level DWT, PSNR = 24.61 dB.



Abbildung H.162: SPIHT-Codec-I, 16×16 DCT, PSNR = 24.82 dB.



Abbildung H.163: SPIHT-Codec-II, 16×16 DCT & 2-level DWT, PSNR = 24.94 dB.



Abbildung H.164: SPIHT-Codec-I pp, 16×16 DCT, PSNR = 25.22 dB.



Abbildung H.165: SPIHT-Codec-II pp, 16×16 DCT & 2-level DWT, PSNR = 25.32 dB.



Abbildung H.166: AGU-Codec-I, 16×16 DCT, PSNR = 25.17 dB.



Abbildung H.167: AGU-Codec-II, 16×16 DCT & 2-level DWT, PSNR = 25.22 dB.



Abbildung H.168: AGU-Codec-I pp, 16×16 DCT, PSNR = 25.61 dB.



Abbildung H.169: AGU-Codec-II pp, 16×16 DCT & 2-level DWT, PSNR = 25.67 dB.



Abbildung H.170: SPIHT-Codec-I, 32×32 DCT, PSNR = 25.53 dB.



Abbildung H.171: SPIHT-Codec-II, 32×32 DCT & 1-level DWT, PSNR = 25.56 dB.



Abbildung H.172: SPIHT-Codec-I pp, 32×32 DCT, PSNR = 25.75 dB.



Abbildung H.173: SPIHT-Codec-II pp, 32×32 DCT & 1-level DWT, PSNR = 25.77 dB.



Abbildung H.174: AGU-Codec-I, 32×32 DCT, PSNR = 26.06 dB.



Abbildung H.175: AGU-Codec-II, 32×32 DCT & 1-level DWT, PSNR = 26.05 dB.



Abbildung H.176: AGU-Codec-I pp, 32×32 DCT, PSNR = 26.40 dB.



Abbildung H.177: AGU-Codec-II pp, 32×32 DCT & 1-level DWT, PSNR = 26.39 dB.

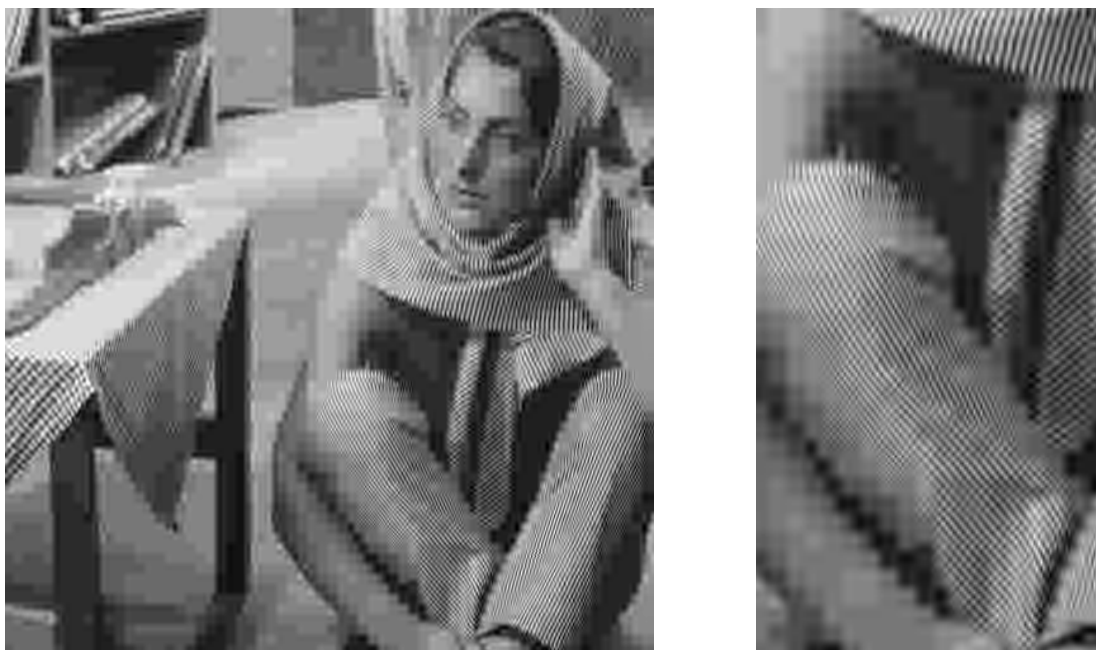
Zweifachüberlappung, Malvar-Fenster

Abbildung H.178: L-CEB-Codec-I, 8×16 LCT-2, PSNR = 25.36 dB.

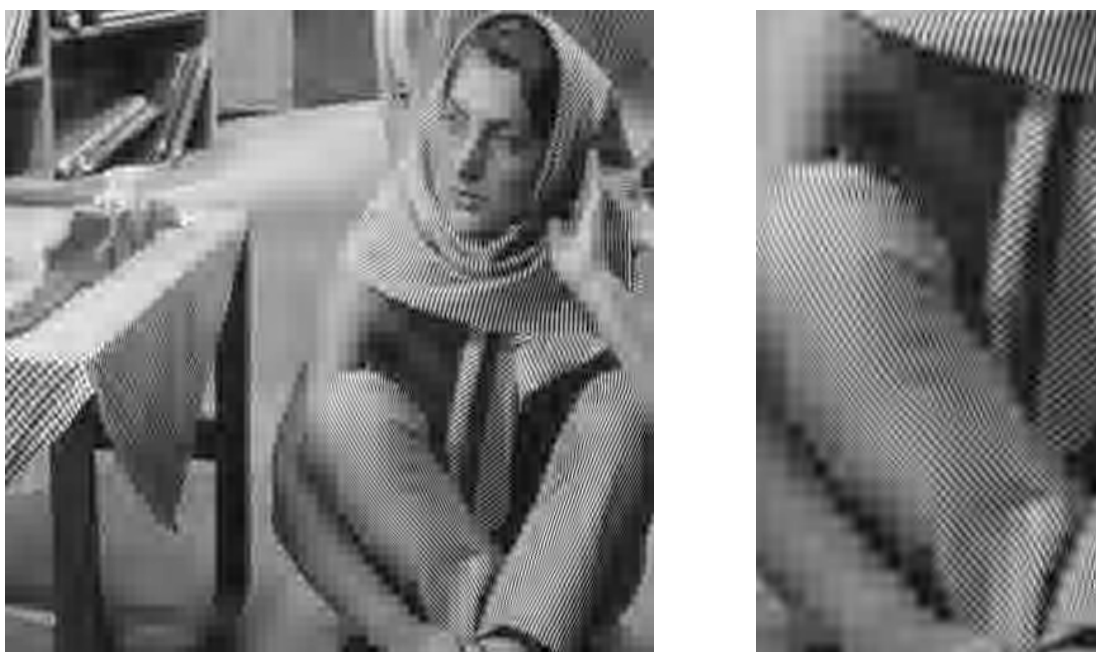


Abbildung H.179: L-CEB-Codec-II, 8×16 LCT-2 & 3-level DWT, PSNR = 25.45 dB.

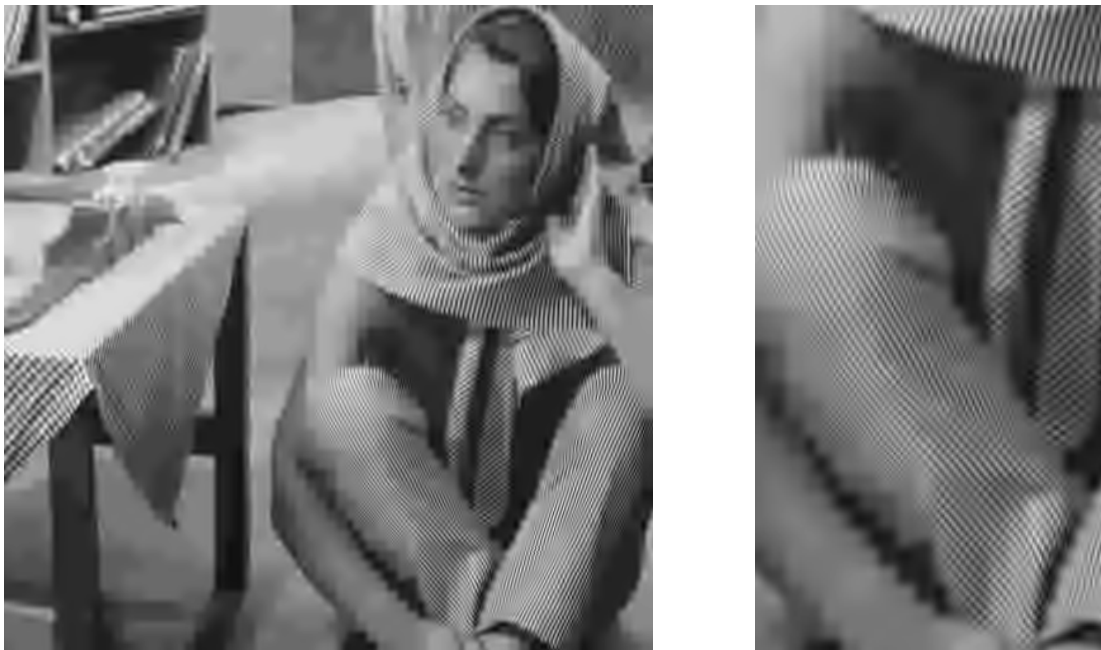


Abbildung H.180: L-CEB-Codec-I pp, 8×16 LCT-2, PSNR = 25.82 dB.

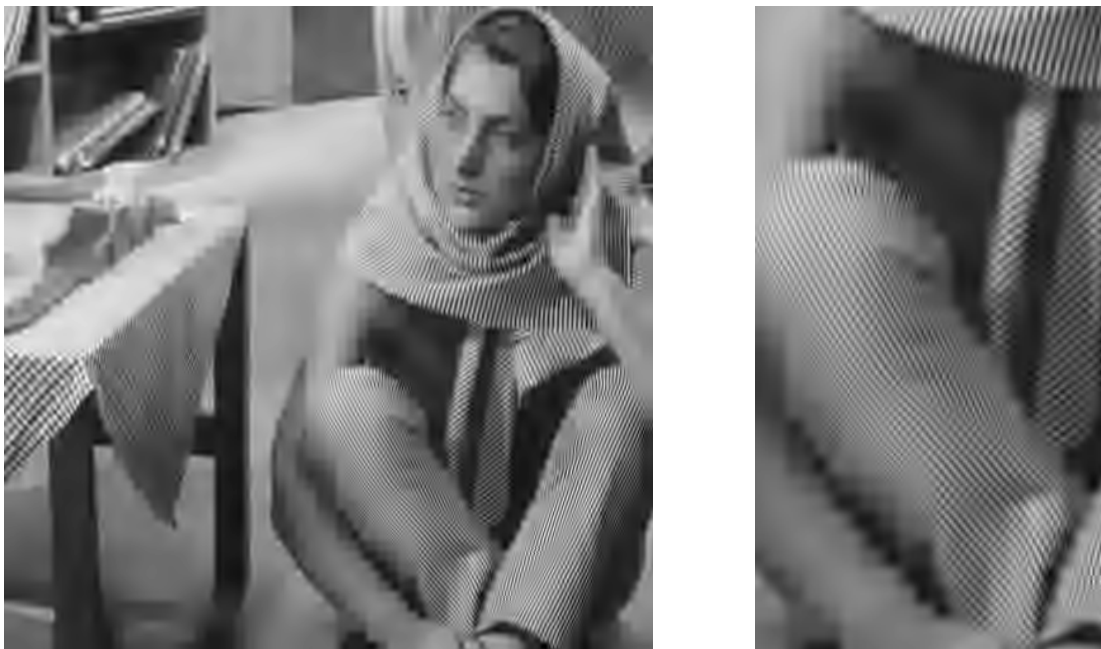


Abbildung H.181: L-CEB-Codec-II pp, 8×16 LCT-2 & 3-level DWT, PSNR = 25.88 dB.

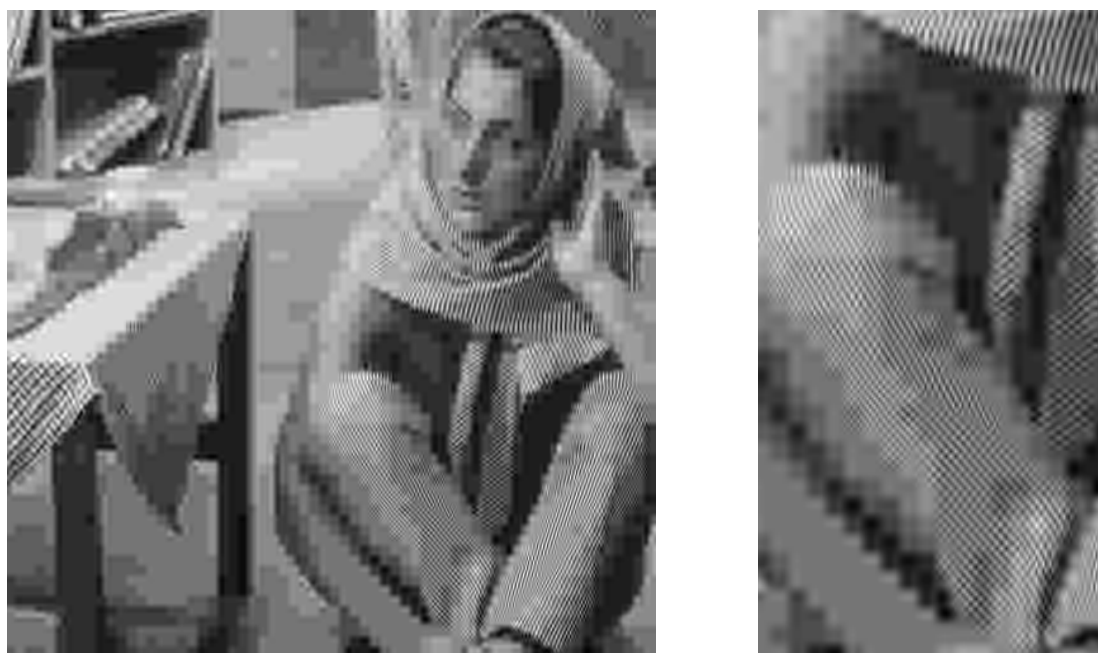


Abbildung H.182: SPIHT-Codec-I, 8×16 LCT-2, PSNR = 24.26 dB.



Abbildung H.183: SPIHT-Codec-II, 8×16 LCT-2 & 3-level DWT, PSNR = 25.07 dB.



Abbildung H.184: SPIHT-Codec-I pp, 8×16 LCT-2, PSNR = 24.62 dB.

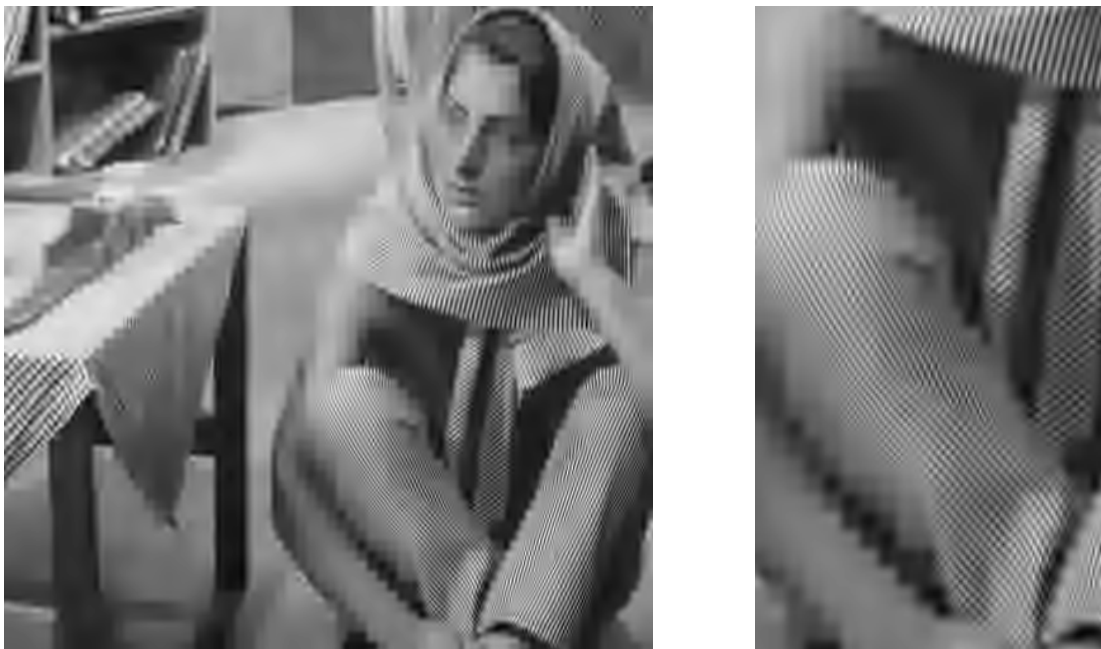


Abbildung H.185: SPIHT-Codec-II pp, 8×16 LCT-2 & 3-level DWT, PSNR = 25.37 dB.



Abbildung H.186: SPIHT-Codec-I, 16×32 LCT-2, PSNR = 25.85 dB.



Abbildung H.187: SPIHT-Codec-II, 16×32 LCT-2 & 2-level DWT, PSNR = 25.99 dB.



Abbildung H.188: SPIHT-Codec-I pp, 16×32 LCT-2, PSNR = 26.00 dB.



Abbildung H.189: SPIHT-Codec-II pp, 16×32 LCT-2 & 2-level DWT, PSNR = 26.13 dB.



Abbildung H.190: AGU-Codec-I, 16×32 LCT-2, PSNR = 26.18 dB.



Abbildung H.191: AGU-Codec-II, 16×32 LCT-2 & 2-level DWT, PSNR = 26.23 dB.



Abbildung H.192: AGU-Codec-I pp, 16×32 LCT-2, PSNR = 26.44 dB.



Abbildung H.193: AGU-Codec-II pp, 16×32 LCT-2 & 2-level DWT, PSNR = 26.49 dB.



Abbildung H.194: SPIHT-Codec-I, 32×64 LCT-2, PSNR = 26.14 dB.



Abbildung H.195: SPIHT-Codec-II, 32×64 LCT-2 & 1-level DWT, PSNR = 26.16 dB.



Abbildung H.196: SPIHT-Codex-I pp, 32×64 LCT-2, PSNR = 26.28 dB.



Abbildung H.197: SPIHT-Codex-II pp, 32×64 LCT-2 & 1-level DWT, PSNR = 26.30 dB.



Abbildung H.198: AGU-Codec-I, 32×64 LCT-2, PSNR = 26.72 dB.



Abbildung H.199: AGU-Codec-II, 32×64 LCT-2 & 1-level DWT, PSNR = 26.70 dB.



Abbildung H.200: AGU-Codec-I pp, 32×64 LCT-2, PSNR = 26.98 dB.



Abbildung H.201: AGU-Codec-II pp, 32×64 LCT-2 & 1-level DWT, PSNR = 26.96 dB.

Dreifachüberlappung, Fenster 3

Abbildung H.202: L-CEB-Codec-I, 8×24 LCT-3, PSNR = 25.59 dB.

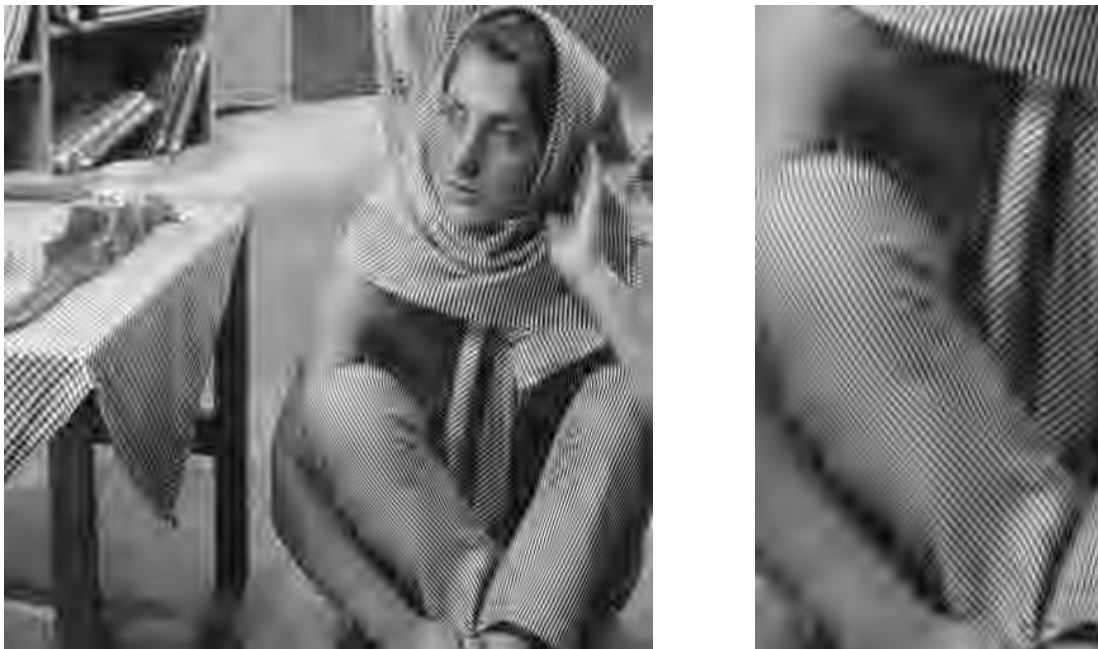


Abbildung H.203: L-CEB-Codec-II, 8×24 LCT-3 & 3-level DWT, PSNR = 25.65 dB.



Abbildung H.204: L-CEB-Codec-I pp, 8×24 LCT-3, PSNR = 26.03 dB.

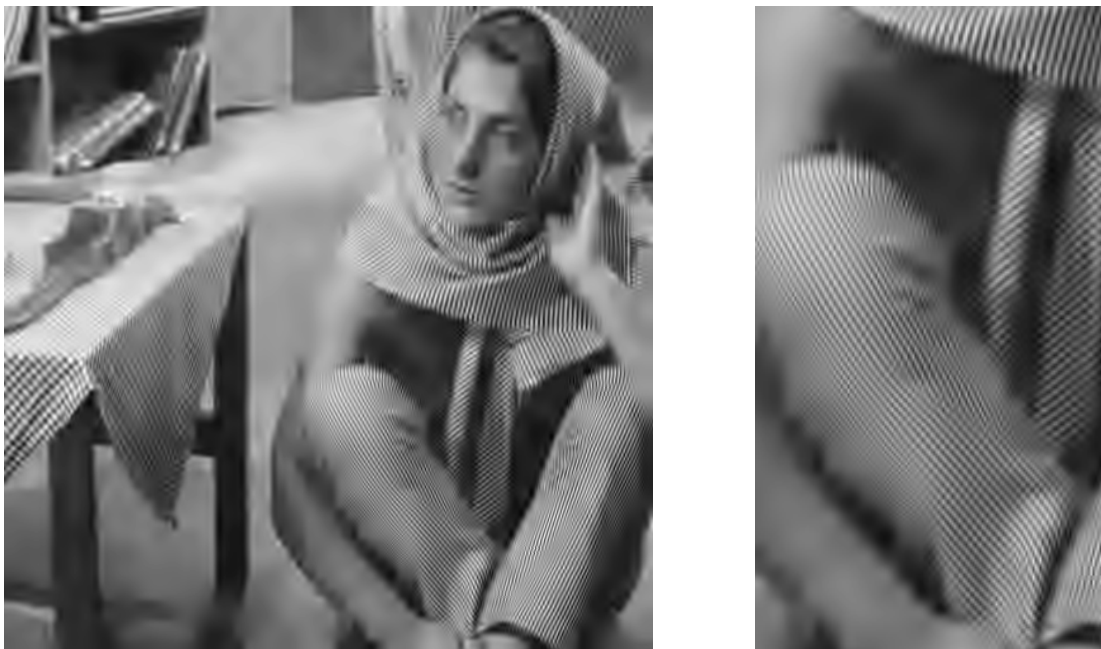


Abbildung H.205: L-CEB-Codec-II pp, 8×24 LCT-3 & 3-level DWT, PSNR = 26.06 dB.



Abbildung H.206: SPIHT-Codec-I, 8×24 LCT-3, PSNR = 23.99 dB.



Abbildung H.207: SPIHT-Codec-II, 8×24 LCT-3 & 3-level DWT, PSNR = 25.17 dB.



Abbildung H.208: SPIHT-Codec-I pp, 8×24 LCT-3, PSNR = 24.35 dB.

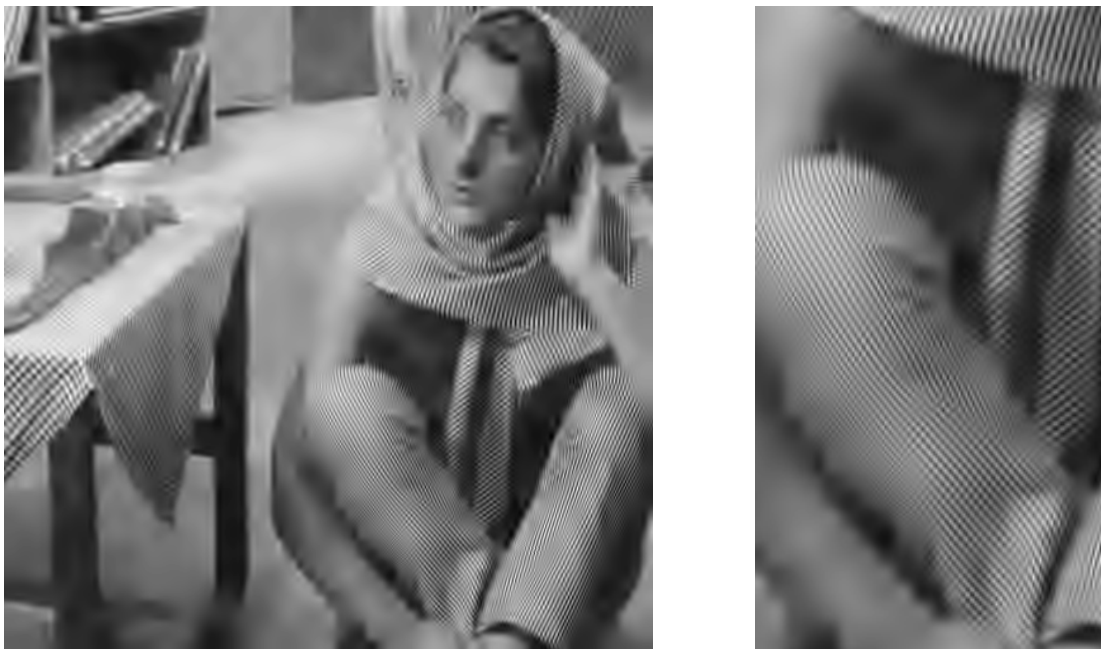


Abbildung H.209: SPIHT-Codec-II pp, 8×24 LCT-3 & 3-level DWT, PSNR = 25.34 dB.



Abbildung H.210: SPIHT-Codec-I, 16×48 LCT-3, PSNR = 25.52 dB.



Abbildung H.211: SPIHT-Codec-II, 16×48 LCT-3 & 2-level DWT, PSNR = 25.78 dB.



Abbildung H.212: SPIHT-Codec-I pp, 16×48 LCT-3, PSNR = 25.73 dB.



Abbildung H.213: SPIHT-Codec-II pp, 16×48 LCT-3 & 2-level DWT, PSNR = 26.08 dB.



Abbildung H.214: AGU-Codec-I, 16×48 LCT-3, PSNR = 25.91 dB.



Abbildung H.215: AGU-Codec-II, 16×48 LCT-3 & 2-level DWT, PSNR = 25.97 dB.



Abbildung H.216: AGU-Codec-I pp, 16×48 LCT-3, PSNR = 26.27 dB.



Abbildung H.217: AGU-Codec-II pp, 16×48 LCT-3 & 2-level DWT, PSNR = 26.32 dB.

Vierfachüberlappung



Abbildung H.218: L-CEB-Codec-I, 8×32 LCT-4, PSNR = 26.09 dB.

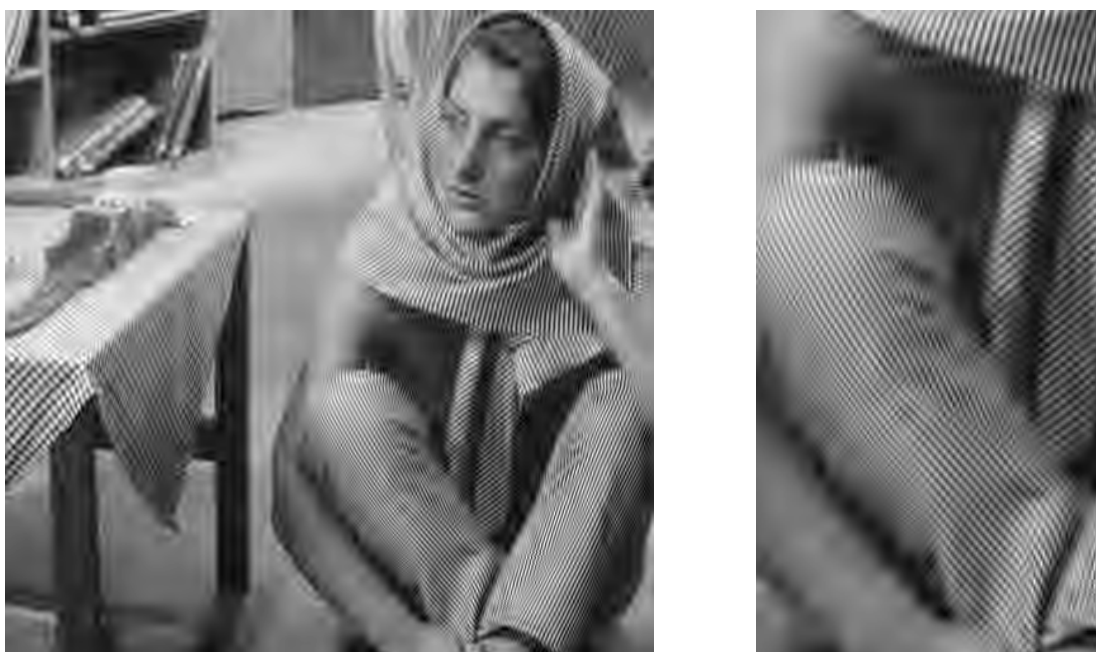


Abbildung H.219: L-CEB-Codec-II, 8×32 LCT-4 & 3-level DWT, PSNR = 26.16 dB.



Abbildung H.220: L-CEB-Codec-I pp, 8×32 LCT-4, PSNR = 26.33 dB.



Abbildung H.221: L-CEB-Codec-II pp, 8×32 LCT-4 & 3-level DWT, PSNR = 26.38 dB.



Abbildung H.222: SPIHT-Codec-I, 8×32 LCT-4, PSNR = 24.65 dB.



Abbildung H.223: SPIHT-Codec-II, 8×32 LCT-4 & 3-level DWT, PSNR = 25.55 dB.



Abbildung H.224: SPIHT-Codec-I pp, 8×32 LCT-4, PSNR = 24.90 dB.



Abbildung H.225: SPIHT-Codec-II pp, 8×32 LCT-4 & 3-level DWT, PSNR = 25.66 dB.



Abbildung H.226: SPIHT-Codec-I, 16×64 LCT-4, PSNR = 25.85 dB.



Abbildung H.227: SPIHT-Codec-II, 16×64 LCT-4 & 2-level DWT, PSNR = 26.03 dB.



Abbildung H.228: SPIHT-Codec-I pp, 16×64 LCT-4, PSNR = 25.99 dB.



Abbildung H.229: SPIHT-Codec-II pp, 16×64 LCT-4 & 2-level DWT, PSNR = 26.17 dB.



Abbildung H.230: AGU-Codec-I, 16×64 LCT-4, PSNR = 26.30 dB.



Abbildung H.231: AGU-Codec-II, 16×64 LCT-4 & 2-level DWT, PSNR = 26.37 dB.



Abbildung H.232: AGU-Codec-I pp, 16×64 LCT-4, PSNR = 26.54 dB.



Abbildung H.233: AGU-Codec-II pp, 16×64 LCT-4 & 2-level DWT, PSNR = 26.61 dB.

Sechsfachüberlappung



Abbildung H.234: L-CEB-Codec-I, 8×48 LCT-6, PSNR = 25.84 dB.



Abbildung H.235: L-CEB-Codec-II, 8×48 LCT-6 & 3-level DWT, PSNR = 25.99 dB.



Abbildung H.236: L-CEB-Codec-I pp, 8×48 LCT-6, PSNR = 26.20 dB.



Abbildung H.237: L-CEB-Codec-II pp, 8×48 LCT-6 & 3-level DWT, PSNR = 26.34 dB.



Abbildung H.238: SPIHT-Codec-I, 8×48 LCT-6, PSNR = 24.03 dB.

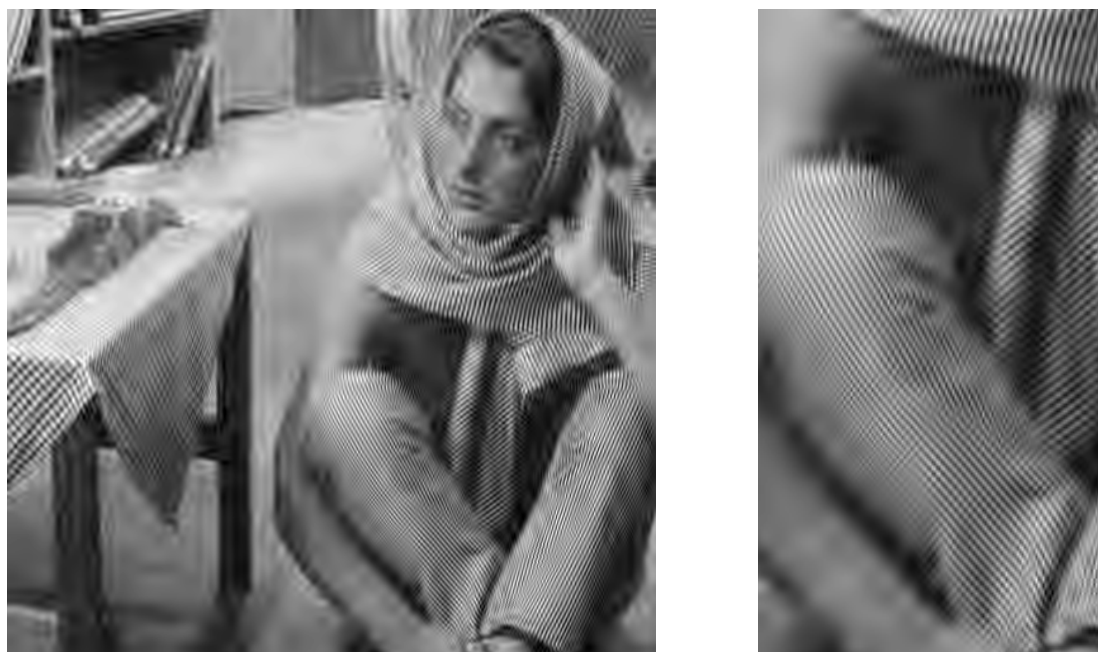


Abbildung H.239: SPIHT-Codec-II, 8×48 LCT-6 & 3-level DWT, PSNR = 25.54 dB.



Abbildung H.240: SPIHT-Codex-I pp, 8×48 LCT-6, PSNR = 24.41 dB.



Abbildung H.241: SPIHT-Codex-II pp, 8×48 LCT-6 & 3-level DWT, PSNR = 25.67 dB.



Abbildung H.242: SPIHT-Codec-I, 16×96 LCT-6, PSNR = 25.45 dB.



Abbildung H.243: SPIHT-Codec-II, 16×96 LCT-6 & 2-level DWT, PSNR = 25.82 dB.



Abbildung H.244: SPIHT-Codec-II pp, 16×96 LCT-6 & 2-level DWT, PSNR = 25.72 dB.



Abbildung H.245: SPIHT-Codec-II pp, 16×96 LCT-6 & 2-level DWT, PSNR = 26.00 dB.



Abbildung H.246: AGU-Codec-I, 16×96 LCT-6, PSNR = 25.79 dB.



Abbildung H.247: AGU-Codec-II, 16×96 LCT-6 & 2-level DWT, PSNR = 25.89 dB.



Abbildung H.248: AGU-Codec-I pp, 16×96 LCT-6, PSNR = 26.22 dB.



Abbildung H.249: AGU-Codec-II pp, 16×96 LCT-6 & 2-level DWT, PSNR = 26.32 dB.

TDLT

Abbildung H.250: L-CEB-Codec-I, 8×16 TDLT, PSNR = 25.57 dB.



Abbildung H.251: L-CEB-Codec-II, 8×16 TDLT & 3-level DWT, PSNR = 25.62 dB.



Abbildung H.252: L-CEB-Codec-I pp, 8×16 TDLT, PSNR = 25.83 dB.



Abbildung H.253: L-CEB-Codec-II pp, 8×16 TDLT & 3-level DWT, PSNR = 25.85 dB.



Abbildung H.254: SPIHT-Codec-I, 8×16 TDLT, PSNR = 24.04 dB.



Abbildung H.255: SPIHT-Codec-II, 8×16 TDLT & 3-level DWT, PSNR = 25.33 dB.



Abbildung H.256: SPIHT-Codec-I pp, 8×16 TDLT, PSNR = 24.21 dB.

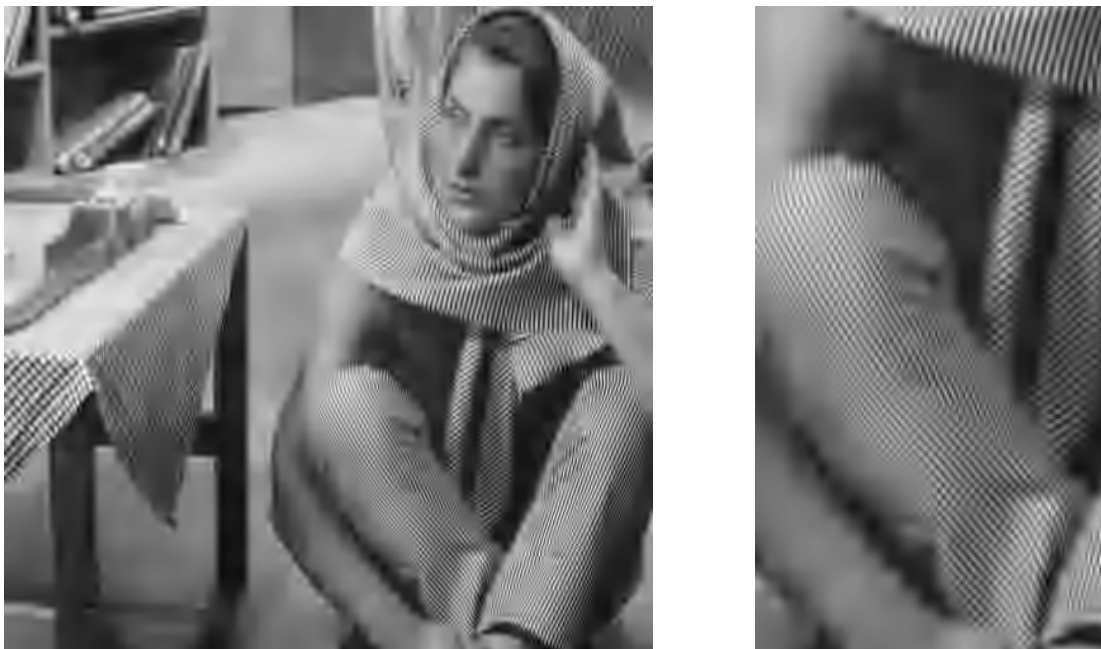


Abbildung H.257: SPIHT-Codec-II pp, 8×16 TDLT & 3-level DWT, PSNR = 25.49 dB.

LOT Typ II

Abbildung H.258: L-CEB-Codec-I, 8×16 LOT-II, PSNR = 25.33 dB.



Abbildung H.259: L-CEB-Codec-II, 8×16 LOT-II & 3-level DWT, PSNR = 25.37 dB.



Abbildung H.260: L-CEB-Codec-I pp, 8×16 LOT-II, PSNR = 25.85 dB.



Abbildung H.261: L-CEB-Codec-II pp, 8×16 LOT-II & 3-level DWT, PSNR = 25.85 dB.



Abbildung H.262: SPIHT-Codec-I, 8×16 LOT-II, PSNR = 24.14 dB.



Abbildung H.263: SPIHT-Codec-II, 8×16 LOT-II & 3-level DWT, PSNR = 25.00 dB.



Abbildung H.264: SPIHT-Codec-I pp, 8×16 LOT-II, PSNR = 24.50 dB.



Abbildung H.265: SPIHT-Codec-II pp, 8×16 LOT-II & 3-level DWT, PSNR = 25.37 dB.

GenLOT

Abbildung H.266: L-CEB-Codec-I, 8×32 GenLOT, PSNR = 25.53 dB.



Abbildung H.267: L-CEB-Codec-II, 8×32 GenLOT & 3-level DWT, PSNR = 25.57 dB.



Abbildung H.268: L-CEB-Codec-I pp, 8×32 GenLOT, PSNR = 25.93 dB.



Abbildung H.269: L-CEB-Codec-II pp, 8×32 GenLOT & 3-level DWT, PSNR = 25.96 dB.



Abbildung H.270: SPIHT-Codec-I, 8×32 GenLOT, PSNR = 24.41 dB.



Abbildung H.271: SPIHT-Codec-II, 8×32 GenLOT & 3-level DWT, PSNR = 25.18 dB.



Abbildung H.272: SPIHT-Codec-I pp, 8×32 GenLOT, PSNR = 24.77 dB.



Abbildung H.273: SPIHT-Codec-II pp, 8×32 GenLOT & 3-level DWT, PSNR = 25.46 dB.



Abbildung H.274: L-CEB-Codec-I, 8×48 GenLOT, PSNR = 25.89 dB.



Abbildung H.275: L-CEB-Codec-II, 8×48 GenLOT & 3-level DWT, PSNR = 25.91 dB.



Abbildung H.276: L-CEB-Codec-I pp, 8×48 GenLOT, PSNR = 26.20 dB.

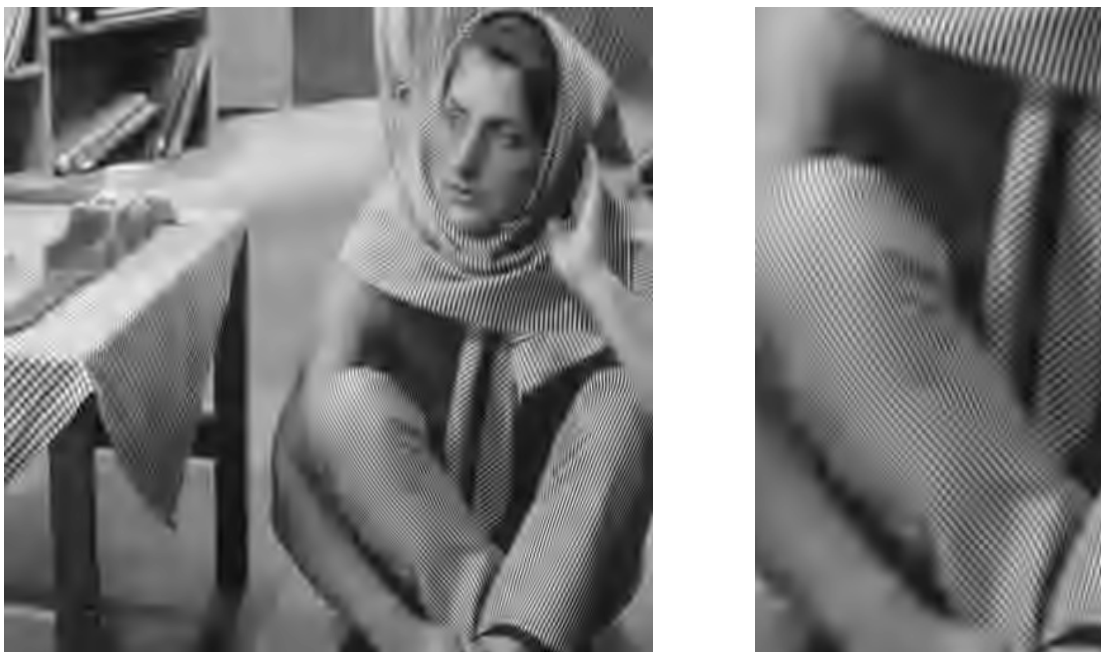


Abbildung H.277: L-CEB-Codec-II pp, 8×48 GenLOT & 3-level DWT, PSNR = 26.21 dB.



Abbildung H.278: SPIHT-Codec-I, 8×48 GenLOT, PSNR = 24.78 dB.



Abbildung H.279: SPIHT-Codec-II, 8×48 GenLOT & 3-level DWT, PSNR = 25.52 dB.



Abbildung H.280: SPIHT-Codec-I pp, 8×48 GenLOT, PSNR = 25.02 dB.



Abbildung H.281: SPIHT-Codec-II pp, 8×48 GenLOT & 3-level DWT, PSNR = 25.69 dB.

ELT

Abbildung H.282: L-CEB-Codec-I, 8×32 ELT, PSNR = 25.99 dB.



Abbildung H.283: L-CEB-Codec-II, 8×32 ELT & 3-level DWT, PSNR = 26.06 dB.



Abbildung H.284: L-CEB-Codec-I pp, 8×32 ELT, PSNR = 26.26 dB.



Abbildung H.285: L-CEB-Codec-II pp, 8×32 ELT & 3-level DWT, PSNR = 26.32 dB.



Abbildung H.286: SPIHT-Codec-I, 8×32 ELT, PSNR = 24.75 dB.



Abbildung H.287: SPIHT-Codec-II, 8×32 ELT & 3-level DWT, PSNR = 25.59 dB.

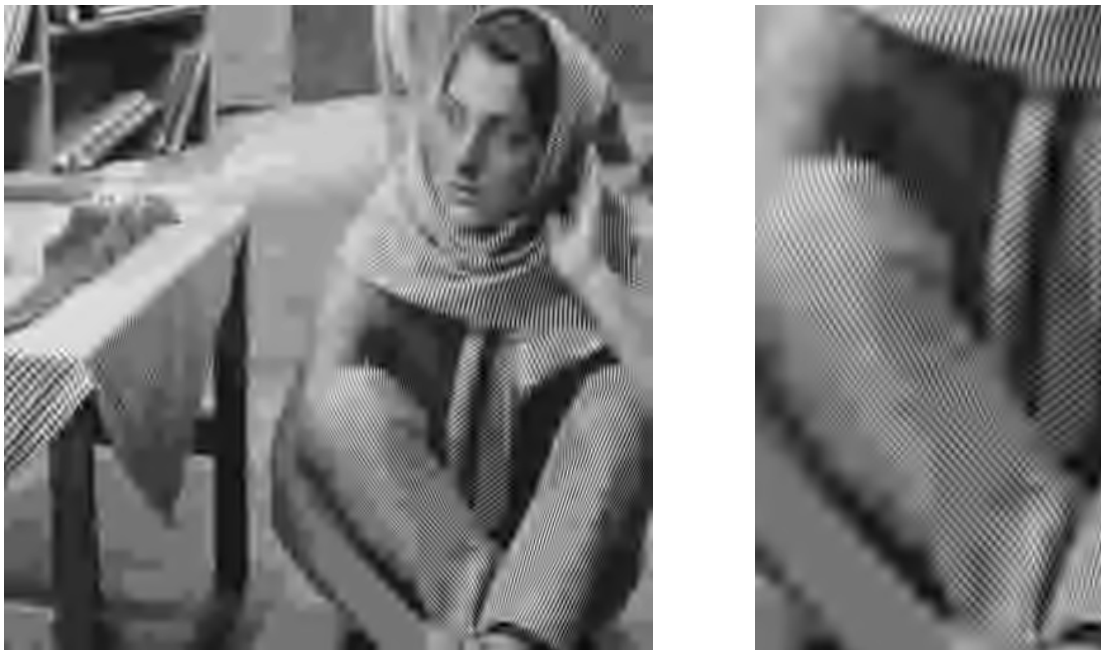


Abbildung H.288: SPIHT-Codec-I pp, 8×32 ELT, PSNR = 25.01 dB.



Abbildung H.289: SPIHT-Codec-II pp, 8×32 ELT & 3-level DWT, PSNR = 25.73 dB.

Wavelet Transformation, 9/7-Wavelet

Abbildung H.290: L-CEB-Codec-I, 3-level DWT, PSNR = 24.15 dB.



Abbildung H.291: L-CEB-Codec-II, 6-level DWT, PSNR = 24.22 dB.



Abbildung H.292: L-CEB-Codec-I pp, 3-level DWT, PSNR = 24.42 dB.



Abbildung H.293: L-CEB-Codec-II pp, 6-level DWT, PSNR = 24.50 dB.



Abbildung H.294: AGU-Codec-I, 5-level DWT, PSNR = 24.47 dB.



Abbildung H.295: AGU-Codec-II, 6-level DWT, PSNR = 24.47 dB.



Abbildung H.296: AGU-Codec-I pp, 5-level DWT, PSNR = 24.67 dB.



Abbildung H.297: AGU-Codec-II pp, 6-level DWT, PSNR = 24.66 dB.



Abbildung H.298: SPIHT, 3-level DWT, PSNR = 23.56 dB.



Abbildung H.299: SPIHT, 6-level DWT, PSNR = 24.25 dB.



Abbildung H.300: SPIHT pp, 3-level DWT, PSNR = 23.70 dB.



Abbildung H.301: SPIHT pp, 6-level DWT, PSNR = 24.38 dB.



Abbildung H.302: JPEG2000 Coder, PSNR = 24.63 dB.

H.3.2 Codec-I und Codec-II im Vergleich

Testbild „Lena“ bei einer Bitrate von 0.0625 bpp

L-CEB-Codec-I, 8×8 DCT
PSNR = 26.90 dB



L-CEB-Codec-II, 8×8 DCT
PSNR = 27.04 dB



L-CEB-Codec-I, 8×16 LCT-2
PSNR = 27.68 dB



L-CEB-Codec-II, 8×16 LCT-2
PSNR = 27.84 dB



Abbildung H.303: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-I, 8×32 LCT-4
PSNR = 28.21 dB



L-CEB-Codec-II, 8×32 LCT-4
PSNR = 28.33 dB



L-CEB-Codec-I, 9/7-Wavelet
PSNR = 28.14 dB



L-CEB-Codec-II, 9/7-Wavelet
PSNR = 28.26 dB



Abbildung H.304: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

Testbild „Zelda“ bei einer Bitrate von 0.04 bpp

L-CEB-Codec-I, 8×8 DCT
PSNR = 27.75 dB



L-CEB-Codec-II, 8×8 DCT
PSNR = 28.55 dB



L-CEB-Codec-I, 8×16 LCT-2
PSNR = 28.84 dB



L-CEB-Codec-II, 8×16 LCT-2
PSNR = 29.66 dB



Abbildung H.305: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-I, 8×32 LCT-4
PSNR = 29.70 dB



L-CEB-Codec-II, 8×32 LCT-4
PSNR = 30.56 dB



L-CEB-Codec-I, 9/7-Wavelet
PSNR = 29.91 dB



L-CEB-Codec-II, 9/7-Wavelet
PSNR = 30.63 dB



Abbildung H.306: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

Testbild „Clown“ bei einer Bitrate von 0.1 bpp

L-CEB-Codec-I, 8×8 DCT
PSNR = 26.52 dB



L-CEB-Codec-II, 8×8 DCT
PSNR = 26.58 dB



L-CEB-Codec-I, 8×16 LCT-2
PSNR = 27.37 dB



L-CEB-Codec-II, 8×16 LCT-2
PSNR = 27.44 dB



Abbildung H.307: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-I, 8×32 LCT-4
PSNR = 27.65 dB



L-CEB-Codec-II, 8×32 LCT-4
PSNR = 27.80 dB



L-CEB-Codec-I, 9/7-Wavelet
PSNR = 27.17 dB



L-CEB-Codec-II, 9/7-Wavelet
PSNR = 27.23 dB



Abbildung H.308: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

H.3.3 Codec-II und JPEG2000 im Vergleich

Testbild „Lena“ bei einer Bitrate von 0.1 bpp

L-CEB-Codec-II, 8×8 DCT
PSNR = 28.84 dB



AGU-Codec-II, 32×32 DCT
PSNR = 29.82 dB



L-CEB-Codec-II, 8×16 LCT-2
PSNR = 29.73 dB



AGU-Codec-II, 16×32 LCT-2
PSNR = 30.13 dB



Abbildung H.309: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-II, 8×24 LCT-3
PSNR = 29.74 dB



L-CEB-Codec-II, 8×32 LCT-4
PSNR = 30.23 dB



L-CEB-Codec-II, 8×48 LCT-6
PSNR = 29.94 dB



JPEG2000
PSNR = 29.87 dB



Abbildung H.310: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

Testbild „Goldhill“ bei einer Bitrate von 0.3 bpp

L-CEB-Codec-II, 8×8 DCT
PSNR = 30.79 dB



AGU-Codec-II, 32×32 DCT
PSNR = 31.34 dB



L-CEB-Codec-II, 8×16 LCT-2
PSNR = 31.38 dB



AGU-Codec-II, 16×32 LCT-2
PSNR = 31.53 dB



Abbildung H.311: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-II, 8×24 LCT-3
PSNR = 31.10 dB



L-CEB-Codec-II, 8×32 LCT-4
PSNR = 31.43 dB



L-CEB-Codec-II, 8×48 LCT-6
PSNR = 31.07 dB



JPEG2000
PSNR = 31.16 dB



Abbildung H.312: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

Testbild „Lighthouse“ bei einer Bitrate von 0.25 bpp

L-CEB-Codec-II, 8×8 DCT
PSNR = 27.78 dB



AGU-Codec-II, 32×32 DCT
PSNR = 28.32 dB



L-CEB-Codec-II, 8×16 LCT-2
PSNR = 28.25 dB



AGU-Codec-II, 16×32 LCT-2
PSNR = 28.47 dB



Abbildung H.313: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-II, 8×24 LCT-3
PSNR = 27.99 dB



L-CEB-Codec-II, 8×32 LCT-4
PSNR = 28.35 dB



L-CEB-Codec-II, 8×48 LCT-6
PSNR = 28.01 dB



JPEG2000
PSNR = 27.96 dB



Abbildung H.314: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

Testbild „Roof“ bei einer Bitrate von 0.125 bpp

L-CEB-Codec-II, 8×8 DCT
PSNR = 23.94 dB



AGU-Codec-II, 32×32 DCT
PSNR = 25.87 dB



L-CEB-Codec-II, 8×16 LCT-2
PSNR = 25.32 dB



AGU-Codec-II, 16×32 LCT-2
PSNR = 26.26 dB



Abbildung H.315: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-II, 8×24 LCT-3
PSNR = 25.31 dB



L-CEB-Codec-II, 8×32 LCT-4
PSNR = 25.90 dB



L-CEB-Codec-II, 8×48 LCT-6
PSNR = 25.75 dB



JPEG2000
PSNR = 24.35 dB



Abbildung H.316: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

Testbild „Houses“ bei einer Bitrate von 0.5 bpp

L-CEB-Codec-II, 8×8 DCT
PSNR = 26.66 dB



AGU-Codec-II, 32×32 DCT
PSNR = 26.78 dB



L-CEB-Codec-II, 8×16 LCT-2
PSNR = 27.09 dB



AGU-Codec-II, 16×32 LCT-2
PSNR = 27.04 dB



Abbildung H.317: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-II, 8×24 LCT-3
PSNR = 26.50 dB



L-CEB-Codec-II, 8×32 LCT-4
PSNR = 26.95 dB



L-CEB-Codec-II, 8×48 LCT-6
PSNR = 26.36 dB



JPEG2000
PSNR = 26.76 dB



Abbildung H.318: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

Testbild „Peppers“ bei einer Bitrate von 0.1 bpp

L-CEB-Codec-II, 8×8 DCT
PSNR = 28.25 dB



AGU-Codec-II, 32×32 DCT
PSNR = 29.08 dB



L-CEB-Codec-II, 8×16 LCT-2
PSNR = 29.04 dB



AGU-Codec-II, 16×32 LCT-2
PSNR = 29.41 dB



Abbildung H.319: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-II, 8×24 LCT-3
PSNR = 29.24 dB



L-CEB-Codec-II, 8×32 LCT-4
PSNR = 29.51 dB



L-CEB-Codec-II, 8×48 LCT-6
PSNR = 28.99 dB



JPEG2000
PSNR = 29.64 dB



Abbildung H.320: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

H.3.4 Reduktion von Kompressionsartefakten

Testbild „Lena“ bei einer Bitrate von 0.125 bpp

AGU-Codec-II, 32×32 DCT
PSNR = 30.76 dB



AGU-Codec-II pp, 32×32 DCT
PSNR = 31.28 dB



AGU-Codec-II, 16×32 LCT-2
PSNR = 31.10 dB



AGU-Codec-II pp, 16×32 LCT-2
PSNR = 31.51 dB



Abbildung H.321: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-II, 8×32 LCT-4
PSNR = 31.18 dB



L-CEB-Codec-II pp, 8×32 LCT-4
PSNR = 31.53 dB



SPIHT, 6-level DWT, 9/7-Wavelet
PSNR = 31.09 dB



SPIHT pp, 6-level DWT, 9/7-Wavelet
PSNR = 31.24 dB



Abbildung H.322: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

Testbild „Lena“ bei einer Bitrate von 0.0625 bpp

AGU-Codec-II, 32×32 DCT
PSNR = 27.92 dB



AGU-Codec-II pp, 32×32 DCT
PSNR = 28.44 dB



AGU-Codec-II, 16×32 LCT-2
PSNR = 28.26 dB



AGU-Codec-II pp, 16×32 LCT-2
PSNR = 28.65 dB



Abbildung H.323: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-II, 8×32 LCT-4
PSNR = 28.33 dB



L-CEB-Codec-II pp, 8×32 LCT-4
PSNR = 28.71 dB



SPIHT, 6-level DWT, 9/7-Wavelet
PSNR = 28.36 dB



SPIHT pp, 6-level DWT, 9/7-Wavelet
PSNR = 28.48 dB



Abbildung H.324: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

Testbild „Roof“ bei einer Bitrate von 0.1 bpp

AGU-Codec-II, 32×32 DCT
PSNR = 24.92 dB



AGU-Codec-II pp, 32×32 DCT
PSNR = 25.16 dB



AGU-Codec-II, 16×32 LCT-2
PSNR = 25.30 dB



AGU-Codec-II pp, 16×32 LCT-2
PSNR = 25.43 dB



Abbildung H.325: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-II, 8×32 LCT-4
PSNR = 24.91 dB



L-CEB-Codec-II pp, 8×32 LCT-4
PSNR = 25.07 dB



SPIHT, 6-level DWT, 9/7-Wavelet
PSNR = 23.07 dB



SPIHT pp, 6-level DWT, 9/7-Wavelet
PSNR = 23.23 dB



Abbildung H.326: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

Testbild „Peppers“ bei einer Bitrate von 0.0625 bpp

AGU-Codec-II, 32×32 DCT
PSNR = 27.17 dB



AGU-Codec-II pp, 32×32 DCT
PSNR = 27.85 dB



AGU-Codec-II, 16×32 LCT-2
PSNR = 27.43 dB



AGU-Codec-II pp, 16×32 LCT-2
PSNR = 27.98 dB



Abbildung H.327: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

L-CEB-Codec-II, 8×32 LCT-4
PSNR = 27.50 dB



L-CEB-Codec-II pp, 8×32 LCT-4
PSNR = 28.04 dB



SPIHT, 6-level DWT, 9/7-Wavelet
PSNR = 27.56 dB



SPIHT pp, 6-level DWT, 9/7-Wavelet
PSNR = 27.76 dB



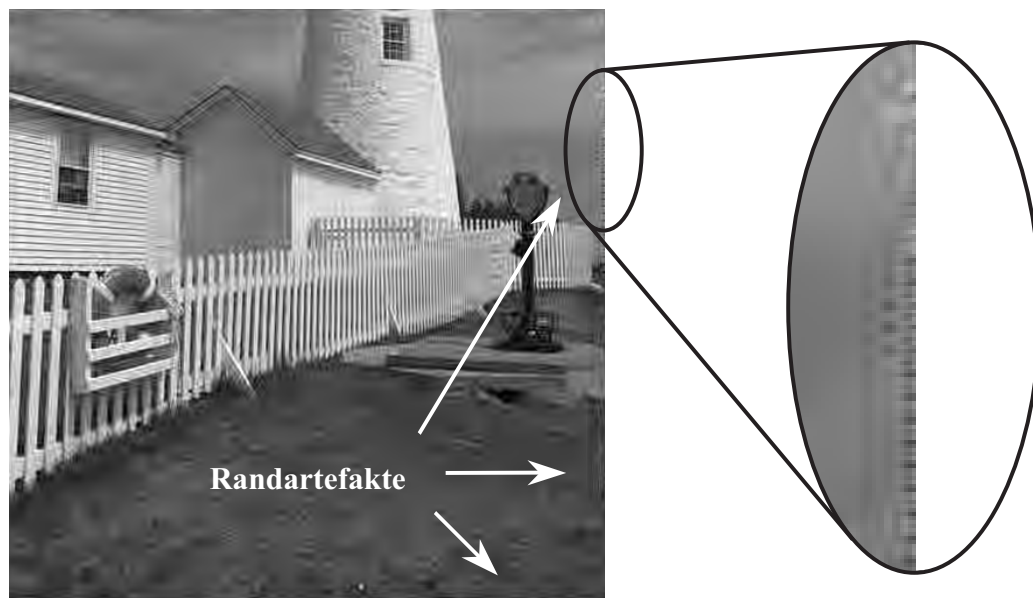
Abbildung H.328: Rekonstruierte Bilder bei verschiedenen Kompressionsstrategien.

H.3.5 Basen auf dem Intervall und Signalfortsetzung im Vergleich

Verwendung von vierfach überlappenden Kosinus-IV-Basen auf der reellen Achse:

Das Signal wird durch Periodisierung fortgesetzt. Es entstehen sichtbare Randartefakte.

PSNR = 25.96 dB, R_8 -PSNR = 24.96 dB, R_{12} -PSNR = 25.48 dB, R_{16} -PSNR = 25.77 dB.



Verwendung von vierfach überlappenden Kosinus-IV-Kosinus-II-Basen auf dem Intervall:

Eine Signalfortsetzung ist nicht erforderlich. Es entstehen keine Randartefakte.

PSNR = 26.24 dB, R_8 -PSNR = 27.33 dB, R_{12} -PSNR = 27.23 dB, R_{16} -PSNR = 27.19 dB.

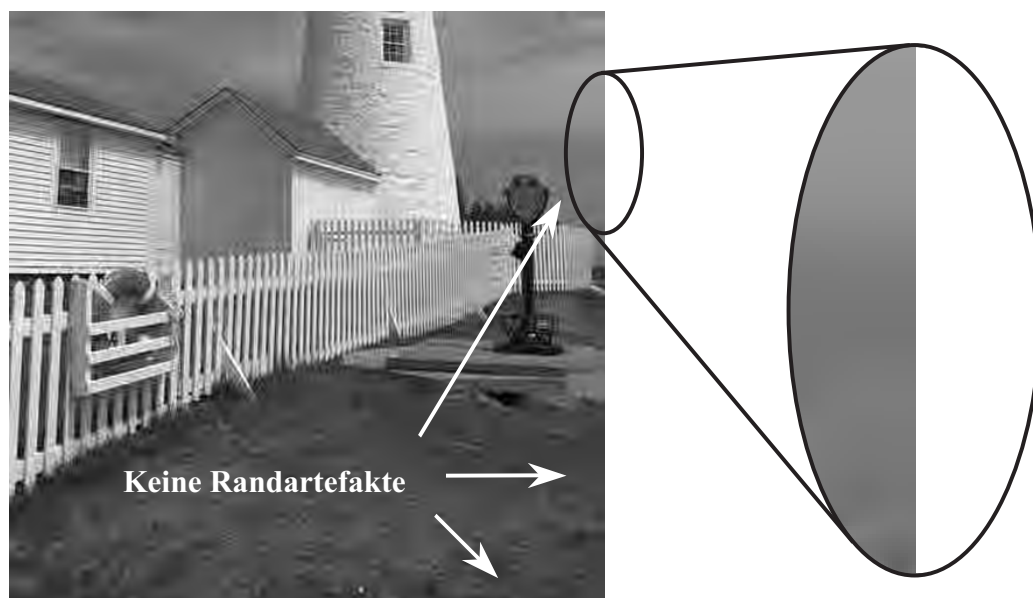


Abbildung H.329: L-CEB-Codec-II: Kompression des Bildes „Lighthouse“ bei einer Bitrate von 0.125 bpp; oben: Nutzung von Basisfunktionen, die auf der reellen Achse definiert sind; unten: Nutzung von Basisfunktionen, die auf dem Intervall definiert sind.

Thesen

Mit der zunehmenden Verbreitung von Systemen zur Erfassung, Digitalisierung, Speicherung und Übertragung visueller Informationen gewinnt die Entwicklung leistungsfähiger Verfahren zur Kompression von Bild- und Videodaten immer mehr an Bedeutung.

Überlappende trigonometrische Basisfunktionen

1. Mehrfach überlappende lokale trigonometrische Basisfunktionen eignen sich für die Entwicklung leistungsfähiger Verfahren zur verlustbehafteten Bilddatenkompression.
2. Bei der Darstellung von Bildern durch trigonometrische Basisfunktionen lassen sich sichtbare Blockartefakte vermeiden, indem das Überlappen benachbarter Basisfunktionen zugelassen wird.
3. Die Mehrfachüberlappung bewirkt im Vergleich zur Zweifachüberlappung eine bessere Reduktion von Blockartefakten.
4. Überlappende trigonometrische Basen erlauben eine effiziente Darstellung von höherfrequenten Bildinhalten mit wiederkehrenden Mustern.
5. Die Eigenschaft der mehrfach überlappenden lokalen trigonometrischen Basen, Polynome höherer Ordnung durch die Basisfunktionen der niedrigsten Frequenz zu rekonstruieren, ermöglicht eine effiziente Darstellung lokal glatter Bildbestandteile.

Verarbeitung von endlich langen Signalen

6. Die Transformation von Signalen, die nur auf einem beschränkten Intervall $[0, L]$ definiert sind, mit überlappenden trigonometrischen Basen, die sich auf die gesamte reelle Achse \mathbb{R} beziehen, erfordert Signalwerte, die außerhalb des beschränkten Intervalls $[0, L]$ liegen und somit nicht definiert sind.
7. Die Definition der erforderlichen Signalwerte durch periodische Signalfortsetzung oder symmetrische periodische Signalfortsetzung stellt für die Anwendung der untersuchten überlappenden Kosinus-IV-Basisfunktionen in der Bilddatenkompression keine geeignete Methode dar.
8. Basen auf dem Intervall, die aus bestehenden überlappenden Kosinus-IV-Basisfunktionen und aus neu entwickelten überlappenden Kosinus-IV- und Kosinus-II-Randbasisfunktionen gebildet werden, eignen sich zur effizienten Darstellung von endlich langen Signalen.
9. Die Transformation von endlich langen Signalen $f \in L^2([0, L])$ mit Kosinus-IV-Kosinus-II-Basen auf dem Intervall $[0, L]$ erfordert keine Signalwerte, die nicht definiert sind.
10. Bei der Anwendung von überlappenden Kosinus-IV-Basisfunktionen in der verlustbehafteten Bilddatenkompression wird durch die Nutzung der Kosinus-IV- und Kosinus-II-Randbasisfunktionen auch an den Bildrändern eine gute Bilddarstellung erzielt. Es treten keine sichtbaren Randartefakte auf und es wird kein zusätzlicher Codieraufwand für die gute Darstellung der Bildränder benötigt.

Schnelle überlappende trigonometrische Transformationen

11. Für den praktischen Einsatz von transformationsbasierten Kompressionsverfahren in Systemen, die nur über vergleichsweise geringe Ressourcen verfügen, werden Transformationsalgorithmen mit geringer arithmetischer Komplexität benötigt.
12. Die Klapp- und Aufklappoperatoren, die für die neu eingeführten Kosinus-IV-Kosinus-II-Basen auf dem Intervall hergeleitet wurden, ermöglichen die Entwicklung schneller Algorithmen zur effizienten Realisierung der überlappenden Kosinustransformation von endlich langen Signalen.
13. Klapp- und Aufklappoperatoren erlauben die Nutzung schneller diskreter trigonometrischer Blocktransformationen.
14. Für die Entwicklung von Algorithmen zur effizienten Realisierung der Klapp- und Aufklappoperatoren sind Fensterfunktionen mit kurzem Träger von Vorteil.
15. Für duale Fenster mit langem Träger lassen sich schnelle Klappoperatoren entwickeln, die auf dem Lösen von linearen Gleichungssystemen mit Bandstruktur unter Verwendung von LU-Faktorisierungsverfahren beruhen.
16. Bei der Transformation von Bilddaten ist der relative Berechnungsaufwand für die LU-Faktorisierung verschwindend gering.

Kompressionsverfahren

17. Bei der verlustbehafteten Transformationscodierung haben sowohl die Darstellung wesentlicher Bildinhalte durch Überlagerung von wenigen signifikanten Basisfunktionen als auch die effiziente Codierung der quantisierten Wichtungskoeffizienten entscheidenden Einfluss auf die Leistungsfähigkeit des Kompressionsverfahrens.
18. Zur Entwicklung leistungsfähiger LCT-basierter Kompressionsverfahren kann auf bestehende effiziente blockangepasste oder waveletangepasste Codierungsstrategien zurückgegriffen werden. Die Nutzung waveletangepasster Codierungsstrategien wird durch eine Umsortierung der LCT-Koeffizienten ermöglicht.
19. Die Anwendung der 8×24 LCT-3, 8×32 LCT-4 und 8×48 LCT-6 führt bei der Rekonstruktion tieffrequenter Bildanteile zu Kompressionsartefakten in Form von schnellen Grauwertwechseln wie sie auch bei Anwendung der 3-level 9/7-DWT auftreten.
20. Die Wavelet-Zerlegung der LCT-Koeffizienten der niedrigsten Frequenz stellt ein geeignetes Verfahren zur Reduktion der genannten Kompressionsartefakte dar.
21. Zu hohe Überlappungsfaktoren und Blockgrößen führen zu deutlich sichtbaren Ringing-Artefakten.
22. Beim Vergleich zwischen 8×8 DCT, 8×16 LCT-2, 8×24 LCT-3, 8×32 LCT-4 und 8×48 LCT-6 wird im Allgemeinen die höchste Rekonstruktionsqualität mit der 8×32 LCT-4 erzielt.
23. Bereits das Ersetzen der Wavelet-Transformation in einem Wavelet-Codec wie dem SPIHT-Codec durch die vierfach überlappende Kosinustransformation führt bei fast allen untersuchten Testbildern zu einer Erhöhung der Rekonstruktionsqualität.