

A Framework for Hull Form Reverse Engineering and Geometry Integration into Numerical Simulations

Rumpfform Reverse Engineering und deren geometrische

Integration mit numerischen Simulationen

Doctoral Thesis

for

the academic degree of

Doktor-Ingenieur (Dr.-Ing.)

submitted to the Board of the Faculty of

Mechanical Engineering and Marine Technology

of the University of Rostock

Rostock, 2016

submitted by

Desta Milkessa Edessa

Born on 01.06.1984 in Ambo/Ethiopia

Gutachter:

1. Gutachter:

Prof. Dr.-Ing. Robert Bronsart,
Fakultät für Maschinenbau und Schiffstechnik, Universität Rostock

2. Gutachter:

Prof. Dr.-Ing. habil. Nikolai Kornev,
Fakultät für Maschinenbau und Schiffstechnik, Universität Rostock

Datum der Einreichung: 22. Januar 2016

Datum der Verteidigung: 04. Juli 2016

Abstract

The thesis presents a hull form specific reverse engineering and CAD integration framework. The reverse engineering part proposes three alternative and suitable surface reconstruction approaches namely curves network, direct surface fitting, and triangulated surface. The curves network approach fits a set of cross-sectional curves to an unorganized point cloud. The direct surface fitting approach reconstructs NURBS surfaces directly from an organized point cloud. The last approach reconstructs a triangulated surface from an unorganized point cloud based on existing techniques.

The CAD integration part includes surface healing, region identification, and domain preparation strategies which used to adapt the CAD model produced by forward or reverse engineering processes to downstream application requirements. The surface healing repairs both parametric surfaces read from an IGES file format and triangulated surfaces read from STL file format. The parametric surface healing technique consists of two modules: shape fixing and sewing. Shape fixing resolves inconsistencies in entities while sewing deals with inconsistencies between entities. The triangulated surface healing resolves different flaws in triangular representation. The region identification and domain preparation specifically prepare the hull form CAD model for numerical simulation. This helps to save the cost and resources associated to the surface healing, region identification and domain preparation. In general, the developed framework bridges the point cloud data obtained from a real world object (i.e. hull form) to downstream application (i.e. hydrodynamic computation). The CAD integration repairs commonly found topological and geometrical inconsistencies that originate from reverse and forward engineering processes.

CONTENTS

Abstract	iv
List of Figures	vii
List of Tables	xi
Abbreviations	xiii
1 INTRODUCTION	1
1.1 Context and Motivation	1
1.2 Scope of the Thesis	3
1.3 Contributions	4
1.4 Thesis Overview	5
2 BACKGROUND	7
2.1 Introduction	7
2.2 Reverse Engineering Applications in Maritime Industry	7
2.3 Point Cloud Acquisition	10
2.4 Geometry Based Reverse Engineering	15
2.5 Knowledge Based Reverse Engineering	37
2.6 CAD Model Integration into Numerical Simulation	43
2.7 Base Information Model	47
2.8 Neutral File Formats	50
3 CURVES NETWORK HULL FORM REVERSE ENGINEERING	53
3.1 Introduction	53
3.2 Related Works	53
3.3 The Methodology	55
3.4 Point Cloud Pre-Processing for Curves Network Fitting	55
3.5 Curves Fitting	62
3.6 Surface Generation	64
3.7 Case Study and Discussion	67
3.8 Conclusion	71
4 DIRECT SURFACE FITTING HULL FORM REVERSE ENGINEERING	73
4.1 Introduction	73
4.2 Related Works	74
4.3 NURBS Hull Form Surface Fitting to an Unorganized Point Cloud	76
4.4 Methodology	80
4.5 Results and Discussions	83
4.6 Summary	84
5 TRIANGULATED SURFACE HULL FORM REVERSE ENGINEERING	87
5.1 Introduction	87

5.2	Related Works	87
5.3	Surface Reconstruction Evaluation	88
5.4	Hull Form Related Knowledge	93
5.5	Triangulated Surface Reconstruction	94
5.6	Discussion	102
5.7	Summary	103
6	SURFACE HEALING	105
6.1	Introduction	105
6.2	Related Works	106
6.3	Repairing Geometrical and Topological Inconsistencies	107
6.4	Summary	115
7	REGION IDENTIFICATION AND DOMAIN PREPARATION	117
7.1	Introduction	117
7.2	Related Works	117
7.3	Ship Hull Form Region Identification	117
7.4	Domain Preparation for Downstream Applications	121
7.5	Output File Format	124
7.6	Results and Discussions	125
7.7	Summary	127
8	INTEGRATION AND WORK-FLOW	131
8.1	Introduction	131
8.2	The Framework Components	131
8.3	The Components Integration	132
9	DISCUSSION AND CONCLUSION	137
9.1	General Discussion	137
9.2	Detail Discussion and Conclusion	137
9.3	Limitations	140
9.4	Further Work	140
	Bibliography	143
	Declaration of Authorship	151
	Acknowledgements	152

LIST OF FIGURES

1.1	FE and RE in the context of CAD field and the respective engineering applications.	2
1.2	The main components and workflow of the developed framework	4
2.1	The scanned aft part of the ship at a construction site for ship check purpose. . .	8
2.2	Redesigning using RE techniques.	9
2.3	Retrofitting using RE techniques.	9
2.4	Collision damage repairing using RE techniques.	10
2.5	Different types of point cloud inconsistencies.	14
2.6	Working principle of a kd-tree in 2D space.	16
2.7	Convex hull (left) and concave hull (right)	17
2.8	Point cloud down-sampling using voxelize grid method	18
2.9	Raw point cloud outliers removal and registration.	19
2.10	Point cloud normals estimation	20
2.11	Point cloud normals estimation with the Least Square Regression (left) and estimation with sharp feature sensitive computation (right).	20
2.12	Illustration for primitives segmentation	22
2.13	NURBS curves with different magnitude of weight values.	25
2.14	Different NURBS curves based on the different definition of knot vectors.	26
2.15	Typical direct NURBS surface fitting to a point cloud.	27
2.16	Alpha shapes working principle in 2D.	28
2.17	Greedy projection reconstruction principle.	29
2.18	Grid triangulation method working principle in 2D.	31
2.19	Estimation of a tangent plane from nearest neighborhood points for each points in the cloud.	32
2.20	Marching cubes method, signed distance function between arbitrary point p and its projection on the plane.	32
2.21	Marching cubes method iso-surface extraction based on cubes and oriented planes.	33
2.22	Ball pivoting illustration in 2D.	33
2.23	Illustration of Poisson surface reconstruction from an oriented 2D point cloud. . .	34
2.24	Possible flaws of the reconstructed triangular surface.	35
2.25	Hole filling and smoothing based on a point cloud.	36
2.26	Reducing a reconstructed triangular surface from a point cloud.	37
2.27	Surface smoothing.	38
2.28	Scene objects grouping for the scanned ship.	40
2.29	Hull form geometric and topological features (typical regions of a hull form). . . .	40
2.30	Hull form geometry class hierarchy.	41
2.31	Specific characteristics geometric class.	41
2.32	Hull planar features as a semantic network	43
2.33	Independent and dependent geometrical inconsistencies.	46
2.34	Independent and dependent topological inconsistencies.	46
2.35	General OCCT structure.	49
2.36	Topological entities.	49
2.37	Tolerance definition at vertex (left), edge (middle) and face (right) levels	50

2.38	Parametric surface patches read from an IGES (left) and triangulated surface representation read from a STL (right)).	51
3.1	The general layout of the developed methodology.	55
3.2	The outliers removal methods	57
3.3	The working principle of the statistical outliers removal	58
3.4	Cross-sectional points extraction method.	59
3.5	Cross-sectional points extraction algorithm chart.	59
3.6	Pre-processing (i.e. filtering, down-sampling and smoothing) of a 2D cross-sectional point cloud algorithm chart.	60
3.7	2D point sets filtering and smoothing using the angle method.	61
3.8	The angle filtering and smoothing method.	61
3.9	The rectangular centroid method with different rectangle sizes	62
3.10	Demonstration of curve interpolation.	63
3.11	Demonstration of curve approximation.	63
3.12	Surface lofting from curves network.	66
3.13	NURBS surface generation using patching method (left) and surface from curves network method (right).	66
3.14	A hull form (bow region) before and after pre-processing of a 3D point cloud which includes outliers removal, filtering and down-sampling.	67
3.15	Ship hull form point cloud division.	68
3.16	Extracted waterline and deck points.	68
3.17	Extracted transversal sections point cloud.	68
3.18	Extracted waterline sections point cloud.	68
3.19	The transversal, waterline sections, and centerline and deck curves fitted using the developed method.	68
3.20	A hull form NURBS surface generated from transversal and waterline section curves using lofting.	69
3.21	Surface creation process using the patching method from curves network.	69
3.22	Surface generation process using the surface from curves network method.	70
3.23	A ship hull form NURBS surface reconstructed from curves network using a surface from curves network method.	70
3.24	Model ship curves network RE, reconstructed from measured data	70
4.1	The effect of changing control points (a) and points on surface (b, c) position. . .	77
4.2	Direct surface fitting to a point cloud process flow.	80
4.3	Segmentation of a hull form into simple surfaces.	81
4.4	An unorganized point cloud regularization type and working principle.	81
4.5	Extraction of boundary points using the angle criterion.	82
4.6	The direct surface fitting process.	84
4.7	A reconstructed hull form after segmented using different segmentation methods. .	85
5.1	Ship hull form (aft part) surface reconstruction from a noise free point cloud. . .	92
5.2	The reconstruction of triangular surfaces from a noisy point cloud.	93
5.3	Triangulated surface reconstruction layout.	95
5.4	Noise reduction using the statistical method.	96
5.5	A point cloud down-sampling using the 3D voxel grid method.	96
5.6	Point cloud normals estimation.	97
5.7	Reconstructed objects without knowledge and primitive detection (left) and reconstruction with knowledge and detection integration.	98

5.8	Hull form primitives detection (i.e. flat of side, flat of bottom, and transom).	98
5.9	Triangular surface reconstructed using different approaches.	99
5.10	Ball pivoting reconstruction approach and its driving parameter.	100
5.11	Poisson reconstruction approach and its dependence on a normal orientation.	101
5.12	Reconstructed triangular surface post-processing.	102
5.13	Triangulated surface trimming using the centerline and weather deck planes.	102
6.1	The magnitude (all in mm) of gaps and overlaps observed in the KCS ship hull form	108
6.2	A repair algorithm for parametric surface patches (reconstructed or read from an IGES file format).	108
6.3	Surface patches topology recovery (sewing) steps.	111
6.4	Sewing tolerance management.	111
6.5	The KCS ship hull form (open shell) automatically repaired, before (left) and after (right) repair	112
6.6	A closed shell automatic repairing, the case study of a simple box with gaps and overlaps.	113
6.7	Automatic ship hull form closing and repairing steps	114
6.8	Automatic closing and repairing of the open shell hull form.	114
6.9	Hole filling using a user defined tolerance.	115
7.1	Sectioning the triangulated surfaces using the section plane.	118
7.2	Data range method region identification.	119
7.3	Determination of a surface curvature at a point.	119
7.4	Procedure used to estimate the curvature of triangulated surfaces.	120
7.5	Determination of the knuckle between two adjacent surface patches.	121
7.6	Ship hull form knuckles detected with different angles.	121
7.7	Ship hull form region identified for <i>snappyHexMesh</i>	122
7.8	Boolean operation method domain construction	123
7.9	Direct bounding domain construction.	124
7.10	Double body computational domain (left) and free surface computational domain (right)	124
7.11	Colored STL file format for <i>snappyHexMesh</i> and other softwares.	125
7.12	Colored STL file format for <i>Numeca HEXPRESS</i>	126
7.13	DTC-Container ship hull form repaired and ready for mesh generation in <i>snappyHexMesh</i>	126
7.14	Mesh generated based on the framework output. Different refinement level a) with <i>subdivision</i> = 8 and b) with <i>subdivision</i> = 16	127
7.15	Different mesh refinement level for different regions.	128
7.16	Wave elevation of DTC ship in model scale at $Fr = 0.218$	128
7.17	Total resistance of DTC in model scale at $Fr = 0.218$ over simulation time	129
8.1	The pre-processing stage of the framework.	132
8.2	The reconstruction stage of the framework.	133
8.3	Hull form with region identified and mesh generated based on it.	134
8.4	Hull form surface reconstruction from an unorganized point cloud.	135

This page intentionally left blank.

LIST OF TABLES

5.1	Different triangular surface reconstruction types and their main input parameters	90
5.2	Evaluation of reconstruction algorithms against evaluation criteria	91
5.3	Triangulated surface reconstruction using different reconstruction methods, integration of knowledge, the required pre- and post-processing processes.	104
6.1	Different ship hull form test cases	113
7.1	The colored STL file format for <i>Numeca HEXPRESS</i> (the original file format and the corresponding property file.)	127

This page intentionally left blank.

ABBREVIATIONS

FE	F orward E ngineering
RE	R everse E ngineering
CAD	C omputer A ided D esign
CFD	C omputational F luid D ynamics
CSD	C omputational S olid D ynamics
IGES	I nitial G raphics E xchange S pecification
STL	S Tereo T ithography
DoF	D egree of F reedom
ICP	I terative C losest P oint
PCA	P rincipal C omponent A nalysis
RANSAC	R andom S ampled C onsensus
NURBS	N on U niform R ational B - S pline
ES	E xtremal S urfaces
CS	C ritical S urface
VTK	V isualization T ool K it
GIS	G eographic I nformation S ystem
B-Rep	B oundary R epresentation
FEA	F inite E lement A nalysis
ITTC	I nternational T owing T ank C onference
PCL	P oint C loud L ibrary
OCCT	O pen C ASCADE T echnology
STEP	S Tandard for the E xchange of P roduct model data
VRML	V irtual R eality M odeling L anguage
OFF	O bject F ile F ormat
PLY	P olygon F ile F ormat
CSG	C onstructive S olid G eometry
CGAL	C omputational G eometry A lgorithms L ibrary
KCS	K RISO C ontainer S hip
KRISO	K orean R esearch I nstitute for S hips
RANSE	R eynolds A veraged N avier S tokes E quations
LOA	L ength O ver A ll
DTC	D uisburg T est C ase

This page intentionally left blank.

CHAPTER 1

INTRODUCTION

1.1 Context and Motivation

Nowadays the use of Computer Aided Design (CAD) in product development is inevitable, because it certainly provides increased productivity, lower product development costs, improved product quality and faster time to market. These can be achieved through CAD system's better final product and sub-assemblies visualization which speeds up the design process, leads to greater accuracy and more robust documentation of the design, allows easy to re-use design data, and ensures best practices. Product development is a process of designing, creating and marketing new products or services to benefit customers. It involves either developing a new product or improving an existing product to target a particular market segments. Any new product development gets its birth from innovative concept abstraction and logic or from existing documentations. This is a traditional Forward Engineering (FE) process which has been developed over decades and has a strong impact on the life of humankind and the environments. On the other hand, products can be optimized, re-produced, retrofitted from existing real world objects using Reverse Engineering (RE) process. It is an activity which aims to digitize a real physical object to create a virtual model. Currently, there are stable and more matured FE processes compared to RE, because FE is a daily basis product development phenomena and much more has been done compare to the development in RE processes. In addition RE involves many mathematically challenging and error-prone processes. RE with regards to product development has received extensive focus with the development of laser scanner technology, photogrammetry and other data acquisition systems. Many researchers contributed extensive work related to RE processes and much efforts have been paid to figure out robust, time saving and accurate reconstruction strategies for specific or general applications. However, there is no straightforward recipe how to obtain a good quality surface when it comes to complex surfaces. The problem starts from point cloud data acquisition systems (i.e. calibration, accuracy, placement, and multiple views), kind of physical objects (i.e. occlusion, surface finish, accessibility), the nature of data (i.e. extremely unorganized, noisy, and incomplete) and extremely non-linear problems (i.e. curve and surface fitting). The data obtained through all these procedures are the main source of information for downstream applications such as numerical analysis, production, maintenance and documentations. Therefore, it is not trivial to find a general RE method which could be applicable in all applications and for any kind of objects.

Indeed, the 3D CAD models obtained through FE or RE are usually the starting point for downstream applications such as Computational Fluid Dynamics (CFD), Computational Solid Dynamics (CSD), computational electromagnetic analysis, specification and verification of any product design. In addition to their services as virtual laboratory (i.e. CFD, CSD, etc), they play a very important role in an automatic manufacturing, inspection, maintenance and assembly operations as well, which result in a comprehensive archival design database for reference; which otherwise require expensive and time-consuming physical experimentation and documentation. Figure 1.1 shows the processes involved to reach downstream applications from real world objects through RE or from new ideas and abstractions using conventional FE processes.

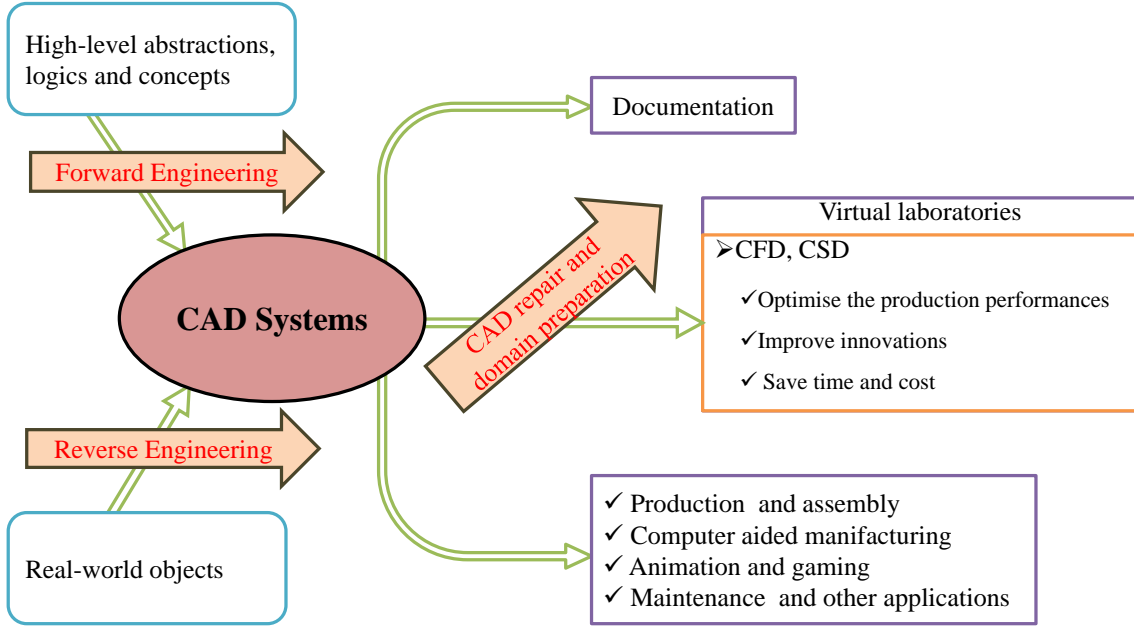


Figure 1.1: FE and RE in the context of CAD field and the respective engineering applications.

CAD has a widespread and ubiquitous application and every specific downstream application dealing with 3D geometry has its own quality requirements that restrict the direct usage of CAD models. Actually the modern CAD systems have attained a certain degree of maturity, but their efficiency, reliability, and compatibility with subsequent analysis tools remain an active research topic. At the heart of this problem lie some mathematical issues, concerned with the computation, representation, and manipulation of complex geometries, which have stubbornly resisted the best efforts of the research community to formulate rigorous and efficient solution procedures. These problems are usually introduced during RE or FE processes and often need to be analyzed and processed toward downstream applications that typically have strict requirements on the quality and integrity of their input. In practice, these requirements are often not met by models originating from these sources. Thus, adapting imperfect 3D models to such requirements is a task of high importance.

Conceptually geometric objects belong to a continuous domain, yet they are almost always analyzed by algorithms doing discrete computation. Due to numerical problems, imprecise design, software idiosyncrasies, or data exchange issues, the CAD model obtained through FE as well as RE processes may abut within unpredictable tolerances resulting in gaps, noise, overlaps, self-intersections, degenerated elements, etc. These necessitate the development of CAD repairing and adapting strategies. These CAD repairing and adapting toward downstream applications input requirements usually involve tedious, time and resource consuming tasks. Over the years, different techniques were proposed by CAD, meshing and computer graphics communities to solve CAD model inconsistencies. However it is rather vast and complicated to devise a common strategy which is able to adapt a CAD model to downstream application input requirements. In addition, such an attempt can be inevitably incomplete as particular cases and new applications arise too frequently.

Therefore, it is important to develop RE and CAD repairing methodologies for specific scenario to improve the accuracy and efficiency. In addition it helps to include expertise knowledge in RE processes as well as in CAD repairing and domain preparation steps. This work entirely considers ship hull forms as a specific application, because the use of RE in maritime industries is increasing with the rapid development of modern digitizing technologies such as laser scanners

and photogrammetry. On the other hand maritime companies, organizations and suppliers need to reconstruct CAD models from physical objects for quality control, retrofit, redesign and maintenance purposes to improve their design efficiencies so that they could meet the environmental constraints and also to be competitive in the new economic environments. The reconstructed CAD models are usually analyzed, optimized, and verified with the help of downstream applications. Therefore, an efficient CAD model adapting method to downstream application's requirements is a fundamental need.

1.2 Scope of the Thesis

RE can be used for many applications such as redesign, retrofit, inspection, maintenance, verification and historical archive. In maritime industry there are two typical situations for the need of RE: the first situation is when the documentations of the ships in service are unavailable, incomplete or in a form incompatible with CAD and manufacturing softwares, and the redesign, maintenance and retrofitting processes are required. The second is when as-built shape of the ship hull forms or other desired parts are required. In general the main aims of ship hull form RE is to ensure the production quality control and/or to optimize the performances using virtual laboratories. For instance, despite the fact that hull forms currently under construction are designed based on 3D data, they suffer the same lack of documentation of their true as-built geometries. Due to an inherent inaccuracy of the building process and the tolerances of engineering techniques involved, there is always a mismatch between design and as-built data. The actual as-built shape of the ship, from the hull to the layout of internal features such as piping does not accurately reflect prints (i.e. design data). Therefore, precise RE process offers marine engineers a way to ensure whether their ships are built correctly from the very beginning to the end, this ultimately save time and cost. On the other hand nowadays simulation based ship design approach is taking a lion's share at early stage of the ship design process while build-and-test approach comes at the final stage. Therefore, CFD computations are extensively used to check the hydrodynamic performances. In this context, the modifications applied in reconstructed old ships and newly as-built hydrodynamic performances are mainly verified using CFD.

The goal of the thesis is to develop a framework which efficiently bridges the point cloud data from the real world object (i.e. hull form) to the downstream application (i.e. hydrodynamic simulation). The developed framework comprises three alternative RE processes namely curves network, triangulated surface and direct fitting method. In addition, it includes CAD repairing, region identification and further domain preparation strategies as shown in Figure 1.2.

In addition to the CAD model reconstructed from existing objects, the framework also includes the possibility to adapt the CAD model represented in parametric surfaces read from Initial Graphics Exchange Specification (IGES) and triangulated surfaces read from STereoLithography (STL). The thesis aims not only the development of suitable RE processes but also the adaptation of CAD models to the requirements of downstream applications so that the process will be complete and time saving.

Curves network method fits a set of an unorganized noisy 3D point cloud to cross-sectional curves network (i.e. transversal and waterline sections). It is found to be an efficient and practically preferable for the ship hull form reconstruction because it turns the RE process into traditional hull form design procedures. The triangulated surface RE method is the most employed and used method so far. With many solutions and techniques in place, the framework incorporates representative methods and integrates with knowledge in order to further improve the efficiency and quality of the output. The direct fitting method offers the possibility to fit the point sets to the surfaces directly.

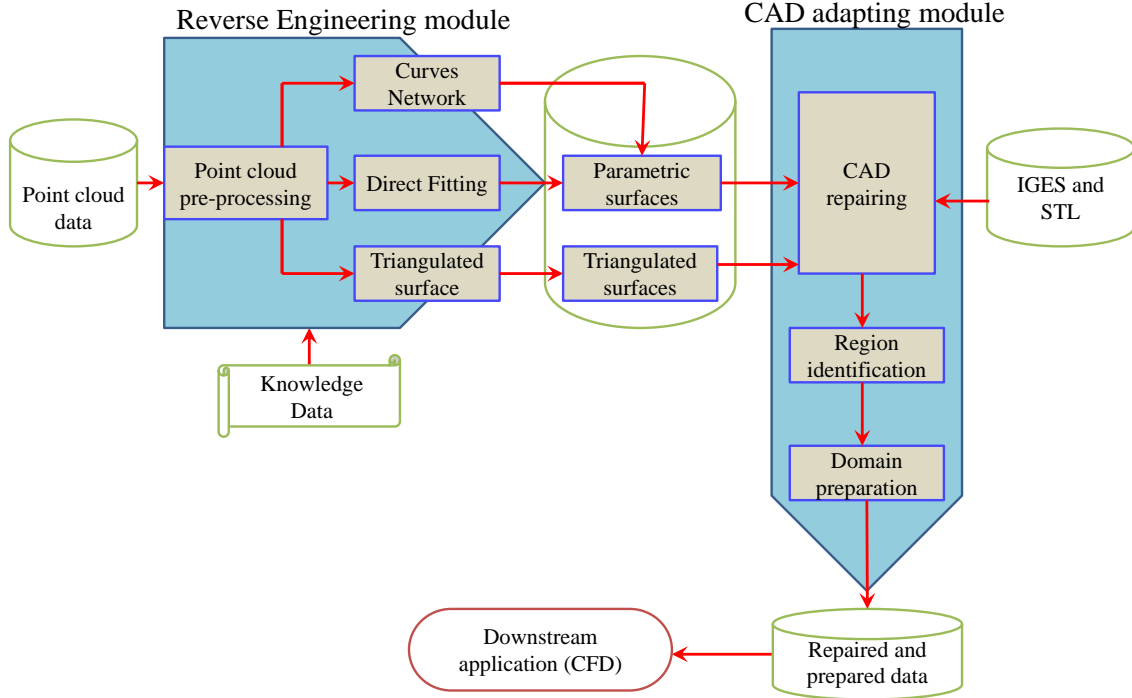


Figure 1.2: The main components and workflow of the developed framework

The CAD adaptation module receives an input from the three RE processes or from any FE or documentation. The repair algorithms clean up commonly found geometrical and topological inconsistencies. After the errors are healed, the regions are identified to further facilitate the model for the mesh generation and then numerical simulation.

1.3 Contributions

The thesis aims to develop a time saving and efficient hull form RE and geometry integration into numerical simulation. The principal contributions are:

- It presents a comprehensive review of the state-of-the-art techniques in RE, knowledge based RE, CAD repairing, region identification, and domain preparation.
- It proposes a ship hull form specific RE techniques, CAD repairing, region identification and domain preparation framework. The RE techniques include curves network, triangulated surface and direct surface fitting.
- In the developed framework, it introduces several structured and integrated numerical processing algorithms to perform the RE tasks such as point cloud pre-processing, segmentation, normal estimation, geometric fitting, etc.
- It incorporates prior knowledge in different parts of the RE algorithms and processes of the developed framework to improve the quality of the results.
- It proposes an automatic approach which detects and repairs commonly found geometrical and topological inconsistencies in CAD data read from IGES and STL file formats. The reconstructed surfaces from RE processes can also be adapted to downstream application using this module.
- To further adapt and save the CAD processing time, it implements region identification and domain preparation strategies which can be used directly by mesh generator tools specifically for *snappyHexMesh* and *Numeca HEXPRESS*.

1.4 Thesis Overview

The thesis is organized as follow:

The general background related to RE and its applications in maritime industry, the available measurement techniques, the point cloud properties and pre-processing methods, the reconstruction techniques, the type and sources of CAD inconsistencies and the need for adapting for downstream applications, supporting theories and tools are presented in chapter 2. In addition, this chapter discusses different kind of knowledge that can be introduced to improve the RE processes. It also provides the general background with the help of relevant literatures. The curves network, the direct fitting and the triangulated surface RE techniques are discussed and demonstrated in chapter 3, 4, 5 respectively. Chapter 6 presents automatic CAD repairing techniques and chapter 7 discusses and demonstrates the region identification and domain preparation strategy for a hull form. Chapter 8 is dedicated to integrate and demonstrate the developed framework. The conclusion and future work are presented in chapter 9.

This page intentionally left blank.

CHAPTER 2

BACKGROUND

2.1 Introduction

The growing interest in a 3D CAD model reconstruction has two faces: the surge in scanning technologies and capabilities on one face and the continuous need to devise an efficient RE strategy to meet the economical and environmental constraints on the other face. Despite the existence of several RE techniques, there are still gaps to achieve the respective quality requirements. In this thesis a hull form specific RE, surface repairing and domain preparation framework is presented. The framework incorporates two main parts: RE and CAD model adaptation to downstream application's input requirements. The RE part includes geometric and knowledge based point cloud pre-processing, surface reconstruction and post-processing. The CAD model adaptation includes surface healing, region and domain preparation techniques. This chapter discusses the theoretical background behind the thesis topic and the tools used in the developed framework with the help of the literature review.

2.2 Reverse Engineering Applications in Maritime Industry

In today's strong and competitive economic environments, minimizing production time and cost is very essential. Therefore, industries are paying enormous effort in improving their products and services by using different technologies. RE is one of the method which used to assist industrial product developments. It finds several applications in maritime industry. The most important applications are:

- ship construction quality control,
- redesign,
- retrofitting,
- maintenance and
- historical archive.

2.2.1 Ship Construction Quality Control

The first step in a ship construction is the building of the hull skeleton. Invariably the as-built skeleton is not exactly the same as the plan drawings due to inaccuracies in the production process. Therefore, shipbuilders require hull actual dimensions in order to fit the important components such as the windows, deck, cabins and various other subsystems. Hence partial or full ship hull RE can be performed at intermediate stages of the construction process to make sure that the hull does not deviate much from its as-designed shape. This ensures the proper fitting of the windows, decking, cabin areas, propulsion systems, electrical systems, as well as design features on the bow and the stern. With accurate scan data, a CAD model can be reconstructed

to monitor whether the ship's parts will fit correctly at the first trial. This helps to save the associated time and cost. The 3D laser scanning could reduce the cost by 37% and time by 39% of ship check data capture and post-processing compared to traditional methods using tape measures, plumb bobs and 2D sketches [1]. Figure 2.1 shows the scanned aft part of the ship at the construction site for ship construction quality control purpose. The report presented in [1], further explains how the benefits of 3D laser scanning for ship production quality control can be evaluated, and called it ship checks. The benefits include:

- the creation of an as-built 3D models and validation of as-built models to design models;
- the reduction of costly design changes and improved design capability;
- the reduced construction rework;
- the accurate and less costly factory fabrication in steady of field fabrication;
- the reduced ship check costs - fewer days, fewer personnel;
- the elimination of return visits to the site to obtain measurements that are difficult or unsafe for human reach;
- the generation of as-built scan data (i.e. point clouds and associated 3D models) for documentation.

The project [1] has been conducted more than five years ago and now, with the scanning technology development one can easily anticipate even more gain with this method. The above list of benefits are applicable to the following applications of RE in maritime industry with regards to cost and time saving.

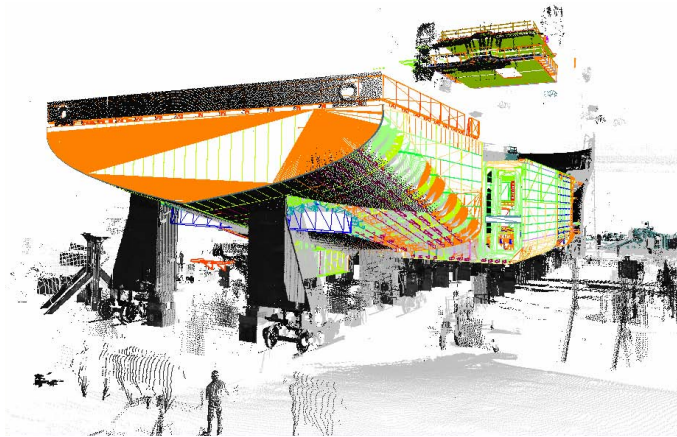


Figure 2.1: The scanned aft part of the ship at a construction site for ship check purpose. Figure adopted from [1]

2.2.2 Redesign

The majority of marine vessels that are in service today do not have digital documentation or CAD models. When the redesign, test, or update of the ship hull is needed, it is difficult to measure and model with traditional measurement methods because it involves complex free-form geometrical shapes. In this case, scanned data allows the naval architects to create precise 3D models that can be used to ensure the quality of interior construction, as well as for design, simulation and inspection purposes.

In some cases the condition and material of the original ship makes the conventional repairing method uneconomical. Therefore, re-engineering of the corroded, flat construction elements in bulk carriers and oil tankers is needed [2]. Figure 2.2 illustrates the redesign process of hull form from very old wooden ships.

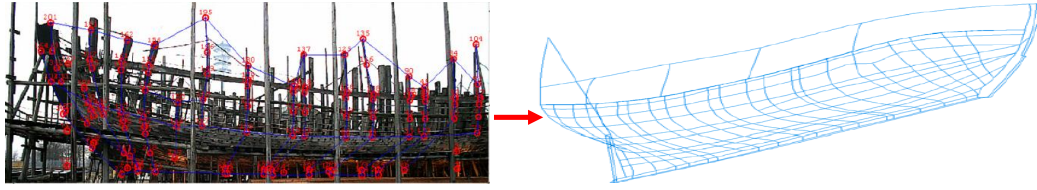


Figure 2.2: Redesigning using RE techniques. The landmarked points (left) and the reconstructed surfaces (right). Picture adopted from [2].

2.2.3 Retrofitting

Retrofitting is the installation of innovative components or systems, and could in principle be driven by the need to meet new regulatory standards or by the ship owner's interest to upgrade to higher operational standards [3]. Many old ships do not fulfill the modern economical and environmental standards. By retrofitting these ships with new technologies, the efficiency and the environmental footprint could be effectively improved. There are potentials to fetch economical and environmental benefits out of ships bulbous bow, rudder, propeller, etc. retrofitting. Figure 2.3 shows the improvement of a bulk carrier propulsion system by retrofitting the wake equalizing duct with fins. At the same time, a ship's downtime for whatever reason should be as short as possible from a profitability point of view. Thus, ships should not stay in shipyard for retrofitting for longer than absolutely necessary. As a consequence, shipyards who engaged in retrofitting require powerful customized methodologies and tools to remain competitive [3]. However, due to an inherent inaccuracy of the ship building process and the tolerances of the engineering techniques involved, there is always a mismatch between design data and reality. The actual as-built shape of the vessel does not accurately reflect the original design data. Therefore, RE techniques should be used to ensure the accuracy, efficiency and safety of the vessels retrofit process.

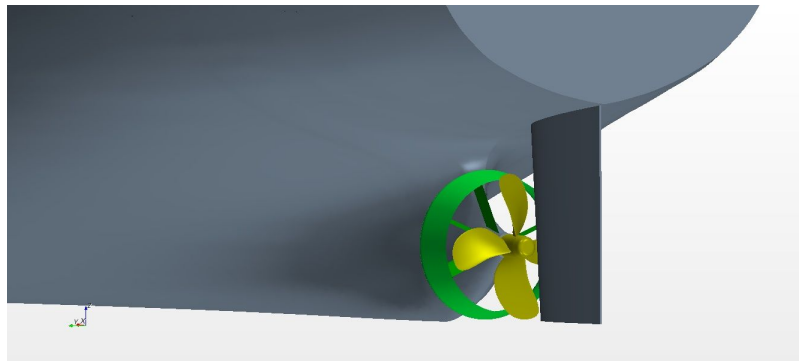


Figure 2.3: Retrofitting using RE techniques. Retrofitting wake equalizing duct with fins to reduce the fuel consumption of the bulk carrier by increasing the efficiency of the propellers. Picture adopted from [4].

2.2.4 Maintenance

A ship's lifespan may reach 20-40 years. The total lifetime costs of a ship is just as important as the initial price. For commercial ship owners, having a ship out of service for a few days may result in high revenue loss, a tremendous impact on profits. Specially in case of collision damage, the part should be quickly maintained to decrease the loss. This needs efficient and fast RE techniques. Figure 2.4 shows how damaged vessels hull could be reconstructed.

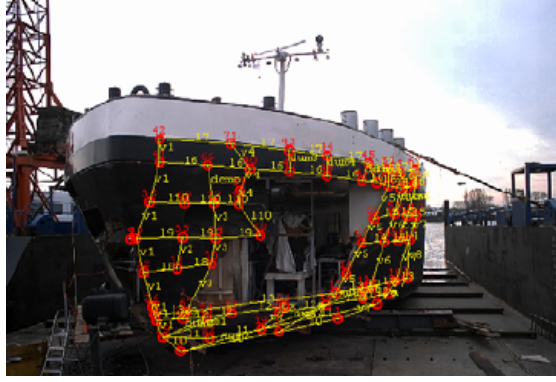


Figure 2.4: Collision damage repairing using RE techniques. The picture adopted from [2].

2.2.5 Historical Archive

This emphasize on the role of RE application for preservation, communication, restoration and replication of historic ships. The historical ships preservation composes of full-scale historic ships and boats, as well as of models and replicas of famous vessels, of objects and exhibits relating to ships, navigation and travels collected in dedicated museums [5]. The preservation of 3D virtual models of the historic ships have significant usage in creating databases and archive in a valuable means for documentation, education and preservation purposes.

2.3 Point Cloud Acquisition

The accuracy of a RE output is greatly dependent on the quality of input data. The quality of input data (i.e. point cloud) strongly rely on the measurement methods and environments. Therefore, selecting suitable measuring technologies and methods is crucial in getting the required level of accuracy out of the RE process. The possible representative measuring technologies and variables used to select suitable measuring methods and technologies are roughly discussed in the following subsection.

2.3.1 Point Cloud Acquisition Methods

There are large varieties of measurement methods used to retrieve the point data from shapes. However, the measurement methods specific to large-scale objects such as ships are a major focus. Koelman [2] categorized into four: conventional manual measurement, mechanical device, laser scanning and photogrammetry.

2.3.1.1 Conventional Manual Measurement

This method usually uses a measurement tape or a laser distance meter. It is flexible and does not rely on an expensive equipment or special skills, but is not efficient. Furthermore, the size of an object which can be measured accurately is limited to an order of magnitude of tens of meters.

2.3.1.2 Mechanical Device

It is also known as a contact scanner or a coordinate measuring machine. An example of an apparatus intended to measure yachts, is a 2D triangulation device where a rod is attached to two running wires, and on the basis of the lengths of the two wires by means of triangulation, the 2D coordinates of the end of the rod can be derived. The advantages of this method are its modest costs and the fact that measured coordinates are directly available in electronic form. The disadvantages are its 2D nature, as well as the limited size of objects (which in case of the 2D device can more or less be as large as the rod).

2.3.1.3 Laser Scanning

The principle is very simple, it is mounted on a tripod next to the area to be scanned. The scanner fires a laser and measures the time it takes for the laser to be reflected back, with which it can calculate the distance from the reflected point. With this principle it scans the environment in a high density based on the time delay of a radiated and captured beam. Laser scanners have a vast variety based on their capabilities, functionalities and application. Large scale structures (i.e. ships, offshore platforms, airplanes, and etc.) are digitized using large range 3D scanners. Using the highly accurate 3D scan data generated with long range scanning, engineers and designers can create CAD model of actual object by processing the scan data into universally usable CAD models.

2.3.1.4 Photogrammetry

It has been applied in a real surveying and architecture for many decades, but also finds many application in fields where it is required to determine the spatial shape of an existing object. Photogrammetry belongs to the category of image-based modeling, a group which also belong to some more exotic technologies, such as shape from shading, shape from silhouette and shape from texture. With regards to RE purpose, multiple photos of an object are captured from different directions and angles. These allow to find identical scenes or points of an object on different photos which leads to a solvable system of equations to reconstruct the object [2, 6].

After all laser scanning and photogrammetry are by far the most employed methods compare to others. Koelman [2] presents detail informations (i.e. suitability, sensitivity, accuracy, cost, etc.) regarding the two measurement methods with regard to their applications for a ship building industry.

2.3.2 Selection Variables of Point Cloud Acquisition Methods

The first and early stage of the RE process is to select a suitable measurement method. This highly affects the whole process and essentially the output quality. Thus, the morphology, object size, measuring environment, accuracy, and economy should be carefully studied before starting any action into the RE process.

2.3.2.1 Morphology

For simple objects such as a plane, many measurement methods are suitable to obtain accurate result. However, with complex physical objects not all measurement technologies are applicable.

Therefore the form and structural features such as the mixture of flat and curved regions, surface finish and sharp or gradual transitions between those regions should be well studied. This is an effort to find out a suitable measurement method which is able to capture less accessible object, or surfaces with different surface finish.

2.3.2.2 Object Size

The object size is very important depending on the area that needs to be scanned. In some cases, it is only necessary to scan desired areas. Specific to marine structures, it depends on whether the whole object is needed to be scanned or partially. In most cases large range laser scanners are needed to scan large areas but sometime limited area of the object is needed for retrofitting purpose where short or medium range laser scanners are sufficient.

2.3.2.3 Measurement Environment

The measurement environment is the most important variable which greatly influences the data accuracy, therefore the environment should be known before the selection of the equipment. The environments include for instance whether the scan will be performed inside or outside the ship? If outside, dry or wet? If wet, the purity of the fluid? Dirty and salty? All environmental constraints or effects should be well know prior to data acquisition equipments selection.

2.3.2.4 Accuracy

It is also challenging to obtain the required level of accuracy in relation to object size. Therefore, the required accuracy should be well know, as the accuracy requirements for different downstream applications (e.g. CFD, FEA, manufacturing and etc.) are different.

2.3.2.5 Economy

Conventionally floating objects such as ships and offshore structures digital data measurements are performed with the object beached or docked. With this method, high precision measurements can be achieved but with high cost for shipowners as the unit should be set out of service. Other practical difficulties may arise due to the setting (dry or floating docks) where the survey has to be conducted, often characterized by restricted spaces, disadvantageous environment conditions (e.g., water, wetness, saltiness), etc [6]. Efficient measurement technologies and methods should be considered to minimize the economical loss during the measurement time.

In general, recent researches on RE of marine structures suggested different measurement technologies. Matthias et al. [3] suggested photogrammetry for interior components of the ship and laser scanner for ship hull and other underwater components for retrofit purpose. Menna et al. [6] on the other hand presented the suitability of photogrammetry for underwater and terrestrial applications.

2.3.3 Point Cloud Characteristics

Digitized point clouds are typically susceptible to unwanted or missed points. As a result the processes based on the point cloud (e.g. normal and curvature estimation, triangulation, curve

or surface fitting, etc.) are complicated or even leads to failure. Hence digitized points can be hardly used directly without pre-processing. The point cloud pre-processing includes the identification of unwanted or missed points and treats accordingly so that it is suitable for downstream applications. In this section the type and the sources of point cloud imperfections are discussed.

2.3.3.1 Types of Point Cloud Inconsistencies

Having information about the input point cloud properties has an important impact on the selection of pre-processing, reconstruction and even post-processing algorithms and methods. In literature point cloud inconsistencies are categorized based on their primary sources and based on their effect on reconstruction algorithms. Kanzok et al. [7] categorize based on the sources of the inconsistencies as: inconsistencies in the scanning equipment itself and inconsistencies in the scene of the scanned object. They only consider the sources of inconsistencies that might occur during measurement. However, there might be also inconsistencies arise from point cloud pre-processing algorithms for instance registration which might usually leads to misalignments. Based on their impact on reconstruction algorithms, Berger et al. [8] characterize the point cloud inconsistencies as: varying density, noise, outliers, misalignment, and missing data. In this thesis the classification based on their impact on reconstruction algorithms is found useful, as the RE part of the thesis focuses on pre-processing, surface reconstruction and post-processing.

Outliers: Points that are far from the true surface are classified as outliers. Outliers are commonly introduced due to structural artifacts in the acquisition process. In some instances, outliers are randomly distributed in the volume, where their density is smaller than the density of the points that sample the surface. Outliers can also be more structured, however, where high density clusters of points may exist far from the surface, see Figure 2.5 (a). Unlike noise, outliers are points that should not be used to infer the surface rather should be removed either explicitly through detection [9], or implicitly through robust methods [10].

Noise: Points that are randomly distributed near the surface are traditionally considered to be noise, see Figure 2.5 (b). The specific distribution is commonly a function of scanning artifacts such as sensor noise, depth quantization, and distance or orientation of the surface in relation to the scanner. For some popular scanners, noise is introduced along the line of sight, and can be impacted by surface properties, including scattering characteristics of materials. In the presence of such noise, the typical goal of surface reconstruction algorithms is to produce a surface that passes near the points without over-fitting to the noise. Robust algorithms that impose smoothness on the output [11], as well as methods that employ robust statistics [12], are common ways of handling noise. Here an important question is that, whether the noise should be eliminated before, after, or during the reconstruction stage, because noise filtering might destroy the sharp feature of the model, which might be tolerable in some cases but may also lead to serious problems in case features are important to preserve.

Missing data: A motivating factor behind many reconstruction methods is dealing with missing data. Missing data is due to such factors as limited sensor range, high light absorption, and occlusions in the scanning process where large portions of the shape are not sampled. Although some of these artifacts may be reduced as scanning technology advances with higher precision, denser sampling, and lower noise levels, occlusion remains a persistent problem due to the physical constraints of the device. Missing data differs from a non-uniform sampling, as the sampling density is zero in such regions, see Figure 2.5 (c). Many methods deal with a missing data by assuming that the scanned shape is watertight [11, 13–15]. If the level of the missing data is significant, for instance a single scan, then trying to infer the entire shape can be too ambiguous. Some methods focus on performing reconstruction only on the available information, effectively

preserving the boundaries from the scan [16]. Other approaches make a prior knowledge and assumptions on the missing region, permitting the reconstruction of higher-level information. These can range from inferring a skeleton [17], shape primitives [18], symmetry relationships [19], and canonical regularities [20].

Varying density: The sampled point cloud distribution is usually susceptible to varying density due to the distance from the shape to the scanner position, the scanner orientation, as well as the shape's geometric features and colors. Figure 2.5 (d) illustrates the varying point cloud density due to the object color differences. Many surface reconstruction algorithms should be able to estimate a notion of sampling density at every point, see e.g. [21], and hence the level of non-uniformity in the sampling can have a great impact on estimation accuracy.

Misalignment: The imperfect registration of range scans results in misalignment. Misalignment tends to occur for a registration algorithm when the initial configuration of a set of range scans is far from the optimal alignment, see [22] for a survey on registration techniques. Misalignment is a significant challenge for surface reconstruction, as it introduces structured noise via scans that are slightly offset from the surface; see Figure 2.5 (e).

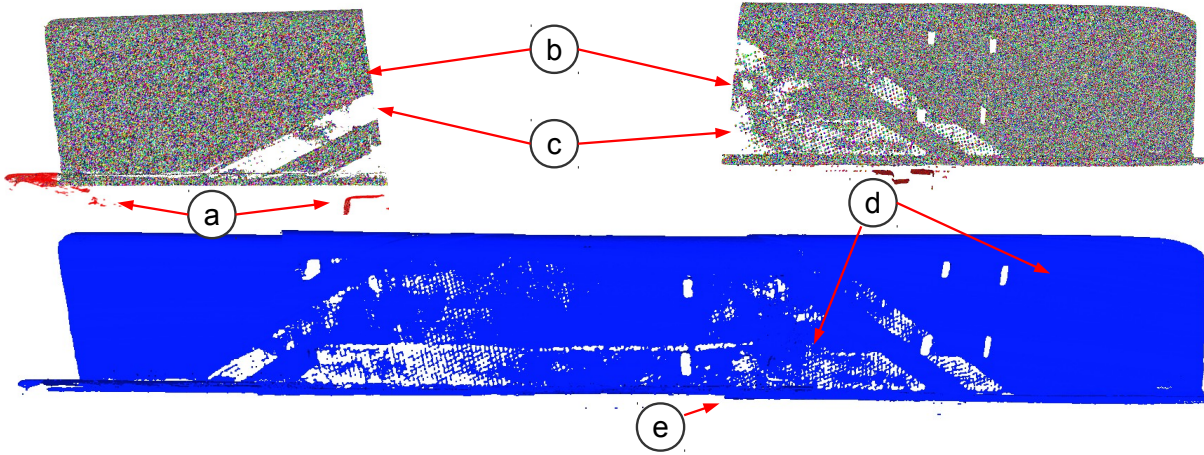


Figure 2.5: Different types of point cloud inconsistencies. a) outliers, b) noise (the color shows the extent of noise), c) missing data, d) varying density, e) misalignment

2.3.3.2 Sources of Point Cloud Inconsistencies

It is essential to know the sources of the point cloud inconsistencies in order to devise the pre-processing methods. The sources of the point cloud inconsistencies are all or either of the following reasons:

Acquisition system: This includes the calibration and accuracy of sampling equipments and their fixture placement. Calibration is an essential part of setting up and operating a position-measuring device. Any sensing unit should be calibrated before the sampling started to accurately determine parameters such as camera points and orientations. In addition the sampling procedures such as distance from measured surface and accuracy of the moving parts of the scanning systems especially for the multi view sampling all contribute to the overall measurement errors.

Physical object: The object might include occlusion, different surface finish and might be complex object configuration or topology which may lead to a problem of accessibility. This can make some data impossible to acquire with certain methods. Through holes are typical examples of inaccessible surfaces. Occlusion is the blocking of the scanning medium due to shadowing or obstruction. It could be due to occlusion on the objects itself or from the fixtures clamped on

or around the object. Multiple views scanning might decrease the extent of this problem but leads to the inconsistencies that might be introduced by registration algorithms. Surface finish which basically implies for smoothness and material coatings can also affect the data acquisition system and resulted in point cloud inconsistencies.

Pre-processing algorithms: This includes the inconsistencies those introduced during pre-processing stage e.g. misalignment, loss in objects feature, etc.

Measurement environment: The measurement environments such as brightness, wet/dry, surrounding physical objects are usually the reason for a missing data, extremely unorganized, noisy, and incomplete point clouds.

Measuring personnel: The personnel involved in scanning activities should be well experienced with measuring set of equipments and should have a prior knowledge about the features and the design procedure of the object to be scanned.

2.4 Geometry Based Reverse Engineering

Over the last several years many mathematical approaches have been proposed to reconstruct objects and structures from 3D datasets. The approaches include completely mathematical based pre-processing (i.e. noise reduction, normal computation, outliers removal, etc.), surface reconstruction (i.e. triangulation, surface fitting, etc.) and post-processing (i.e. smoothing, hole filling, etc.). These approaches have proven successful for relatively simple parts and objects, but have resulted in reconstructions that have 'frozen-in' errors. Typical errors are surfaces at incorrect relative positions or artifacts arising from noisy or missing data [23].

2.4.1 Point Cloud Pre-Processing

Point cloud pre-processing includes all attempts to eliminate point cloud inconsistencies and the preparation of a point cloud for the success of surface reconstruction algorithms. Under this section some frequently used pre-processing algorithms are described.

2.4.1.1 Nearest Neighborhood

Nearest neighborhood determination is one of the algorithms extensively used in a point cloud pre-processing. Mathematically the problem of determining the point's neighbors P^k is closely related to the specific metric space that needs to be used. Let P_q be a query point, and $P^k = P_1^1, ..., P_k^n$ a set of points located in the neighboring vicinity of P_q . The neighbor is given as:

$$\| P_i^k - P_q \|_x \leq d_m \quad (2.1)$$

where d_m is a specified maximum allowed distance from the neighbor to the query point. In order to determine the k (k is a positive integer) closest points of a query point $P_q \in P$, all distances from P_q to all the points in P must be estimated and ordered, with the first smallest k ones corresponding to the closest set of points P^k . This brute-force process however is extremely costly and it makes little sense to use it for applications where closest queries are frequent [24]. Hence there are simplified procedures to determine the neighbor points using different methods. Applying a k-d or k-dimensional tree data structure is a quick and widely used method which organizes the number of points in space. Figure 2.6 illustrates a 2D case. This search can be

done efficiently by using the tree properties to quickly eliminate large portions of the search space [25, 26].

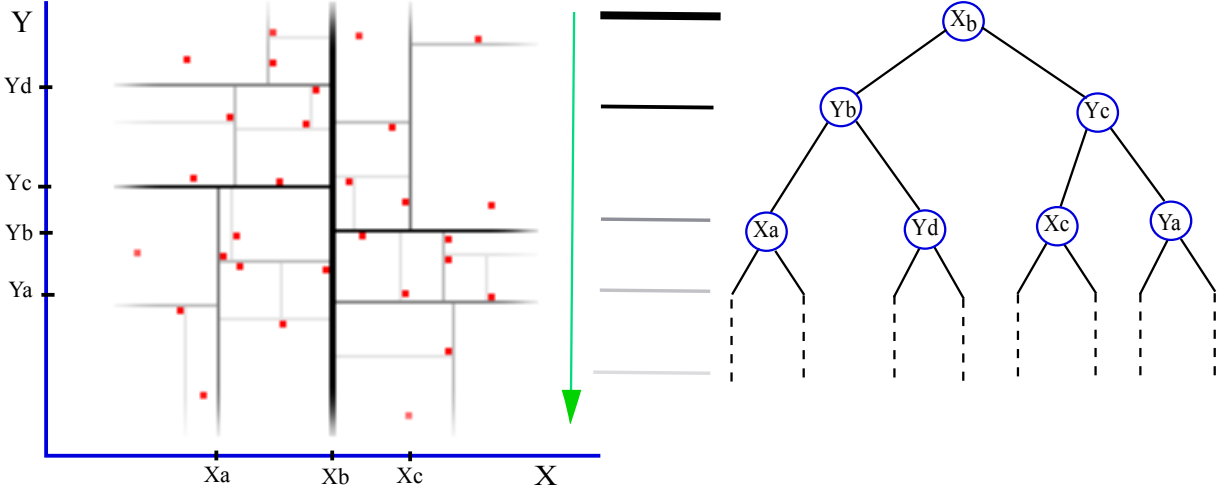


Figure 2.6: Working principle of a kd-tree in 2D space. Red squares are dataset points, black lines are splits. The thinner the line is, the deeper the node is which corresponds to the split.

From the usage point of view, there are two specific application examples for the determination of P^k for a query point P_q , namely:

- determination of closest k neighbors of the query point (k search);
- determination of all k neighbors of the query point up to a radius d_m (radius search).

The specifics of the nearest-neighbor estimation problem raise the question of the right scale factor: given a sampled point cloud dataset P , what are the correct k or r values that should be used in determining the set of nearest neighbors P^k of a point? This issue is of an extreme importance and constitutes a limiting factor in the automatic estimation (i.e. without user given thresholds) of a point feature representation. There are some particular studies regarding this problem presented in [27, 28], in the context of the surface normal estimation, where the authors attempt to automatically estimate the correct scale by iteratively determining the patch density.

2.4.1.2 Convex and Concave Hull

Convex and concave hull algorithms are also frequently used in RE processes. The convex hull is the boundary of the minimal convex set containing a given non-empty finite set of points in the plane (or n -dimensional space). The convex hull does not fully reflect the geometrical characteristics of datasets as a result, concave hull is developed which is a better choice for geometrical evaluation. See Figure 2.7 for the difference between convex and concave hull.

2.4.1.3 Outliers and Noise Filtering

The huge point cloud data obtained by the applied data acquisition system is usually uneven in density, extremely unorganized including noise and outliers. These mostly impede the point cloud surface reconstruction and post-processing steps in a total failure. It is usually originated from inaccuracies in the scanning equipment and in the scene to be scanned. Many strategies

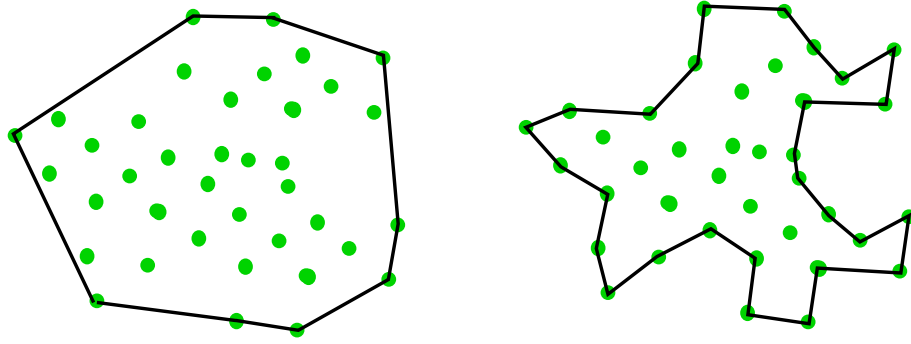


Figure 2.7: Convex hull (left) and concave hull (right)

are developed to detect and remove the outliers and noise in the raw data. Outliers filtering is a mandatory step as it is a serious difficulty compared to noise. Outliers detection in point cloud data is not trivial because the points are usually unorganized, noisy, sparse, have inconsistent point density, no knowledge about the statistical distribution of the points, have geometrical discontinuities, and arbitrary surface shape with sharp features [29]. Over the years several outliers detection and removal approaches are proposed. The approaches include distribution-based [30], depth-based [31], Clustering [32], distance-based [33], and density-based [34].

Distribution-based approaches deploy some standard stochastic distribution model (i.e. normal, Poisson, etc.) and flag as outliers those objects that deviate from the model according to a significant level. The depth-based approach is based on computational geometry and computes different layers of k -dimensional convex hulls [31]. Objects in the outer layer are detected as outliers. However, it is a well-known fact that the algorithms employed cannot cope with large, arbitrary data sets in 3D dimensions [35].

Many clustering algorithms detect outliers as by-products [32]. From the viewpoint of a clustering algorithm, outliers are objects not located in clusters of a dataset. However, since the main objective of a clustering algorithm is to find clusters, they are developed to optimize clustering, and not to optimize outliers detection. These algorithms, in general, consider outliers from a more global perspective, which also has some major drawbacks [34].

The distance-based approach was originally proposed by Krorr et al. [33]. An object in a data set is a distance-based outliers if at least a fraction or percentage of the objects in the object set is farther than given radius from it. This outliers definition is based on a single, global criterion determined by the parameters: the radius and percentage. This can lead to problems when the data set has both dense and sparse regions [34].

The density-based approach was proposed by Breunig et al. [34] for knowledge discovery of databases applications. It relies on the local outliers factor of each object, which depends on the local density of its neighborhood. The algorithm is not only independent of the prior knowledge of the scanned objects, distribution or density of sampled points but also does not suffer from the different local point densities.

These approaches are further developed with different techniques to improve the performance of the detection and removal strategies. Sotoodeh [35] further develops a semi automatic point cloud density based outliers removal strategy. Abdul et al. [29] develop a method which is based on statistical approaches. It fits the best planes to the majority data (inliers) and then discovers the outliers locally for every neighborhood based on the majority results.

Scientific literature suggests distance-based and density-based approaches for large range of data [7] but with prior knowledge about the scene and the measurement technology, there are multiple

approaches available to remove the outliers from the data. In case of little knowledge about the scene, the removal strategy might range from simple bounding domain trimming to statistical and geometrical methods described above. The outliers removal approaches used in this thesis will be described in successive RE strategies.

2.4.1.4 Down-Sampling

Data acquisition with 3D laser scanner is virtually unlimited with respect to resolution. Modern laser scanners can acquire more than a million points per second [36]. This leads to point clouds with large numbers of points. Data reduction becomes a necessary step to efficiently perform operations on these point clouds. Reduced point clouds will notably increase the speed of subsequent processes such as registration, feature estimation, and surface reconstruction. On the other hand, point cloud reduction tends to be computationally more efficient or less memory demanding compares to mesh reduction since mesh data structures needs to be maintained [37]. In addition a huge amount of dense points directly affects the point cloud display and storage [38]. Thus point cloud data simplification is an inevitable step to maintain the smooth point cloud subsequent processing [39] [40].

The traditional point cloud reduction methods include the random sampling method, the bounding box method, the grid-based reduction based on curvature and reduction based on clustering algorithm [41] [42]. In recent years, there are many improvements to preserve the features of the original object based on existing methods [38]. In the developed framework, the voxelized grid approach is primarily used to reduce the number of points. Over each point sets, a 3D voxel grid is generated with a user defined voxel size (resolution values), then in each voxel, all points inside the voxel are approximated with their average. For illustration a 2D point cloud down-sampling using voxel grid is shown in Figure 2.8.

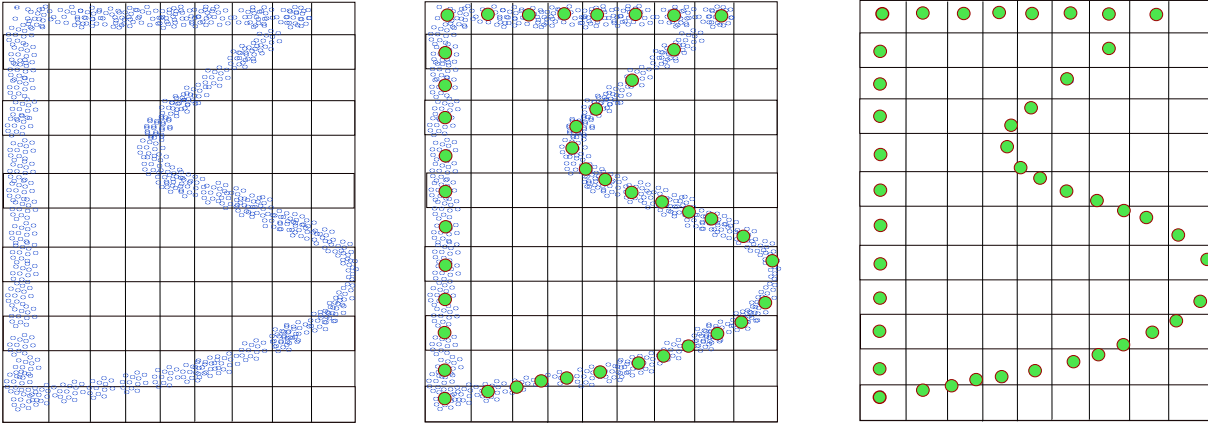


Figure 2.8: Point cloud down-sampling using voxelized grid method, illustration for 2D cases. The point cloud with some noise and many in number (left). Generating a grid using user defined resolution (middle), and find the average point of the points in the same grid and eliminate the other points (right)

2.4.1.5 Registration

Large scale structures, objects and environments RE requires multiple scans from which a complete 3D model is to be assembled together by registration techniques. The registration process requires the selection of initial features or point correspondences for every two acquired scan

pairs. The relative orientation between scan pairs is computed with the aforementioned correspondences. Registration could be roughly categorized into rigid and non-rigid approaches [43]. Rigid approaches assume a rigid environment such that the transformation can be modeled using only 6 Degrees of Freedom (DoF). Non-rigid methods on the other hand are able to cope with articulated objects or soft bodies that change shape over time. There are many registration approaches proposed over the years, simple singular value decomposition [44], principal component analysis based registration [43], iterative closest point algorithm [45]. Many improvements have also been made based on the original iterative closest point algorithm such as non-linear iterative closest point [46], generalized iterative closest point [47], and non-rigid iterative closest point [48].

The choice of one of the algorithms generally depends on several important characteristics such as accuracy, computational complexity, and convergence rate, each of which depends on the application of interest [43]. Moreover, the characteristic of most registration algorithms heavily depend on the data used, and thus on the environment itself [43].

Considering the relative large size of ships, multiple scanning strategy is inevitable. For hull form RE the starboard or the port side is usually enough for hull surface reconstruction, however still requires multiple scanning. Figure 2.9 shows the different patches of scanned point cloud data aligned together using Iterative Closest Point (ICP) registration algorithm.

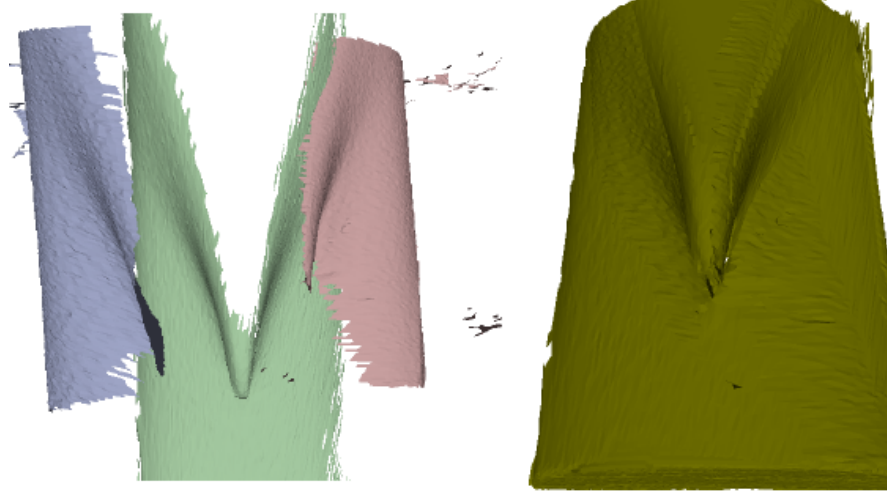


Figure 2.9: Raw point cloud outliers removal and registration. Scanned point cloud data from different directions and locations (left). After removing the outliers, the point cloud patches are aligned together using ICP registration approach.

2.4.1.6 Normal Estimation

Normal vectors are heavily used in CAD data representation, visualization, manipulation and virtual laboratories based on CAD models. For instance, numerous points based rendering [49, 50], surface reconstruction [51, 52], clustering based on normals [51], primitive extraction [53] are highly dependent on the accuracy of points normal.

Point cloud acquired from photogrammetry or lasers typically consist of noises, therefore the objective of point cloud normal estimation is to develop noise insensitive estimation method. Figure 2.10 shows the sensitivity of normal estimation to noise (left) and the robustness of normal estimation regardless of a noisy data [54].

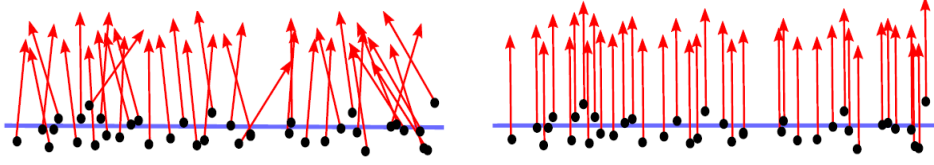


Figure 2.10: Point cloud normals estimation, sensitive to noise (left) and robust to noise (right).
Figure adopted from [54]

The approaches to normal estimation on point clouds can be classified into two: regression based and Delaunay/Voronoi based [55]. The robustness of normal estimation approaches is strongly dependent on how they deal with point cloud outliers, noise and sharp features. Regression based estimation is first proposed by Hoppe et al.[56]. For each query point, a least squares local plane is fitted to its k nearest neighbors by using Principal Component Analysis (PCA). The normal of a point is the eigenvector corresponding to the smallest eigenvalue of the covariance matrix. There are many techniques developed to further improve this method [28, 57–60]. Most of them improve the robustness of the approach towards handling noise, outliers but many fail to properly preserve sharp features. Delaunay/Voronoi based approaches for noise-free point cloud regard the line through a point and the furthest voronoi vertex in a point's Voronoi cell, named pole, as the approximation of the normal for a point [61]. Some upgrades are developed based on these approaches [61–63].

A combination between PCA and Voronoi based methods is developed to increase the robustness of the method towards noise but do not consider sharp features[14].

Recent works presented, techniques which are capable to handle noise, outliers and sharp features while estimating the point cloud normal [54, 64]. Figure 2.11 illustrates the feature sensitive normal estimation algorithm.

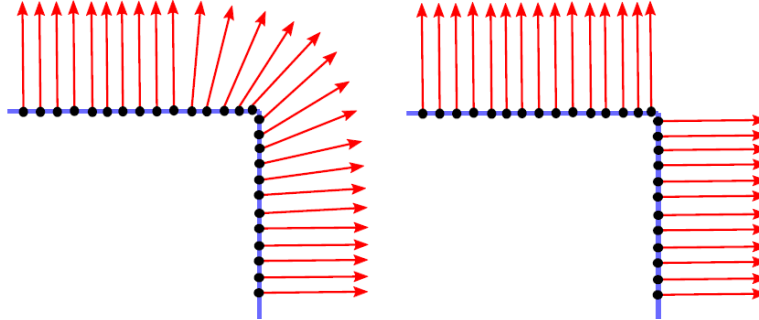


Figure 2.11: Point cloud normals estimation with the Least Square Regression (left) and estimation with sharp feature sensitive computation (right). Figure adopted from [54]

As part of this work the estimation of the surface normal based on neighborhood points least-square fitting to a plane, tangent to the surface is implemented. The method estimates the surface normal at a point from the surrounding neighborhood points. The numbers of neighborhood points are determined by defining the radius of the sphere around the query point. Hence the accuracy of the result depends on the feature of the object and the density of the point cloud as studied in [24].

2.4.1.7 Segmentation

Segmentation is the process of dividing a point cloud or mesh into a 'meaningful' point clusters (segments). The problem of segmentation is not well posed since the desired segmentation result

depends on the application and is related to the definition of the word 'meaningful'.

Segmentation is indispensable in RE process, because the 3D model is usually too complex and too large to reconstruct the surface using only a single surface patch. Segmentation is also required to modify a specific region of a 3D model or to recognize features. In literature there are two main approaches such as edge-based and face-based approaches. Some authors also combine the two methods which they usually call hybrid method [65].

Edge-based methods detect the points on the borders between different surface patches by searching for discontinuities in the normal vectors and/or curvature values. The main challenge of this method is to connect these border points (or mesh edges) to obtain a closed polygonal lines to assure a good segmentation [24]. Moreover, the exact location of these points is difficult to determine in the presence of noise. Most of these algorithms require a threshold to restrict the number of candidate edge points, e.g. an angle threshold between two adjacent facets or a curvature threshold. Many improvements are performed based on this approach [66].

The face-based methods, on the other hand, group similar points according to some geometric property, e.g. normal vector or curvature. An extensively studied approach for such methods is the use of traditional cluster analysis [67, 68]. Another approach used in face-based algorithms is applying region growing algorithms [69]. This segmentation approach is improved by iteration between segmentation classifications, i.e. recognition and fitting of simple algebraic surfaces (plane, sphere, cylinder and cone), an extended review is given in [65]. Some algorithms cannot be classified in any of the two segmentation groups, since they are based on a combination of both approaches. Many similar developments are conducted in this regard [65, 70, 71].

The use of point cloud or mesh segmentation varies from application to application. In the context of the application presented in this thesis, the primitive detection and segmentation in the desired scene is the primary interest. Many engineering construction elements are usually made of different regular primitives such as planes, spheres, cylinders, and etc. These primitives have a significant contribution in generating accurate 3D models from an unorganized noisy point cloud. Once the regular features are clustered the geometrical or prior knowledge can be introduced to ensure accurate results even in case of missing data. Therefore the functionality of primitive segmentation or detection such as plane and cylinder detection is incorporated in the develop framework.

Usually primitives are fitted to point cloud using an iterative method called *Random Sampled Consensus (RANSAC)*. Unlike simple squares method (fits to all points, including the outliers) the RANSAC only fits to the inliers, provided that the probability of choosing only inliers in the selection of points is sufficiently high. For more detail on RANSAC refer to [72].

The segmentation procedure of the primitives in the point cloud includes the following main steps:

- randomly select three non-collinear unique points (p_i, p_j, p_k) from P ;
- compute the plane model coefficients from the three points $(ax + by + cz + d = 0)$;
- compute the distances from all $p \in P$ to the plane model (a, b, c, d) ;
- count the number of points $p^* \in P$ whose distance d_p to the plane model falls between $0 \leq |d_p| \leq |d_t|$, where d_t represents a user specified threshold. The threshold determines how close a point must be to the model in order to be considered as an inlier.

Additional constraints could be imposed to simplify the process, for example horizontal or vertical planes. A similar procedure works for cylinder primitives implemented in the developed framework replacing the mathematical model. The result of the plane and cylinder segmentations incorporated in the developed framework is shown in Figure 2.12.

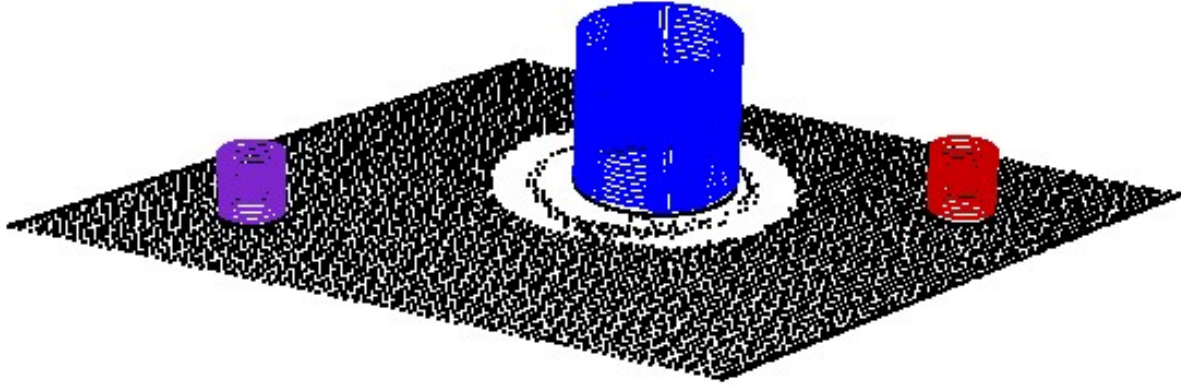


Figure 2.12: Illustration for primitives segmentation such as plane and cylinders from the given point cloud scene.

2.4.2 Surface Reconstruction

Surface reconstruction is the representation of the pre-processed point cloud using different surface representation techniques. Surface reconstruction and representation involve several challenging tasks to be performed, and many objectives to meet. In general the task is ill-posed, i.e., there is no unique solution. Furthermore, it should be able to handle a complicated topology and geometry as well as noise and non-uniformity of the point cloud. With these conditions, it is required to construct a surface with good approximation of the measured data and with desired smoothness. Above all, the reconstructed surface should have a representation that is not only good visually but also compatible with downstream applications and suitable for any other dynamic operation on the surfaces.

2.4.2.1 Surface Representation

In literature there are two main groups of surface representations: explicit and implicit [73–75]. Explicit surfaces prescribe the precise location of a surface while implicit surfaces represent a surface as a particular isocontour of a scalar function [74].

Explicit Surface Representation: Explicit surface representation includes polygonal meshes, parametric surfaces, generalized cylinders, and subdivision surfaces [75]. Polygon meshes are the most popular and can be deformed either by directly pulling their vertices or by using some algorithms that displace vertices by deforming the volume in which the mesh is enclosed. The parametric representation is a powerful method that represent generic surface shapes as a 2D manifolds embedded in 3D, given the set of variable parameters. It consists of Bézier and B-spline curves and surfaces represented by closed-form mathematical expression. The parametric representations of a complex geometry usually resulted in a high number of surface patches e.g. upto 4000 surfaces for a single ship hull form. A new set of methods, which utilize a mesh of polygonal shapes or a sequence of meshes to describe a surface, is now becoming popular. By doing this, a freedom from the closed-form mathematical expression can be achieved, and a wide variety of surface types can be expressed. The surfaces are commonly called subdivision surfaces as they are based upon the binary subdivision of the uniform B-spline curve/surface. In general, they are defined by an initial polygonal mesh, along with a subdivision (or refinement) operation which, given a polygonal mesh, will generate a new mesh that has a greater number of polygonal elements, and is “closer” to some resulting surface. By repetitively applying the subdivision procedure to the initial mesh, it is possible to generate a sequence of meshes that converges to a resulting surface [75]. The other explicit surface representation is generalized cylinder which is a

solid defined by its axis, cross-section curve and scaling function. They are flexible, special class of parametric shapes capable of modeling many real-world objects.

Implicit Surface Representation: An implicit surface also known as a volumetric representation or a function fitting is represented by a closed mathematical form with a single equation. For example the equation for a sphere: $x^2 + y^2 + z^2 - r^2 = 0$ describes infinite number of points (x, y, z) at the distance r , that lie on the common surface. If any other point in space is taken and plugged into the sphere equation, a non-zero value is the result. The returned value indicates whether the point is inside or outside an implicit surface. Implicit surfaces are represented by a particular iso-contour of a scalar function [74]. The methods in this group can be categorized into global or local fitting [76]. In surface reconstruction techniques, the implicit forms are represented using a signed distance function, radial basis function, moving least square and indicator function [77]. As mentioned in [78], implicit surfaces can provide a better topology and can fill up the holes automatically. Based on implicit surfaces, there are many methods used to reconstruct surfaces i.e. least square [79], Poisson surface reconstruction [11, 76], partial differential equation [74, 80], level set method [74], etc.

In general, explicit surface representations are intuitive and easy to manipulate, and they are widely accepted among graphics designers. However, they are not necessarily ideal for fitting surfaces to potentially noisy and incomplete 3D points produced by scanners or 2D points from image contours, because fitting them involves minimizing a non-differentiable distance function. Implicit surfaces are well-suited for simulating physical based processes and for modeling smooth objects [75], because the algebraic distance to an implicit surface is differentiable, in other word, they do not suffer from the drawbacks discussed above when it comes to fitting them to 2 or 3D data. Conclusively, explicit surface representations are well suited for graphics purposes, but less for fitting and automated modeling. The reverse can be stated for implicit surface representations [75].

2.4.2.2 Surface Reconstruction: Discrete and Continuous

Surface reconstruction techniques can be categorized as either discrete or continuous based on reconstruction techniques involved [81].

Discrete Methods: Discrete methods utilize the point set directly, or structures from computational geometry to define the surface. The computational geometry methods such as alpha shapes [82], power crust [83] and ball pivoting [84] methods are categorized under this method. From these partitions of space, a labeling process defines each partition as either interior or exterior to the shape, and the reconstructed surface is defined as the set of faces between interior and exterior regions. The surfaces resulted from these methods typically interpolate most or all of the input points.

Continuous Methods: Continuous methods on the other hand take either a surface fitting approach, where a surface is fitted directly to the samples, or a function fitting approach, where an implicit function is first fitted to the samples and is used to define a surface. It includes the reconstruction techniques such as point set surfaces [85], Hoppe et al. method [86], fast Fourier transform [13], Poisson method [11], wavelet method [87] and fast level set method [74].

2.4.2.3 Surface Reconstruction: Input Data Based Classification

There is also a classification based on the input data set requirements [81]. There could be two categories of surface reconstruction techniques based on the input data requirements excluding

the range image case, point cloud and point cloud with orientation. Alpha shapes, ball pivoting, power crust, point set surfaces and fast level set methods are based on point cloud without normal information, and Hoppe et al., fast Fourier transform, wavelets and Poisson methods are based on point cloud with orientation (normal) information.

2.4.2.4 NURBS Surface Reconstruction Approaches

The NURBS surfaces can be reconstructed from curves network, directly from point cloud and triangular surface. The curves network surface reconstruction approach needs the extraction of cross sectional point clouds on which curves network are fitted. The direct NURBS fitting to point cloud involves the point cloud pre-processing such as noise elimination, outliers removal and more importantly a regularization. The triangulated surface method, converts point cloud to triangulated surfaces on which NURBS surfaces can be built. Here it is important to mention that, the reconstruction of triangulated surfaces can be directly used in downstream applications, because there are many downstream applications which offer the functionality to generate meshes and then virtual laboratory analysis based on triangulated surfaces.

Curves Network Reconstruction: There are many curves generation techniques from point cloud [88] however in this work the generation of free-form curves, i.e., curves of an arbitrary shape, e.g., ship lines is the main interest. Before proceeding to curves generation it is worth to look at the powerful curves representations. The introduction of well known flexible polynomial parametric representations such as Bézier and B-Splines curves and surfaces increases the efficiency, capability and the geometric interpretations of polynomial parametric forms. Bézier curve is based on Bernstein polynomials which has desirable and undesirable properties [89]. The main negative aspect is that a change of one vertex changes the whole Bézier curve and re-computation of all points on the curve is necessary. Therefore, Bézier curve is extended and the Bernstein basis is replaced with Spline basis polynomials. As a result, the problem of global changes has been solved using the linear combination of Splines to form a B-Spline basis, due to its excellent properties, the B-Spline curve has become an essential standard tool of a computational geometry and geometric modeling. However, B-Spline curves are unable to represent simple geometric shapes such as circles, ellipses, conics and any other curves not easily represented by polynomials. Therefore there was a need to encompass both free-form shapes as well as the representation of well-known geometric primitives (circle, ellipse, etc). It was with this in mind, Versprille [90] introduces the extended B-Splines called NURBS in 1975. This method introduces a new variable called weight, which is greatly used to alter the curves locally. The value of a weight could be range from zero (the control point does not influence the curve) and infinity (the curve interpolates the control point). As the weight of a specific control point increases the curve moved towards the control point and vice versa. See Figure 2.13.

The B-Spline curve mathematical representation is usually expressed by an extended NURBS form.

The NURBS curve is mathematically defined by

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_iP_i}{\sum_{i=0}^n N_{i,p}(u)w_i} \quad (2.2)$$

where the w_i are the weights, the P_i are the control points, and the $N_{i,p}(u)$ are the normalized B-Spline basis functions of p degree. The weight w_i determines the influence of the i^{th} control vector P_i on the curve. The i^{th} basis function $N_{i,p}(u)$ is defined on a knot vector $U = (u_0, u_1, \dots, u_{n+p+1})$ and is recursively defined by

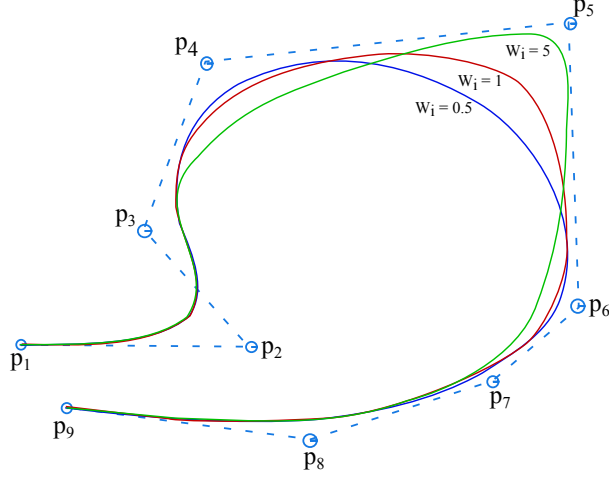


Figure 2.13: NURBS curves with different magnitude of weight values. Weight ($w_i = 1$) is a basic B-Spline curve (red), the decrement of the weight pull the curve away (blue) from the control point (p_5) with weight ($w_i = 0.5$) and as weight increases from ($w_i = 1$) to ($w_i = 5$) the curve (green) moves towards the control point (p_5).

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{if otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (2.3)$$

The knot vector U is non-decreasing numbers, and the half-open interval $[u_i, u_{i+1}]$ the i^{th} knot span. Note that since some u_i 's may be equal, some knot spans may not exist. If a knot u_i appears k times (i.e., $u_i = u_{i+1} = \dots = u_{i+k-1}$), where ($k > 1$), u_i is a multiple knot of multiplicity k , written as $u_i(k)$. Otherwise, if u_i appears only once, it is a simple knot. If the knots are equally spaced (i.e., $u_{i+1} - u_i$ is a constant for $(0 \leq i \leq n+p)$), the knot vector or the knot sequence is called uniform; otherwise, it is non-uniform. The knots can be considered as division points that subdivide the interval $[u_0, u_{n+p+1}]$ into knot spans. All B-Spline basis functions are supposed to have their domain on $[u_0, u_{n+p+1}]$. Usually the knot vector is taken over the closed interval $[0, 1]$.

NURBS can be *open*, *clamped* and *closed*. If the first $p+1$ knots and the last $p+1$ knots are equal to the left end and right end of the domain, the curve is clamped otherwise open. See Figure 2.14.

Direct Reconstruction: Complex objects such as ship hull forms, aircrafts, automobiles, etc. are often represented by easily changeable free form surfaces. It is usually convenient to represent free-form surfaces in terms of surface patches, often given as parametric surfaces such as Bézier and B-Spline surfaces. Parametric surfaces are defined by a set of three functions, one for each coordinate as $f(u, v) = (x(u, v), y(u, v), z(u, v))$, where parameters u and v are in a certain domain. For instance the B-Spline surface is expressed by parametric equation in two-dimensional local coordinates (u, v) with $n+1$ and $m+1$ control points respectively. The B-Spline surface is expressed mathematically as

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{ij} \quad (2.4)$$

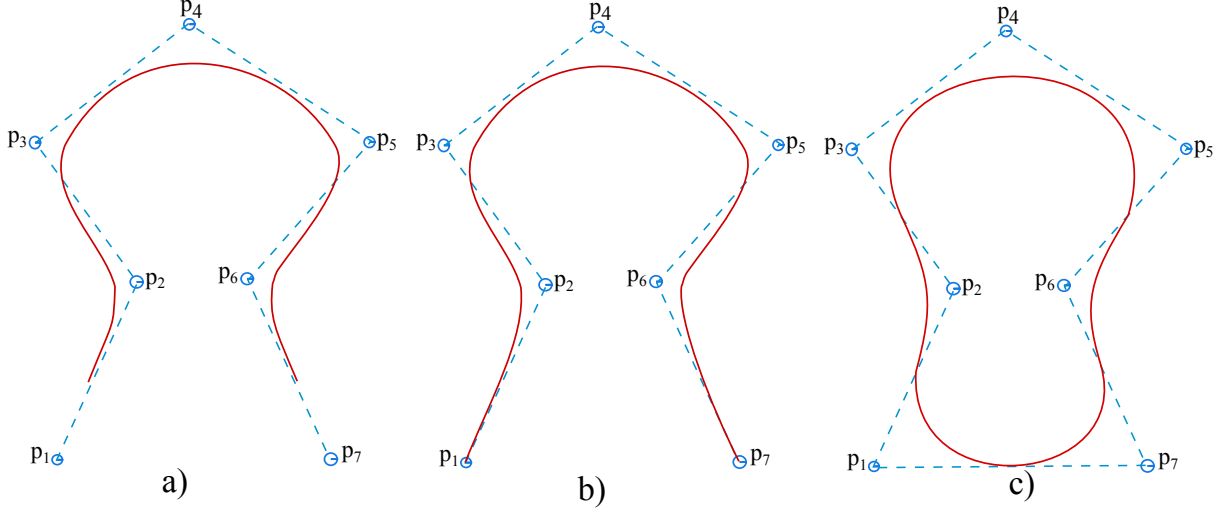


Figure 2.14: Different NURBS curves based on the different definition of knot vectors. a) open curves, b) clamped curves and c) closed curves

where $N_{i,p}(u)$ and $N_{j,q}(v)$ are the B-Spline basis functions, P_{ij} are the control points, with i in the range between 0 to n , and j in the range between 0 to m .

As the NURBS surface is an extension of B-Spline surfaces, it is represented as follow with weight incorporation.

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{ij} P_{ij}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{ij}} \quad (2.5)$$

where $0 \leq u, v \leq 1$ and P_{ij} form a bi-directional control net, the w_{ij} are the weights, and the $N_{i,p}(u)$ and $N_{j,q}(v)$ are non-rational B-Spline basis functions defined on the knot vectors.

$$\begin{aligned} U &= \{\underbrace{0 \dots 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1 \dots 1}_{p+1}\} \\ V &= \{\underbrace{0 \dots 0}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{1 \dots 1}_{q+1}\} \end{aligned} \quad (2.6)$$

where $r = n + p + 1$ and $s = m + q + 1$. Introducing the piecewise rational basis functions,

$$R_{i,j}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{ij}}{\sum_{k=0}^n \sum_{l=0}^m N_{k,p}(u) N_{l,q}(v) w_{kl}} \quad (2.7)$$

then the NURBS surface equation can be written as

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) P_{ij} \quad (2.8)$$

NURBS surface is become a standard in CAD systems due to its stability, flexibility, and local modification properties. A NURBS surface fitting to an unorganized and scattered set of points, and the representation of sharp features like edges, corners, and high curvatures are both challenging [91]. Several approaches have been developed to fit NURBS to unorganized point cloud using different techniques [92–95]. The direct NURBS surface fitting to an unorganized point cloud is studied, see [96, 97]. Figure 2.14 shows typical direct NURBS fitting procedures.

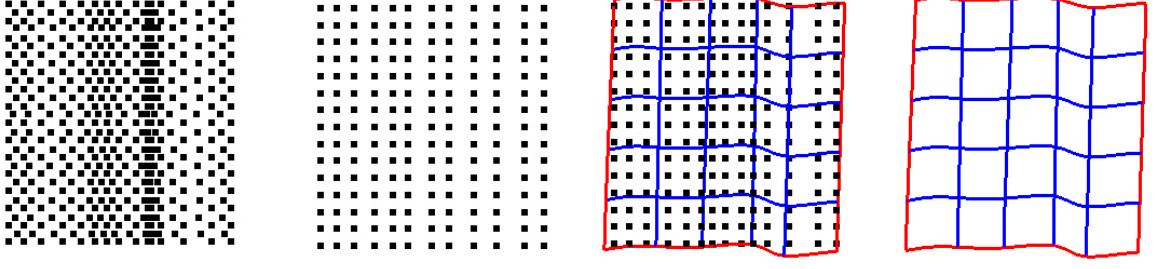


Figure 2.15: Typical direct NURBS surface fitting to point cloud. The process from left to right stands for unorganized point cloud, regularized point cloud, NURBS fitting stage, Fitted NURBS surface as an output respectively.

2.4.2.5 Triangular Surface Reconstruction

Triangular or other polygon surfaces reconstruction is well employed compared to curves network or direct fitting. Different aspects (i.e. pros and cons) of triangular surface reconstruction from an unorganized point cloud are presented in many literatures [11, 82–84, 86, 98–102]. It finds two distinct application directions, a direct usage of triangular meshes for downstream applications and a conversion or fitting to higher order surfaces (i.e. B-Spline and NURBS or other higher order surfaces) based on triangular meshes. Since many downstream applications e.g. mesh generators, offer the functionality to generate meshes based on triangular surfaces, it is possible to use the reconstructed triangular surfaces from a point cloud for this purpose. On the other hand, it eases also the generation of higher order surfaces.

There are explicit and implicit surface representations which lead to triangular surface output. Triangular surfaces could also be obtained by continuous and discrete reconstruction techniques. In this section, some methods are selected for detail discussion. These are: Alpha shapes, greedy projection, grid reconstruction, marching cubes or Hoppes et al., ball pivoting and Poisson methods.

Alpha Shapes: The alpha shapes is a generalization of the concept of convex hull and subgraph of the Delaunay triangulation and used to reconstruct triangulated surfaces from a point cloud. It was first introduced by Edelsbrunner et al. [103] for a plane case and later further developed by Edelsbrunner et al. [104] for 3D. It uses Delaunay triangulation and a user defines the radius (the α value ranges over 0 to ∞) to represent the surface bounded by a point cloud. For instance given a finite point set, a family of shapes can be derived from the Delaunay triangulation of the point set; while the parameter alpha (α) is used to control the level of the surface detail capturing. For $\alpha = \infty$, the α -shape is identical to convex hull of the point set. However, as α decreases, the α -shape shrinks by gradually developing cavities. These cavities may join to form tunnels, and even holes. From this α -shape property one might see that every convex hull is an α -shape but not every α -shape is a convex hull as shown in Figure 2.16.

Edelsbrunner et al. [104] provide an intuitive description to the working principle of α -shape surface reconstruction from point cloud as follow: imagine a huge mass of ice-cream making up the 3D space and containing the points as "hard" chocolate pieces. Using one of these sphere-formed ice-cream spoons and then carve out all parts of the ice-cream block that can be reached without bumping into chocolate pieces, thereby even carving out holes in the inside (e.g. parts not reachable by simply moving the spoon from the outside). Then it will be eventually ended up with a (not necessarily convex) object bounded by caps, arcs and points. If the all round faces are straighten to triangles and line segments, then an intuitive description of what is called the α -shape is obtained. Here is an example for this process in 2D (where the ice-cream spoon is simply a circle), in Figure 2.16.

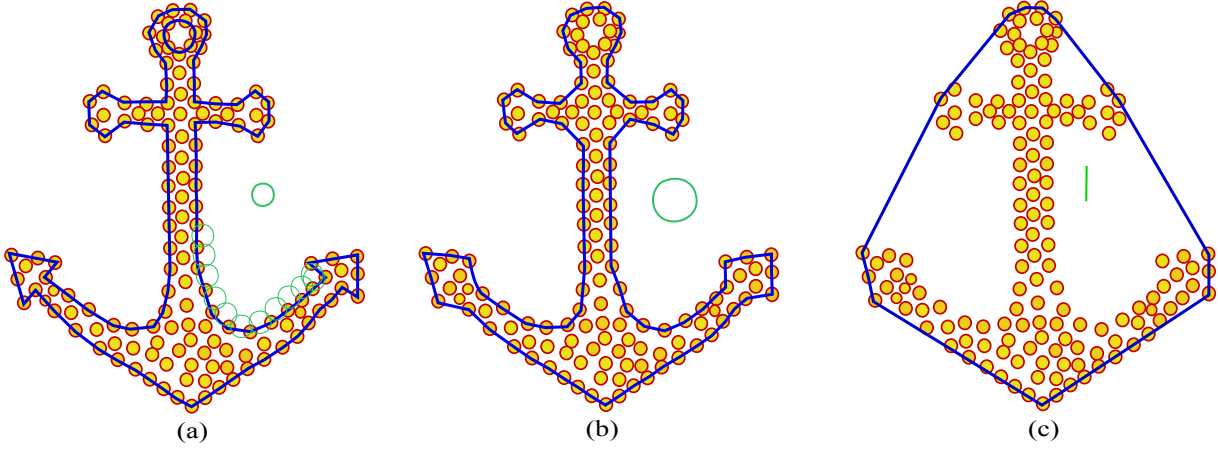


Figure 2.16: Alpha shapes working principle in 2D. An anchor reconstruction using alpha shapes in 2D, a) reconstructed with reasonable α value, b) reconstructed with larger value of circle radius which leads to loss of some details (e.g. anchor crown hole and other details loss), c) reconstructed with α value ∞ which is equivalent to convex hull.

The selection of an appropriate α value is challenging and the random selection of the α value resulted into different results which is called the family of α -shapes of the point cloud. Therefore, the complexity of α -shape is $O(n^2)$, where n is the number of sample points. The α value of the α -shape algorithm is usually defined randomly however there are some attempts to automatize the setting, for instance, Xu and Harada [105] propose the selection of the α value based on the density of sample points. However, this still fails to provide the exact α value that can be applied to all kind of sample points. Moreover Cazals et al. [106] propose a new approach based on α -shape where the α value is chosen locally in steady of defining as a single global parameter.

Greedy Projection: The method reconstructs triangular surfaces from an unorganized 3D point cloud. Marton et al. [107] further develop the projection-based reconstruction algorithm presented in [108] to achieve adaptability to a variable point densities, and noises. The triangulation is performed locally and starts by selecting a starting triangle's vertices and connects new triangles to it until either all points are considered or no more valid triangles can be connected to the resulting mesh. In a second case a new seed triangle is placed in the unconnected part and the triangulation is restarted. Having read an unorganized point cloud, the algorithm comprises three main steps:

- for each reference point R as shown in Figure 2.17, a k -neighborhood is selected by searching for the point's nearest k -neighbors in a sphere (S_R) with radius $r_R = \mu d_0$ that adapts to the local point density (d_0 is the distance from R to its closest neighbor and μ is a user-specified constant);
- the neighborhood is projected on a plane that is approximately tangential to the surface formed by the neighborhood and ordered around R ;
- then the points are pruned by visibility and connected to R and to consecutive points by edges, forming triangles that have a maximum angle criterion and an optional minimum angle criterion.

At the first step, the points in the cloud are assigned with various states depending on their interaction with the algorithm: *free*, *fringe*, *boundary*, and *completed*. Initially, all points in the cloud are in the free state and free points are defined as those points which have no incident triangles. The completed points have all their incident triangles determined. Points that lie along the current surface boundary are either fringe or boundary points. When a point has been chosen as a reference point but has some missing triangles due to the maximum allowable angle

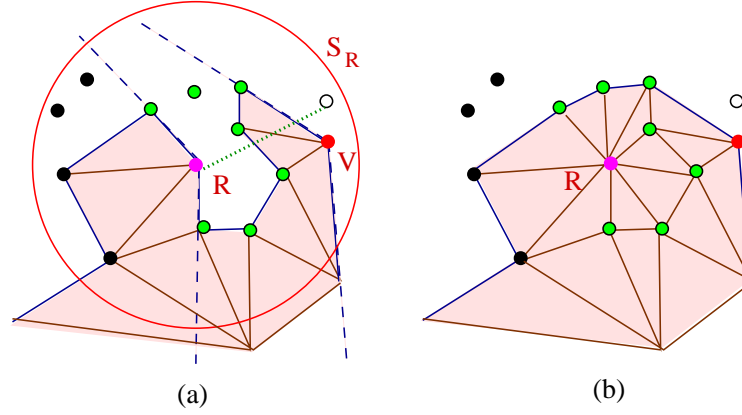


Figure 2.17: Greedy projection reconstruction principle. a) Visibility test around R . The black points are behind R 's boundary edges, the white points are occluded by other edges, and the point V is eliminated as R is behind its boundary edges. b) Completed mesh at R . Adopted from [108].

parameter, it is referred to as a boundary point. Fringe points are points that have not yet been chosen as a reference point.

At the last step, the points in the cloud are pruned depending on the following criterion:

- *Pruning by distance criterion:* a distance criterion is applied to prune down the search for candidate adjacent points in the spatial proximity of current reference point (R) using kd-tree algorithm. Further points which lie outside the sphere of influence centered at reference point are rejected. The chosen points are referred to as the candidate points.
- *Choice of projection plane:* the candidate set of points obtained after applying the distance criterion are projected on the approximate tangent plane.
- *Angle ordering:* a new local coordinate system is defined with the reference point as the origin and project plane. All points in the candidate set are projected to this plane, and order the projected candidate points around the reference point (R) based on the angle between the local coordinate system and the vector from origin to the projected candidate point.
- *Visibility:* the points which potentially form a self-intersecting mesh are discarded. The algorithm defines two edge types for checking this condition, boundary edges and internal edges. Boundary edges are the edges with only one triangle incident on it, they connect fringe and/or boundary points. Internal edges connect the completed points with any other points. The plane is projected using the reference point, candidate set of points and the boundary edges. In case, the line of sight from the reference point to a candidate vertex is obstructed by an edge, the point is occluded.

The algorithm implemented by [107] performs a triangulation locally by projecting the local neighborhood of a point along the point's normal, and connecting unconnected points and regulated by the following parameters:

- maximum number of nearest neighbors which controls the size of the neighborhood points;
- maximum acceptable distance for a point to be considered relative to the distance of nearest point (in order to adjust to changing densities);
- maximum possible triangles edge length which defines the biggest triangles that should be possible;
- the maximum and minimum triangles angles;

- the maximum surface angle and normal consistency.

The maximum surface angle and normal consistency are used to deal with sharp edges or corners where two sides of a surface run very close to each other. To achieve this, points are not connected to the current point if their normals deviate more than the specified angle (note that most surface normal estimation methods produce smooth transitions between normal angles even at sharp edges). This angle is computed as the angle between the lines defined by the normals (disregarding the normal's direction) if the normal-consistency-flag is not set, as not all normal estimation methods can guarantee consistently oriented normals. The method generally works well for locally smooth surfaces and for smooth transitions between areas with different point densities. It has also advantage of saving time and computer memory as edges are written directly and are never deleted.

Grid Triangulation: The general idea of a grid triangulation method is an extraction of Extremal Surfaces (ES) using a pair of scalar (point coordinate) and unoriented vectors (normals) functions, as well as a spatial grid over the domain. It is a polygonal approximation of the underlying surface presented in [109].

Having the scalar and unoriented vector functions and spatial grid, the polygonal approximation of the ES consists of the following procedures:

- identifying critical edges;
- identifying extremal and non-extremal edges;
- polygonization.

For simplification, the steps are illustrated in a 2D for input scalar in a gray color (larger values are lighter) and vector (short lines) function in (a) and the quadtree grid in (b) as shown in Figure 2.18. Then, the grid edges that are crossed by a critical surface (CS) are identified and named as critical edges as highlighted in a red and blue in (b). Critical surface is made up of all critical points of the scalar field (where the derivative is zero) restricted to the line as a function of the scalar and vector. The subset of critical edges intersected by the ES, called extremal edges, are the ones colored red. Next the polygonal critical (extremal) surfaces that crosses the identified critical (extremal) edges (thick curves in (c)) will be created.

Furthermore, assuming S_e is a smooth function with continuous second derivatives, the ES is bounded on the CS by inflection points of s_e restricted to lines $l(x, \mathbf{n}(x))$ which a function of scalar and vector. These inflection points generally form a manifold at a lower dimension than ES (e.g., curves in 3D), as they are the intersection of the CS and the zero-set surface of the second derivative of S_e restricted to normal (\mathbf{n}) .

Let $\nabla S_e(x)$ be the gradient vector of S_e . Then the CS is the zero-set of the following scalar function:

$$g(x) = \overrightarrow{\mathbf{n}(x)} \cdot \nabla S_e(x) \quad (2.9)$$

Identifying critical edges: The critical points of the S_e restricted to \mathbf{n} are the loci where $g(x)$ (see equation 2.9) evaluates to zero, assuming the vector function (\mathbf{n}) can be continuously oriented. If the continuous-orientation assumption holds, a sufficient condition for a grid edge to be critical is that the value of $g(x)$ at the grid ends have different signs, meaning that $g(x)$ equal to zero somewhere along the line. In general, an edge with end points x_1, x_2 is critical if $g(x_1)g(x_2) < 0$, where g is defined in Equation 2.9 and $\overrightarrow{\mathbf{n}}$ is the orientation of \mathbf{n} along the edge.

Identifying extremal edges: Extremal edges are identified as those critical edges whose critical points are local minimum of S_e restricted to \mathbf{n} . The location of the critical points are computed

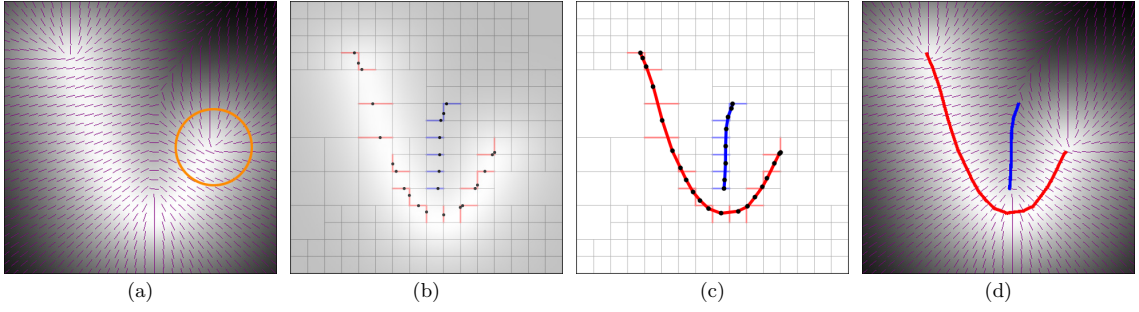


Figure 2.18: Grid triangulation method working principle in 2D. Given the scalar (gray level in (a)), an unoriented vector function (short lines in (a)) and a spatial grid (the quadree in (b)), the method first identifies grid edges intersected by all critical curves (blue and red edges in (b)), a subset of which are intersected by the extremal curves (red edges). Next, the algorithm constructs a polyline that crosses those grid edges (c). The final curves in the original vector field is shown in (d). Adopted from [109]

by a linear interpolation along the critical edge x_1, x_2 using the magnitude of $g(x_1), g(x_2)$. A critical point x is considered a local minimum if the second derivative of S_e along $l(x, \mathbf{n}(x))$ is positive. Figure 2.18 (b) shows an example of extremal edges (red) and non-extremal critical edges (blue) identified using the method.

Polygonization: After identifying the extremal edges, the polygonal surface crossing the extremal edges are generated while approximating the locations of the critical points along these edges. The non-extremal part of the critical surface are built using non-extremal critical edges. The polygonization is achieved by adapting an iso-surfacing algorithm called Dual Contouring [110]. Dual contouring places vertices within grid cells that exhibit a sign change, and constructs polygons that cross the grid edges with a sign change. To extract extremal surfaces, a vertex within any grid cell that contains an extremal edge is created, and a polygon for each extremal edge is created by connecting the vertices within the cells sharing that edge. To determine the location of the vertex within a grid cell, the centroid of the critical points are computed previously on all of the cell's extremal edges, then project this point onto the ES using the projection strategy. Detail information regarding this method is presented in [109].

Marching Cubes Method: Marching cubes also known as Hoppe et al. method is one of the first function fitting approaches to reconstruct a surface from an unorganized point set (i.e. a set of three-dimensional points with no topological information). It approximates an underlying surface from an unorganized set of points. Neither the topology, the presence of boundaries, nor the geometry of the surface are supposed to be known in advance [86]. The method consists of the following major steps to meet the required result:

- determine the neighbor points for each point subscribed in the sphere centered at each point and defined radius;
- compute the average point of each neighborhood and estimation of the corresponding tangent plane for each input point;
- establishment of a consistent tangent plane orientation;
- determination of a signed distance function on a voxel grid;
- extract an iso-surface using marching cubes method.

The estimation of a tangent plane $T_p(x_i)$ for each point x_i alone includes the determination of nearest neighborhood of each point lying in the sphere of the radius R with the center in this point. These points are denoted by $Nbhd(x_i)$ and the next task is to compute the average point O_i with a normal \mathbf{n}_i of the local neighborhood points and a corresponding plane representing

the linearized local surface at this point. Then, the signed distance function of any arbitrary point $p \in \mathbb{R}^3$ to $T_p(x_i)$ is defined to be $dist_i(p) = (p - O_i) \cdot \mathbf{n}_i$. The center and unit normal are computed so that the plane $dist_i(p) = 0$ is the least squares best fitting plane to $Nbhd(x_i)$.

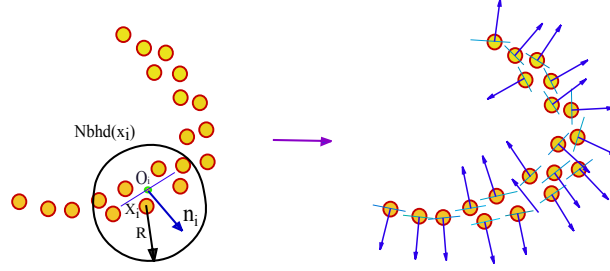


Figure 2.19: Estimation of a tangent plane from nearest neighborhood points for each points in the cloud.

Having the tangent plane and its orientation determined as shown in Figure 2.19, it is important to make the orientation consistent. To assign orientation to an initial plane, the unit normal of the plane whose center has the largest z coordinate is forced to point towards the $+z$ axis [86]. Rooting at this initial orientation, the rest planes will be visited and reoriented. For instance, if $T_p(x_i) = (O_i, \mathbf{n}_i)$ and $T_p(x_j) = (O_j, \mathbf{n}_j)$ are nearly parallel, i.e., $\mathbf{n}_i \cdot \mathbf{n}_j$ approximately positive or negative one ($+/-1$). If the planes are consistently oriented, then $\mathbf{n}_i \cdot \mathbf{n}_j$ is positive one; otherwise, either \mathbf{n}_i or \mathbf{n}_j should be flipped.

After determining a tangent plane at O_i with \mathbf{n}_i as shown in Figure 2.19 (left), which is a local linear approximation of an unknown surface S , then, the signed distance function $f(p)$ to S to be the signed distance between p and its projection onto $T_p(x_i)$ as shown in Figure 2.20, that is, $f(p) = dist_i(p) = (p - O_i) \cdot \mathbf{n}_i$

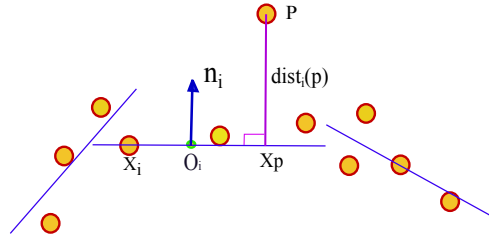


Figure 2.20: Marching cubes method, signed distance function between arbitrary point p and its projection on the plane.

The next task of the method is to contour tracing, the extraction of an iso-surface from a scalar function, which is marching cubes algorithm in this case. To use marching cubes, the distance between the edges of the cubes and the planes must be computed. To do so, the distance between cubes vertices and the center point O_i should be computed, from which the minimum distance will be considered and then finds the corresponding distance from arbitrary point p to the plane, if the projected point (X_p) lies in the neighborhood $Nbhd(O_i)$, then this distance is acceptable, otherwise not and will be set to infinity.

At this position, the main idea is to find the position where the object surface cut the cubes. Dividing into two parts (say hot and cold) depending on the plane surface orientation as shown in Figure 2.21. Finally the weighted “hot” and “cold” vertices’s are used to triangulate the planes in the right order. Many improvements are proposed on iso-surface extraction of the marching cubes method, the work presented in [111] offers the local refinements of the cubes in order to connect the surfaces in neighboring cubes consistently.

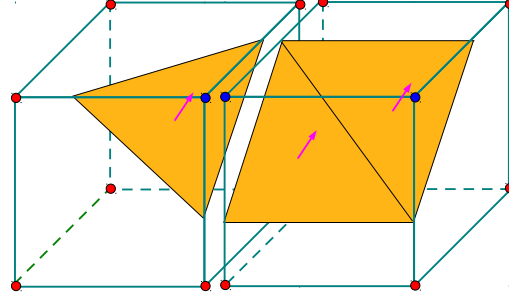


Figure 2.21: Marching cubes method iso-surface extraction based on cubes and oriented planes. The blue vertices are “hot” and red vertices are “cold”. The “cold” and “hot” vertices lies on different side of the vertices.

Ball Pivoting Method: The ball pivoting algorithm is conceptually simple surface reconstruction from a point cloud in which three points form a triangle if a sphere of a user defined radius (ρ) touches them without containing any other point. It was first introduced by Bernardini et al. [84] and provides a way to build triangulated surfaces by interpolating most of the points. Having read the point cloud, the algorithm starts with a seed of triangle which pivot the sphere of a defined radius (ρ) in 3D or a radius of the circle (ρ) in 2D. Then, the sphere pivots around an edge and revolves around an edge until the sphere is in touch with new point, and form a new triangle. The process continues until all reachable points are visited. Two questions might rise here, how to decide the interior and exterior region, how to define the proper radius so that, the missing data and the features can be preserved. In case of a big holes larger than ρ , in the input data, it is challenging to distinguish an interior and an exterior region. Bernardini et al.[84] improve this problem by determining the surface normals and check whether the seed triangle’s normals at the three vertices are consistently oriented or not.

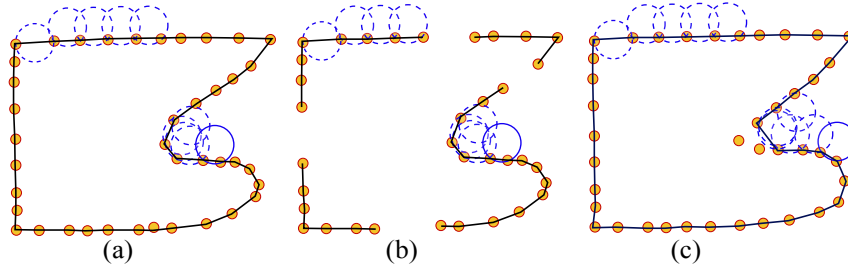


Figure 2.22: Ball pivoting illustration in 2D. a) a circle of radius ρ pivoted from sample point to sample point, connecting them with edges. b) when the variation of density of the point cloud is too much, in some areas, the distance between the two points is greater than the diameter of the circle, which leads to holes. c) shows again the shortcoming of ball pivoting method when it comes to high curvature areas, some points are missed, which leads to loss of object features.

The case of a missing data or high density variation and a high curvature features are the most challenging for this method. As shown in Figure 2.22 for 2D case, the method fails to generate watertight surface if the density of the point cloud is varying to much, or if the distance between two points are greater than the diameter of the sphere as shown Figure 2.22 (b). This problem might be solved by increasing the radius of the sphere, but a large sphere radius might leads to loss of a high curvature areas as shown in Figure 2.22 (c).

Laser scanner high density sample capturing capability, the knowledge of a sampling density might help to choose an appropriate sphere radius. However, the presence of noise, misalignment and complexity of the object challenge the quality of the method’s output.

Further investigations are performed on this method to make it better, Digne et al. [112] attempt to increase the method's feature preservation capability. Digne [113] also presents a method which allow a faster triangulation of an oriented point cloud.

Poisson Surface Reconstruction: Poisson surface reconstruction method expresses the surfaces reconstruction as the solution to a Poisson equation [11]. It is a well know approach which used to reconstruct a watertight surfaces from an oriented point samples. The method is resilient to noisy point cloud and misalignments of point cloud patches. In general, the method first computes a 3D indicator function (defined as 1 at points inside the model, and as 0 at points outside) as shown in Figure 2.23 (c), and then obtain the reconstructed surface by extracting an appropriate iso-surface [11]. The key finding of the approach is that, there is an integral relationship between an oriented point cloud from the surface of a model and the indicator function of the model. More specifically, the gradient of the indicator function is a vector field that is zero almost everywhere (since the indicator function is constant almost everywhere) as shown in Figure 2.23 (b), except at the points near the surface, where points in the direction of the inward surface normal. Thus, the oriented points can be viewed as samples of the gradient of the model's indicator function as illustrated in Figure 2.23 for 2D case.

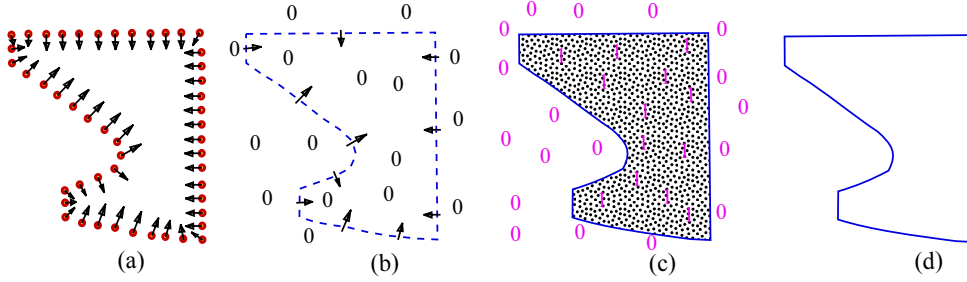


Figure 2.23: Illustration of Poisson surface reconstruction from an oriented 2D point cloud. a) oriented points, b) indicator gradient, c) indicator function and d) a surface.

Suppose an input to the Poisson surface reconstruction is an oriented point set that consists of a sample points with a position and an inward-facing normal, assumed to lie on or near the surface ∂M of an unknown surface model M (see Figure 2.23 (d)). Having consistently oriented point's normal, see Figure 2.23 (a) which can be considered as samples of the gradient of the model's indicator function. Then the problem is reduced to finding indicator function χ whose gradient best approximates a vector field \vec{V} defined by an input points. The problem of computing the indicator function therefore reduces to inverting the gradient operator; that is, finding the scalar function whose gradient $(\nabla\chi)$ best approximates the vector field (\vec{V}) see Figure 2.23 (a) and (b) defined by the input oriented samples (i.e. $\min_{\chi} \|\nabla\chi - \vec{V}\|$). Applying the divergence operator, this problem can be transformed into a standard Poisson problem which computes the scalar function χ whose Laplacian (divergence of gradient) equals to divergence of the vector field \vec{V} :

$$\Delta\chi = \nabla \cdot \nabla\chi = \nabla \cdot \vec{V} \quad (2.10)$$

The implicit function χ is represented using an adaptive octree rather than a regular grid from which appropriate iso-surfaces are extracted. Formulating surface reconstruction as a Poisson problem offers a number of advantages. Many implicit surface fitting methods first segment the data into regions for local fitting, and then combine these local approximations using blending functions. In contrast, Poisson reconstruction is a global solution that considers all the data at once, without resorting to heuristic partitioning or blending [11]. However, some authors pointed out that it has a tendency to over-smooth the features [14, 87, 112]. Kazhdan and Hoppe [114] further improve the capability of its feature preservation, and reduce the time complexity of the

solver to linear with the number of points, thereby enable a faster in terms of a processing time and higher-quality surface reconstruction.

2.4.3 Post-Processing

The requirement of post-processing step depends on the quality of a digitized point cloud, and the type and capability of the reconstruction methods. Due to the complexity of the concerned object's shape and incapacibilities of the reconstruction method, there might be inconsistencies in a re-engineered CAD model. The inconsistencies usually include an inconsistent normal orientation, holes, overlaps, non-manifolds, tunnels, isolated vertex, etc, see Figures 2.24, 2.33, 2.34 for representative inconsistencies. Some reconstruction methods are sensitive to noises as a result they generate a rough surface which needs to be smoothed, especially hull forms are typically need to be smoothed or faired well. In order to repair the inconsistencies introduced at the pre-processing and reconstruction steps, the post-processing methods are incorporated in the developed framework.

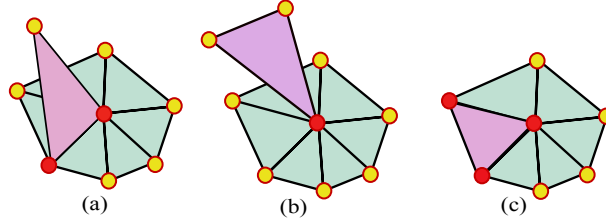


Figure 2.24: Possible flaws of the reconstructed triangular surface. a) the surface edge is incident to more than two edges, b) a non-manifold where a vertex of the surface is incident to open triangle and c) regenerated triangles.

2.4.3.1 Triangular Surface Hole Filling

Despite extensive studies and progresses in creating accurate models of the real objects and environments, it is still non-trivial to rely on a single technique to achieve the required objective. There are inconsistencies introduced at different stages of the CAD model development, which complicate or even sometime halt the operation of the downstream applications. The existence of unnecessary holes in the model is one of the challenging problem, that has to be solved before the CAD model can be used in downstream applications. In the RE context, the holes can be categorized into two groups: holes in a point cloud, and in the surface. The holes in a point cloud are originated from inaccuracies in the data acquisition systems, object's surface properties (e.g. low or specular reflectance) and object's complexity (e.g. occlusions and accessibility limitations). The holes in a point cloud are usually large and it is challenging for the hole filling algorithms to preserve features. The holes in the surfaces are usually introduced due to incapacibilities of reconstruction algorithms.

In literature, there are hole filling techniques which are able to fill holes in point clouds and in surfaces. Therefore, in this work, the techniques are categorized as: hole filling before, during and after surface reconstruction process.

The hole filling before surface reconstruction stands for methods used to fill holes in the point cloud. There are some authors who deal with the hole filling based on a point cloud. Weyrich et al. [115] present a hole filling, point cloud regularization and down-sampling tool, and show the result as shown in Figure 2.25.

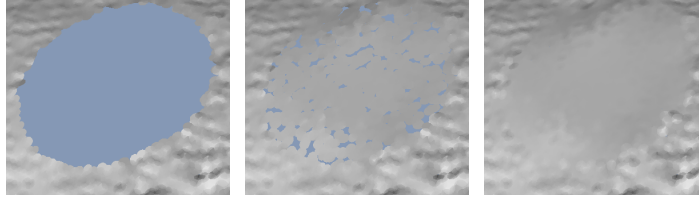


Figure 2.25: Hole filling and smoothing based on a point cloud. From left to right: holes in a point cloud, filled holes, and further up-sampling and smoothing. Adopted from [115]

Li et al. [116] on the other hand, note that the hole filling based on a point cloud usually fails in boundary detection, and propose a hole filling technique which does not require the boundary detection. The method is based on a scan line point cloud and the points that belong to same scan line are fitted to a curve by a cubic parametric spline function.

The hole filling during surface reconstruction refers to the reconstruction techniques that are capable to provide watertight surfaces. As discussed in the surface reconstruction sections, there are explicit and implicit surface representations. The surface reconstruction methods categorized under explicit representations usually tend to leave holes in under-sampled or missing data regions. For instance, the ball pivoting and alpha shapes are usually susceptible to this problem. The most popular approaches based on implicit functions produce hole-free output except Marching Cubes method, however most of them cannot directly reconstruct surfaces with boundaries [116]. Therefore, holes which are produced during reconstruction due to incapacibilities of the reconstruction method should be filled after the reconstruction is performed.

Hole filling after reconstruction usually deals with inconsistencies introduced into the model during the whole processes before this stage (e.g. data acquisition, point cloud pre-processing, point cloud reconstruction, and meshing). Surface reconstruction usually aims to generate parametric surfaces (e.g. B-Spline surfaces or NURBS) or triangulated surfaces. The surface inconsistencies and their repairing strategies will be discussed in the following sections, but here the triangular surface hole filling strategies are discussed as follow. There are various triangular mesh hole filling approaches proposed over the last years and can be mainly classified into two: volume-based and triangular-based. The volume-based approaches indirectly repair the model using an intermediate volumetric grid, while the triangular-based approaches identify and fill the holes directly on the model [117].

Volume-based approaches: The input model is first converted into an intermediate volumetric grid, where each grid point is associated with a positive or negative sign indicating it is inside or outside the model. Next, a polygonal surface is reconstructed that separates the grid points of different signs [117]. These methods excel in their robustness in resolving complex holes, however, they are time consuming because they are computationally extensive.

Mesh-based approaches: The holes are explicitly searched and filled directly on the triangle mesh. There are several authors who attempt to fill holes with this approaches, refer to [117] and the references therein.

In general hole filling techniques have two distinct parts, detecting holes and filling. Usually, there are local hole filling techniques which helps to fill holes at a local level (e.g. small holes or desired holes that have to be repaired). This method has advantages in case there are holes and boundaries that have to be preserved. On the other hand, there are hole filling techniques at a global level, and they provide a watertight model as an output.

2.4.3.2 Triangular Mesh Simplification and Smoothing

This section introduces the triangular surface post-processing methods that are integrated into the developed framework from the Visualization Toolkit (VTK) (<http://www.vtk.org/>).

Simplification: Surface mesh simplification is the process of reducing the number of faces in a reconstructed surface mesh while preserving the overall shape. There are several mesh simplification algorithms in literature, however most of them fall under: vertex decimation, vertex clustering and iterative edge contraction. Surface mesh reduction algorithms based on vertex decimation iteratively select a vertex for removal, remove all adjacent faces, and re-triangulates the resulting hole [118]. In vertex clustering a bounding box is placed around the input model and divides it into a user defined grid. Within each cell, the cell's vertices are clustered together into a single vertex, and the model faces are updated accordingly [118]. An iterative edge contraction, as the name implies, iteratively contracts edges to simplify meshes for details refer to [118]. In this work a surface mesh simplification algorithms from VTK are incorporated in the developed framework. The simplification consists of the *vtkQuadricDecimation* and *vtkQuadricClustering* classes. The *vtkQuadricDecimation* works based on the principle of the vertex decimation algorithm and iteratively reduces the mesh. The *vtkQuadricClustering* class reduces the mesh based on vertex clustering method and is mainly driven by the size of the grid as shown in Figure 2.26.

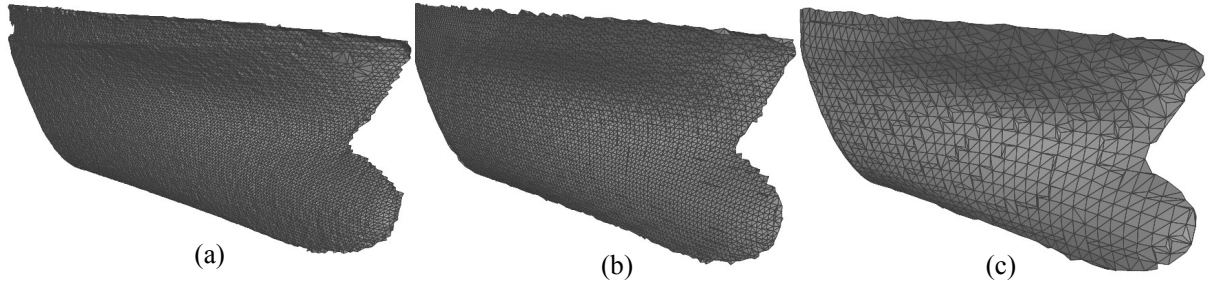


Figure 2.26: Reducing reconstructed triangular surface from a point cloud. a) Reconstructed surface without a reduction (number of triangles = 16265), b) triangular surface reduced by *vtkQuadricClustering* using a grid size of $1 \times 1 \times 1$ mm (number of triangles = 10714). c) The same method but with a grid size of $3 \times 3 \times 3$ mm (number of triangles = 1595).

Smoothing: Some surface reconstruction algorithms are sensitive to noise. Those methods usually produce undulant, noisy, and tunnel kind of triangular surfaces. These surfaces have to be further processed and to be used in downstream applications. For instance, producing a smooth ship hull form surface is necessary and a norm in shipbuilding industry. Therefore, the developed framework includes a triangular surface smoothing algorithms from VTK. The adopted methods are *vtkWindowedSincPolyDataFilter* and *vtkSmoothPolyDataFilter*. The earlier reduces high frequency information in the geometry using a windowed sinc function interpolation, while the latter is related to the well know Laplacian smoothing algorithm which finds a new position for each vertex in the mesh based on a local informations such as positions of the neighbors. Figure 2.27 shows the result of the Laplacian smoothing with different smoothing iterations.

2.5 Knowledge Based Reverse Engineering

Despite substantial developments in RE techniques there are still much to do in achieving high quality CAD models especially when it comes to complex objects. Over the last few years the concept of knowledge based RE receives extensive attention with an increase in the demand of a

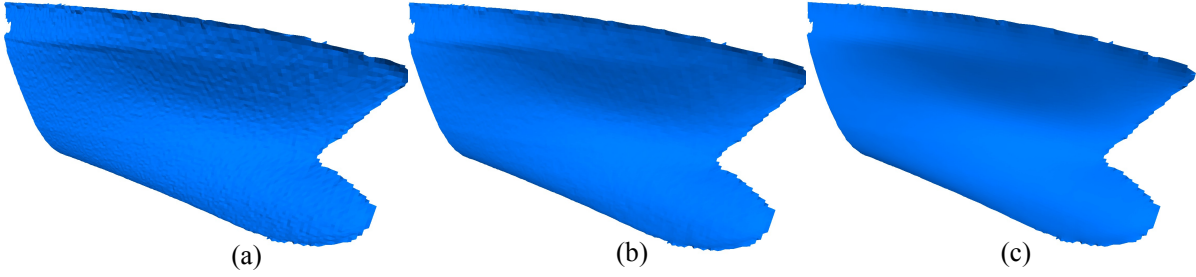


Figure 2.27: Reconstructed surface smoothing. a) Reconstructed surface (original). b) smoothed by the Laplacian method using 100 iterations. c) smoothed using 500 iterations.

quality physical object reconstruction. Despite its application specific behavior, the knowledge based RE is proved that it increases the quality of the output. This section presents a brief description of what knowledge is, and how it is perceived in the ship hull form RE process.

2.5.1 What is Knowledge

In order to understand what knowledge is, it might be useful to clarify the difference between data, information and knowledge. Truong [119] well describes what data, information and knowledge are, in the computer science community sphere. Data are raw facts or figures about an event. They have no meaning on their own. Data can be any numerical quantities (i.e. cost, time, speed, and capacity), text, symbols or attributes derived from an observation. These materials do not provide judgment or interpretation, and they are not organized in any way, but raw material of a decision making may include data. Data, when organized with relevance and for a certain purpose, becomes information. Information is a collection of data, or associated interpretations to describe a particular meaning. Knowledge is the combination of framed experience, contextual information and expert insight, all of which are mixed to become a framework that is able to evaluate and incorporate new experiences and information. In each specific field, knowledge appears not only in documents, books but also in routine activities and processes. Knowledge is derived based on collecting information in an appropriate way that is carried out by humans [119]. Therefore, one might conclude in this context that data is the base for information, and information is the base for knowledge.

2.5.2 Why Knowledge Based RE

Many research efforts have paid to reconstruct CAD models from point clouds efficiently; however there are still several open research questions with respect to accuracy, simplification and efficiency of reconstruction techniques. There are many geometric based RE approaches but the high reconstruction accuracy requirements and newly arisen applications make it still insufficient. Therefore research communities introduce geometric feature sensitive and knowledge based reconstruction strategies. In the geometric reconstruction approaches, data fitting to B-spline surfaces, using both the parameterization and the choice of the knot vectors are difficult [120]. Therefore, the feature sensitive parameterization is introduced which automatically provides more control points for feature areas and allocates more parameter space for feature regions such as sharp edges, smoothed edges, ridges, valleys and prongs [121].

Feature extraction is either performed by estimating differential quantities via a local or global surface fitting (see [92] and the references therein) or based on appropriate integral invariants such as moments of local neighborhoods [122]. Feature sensitive parameterization algorithms are

usually developed for specific applications, e.g. feature sensitive surface extraction from volume data [9], or feature sensitive sampling for remeshing [123], feature sensitive remeshing based on curvature estimation [124], feature sensitive geometry images [97]. For fitting the measurement data, work on feature sensitive filtering and smoothing is certainly of interest [11, 98, 121]. Recent work by Bin et al. [125] propose a feature preserving geometric reconstruction on static models and time-varying data sets. The most explored and commercially available reconstruction approaches are geometric, which usually resulted in frozen CAD models (i.e. not parameterizable and not easy to modify) [126] and not re-usable (meshed surfaces approaches)[23]. Consequently, the possibility of re-engineering or re-design is limited. For example, in a meshed model, a hole has neither diameter nor axis. Moreover, applying constraints like parallelism or a fillet between two faces is difficult. Inevitably, the CAD model provided by RE has to enable engineers to improve, to repair and to update the concerned objects. The current commercially available softwares (*CATIA V5*, *PRO-Engineer*, *RapidForm*, *GeoMagic* etc.) mainly provide more geometric models than real CAD models. They mainly provide sets of complex free form surfaces or primitive features. These are fitted to the 3D point cloud using geometric parameters. In a real CAD model, parameters are not only geometric but also functional and manufacturing. The functional parameters consist the model features (i.e. symmetry, parallelism, known angle between sub features, concentricity, and any relationship between features) and manufacturing parameters include the processes such as a draft angle. The RE processes can be very difficult more over error-prone, if the parameterization of the CAD model is only *geometric* and not *functional*. Therefore the research communities came up with an idea which greatly increases the success of the RE, having prior knowledge about the object. The main concept of knowledge based RE is to define a feature according to several points of views (e.g. a functional point of view, a manufacturing point of view), assuming the design intents. Then, this feature is fitted into the point cloud reconstruction and if needed, is constrained with other features of the part (i.e. parallelism, symmetry etc.) [23, 126, 127]. It is not only limited to feature definition but also includes the use of prior knowledge about data acquisition system, and the reconstruction algorithms. Inclusion of any available knowledge allows an improved RE output even when the data is very noisy, sparse or incomplete [23]. Therefore, integration of knowledge in RE processes for specific object will upgrade the accuracy and efficiency as well.

2.5.3 Type of Knowledge

Knowledge based RE methods have been published for different applications [23, 119, 126–128]. With the increasing complexity of products, comes an increasing quantity of associated knowledge. Thus, many authors incorporated knowledge in RE process, but with different sense, classification, management, modeling and methodology. Ikujiro et al. [129] classify knowledge based on their representation into two: *explicit* and *implicit*. Explicit knowledge can be expressed in words and numbers and shared in the form of data, scientific formulae, specifications, manuals and the like. While, implicit is highly personal and hard to formalize, making it difficult to communicate or share with others. Subjective insights, intuitions, and hunches fall into this category of knowledge. Implicit knowledge is deeply rooted in an individual's actions and experience as well as in the ideals, values, or emotions he or she embraces [129]. Mohamed-Islem et al. [128] discuss the way to integrate the implicit and explicit knowledge in RE. Lexandre et al. [127] consider the knowledge from two point of views (i.e. manufacturing and functional specification) and propose a way to integrate those knowledge into RE. Many consider knowledge for the improvement of the RE output without classification, for instance [23]. Truong [119] classifies the knowledge into four distinct parts: Scene, spatial, data and algorithm. This classification is found very interesting as it includes several aspects of integrated RE process assisted by knowledge. Hence the developed framework in this thesis is using this classification.

2.5.3.1 Scene Knowledge

The scene knowledge contains information related to the content of the scene to be processed, important characteristic of objects (i.e. geometric features, appearance and texture), and the geometry that composes its structure. It can be layered into three classes: *Domain Concept*, *Geometry* and *Characteristic*. The three classes are discussed taking a ship with special interest to the hull form. For instance, domain concept contains classes of objects of interest in the scene. For a scanned ship, one might identify different scene names and particular features such as hull form, superstructure, equipment and outfitting, propulsion system, exhaust system, maneuvering system, as shown in Figure 2.28.

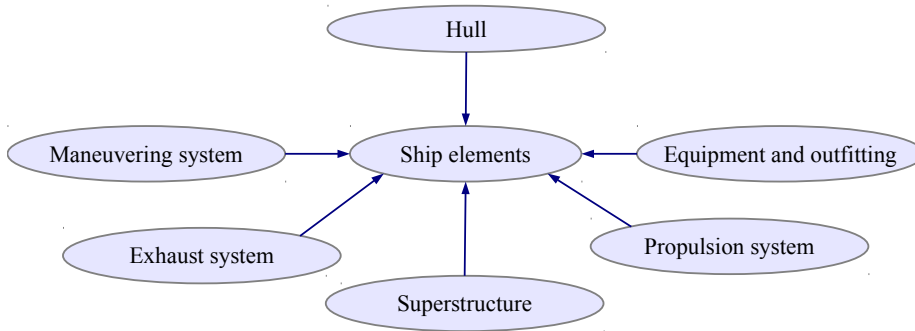


Figure 2.28: Scene objects grouping for the scanned ship.

Taking the hull form from the domain concept, further geometric shapes (geometric class) can be extracted based on features such as planar surface, linear and curved surfaces. The hull could be generally expressed with different features such as a stern region, free to stern region, bulwark curves, bow region, centerline curves, flat of side, flat of bottom and others as shown in Figure 2.29. The geometric class hierarchy can be shown as in Figure 2.30

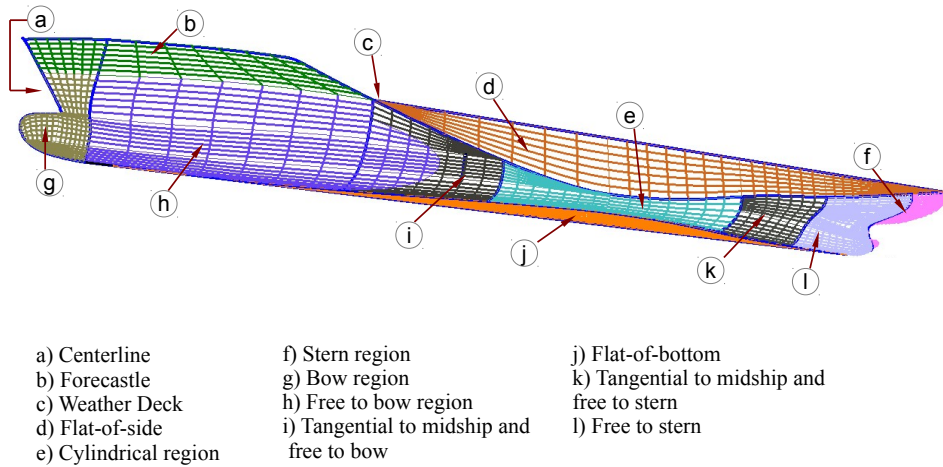


Figure 2.29: Hull form geometric and topological features (typical regions of a hull form).

Additionally characteristic class consists of instances which describe the characteristic of the geometry or data, for example: thin, thick, low density, high density, color, etc. Taking flat of side from geometric class, the characteristic class is shown in Figure 2.31. Scene knowledge is not only important for the identification and classification processes but also supports the selection and guidance of the algorithms.

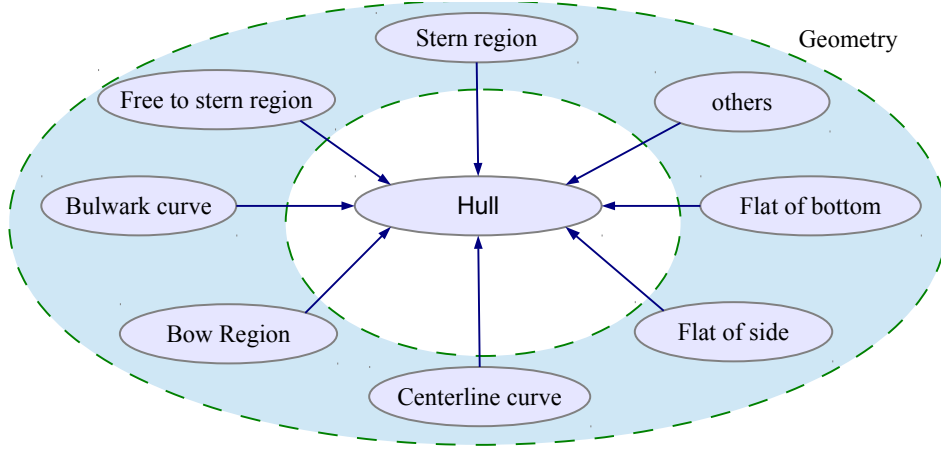


Figure 2.30: Hull form geometry class hierarchy.

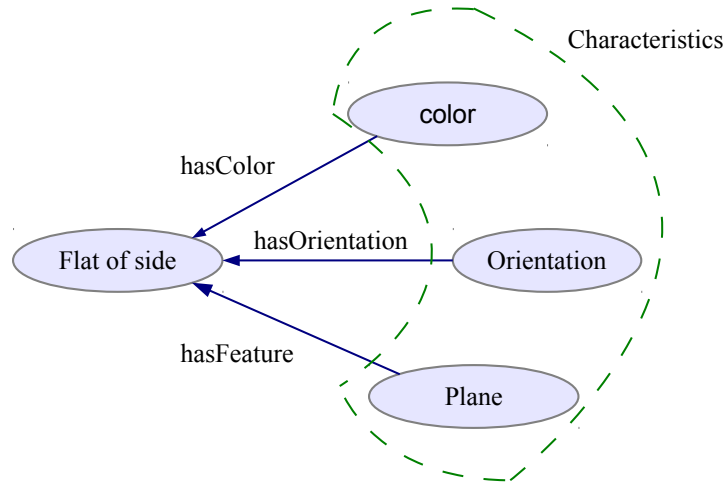


Figure 2.31: Specific characteristics geometric class. Object and data properties characterizing the semantic object.

2.5.3.2 Spatial Knowledge

The spatial knowledge models the relationships between objects in the scene. It is an important factor for the classification process because it supports an object's state disambiguation based on its relationship with the common environment. The spatial knowledge includes standards like the *3D topological knowledge*, *3D metric knowledge* and *3D processing knowledge*. Spatial knowledge contains relationships such as: disjoint, contains, inside, equals, covers and overlaps. They represent the geometric relationships either between components in an object or between objects in the scene. For instance, flat of bottom is parallel to the waterline plane and the flat of side is parallel to the centerline plane and they are perpendicular to each other. Metric knowledge presents important information, because different elements fulfill metric rules that can also be used in the detection and classification process while topological knowledge represents adjacency relationships between scene elements. Certain relations such as *isParallelTo*, *isPerpendicularTo* and *isConnectedTo* will help to characterize and exploit certain spatial relations and make them accessible during the reasoning steps.

2.5.3.3 Data Knowledge

There are different kinds of input data that could be used in a 3D processing task: images, range images, 3D point clouds and other domain related documents, CAD, Geographical Information System (GIS), etc [119]. A collection of necessary resources allows processing to be able to quickly detect/identify targeted objects and achieve reliable results. To do that, different data features such as data characteristics (i.e. in terms of data quality, attributes and errors in data) that influence the numerical processing should be exploited. For example, an object detection algorithm can succeed or fail at a certain point during the process due to the quality of data. To understand data characteristic, a process is required to extract information from data and draws rules from such information. Focusing on point cloud data, the data knowledge can be considered in some aspects such as quality, measurement error and features of measurement devices.

The quality of the data has a tremendous effect during the execution of an algorithm. The thresholds in the algorithm should be properly set to adapt to the quality of data. For instance the threshold for low density point cloud is different from high density. The measurement errors mostly depend on the distance and angle from the scanner to the surface of the object [119]. Basically, point clouds representing an object that is far from the scanner have high probability of containing errors. Therefore any result obtained from such data is uncertain. By a relative evaluation, the assigned probability values for these results indicating how much trust is put into each. The relationship between measurement error and probability of accuracy of the results are modeled in data knowledge. Measurement devices directly influence the way an algorithm should be modeled. Depending on the type of measurement devices, the outputs are also different in terms of data format, data quality and density, therefore, the developed algorithm should be adapted to the output of the devices.

2.5.3.4 Algorithm Knowledge

The integration of the 3D point cloud processing algorithms into the semantic framework requires an interaction between scene data and algorithms. Algorithm knowledge helps the 3D point cloud processing algorithms to adapt to various conditions of input data (point clouds) as well as different scenes. Algorithm knowledge contains all relevant aspects needed to select processing algorithms, generated processing sequences and set parameters for individual algorithms in different situations. Regarding the numerical processing algorithms, effectiveness depends on the quality of data (i.e. resolution, noise), the characteristics of the object that needs to be detected, or other factors depending on a specific case. Finally, the algorithm knowledge characterizes the behavior of algorithms and determines which purpose they fulfill, what inputs they expect, what output they could generate and for which geometries or objects they are designed or what geometries they could properly handle? Based on this knowledge, a dynamic algorithm selection is possible, allowing for a dynamic adaptation to processing situations over a given domain.

2.5.4 Knowledge Acquisition

Usually human beings develop their knowledge about a particular field from the related facts and develop or learn best from their experience over time. Human perceive information through the basic senses (i.e. touch, taste, smell, sight, and hearing) and transfer to the brain. In order to operate correctly any knowledge driven system needs knowledge data in the domain of interest. Unlike human beings, the knowledge driven system is not able to perceive knowledge through senses. It, however, needs support from both human experts to extract knowledge from different

sources and knowledge engineers to pass the encoded knowledge (in the form of the knowledge representation used) from experts to the system. This process is known as knowledge acquisition or knowledge engineering. Knowledge acquisition includes collecting, modeling, analyzing and validating knowledge. Knowledge about a specific domain can be of any form such as symbols, text, sound, pictures, etc., which are often raw data.

2.5.5 Knowledge Representation

Computer programs are not capable to understand the knowledge captured by the knowledge engineers from experts [119]. A knowledge representation is used to encode the acquired knowledge into forms that are usable for computer programs. A knowledge representation appears in different forms, the most popular ones are semantic networks, rules and logical presentations. Semantic networks provide a way of interpreting the meaning of knowledge by a visual graph. Semantic networks formalisms express the taxonomic structure of categories of concepts and relationships between them. A semantic network is a graph which consists of nodes and arcs. Nodes represent concepts while arcs denote relations between concepts. Statements about a domain of interest are represented through nodes and arcs in the graph as shown in Figure 2.32.

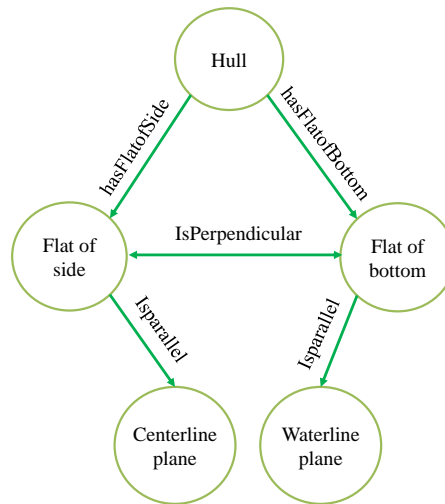


Figure 2.32: Hull planar features as a semantic network

Knowledge can also be represented by rules which usually appear as “if-then” constructs. The “if-then” rule statements are used to formulate the conditional statements that comprise fuzzy logic. The if-part is referred to as a premise and the then-part is referred to as a conclusion. For example: *if a hull has a flat of bottom then it is parallel to a waterline plane*. Such phrases are formalized to use predicates and variables over objects of the domain of interest for more detail refer [119]. The other way to represent knowledge is logic, which can be formulated through semantic networks and rules to give them a precise semantics, for details refer to [119, 130].

2.6 CAD Model Integration into Numerical Simulation

The CAD model preparation for downstream applications is the most tedious and time consuming task if proper tools are not in place. This section introduces different CAD data inconsistencies with their origins. The surface healing methods and their challenges are presented. Ship hull form surface healing, region identification and domain preparation requirements are presented subsequently.

2.6.1 CAD Healing

CAD systems allow designers to build shapes by modeling the surface patches or polygons that comprise its boundaries. These boundaries are often represented as a composite parametric surfaces, or employ a discrete representation in terms of triangular facets. Ideally the CAD software generates at least a watertight (C^0 continuity) surface representation which is then stored in a format that maintains both a geometric description of each patch as well as topological connectivity between patches [131]. In addition to generating and storing geometrical and topological data, the capability of exchanging information with other systems in a large range of data formats; is also one the most important requirement for CAD systems.

Virtually all computer-based design tasks commence with the use of CAD systems to create detailed geometrical models. These models serve as the base for diverse analysis tools, such as CFD, stress analysis, geophysical data exploration, computational electromagnets, etc. The models are also employed in many manufacturing processes, such as numerical-control machining, injection molding, and casting. The success of such downstream applications is, of course, predicated based on the receipt of geometrical models that are accurate, self-consistent, and economical in data volume [132]. The representation of a CAD model includes feature based data and a resulting Boundary-Representation (B-Rep) model. The B-Rep model consists of more than just geometry, and indeed one of the major problems in accessing CAD geometry is the oversimplification of what constitutes a valid B-Rep model. B-Rep models contain geometry (i.e. shapes), topology (i.e. how objects are connected), and tolerances (i.e. how closely do they actually fit together). These constitutes of a CAD model can be accessed by the CAD systems to define a valid B-Rep model. Therefore, a valid B-Rep model should be considered to consist of geometry, topology, tolerances, and methods used by the CAD system it was defined within [133]. Therefore, the CAD models should be adapted to the requirements of downstream applications through advanced algorithms.

2.6.1.1 CAD Inconsistencies and Its Origins

Despite much work and major advances in geometric and solid modeling, practical implementations of geometric modeling operations remain error prone, and the goal of implementing correct, efficient, and robust systems has not yet been attained. There is an agreement that the problem is serious, but the strategy towards the solution is unclear [134]. This statement has been made almost 20 years back, but the problem remains; as the computational capability advanced a lot for the last two decades because the problem difficulty seems to be rooted in the interaction of an approximate numerical and exact symbolic data. Geometric objects belong conceptually to a continuous domain, yet they are almost always analyzed by algorithms doing discrete computation.

The sources of geometrical and topological inconsistencies can be further explained looking at the two CAD data development main categories: the *nature* and *approach*.

Nature refers to the digitization of a real-world object (i.e. RE) and the development of a virtual CAD model from high-level concepts abstraction and logics (i.e. FE). The FE CAD processes introduce different kinds of inconsistencies such as gaps, intersections, overlaps, degeneracies and singularities. These defects are originated either from inaccuracies in the modeling process or in representations. On the other hand, digitization of physical objects mainly leads to point cloud inconsistencies.

Approach includes all algorithms used to upgrade or exchange one form of data structure to another. The approach includes generation of parametric surface from curves network (i.e. which

may lead to gaps, overlaps), tessellation (i.e. which may lead to gaps, intersections, degeneracies), surface reconstruction from points (i.e. which may lead to topological noise, holes, gaps, and etc), solid model boundary extraction (i.e. singularities), translation from one CAD data standard file format to another, etc.

2.6.1.2 CAD Inconsistencies Classification

It is useful to categorize the inconsistencies in CAD geometries to design a successful repairing algorithm. The CAD inconsistencies can be classified based on the application process or based on its geometry and topology. Based on a general numerical simulation process, CAD geometric inconsistencies can be categorized into two: dependent and independent CAD errors. Independent CAD errors are introduced in the design process from concept abstraction or digitization from real-world object processes. These include the errors in measurement methods, modeling processes and data exchange issue and so on. On the other hand, dependent CAD errors are usually introduced during grid generation and refer to errors created between the CAD data model and virtual laboratories such as CFD, FEA, CAM, etc. In general, independent CAD errors comprise all errors that can be created before mesh generation while dependent errors are introduced after mesh generation.

CAD inconsistencies can also be categorized into geometrical and topological inconsistencies. Dependent and independent CAD errors can fall into both topological and geometrical errors.

Independent CAD errors refer to global continuity of the geometric model and further classified as geometric errors such as non-abutting surface patches; curve self intersections; surface self intersections; inconsistent normal vectors between surface patches and topological errors such as unshared vertices's of neighboring edges; unshared adjacent edges; vertices's non-matching no underlying curve; unclosed loops of the face; non-lying of the boundary curve on a surface; incorrectness of face loops; overlapping faces; intersecting faces. On the other hand, dependent CAD errors are also categorized into geometrical and topological errors and refer to badly meshed geometries and result from the meshing algorithms. Some representative geometrical and topological inconsistencies are shown in Figures 2.33 and 2.34 respectively. For more details on the classification of CAD inconsistencies refer to [135] and for several list of a collective topological and geometrical inconsistencies refer to [136].

2.6.1.3 Needs and Challenges of CAD Repair

Downstream applications such as mesh generation, structural/fluid/thermal analysis, rapid prototyping, numerical controlled machining, casting, computer graphics, and real-time rendering have specific requirements on the input geometry. Hence, the success of the downstream applications is strongly dependent on the accuracy and consistency of the input geometry [137]. In most cases the problems of CAD model errors do not affect the efficiency of graphical applications, as these errors are too small to be observed visually, however major problems encounter in downstream applications [135]. For instance, quality mesh is essential for engineering analysis such as CFD, FEA. Therefore, geometric data models that have been created by mesh generation software or imported from external CAD systems have to satisfy quality constraints which usually impede automatic mesh generation. These quality constraints required by mesh generators usually include C^0 continuity and a consistent representation of geometric models [131, 136]. Therefore it is necessary to adapt/repair the geometrical entities by changing their mathematical description while maintaining the same geometrical shape. The research community in this field addresses the importance and the challenge difficulty of the CAD repair. Despite

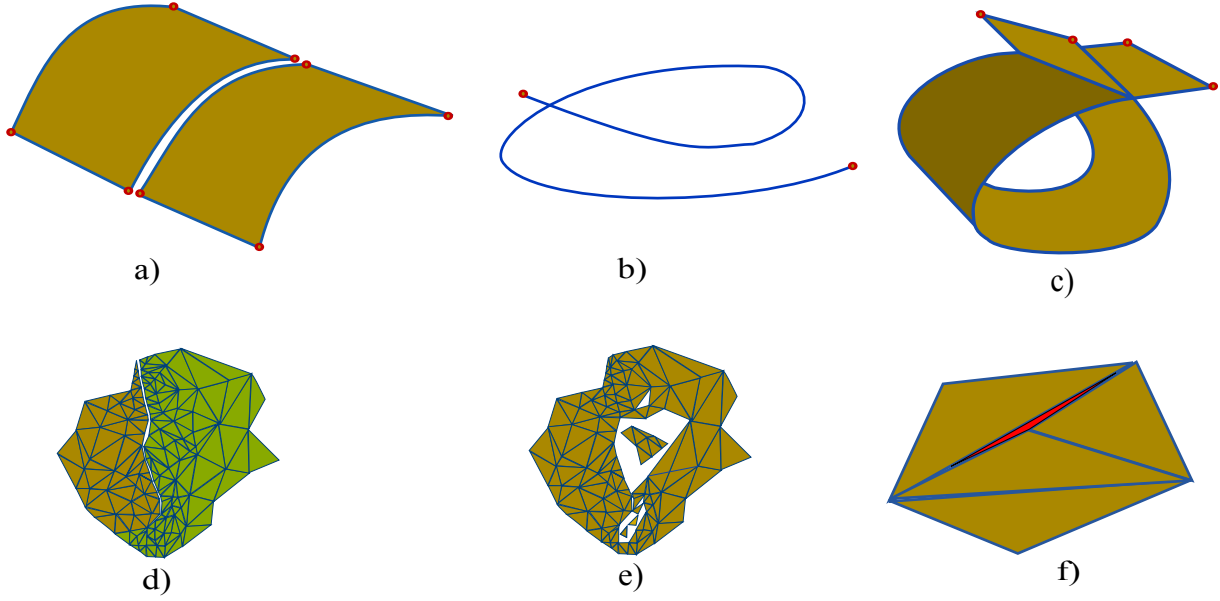


Figure 2.33: Independent and dependent geometric inconsistencies. a) Non-abutting surface patches, b) self intersecting curve, c) self intersecting surface, d) gaps or missing pieces between surfaces, e) missing pieces within surfaces (holes), f) degenerate elements. The inconsistencies in a, b, c are independent while d, e, f are dependent.

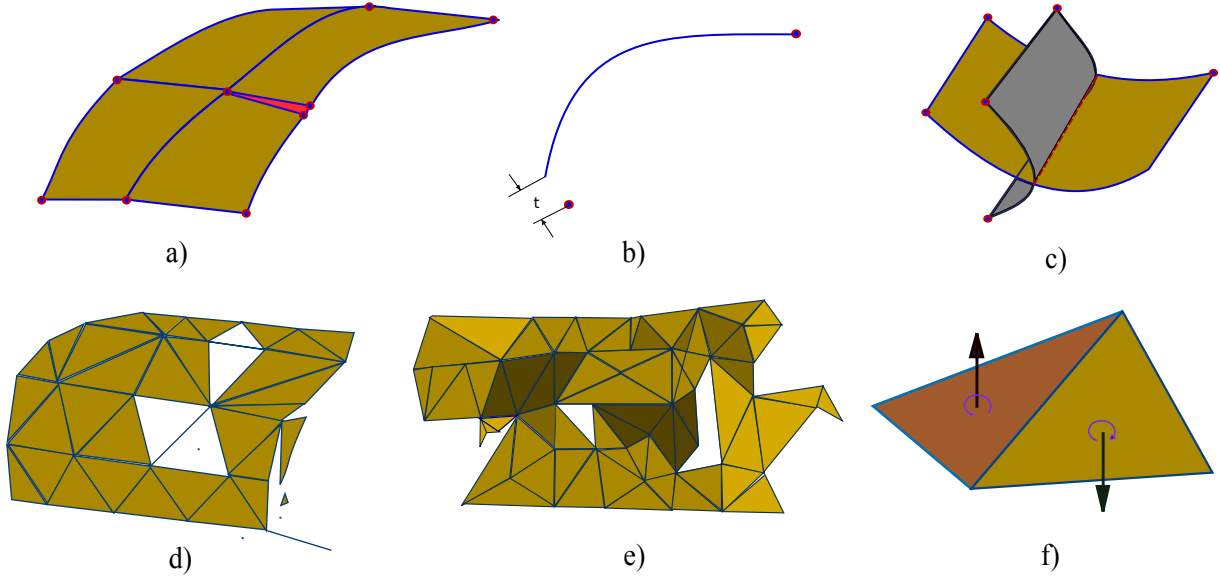


Figure 2.34: Independent and dependent topological inconsistencies. a) Small area surfaces, b) vertex on edge does not match underlying curve, c) intersecting faces, d) isolated vertices, e) topological noise (tiny disconnected components and tunnels), f) wrong orientation. The inconsistencies in a, b, c are independent while d, e, f are dependent.

meshing algorithms have attained a high degree of sophistication and reliability, they need an error-free geometrical input. Rida [132] presents a very explicit breakdown of a fighter aircraft by Kenneth Morgan. He shows detailed meshes (with 50 million elements); and characterizes the meshing algorithm as essentially 100% reliable, provides the input CAD model is error-free. He presents the following 'typical' breakdown of the effort in a realistic CFD analysis: 1 to 4 weeks for geometry repair and preparation, 10 to 20 minutes for surface meshing, 3 to 4 hours for volume meshing, and about 1 hour for the actual flow analysis. Other authors [138] discuss process stages of CFD simulation (i.e. pre-processing, flow solution and post-processing of the

results). Pre-processing includes a geometry cleanup and mesh generation to discretize the computational domain. They discuss the inconsistencies related to IGES and STL file formats, and the difficulty to generate an automatic mesh generation directly from these file formats. They describe the situation as the analyst has to manually clean the geometry to make it suitable for grid generation. This cleanup (pre-meshing) process is a very time consuming, expensive and a tedious task for a design/analysis engineer. They finally note that for realistic simulations, this is the most labor-intensive task in the process, preventing true auto-meshing. Bronsart et al. [139] discuss the situation with the generation of a panel mesh around a ship hull to calculate the wave resistance. They reveal that creating panels on the hull surface is a labor intensive task requiring a profound experience-based knowledge and estimated the time required to prepare the geometry and to generate the panel mesh about 30 to 90 % of the total time to perform the wave resistance analysis. In 2011, ITTC computational fluid dynamics specialist committee conduct a questionnaire over the difficulties and limitations of CFD for hydrodynamic analysis. They also reveal that around 42% of the participants have difficulties in grid generation. These difficulties are attributed to mesh generation softwares and CAD model preparation processes. The above literatures reveal that CAD repair is challenging in terms of time and cost associated to it.

2.6.2 Region Identification and Domain Preparation

Ensuring error-free CAD models alone does not guarantee a good mesh or good computational analysis results. Most downstream applications have their own input geometry requirements from which geometric consistency and error-freeness are usually common. However, several requirements are application specific and need to be fulfilled based on individual requirements. The specific requirements include region identification, domain preparation, knuckle detection and other data format related requirements. In order to perform effective, time saving and reasonable computational analyses, the input geometry should be equipped with the necessary input requirements. However, it is rather complex to devise a certain approach for all downstream applications. The attempt might be inevitably fall short as particular cases and new applications arise too frequently. Nevertheless, there are rooms to devise specific strategy for specific downstream application. In this thesis, the region identification and domain preparation strategies are developed and are integrated in the developed framework.

2.7 Base Information Model

The RE, CAD repair and domain preparation framework developed in this work consists of many algorithms to perform varies necessary tasks. The framework is developed mainly based on the Point Cloud Library (PCL) and the Open Computer Aided Software for Computer Aided Design and Engineering (Open CASCADE) Technology which is shortly known as OCCT. Some of the algorithms used in RE processes are implemented from PCL and the curves fitting, surface patches healing, etc. are based on OCCT.

2.7.1 PCL

The PCL is a standalone, large scale, open project for 2D/3D image, point cloud and 3D geometric processing (<http://pointclouds.org/>). The library's algorithms are written in C++ and optimized for solving numerical processing problems. It incorporates several useful algorithms to process digitized data. The library allows using algorithms for filtering, feature estimation, surface reconstruction, registration, model fitting and segmentation.

PCL is split into a series of modular libraries which makes it simple to use and develop [140]. The most important set of libraries includes:

pcl_io library contains classes and functions for reading and writing point cloud data.

pcl_filters library contains outliers and noise removal mechanisms, down sampling, indices extraction and projections, etc. for 3D point cloud data applications.

pcl_features library contains data structures and mechanisms for 3D feature estimation from point cloud data. 3D features (such as surface normal, curvatures, boundary points estimation, etc.) are representations at a certain 3D point or position in space, which describe geometrical patterns based on the information available around the point.

pcl_keypoints library contains implementations of two point cloud keypoint detection algorithms. Keypoints or interest points are points in an image or point cloud that are stable, distinctive, and can be identified using a well-defined detection criterion.

pcl_registration library implements a plethora of point cloud registration algorithms for both organized and unorganized datasets.

pcl_kdtree library provides the kd-tree data-structure, using nearest neighbor searches.

pcl_octree library provides efficient methods for creating a hierarchical tree data structure from point cloud data. This enables spatial partitioning, down-sampling and search operations on the point data set.

pcl_segmentation library contains algorithms for segmenting a point cloud into distinct clusters. These algorithms are best suited to processing a point cloud that is composed of a number of spatially isolated regions.

pcl_sample_consensus library holds SAmple Consensus (SAC) methods like RANSAC and models like planes and cylinders. These can be combined freely in order to detect specific models and their parameters in point cloud.

pcl_surface library deals with reconstructing the original surfaces from 3D scans.

pcl_visualization library is built for the purpose of being able to quickly prototype and visualize the results of algorithms operating on 3D point cloud data.

It also consists of an object recognition module which can be used to recognize objects.

The point cloud processing algorithms implemented in the framework are partially developed based on PCL.

2.7.2 OCCT

OCCT is a software development platform available as open source. It is an object-oriented C++ library designed for rapid production of sophisticated domain-specific design applications. A typical application developed using OCCT deals with two or three-dimensional (2D or 3D) geometric modeling in general-purpose or specialized CAD systems, manufacturing or analysis applications, simulation applications, or illustration tools. It provides a wide range of CAD application from creation of simple primitives (i.e. prism, cylinder, cone and torus) to a complex NURBS parameterization. It provides also several functionalities to design, manipulate, project, interpolate, approximate geometries. It incorporates also Boolean operations (i.e. addition, subtraction,

intersection), tweaking constructions (i.e. fillets, chamfers, and drafts), modeling constructions (i.e. offsets, shelling, hollowing and sweeps), computation of properties (i.e. volume, curvature, center of gravity, etc), etc. It handles also data exchange between different standards which ensures the openness of the OCCT in a multi-software environment, by allowing it to process external data and providing a good level of integration.

OCCT is designed with reference to the STEP standard ISO-10303-42 and consists of different modules which contain several libraries, each library contains classes grouped into packages as shown in Figure 2.35.

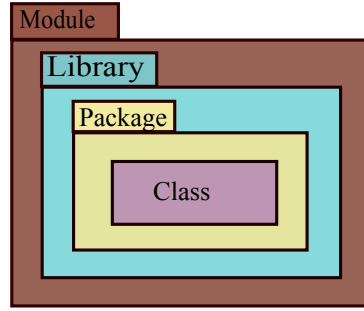


Figure 2.35: General OCCT structure.

It is essential to precisely understand the difference between geometry and topology to work with OCCT. Both analytical geometry which is defined by explicit equations of type $f(x,y) = 0$ and parametric geometry defined by a set of equations depending on one or more parameters are implemented. For complex surfaces, two numerical methods (i.e. interpolation and approximation) or a combination of them are implemented. Geometry is a representation of simple shapes which have mathematical descriptions (such as points, curves, and surfaces) and is used to define the actual dimensions of entities. While, topology defines the connectivity and associativity of the entities. Topology consists of entities as shown in Figure 2.36, and are called “shapes” in OCCT.

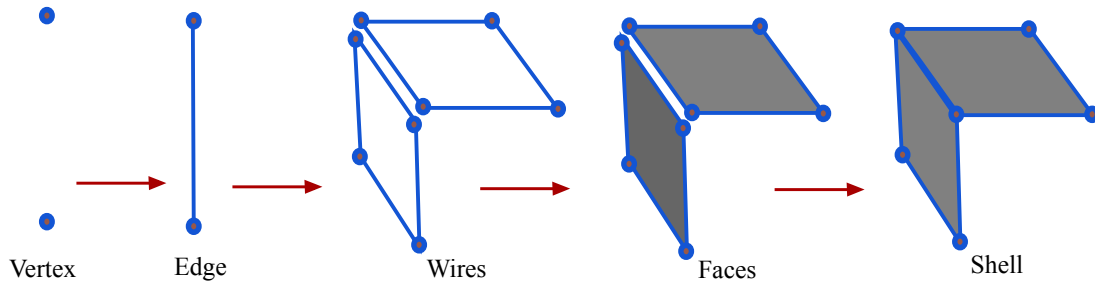


Figure 2.36: Topological entities.

In this library, objects are topologically represented by entities such as a vertex (i.e. a point in 3D space), an edge (i.e. a linear or curved segment), a wire (i.e. a set of consequently connected edges), a face (i.e. a surfaces limited by wire(s)), a shell (i.e. a set of faces connected by their edges), solid (i.e. a part of space limited by shells with material defined in), a compound (i.e. a group of any type of the topological objects), a compsolid (i.e. a set of solids connected by their faces). Therefore, one can construct, explore, visualize, transform and manipulate all geometrical and topological entities in OCCT. Another attractive feature of OCCT is the usage of the *Brep* which describes the data structure for modeling objects in three-dimensions. In Brep modeling, entities are represented by their frontiers or their boundaries. It mixes the geometry with topology. As geometry in Brep is face lies on a surface, an edge lies on a curve and a vertex lies on a point, while topology is related to the connectivity of shapes. For more information on topology and geometry in *Brep* refer to the *TopoDS* and *Geom/Geom2d* packages respectively.

2.7.3 OCCT Tolerance Management

OCCT enables the user to define tolerance as local property. The most commonly used local tolerance values are vertex, edge, and face tolerances. For example, the three vertices of the three different edges are considered sharing a single vertex if the vertex tolerance is equal to or less than the radius of the sphere as shown in Figure 2.37 (left). Two wires are considered corrected if the tolerance value of the edge is less than or equal to the radius of the cylinder around the edges as shown in Figure 2.37 (middle). The face tolerance is the thickness of the pie surrounding surfaces as shown in Figure 2.37(right). In OCCT the default tolerance value is $1e^{-7}$ mm and it is required that the face tolerance should be less or equal to the edge tolerance and the edge tolerance should also be less than or equal to the vertex tolerance. This should be fulfilled for valid geometries in OCCT and the same is true in the developed CAD repairing module.

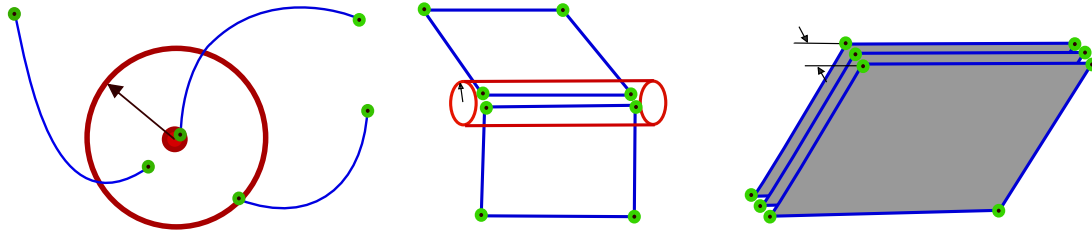


Figure 2.37: Tolerance definition at vertex (left), edge (middle) and face (right) levels

The possibility to define the tolerance values at a vertex, edge and face levels makes OCCT favorable to deal with the inconsistencies in a CAD data model and repair. One can modify and /or enforce tolerance values of an entity in a CAD model, which enables to manipulate the CAD model for better quality.

2.8 Neutral File Formats

Neutral files and neutral file interfaces are needed in order to exchange product data between CAD systems; between CAD systems and virtual laboratories (CFD, FEA, and etc.); between CAD systems and computer aided manufacturing and other applications. One of the most important characteristic of CAD systems is their capability to exchange information with other systems in a large range of formats. There are many neutral file formats available to exchange CAD data. To mention a few: IGES, STandard for the Exchange of Product model data (STEP), Virtual Reality Modeling Language (VRML), STL, Object File Format (OFF), Polygon File Format (PLY) and many more. The developed framework reads a point cloud from *.xyz, *.pcd, and *.ply, parametric surface patches from *.iges, triangulated surfaces from *.stl. The framework provides the output in IGES, STL, STEP, and OFF depending on the input file format and the desired output. IGES and STL are the two commonly used file formats in CAD systems and also primarily used in the developed framework.

2.8.1 IGES

IGES is a neutral representation format for the exchange of a product definition data between CAD systems. Since the release in 1980 of the first version of IGES, the IGES/PDES (Product Data Exchange Specification) organization has added increasingly sophisticated data constructions to the IGES specification. As the capabilities of IGES have been expanded to accommodate more applications, the specification has become more pliable. Some of these changes have added

the complexity and ambiguity to the specification, and this has increased the difficulty of using IGES effectively. However IGES is one of the common information exchanging format between CAD systems using the fundamental entities such as points, curves, surfaces, solids and their combination. Using these entities, IGES is a mechanism for the digital exchange of a database information among CAD systems. Not only between CAD systems, but also with CFD and FEA programs. However, usually it creates a problem for the downstream application as it is unable to provide topological information. As a result, CAD healing is necessary to resolve geometrical and topological errors in IGES file format and adapts for downstream applications (mainly grid generators). In IGES file format, a ship hull form is usually represented by a combination of surface patches as shown in Figure 2.38 without topological informations.

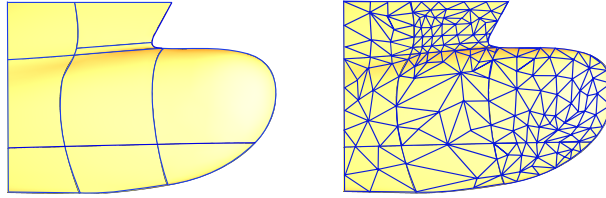


Figure 2.38: Parametric surface patches read from an IGES (left) and triangulated surface representation read from a STL (right)).

2.8.2 STL

STL is a triangular representation of a 3D surface and solid entities as shown in Figure 2.38. The surfaces or solid entities are tessellated into a set of oriented triangles (facets). STL is a widely used data exchange format in the rapid prototyping industry. The generation of STL triangulation is very efficient and can approximate most geometries very precisely. In addition there are various CAD systems and grid generators which offer the functionality to import and export STL surface representation. The developed framework provides standard STL (without color) and colored STL file format for downstream applications. Many commercial (e.g. *Numeca HEXPRESS*) and opensource (e.g. *SnappyHexMesh*) tools are claiming to consistently generate meshes based on STL file format. Due to this fact, the framework targets watertight (C^0 continues) and consistent STL file as an output, but also offers the functionality to write IGES, STEP and OFF file formats.

2.8.3 Others Data Formats

In addition to the above commonly used neutral file formats, other file reading and writing functionalities are integrated. To read and write point clouds three file formats (*.xyz, *.ply and *.pcd) are integrated. Point cloud data format (*.pcd) is a file format from PCL which is able to read point clouds with only three coordinates (xyz) or coordinates with features (normal vectors, and curvatures). Also *.ply which is able to manage both point cloud and polygons with additional features such as color, transparency, surface normals and textures coordinates. The coordinate points (xyz) reading can also be performed by the (*.xyz) file format.

This page intentionally left blank.

CHAPTER 3

CURVES NETWORK HULL FORM REVERSE ENGINEERING

3.1 Introduction

The involvement of several technologies and methodologies makes the RE process always computationally expensive, time taking and more over error-prone. As a result, developing a general RE approach for all kind of objects is challenging, therefore developing an application specific RE method can establish the possibility to obtain a quality reconstructed object with a reasonable associated time and then cost. The main reasons are that it opens the possibility to select or develop a specific algorithm, to define driving parameters accordingly and to incorporate prior knowledge into the RE process. In this chapter, a curves network hull form RE approach is presented. The approach reconstructs a hull surface from an unorganized point cloud via a curves network. Ship hull form geometry is traditionally designed and represented by a curves network, called ship lines plan. This lines plan consists of primarily transversal sections, waterlines and buttock lines and in addition diagonals, i.e. planar intersections of the ship surface in incline planes, for checking purposes; non-planar curves such as side of deck, knuckle lines to delimit the ship surface and its region; in some cases boundary curves of flat of side, and flat of bottom [89]. These curves are combined together to form a curves network from which points on the ship's surface can be interpolated. The curves network hull form representation has been developed centuries ago [89] and is well developed. In addition it is the state of the art in maritime CAD systems and naval architects have a very good experience and knowledge. Hence, the reconstruction of curves network (lines plan) from an unorganized point cloud uses the already well developed traditional surface generation based on curves network.

3.2 Related Works

The use of CAD, computational analysis, and optimization in any industry with regard to complex high-performance products is inevitable. For rapid product development two technologies (i.e. RE and rapid prototyping) have received extensive attention recently from both research and industrial areas. In particular, RE is an important method to reconstruct the CAD model from a physical object that already exists [122, 123]. The process starts from digitizing the existing objects, i.e., capturing a point cloud data of the part surface with digitizers or coordinate measuring devices. The measured points are then transformed into a CAD model using approximation or interpolation techniques [141]. Mathematical attempts to reconstruct surfaces from point clouds was published a long time ago [142], several surface reconstruction techniques have been proposed over the last decades. In recent years, the laser scanning technology has developed rapidly and has become a powerful tool in acquiring the point cloud data of a large and complex object models [141]. Therefore research communities have paid a special attention to develop an efficient and robust surface reconstruction algorithm and some software such as (*3D Reshaper, Geomagic, Polyworks and etc.*) claim to do so. Although various reconstruction methods have been proposed [9, 60, 86, 98, 99, 143, 144], many problems still remain to be addressed due to a geometric complexity of the shapes and noise or outliers in a measured data,

to the contrary high accuracy reconstruction requirements and newly arisen applications [100]. There exists an extensive literatures, which address different questions of RE techniques. The problem starts from point cloud data acquisition systems (i.e. calibration, accuracy, placement, and multiple views), kind of a physical object (i.e. occlusion, surface finish, accessibility) and the nature of data (i.e. extremely disorganized, noisy, and incomplete). The data obtained through all these procedures are the main source of information in the 3D surface reconstruction and are not suitable to integrate directly into the CAD systems [92, 96].

Several strategies have been published to reconstruct CAD models from a point cloud [97, 102, 120, 145, 145–149], but it is not trivial as surface fitting is a highly non-linear problem as the ideal number of control points to be searched in an unknown dimensionality, the knot vectors, and the parameters values of the data points, and the ideal weights of the smoothness functional are unknown. Many research efforts have been paid to figure out effective, robust, time saving and accurate reconstruction strategies for specific or general applications. However there is no straightforward recipe how to proceed to obtain a good quality surface when it comes to complex surfaces.

In RE, the most employed reconstruction method is 3D triangulation of the point cloud. This approach seems to be slow as the number of data points gets larger and larger, and might have limitations when it comes to concave parts and hole loops in a point cloud [98–102]. Furthermore, the surface fitting based on a triangular surface is computationally expensive in terms of the computer memory and processing time as it includes complicated procedures of refinement, parametrization and maintainability of the continuity between networks of patches. Noting the above limitations, an alternative direct surface fitting to point cloud [92, 97, 145, 147, 148] and cross sectional curves network methods [150–152] are proposed. A direct surface fitting is unable to reconstruct occlude surfaces [120]. The possibility to use the direct surface fitting and triangulated surface approaches for a ship hull form reconstruction will be investigated in the following subsequent chapters, while this section only deals with the curves network approach.

Attempts have been made to develop methods to reconstruct maritime structures from a point cloud data [2, 5, 6, 153–156]. However an effective and robust procedures are not in place because of the RE related complexity, and insufficient maritime specific studies. It is also partially because of the difference in traditional RE design processes and strategies with other industries (e.g. automotive, robotics, etc.) where considerable RE practices are well developed.

Therefore the development of specific reconstruction method for a specific application improves a quality as it enables to specifically adjust the algorithm based on the shape and features of a specific object. This chapter presents, a RE method for a specific object (i.e. ship hull form) based on a curves network surface fitting. The proposed method is shared to the scientific community [157] and has the following advantages compare to traditional RE methods:

- avoids the need to compute the local properties (e.g. surface normal and curvature) based on a point cloud, which is always computationally expensive and error prone.
- decreases a large number of input points to manageable size very quickly.
- avoids the 3D triangulation based on a point cloud which is computationally expensive in terms of the computer memory and processing time as it includes complicated procedures of the refinement and parametrization.
- provides a semi-automatic curves (i.e. transversal and waterline sections curves) fitting to a point cloud.
- NURBS surface fitting based on transversal section and waterline curves is well developed and it fits the hull form RE into the traditional hull form design procedure.

3.3 The Methodology

This section presents a semi-automatic curves network based hull form surface reconstruction approach. It consists of three parts (i.e. point cloud pre-processing, cross-sectional curves fitting, and surface generation) as shown in Figure 3.1.

The point cloud pre-processing is developed based on *PCL* and additionally own developed codes. The cross-sectional curves interpolation and approximation after point cloud pre-processing are developed based on *OCCT*. Thereby the NURBS surfaces can be generated based on the curves network using a well developed traditional procedures. The developed methodology reads point cloud data from *xyz-coordinates (*.txt)*, *Point Cloud Data (*.pcd)*, and *Polygon File Format (*.ply)*. The developed system and its components will be explained step by step in the following sections.



Figure 3.1: The general layout of the developed methodology.

3.4 Point Cloud Pre-Processing for Curves Network Fitting

Most often the 3D digitization results in numerous unwanted points. These points frequently belong to objects which surround the object being digitized, such as fixtures, measurement area, or some other part of the assembly to which the digitized part belongs. However, in case of non-contact methods, such as the laser triangulation, those points can originate from any objects in the scene of the measurement. To some extent, unwanted points can also be the result of the measurement errors (i.e. due to operator errors, system-specific errors and/or errors due to specific nature of the digitized object, some external disturbance i.e., vibration), etc. These points, have to be eliminated in order to maintain the quality of the surface reconstruction. The point cloud pre-processing is developed to allow elimination of unwanted points. The pre-processing of a point cloud for hull form curves network reconstruction consists of three modules: outliers removal, cross-sectional point data extraction, and cross-sectional filtering and smoothing.

3.4.1 Outliers Removal

This module removes unwanted points such as Not a Number (NaN) and outliers imported with the input file. The detection of outliers in a point cloud is not a trivial task since there are: geometrical discontinuities caused by occlusions in silhouette boundaries, no prior knowledge about the statistical distribution of points, the existence of noise, and different local point densities [35]. In this work, different outliers removal strategies are implemented to eliminate the outliers exist in noisy sets of points as shown in Figure 3.2 (a). Four robust methods are implemented for this specific application: bounding domain, conditional outliers removal, radius neighborhood and statistical methods.

3.4.1.1 Bounding Domain Method

This is the simplest and rough outliers filtering method next to NaN removal. The user defines the maximum and the minimum 3D coordinates of the bounding domain. Hence the algorithm deletes all point sets outside the bounding domain. It is used to remove extreme outliers points originally not belong to the target object but the fixtures and the surrounding objects or extreme reflections.

3.4.1.2 Conditional Outliers Removal Method

A conditional filtering method removes all indices in a given input cloud that do not satisfy one or more given conditions. This method helps to retrieve some interest area by setting a certain conditions which the given point must satisfy for it to remain in the required point cloud. Comparisons in a 3D space are implemented to retrieve the interest area, for instance, one can retrieve the bow or stern region of the hull form using comparison conditions.

3.4.1.3 Radius Neighborhood Method

It iterates through the entire input points, and determines the number of neighboring points based on the defined radius of the sphere. The points with too few neighbors will be considered as outliers. Figure 3.2 (b) demonstrates the radius outliers removal method, with two user defined variables (i.e. the radius of the sphere and the number of neighboring points which define whether the points are outliers or not). In the figure, the radius r_1 is greater than r_2 , and with the radius r_1 set and the number of neighboring points set to 1, then only the point inside the sphere (r_1) will be considered as outliers. With the radius set to r_2 and the number of allowed neighboring points are 2, then all the points in all spheres shown in the figure will be eliminated from the rest of the point cloud. In this manner, radius neighborhood removal eliminates the points out of the local intended surface.

3.4.1.4 Statistical Method

Digitized point clouds are typically susceptible to a varying point densities and also lead to sparse outliers which complicate the proceeding processes (i.e. normal estimation, curvature estimation, triangulation, curves and surface fitting, etc.) or even lead to failure. The statistical outliers removal is implemented to reduce the irregularities of a point cloud by computing the distribution of a point to neighbors distances in the input dataset. For each point, the mean distance from it to all its neighbors are computed. By assuming that the resulted distribution is Gaussian with a mean (μ) and a standard deviation (σ), all points whose mean distances are outside an interval defined by the global distances mean and standard deviation can be considered as outliers and trimmed from the dataset. Figure 3.2 (c) shows the standard normal Gaussian distribution with the corresponding coefficients and statistical percentage. The algorithm requires to define the coefficient value which is corresponding to the number of points to be deleted or not. For 1σ , 2σ and 3σ about 68, 95, 99.7% of the points will be kept in the point cloud respectively. Figure 3.3 shows an arbitrary plane object point cloud with outliers, and the treatment with different standard deviation coefficient values and the corresponding treated results. This method is robust and efficient outliers removal method and also used to down-sample the point cloud.

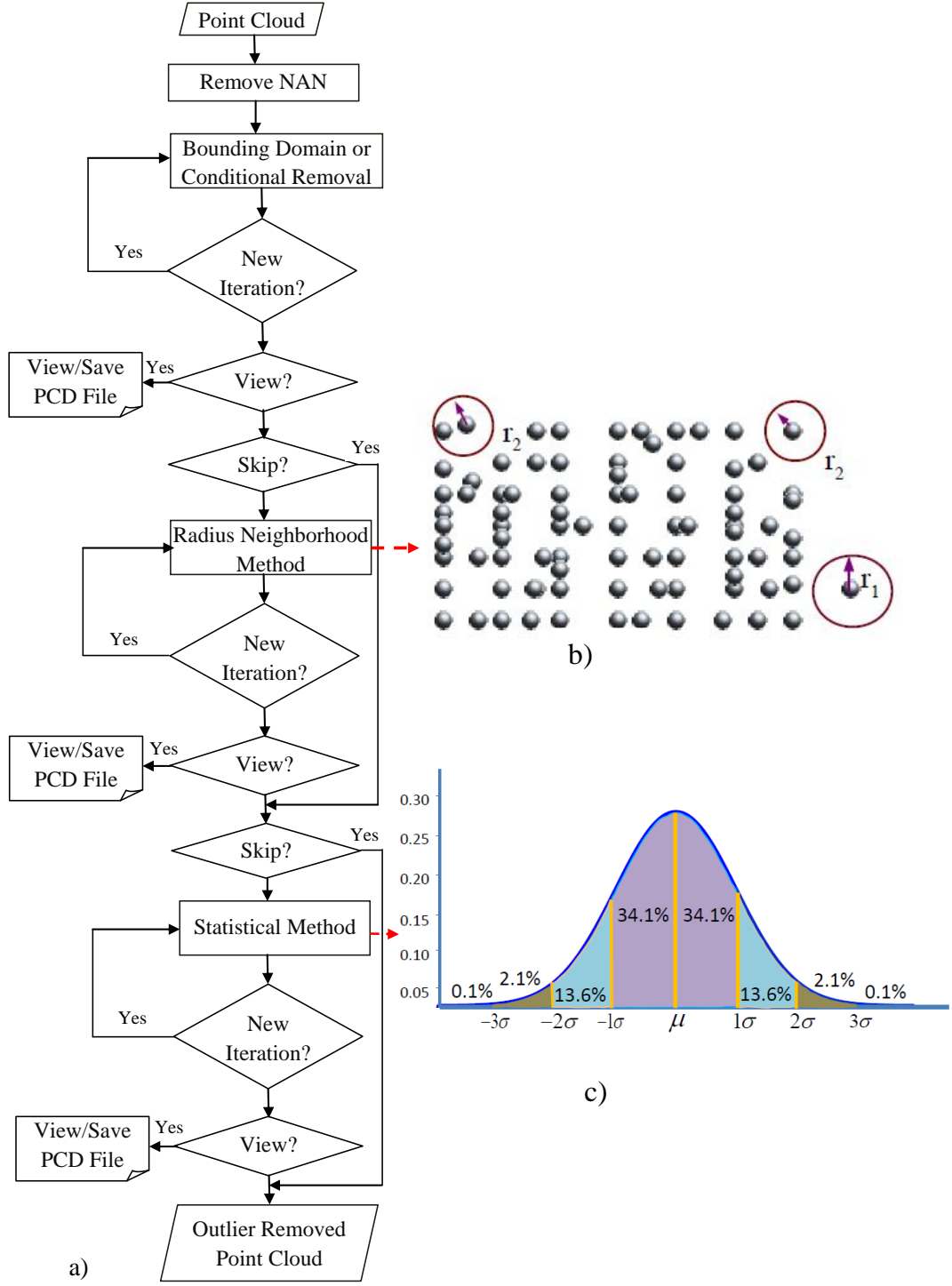


Figure 3.2: The outliers removal methods, a) the general algorithm layout implemented to remove the outliers in the point cloud. b) the working principle of the radius neighborhood outliers removal method. c) the normal Gaussian distribution with different number of standard deviation coefficients used in statistical outliers removal.

3.4.2 Cross-Sectional Point Data Extraction

Once the outliers are eliminated from the raw point cloud. The point cloud is sectioned by transversal and waterline planes to extract the corresponding section curves. For instance transversal section curves can be interpolated or approximated once the stations point cloud

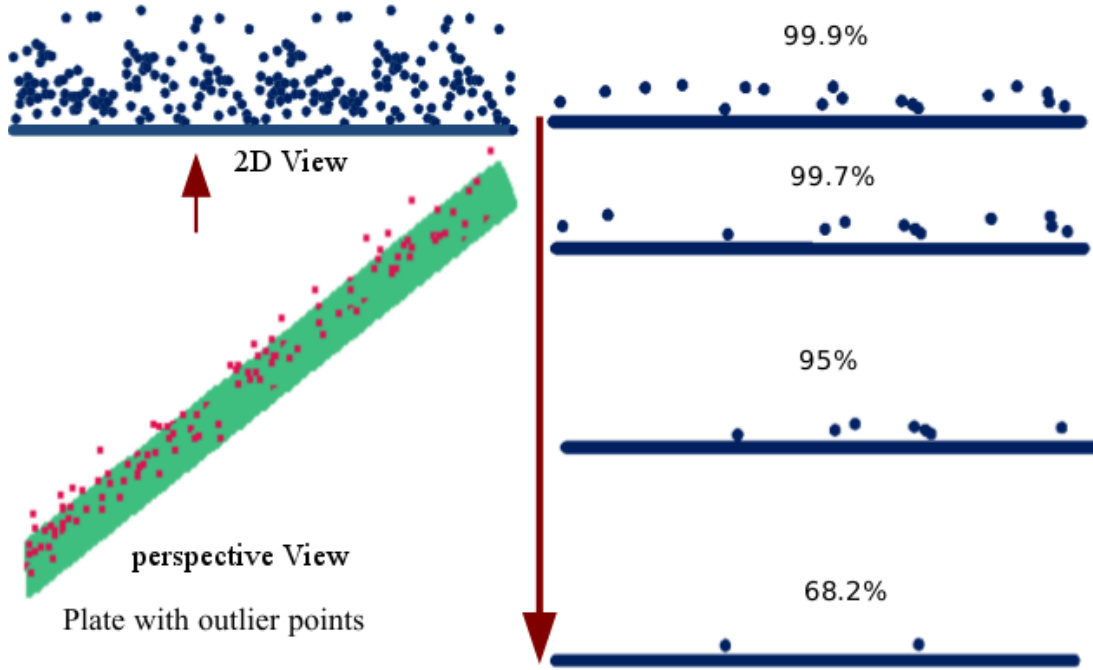


Figure 3.3: The working principle of the statistical outliers removal with different number of standard deviation tolerance coefficients. The object is a plate.

are properly extracted. The same holds for waterline curves. The proposed approach extracts and treats the section point cloud as shown in Figures 3.4 and 3.5. These include the definition of the number of sections and planes, the determination of the point cloud around the section plane within the defined tolerance, and the projection of the points to the center plane. For instance the user defines the axis (i.e. x - for transversal sections and z - for waterline sections) and the desired number of sections. Then, the algorithm automatically extracts the sections points based on the defined diameter (*range*), for example, for transversal sections $X(i+1)-X(i)$.

The extracted cross-sectional points are projected to the mean or center points plane to change the coordinate system from 3D to 2D as shown in Fig 3.4. Hence, it is easy to deal with 2D point cloud for further treatments and curves fitting. At this stage the waterlines and transversal sections point cloud are identified, but still needs more treatment. It is obvious that a wider range ($X(i+1)-X(i)$) implies lower accuracy of the result.

3.4.3 Ordering the Point Cloud

Some of the point cloud pre-processing algorithms and the curve fitting process need strictly ordered point cloud. Therefore, four 2D point sets ordering options are incorporated: ascending/descending, two variables ordering, center-points-angle method, and nearest point method. Ascending/descending is the simplest method which orders based on the increment/decrement of one of the coordinate variable of the points (i.e. x or y), while two variables ordering considers the second variable too (i.e. based on x and then y). Center-points-angle method orders point sets into clockwise or anti-clockwise based on the point sets center of gravity. It orders based on the magnitude of the angles each points makes with the center point. This method is efficient for objects without occlusion. The center point can be manipulated by the user based on the complexity of the shape. The nearest point method is relatively robust compared to the above three methods. For open curves, the outer point should be first determined, and the algorithm determines the nearest point to a query point. Once a nearest point is determined, then a point

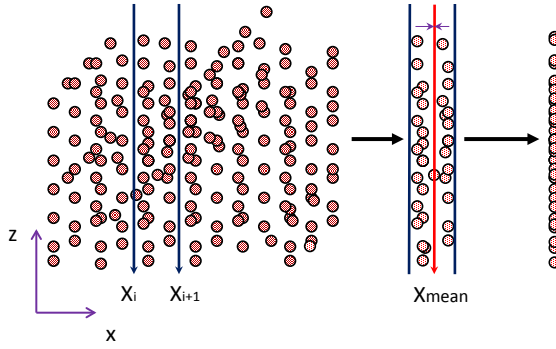


Figure 3.4: Cross-sectional points extraction method.

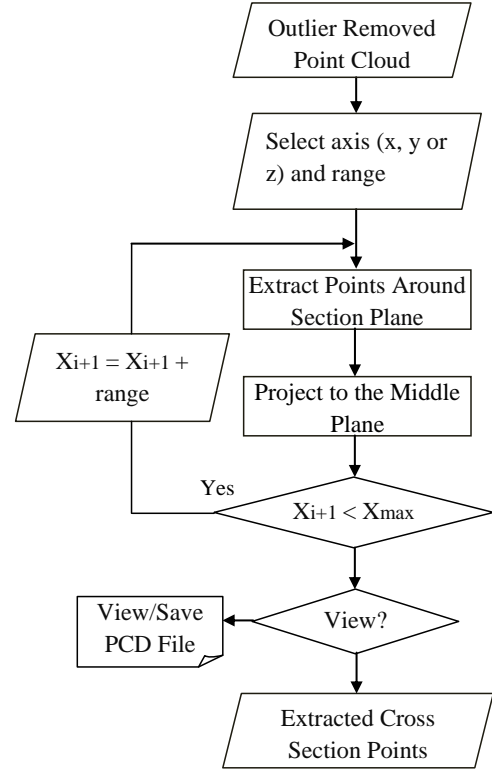


Figure 3.5: Cross-sectional points extraction algorithm chart.

is deleted from the array of the point cloud. Based on this process, the points are ordered for further treatment or curve fitting process as shown in Figure 3.6.

3.4.4 Cross-Sectional Filtering and Smoothing of Point Data

At this stage, the 2D cross-sectional point sets in the desired cross-sectional sections are extracted. The next step is to filter, down-sample and smooth the 2D point sets for an automatic curves interpolation or approximation. As a result three different filtering, smoothing and down-sampling options (i.e. mean, angle and rectangular centroid methods) are incorporated in the developed methodology as shown in Figure 3.6.

3.4.4.1 Mean Method

A mean method works based on a stepped increment and a statistical mean of the data array in a specified step/gap. For instance, for transversal section point sets, the stepped increment (st) is defined by the user and the points fall in that gap will be summed up and the mean of those points will be computed. Let $st(i) = Z(i+1) - Z(i)$, where $i = 0, 1, 2, \dots, n$ and n is the number of steps. The mean of the points between $Z(i+1)$ and $Z(i)$ is computed, for instance say $Xmean(i) = Mean[x(j) + x(j+1) + x(j+2) + \dots + x(m)]$, where $j = 0, 1, 2, \dots, m$ and m is the number of points fall in a single step increment. The mean method is able to filter a point cloud noise in 2D and also smooths. The definition of an appropriate step increment value can produce good result, and a too low value definition leaves noise and outliers and a too high value damages the detail of the object. Figure 3.8 (a) shows the results of the mean method with different step

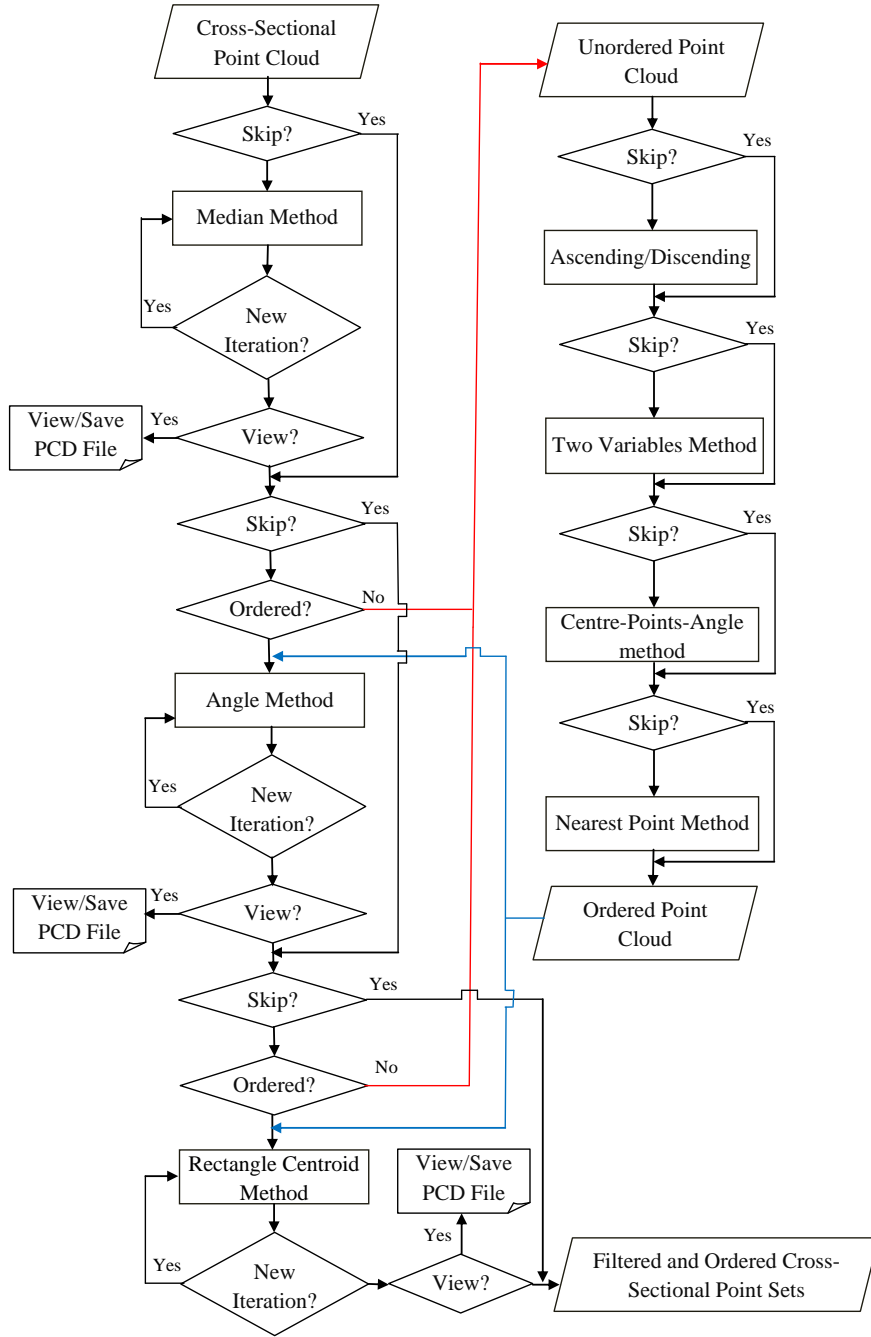


Figure 3.6: Pre-processing (i.e. filtering, down-sampling and smoothing) of a 2D cross-sectional point cloud algorithm chart.

increment values. It is shown that $st = 4mm$ is too high to catch the curvature of the curve and $st = 1mm$ is too low to eliminate unnecessary outliers. It is shown that $st = 2mm$ sufficiently filters and smooths the point cloud for this specific test case.

3.4.4.2 Angle Method

It is very fast and robust but requires a pre-ordered point cloud as an input. The algorithm iterates through all points and checks the angle (β) between the three consecutive points $D(i-1)$,

$D(i)$ and $D(i+1)$ as shown in Figure 3.7. If the angle is less than the default (90^0) or user defined angle, then the algorithm eliminates $D(i)$.

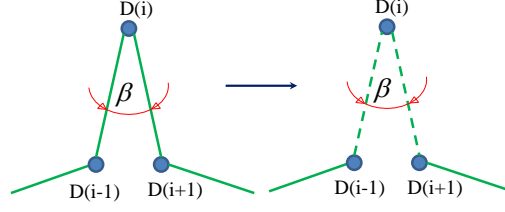


Figure 3.7: 2D point sets filtering and smoothing using the angle method.

The same point cloud, used in mean method, is filtered using angle method as shown in Figure 3.8 (b), it is stable and effective in eliminating impulse noise compared to mean method. In addition to filtering outliers, it also smooths the point point sets. However an appropriate angle value should be defined to preserve the curvature or knuckle of the curves. Increasing the magnitude of an angle filters, smooths and reduces the number of points, but could also be resulted in the loss of geometric details. The possible maximum and minimum magnitude of the angle are 0^0 and 180^0 respectively. Defining an angle value as 180^0 deletes any point out of plane, while 0^0 doesn't affect the original point cloud.

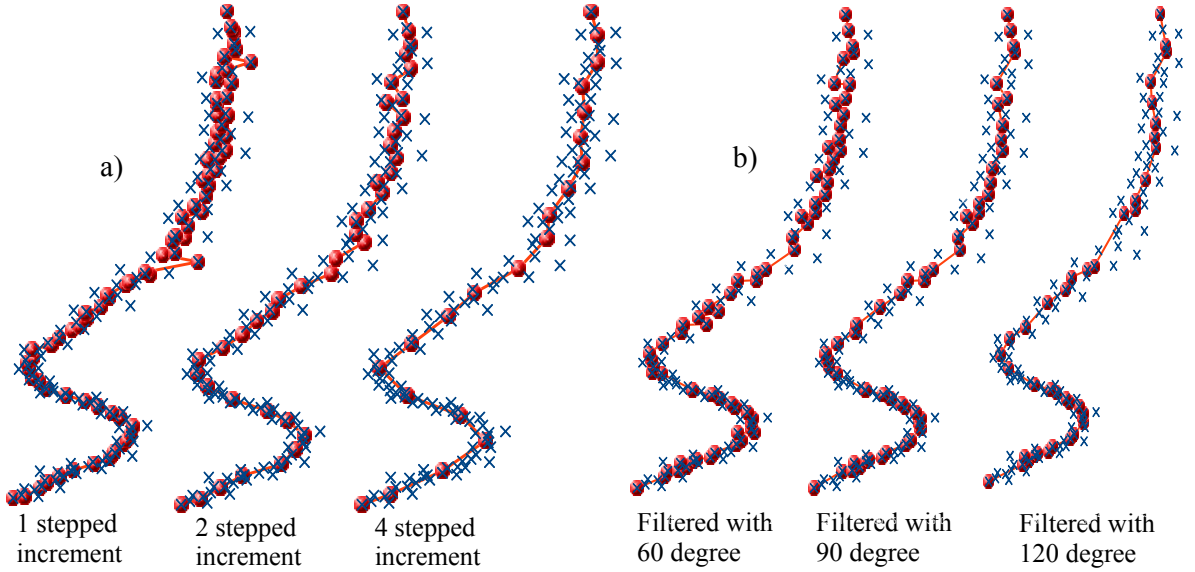


Figure 3.8: The angle filtering and smoothing method. a) the result of the mean method with different step increment values and b) the result of the angle method with different angles. The crossing points are a raw points and the red dot points are the filtered point cloud with corresponding variable and methods.

3.4.4.3 Rectangle Centroid Method

A rectangular centroid method (2D voxel grid) is a 2D point sets filtering and down-sampling method adapted from the 3D voxel grid sampling algorithm in PCL. It creates 2D rectangles over the point sets domain based on a user define length(L) and height(H) of the rectangle. Then, all the points fall in each rectangle are approximated by the centroid of the rectangle. It is efficient in down-sampling and regularization of point sets in addition to outliers filtering and smoothing.

With an increment in the rectangle size, the number of points gets reduced and regularized but also risks the preservation of the features as shown in Figure 3.9.

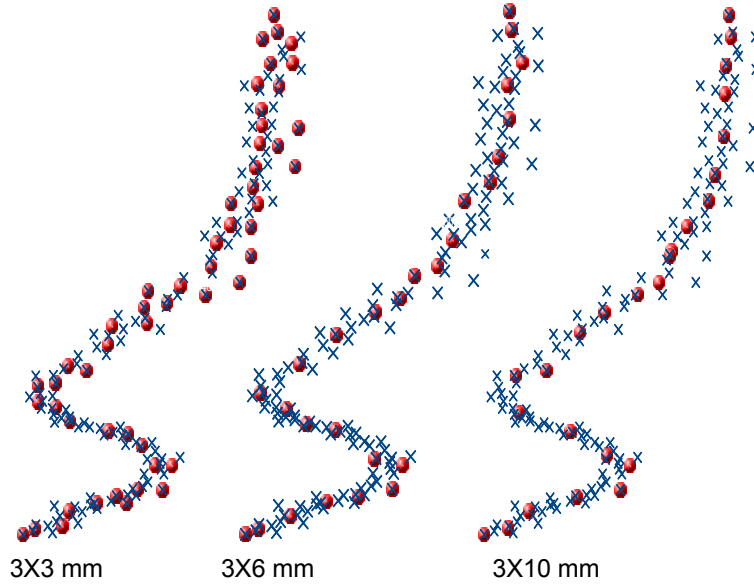


Figure 3.9: Rectangular centroid method with different rectangle sizes ($3\text{mm} \times 3\text{mm}$ with the number of points reduced from 130 to 53 and with $3\text{mm} \times 6\text{mm}$ reduced to 25 and with $3\text{mm} \times 10\text{mm}$ reduced to 24). The input point cloud has 130 data points with artificially introduced noise and outliers.

3.5 Curves Fitting

The development of the polynomial parametric representation has played a major role in advancing the field of the computational geometry. Specially the introduction of a well known flexible polynomial parametric representations such as Bézier and B-Splines curves and surfaces increases the efficiency, capability and the geometric interpretations of polynomial parametric forms. Spline and Bézier curves have many nice properties for the curves design, but not able to represent the simplest curves such as circle, ellipses and other curves that are not represented by polynomials. Therefore both Bézier and B-Spline curves are generalized to rational Bézier curves and Non-Uniform Rational B-Splines (NURBS) respectively to enable the representation capable to manage simple curves mentioned above.

The mathematical description, advantages and limitations of B-Splines, NURBS curves and surfaces are briefly described in chapter 2, for more detail refer to [89, 158, 159]. This section introduces the techniques used to fit the points into transversal and waterline section B-spline curves. The interpolation and approximation are the two well known curves fitting techniques which used to fit curves to arbitrary set of geometric data, such as points and derivatives vectors. The input to a B-Spline interpolation/approximation algorithm usually consists of a set of data points. Thus, the first step is to find a set of parameters that can *fix* these points at certain values. More precisely, if the data points are D_0, D_1, \dots, D_n , then $n+1$ parameters t_0, t_1, \dots, t_n in the domain of the curve must be found so that data point D_k corresponds to parameter t_k for k between 0 and n . This means that if $C(u)$ is a curve that passes through all data points in the given order, then we have $D_k = C(t_k)$ for all $0 \leq k \leq n$.

There are infinite number of possibilities to select the set of parameters (i.e. t_0, t_1, \dots, t_n). However, poorly chosen parameters could be resulted in an unpredictable result by affecting the

shape of the curve [89, 158, 159]. Consequently, this affects the parametrization of the curves. As a result there are many parameter selection methods proposed in literature, which includes the *uniformly spaced*, *chord length*, *centripetal* and *universal methods*. For more details refer to [158, 159].

3.5.1 Curve Global Interpolation

Taking the point sets D_0, D_1, \dots, D_n and fit them with a B-Spline curve of degree p , where $p \leq n$ is an input. The parameter values and the knot vectors are determined using the parametric selection methods and knot vector generations respectively. Once the knot vector and the degree p are determined, the only missing part is a set of $n + 1$ control points. A global interpolation is a simple way of finding these control points of the curves pass through all points as shown in Figure 3.10. For detail mathematical description refer to [89, 158].

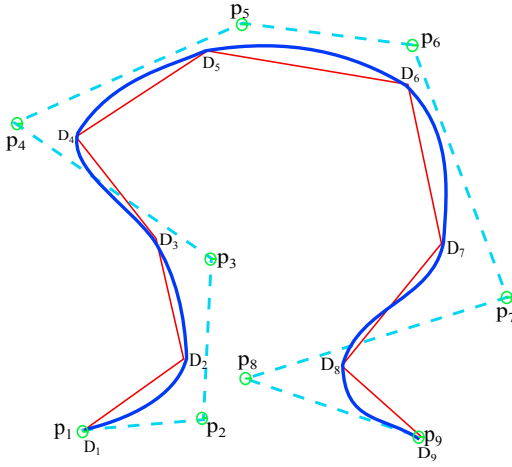


Figure 3.10: Demonstration of curve interpolation.

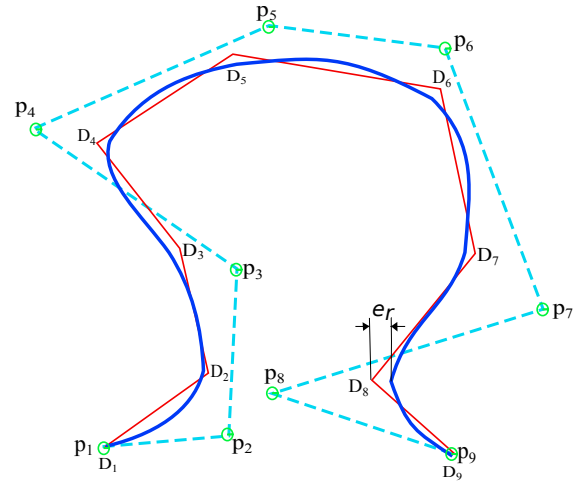


Figure 3.11: Demonstration of curve approximation.

3.5.2 Curve Global Approximation

In interpolation, the curve passes through *all* given data points in a given order. An interpolating curve may wiggle through all data points rather than following the data polygon closely. The approximation technique is introduced to overcome this problem by relaxing the strict requirement that the curve *must* contain all data points. In global approximation, except for the first and last data points, the curve does not have to contain every point. To measure how well a curve can *approximate* the given data polygon, the concept of error distance (e_r) is introduced. The error distance is the distance between a data point and its corresponding point on the curve as shown in Figure 3.11. Thus, if the sum of these error distances is minimized, the curve should follow the shape of the data polygon closely. An interpolating curve is certainly such a solution because the error distance of each data point is zero.

Given a set of $n + 1$ data points, D_0, D_1, \dots, D_n , a degree p , and a number h , where $n > h \geq p \geq 1$, find a B-Spline curve of degree p defined by $h + 1$ control points that satisfies the following conditions: this curve contains the first and last data points (i.e., D_0 and D_n) and this curve approximates the data polygon in the sense of least square. Thus, an approximation is more flexible than interpolation as it not only enables the users to select the degrees but also the number of control points. For detail mathematical descriptions refer to [89, 158].

3.5.3 Semi-Automatic Ship Hull Form Curves Fitting

In the developed framework, both interpolation and approximation curves fitting approaches are implemented to fit B-Spline curves to cross-sectional point sets. Once the point cloud is treated using the methods discussed in the pre-processing section, the pre-processed point sets are fitted to B-Spline curves using the interpolation or approximation algorithms implemented based on OCCT library. The interpolation and approximation algorithms are implemented based on OCCT classes *GeomAPI_Interpolate* and *GeomAPI_PointsToBSpline* respectively. The resulting B-Spline curve is C^2 continuous, except where a tangency constraint is defined on a point through which the curve passes. In this case, it could be decreased to C^1 continuity. The definition of a tangency constraint needs a parameter for each of the input point sets given by parallel table of parameters. For a hull form, the tangent constraint is applied at two extreme points of the curves. Based on the defined parameters the class *Geom_BSplineCurve* returns a B-Spline curve. It is implemented as a semi-automatic hull form cross-sectional curves fitting to point sets as described in *Algorithm 1* for interpolation. The approximation of the cross-sectional curves follow the same procedure except the need for additional user inputs such as degree range (i.e. minimum and maximum), the error distance (e_r) and the continuity. The developed pre-processing methodology prepares a consistently filtered, smoothed and ordered inputs for the interpolation and approximation algorithms. Once the above parameters are defined the points to curves approximation is performed automatically. In *Algorithm 1*, *TopoDS_Wire* and *TopoDS_Compound* are the classes from OCCT used to return wire and a compound of wires respectively.

3.6 Surface Generation

Once the point cloud is pre-processed using the described cross-sectional method, the B-spline curves are interpolated or approximated based on the user preferences. The generation of surfaces based on curves network is well studied and widely incorporated in CAD systems. In this work, Rhinoceros [160] is selected because it provides an extensive NURBS surface fitting, consists of many plug-ins for different specific applications, and includes possibilities to integrate own code through scripting language. It incorporates several NURBS fitting approaches based on curves network such as lofting, extruding, sweeping, revolving, patching and surface from curves network. From these possibilities, lofting, patching and surface from curves network are found to be suitable for the hull form surface reconstruction from a cross-sectional curves network.

3.6.1 Lofting

A lofting creates a smooth surface that blends between the selected curves network. The resulting surface quality is strongly dependent on the difference between the number of control points of consecutive curves and the fairness. The consecutive curves control points number closeness promise a better surface quality in lofting and the curves should be selected in a consistent order. The limitation of lofting is that, it does not manage the crossing curves.

3.6.2 Patching

A patching creates a trimmed NURBS surface from curves network. If the boundary curves (exterior curves) are closed, then the patching trims the surface using the boundary curves automatically. For instance in Figure 3.13 (left) four closed boundary curves are created and the

rest curves (interior curves) help to catch the curvature of the surfaces. In this case, the user does not require to define the ordering of the curves network. The patching method builds the surface patch, first by finding the best fit plane through the selected and sampled points along curves. Then, the surface deforms to match the sampled points. It also offers the possibility to constrain the surface by defining the neighboring surface edge which fits the tangent direction of the surface. The bold curve in Figure 3.13 (left) shows the selection of the surface edge to keep the tangency between adjacent surfaces.

Algorithm 1: The general layout of the developed algorithm for transversal section curves interpolation..

Input : Registered hull form point cloud in (x, y, z coordinates) with file formats (*.pcd, *.txt and *.ply).

User inputs for pre-processing and curves fitting purposes.

Output: B-Spline cross-sectional curves in IGES file format

Algorithm:

```

-Select the desired methods and perform the pre-processing of the point cloud;
-Outliers removal (i.e. bounding domain, conditional, radius neighbor, statistical);
-Define the number of cross sections (n);
/* Pre-processed 3D point cloud available */;
Define the radius of the extraction.
for i: = 0 to n do
    -Extract the points at each section and project to the mean plane;
    /*Cross-sectional 2D point sets available*/;
    -Determine the number of points at each sections (m = Number of points at section i);
    for j: = 0 to m do
        -Cross-sectional filtering and smoothing;
        -Ordering the point cloud ;
        -Select from (mean, angle, and rectangular centroid method);
        -Be sure that the points are consistently ordered;
        /*An array of filtered, smoothed, and ordered point sets available*/;
        -Curves interpolation to array of points;
        -Define the constraints (optional) such as tangential vectors, the number of points;
        -GeomAPI_Interpolate returns a clamped B-Spline curves;
        -Change into topological representation (a wire TopoDS_Wire);
        /*Successfully fitted wire is available*/;
        -Add each wires at each section (j) to a compound (TopoDS_Compound);
    /* A compound of wires at each sections are available */
-Write the wires to IGES file format.
```

3.6.3 Surface from Curves Network

A surface from curves network method creates a NURBS surface from a network of crossing curves. This method offers an automatic creation of a patch boundary (edge curves) which makes it easy and fast compares to the patching method. The edge curves are selected by a user and used to set the allowable tolerance the surface may deviate from the edge curves. The label A, B, C, D in Figure 3.13 (right) shows the edge curves of a surface to be fitted. Similar to the patching method, the surface from curves network offers the use of curves geometry or existing surface edges as a boundary wire (edges). This functionality helps to ensure the continuity and smoothness between adjacent surfaces, thereby it is highly recommended to use surface edges

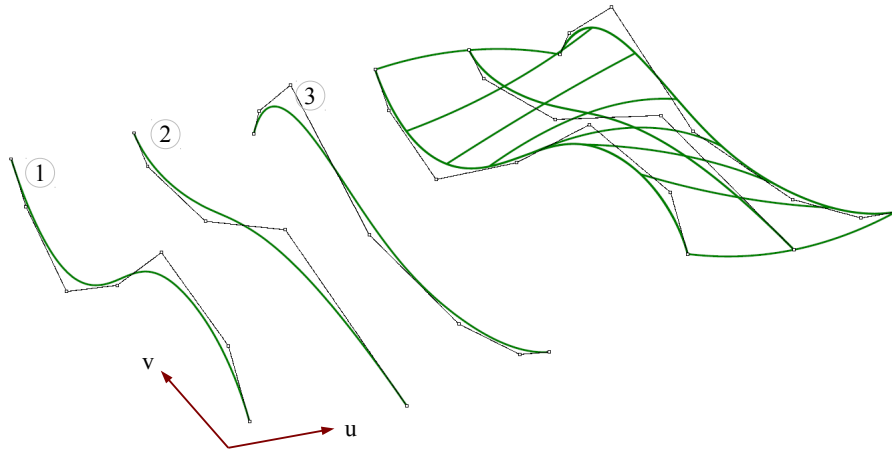


Figure 3.12: Surface lofting from curves network.

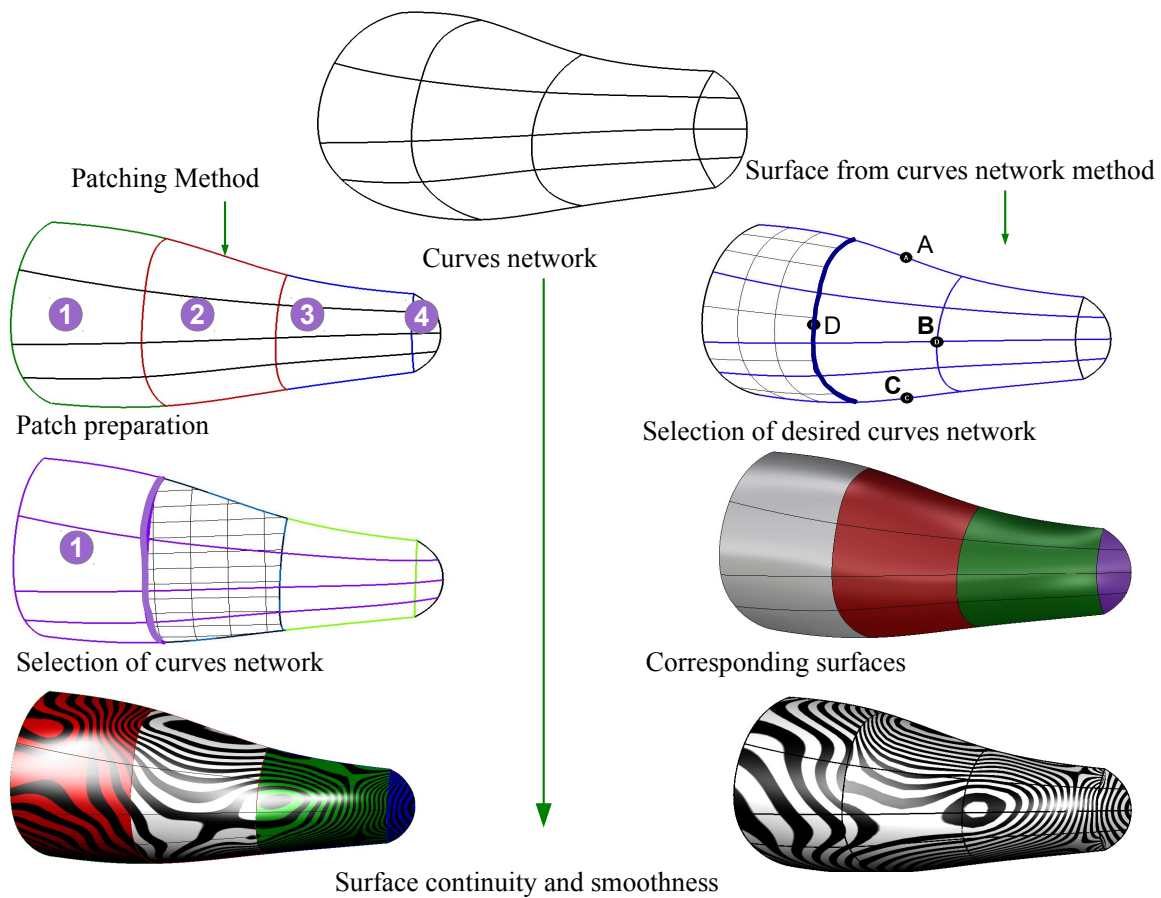


Figure 3.13: NURBS surface generation using patching method (left) and surface from curves network method (right).

rather than curve geometry. The label **D** (bold) in the Figure 3.13(right) shows the selection of the surface edge to create continuity between adjacent surface patches.

3.7 Case Study and Discussion

This section illustrates the developed RE methodology against a ship hull form point cloud data. The case study is categorized into two based on their input sources: artificially produced point cloud from existing hull form surface and measured data from ship model.

3.7.1 Fitting on Produced Point Data

To check the efficiency and robustness of the developed methodology, it should be tested against different types of hull forms. Hence, a software tool which reads an existing hull form CAD data (IGES file format) and generates a point cloud with a random noise and outliers is developed. The developed tool (converter) helps to produce a point cloud from different test cases (i.e. ship hull forms). This part of the case study is demonstrated with a ship hull form represented with around half million points including an artificially introduced noise and outliers.

The developed RE methodology reads a point cloud and offers the user to select outliers removal methods (i.e. bounding domain, conditional, radius neighborhood or statistical). It is possible to use all methods or to select the desired method from the list. At this stage the outliers should be eliminated as shown in Figure 3.14.

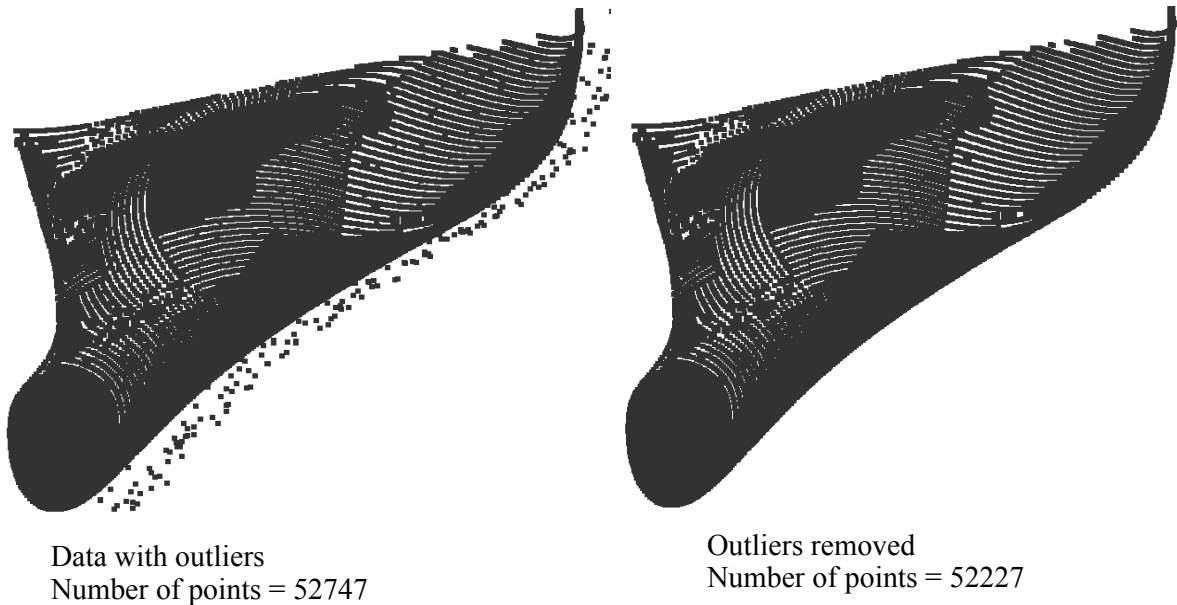


Figure 3.14: A hull form (bow region) before and after pre-processing of a 3D point cloud which includes outliers removal, filtering and down-sampling.

Once outliers are eliminated, the developed system automatically divides the hull form into five regions based on two user defined critical points i.e. bow (*bcp*) and stern (*scp*) as shown in Figure 3.15. The definition of the two critical points avoids the difficulty to fit transversal section curves fitting automatically, and enables the possibility to fit more denser curves to high curvature regions (i.e. bow and stern). Having the five regions segmented, the transversal sections points are extracted. The hull form division is not necessary for the extraction of points in waterline sections. The extracted points should be treated based on the described procedure in point cloud pre-processing section.

The developed pre-processing step generates transversal and waterline sections as shown in Figs. 3.17 and 3.18 respectively. In addition to transversal and waterline sections, centerline and



Figure 3.15: Ship hull form point cloud division.

deck points are extracted using the boundary extraction method implemented in PCL specifically *Concave* method [161] as shown in Figure 3.16.

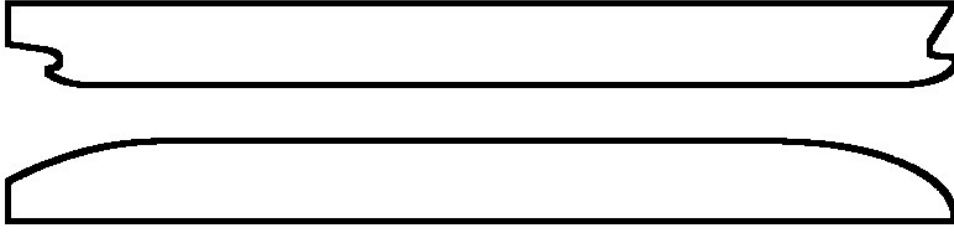


Figure 3.16: Extracted waterline and deck points.

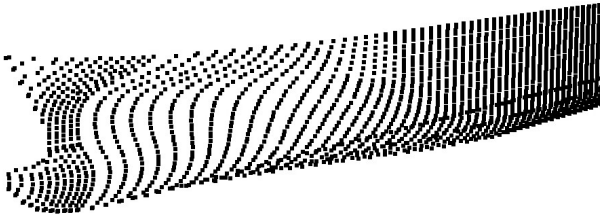


Figure 3.17: Extracted transversal sections point cloud.

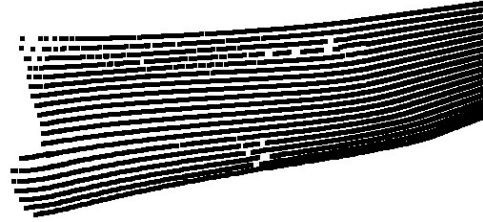


Figure 3.18: Extracted waterline sections point cloud.

The interpolation and approximation functions are implemented in the developed methodology, and both are applicable for the curves fitting as the point cloud pre-processing could filter and smooth the point cloud successfully. In some cases, if the section points are denser and not well smoothed, the approximation could be preferred while for a well treated point cloud, the interpolation function works well. In this work, the deck and centerline points are extracted to increase the accuracy of the exterior boundary. Here one can introduce prior knowledge to improve the accuracy of the reconstruction. For centerline and deck curves fitting the interpolation is generally preferable. After all curves are interpolated or approximated they are all put together as shown in Figure 3.19 and ready for the surface fitting.

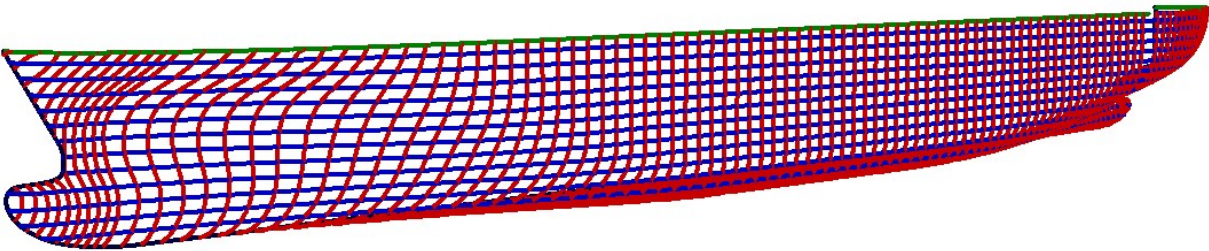


Figure 3.19: The transversal, waterline sections, and centerline and deck curves fitted using the developed method.

In this method, the qualities of the interpolated and approximated curves are highly dependent on the point cloud pre-processing result. However both interpolation and approximation functions

could be used interchangeably for transversal and waterline section curves as the pre-processing treats well the point cloud towards the curves network fitting. The surface fitting using lofting, patching and surface from curves network methods are experimented based on the fitted cross-sectional curves. The traditional lofting only considers either transversal or waterline sections at a time. This is the disadvantage of lofting for the intended curves network based RE. However, it is possible to use transversal and/or waterline section curves accordingly. Figure 3.20 shows how transversal and waterline section curves could be used to generate NURBS hull form surfaces using the traditional lofting. The illustrated method is recommendable to generate hull form surface from the curves network.

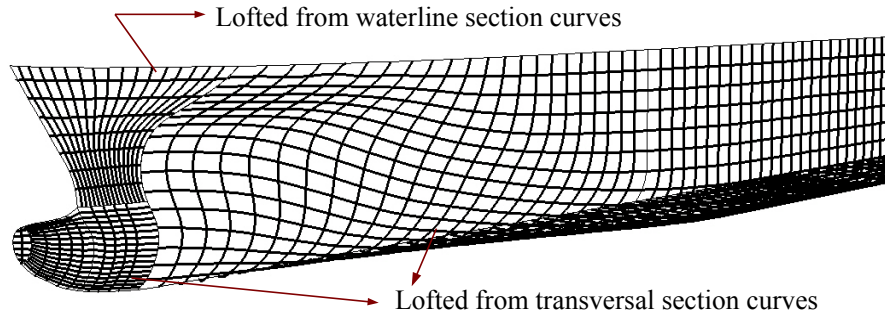


Figure 3.20: A hull form NURBS surface generated from transversal and waterline section curves using lofting.

The surface generation from curves network using the patching method is time consuming because the user has to create the boundary curves manually as shown in Figure 3.21, but the surface from curves network method offers an automatic creation of boundary curves which accelerates the time required to generate a surface from the curves network. It also provides a better surface continuity compared to lofting and patching as shown in Figs. 3.22 and 3.23.

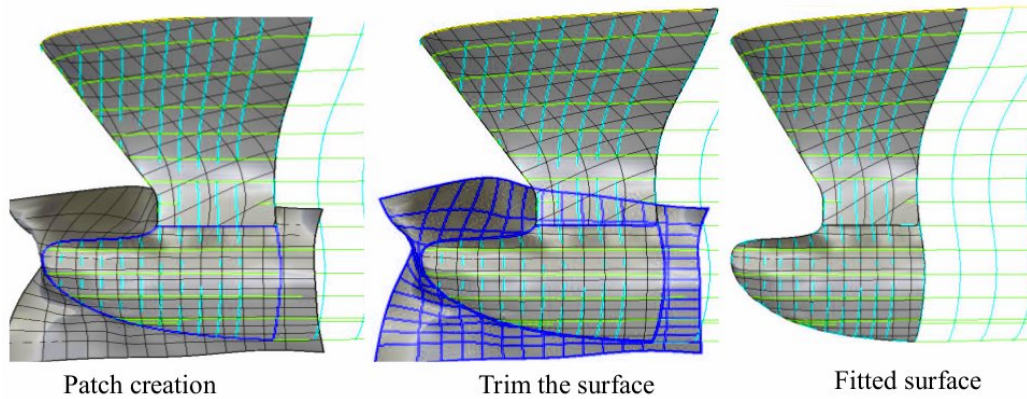


Figure 3.21: Surface creation process using the patching method from curves network.

3.7.2 Fitting on Measured Point Data

A small scale model ship ($LOA = 40cm$) is scanned using DAVID Laserscanner [162] at our department. The measurement is performed from four directions (i.e. front, back, right and left sides). The four patches of the point cloud are transformed to a common global coordinate system. The registered point cloud is used as an input to the developed methodology. It is tested against the developed method using a similar procedure described above and the successive results are shown in Figure 3.24. The developed method shows similar behavior against artificially

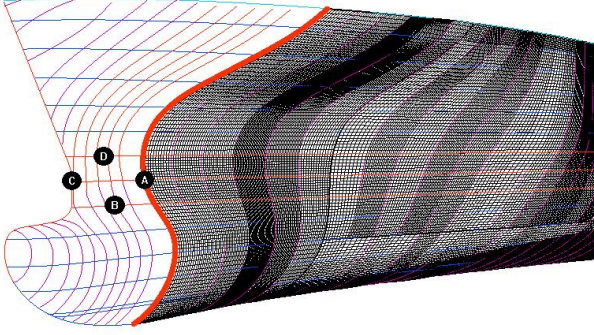


Figure 3.22: Surface generation process using the surface from curves network method. The label A, B, C, D show an automatic creation of boundary curves of the patch to be created.

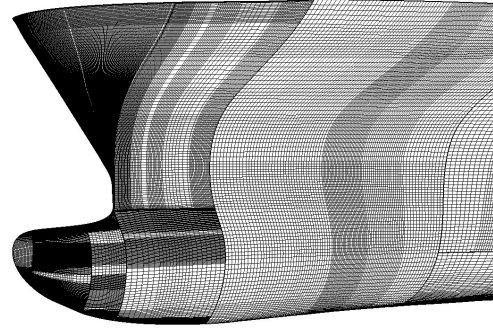


Figure 3.23: A ship hull form NURBS surface reconstructed from curves network using a surface from curves network method.

generated and measured data, however there are some ripples observed on the surfaces reconstructed from the measured data. These ripples are attributed to inaccuracies in laser scanner to objects reference during measurement and poor registration between the measured point cloud patches. This implies that, the measurement techniques and the registration accuracy affect the quality of the downstream processes such as point cloud pre-processing, curves and surface fitting results.

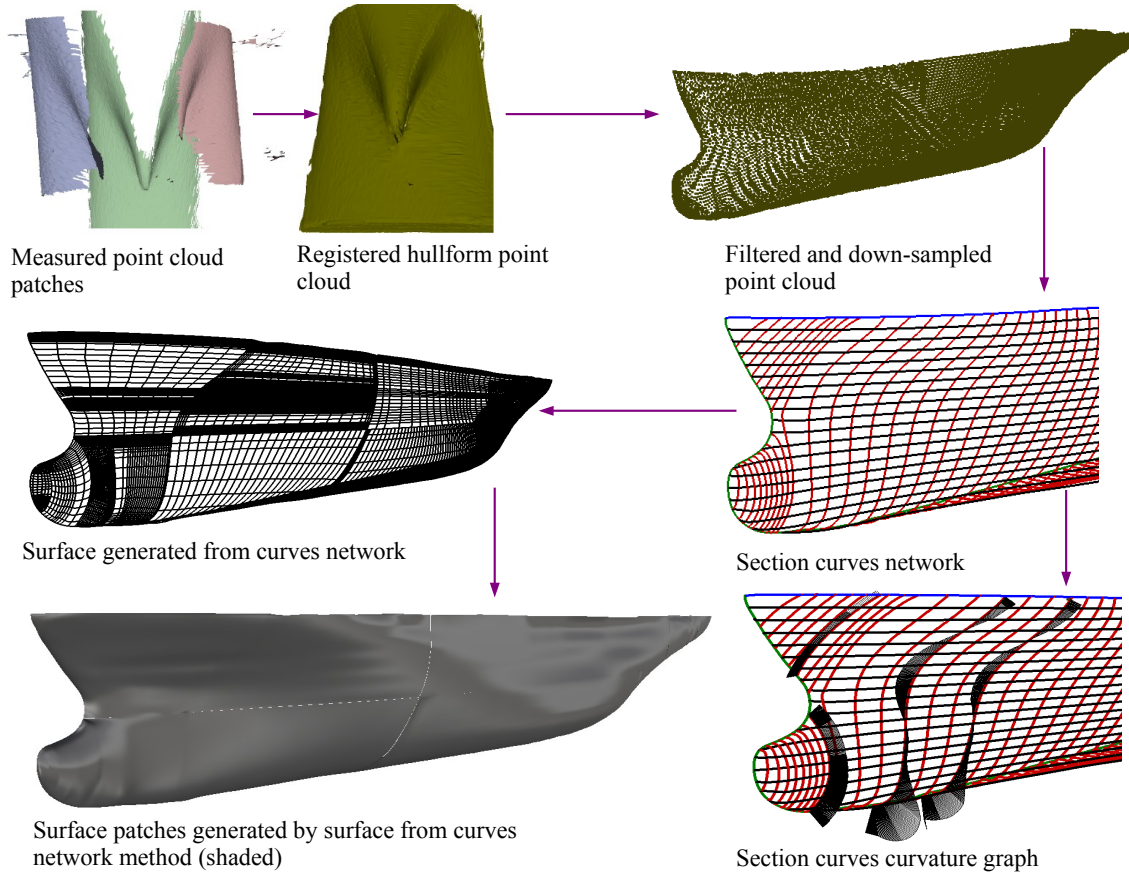


Figure 3.24: Model ship curves network RE, reconstructed from measured data

3.8 Conclusion

The developed methodology presents a B-spline curves network (i.e. transversal and waterline sections) ship hull form reconstruction from an unorganized point cloud. With a receipt of consistently aligned point cloud from various patches of 3-D point cloud data views, the developed methodology provides efficient and time saving reconstruction strategy. The method quickly reduces a large range of data to a manageable size, which is essential for a large objects like ship hull form. In CAD systems, the NURBS surface design based on curves network is well developed, this makes the developed method preferable and easier. The transversal sections, waterline sections, centerline and deck curves are successfully generated using either interpolation or approximation. The efficient point cloud pre-processing enables either the interpolation or approximation to be used interchangeably. However with extreme differences in a density and a large irregularities in an input data, it is generally recommendable to choose approximation. The performed case studies show that the developed methodology has a significant contribution to the hull form reconstruction processes, because a NURBS generation from curves network is well developed. The reconstructed curves network is used to generate NURBS surfaces using three existing surface generation methods (i.e. lofting, patching and surface from curves network). The results reveal that the surface from curves network method provides a better surface quality and also save time.

Generally with the inclusion of prior knowledge and further development in the developed reconstruction process, the method will be a promising strategy to re-engineer the ship hull form.

This page intentionally left blank.

CHAPTER 4

DIRECT SURFACE FITTING HULL FORM REVERSE ENGINEERING

4.1 Introduction

In order to use the output of the RE process in downstream applications, its representation, quality and consistency should fulfill the requirements of those applications. Specific to ship hull forms, downstream applications such as hydrostatic calculations, hydrodynamic performance optimizations, manufacturing processes, sharing data between corporates, etc. rely on the quality and the representation of the CAD model. Therefore, developing a suitable and capable reconstruction approach which leads to a high quality and efficient ship hull form surface representation is indispensable. As discussed in chapter 2, there are several approaches to retrieve surfaces from an unorganized point cloud. In this chapter a direct parametric surface (i.e. NURBS) fitting to an unorganized point cloud will be dealt with.

A direct surface fitting to an unorganized point cloud is not trivial as the surface fitting is a highly non-linear problem which imposes many difficulties. The difficulties include the search for an ideal number of control points in unknown dimensionality, the determination of several unknowns such as knot vectors, the parameter values of the data points, and the ideal weights of the smoothness functional. Many research efforts have been paid to figure out effective, robust, time saving and accurate reconstruction strategies for specific or general applications. However, there is no clear cut recipe how to proceed to obtain the best surface out of a messy and unorganized point cloud, especially when it comes to complex surfaces consisting of planes, arcs and free surfaces.

The key question and labor intensive part of a surface reconstruction is assigning good parameter values to the data points, i.e., to locate points on the smooth surface which are close to the data points [146]. These might include aspects such as finding a reasonably good initial parameterization, optimizing not only the surface shape but also the parameter values of each data points. Several methods exist to solve these problems, which can be applied under certain circumstances. These methods include simple base surfaces such as a plane or a cylinder etc. [97, 146]. Another common and alternative way for an initial parameterization is a 3D triangulation of the point cloud. This approach seems to be slow as the number of data points gets larger and larger, and might have limitations when it comes to concave parts and hole loops within a point cloud [98, 99, 114, 125]. Furthermore the surface fitting based on triangular surfaces is computationally expensive in terms of the computer memory and processing time as it includes complicated procedures of a refinement, parameterization and maintainability of the continuity between neighboring patches. Noting the above limitations, a simple surface fitting algorithm which does not require prior construction of polygon meshes, mesh refinement, and data parameterization is developed. The method projects the data points on a regression plane of the cloud, and then detects regions where the cloud lacks points and regions where the cloud is too dense. By inserting and removing points, the point cloud is regularized and approximated by a NURBS surface. However, it has a limitation to reconstruct complex objects with occlusion

[97] and different features such as sharp edges, smoothed edges, ridges, valleys and prongs. Attempts are made to preserve features by providing more control points to the feature areas and to allocate more parameter space for the feature regions [126]. Though considerable attempts have been made to develop robust, relatively accurate and resource efficient algorithms, there is still a demand to optimize the most important variables of CAD systems with regards to RE requirements. The variables might include robustness, accuracy, time and cost required for an intended application.

The CAD model provided by a RE process has to enable engineers to redesign the form of the concerned object in order to improve, to repair its geometry or to update it. According to technological survey, the CAD models provided by modern RE CAD systems (i.e. *CATIA V5*, *PRO-Engineer*, *RapidForm*, *GeoMagic*, etc.) are more geometrical models than real CAD models, because the algorithms are designed for general engineering applications and they usually reconstruct a set of free-form surfaces or primitive features. For complex objects, these do not guarantee the required level of the output accuracy for specific applications. Therefore, developing application specific RE techniques is essential, because it offers several possibilities to increase the success and accuracy of the RE process. These possibilities include the selection of a specifically suitable RE process from the start (i.e. data acquisition system) to the end (i.e. exporting a consistent and quality CAD model) with the help of prior knowledge. Hence the development of application specific (e.g. ship hull form) RE processes helps to answer the following questions. The questions aim to develop a suitable direct surface fitting approach for the ship hull form from an unorganized point cloud.

- What measurement techniques are suitable?
- What point cloud pre-processing should be performed?
- What best practice should be followed to achieve the desired level of accuracy?
- How to generate a good initial parameterization?
- How to get a better shape parameter estimation?
- How to efficiently achieve adaptive surface fitting?
- How to extract features in a very noisy data?
- How to automatically allocate additional control points in a highly curved feature regions to better represent the features?
- How to enforce knowledge relationships into data fitting?
- What criteria should be set in order to merge neighboring patches?

Answering the above basic questions requires the knowledge about important features of the object (hull form), the data and the algorithm that is going to be used. This will increase the success of RE processes and thereby its output quality. In this chapter, the developed ship hull form specific direct surface fitting to an unorganized point cloud is presented.

4.2 Related Works

Many surface reconstruction algorithms that fit a point cloud to various type of surfaces have been proposed over the last couple of decades. Some of them are quadratic surface fitting, B-Spline surface fitting, lofted surface fitting, and sweep surface fitting methods [97, 120, 147, 148]. However, Bézier and B-Spline patches are the two most popular and widely used parametric surface representations in the field of computer graphics, geometric modeling, CAD/CAM and RE, because of their elegant geometric properties. Comparing the surface models generated by B-spline patches and Bézier patches, there are two inevitable defects of exploiting Bézier patches to reconstruct smooth parametric surface models (C^2 – *continuity*). One is the difficulty to use Bézier patches to achieve the same approximation accuracy compare to B-spline patches if

the surface layout is given and the degree of the parametric surfaces is fixed. The other is that the global smoothness and fairness of the surface in Bézier form is worse than in NURBS form, this attributes to the requirement of a significant number of constraints imposed on the control points of Bézier patches to a piece of patches together in a continuous composite surface (G^2 – continuity). The tensor product B-Spline surfaces provide continuity without the imposition of constraints in the surface fitting process [149]. Due to these reasons, B-Spline surfaces, especially non-uniform rational B-spline (NURBS) surfaces, are popular.

Several NURBS based surface reconstruction approaches from an unorganized point cloud are proposed over the last years. However, their reconstruction process efficiency, their output accuracy, smoothness and continuity are still unsatisfactory and require further research. Most NURBS surface reconstruction approaches use mainly two pre-processing steps: parameterization and regularization.

Parameterization includes polygonization of an unorganized point cloud, so that the NURBS surface can be fitted to it later. The approaches based on this method are known to be computationally expensive in terms of the computer memory and processing time. Furthermore, they have problems to maintain a required level of continuity between surface patches. Eck and Hoppe [163] first reconstruct a triangular mesh from an unorganized point cloud and then the neighboring triangles are merged to form a quadrilateral mesh on which a network of B-Spline patches are built. To achieve a user-specified error tolerance, the quadrilateral meshes are refined on which new B-Spline patches are fitted. Even though it is an effective approach to reconstruct B-Spline surfaces from an unorganized point cloud; it is computationally expensive and guarantees only G^1 continuity between the patches. Krishnamurthy and Levoy [93] also first construct a polygon mesh from an unorganized point cloud and re-sample the mesh to a regularized grid on which NURBS surface patches can be fitted. The method is unable to handle surfaces with holes and has a poor performance for complex shapes. Park et al.[145] present a methodology to cluster the objects into regions on which polygon meshes are constructed and subdivided into triangular meshes, and blended into quadrilateral meshes on which NURBS surface patch networks are fitted. The method assumes that the point cloud represents a closed surface and does not preserve holes. It shares similar disadvantages with Eck and Hoppe’s approach. Gregorski et al.[92] present an approach that subdivides the set of points into a strip tree structure. This structure is used to fit quadratic surfaces and blended into B-Spline patches. It is unable to handle surfaces running close to each other and occlude surfaces. Most of the approaches attempt to reconstruct NURBS patches, however there are also some contributions to the reconstruction of triangular B-Spline patches from an unorganized point cloud [94, 164].

The regularization omits the parameterization (i.e. polygonization and NURBS patches networks) and regularizes an unorganized point cloud. Basically NURBS surfaces can not be directly fitted to an unorganized and scattered set of points and the representation of sharp features like knuckles, corners, and high curvatures is poor. Therefore, an unorganized point cloud should be regularized to fit NURBS surface to the point cloud. The regularization is usually performed by reducing the dimensionality (i.e. projecting onto the regression plane of the cloud) of the point cloud [96, 97]. The missing data and varying densities can be managed by inserting and removing points during regularization. This approach is simple to implement and produces a low fitting errors but is not efficient for occlude surfaces. In this work a regularization approach is implemented to reconstruct a NURBS hull form surface from an unorganized point cloud.

4.3 NURBS Hull Form Surface Fitting to an Unorganized Point Cloud

NURBS is the dominant mathematical technique used to design complex surfaces such as a ship hull form, because of its high level continuity, fairness and flexibility. In spite of this, there are many influencing variables one has to know to get the most out of the NURBS hull design. It is usually defined by a rectangular grid of vertex or control points, as a result a direct fitting to an unorganized point cloud is impossible. Therefore an unorganized point cloud should be first arranged into rows and columns.

4.3.1 NURBS Surface Formulation

NURBS is a mathematical surface representation and includes a controlling variables therefore it is important to discuss the controlling variables before proceeding to the surface fitting to unorganized points. Recalling the NURBS formulation discussed in chapter 2.

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{ij} P_{ij}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{ij}} \quad (4.1)$$

where $0 \leq u, v \leq 1$ and P_{ij} form a bidirectional control net, the w_{ij} are the weights, and the $N_{i,p}(u)$ and $N_{j,q}(v)$ are non-rational B-Spline basis functions defined on the knot vectors.

$$\begin{aligned} U &= \{\underbrace{0 \dots 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1 \dots 1}_{p+1}\} \\ V &= \{\underbrace{0 \dots 0}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{1 \dots 1}_{q+1}\} \end{aligned} \quad (4.2)$$

where $r = n + p + 1$ and $s = m + q + 1$.

Introducing the piecewise rational basis functions,

$$R_{i,j}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{ij}}{\sum_{k=0}^n \sum_{l=0}^m N_{k,p}(u) N_{l,q}(v) w_{kl}} \quad (4.3)$$

then the NURBS surface equation can be written as

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) P_{ij} \quad (4.4)$$

The designer has four basic variables to control the shape (i.e. stiffness and continuity) of the surface represented by NURBS: Control points, degrees, knots and weights.

Control Points: A NURBS Surface is defined by a regular grid of vertex or control points which do not necessarily lie on the surface. Changing the position of one of a control point deforms the local area of the surface around the control point like it is attached to the control point by a spring. As shown in Figure 4.1 (a), the curve is fitted to five vertices (D_1, D_2, D_3, D_4, D_5) with the corresponding control points (P_1, P_2, P_3, P_4, P_5). Changing the position of a single control point from P_3 to P'_3 deforms the curve as shown in Figure 4.1 (a). The surface does not move as much as the control point moves and the change only affects a small area near the moved control point. The influence area depends on the degree of the surface, a smaller degree means a

smaller area of influence and a larger degree means a larger area of influence. For every control point floating in a space there is an associated vertex point which lies on the surface. There is also a possibility to control the shape of the surface using these vertex points instead of the control point. This can be implemented as a local or global control as shown in Figure 4.1 (b) and (c). A local surface control using a point on the surface affects only the small area of the surface surrounding the point and the effect is similar to moving the associated control point, see Figure 4.1 (b) as D_3 moves to D'_3 . It can also be implemented in such a way that it could have a global effect, that means the movement of one of the point on the surface, moves none of the other points on the surface, see Figure 4.1 (c) as D_3 moves to D'_3 . Control points and points on the surface can be repositioned anywhere in the space to control the shape of the surface even though it is challenging to get the surface to be positioned exactly where one wants.

Degree: A NURBS curve or surface is represented mathematically by a piecewise polynomial blending functions of the form $(a + bx + cx^2 + dx^3 + \dots)$. The highest exponent used in the polynomial is the degree of the polynomial. Curves use one type of polynomial and NURBS surfaces use two different polynomial types one for each grid directions – row and column. The higher the degree of the polynomial, the more flexibility the curve and surface however the more wiggles they might be.

Knot Vectors: NURBS curves and surfaces are polynomials pieced together using knot vectors. Curves use one knot vector and surfaces use two in each local directions. The knot vectors define how the polynomial pieces are blended together with the proper smoothness. Therefore it alters the shape and fairness of the surface. There are generally two kind of knot vector definitions: uniform (i.e. with constant spacing between the knots) and non-uniform (i.e. with varying spacing between knot vectors). A uniform knot vector is the most common technique and works very well in most situations. The non-uniform knot vector definition could be used to control the tension or knuckles over the surface, but it may cause problem with adjoining surfaces. Knot vectors have global effects on the shape of the surface because adjusting the spacing in a row direction changes the surface on a full row length and affect also several columns. If there are neighboring surfaces at the end of the rows, then it must use the same knot spacing to maintain exactly the same edge shape. Therefore, they do have not only global effect on the surface shape itself but also force to change the adjoining surfaces, that is the reason why mostly they are kept uniformly spaced.

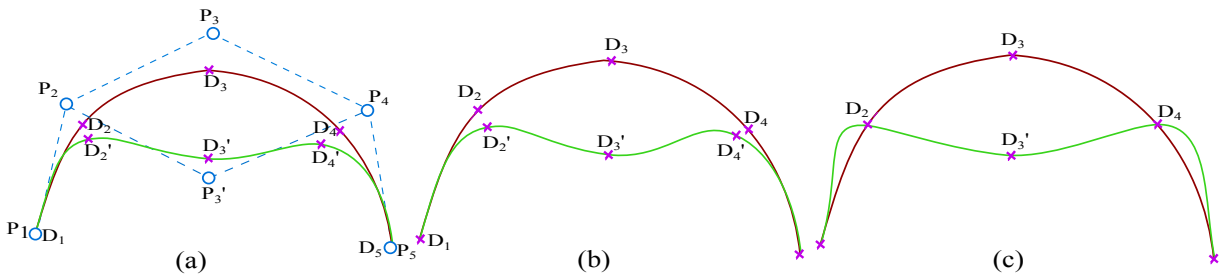


Figure 4.1: The effect of changing control points (a) and points on surface (b, c) position.

Weights: Each points on the surface or control points has an associated weight value that affects the shape of the surface locally near that point. The weight value is the result of the rational aspect of the NURBS, which means that the equation is defined as a fraction or ratio of polynomials. The purpose of the weights is to allow for an exact representation of conic shapes and to add more user control over the shape of the surface. In practice changing a vertex weight value can cause more problems on fairing. The best approach is to apply weight changes only near the end of the design process to achieve very specific local effects. Figure 2.13 shows how the weights affect the local shape of the surface.

If the NURBS representation uses uniform knot vector spacing and fixes the weights to unity, then it is a normal uniform B-Spline. In some CAD systems, even though the knot vectors are kept uniformly spaced and weights are set to unity, still they use NURBS representation. The reason is that NURBS representation is a superset of many different spline techniques and it has a great advantage in exchanging surface geometry between CAD systems.

4.3.2 NURBS Surface Fitting

To approximate an unorganized point cloud with NURBS surface, the points should be organized into a rectangular matrix. Having an unorganized point cloud regularized into a rectangular $(m+1) \times (n+1)$ grid points. The rectangular grid points (D_{ij}) where $(0 \leq i \leq m$ and $0 \leq j \leq n)$ and degrees (p, q) defined in u - and v -directions and the control points (P_{ij}) defined over $(e+1) \times (f+1)$ where $(0 \leq i \leq e$ and $0 \leq j \leq f)$, where input value should satisfy $m > e \geq p \geq 1$ and $n > f \geq q \geq 1$. Then it is possible to approximate the grid by B-Spline surface. Having the grid of points (D_{ij}) known with respective degree (p, q) in u - and v -direction. The next step is the determination of the parameters $(s$ in $u)$ and $(t$ in $v)$ direction. There could be infinite possible parameters and selection possibilities. The parameters selection greatly affect the shape and parameterization of the surface. There are some usually used selection methods such as uniformly spaced, chord length, centripetal and universal. Uniformly spaced parameter selection creates evenly spaced parameters. For instance if the parametric domain is $[a, b]$ in u -direction, then the parameter ends are $s_0 = a$ and $s_n = b$, then the parameters can be determined as follow.

$$\begin{aligned} s_0 &= a \\ s_i &= a + i \times \frac{(b-a)}{n} \quad \text{for } 1 \leq i \leq n-1 \\ s_n &= b \end{aligned} \quad (4.5)$$

The same applies to the v -direction.

Having $(m+1)$ parameters in u -direction and $(n+1)$ parameters in v -direction with corresponding degrees (p, q) . The spline surface knot vectors can be determined, for instance in u -direction, there are $h+1$ knots, where $h = m + p + 1$, and in v -direction there are $k = n + q + 1$. For clamped uniformly spaced knot vectors it could be determined as:

$$\begin{aligned} u_0 &= u_1 = \dots = u_p = a \\ u_{j+p} &= \frac{j}{(h-p+1)} \quad \text{for } j = 1, 2, \dots, h-p \\ u_{h-p} &= u_{m-p+1} = \dots = u_m = b \end{aligned} \quad (4.6)$$

There are other knot vector determination methods similar to parameter selection methods available in any parametric surface modeling books.

Once the parameters and knot vectors are determined, the points on the surface that corresponds to the data point D_{cd} could be determined as:

$$S(s_c, t_d) = \frac{\sum_{i=0}^e \sum_{j=0}^f N_{i,p}(s_c) N_{j,q}(t_d) w_{ij} P_{ij}}{\sum_{i=0}^e \sum_{j=0}^f N_{i,p}(s_c) N_{j,q}(t_d) w_{ij}} \quad (4.7)$$

The squared error distance between D_{cd} and its corresponding point on the surface is

$$|D_{cd} - S(s_c, t_d)|^2 \quad (4.8)$$

The sum of all squared error distances is therefore the following:

$$f(P_{00}, P_{01}, \dots, P_{ef}) = \sum_{c=0}^m \sum_{d=0}^n |D_{cd} - S(s_c, t_d)|^2 \quad (4.9)$$

This is a function with $(e+1) \times (f+1)$ unknown control points P_{ij} 's. To minimize the function $f()$ and determine the result, the partial derivatives could be computed and set to zero:

$$\frac{\partial f}{\partial P_{ij}} = 0 \quad (4.10)$$

This leads to a system of $(e+1) \times (f+1)$ equations whose common zeros are the desired control points. Unfortunately these equations are non linear and solving a system of non-linear equations is time consuming. Instead of aiming for an optimal solution, a non-optimal, but reasonably good solution is usually applied as shown in Algorithm 2.

4.3.3 Advantages and Disadvantages

Due to its ability to represent complex shapes, NURBS is preferred for any smooth, and a combination of planes and curved surfaces such as ship hull forms, car bodies, plane bodies, etc. However the biggest drawback of the NURBS modeling is its inability to represent complex shapes with a single surface. It rather represents with multiple surface patches blended together. Multiple surfaces usually fail to achieve the desired level of continuity and working with it needs more time. Some NURBS advantages and disadvantage are [165]:

Advantages:

- They are more general than Bezier and B-Spline curves and tensor product surfaces.
- Evaluation is straightforward, fast and computationally stable.
- They offer a common mathematical representation for free-form surfaces and commonly used analytical shapes such as natural quadrics, torii, extruded surfaces, and surface of revolution.
- They are affine (i.e. rotation, scaling and translation), parallel and prospectively invariant as well as invariant under shear transformations.
- It is easy to change their shape through the manipulation of control points, weights and knots.
- Degree elevation, splitting, knot insertion and deletion, and knot refinement offer a wide range of tools to design and analyze shape information.

Disadvantages:

- Bad choice of weights can lead to bad curve/surface parameterization.
- More storage is needed to define traditional shapes, such as a circle or a sphere.
- Some algorithms are numerically unstable e.g. inverse point mapping.
- Some algorithms work better with other forms than with NURBS, e.g. computing the intersection curve of two surfaces.

- Point member classification is a difficult problem for parametric surfaces. Therefore, it is particularly difficult to include NURBS as nodes in a constructive solid geometry system.

4.4 Methodology

The developed methodology considers an unorganized point cloud as an input and directly fits a NURBS surface without the need of prior parameterization. The ship hull form specific direct NURBS surface reconstruction methodology consists of three major modules: pre-processing, surface fitting and post-processing. The pre-processing module includes outliers removal, noise reduction, down-sampling, segmentation, regularization and boundary extraction as shown in Figure 4.2. The fitting module fits hull form surface to the regularized point cloud. Because of NURBS rectangular nature, the surface which is not a part of the target object's surface should be eliminated, to do so, the boundary points are detected to create a trimming curve. The boundary points are fitted with a B-Spline curve. At the stage of the post-processing module the additional surfaces are trimmed using the boundary curve and then the final surface will be exported to an IGES file format as shown in Figure 4.2.

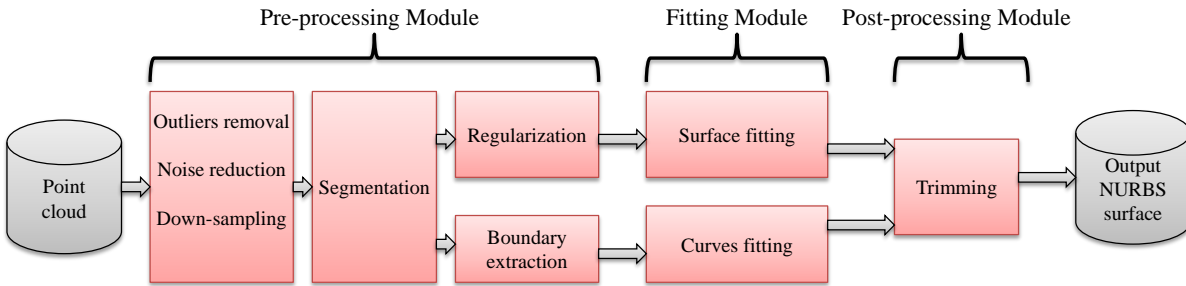


Figure 4.2: Direct surface fitting to a point cloud process flow.

4.4.1 Pre-Processing Module

The pre-processing implemented in this methodology assumes that the point cloud is transformed into global coordinate (registered). Therefore, it consists of the algorithms used to pre-process an unorganized point cloud for direct NURBS surface fitting. It is equipped with outliers removal, noise reduction, down-sampling, segmentation, regularization, and boundary extraction algorithms. For outliers removal, the statistical; for noise reduction and down-sampling the voxel grid approaches are incorporated. The working principle of the statistical and the voxel grid approaches are described in chapter 2.

4.4.1.1 Hull Segmentation

Representing a hull form with a single NURBS surface is challenging. As a consequence several NURBS patches are stitched together to represent a complex object. Hence hull form point cloud segmentation algorithms are incorporated in the framework. It divides the hull form point cloud into simple regions so that it is easy to fit the NURBS surfaces. To do so, three alternative automatic hull form segmentation methods are integrated. The first approach divides the hull form into five regions using section planes, as shown in Figure 4.3(a). The second approach first divides the hull form into wet and dry regions, and then divides with perpendicular planes which in total creates nine simple surface regions as shown in Figure 4.3(b). The third approach first

detects the flat of bottom, flat of side and transom, and then separates from the whole hull form. The rest of the surface is divided into four regions which in total results in seven simple regions as shown in Figure 4.3(c).

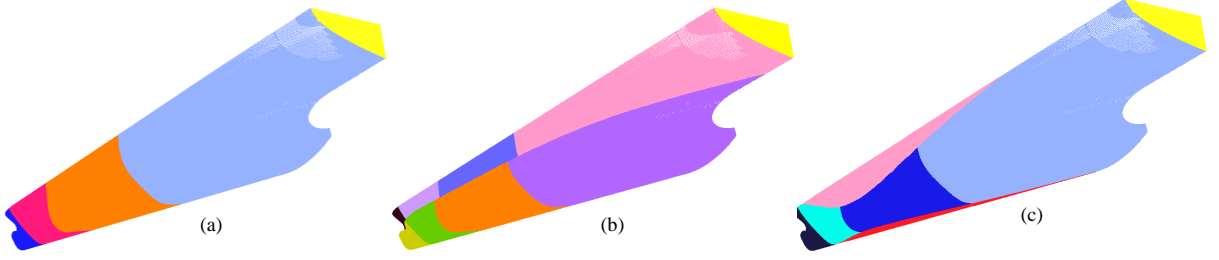


Figure 4.3: Segmentation of a hull form into simple surfaces.

4.4.1.2 Regularization

Having the hull form point cloud segmented into simple regions, the next step regularizes an unorganized point cloud to a regular rectangular matrix. Two approaches are implemented to regularize the points. The first approach projects the points on a plane on which the regularization process is carried out. For instance, as shown in Figure 4.4(a), the point cloud is projected on xz plane on which the regularization is performed using the average of the points which fall into pixel size $(x_{i+1} - x_i, z_{i+1} - z_i)$. However the boundary points are preserved, that means averaging does not apply to the boundary points. This method works well in most cases, but has a difficulty to evenly regularize in case the surface composes perpendicular surfaces to the plane as shown in Figure 4.4(b). An unevenly regularized point cloud has a negative effect on the stiffness of the surface. As a consequence a second and better regularization strategy is implemented. It takes a strip of points as shown in Figure 4.4(c) and determines the centroid of the points that fall into the strip. Afterwards, the point cloud is regularized based on the angle and the strip size $(x_{i+1} - x_i)$ for the case shown on the figure). This method offers a better regularization as shown in Figure 4.4(d).

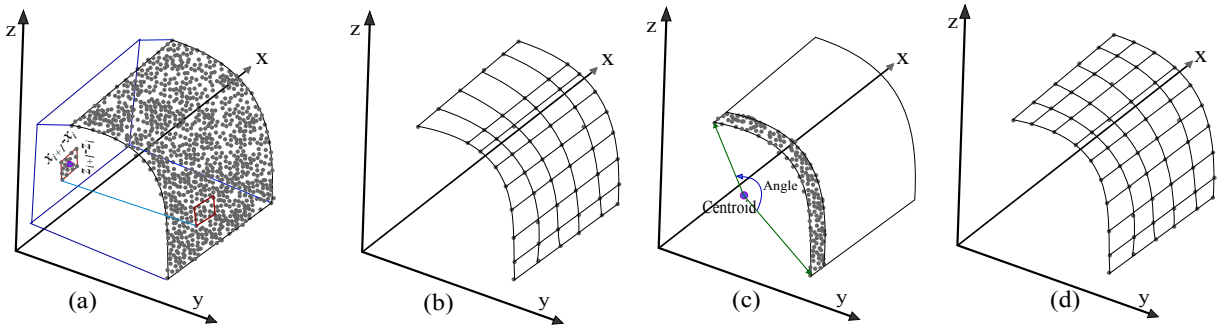


Figure 4.4: An unorganized point cloud regularization type and working principle.

4.4.1.3 Boundary Extraction

After fitting the NURBS surface, the regions of the surface which are not part of the target object should be removed. As a result the boundary of the original point cloud should be extracted. The boundary points can be determined by analyzing the angles they form with their neighbors. First the neighboring points are determined for a query point (P_q) as shown in Figure 4.5(a) and projected to a local plane. The local plane is determined using the normal of the query point.

On the plane, the angles created between neighboring points and the query point are determined. For instance the angle $(\angle P_i P_q P_n)$ as shown in Figure 4.5(b). Let $\beta_i = (\beta_1, \dots, \beta_n)$ be the list of sorted angles between the lines formed by two points P_i and P_{i+1} in the neighborhood with the query point P_q . Then P_q is a point located on the boundary of a surface, if:

$$\max(\beta_i) \geq \gamma_{th} \quad (4.11)$$

where γ_{th} is a maximum angle threshold. The value $\Pi/2$ is found to be good estimation for most cases and it is set to $\Pi/2$ for all test cases in this thesis.

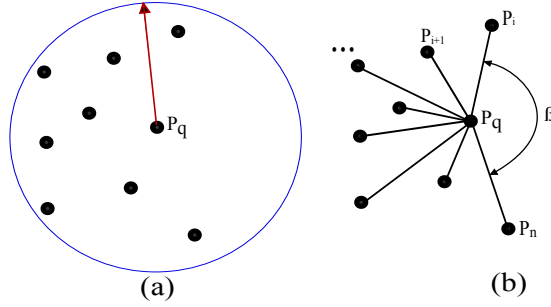


Figure 4.5: Extraction of boundary points using the angle criterion, a) determination of neighboring points and normal and b) determination of the angle between the query point and neighboring points to extract boundary points.

4.4.2 Surface Fitting Module

The surface fitting is the most important element of the developed reconstruction framework. Once the points are regularized into a rectangular grid (u, v) then the NURBS surface is approximated. The approximation process allows the user to control the degree, the tolerance, and the continuity of the surface. The degrees in u -direction p and in v -direction q are by default set to degree three. The tolerance (i.e. the maximum possible deviation of the point on the surface and the real point) is set by the user. When the tolerance value approaches to zero then the surface approximation is similar to surface interpolation. The continuity is set to (C^2) in the surface. The NURBS surface with unity weight is automatically fitted to the organized point cloud as shown in Algorithm 2. The surface approximation is implemented based on the OCCT class *GeomAPI_PointsToBSplineSurface()*.

The surface parts added during regularization should be removed. For this reason the boundary points are extracted as shown in Figure 4.6 (b). The extracted boundary points are interpolated with B-Spline curves. The NURBS surface and B-Spline curves are fitted as shown in Figure 4.6 (c).

4.4.3 Post-Processing Module

Once the NURBS surface is fitted to the regularized point cloud, the surface which is not part of the intended surface should be trimmed using the boundary B-Spline curve. Due to pre-processing, surface and curve fitting approximation natures, there are often gaps between the surface and the boundary curve. These gaps usually prevent the trimming process. In that case a curve pulling towards the surface is used. The NURBS surface and boundary curves are exported to IGES file format and the trimming process is performed in a CAD system – Rhinoceros [160].

Algorithm 2: Surface fitting to a regularized point cloud.

input : $(m + 1) \times (n + 1)$ data points D_{ij} , degree (p, q) , continuity and tolerance.

output: A NURBS surface with unity weights (B-Spline surface) of degrees (p, q) that approximates the data points.

Algorithm:

Compute parameters in the u-direction $s_0, s_1, s_2, \dots, s_m$ and the knot vector U ;

Compute parameters in the v-direction $t_0, t_1, t_2, \dots, t_n$ and the knot vector V ;

*/*computing “intermediate data points” Q ’s*/ ;*

for $i := 0$ **to** m **do**

/ for column i of D’s */ ;*

Apply curve approximation to column i of the points (i.e., $D_{0i}, D_{1i}, \dots, D_{mi}$);

using ;

degree p ;

parameters s_0, s_1, \dots, s_m ;

knot vector U ;

The result is column i of the “intermediate data points” $Q_{0i}, Q_{1i}, \dots, Q_{ei}$;

/ Q ’s form a $(e + 1) \times (n + 1)$ matrix available*/ ;*

*/*computing the desired control points P ’s */ ;*

for $j := 0$ **to** n **do**

/ for row j of Q’s */ ;*

Apply curve approximation to row j of the Q ’s (i.e., $Q_{j0}, Q_{j1}, \dots, Q_{jn}$);

using ;

degree q ;

parameters t_0, t_1, \dots, t_m ;

knot vector V ;

The result is row j of the desired control points $P_{j0}, P_{j1}, \dots, P_{jf}$;

/ P ’s form a $(e + 1) \times (f + 1)$ matrix available*/ ;*

The computed $(e + 1) \times (f + 1)$ control points P_{ij} , degree (p, q) , and knot vectors U and V define an approximated B-spline surface.

4.5 Results and Discussions

Modern data acquisition devices and scanners are able to generate a very large point cloud which helps to reconstruct even geometrically complicated objects. Due to its large size, a ship hull form data acquisition is typically performed from multiple scan viewpoints. As a result, the acquired massive unorganized point clouds are not suitable to be integrated into CAD systems. Therefore, several consecutive processes should be performed to make the acquired data usable for further processing in CAD systems or downstream applications. The developed framework considers registered unorganized point cloud as an input. The framework is equipped with outliers removal, noise reduction and down-sampling algorithms so that the successive operations can be performed successfully. Having the pre-processed unorganized point cloud, the hull form is segmented into simple regions as fitting a complete hull form with a single NURBS surface is challenging. At this stage, the user has three alternative segmentation strategies which decompose the hull form into different regions. Once the hull form is decomposed into desired regions, then each regions is fitted with a single NURBS surface which are later stitched together. In order to fit NURBS surface to unorganized points, the point cloud should be regularized into a four sided array. The regularization is performed automatically having the number of rows and columns, and a pixel

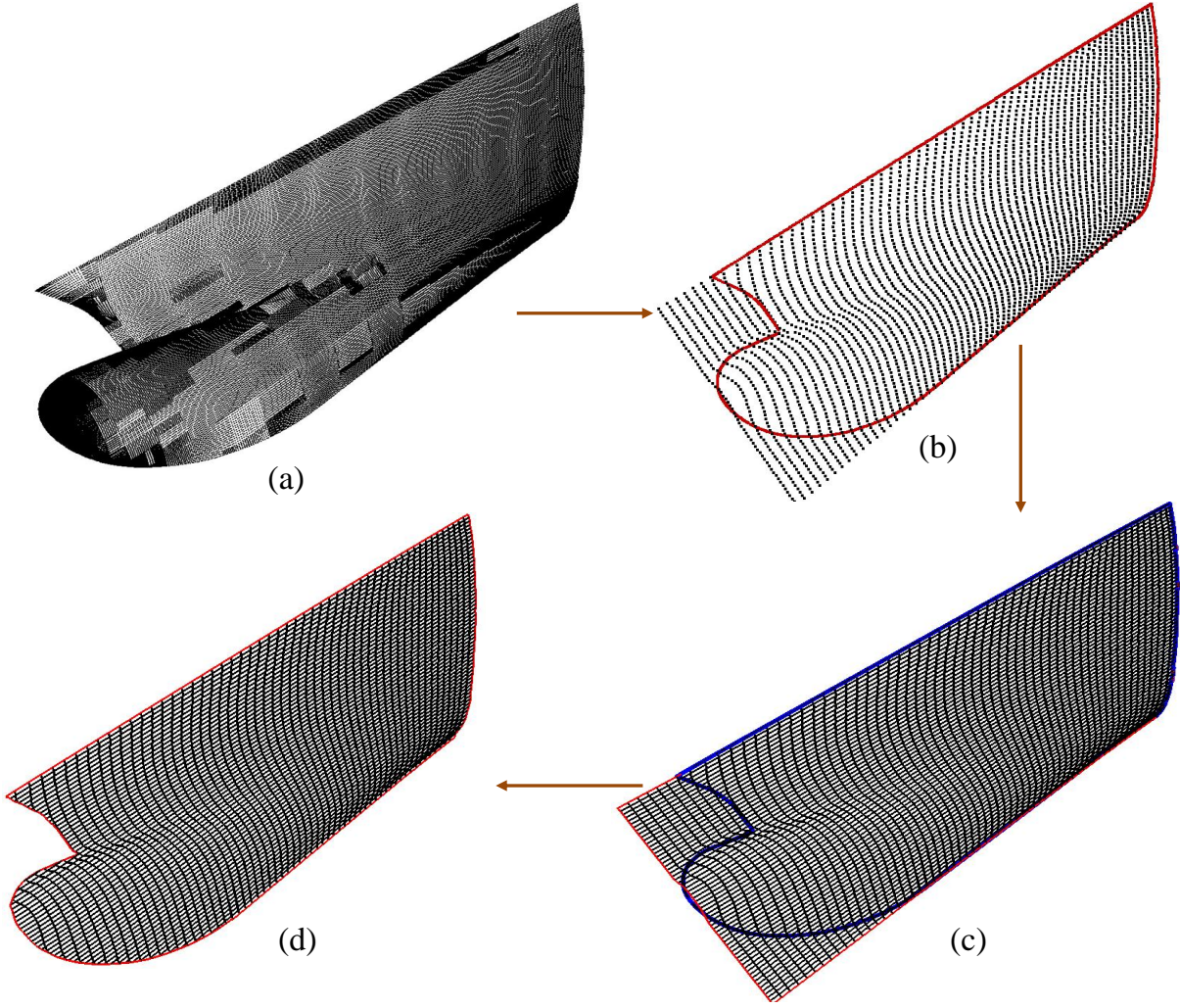


Figure 4.6: The direct surface fitting process. a) unorganized point cloud, b) regularized points and extracted boundary points, c) the fitted surface and boundary curves, d) the trimmed surface.

size defined. Then, the regularized point cloud is approximated with a NURBS surface. The general algorithm layout of the developed framework is shown in Algorithm 3.

Despite good achievements, the methodology needs improvements. Especially the topological problems between surface patches as shown in Figure 4.7, the continuity between surfaces from different regions, the different features (e.g. knuckles), etc should be addressed. The fitting process should be optimized for a high curvature areas and sharp features using a weighted fitting process. In addition, the segmentation based on a point cloud mostly results in zigzag boundaries which is problematic for the boundary extraction process.

4.6 Summary

In engineering companies (i.e. marine, car, aircraft, etc) NURBS surface representation has a particular interest because of its elegant and precise modeling control, descriptive flexibility (i.e. ability to construct conics), and a modeling logic that is in tune with redesign, retrofit, and fabrication considerations in general. In addition, it can be saved to a powerful data formats such as IGES, STEP, etc which are capable to convey considerable object's information when

Algorithm 3: A ship hull form NURBS surface from an unorganized point cloud, a general algorithm.

input : Unorganized point cloud (Hull form)

output: NURBS represented hull form in IGES file format

Algorithm:

input : Unorganized point cloud (Hull form), number of neighboring points, standard deviation multiplier

Remove outliers;

input : Processed unorganized point cloud, leaf size

Reduce noise and Down-sampling;

/*Pre-processed unorganized point cloud available*/;

input : Pre-processed unorganized point cloud, type of segmentation, plane detection iteration, distance threshold and number of divisions

Segment the hull form into desired parts;

/*Segmented point cloud available*/;

input : Segmented point cloud

for $i := 0$ **to** R_p **do**

/* R_p number of segmented regions***/** ;

input : Number of rows and columns, pixel size

Regularize points into u and v direction;

/*Regularized point cloud available*/;

input : Angle, radius for normal estimation

Extract the point cloud boundary;

/*Boundary points available*/;

input : Boundary points

Fit the B-Spline curves;

input : Regularized points, continuity, degree (p, q), and tolerance

Fit the NURBS surface;

/*B-Spline surface available*/ and /*B-Spline curves available*/;

Trim B-Spline surface using B-Spline curve;

/*Trimmed surfaces available as an IGES file format*/;

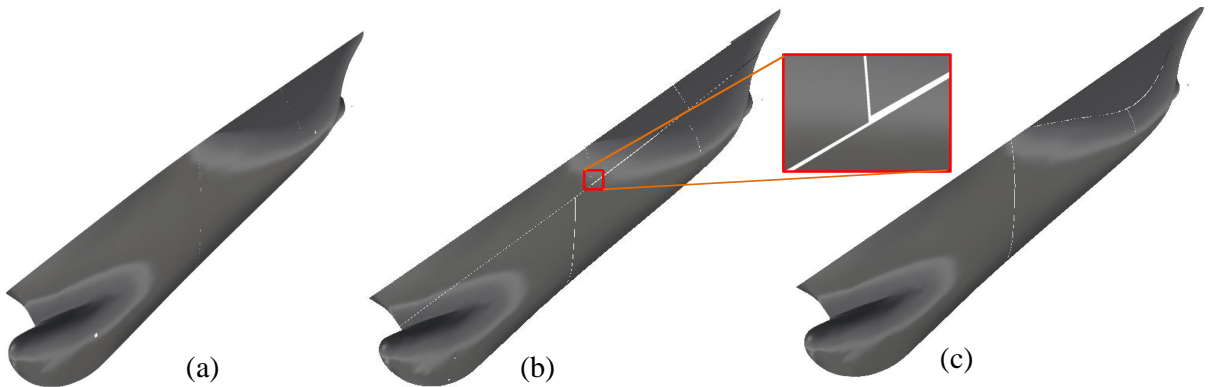


Figure 4.7: A reconstructed hull form after segmented using different segmentation methods. a) simple transversal cutting plane segmented, b) segmented to wet and dry region in addition to the transversal sectioning, c) includes the flat of bottom and the flat of side detection.

data exchange and documentation is required. Consequently, CAD systems adopted the NURBS representation which plays a great role in a complex CAD model design. However, existing automated approaches to fit the NURBS surface to an unorganized point cloud are still limited. They either fail to account for all the topological variations (i.e open or closed) or struggle with imperfections in the input data (i.e. incomplete and noisy). In this chapter, a ship hull form specific RE approach which fits NURBS surface to an unorganized point cloud in the presence of noise and incomplete data is presented. The framework consists of automatic point cloud pre-processing, surface fitting and post-processing modules with a minimum number of user interaction. The pre-processing module prepares the point cloud so that the NURBS surface can be directly approximated based on the point cloud. It includes outliers removal, noise reduction, down-sampling, segmentation, regularization and boundary estimation. The surfaces and curves are fitted to the regularized and boundary points in the second module respectively. The boundary curve is used to trim the surface added as a result of the points added at the regularization step. In general the approach fits NURBS surfaces directly to the point cloud without prior construction of either polygon meshes or NURBS patch networks. As a result, the proposed strategy is computationally efficient. In addition, the segmentation based primitives preserves some hull form features however, it does not preserve features such as knuckles and holes to the required level. The other limitations are smoothness, and continuity between surface patches which require further work. The solution to the topological problem is presented in chapter 6, but the smoothness issue needs improvement in the developed framework. With improvement in feature preservation capability and the smoothness issue, the framework can be a good choice for NURBS hull form surface reconstruction.

CHAPTER 5

TRIANGULATED SURFACE HULL FORM REVERSE ENGINEERING

5.1 Introduction

The main applications of RE in maritime industry (i.e. retrofit, maintenance, redesign, inspection and historical archive) are briefly discussed in chapter 2. From these applications, redesign and retrofit processes usually require performance checking virtual laboratory techniques such as CFD and FEA. These techniques usually offer the functionality to perform the investigation based on triangulated surfaces or meshes. In this case, the reconstruction towards parametric surfaces and curves might be not necessary because of the fact that the process is in a sense of repeating itself and comes back to where it is started but with deviations or inaccuracies far in excess of the original scanned point cloud. Therefore, neither the curves network nor the direct surface fitting RE approaches are effective and efficient in this case, because parametric surface fitting to an unorganized point cloud usually involves many consecutive error-prone and lengthy RE processes. Therefore a consistent triangulated surface reconstruction decreases the possibility of object's feature loss and the number of required RE processes, which eventually makes the method efficient and easy to use for downstream applications directly.

The reconstruction of a consistent triangular surface from an unorganized point cloud offers the following major applications:

- it provides a rapid process chain by providing the design tools required to reconstruct and manipulate the scanned point cloud directly, transforming it into a mesh ready for hydrodynamic or finite element computation;
- eases the way towards parametric surface fitting;
- can be directly used for production purpose;
- can be used for visualization and documentation directly.

5.2 Related Works

Over the last two decades a vast variety of surface reconstruction approaches, techniques and methods are proposed as discussed in chapter 2. Due to a large variety of approaches, it is challenging for the end users to select a suitable approach for their specific application. Hence the research community in this field present the state-of-the-art on surface reconstruction approaches, providing categorization, comparison, effectiveness, capability and limitation of different methods and techniques. This provides a knowhow about the quality, capability, efficiency and suitability to specific cases of the reconstruction approaches. Berger et al. [8] categorize and provide a detailed characterization of the surface reconstruction, highlight similarities between a diverse reconstruction techniques, and extract the advantage and disadvantage of each technique with future possible improvement ideas. Chang [166] performs a survey of some reconstruction techniques towards what he means “key ingredient” in an excellent surface reconstruction algorithm,

which guarantee a continuous surface from a given unknown surface point cloud. These include the techniques robustness to different point cloud artifacts, capability to preserve sharp features and provable reconstruction guarantees. He compares various techniques and suggests the idea of combining the strength of different techniques for a better result. Berger et al. [167] present a benchmark for the evaluation and comparison of surface reconstruction algorithms, which takes an oriented point cloud as an input and produces an approximating surface as an output. They unveil the advantages and disadvantages of the existing approaches which help to find out potential further improvements. Lim et al. [73] also reveal that each technique has its own pros and cons, in dealing with point cloud artifacts and guaranteeing the required level of an output surface quality. In addition to an investigation for new methods, they also suggest extensive study of the existing approaches pros and cons, comparison of each method based on same input point cloud. Despite many efforts made to draw conclusive ideas from diversified reconstructions approaches, it is not an easy task because of the wide differences in the respective approaches' pros and cons. An approach might be robust to noise but does not preserve sharp edge, for others the reverse might hold true. Therefore, the existing approaches evaluation should be investigated for a specific application in order to make better approaches available for the end user.

In this chapter, different triangulated surface RE approaches are selected for evaluation with regards to a hull form reconstruction from an unorganized point cloud. It does not only evaluate the existing methods but also identifies the way approaches can perform even better. Based on the evaluation, a hull form specific triangular surface reconstruction approach is developed. The developed approach is categorized into three stages: point cloud pre-processing, triangular surface reconstruction, and post-processing. In addition hull form specific knowledge are integrated into different stages to improve the performance and accuracy of the output as shown in Figure 5.3.

5.3 Surface Reconstruction Evaluation

The diversity of surface reconstruction techniques ranges from methods that assume a well sampled point cloud to that make very loose assumptions on the quality of the point cloud; from methods that can be used to reconstruct arbitrary shapes to that restricted to specific classes of shapes; from methods that produce a watertight surface mesh to the methods that offer a non-mesh based surface representation, etc. However, this section evaluates the reconstruction methods which primarily offer triangulated surfaces (mesh) as an output. There are several surface reconstruction evaluation criteria in the literature. Berger et al. [8] consider different surface reconstruction evaluation techniques such as: geometric accuracy, topological accuracy, structural recovery, shape recognition, ease of use and reproducibility. Geometric accuracy commonly compares the geometry of the reconstructed output and the ground truth surface from which the scan was obtained. On the other hand topological accuracy deals with the recovery of the higher-level information of the shape and in particular, its topology. When dealing with large scale scenes, not only the object as whole is important but also its structures. Hence structural recovery is desirable during reconstruction specially for complex objects. In this situation the meaning of structure ranges from the dimension of geometric entities (i.e. manifolds, non-manifolds) to adjacency relationships through canonical geometric relationships (i.e. parallelism, co-planarity, orthogonality, concentricity, co-axiality) and regularities (i.e. repetitions, symmetries). Shape recognition deals with the quality of surface reconstruction to recognize the object's whole shape or parts of a shape from an incomplete point cloud. Ease to use as a quality measure is a common method to evaluate the effectiveness of an algorithm. It discloses whether the algorithm is automatic or not, whether it is too much sensible to an input parameter or not, whether it

requires few or many input parameters, etc. Reproducibility unveils the robustness of the reconstruction algorithm. Chang [166] also presents a high-level reconstruction evaluation criteria which includes the quality and capability of the underlying reconstruction algorithm.

5.3.1 Evaluation Criteria

In this thesis the evaluation methods presented in [8] and [166] are combined and categorized into three criteria: quality, capability, and usage and time.

Quality Criteria

Sharp: Does it preserve sharp features?

Smooth: Does it produce a smooth surface?

Fill holes: Does it fill holes or preserve topology?

Accuracy: Does it produce an approximating or interpolating surface?

Global/Local: Does it solve for the entire surface at once (global) or does it compute localized fits to the data?

Capability Criteria

Normals: Does it require normals or not?

Noisy Data: Is it robust to noisy data?

Large Data: Is it capable to manage a large data size?

Missing data: Is it capable to manage missing data?

Complex object: Is it capable to reconstruct complex object?

Reproducibility: Is it stable and reliable?

Post-processing: Does it require post-processing?

Usage and Time Criteria

Usage: Is it user friendly? Is it automatic or manual? How many input variables does it require?

EIapse time: How long it takes to reconstruct?

5.3.2 Evaluation and Improvements

An ideal surface reconstruction algorithm is expected to completely fulfill the quality, capability and usability criteria listed above. For instance the best approach: does not require normals, robust to noisy data, ables to manage missing and large data, is capable to reconstruct a complex shape, preserves sharp features while guaranteeing a topology and smoothness, produces both interpolating and approximating surfaces, reconstructs geometrically accurate model, and is user friendly and time saving. However, none of the reconstruction techniques proposed so far satisfies all the criteria.

Six triangular surface reconstruction approaches, which are implemented in the developed framework are evaluated against the above evaluation criteria. The methods include: alpha shapes, ball pivoting, greedy projection triangulation, grid projection, marching cubes, Poisson surface reconstruction. The method's surface representation strategy and main driving parameters are presented in Table 5.1. The first three methods (i.e. alpha shapes, ball pivoting and greedy projection triangulation) use explicit type surface representation while the last three methods (i.e. grid projection, marching cubes and Poisson) use implicit surface representation. Greedy projection triangulation and Poisson methods require relatively higher number of input parameters.

Table 5.1: Different triangular surface reconstruction types and their main input parameters

Methods	Type	Main parameters	Description
Alpha shapes	Explicit	α	The maximum magnitude of the Delaunay simplex
Ball pivoting	Explicit	ρ	The maximum magnitude of a pivoting ball radius
Greedy projection triangulation	Explicit	N ρ $\theta_{max}, \theta_{min}$ β_{max}	<p>The maximum number of the nearest neighbors which control the size of the neighborhood points.</p> <p>The maximum acceptable distance for a point to be considered relative to the distance of the nearest point (in order to adjust to changing densities).</p> <p>The maximum and minimum angles of each triangle.</p> <p>The maximum surface angle which deals with sharp edges and when two surfaces run close to each other.</p>
Grid projection	Implicit	N d_g p_s	<p>The maximum number of the nearest neighbors points.</p> <p>The resolution (the size of a grid cell)</p> <p>Padding size: which helps to fix holes smaller than the padding size.</p>
Marching cubes	Implicit	d_x, d_y, d_z r_p	<p>Resolution: the marching cubes grid resolution along x-axis, y-axis and z-axis respectively.</p> <p>Percentage: the percentage of an empty space between the bounding box and the grid limits.</p>
Poisson	Implicit	p_d p_s p_{sn} p_{sd} p_{id}	<p>Depth: the maximum depth of the tree that will be used for the surface reconstruction. A higher value means deeper the cubes and finer the voxel grid. It is only an upper bound since an octree adapts to the sampling density. The default value is 8.</p> <p>Scale: ratio between the diameter of the cube used for reconstruction and object's bounding cube.</p> <p>Sample per node: specifies the minimum number of sample points that should fall within an octree node as the octree construction is adapted to a sampling density. Depends on a noise level, for noisy point cloud (15 to 20) is recommended and for noise free point cloud (1 to 5) is generally recommended [11].</p> <p>Solver divide: the depth at which a block Gauss-Seidel solver is used to solve the Laplacian equation. This parameter helps to reduce the memory overhead at the cost of a small increase in construction time.</p> <p>Iso divide: the depth at which a block iso-surface extractor should be used to extract the iso surface. It helps to reduce the memory overhead at the cost of small increase in extraction time.</p>

Table 5.2: Evaluation of reconstruction algorithms against evaluation criteria

	Ideal	Alpha shapes	Ball pivoting	Greedy projection triangulation	Grid projection	Marching cubes	Poisson
Normals	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>
Noisy data	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Yes?</i>	<i>Yes</i>
Approx.	<i>Both</i>	<i>Inter.</i>	<i>Inter.</i>	<i>Inter.</i>	<i>Approx.</i>	<i>Approx.</i>	<i>Approx.</i>
Fill holes	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>
Smooth	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>
Sharp	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes?</i>	<i>Yes</i>	<i>No</i>	<i>No</i>
Guarantee	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>
Global/local	<i>Any</i>	<i>Global</i>	<i>Global</i>	<i>Local</i>	<i>Local</i>	<i>Global</i>	<i>Global</i>
Reproducibility	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>
Complex-object	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>
Usage	<i>Yes</i>	<i>Yes?</i>	<i>Yes?</i>	<i>No</i>	<i>No</i>	<i>Yes?</i>	<i>Yes?</i>
Post-processing	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>

All the selected methods are evaluated against various criteria as shown in Table 5.2. The evaluation criteria for an ideal surface reconstruction are listed in the first column of the table and the reconstruction algorithms capability, quality and usage are evaluated respectively. The evaluation with the question mark (Yes?) in the table shows that the method is capable but not satisfactory. The evaluation clearly reveals that every methods has its own advantages and disadvantages.

All methods are implemented and the ship hull form stern region is reconstructed with each method as shown in Figure 5.1. During the testing of the implemented algorithms, the following algorithm general behaviors are observed in addition to the evaluation results listed in Table 5.2.

The alpha shape is a simple concept algorithm and driven by a single variable. However, it is tough to set a suitable variable (alpha value) to get a watertight output surface. The method preserves sharp edges and boundaries well, but produces triangles with many flaws such as unoriented normals and non-manifold triangles (a color difference) as shown in Figure 5.1 (a).

The ball pivoting shares similar reconstruction characteristic with alpha shapes method but provides a consistent triangular surface. It also preserves sharp edges and boundaries as shown in Figure 5.1 (b). It is not robust enough to reconstruct complex shapes because, defining a too large value of the ball pivoting radius leads to the loss of occlude features, and defining a too small radius value leads to holes in the reconstructed surface.

The grid projection requires an oriented point cloud as an input and provides an approximated triangulated surface as an output. The output result is largely dependent on the resolution value

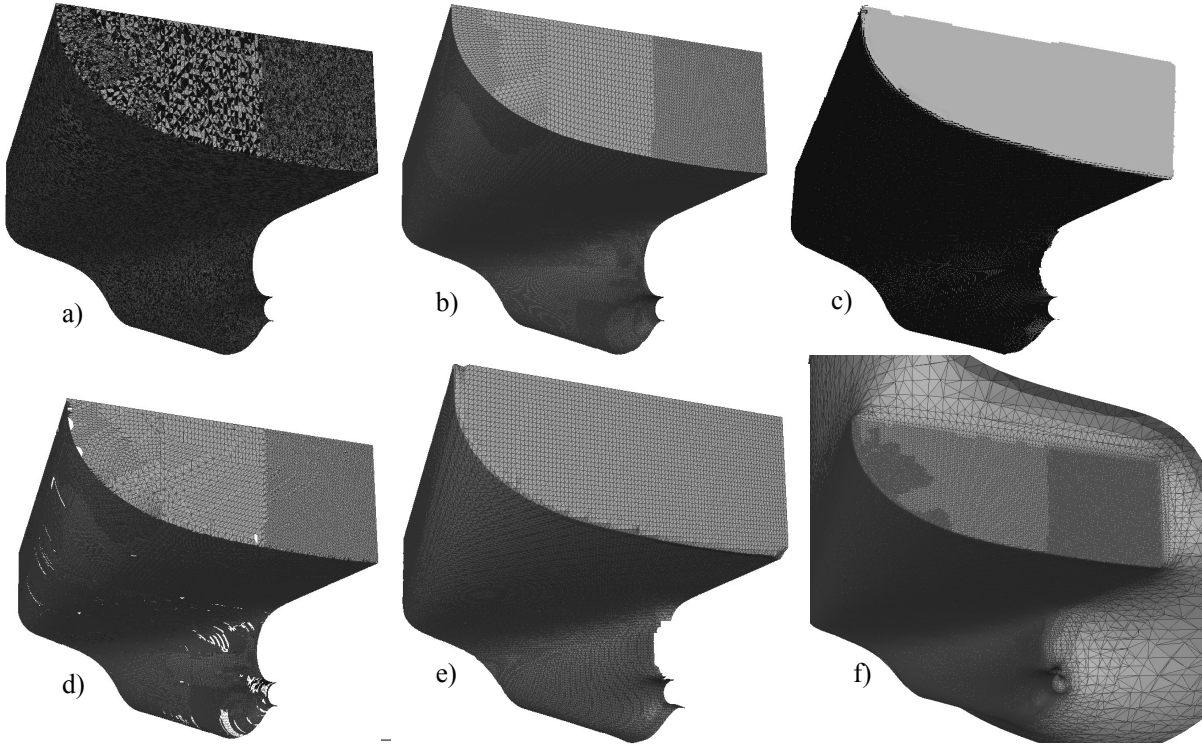


Figure 5.1: Ship hull form (aft part) surface reconstruction from a noise free point cloud. A comparison between six triangular surface reconstruction. a), b) and c) are reconstructed by alpha shapes, ball pivoting and grid projection methods respectively. Greedy projection triangulation, marching cubes and Poisson methods are used to reconstruct surfaces labeled by d), e) and f) respectively. The non-uniformity in color in a), b) and c) shows the lack of consistent surface normal orientation.

(the grid cell size). A too small resolution produces smooth and fine triangular mesh but needs high computer memory and time, a too large resolution value does not produce a smooth surface. It is very weak in preserving boundaries and sharp edges, and usually results in holes as shown in Figure 5.1 (c).

The greedy projection triangulation requires an oriented and non duplicated point cloud as an input. The method preserves sharp edges but is not robust to noise and usually produces holes as shown in Figure 5.1 (d).

The marching cubes algorithm is very sensitive to the normal orientation and also depends on the grid resolution. It fails to reconstruct a proper surface with an unoriented point cloud. The method preserves neither sharp edges nor boundary edges. Hence the method requires post-processing to extract boundary edges from the reconstructed surfaces. The result of the implemented algorithm is shown in Figure 5.1 (e).

The Poisson method is a powerful and robust reconstruction algorithm compared to the other implemented methods, however it is unable to preserve sharp and boundary edges. It is robust to noisy and large data but requires consistently oriented point cloud normals. The requirement of a consistent normal orientation makes the method not easy to use because obtaining a consistent orientation of a complex object is challenging by itself. The method requires post-processing steps to extract a proper boundary edges. As shown in Figure 5.1 (f) the method usually attempts to produce a closed shell and produces a surface which is not part of the intended object.

All the above evaluation of the methods are performed with a noise free hull form point cloud. To check the methods robustness to noisy data, two reconstruction methods one from an explicit

and the other from an implicit surface representation are considered: ball pivoting and Poisson methods. To do so, noisy data is introduced to the same point cloud used before. As shown in Figure 5.2 (left) ball pivoting is ineffective and inappropriate for noisy data. To the contrary, the Poisson method produces smooth surface, and it is suitable and robust to noisy data as shown in Figure 5.2 (right).

All reconstruction algorithms require pre-processing and post-processing steps in order to reconstruct usable surfaces from measured points. For instance, the first three methods generally require proper denoising algorithm and the last three methods require an oriented point cloud as an input, see Table 5.2. Marching cubes and Poisson methods require trimming algorithms in order to extract proper boundary edges. Except the Poisson method, the others require local hole filling. The alpha shapes method particularly produces triangulated surface with flaws which needs to be repaired in a post-processing step. In the following sections the developed triangulated surface RE approach will be discussed. The developed approach provides statistical, numerical and knowledge based solutions to the problems of the reconstruction methods.

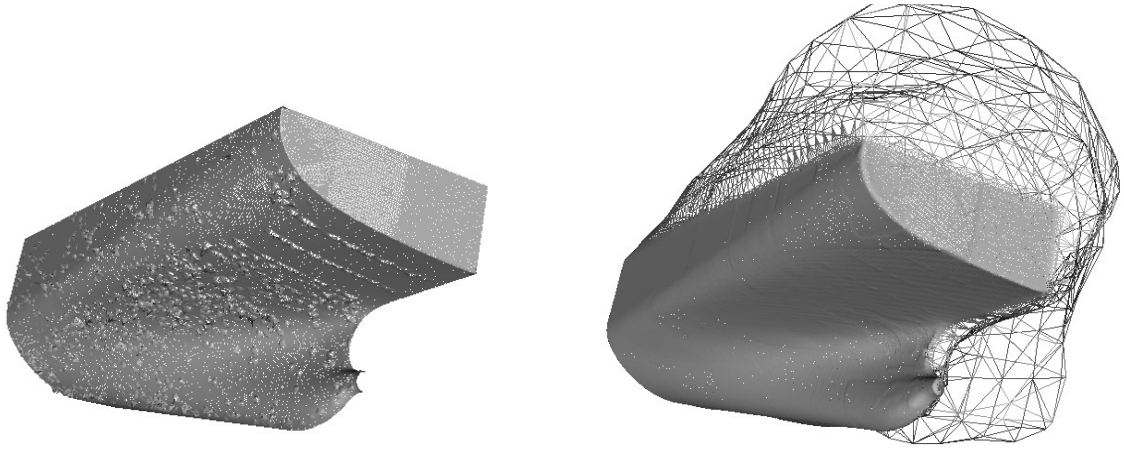


Figure 5.2: The reconstruction of triangular surface from a noisy point cloud. Ball pivoting reconstructed surface (left) and Poisson method reconstructed surface (right).

5.4 Hull Form Related Knowledge

The nature of the point cloud and the behavior of the algorithms are indeed the most driving elements in reconstruction processes. The nature of the point cloud is mainly a result of the data acquisition system capability, measurement method and object's complexity. The behaviors of the algorithms include the core approaches used for reconstruction, the nature and the number of determining input variables, the required input data type and the type of output surfaces. The increment in a complexity of the object usually makes the data acquisition systems challenging and is usually resulted in a less quality point cloud, which is challenging for reconstruction processes and indeed resulted in unreliable outputs. In general the success of a RE process is mainly dependent on the capability of the data acquisition technology, the reconstruction processes and the users knowledge. Over the last years the development in data acquisition systems and the reconstruction techniques lay down many options. Therefore, it is crucial to have the knowledge about the data acquisition systems and reconstruction techniques, to be able to select suitable approaches for specific applications. It is usually the users subjective responsibility to select suitable algorithms for any particular kind of object and/or point sets based on his/her own experience. This requires considerable knowledge about the property of available algorithms, the behavior of the point cloud, and the complexity of the object. Otherwise, the reconstruction

performed by a user with limited knowledge might certainly result in unreliable output. Hence, the availability of algorithm and data knowledge in general improves the quality of data-driven numerical reconstruction processes. In addition, the integration of scene and spatial knowledge could further enhance the quality and capability of the data-driven algorithms. Knowledge based RE is hardly suitable for a general case but it is rather effective when it comes to specific applications. Hence, application specific knowledge based reconstruction approaches are possible but what kind of knowledge and how to integrate it into the reconstruction processes is always an open question. In this section, the knowledge that can be used for an effective ship hull form reconstruction is identified based on the knowledge classification discussed in chapter 2. Data knowledge is very much helpful in selecting a suitable pre-processing and surface reconstruction approach for the intended application. The data knowledge includes a point cloud density, noise level, missing data and data size. Data knowledge alone does not guarantee to reconstruct a hull form without the knowledge of an algorithm. Therefore, algorithm knowledge is equally important in this case. Algorithm knowledge includes the knowledge about its inputs (i.e. point cloud with or without normals), the working principle of the algorithm, whether the algorithm is capable to preserve features or not, whether the algorithm provides the required level of smoothness or not, knowledge about the effect of each driving parameters, knowledge about the output offered by the algorithm, etc. Data and algorithm knowledge are used to figure out a suitable reconstruction approach for a specific application and they interact and support each other. On the other hand, the knowledge about the concerned object such as scene and spatial are also the means to improve the RE processes. This directly concerns prior expert's knowledge regarding the concerned objects. As discussed in chapter 2 a ship hull form scene and spatial knowledge comprises: main dimension, centerline, weather deck line, flat of side surface, flat of bottom surface, flat of side curve, flat of bottom curve, stern tube and transom surface. Still the mentioned knowledge cannot be generalized for all kind of ships, as the shapes and features of hull forms for different kind of ships are different from each other.

5.5 Triangulated Surface Reconstruction

In literature there exist various surface reconstruction methods which range from those capable to reconstruct simple shapes to those able to handle complex shapes and data imperfections. The main differences between various reconstruction algorithms lay in the capabilities to handle different aspect of a point cloud properties and imperfections, because point cloud imperfections are the main challenge to surface reconstruction algorithms. Apart from this, the differences in the reconstruction methods also depend on surface representations (i.e. explicit or implicit), fitting approaches (i.e. approximation or interpolation), the nature of the input point cloud (i.e. only points, points with oriented normals, points with unoriented normals) and the output of the reconstruction algorithms (i.e. triangulated, parametric or other). The reconstruction algorithms proposed over the last two decades are capable to reconstruct simple objects successfully but usually resulted in not parameterizable and not easy to modify for complex shapes and even resulted in not usable models in the worst case. Among the various reconstruction algorithms there are *common* and *specific* properties, therefore exploring and evaluating the reconstruction algorithms can help to devise a better reconstruction strategy. Having this in mind, this section aims to develop a ship hull form specific triangular surface reconstruction strategy which consists of the approaches and procedures used to improve the quality, capability and usability of the existing reconstruction algorithms. The evaluation of reconstruction algorithms performed in previous topics provides the weak and strong side of the considered reconstruction algorithms. These lay a foundation for the improvement of the reconstruction capability, quality and usability. The developed strategy consists of three main modules: triangulated surface reconstruction pre-processing, reconstruction approaches and post-processing as shown in Figure 5.3. Each module is developed

with the help of geometrical processing algorithms and additionally integrated with knowledge. The statistical and numerical geometric processing include: pre-processing (i.e. filtering, down-sampling, normal estimation and geometric detection), surface reconstruction algorithms (i.e. alpha shapes, ball pivoting, greedy projection triangulation, grid projection, marching cubes and Poisson method) and post-processing (i.e. hole filling, smoothing, trimming, normal estimation and down-sampling). Purely mathematical based surface reconstruction approaches are usually ill-posed, because an infinite number of surfaces could pass through or near the data points. This problem is attributed to the incapacibilities of the reconstruction algorithms, the point cloud artifacts and the complexity of the underlying objects. Therefore, previous studies related to knowledge based RE reveal that the incorporation of prior knowledge in RE processes improves the quality of the output result. Hence, different kind of knowledge (i.e. scene, spatial, algorithm and data) are integrated into the developed framework to help the effort to recover as much information of the shape as possible, as shown in Figure 5.3. The developed method is described in detail in the following subsection.

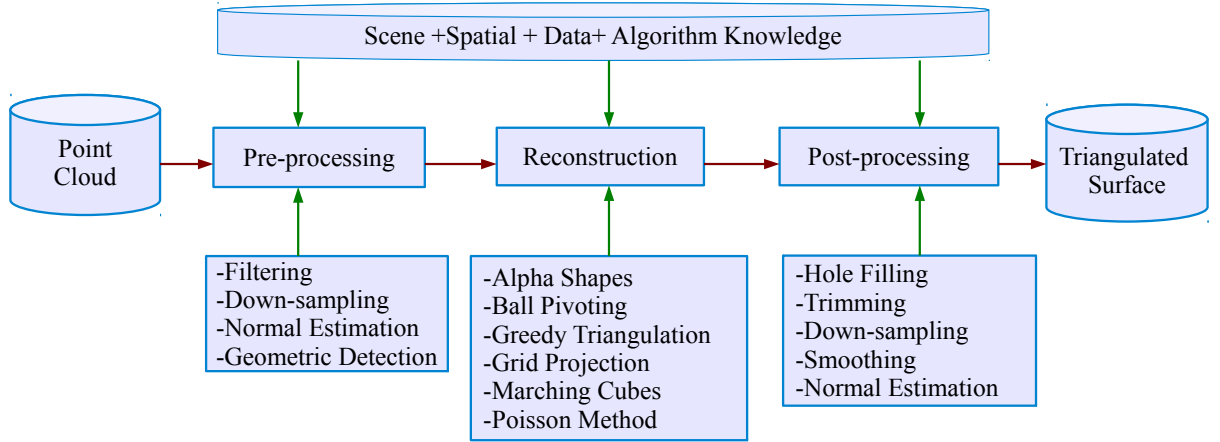


Figure 5.3: Triangulated surface reconstruction layout.

5.5.1 Triangulated Surface Reconstruction Point Cloud Pre-Processing

The capabilities of the modern point cloud sampling systems have increased over the last years which primarily make the high sampling density of the surfaces economically feasible. However, point cloud artifacts remain the main challenge of surface reconstruction algorithms. Point cloud artifacts (i.e. outliers, noise, missing data, and varying density) are not the only treat to surface reconstruction, the huge density of a point cloud from scanners also has significantly slowed down the reconstruction process. Not only slowing down but also brought to a complete halt in extreme cases, despite the high processing power of the modern computers. Therefore, the point cloud pre-processing is not a choice but it is an unavoidable step. Point cloud pre-processing is necessary to reconstruct a geometrically correct object by removing the point cloud artifacts, and also makes the reconstruction process possible, affordable and easy. However, it also usually results in the loss of features such as sharp edges and boundaries. Therefore, the pre-processing algorithms should be wisely used in such a way that the object's features are preserved. As indicated in the surface reconstruction algorithm evaluation, every algorithm has its own input requirements which have to be fulfilled before the reconstruction processes. Therefore, pre-processing can not be ignored or reduced to outliers removal or noise elimination or data reduction, but it should include all input requirements of the respective reconstruction algorithms. Based on the evaluation, necessary point cloud pre-processing algorithms are identified and implemented in the developed triangulated surface methodology.

5.5.1.1 3D Filtering and Down-Sampling

The filtering implemented in this section includes outliers removal and noise elimination strategies. The outliers removal is a mandatory step, while noise elimination is not, because some reconstruction algorithms such as the Poisson method, are robust to noisy point cloud. The statistical outliers removal algorithm is implemented in the developed framework to remove outliers and eliminate noise as shown in Figure 5.4. It requires to define two main parameters, the number of the nearest neighbors and standard deviation multiplier which used to classify the points as inlier or outliers.

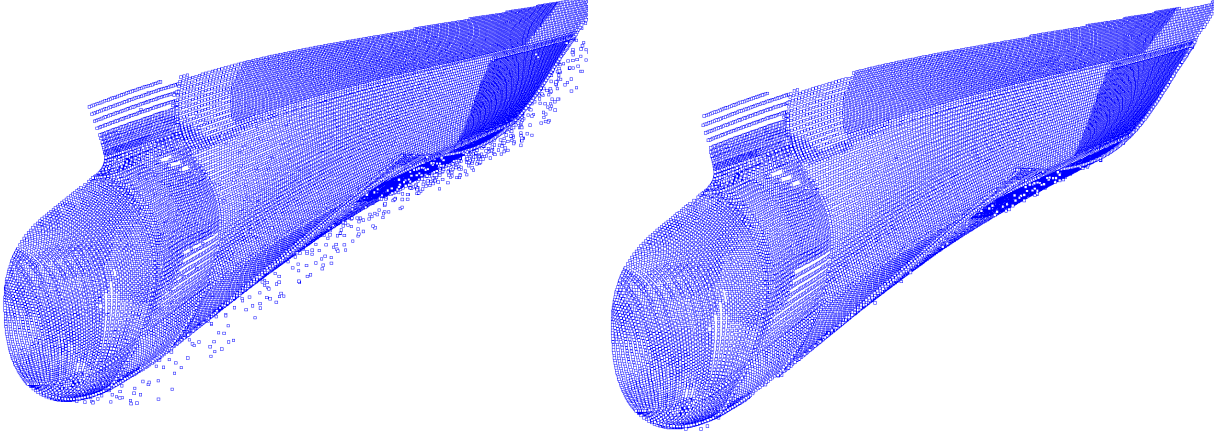


Figure 5.4: Noise reduction using the statistical method.

Additionally the 3D voxel grid algorithm is integrated to down-sample and de-noise the raw point cloud. The algorithm mainly requires to define the resolution (grid size). The points fall into a 3D box created from the resolution values are approximated by their centroid. Figure 5.5 shows the down-sampling and filtering of a scanned boat point cloud with different value of resolution (i.e. the three resolution values are set to equal). An unreasonably high resolution value usually results in loss of the object's features. The working principle of both statistical outliers removal and 3D voxel grid algorithms are described in chapter 2.

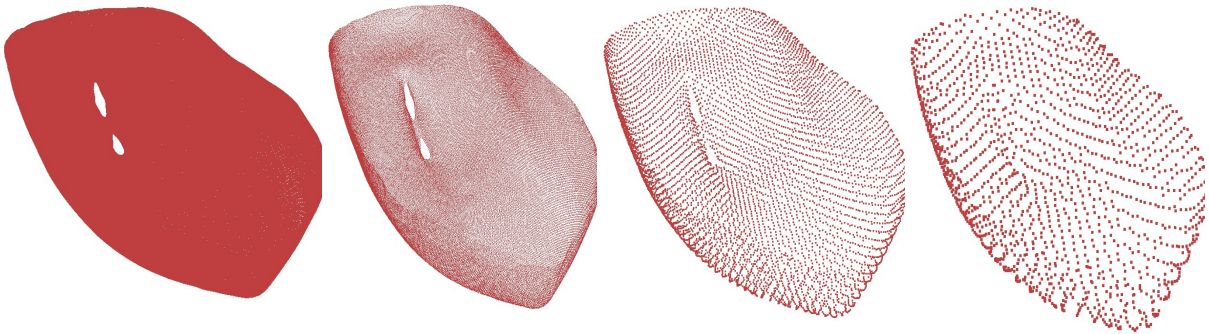


Figure 5.5: Point cloud down-sampling using 3D voxel grid method. Number of points (1074993, 142303, 6431, and 1683) with grid resolution (original, 5mm, 25mm and 50mm) from left to right respectively.

5.5.1.2 Normal Estimation for Pre-Processing and Surface Reconstruction

Many surface reconstruction approaches require a consistently oriented points while some others reconstruct only based on points. For methods which require point cloud normals, the point cloud

normal estimation method is implemented. It is approximated by the problem of estimating the normal of a plane tangent to the surface, which in turn becomes a least square fitting estimation problem. A plane is fitted to the query point and its neighborhood points. The number of neighbor points are determined using a radius which the user needs to define. As shown in Figure 5.6 once the user defines the radius of the sphere, the points fall inside the sphere are fitted to a plane from which the point normal could be estimated. A normal estimation could be resulted into an oriented or unoriented normals. Unoriented normals do not possess direction which means the points orientation could point either inside or outside direction of the surface as shown in Figure 5.6 (right). Oriented normals have a consistent direction throughout the whole points as shown in Figure 5.6 (left). The estimation algorithm integrated in the framework is described in chapter 2.

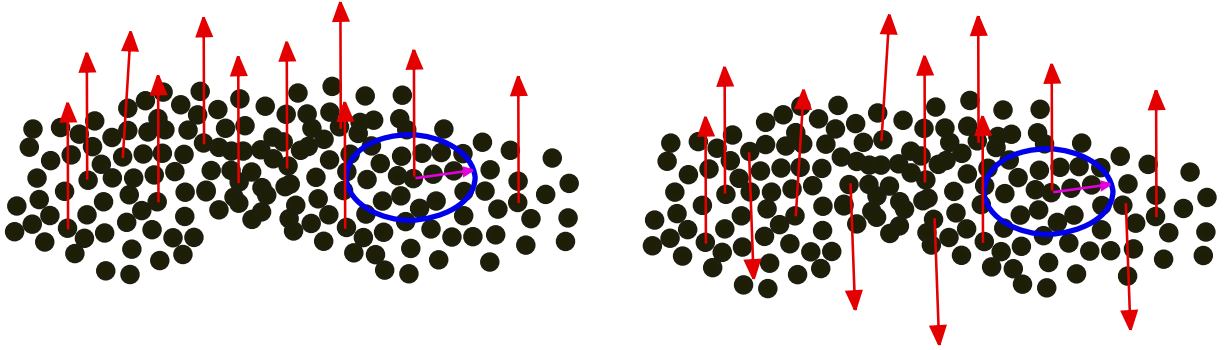


Figure 5.6: Point cloud normals estimation. Consistently oriented estimated normals (left) and unoriented normal (right).

5.5.1.3 Geometric Detection

The demand for a geometric detection is ever growing due to the need for a concise and meaningful down-sampling and simplification of an increasing size and complexity of the point cloud data. Geometric primitives such as planes, cylinders, spheres, cones, tori, etc are the most construction elements in many engineering applications, therefore, the functionality of a geometric detection in RE methods has a tremendous positive effect in reconstructing quality results as shown in Figure 5.7. It is a divide and conquer approach which is extremely helpful specially for the mechanical systems where the objects are made of many primitive shapes. It divides the entire objects so that every part is well approximated by a corresponding geometric primitives. From general knowledge, the use of geometric detection in a hull form RE is limited to the detection of planes (i.e. flat of bottom, flat of side and transom), cylinder (i.e. stern tube) and boundary detection e.g. centerline and weather deck, if applicable. In this section, the implemented flat of bottom, flat of side, transom and stern tube detection techniques such as normal and curvature method and plane cylinder detection are described.

Normal and Curvature Method: The ship hull form flat of side, flat of bottom and transom can be detected using points normal and curvature value criteria. The point cloud normal and curvature values are determined using the moving least square implemented from PCL. To simplify the detection, the hull form region near to the planes are automatically separated. For instance to detect flat side, the algorithm only considers 90% of the total hull form breadth, and finally the normal and curvature of the separated point cloud is determined and finally the points with an approximate unit normal $(0.0, \pm 1.0, 0.0)$ and curvature (0.0) are identified. The result obtained using this method is shown in Figure 5.8 (left).

Plane and cylinder Detection : The plane and cylinder fitting approaches described in chapter 2 are used to detect flat of side, flat of bottom, transom and stern tube. The plane

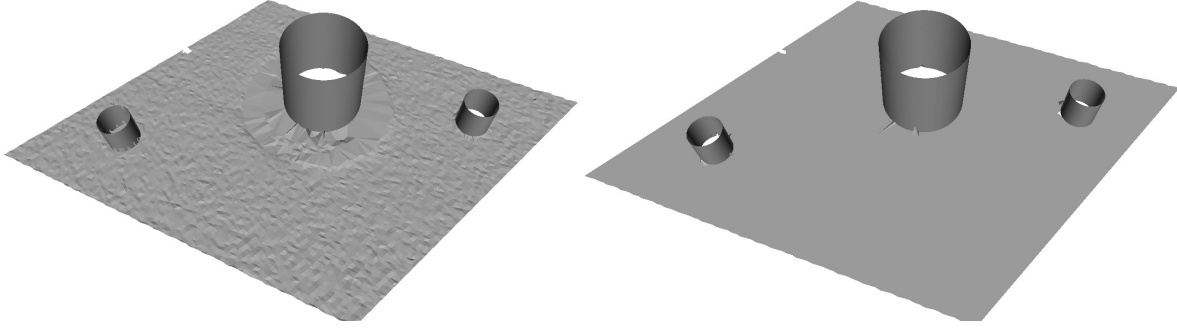


Figure 5.7: Reconstructed objects without knowledge and primitive detection (left) and reconstruction with knowledge and detection integration.

fitting depends on the distance threshold and the number of iteration, both needs to be defined by user. The hull form flats detected with this approach (see Figure 5.8 (right)) are more robust compares to the normal and curvature value criteria as shown in Figure 5.8 (left).

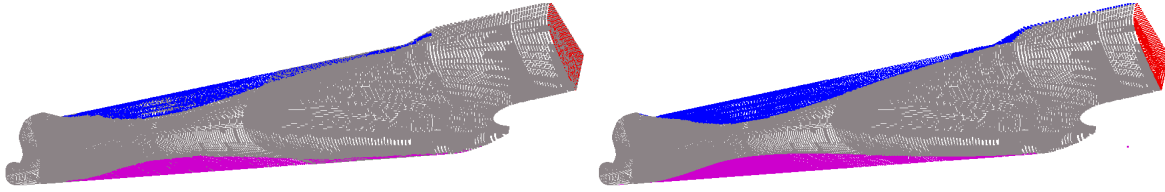


Figure 5.8: Hull form primitives detection (i.e. flat of side, flat of bottom, and transom). Detected using the normal and curvature criteria (left) and detected using the plane fitting approach (right).

5.5.1.4 Knowledge Based Pre-Processing

Knowledge based pre-processing refers to the use of any prior knowledge which ranges from basic metric knowledge (i.e. over all dimension, stern tube radius) to the spatial knowledge (symmetry, parallelism, perpendicularity) of different scenes of the hull form. In the developed framework, overall dimensions are used to trim extreme outliers. Different hull form knowledge are used in combination with geometric detection algorithms. For instance, the centerline always lays on constant x-axis and this knowledge is enforced to make the centerline position accurate. Likewise, the outliers, the noise and the misalignment errors could be removed from the point cloud using the stated knowledge. Sufficient knowledge does not only provide useful information about the object but also helps to use the involved algorithms effectively. For instance, the success rate of a primitive detection is increased by integrating the location of the flat of side, flat of bottom, transom surfaces.

5.5.2 Surface Reconstruction Approaches

The surface reconstruction step consists of six alternatives to reconstructed triangulated surfaces from a pre-processed point cloud. Each reconstruction method has its own input data requirements. Some requires only point cloud filtering and down-sampling to recover the underlying surfaces while others require a time consuming point cloud normal estimation. Pre-processing algorithms are incorporated into the developed framework to make the reconstruction successful and efficient.

Once the input data requirements for the surface reconstruction are fulfilled, the reconstruction could be performed by choosing the suitable reconstruction method for an intended application. All the selected methods offer their own advantages with their respective disadvantages. Figure 5.9 shows the reconstructed hull form applying the implemented reconstruction algorithms. In general the first three (i.e. a, b, c) are efficient in preserving the boundary edges. They also generally preserve features such as knuckles for non-occlude surfaces. For an occlude surface and surfaces running near to each other, they face a challenge to preserve the object's detail features in case the point cloud density is courser than the distance between the two surfaces. They usually produce holes in less dense regions with inappropriate driving parameters. The last two (i.e. e, f) are strictly dependent on the normal consistency, while (d) can reconstruct a surface based on unoriented normals too. They are known to produce a smooth surfaces even with a presence of noise. However, they usually smooth out the features such as knuckles, and also do not recover an appropriate boundary edges.

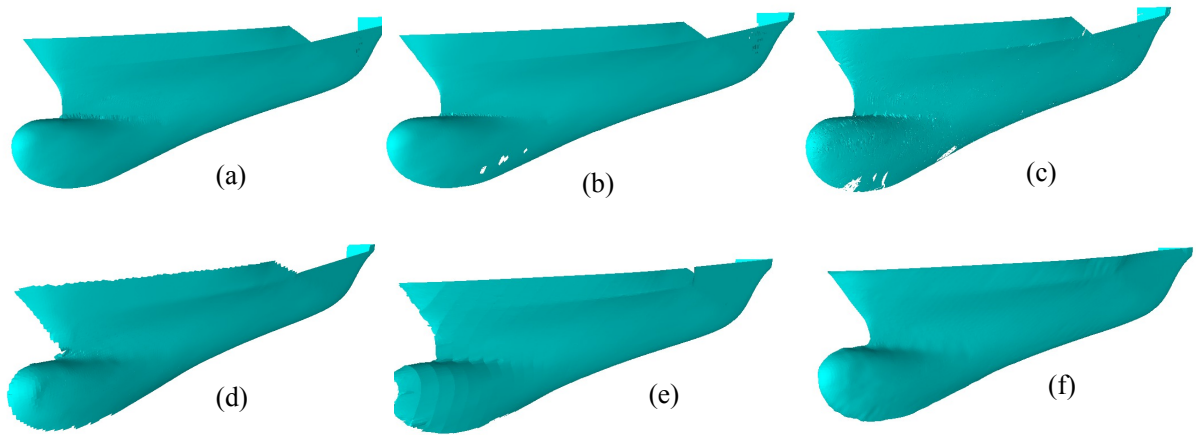


Figure 5.9: Triangular surface reconstructed using different approaches. a) Alpha shapes, b) ball pivoting, c) greedy projection triangulation, d) grid projection, e) marching cubes and f) Poisson methods.

For successful reconstruction processes and the generation of a high quality output surface, the knowledge about reconstruction algorithms and input data are very essential. For instance, to reconstruct the surface using the ball pivoting approach successfully, the user should be able to answer the following basic algorithm knowledge: the working principle of an algorithm; number of input requirements. How the driving parameters affect the reconstruction process? Does eliminating noise very essential? Does it preserve features and boundary edges? In addition the user should have a knowhow about the input data characteristics such as density, level of noise etc., and should also know the spatial and scene knowledge about the object. These provide a knowhow on whether the ball pivoting is suitable or not for the object going to be reconstructed.

Having sufficient algorithm knowledge answers many questions in the RE process. In the presence of an alternative reconstruction approaches, answering the following questions will be half a way to the successful and efficient reconstruction results. The question might include:

- Is it suitable for an intended application (i.e. complexity of an object)?
- Does it handle noisy data?
- What pre-processing steps it requires?
- What post-processing steps it requires?
- How the input driving parameters affect the reconstruction process?

The ball pivoting and Poisson method are discussed in detail to answer the above questions related to algorithm knowledge.

The ball pivoting approach interpolates through most of the input point cloud, therefore the noise should be removed at a pre-processing stage as much as possible. Filtering and down-sampling are essential in this case while a normal estimation is not essential in the pre-processing stage. The algorithm depends on a single parameter: the radius of the sphere (ρ). A too large value of the sphere radius does not preserve features while a too small value results in holes in less densely regions. Figure 5.10 shows the effect of the input parameter (ρ) in the ball pivoting reconstruction approach on an artificially generated container ship (LOA = 206m) low density point cloud. It is reconstructed with different ρ values which results in different outputs. With a sphere radius value of $\rho = 300mm$ the method produces many holes in the low density areas, see Figure 5.10 (b). The black color in the bow area shows the difference in the orientation because they are not a single mesh but a combination of separate triangulated surfaces. In this case, increasing the ρ value will resolves the problem, as a result the ρ value is increased to $400mm$ which removes most of the holes but still few remain, see Figure 5.10 (c). The ρ value is further increased to $500mm$ which removes all the holes, see Figure 5.10 (d). It preserves boundary edges (e.g. centerline and weather deck-line) and knuckles (e.g. the connection between forecastle and free to bow region). However, the method can not properly recover the region between the bulbous bow and the upper bow with $\rho = 500mm$.

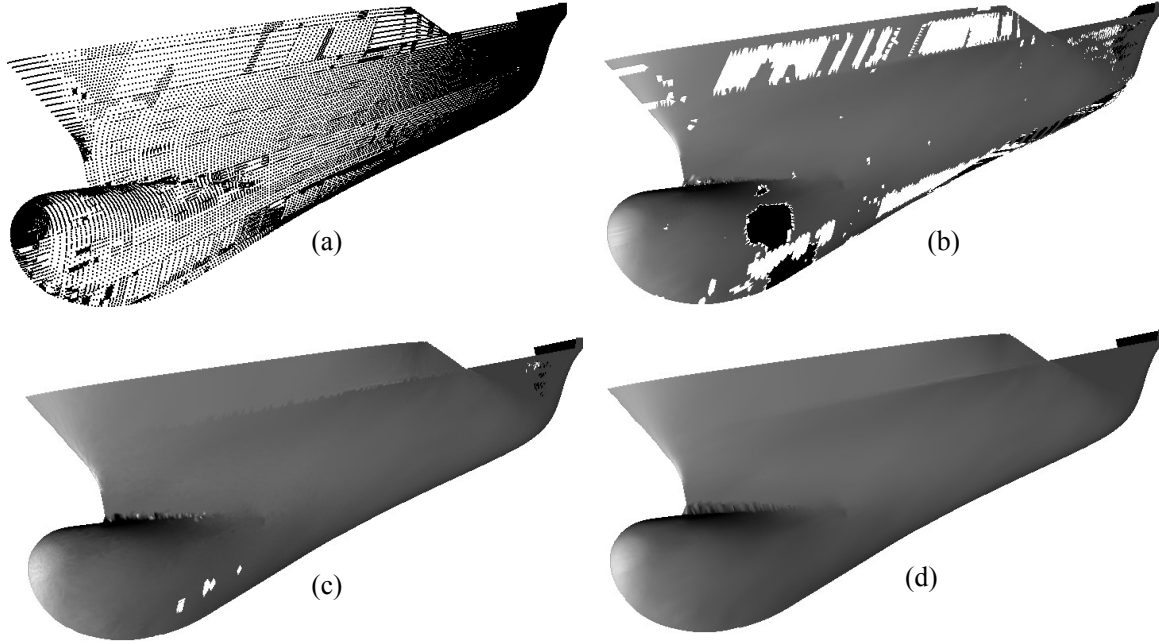


Figure 5.10: Ball pivoting reconstruction approach and its driving parameter. a) input point cloud (artificially generated), b) reconstructed with $\rho = 300mm$ c) reconstructed with $\rho = 400mm$ and d) reconstructed with $\rho = 500mm$.

The Poisson method strictly requires a consistent normal estimation and outliers removal as a pre-processing step. Figure 5.11 shows the effect of a normal orientation on the reconstruction result. With an inconsistent normal orientation, the reconstruction algorithm generates inappropriate surfaces such as topological noise, bump surfaces and tunnels as shown in Figure 5.11 (a). The algorithm might even produce unrelated object as it is strictly dependent on the normal orientation. With point cloud and consistently orientated normals, the algorithm provides a proper surface of the object as shown in Figure 5.11 (b). Therefore, a consistent normal estimation is very essential for Poisson reconstruction approach, while down-sampling and noise removal are optional. As an algorithm knowledge, the proper definition of the main parameters such as depth, solver divide, sample per node, scale and iso divide are very important, see Table 5.1. The result shown in Figure 5.11 is generated with a depth ($P_d = 11$), solver divide ($P_{sd} = 10$), sample per node ($P_{sn} = 2$). These are the most important variables to be properly defined. The

large value of P_d and P_{sd} means a high capability to catch the detail of objects and produces denser triangles, but also consumes more memory and time. It does not extract boundary edges, rather provides surfaces which are not a part of the underlying object. Therefore the Poisson reconstruction requires the post-processing step such as trimming, and down-sampling in case large value of P_d and P_{sd} are applied.

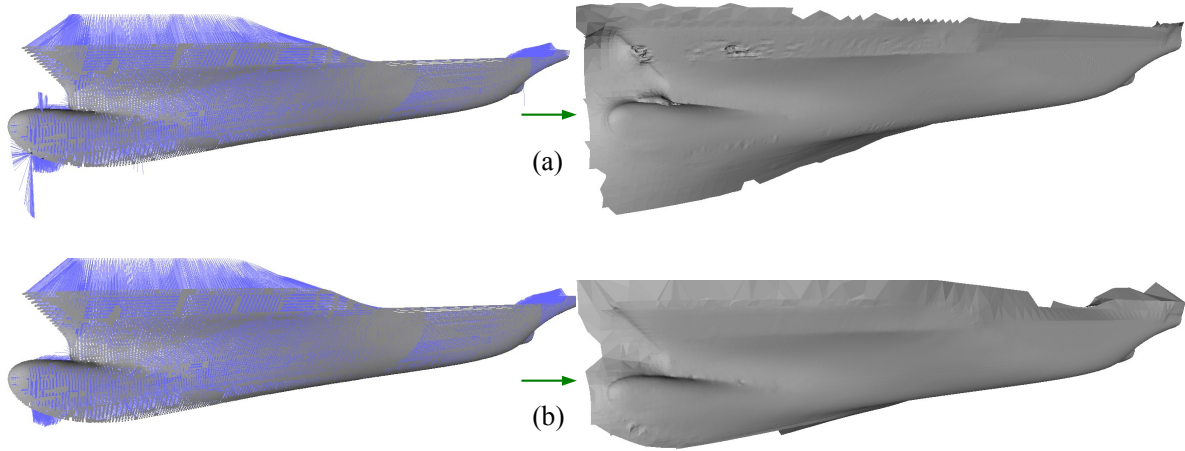


Figure 5.11: Poisson reconstruction approach and its dependence on normal orientation. a) a poor result from not properly oriented normal and b) a result from properly oriented point cloud normals.

5.5.3 Triangulated Surface Post-Processing

5.5.3.1 Hole Filling

The reconstruction approaches, mainly those based on an explicit surface representation require hole filling algorithms due to their incapability to generate watertight surfaces. There are global and local hole filling strategies from which the local filling is found to be suitable for the reconstruction post-processing. Hence an automatic hole identifying and filling of meshes is implemented from VTK. The algorithm locates boundary edges and links them together into loops, and then triangulates the resulting loops. It is driven by the maximum hole size, which is represented by a radius to the bounding circumsphere containing the hole. Figure 5.12 shows the bow part of the boat scanned using markers placed on the surface for a registration purposes. The removal of markers bump outliers creates holes in the point cloud. This point cloud is reconstructed using the ball pivoting method which leaves holes in the mesh as shown in Figure 5.12 (b). The hole filling method automatically detects and fills as shown in (c).

5.5.3.2 Trimming

The marching cubes and Poisson methods do not preserve the boundary edges of the object instead produce extra surfaces which are not part of the underlying surfaces as shown in Figure 5.11. The surfaces which are not part of the underlying object (ship hull form) should be removed using the boundary edge knowledge. For a ship hull form there are two knowledge such as the centerline and weather deck information. These two knowledge can be used to recover the relevant boundary edges of the ship hull form. The centerline and weather deck planes are automatically constructed as shown in Figure 5.13. The reconstructed ship hull form is trimmed fully automatically using the planes. The position of the planes are by default detected from the model automatically but also could be enforced by the user as a knowledge.

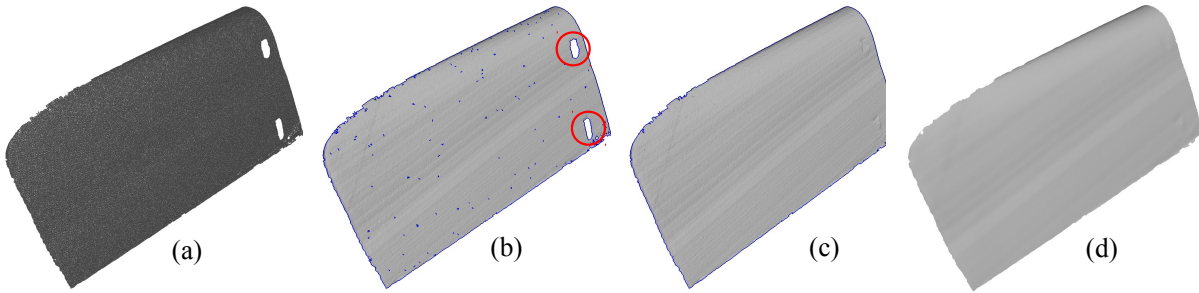


Figure 5.12: Reconstructed triangular surface post-processing. (a) ball pivoting reconstructed surface with holes, (b) boundary edges detection, (c) locally repaired holes and (d) smoothed surface using Laplacian smoothing method. The circles on (b) show the required maximum radius to close the holes.

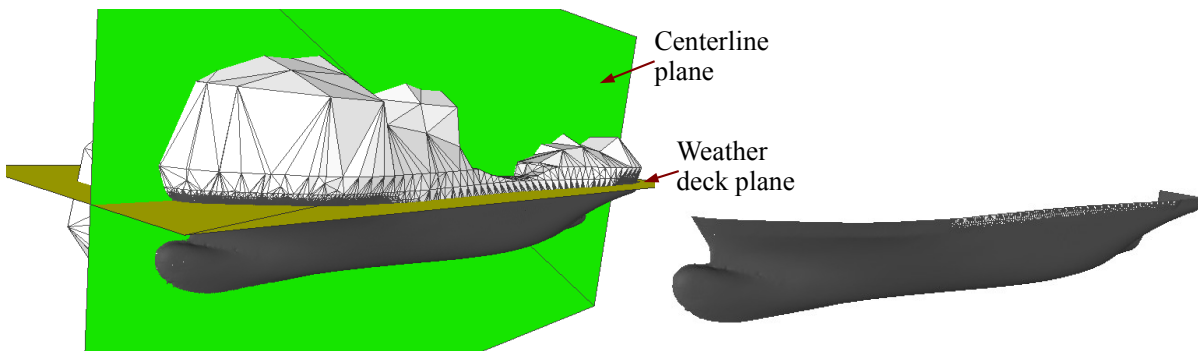


Figure 5.13: Triangulated surface trimming using the centerline and weather deck planes.

5.5.3.3 Down-Sampling and Smoothing

These functionalities are usually optional. Their desirability depends on the down-sampling and smoothing processes performed at pre-processing stage and its downstream application and users desire. In case the reconstruction results are too densely and rough, the implemented mesh down-sampling and smoothing strategies could be used. The reconstruction approaches usually produce rough surfaces if they interpolate point cloud fully or partially. These methods usually require smoothing as a post-processing step, see Figure 5.12.

5.5.3.4 Triangulated Surface Normal Estimation

Surface normals play a major role in downstream applications such as geometric manipulation, visualization, CFD, FEA, etc. Therefore, generating a mesh with a consistent normal orientation is usually essential. Some reconstruction methods such as ball pivoting provides triangles without normal information, others such as alpha shapes and greedy projection triangulation provide unoriented normals. Therefore, the normal estimation based on a triangulated surface is implemented including the algorithm which orients unoriented normals coherently.

5.6 Discussion

This chapter presents a systematic ship hull form RE strategy which consists of the point cloud pre-processing, surface reconstruction alternatives and post-processing. Various approaches are evaluated and implemented to provide hull form specific RE alternatives. The alternatives are

evaluated and their cons and pros are identified. The evaluation results are used to identify what pre-and post-processing methods should be included to improve the quality and usability of the approaches output. The aim is not only limited to improving mathematical techniques, but also further investigates the possibility to integrate prior knowledge to increase the techniques capability, quality, suitability and usability for specific applications. Table 5.3 presents main RE modules of the developed triangulated surface reconstruction approaches. From left to right the consecutive RE processes used to reconstruct a quality triangular surface from an unorganized point cloud are presented. For each alternative, the necessity and unnecessary of the integrated pre-and post-processing modules are identified. The identification is based on four priority levels: necessary, optional, unnecessary and unknown. “Necessary” stands for a mandatory step to be performed before proceeding to the next step while “unnecessary” means that the module is not necessary to be performed. “Optional” means its necessity depends on the size of the object, the density of the point cloud or the mesh, etc. In some cases it is necessary and in other may not. “Unknown” stands for the situation of an unsure opinion. For instance, the alpha shapes method strictly requires the filtering (i.e. outliers removal and noise elimination), down-sampling and geometric detection could be performed to save time and computer memory, and the quality of the result respectively. The point cloud normal estimation is not necessary at all. The post-processing algorithms are categorized under necessary, unnecessary and optional depending on their requirements. One of the important element of this approach is the integration of knowledge in the developed RE process. As shown in Table 5.3, the necessity of knowledge for different RE process is rated. Data and algorithm knowledge are entirely necessary for each necessary pre-processing and reconstruction algorithm. The necessity of scene and spatial knowledge are also identified at different steps of the hull form RE. The table eases the use of multiple alternative RE approaches and assists the user which part of the consecutive algorithms to be considered.

5.7 Summary

In this chapter the triangular surface reconstruction sub-framework is presented. It consists of four key aspects: pre-processing, surface reconstruction approaches, post-processing and knowledge integration. From which the surface reconstruction approaches could be considered as the core, because the identification of a suitable reconstruction is very essential. As a result, different surface reconstruction approaches are implemented and evaluated against different evaluation criteria. The evaluation result reveals that the input requirements, the working principle, the driving input parameters and the output quality of the respective algorithms. This provides a knowhow on what pre-processing, post-processing and knowledge should be integrated to improve the performance and the output quality. Based on the evaluation result, different pre-processing, post-processing algorithms are implemented. After the mathematical RE algorithms are incorporated, the necessary knowledge for each algorithm is identified even to further increase the quality of the output surface. The way to enforce a hull form related knowledge such as centerline, flat of side, flat of bottom is studied. The basic knowledge required by the users are identified for each processing algorithms. In general, the developed strategy provides different alternatives in a combined sub-framework from which the user can choice based on his/her own desire. Not only alternatives but also incorporates different performance and quality improving functionalities.

		Necessary (●)										
		Optional (◐)										
		Not Necessary (●)										
		Unknown (◑)										
			Filtering	Down-sampling	Normal Estimation	Detection	Surface Reconstruction	Hole Filling	Trimming	Down-sampling	Normal Estimation	Smoothing
Point Cloud Data	Mathematical	●	◐	●	◐	●	AS	●	●	◐	●	◐
	Data Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Algorithm Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Spatial Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Scene Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Mathematical	●	◐	●	◐	●	BP	●	●	◐	●	◐
	Data Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Algorithm Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Spatial Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Scene Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Mathematical	●	◐	●	◐	●	GPT	●	●	◐	●	◐
	Data Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Algorithm Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Spatial Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Scene Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Mathematical	●	◐	●	◐	●	GP	●	●	◐	●	◐
	Data Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Algorithm Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Spatial Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Scene Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Mathematical	●	◐	●	◐	●	MC	●	●	◐	●	◐
	Data Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Algorithm Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Spatial Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Scene Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Mathematical	●	◐	●	◐	●	PM	●	●	◐	●	◐
	Data Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Algorithm Knowledge	●	●	●	●	●	●	●	●	●	●	●
	Spatial Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
	Scene Knowledge	●	◐	●	●	●	◐	◐	●	●	●	●
												Output
												T r i a n g u l a t e d
												S u r f a c e s

Table 5.3: Triangulated surface reconstruction using different reconstruction methods, integration of knowledge, the required pre- and post-processing processes. In the table the following abbreviation stands for AS → Alpha Shapes, BP → Ball Pivoting, GPT → Greedy Projection Triangulation, GT → Grid Triangulation, MC → Marching Cubes and PM → Poisson Method

CHAPTER 6

SURFACE HEALING

6.1 Introduction

The availability of robust CFD softwares and high speed computing has lead to the increasing use of CFD for the solution of fluid engineering problems across all industrial sectors and the marine industry is not exceptional [168]. In just about 30 years, CFD for ship hydrodynamics has surpassed all expectations in reaching astronomical progress, capabilities and milestone of providing the first-generation simulation-based design tools for model- and full-scale simulations and optimization enabling innovative cost-saving designs to meet the challenges of the 21st century, especially with regard to safety, energy and economy. CFD is changing the face of ship hydrodynamics as the simulation-based design approach is replacing the build-and-test approach so that model testing is only required at the final design stage. However, towing tank and wave basin facilities are needed additionally for a model development and CFD validation, which requires even more advanced measurement systems for global and local flow variables and more stringent requirements on experimental uncertainty analysis as it plays an important role in validation procedures [169].

Despite the great development and routine usage of CFD in maritime industry, there are still difficulties to apply with a certain level of accuracy and reliability. ITTC computational fluid dynamics specialist committee conducts a questionnaire in 2011 over the difficulties and limitations of CFD for hydrodynamic analyses [170]. The questionnaire asks what are the difficulties and limitations of CFD to achieve a wider usage and acceptance in hydrodynamic computations. The questionnaire was conducted on expertises in maritime industries, universities and model basins. The results reveal that, around 42% of the total participants have reported difficulties in mesh generation. The difficulties and limitations related to mesh generation are directly related to a subject of computation geometries, which includes: geometric repair and domain preparation, and mesh generation (i.e. surface and volume mesh). These problems are attributed to the development of CAD systems, generation of meshes and the data exchange issues.

Modern CAD systems have attained a certain degree of maturity; however their efficiency, reliability, and compatibility with subsequent analysis tools remain an active research topic to-date. At the heart these problems lie some mathematical issues, concerned with the computation, representation, and manipulation of complex geometries, which have stubbornly resisted the best efforts of the research community to formulate rigorous and efficient solution procedures[132].

CAD systems allow designers to build shapes by modeling the surface patches or polygons that comprise its boundaries. These boundaries are often represented as composite parametric surfaces, or employ a discrete representation in terms of triangular facets. Ideally the CAD software generates at least watertight (C^0 continuity) geometries which are then stored in a format that maintains both a geometric description of each patch as well as a topological connectivity between patches [131]. In addition to generating and storing geometrical and topological data, the capability of exchanging information with other systems in a large range of formats is also one of the most important requirement for CAD systems.

Virtually all computer-based design tasks commence with the use of CAD systems to create detailed geometrical models. These models serve as the base for a diverse downstream applications, such as CFD, stress analysis, geophysical data exploration, computational electromagnets, etc. The models are also employed in many manufacturing processes, such as numerical-controlled machining, injection molding, and casting. The success of such downstream applications is, of course, strongly dependent on the receipt of geometrical models that are accurate, self-consistent, and economical in data volume [132]. The representation of CAD models includes feature based data and a resulting B-Rep model. B-Rep models contain geometry (i.e. shapes), topology (i.e. how objects are connected), and tolerances (i.e. how closely do they actually fit together). This combination of the model data is then accessed by the CAD systems methods to define a valid B-Rep model. Therefore, a valid B-Rep model should consist of geometry, topology, tolerances, and methods used by the CAD system it was defined within [133]. In order to perform a downstream application without problems, the CAD model should constitute a valid geometry, topology and tolerances. If not, it appears to be a time consuming and tedious work to adapt a CAD model to a mesh generator and further for CFD analysis algorithms.

6.2 Related Works

Over the years, there are many techniques emerged by CAD, mesh and computer graphics communities to solve CAD model inconsistencies. Some have suggested generic means to represent the geometry, sometimes interfacing directly with the CAD packages themselves [171, 172]. Others attempt to deal with the representations provided by neutral file formats generated by CAD softwares [136, 138, 173–175]. In the mean time two geometric healing approaches are proposed. One which acts on the CAD model and other acts on the mesh [136, 176]. This work includes both the approaches and attempts to insure a global continuity of the geometrical data model produced by the RE part of the framework or originated from any CAD systems. The latter approach can be further categorized into two: surface-based and volume-based methods. The first, operates directly on the input mesh, while the second convert the mesh into a set of voxels before repair [169, 176, 177]. Bischoff et al. [178] combine the advantage of surface oriented and volumetric algorithms to exploit the topological simplicity of a voxel grid to reconstruct a cleaned up surface in the vicinity of intersections and cracks, but keep the input tessellation in regions that are away from these inconsistencies. Thus they are able to preserve any characteristic structure (i.e. iso-parameter or curvature lines) that might be present in the input tessellation. Their algorithm closes gaps up to a user-defined maximum diameter, resolves intersections, handles incompatible patch orientations and produces a feature-sensitive, manifold output that stays within a prescribed error-tolerance to the input model. Many algorithms developed to clean geometric inconsistencies are tolerance driven and require user's interaction which takes significant time. Busaryev et al. [131] develop an algorithm that simultaneously repairs an imperfect geometry and topology while generating Delaunay meshes. They are able to fix many errors in their user input tolerance driven algorithm. Patel et al. [179] also develop a CAD repairing algorithm which is able to detect commonly found geometrical and topological problems e.g. cracks, gaps, overlaps, intersections, T-connections, and no/invalid topology in the model and repairs them and builds correct topological information. Their algorithm is based on an iterative vertex pair contraction and expansion operations called stitching and filling, respectively, to process the model accurately. The algorithm closes small gaps/overlaps via the stitching operation and fills larger gaps by adding new faces through the filling operation. Petersson et al. [174] develop a tool for the preparation of the CAD geometries imported from IGES file formats and maintain in a boundary representation consisting of a patchwork of trimmed and untrimmed surfaces for mesh generation. They claim that the algorithm can identify and remove gross errors automatically while a user interface is provided for the manipulation of geometries such as correcting invalid

trimming curves or removing unwanted details. Petersson [175] demonstrates an application code called *rap*, which can fulfill the needs of mesh generators by cleaning up CAD geometries imported from an IGES file. The topology of the model is computed and a watertight surface triangulation is created on the CAD model which simplifies and speeds up the mesh generation operation. Specific to ship design, numerical simulations used in the design process require a consistent representation of a CAD model as an input. For instance, ship RANSE simulations require objects of interest such as a hull form, duct, volute, etc, which represented either mathematically closed assembly of surfaces within a B-Rep or by a closed volume object within a CSG definition [180]. Different attempts to recover a ship hull form topology are presented, see [180, 181]. In this thesis, an automatic surface healing strategy is developed to reduce the amount of time and cost associated with geometric cleaning [182, 183].

6.3 Repairing Geometrical and Topological Inconsistencies

CAD inconsistencies are usually introduced in different stages of FE and RE processes. The categories of CAD inconsistencies (i.e. dependent and independent) with regard to integration into a numerical simulation are discussed in chapter 2. In this work, two repair modules are developed based on the classification as: repair based on parametric surfaces that aims to repair independent CAD errors and repair based on triangulated surfaces that aims to repair dependent CAD errors. Both repair algorithms are developed to heal geometrical and topological inconsistencies in a CAD model and to transfer it to the region identification and domain preparation modules of the developed framework.

6.3.1 Repair Based on Parametric Surfaces

The repair based on parametric surfaces is developed to heal commonly found invalid representations and to ensure a global continuity of the model. The algorithm is developed based on the software development platform called OCCT. Geometries are read from an IGES file format or directly taken from the reconstruction algorithms, and translated into OCCT shapes object on which a geometric healing is performed. Finally the repaired model is transferred to the next level.

The core reason for selecting OCCT was its capability to handle NURBS surfaces, IGES file format, different level tolerance definition and varies functionalities compared to other open source CAD modeling libraries such as CGAL and others. Therefore the advantages of this platform are used to achieve the objectives of this work.

An IGES file format is considered as a primary input file format in addition to surface patches received from the RE process. Because IGES file format usually fails to preserve topological information between surface patches, and is mostly unable to convey a connectivity between surfaces. If the IGES file format is opened with different CAD systems, various magnitude of gaps and overlaps can be observed as shown in Figure 6.1. This leads to tedious, time and resource consuming pre-processing steps to prepare this and similar hull form for numerical simulations.

After reading surface patches from IGES file format or obtains from the RE process, all the entities are translated into OCCT data structures on which commonly found topological and geometrical errors are repaired. The general layout of parametric surface patches healing algorithm is shown in Figure 6.3 and the detail explanations are given in the following sections.

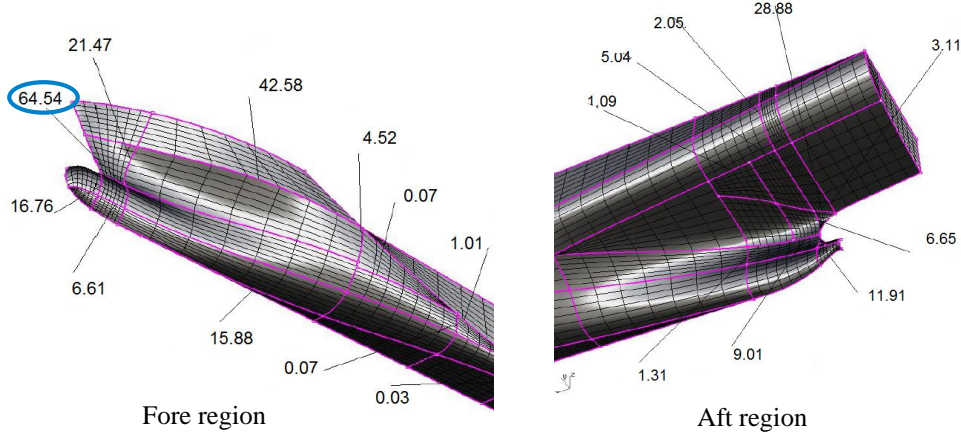


Figure 6.1: The magnitude of gaps and overlaps observed in KCS ship hull form (Length = 230m) read from the IGES file format.

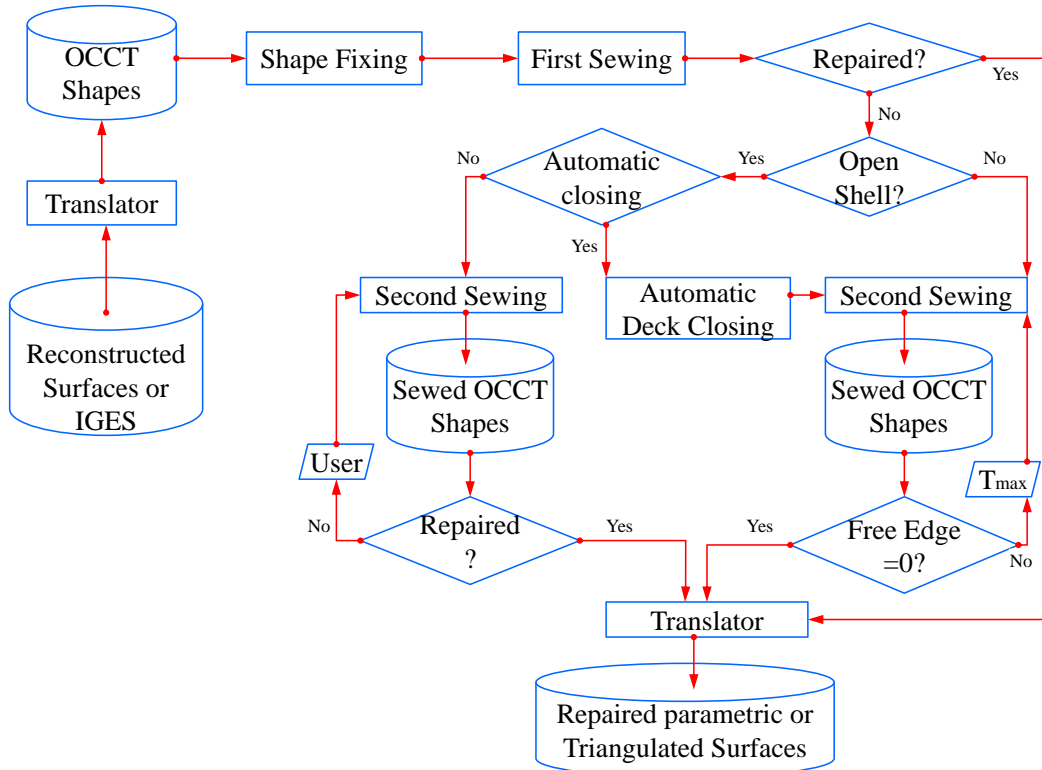


Figure 6.2: A repair algorithm for parametric surface patches (reconstructed or read from an IGES file format).

6.3.1.1 Data Translation and Statistical Extraction

The imported geometries are translated into OCCT information model (see chapter 2 for detail) on which the repairing algorithms are performed. Afterwards, the statistical extraction package extracts some important parameters from an IGES global section and the model itself. These parameters include:

- overall dimensions;
- number of free edges in the model before and after repair;
- the minimum and maximum edge length in the model;

- the number and type of surface patches;
- the IGES global section parameters (such as the unit used to write the file, IGES resolution, the scale, etc.) and
- the minimum edge length greater than IGES resolution.

Some of these parameters are used for the subsequent shape fixing and sewing algorithms. Others are simply for the user's information, to give an overview about the model and the file format constitutes.

6.3.1.2 Shape Fixing

Due to various reasons described in chapter 2, there might exist intra and inter entity inconsistencies in the CAD model. Shape fixing is developed to solve inconsistencies in individual topological entities (i.e. edges, wires, faces) or geometrical entities (i.e. curves, surfaces) which are intra entity errors. Intra entity inconsistencies (i.e. inconsistencies in entities) imported from an IGES file format or introduced during the RE process are repaired using the shape fixing module of the developed repairing method. In the developed method, different shape fixing packages organized under *ShapeFix* classes (from OCCT) are used to solve the problems in entities. It is not necessary for a user to detect problems before using *ShapeFix* because all components of the *ShapeFix* package make an analysis of existing problems before fixing them by a corresponding tool from the package of *ShapeAnalysis* in order to fix the discovered problems. For example, the algorithm explores all faces in an input model and checks whether it fulfills the general requirements to be a face. The same holds for edges and vertices. The inconsistencies in wires are resolved with the help of *ShapeFix_Wire* class which includes the geometrical filling of gaps. These fixing and geometrical filling of gaps include:

- fix disordered edges in the wire (reorder);
- fix small edges (remove edges with a length less than the given tolerance value), degenerate edge;
- fix disconnected edges (adjacent edges having different vertices), ensure whether the end vertex of the preceding edge coincides with the start vertex of the following;
- fix intersections of 2D curves of the edges (self-intersection of 2D curves of individual edges, intersection of 2D curves of the two adjacent edges, intersection of 2D curves of non-adjacent edges);
- fix lacking edges to fill gaps in the parametric space of a surface, or check whether a wire is not closed in the parametric space of the surface. The algorithm computes the gap between the edges by analyzing the positional relationship of the ends of these edges and tries to insert a new edge into the gap or increases the tolerance;
- fix gaps in 2D and 3D wires by means of the geometrical filling (check gaps between the ends of 2D or 3D curves of adjacent edges).

The inconsistencies related to a face are resolved by the class called *ShapeFix_Face*. It fixes problems on a face with regard to its wire. It allows controlling the creation of a face (adding the wires), and fixing its wires by means of a class *ShapeFix_Wire*. The following are common inconsistencies related to a face:

- disorder of wires on the face;
- a face with two wires;
- a face with no wires;
- wires on the surface with degenerate edges.

For more information and figural illustration of the above listed inconsistencies refer to chapter 2.

Another shape fixing class that has been integrated into the shape fixing module is *Shape_Fix_FixSmallFace* which drops small faces from an input model (shape). These small faces might be spot or strip faces. If the size in one dimension of faces is less than the given precision (strip face) or if the size of the faces are less than the given precision (spot face) the algorithm removes those faces. In general the algorithm explores each entities in an input model and checks whether they fulfill the general requirements of the respective entities. If any entity does not fulfill the requirements, the shape fixing module repairs in one or either of the following ways:

- changing topology: adding /removing /replacing an edge in the wire and/or replacing the vertex in the edge.
- changing geometry: shifting a vertex or adjusting ends of an edge curve to vertices, or re-computing a 3D curve or 2D curves of the edge.
- increasing the tolerance of an edge or a vertex.

To fix the errors in entities the respective tolerance values are set to an IGES precision value with coefficients depending on the type of fixing required, to ensure automatic process.

6.3.1.3 Sewing

In maritime industries complex objects such as a hull form and its appendages are preferred to be represented by NURBS surfaces and conveyed by an IGES file format. An IGES file format has topological problems (i.e. gaps and overlaps) which are a major problem for mesh generator algorithms and thereby for numerical computations. Most downstream applications are unable to perform mesh generation and/or computational analysis on this kind of models. Hence, there are needs to repair inter entity topological problems and to adapt the model to downstream applications input requirements. Therefore the sewing algorithm is implemented to close the gaps and overlaps between neighboring surface patches. In general, it ensures the connectivity (C^0 -continuity) and 'water tightness' of the model which are the basic input requirements for downstream applications.

Sewing Principle: The sewing algorithm receives the repaired model from the shape fixing package and solves the inconsistencies between entities. Figure 6.3 shows roughly the successive procedures (i.e. sewing principle) for gaps and overlaps between simple surfaces. In general the algorithm searches each free boundary in the shape, and then identifies the set of candidates to be sewed based on given criteria (i.e. tolerance, manifold/non manifold) and selects those candidates and finds the midpoint to connect the adjacent faces according to the defined criteria. Finally the selected candidates are connected at the midpoint to build the resulting sewed shape. In Figure 6.3 the red edges represent free edges (open edges) and the green stands for the middle point where the candidates (open edges) are connected. The procedures are similar for both in plane and out of plane surfaces.

Sewing Tolerance Management: The sewing algorithm is tolerance driven and allows the creation of a watertight topology (i.e. shells and wires from faces and edges respectively). The algorithm does not change the geometrical representation of the shapes; but adds the information of the topological connectivity. There are three parameters which control the sewing algorithm:

- maximum tolerance (T_{max}) - a maximal distance between topological elements which can be sewed;
- minimal tolerance (T_{min}) - an allowable magnitude of the smallest element in a shape (edge) after sewing and

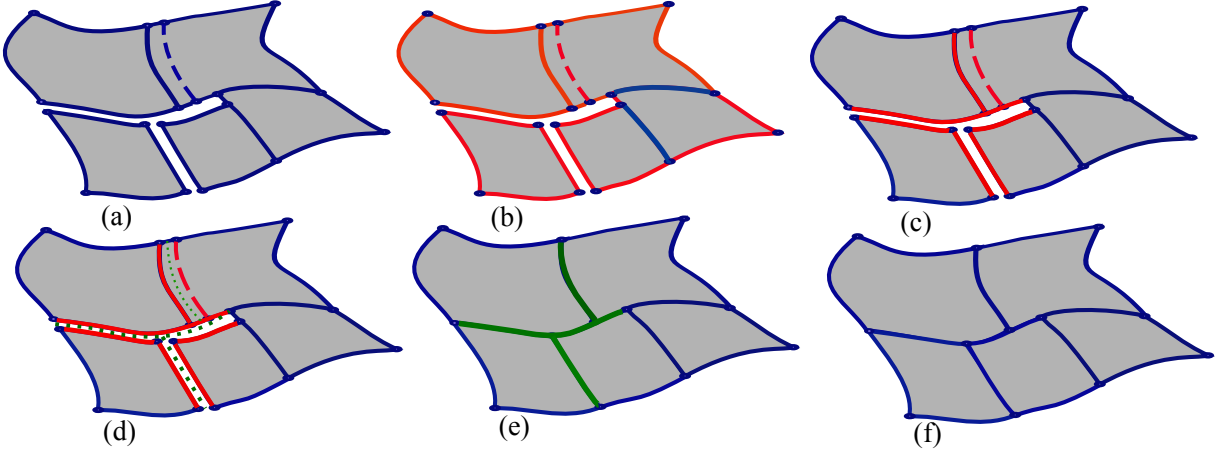


Figure 6.3: Surface patches topology recovery (sewing) steps. a) Surfaces repaired with the shape fixing module - corrupted, b) identify all free edges of the surfaces in the model, c) identify set of candidates to be sewed based on given criteria, d) determine intermediate points of the candidate edges to be merged, e) connected the two adjacent surface patches, f) repaired and consistent surface patches. The red and green color imply for open edges and connected edges respectively.

- manifold/non manifold.

T_{max} is a very essential criterion for the sewing algorithm as it decides whether the gaps or overlaps have to be connected or not. Only gaps and overlaps with a magnitude less or equal to T_{max} are sewed. Definition of a too small tolerance value leaves disconnected surfaces, which certainly results in insufficiently repaired model. On the other hand, a too large T_{max} definition also connects too distant faces which could result in a total damage of the model as shown in Figure 6.4 (c). Therefore, the magnitude of a T_{max} value should be defined in such a way that the shape of the original object does not get damaged. It should be equal to or a bit greater than the maximum size of the gap or overlap.

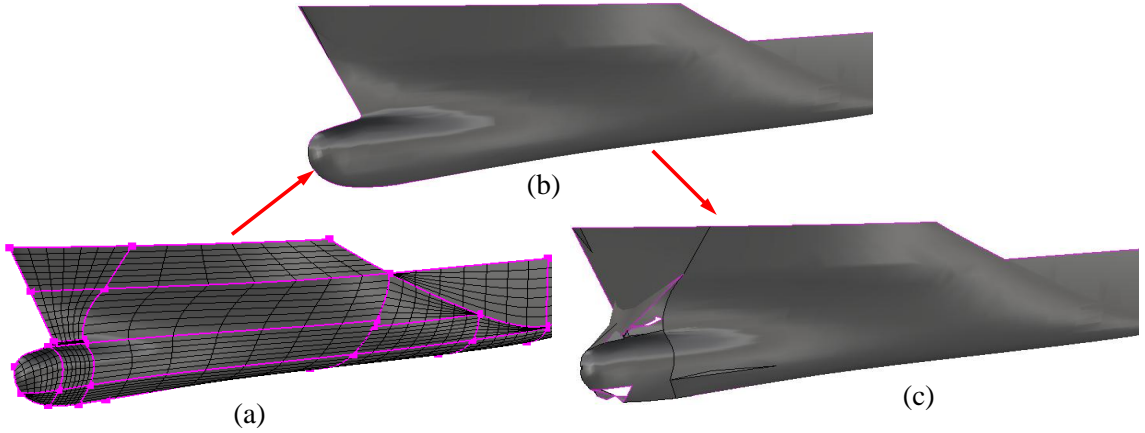


Figure 6.4: Sewing tolerance management. a) the original surface patches of the KCS hull form with gaps (magenta color), b) the hull sewed with ($T_{max} = 5m$), c) sewed with ($T_{max} = 10m$) which leads to total damage of the bulbous bow.

To minimize the problem related to the difficult to define appropriate T_{max} value, two sewing steps are implemented. The first sewing works fully automatically taking T_{min} and T_{max} values from an IGES file format. T_{max} is equated to the minimum edge length greater than IGES resolution (M_{el}) and the IGES resolution is considered as a T_{min} value. The second sewing is also further divided into two: sewing for open and closed shell. Sewing for a closed shell is

an automatic and the tolerance value is determined iteratively between M_{el} and the maximum magnitude of the gaps or overlaps in the model. The sewing for an open shell is a semi-automatic with a user defined T_{max} value. Figure 6.5 shows the maximum gap (64.54mm) between surfaces of the KCS hull form and repaired with a single user defined T_{max} value.

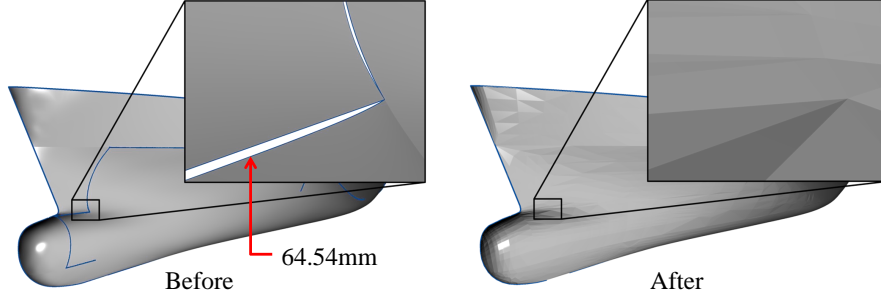


Figure 6.5: The KCS ship hull form (open shell) automatically repaired, before (left) and after (right) repair

Open Shell Sewing: In the developed method, CAD models with holes or any open boundaries (open shells) can not be repaired without the definition of the T_{max} by a user, because, it is difficult to set the T_{max} value which sews the gaps and overlaps between adjacent surfaces but not necessary holes and open boundaries. As a result T_{max} definition for an open shell is left for a user to define depending on the extent of gaps or overlaps present in the model. This enables a user to define the T_{max} to the desired extent and helps to recover the necessary topology. In the first sewing step, both tolerance values are automatically set. If the inconsistencies are not completely healed at this stage, then the geometry will be sent to the second sewing step which requires the definition of the T_{max} value. The automatic definition of the tolerance for the first sewing step ($T_{max} = M_{el}$) gives a clue on what magnitude of T_{max} should be considered for the second sewing step as a first trial. For instance if $M_{el} = n_r$ where $n_r \in \mathbb{R} > 0$ then the user should define $T_{max} > M_{el}$ for the second sewing step in order to repair the remaining inconsistencies. Once T_{max} is defined by the user, the program repairs the gaps and overlaps automatically. Figure 6.5 shows the open shell before and after repair.

Since this work is particularly developed for a ship hull form, a detail case study on sixteen ship hull forms is performed. Table 6.1 shows the number of ship hull forms repaired fully automatically ($T_{max} = M_{el}$) and others semi-automatically ($T_{max} > M_{el}$). If the M_{el} is sufficient enough to repair the model completely then the user does not need to interact. The first six test cases fall into this category which means the inconsistencies are completely repaired without any user interaction. For the last ten test cases, the user needs to define the T_{max} value using the M_{el} as a reference value. Looking into Table 6.1 test cases N^o 9th or 16th, the user can easily reach the appropriate T_{max} value to completely repair the model. On the other hand looking at test case N^o 14th which has the minimum $M_{el} = 0.00744mm$, whereas $T_{max} = 10mm$ and far bigger than M_{el} value, in this case, it is not easy to reach the appropriate T_{max} value.

Closed Shell Sewing: The algorithm determines an appropriate maximum tolerance for closed shells (T_{mcd} value) which is the T_{max} value for closed shells (i.e. no open edge) through an iterative process. The iteration stops when the shape is fully closed (no free edges) and the algorithm exports the watertight (C^0 - continuity) and consistent shape representation to the specified output file format. Figure 6.6 shows a simple box defined by six disconnected faces. This corrupted box is imported and repaired completely in the first sewing step. No action is needed from the second sewing step as no free edge found after the first sewing. In the figure, the blue color stands for free edges.

Table 6.1: Different ship hull form test cases

Nº	Ship Type	LOA [m]	M_{el} [mm]	T_{max} [mm]	T_{mcd} [mm]	Open deck	Closed deck
1	Container ship	224.60	54	54	915	✓	✓
2	Fishing boat	63.41	26	26	305	✓	✓
3	Navy Ship (M)	6.17	228	228	611	✓	✓
4	Kogge	20.88	43	43	110	✓	✓
5	Tanker	333.55	117	117	350	✓	✓
6	Twin Screw Vessel	149.80	311	311	835	✓	✓
7	Container ship (M)	7.69	1.69	51	166		✓
8	Container ship	243.00	53.4	256	994		✓
9	Container ship	205.87	7.78	38	605		✓
10	Container ship	149.30	0.000953	136	650		✓
11	Container ship	207.40	0.00191	382	540		✓
12	Freighter	149.00	0.000954	10	290		✓
13	Ferry	179.66	0.000998	150	590		✓
14	Ferry	72.3	0.000744	10	210		✓
15	Tanker (M)	5.75	0.01528	8	155		✓
16	Twin screw vessel	132.73	2.84	28	597		✓

Where T_{mcd} - Maximum tolerance for closed deck ship hull form, The check-marks imply fully automatically repaired remark.

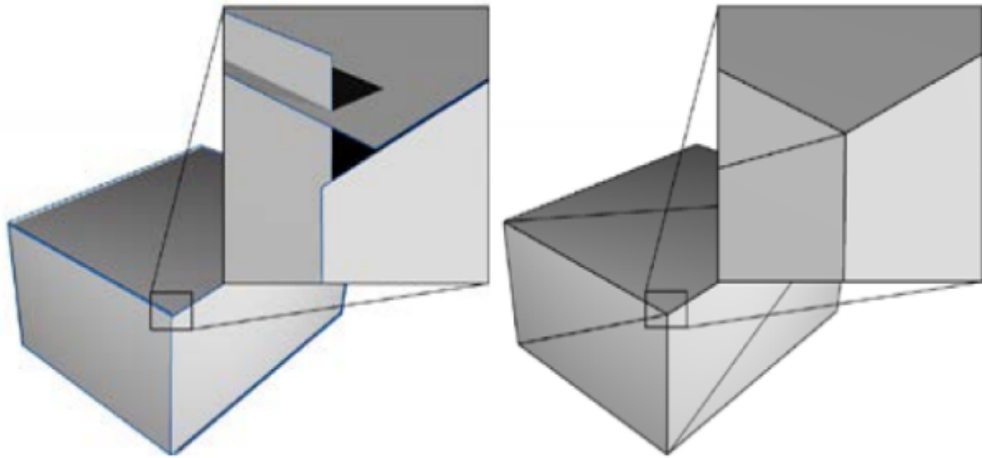


Figure 6.6: A closed shell automatic repairing, the case study of a simple box with gaps and overlaps.

Automatic Hull Form Closing and Repairing: Usually a watertight hull form with the deck closed is required in the grid generators and further in the CFD simulation of ships. Hence, an automatic ship hull form closing strategy is implemented before the sewing algorithm. In addition to fulfilling the requirement of a closed hull form, it enables the possibility of using a fully automatic sewing algorithm. An automatic deck closing includes: detection of the deck

points; ordering the points in clock or anti-clockwise direction; create edges from the points and then wires, from which the deck surface is constructed. After the deck surface is constructed, the port side of the hull is mirrored to the starboard side. A fully automatic construction algorithm requires the port side with the positive Z - axis in the upward direction. After successful construction of the deck and mirroring operation, the second sewing step automatically repairs and provides a watertight ship hull form. Figure 6.7 roughly describes the developed processes to achieve a closed ship hull form and repairing algorithm result for the downstream application. Similarly Figure 6.8 illustrates an automatic sewing of several unconnected surface patches using automatic construction algorithm.

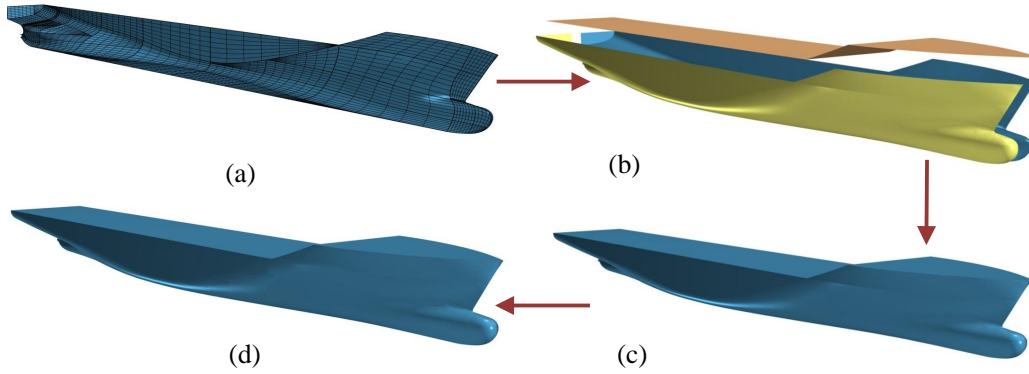


Figure 6.7: Automatic ship hull form closing and repairing steps: a) read the port side hull form from an IGES file format, b) an automatic deck construction and starboard mirroring, c) combining and repairing all together and d) export the fully automatically repaired closed hull form to STL file format or make ready for domain preparation.

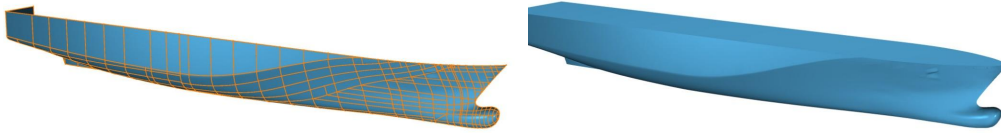


Figure 6.8: Automatic closing and repairing of the open shell hull form consisting of 353 disconnected surface patches.

6.3.2 Repair Based on Triangulated Surfaces

The repair based on triangulated surfaces is intended to resolve dependent CAD errors. This functionality enables the developed framework to heal the inconsistencies introduced from different sources. The inconsistencies in triangulated surfaces are mainly resulted from the triangulated surface reconstruction or meshing algorithms. This module closes the holes based on a user defined tolerance and provides a watertight triangulated surfaces as shown in Figure 6.9. This module adds the following features to the developed framework:

- corrects the magnitude and orientation of facets normals;
- removes degenerated facets, connects neighboring open edges;
- repairs facets by connecting nearby facets that are within a given tolerance;
- fills holes in the mesh by adding facets;
- provides statistics before and after repairing.

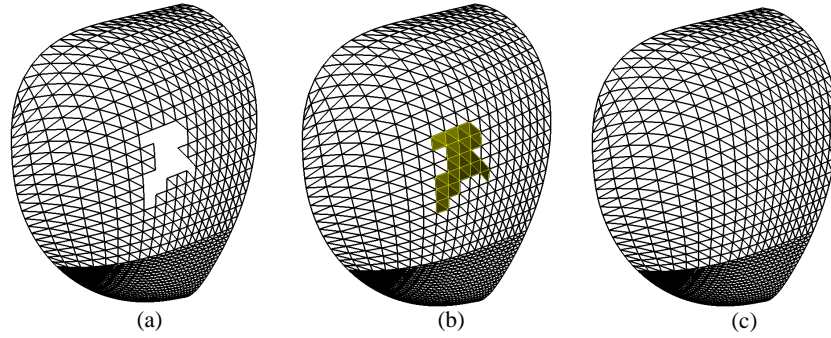


Figure 6.9: Hole filling using user defined tolerance, a) hole on bow part of a hull form, b) hole filling, c) repaired.

6.4 Summary

CAD models are the starting point for many downstream applications such as mesh generation, structural/fluid/thermal analysis, rapid prototyping, numerical controlled machining, casting, and computer graphics. Each of these downstream applications are strongly dependent on the accuracy and consistency of the input geometry, but due to numerical problems, imprecise design, software idiosyncrasies, or data exchange issues, the surface patches produced at the CAD step may abut within unpredictable tolerances, resulting in gaps, cracks, holes, overlaps, T-connections, invalid topology and inconsistent orientation which resulted in an elusive automatic grid generation.

In many cases engineers go through a tedious, time and resource consuming efforts to adapt a corrupted ship hull form CAD data for grid generators and further for hydrodynamic computations. This work presents an automatic CAD repairing approach which aims to reduce the amount of time and cost associated with cleaning/repairing CAD geometric data for grid generation. The approach is capable to repair commonly found geometrical and topological inconsistencies. In addition to repairing 2D and 3D CAD models, the automatic construction of the deck, mirroring of the starboard side are implemented to further facilitate the model for mesh generation and then hydrodynamic computation. As the main target is automatically repairing CAD data errors prior to mesh generation, many test cases are performed for different kind of geometries to check the algorithm consistency and robustness. The repair algorithm is tested against several test cases to evaluate the algorithm robustness. The achieved result reveals that the developed tool substantially decreases the time, and therefore cost, associated with the CAD data repairing and domain preparation.

This page intentionally left blank.

CHAPTER 7

REGION IDENTIFICATION AND DOMAIN PREPARATION

7.1 Introduction

An automatic mesh generation can be usually initialized based on a consistent and watertight CAD models. Unfortunately this alone does not guarantee the required level of mesh quality and successful numerical simulation. Downstream applications (i.e. mesh generator, CFD, FEA, etc.) have their own input requirements based on a specific application. Therefore, it is rather complicated and bulky to device a common strategy which adapts the CAD models for downstream applications. Furthermore, such an attempt would be inevitably incomplete as particular cases and new applications arise too frequently. However, it is possible to develop the region identification and domain preparation strategies which aim to fulfill the input requirements of the specific downstream application. For instance, the preparation of a ship hull form for representative mesh generators and then CFD computation tools is achievable. Therefore, this section receives the repaired ship hull form from the surface healing module and further prepares the model for downstream applications. This section presents the region identification, and domain preparation approaches for two mesh generators *i.e.* *Numeca HEXPRESS* and *snappyHexMesh* which are widely used by marine engineers for a ship hydrodynamic computation.

7.2 Related Works

Specific to a ship hydrodynamic simulation, Bronsart and Knieling [181] implement a hull form knuckle detection strategies. Abt et al. [180] develop a topological recovery and domain preparation strategy based on the Boolean operation. Furthermore Krüger et al. [184] discuss the input geometry requirements of *Numeca HEXPRESS* and the need to identify regions and to prepare domain in order to use the software to its full potential, especially its knuckles detection functionalities. This section identifies the input geometry and domain requirements of *snappyHexMesh* and *Numeca HEXPRESS*, and adapts/prepares the model accordingly [185, 186].

7.3 Ship Hull Form Region Identification

The use of a numerical computation to evaluate ships hydrodynamic performance dramatically reduces the cost and time associate with experimental testing. It can be used to evaluate not only ship's main characteristics (block coefficient, prismatic coefficient, cross-section area curve, etc) but also hull form local details (bulb shape/size, stern shape, skeg size/shape, flap, wedge, interceptor, etc). In order to get a better result and to reduce the computational time, certain regions should be identified, because the quality of the ship hydrodynamic simulation results is strongly dependent on the accuracy of the solver used and the quality of the mesh which represents the hull form and the computational domain.

In this work different region identification methods are investigated to identify relevant local details of a ship hull form. The three different region identification methods are:

- data range method;
- curvature method and
- knuckle detection method.

7.3.1 Data Range Method

The data range method is the simplest method, as it divides the hull form into different regions using cross sectional planes. This method can identify ship regions such as the bow and the stern from the other parts, which are important with regards to the ship hydrodynamic performances. The region identification is performed based on the repaired triangulated surfaces and the sectioning algorithm works with the following procedures:

- First constructs a cutting or a section plane at the desired position, see Figure 7.1.
- Then, categorizes the triangulated surfaces which have no intersection with the section plane into the right and the left side.
- Determines the triangles which have the intersection with the section plane.
- Determines the intersection between the section plane and the triangle. As shown in Figure 7.1 the intersection between section plane and triangle is u_1u_2 .
- Then constructs triangles $v_2u_2u_1$, $v_3u_2v_2$ and $u_1u_2v_1$ respecting the orientation of the original triangle.

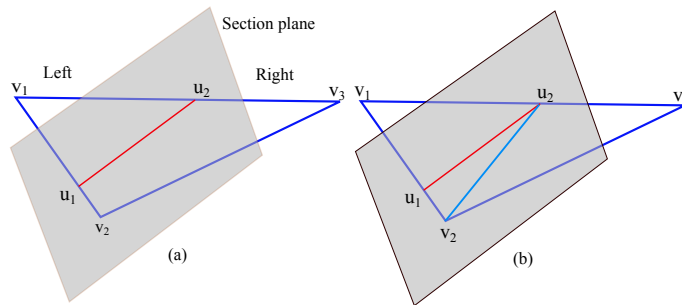


Figure 7.1: Sectioning the triangulated surfaces using the section plane.

The algorithm works automatically and provides three region identification options as an output based on the users desire. The three options are:

- wet and dry regions. Requires only the ship draft as an input.
- wet bow, wet stern, and dry regions. Needs the draft and the coefficient as an input. It helps to single out the bow or the stern region for any kind of special purposes.
- wet bow, wet middle, wet stern and dry regions. This operation requires the definition of the draft, stern coefficient (S_c) and bow coefficient (B_c).

The coefficients S_c and B_c , and the draft are used together with the ship LOA to divide a hull form into different regions as shown in Figure 7.2. Having the coefficients and the draft defined the four regions are defined as follow:

- stern (running) = $[X_{min}, X_{min} + S_c \times LOA]$;
- wet middle = $[X_{min} + S_c \times LOA, X_{min} + (S_c + B_c) \times LOA]$;
- bow region (entrance) = $[X_{min} + (S_c + B_c) \times LOA, X_{max}]$;
- dry region = $[Z_{min} + draft, Z_{max}]$.

Where $LOA = X_{max} - X_{min}$ and X_{min} , Z_{min} and X_{max} , Z_{max} are the minimum and maximum extreme points of the ship in X - axis and Z - axis respectively.

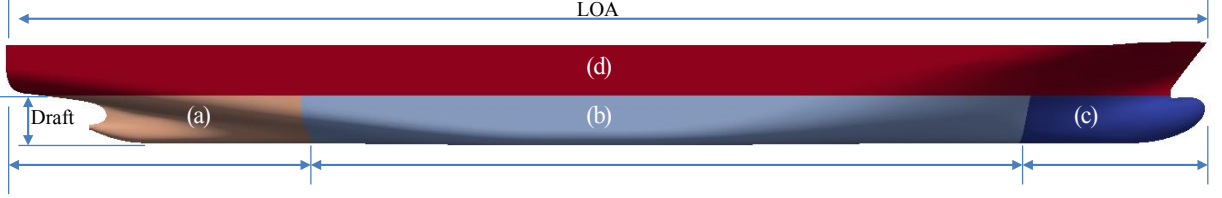


Figure 7.2: Data range method region identification. a) wet stern region, b) wet middle region, c) wet bow region and d) dry region.

7.3.2 Curvature Method

Many scientific applications based on surfaces extensively use the curvature for mesh segmentation, recognition, and registration algorithms. In this thesis work, this geometric quantity is investigated as a region identification methodology. Both Gaussian and mean methods are utilized to determine the surface curvature at a point. A well established curvature determination strategy is implemented in the developed framework. The procedures to determine the curvature are described with the help of the Figure 7.3 as follow:

- At any point (P) on a surface $S(x, y, z)$ find a normal vector (\mathbf{n}).
- Determine a section plane (SP) containing the normal vector.
- Determine the intersection between the plane and the surface which is a section curve ($C(t)$).
- Then determine the curvature of $C(t)$ at P .
- Rotate the section plane with a desired angle to determine curvatures at P with different angles.
- Determine the maximum (K_{max}) and the minimum (K_{min}) curvatures which are known as principal curvatures.
- The product of the two curvatures is Gaussian curvature and the average of the two curvatures is mean curvature at P .

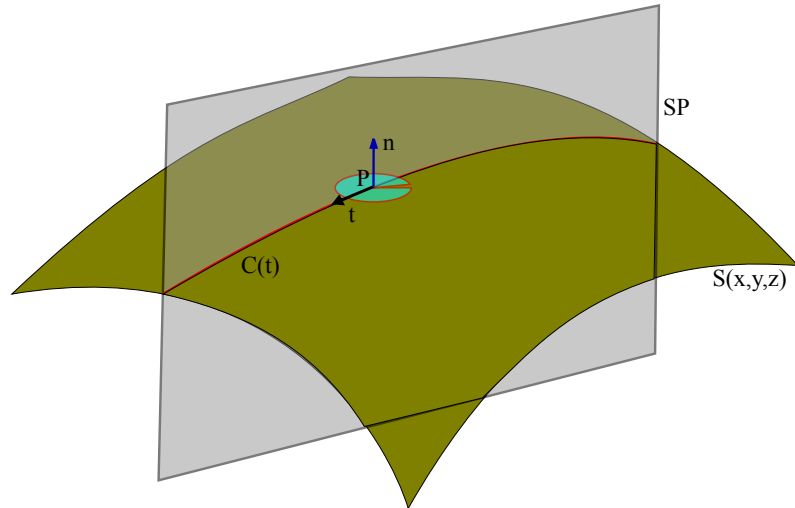


Figure 7.3: Determination of a surface curvature at a point.

The mean and Gaussian curvatures are the two major surface properties employed as a solution for different practical problems such as smoothing or simplifying meshes in a modeling and

manufacturing, also for a surface classification, etc. Unfortunately, they are directly defined only for twice differentiable (c^2) surfaces. Therefore it is not straightforward to determine the curvature of triangular meshes. Hence, in this work parametric surface patches are used as a background for triangular surfaces to determine the curvature values. The vertices of the triangles are considered as the point to determine the curvature values, therefore the algorithm provides the triangular mesh curvatures (i.e. principal, mean and Gaussian) at the vertices of the triangles. The triangulated surface curvatures are determined based on the original parametric surfaces as shown in Figure 7.4.

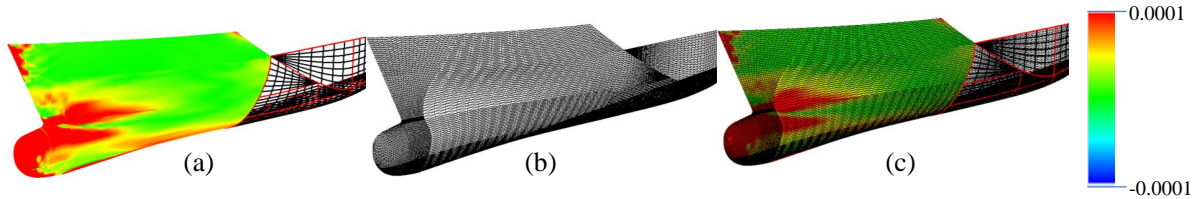


Figure 7.4: The procedures used to estimate the curvature of triangulated surfaces. a) shows the mean curvature of parametric surfaces (bow region), b) the triangulated surface, c) the adaptation of curvature to triangulated surfaces from original parametric surfaces.

The curvature assisted user interactive region identification is possible but the attempt to make it automatic is not trivial. The reason is that, it strongly depends on the complexity of the surfaces and the type of the ship.

7.3.3 Knuckle Detection Method

Almost every ship has at least one knuckle in its hull form definition (i.e. the knuckle between the hull and the transom stern). Often there exist knuckles at a center skeg in the aft part or near the bulbous bow [181]. In the hull design system, the information about knuckle lines is still present, because knuckles are deliberately modeled by the user. Unfortunately this information gets lost while exporting the ship hull form to the IGES or STEP file formats as a set of unconnected surface patches. Even with a correctly defined B-Rep model of the hull surface it is not possible to capture information about knuckle lines [181]. For parametric surfaces, knuckles might exist at the boundary between two patches or inside the surface itself. In this work, two detection strategies recommended in [181] to determine a knuckle between adjacent surface patches are implemented. The first method calculates the points on the edge of the two surfaces and projects them onto both faces. Then it is possible to calculate and compare the surface normals of both faces at the projected points as shown in Figure 7.5 (right). This method is fast as it only needs to calculate normals and compares them, but sometimes the normals are calculated wrongly and the edge is recognized as knuckle although it is not. The second method defines section planes at a number of points along the edge. With these section planes the two faces that share the edge are sectioned. Then the tangents of the section curves at the edge can be compared for the decision whether the edge is a knuckle or not as shown in Figure 7.5 (left). This method requires more computing time however it is stable even for faces that contain degenerated edges.

The user defines the desired angle between tangent vectors and the algorithm identifies knuckle lines between adjacent surfaces patches. The region identification based on knuckles would result in a quality mesh but is rather complex to implement for an automatic usage. The number of knuckles in a ship hull form is different from one ship to another. As shown in Figure 7.6 the number and position of knuckle lines are different with the same magnitude of user defined angle. This makes an automatic region identification based on knuckle line difficult. However manual region identification based on knuckle lines could be effective and has a potential to produce a quality mesh.

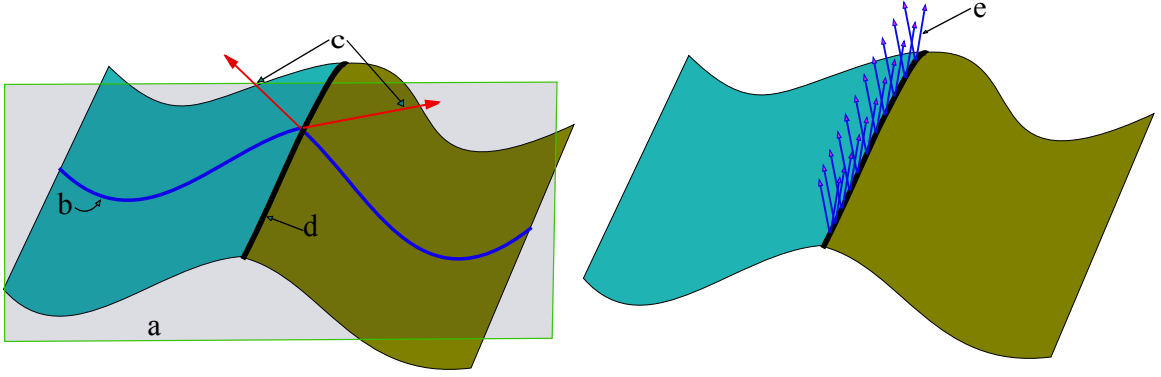


Figure 7.5: Determination of knuckle between two adjacent surface patches. The first method which used the normal vectors at the connection of the edges between adjacent surfaces (right) and the tangent based knuckle detection method (left). The letter a, b, c, d, e stand for cross-sectional plane, section curves, the tangents of the section curves, adjacent surfaces edge and surface normals.

For a perfectly modeled smooth and continues ship hull form, the angle between surface normals is zero. The decision whether the edge between two surfaces is a knuckle or not depends on an angle between normals. A case study was performed to identify an appropriate angle. Even though, what magnitude of angle should be considered as a knuckle or not depends on user's opinion. The case study suggests that the minimum angle to be considered as a knuckle angle is between 5° to 15° depending on the complexity of the model. Some representative results are shown in Figure 7.6 for two container ships at three different knuckle angles (3° , 10° and 15°). For small angle (3°) more knuckle lines are detected where as fewer and relevant knuckle lines are detected at 15° .

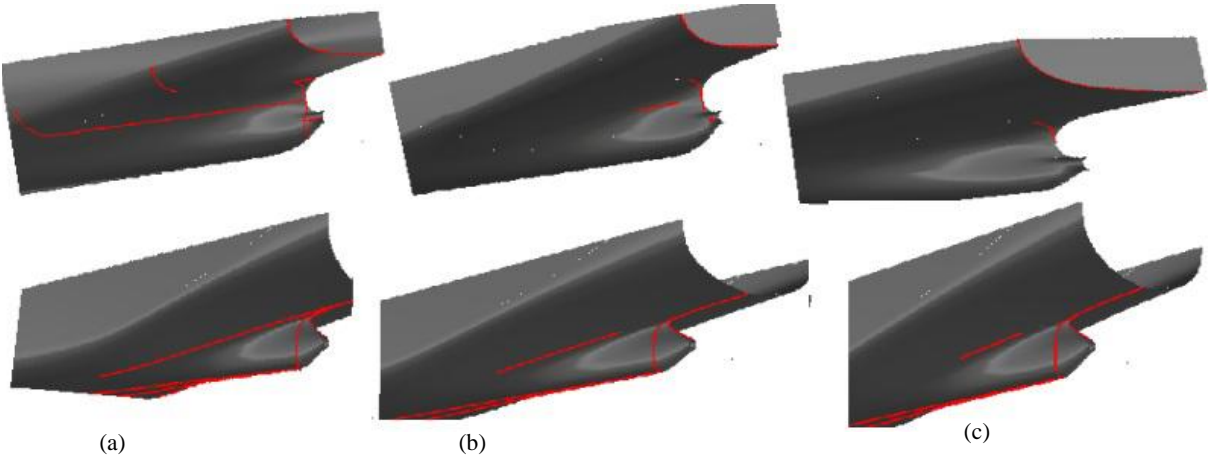


Figure 7.6: Ship hull form knuckles detected with different angles. a), b), c) show the detected knuckle lines with angles 3° , 10° and 15° respectively.

7.4 Domain Preparation for Downstream Applications

Usually the geometric requirements of downstream applications are beyond repaired, consistent and region identified CAD models. For instance the ship hydrodynamic computation requires not only a ship hull form but also the computational domain. In this section the computational domain based on the repaired and region identified hull form is developed for volume mesh generators (i.e. *Numeca HEXPRESS* and *snappyHexMesh*).

7.4.1 Domain for *SnappyHexMesh*

SnappyHexMesh is an open source volume mesh generator supplied with *OpenFOAM*. It generates meshes based on triangulated surfaces, and provides meshes which directly complies with the *OpenFOAM CFD* solvers. The domain preparation for *SnappyHexMesh* uses the deck clothing and mirroring functionalities discussed in chapter 6. These functionalities provide a watertight and consistent triangulated surfaces for the region identification which identifies region based on user's inputs. The developed technique provides three different types of region identification options. The first option separates the dry and wet region of the hull form based on an input draft as shown in Figure 7.7 (left). In the second option the user needs to define a coefficient which is used to separate the bow or the stern region based on the user's desire. This coefficient should always be between zero and unity. The option divides the hull form into three regions as shown in Figure 7.7 (middle). The third option divides the hull form into four regions (i.e. stern, middle, bow and dry) as shown in Figure 7.7 (right). This requires two coefficients (i.e. S_c, B_c) to divide the hull form. The magnitude of the two coefficients should be between zero and unity, and the summation should also be less than unity. After identifying the regions based on data range method, the regions are exported to a colored STL file format which is recognizable by *snappyHexMesh* or other CAD systems.

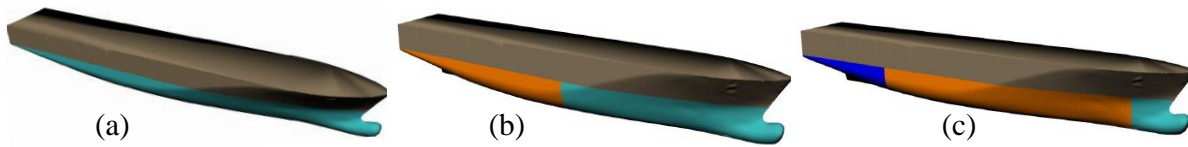


Figure 7.7: Ship hull form region identified for *snappyHexMesh*. Two regions (dry and wetted) identified (left), three regions (i.e. dry, bow and stern) identified (middle), four regions (dry, bow, middle and stern) identified (right).

7.4.2 Domain for *Numeca HEXPRESS*

Numeca HEXPRESS is a mesh generator software designed to generate unstructured hexahedral meshes for complex 2D and 3D geometries. In order to generate a mesh for the RANSE simulation with *Numeca HEXPRESS*, the bounding domain around the ship hull form has to be constructed. The bounding domain is used to represent the surrounding fluid domain for hydrodynamic analysis. The colored STL file format prepared for *Numeca HEXPRESS* should fulfill some specific requirements in order to generate a mesh without a problem. Some of the requirements are:

- the normals of all the triangles should point in or out of the computational domain;
- degenerate triangles should be removed;
- the computational domain should be continues and
- all adjacent facets must share two vertices.

In this section, an automatic computational domain preparation for ships CFD computation is developed for a double body and free surfaces analysis. The domain preparation is developed based on the repaired port side ship hull form. Two different methods are implemented to construct a ship hull form bounding domain automatically: the Boolean operation and direct construction method.

The Boolean operation method is used in CAD systems for different kind of operations. In this case, a box and the repaired hull form are separately prepared and subtracted from each other to get the bounding domain with the hull form as shown in Figure 7.8. Despite some successes,

this method is unable to handle different type of geometries automatically, because the Boolean operation always requires a consistent and error free geometrical representation of the models to provide a consistent result. This method is found to be not robust enough for an automatic construction of the computational domain.

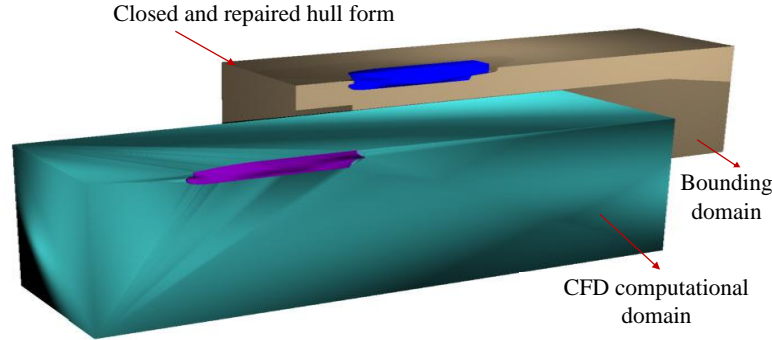


Figure 7.8: Boolean operation method domain construction

As a result an alternative method (i.e. direct construction) which could replace the Boolean operation is implemented. The direct construction method uses the repaired triangulated hull form as an input. Using the *LOA* of the ship and non dimensional user defined coefficient values the bounding domain inlet, outlet, side, bottom, centerline and top (free surfaces) are constructed automatically. If the coefficients are not defined by user, the algorithms considers 1, 2.5, 1.5, 1.2 and 0.5 for in front, behind, side, under keel, above the deck of the hull form respectively, see Table 7.1 and Figure 7.12. The side, bottom, inlet and outlet surfaces are very easy to construct automatically. However the development of a robust and an automatic construction of the free surface and centerline surface are not easy task as they are dependent on the quality and complexity of the ship hull form. The following procedures are followed to obtain the bounding domain for the double body CFD simulation:

- extracts the points on the free surface and the centerline of the wet region;
- calculates the bounding domain points on the free surface and the centerline from input coefficients;
- orders the points on the centerline and free surface in clockwise or anticlockwise orientation;
- triangulates the centerline and the free surface;
- combines inlet, outlet, side, bottom, free surface, centerline and wet part of the ship;
- checks the consistency of normals and orients consistently;
- writes a colored STL file format.

Figure 7.9 shows the way the centerline and waterline surfaces are constructed to achieve a consistent bounding domain. The construction process has two major sensitive areas: the construction of the centerline surface and the free surface. An automatic detection of the centerline and deck points is a challenging part of the method as it depends on the quality of the original design. Once the deck and centerline points are extracted, an automatic surface creation is possible. Figure 7.10 shows the result obtained from the developed techniques for double body and free surface bounding domain. The direct construction method can construct the domain for *Numeca HEXPRESS* automatically, provided the deck or waterline and centerline points are properly extracted. If not it creates gaps between the hull form and bounding domain, which needs further treatment. Despite the limitations, the direct construction is the promising method to construct the bounding domain automatically.

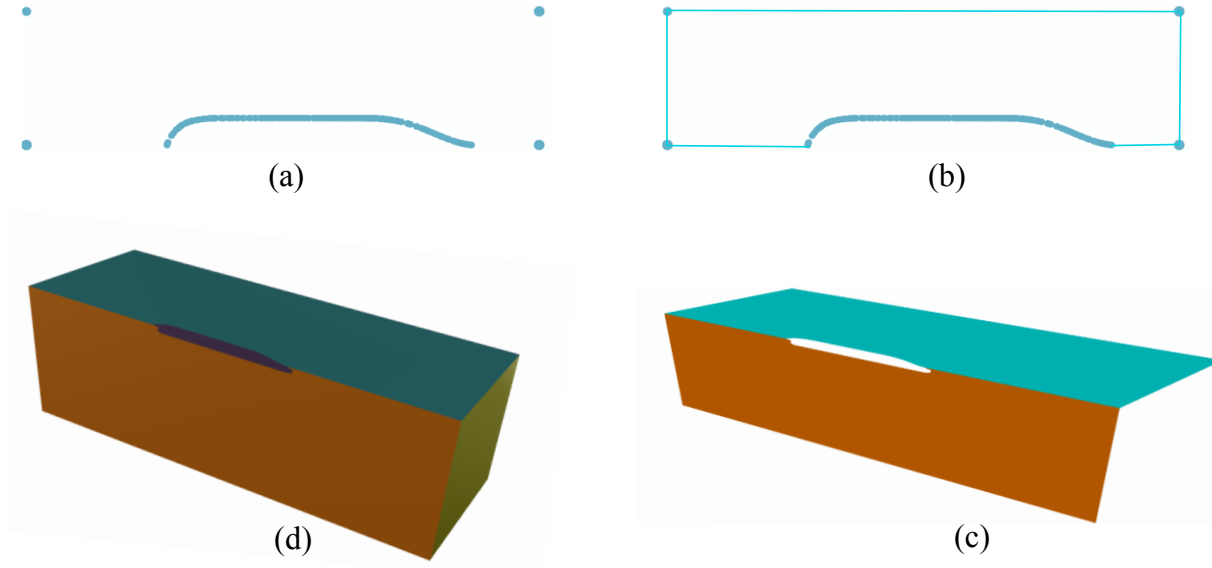


Figure 7.9: Direct bounding domain construction. a) The waterline points are extracted, b) the waterline points are ordered, c) using similar method the centerline points are detected and ordered, and d) finally all sides of the bounding domain surfaces and the hull form are combined.

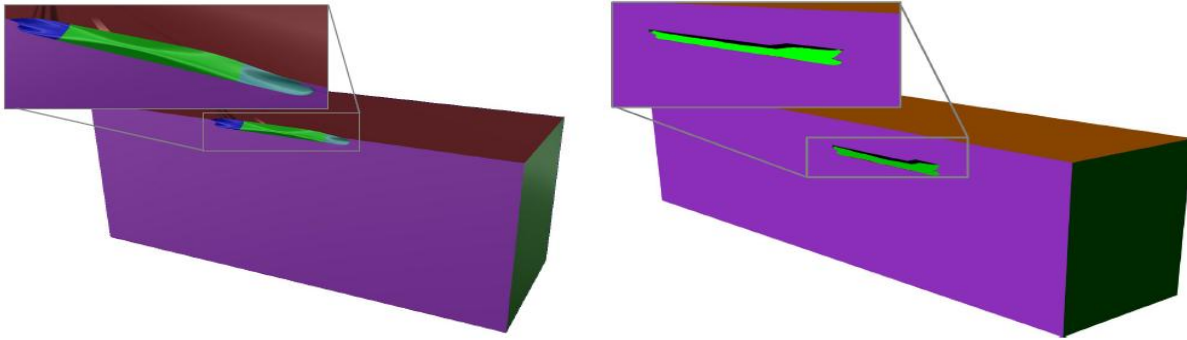


Figure 7.10: Double body computational domain (left) and free surface computational domain (right)

7.5 Output File Format

The developed framework is equipped with the functionality to export the repaired CAD model to IGES, STEP, STL, and OFF file formats. However, most of the downstream applications have the functionality to import STL file format. As a result it is considered as the primary target output file format. Whenever parametric surfaces are changed into triangulated surfaces, many geometric information get lost. Therefore, there are different techniques available to keep some information. In this work, a colored STL file format that can be imported by *snappyHexMesh* and *Numeca HEXPRESS* are studied and the functionality to export this colored STL file format is implemented. This colored STL file format is used to preserve and convey the identified regions and the prepared bounding domain.

7.5.1 Colored STL File Format for *SnappyHexMesh*

In colored STL file format, the identified regions are attached with their region names and this allows *snappyHexMesh* to treat those regions accordingly. In general, identifying regions into

different parts and save it into colored STL file format helps to improve the quality of the mesh. This type of colored STL file format is also recognizable by other CAD systems such as Rhinoceros.

For instance if one wants to write two triangles into different regions, the colored STL file format looks like as shown in Figure 7.11.

```

1 solid region_1 -----> Region name
2 facet normal n1x n1y n1z
3   outer loop
4     vertex x1 y1 z1
5     vertex x2 y2 z2
6     vertex x3 y3 z3
7   endloop
8 endfacet
9 endsolid region_1
10 solid region_2 ---> Region name
11 facet normal n2x n2y n2z
12   outer loop
13     vertex x1 y1 z1
14     vertex x3 y3 z3
15     vertex x4 y4 z4
16   endloop
17 endfacet
18 endsolid region_2

```

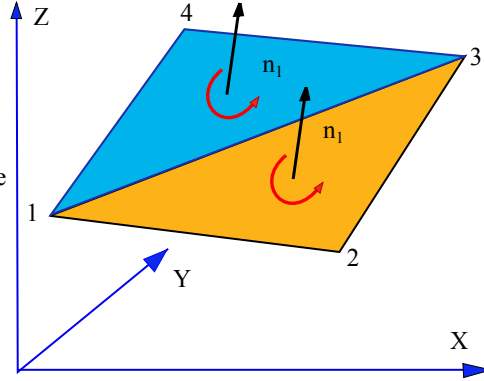


Figure 7.11: Colored STL file format for *snappyHexMesh* and other softwares. The two triangles are considered as two different regions (i.e. blue and yellow).

7.5.2 Colored STL File Format for *Numeca HEXPRESS*

The colored STL file format representation and requirements for *Numeca HEXPRESS* is different from that of *snappyHexMesh*. *Numeca HEXPRESS* expands the main STL file format (.stl) with another property file (.stl.prop) in order to use a full potential of its functionality of resolving regions and knuckles in the geometry by the mesh. The property file assigns each facet in the main STL file format to a certain region. This enables *Numeca HEXPRESS* to create an additional file storing geometry attribute for each triangle in the STL file format. The geometry attribute allows *Numeca HEXPRESS* to reconstruct topology information without the need of using a sophisticated and error prone feature detection algorithm. Figure 7.12 presents the colored STL file format illustrated in Table 7.1. In the figure there are seven regions (i.e. *ship hull form*, *centerline*, *free surface*, *inlet*, *outlet*, *side*, and *bottom*) written into colored STL file format. The property file (*file.stl.prop*) corresponds to each regions in main STL file format (*file.stl*). Therefore *Numeca HEXPRESS* considers those regions as a separate groups of objects, see Table 7.1.

7.6 Results and Discussions

The proper region identification and domain preparation have enormous relevance in performing an efficient, and time saving mesh generation and then hydrodynamic computation. The region identification based on the data range method and the domain preparation for *snappyHexMesh* and *numeca HEXPRESS* are implemented for automatic application. The region identification

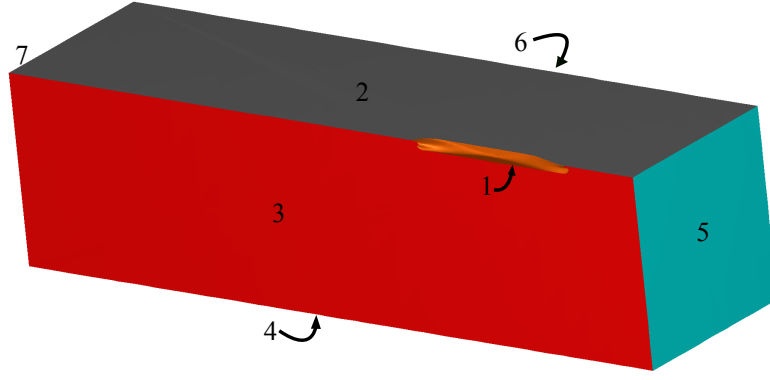


Figure 7.12: Colored STL file format for *Numeca HEXPRESS*.

based on the curvature and knuckle detection needs user interaction which needs visual observation.

It is essential to ensure whether the produced results from the developed framework could be successfully used in downstream application or not. As a result the colored STL file format exported from the developed framework is experimented for *snappyHexMesh*. The experiment is made using the DTC container ship hull form. This hull form is imported from IGES file format to the developed framework, afterwards the repairing, region identification, and domain preparation processes are performed and the treated geometry is exported to colored STL file format as shown in Figure 7.13. The main dimensions and other information of this test case can be found in [187].

Mesh generation and hydrodynamic computation is demonstrated using an opensource CFD-Toolbox OpenFOAM which contains several tools for mesh generation and CFD computation. It includes two grid generation tools (i.e. *blockMesh* and *snappyHexMesh*). This work used an automatic mesh generation process (Python-script) called *hullMesh* developed based on an *OpenFOAM* tool *snappyHexMesh* [188]. The script reads the colored STL file format and performs finite-volume-mesh of the surrounding fluid domain of the ship. Figure 7.14 shows the mesh generated for DTC hull form with different refinement value using *hullMesh*. Using the advantage of region identification, the wet and dry parts of the hull are meshed with different refinement levels as shown in Figure 7.15.

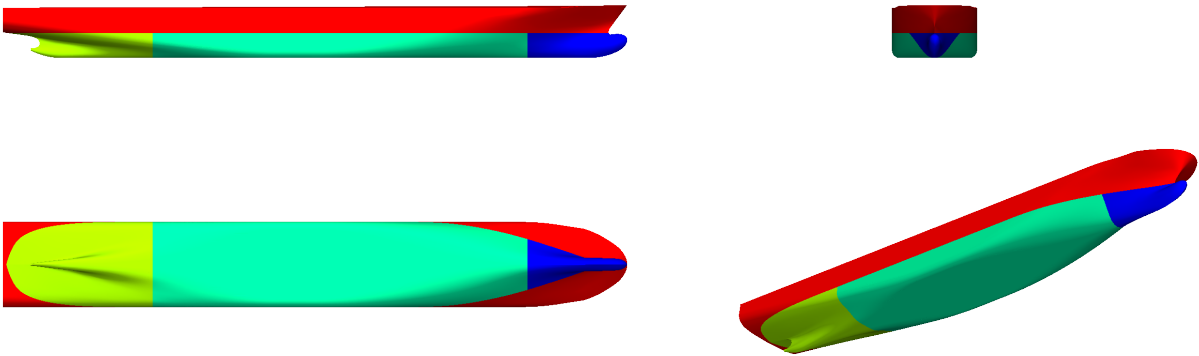


Figure 7.13: DTC-Container ship hull form repaired and ready for mesh generation in *snappyHexMesh*

Table 7.1: The colored STL file format for *Numeca HEXPRESS* (the original file format and the corresponding property file.)

Original file (file.stl)	Property file (file.stl.prop)
Header	Total number of triangles in the file
Triangles on ship hull form	1
Triangles on free surface	2
Triangles on centerline	3
Triangles on bottom	4
Triangles on inlet	5
Triangles on side	6
Triangles on outlet	7

The original file contains the information about STL file format such as triangle vertexes and normals, and the property file separates triangles based on their respective regions and it gives corresponding numbers for each region triangles see Figure 7.12

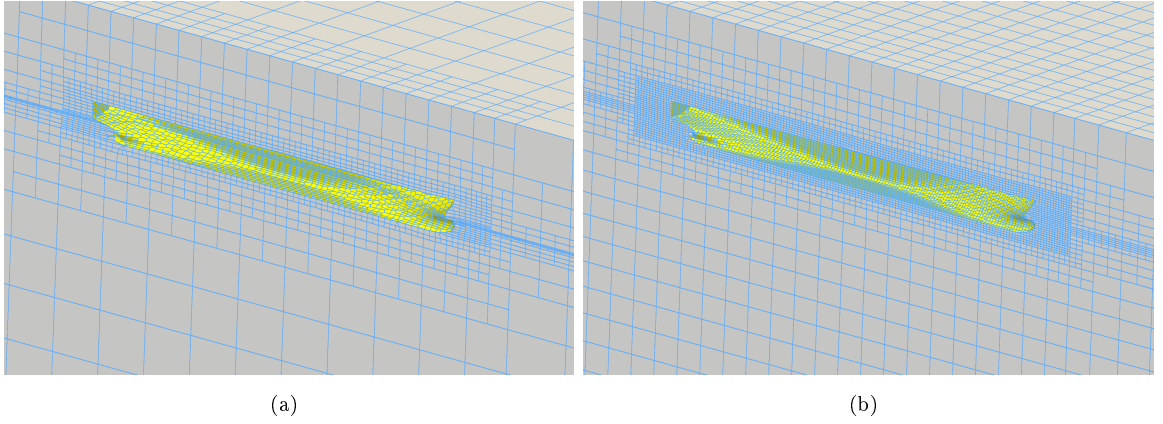


Figure 7.14: Mesh generated based on the framework output. Different refinement level a) with *subdivision* = 8 and b) with *subdivision* = 16

Preliminary DTC ship resistance calculation is performed in model scale. The generated final mesh for this RANS-calculation has four million cells with the free surface refined according to the Froude number. At the ship boundary a layer is inserted with $y^+ = 30$. The obtained results are compared with the experimental results found from [187].

The simulation itself is started with *LTSinterFOAM* and when a steady-state condition of the calculation is found, the solver is changed to *interDyMFOAM* to include trim and sinkage. Due to the 2-DoF calculation the domain is mirrored at the symmetry plane. Figure 7.16 shows the wave elevation of the vessel at $Fr = 0.218$. The simulation result of the total resistance is plotted against the simulation time as shown in Figure 7.17. As it can be seen the simulation results are shown good agreement with experimental results.

7.7 Summary

Repairing 3D models has high importance in decreasing the time and resources required to integrate the CAD models into downstream applications. However downstream applications demand more than that. The variety of the input geometry requirements from one downstream

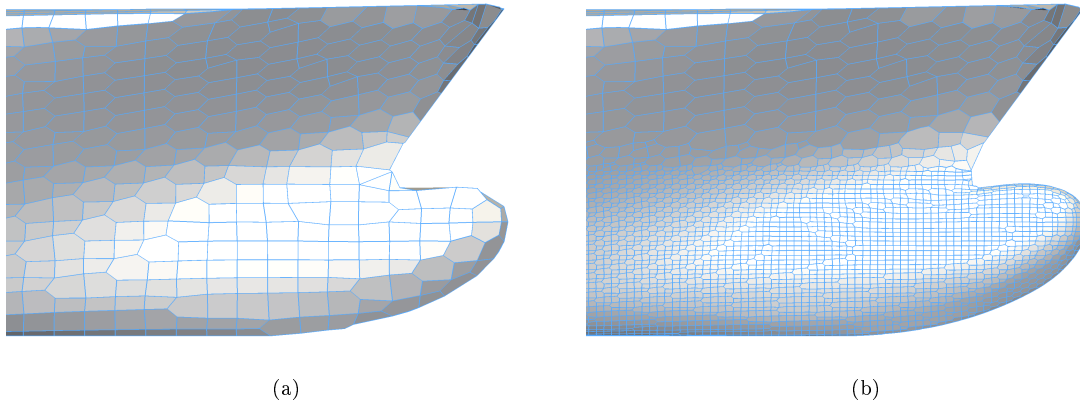


Figure 7.15: Different mesh refinement level for different regions.

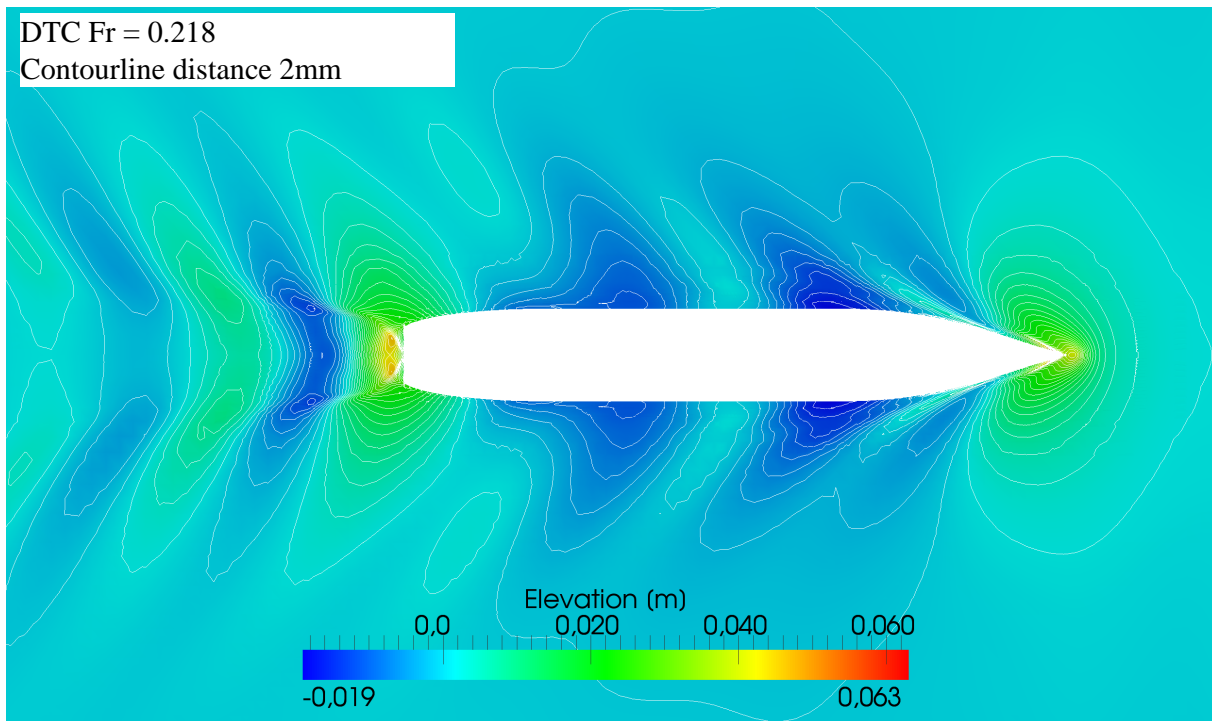
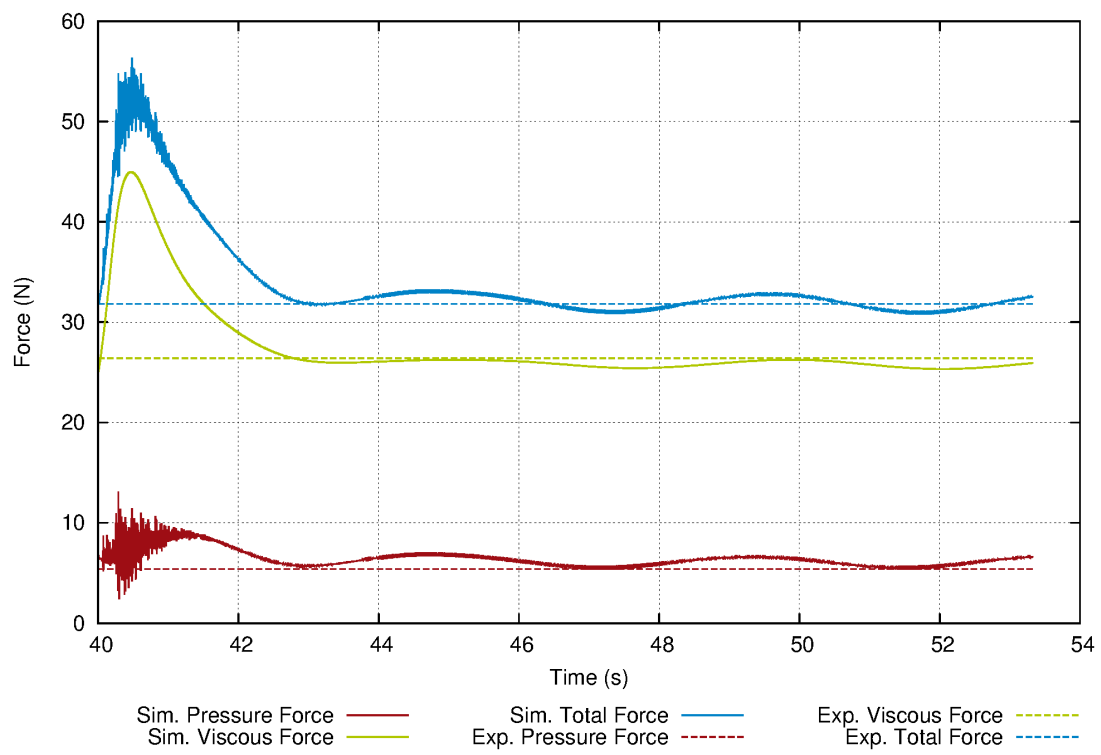


Figure 7.16: Wave elevation of DTC ship in model scale at $Fr = 0.218$

application to another makes the CAD model integration process challenging. This section presents a hull form region identification and domain preparation strategy for hydrodynamic computation. It enables the users to control the construction of the bounding domain and mesh generation steps with few controlling variables, which saves the meshing and mesh manipulation time. The demonstration of ship hydrodynamic computation based on the repaired, region identified and domain prepared hull form model is made. In general the development of CAD model repairing, region identification, domain preparation for downstream applications saves considerable time.

Figure 7.17: Total resistance of DTC in model scale at $Fr = 0.218$ over simulation time

This page intentionally left blank.

CHAPTER 8

INTEGRATION AND WORK-FLOW

8.1 Introduction

There are multiple situations in engineering where a combination of different techniques is needed to create a complete design process from which RE is the typical example. RE consists of several engineering tasks from the initial data acquisition to the stage of final design tasks. The RE process can be roughly categorized into three stages: pre-processing, surface reconstruction, and post-processing. The pre-processing includes the data acquisition systems and data processing techniques. This stage is the most important part of the RE process because it has a major impact on the quality of the RE output. The surface reconstruction stage is the process used to convert point data to surface representations (e.g. NURBS, B-Spline, triangular or other polygons). In addition to the capabilities of the surface reconstruction approaches, the pre-processing stage capabilities and its output qualities have a major impact on the output quality of the reconstructed surfaces. In most cases, the output of the surface reconstruction processes could not be directly used by downstream applications. Due to this reason, post-processing techniques are required to adapt the output of the reconstruction algorithms to the specific requirements of downstream applications. In this thesis the RE process and CAD processing framework is developed aiming to reconstruct and to prepare a ship hull form CAD model for downstream applications. In this chapter, a comprehensive system integration of the developed framework is discussed with the help of representative illustrations. It mainly focuses on the analysis of the developed framework techniques and their relationships as well as the intermediate and final output. Because, system integration is as important as the development of individual techniques capability and efficiency.

8.2 The Framework Components

The developed framework consists of two major engineering parts: RE and CAD processing. The RE part proposes three alternative ship hull form surface reconstruction techniques from an unorganized point cloud and CAD processing aims to fulfill the geometric requirements of the downstream applications. The major components and the work-flow of the developed framework is illustrated in Figure 1.2. The three alternative RE options are: curves network, direct surface fitting and triangular surface reconstruction methods which are found suitable for ship hull form RE. All of them roughly consist of three modules (i.e. pre-processing, reconstruction and post-processing), and they share most pre-processing algorithms such as outliers removal, noise reduction, down-sampling. Their main differences lie in their surface reconstruction techniques and their output. The curves network RE approach aims to reconstruct B-Spline curves from an unorganized point cloud which later fitted to surfaces while direct surface fitting RE approach reconstructs NURBS surface directly from an unorganized point cloud without any prior parameterization. The triangular surface RE is the most employed approach over the last decades and many different methods are proposed. Therefore six different suitable methods are selected for implementation. The selected methods are evaluated against different criteria and integrated into

the developed framework. The CAD processing part consists of two stages: surface healing, and region identification and domain preparation. It takes the input from the RE process or it reads from an existing geometry. Common topological and geometrical inconsistencies are repaired by the surface healing part and the rest requirements of the downstream applications are met in the region identification and domain preparation stage.

8.3 The Components Integration

This section roughly explains how the different techniques are integrated and how they operate towards a common goal.

To illustrate how the different techniques and modules are connected and what intermediate results are produced, two representative cases are considered. The first case reads an unorganized point cloud and follows the process which produces successive intermediate results (i.e. noisy data \rightarrow noise eliminated \rightarrow down-sampled \rightarrow reconstructed surface \rightarrow region identified surface \rightarrow generated mesh) to achieve the objective, as successively shown in Figures 8.1, 8.2, 8.3. This illustration is based on artificially produced point cloud with introduced random noise (gray color) as shown in Figure 8.1 (upper). The introduced random noise is eliminated using statistical outliers removal and noise elimination approaches described in chapter 2 as shown in Figure 8.1 (middle). Due to a large size of the point cloud ($LOA = 324m$ of a container ship), the voxel grid noise elimination and down-sampling approach is used to reduce the points from 2290894 to 18918 points with leaf size (grid resolution) of $1m$ as shown in Figure 8.1 (below). This helps to decrease the normal estimation and reconstruction time.

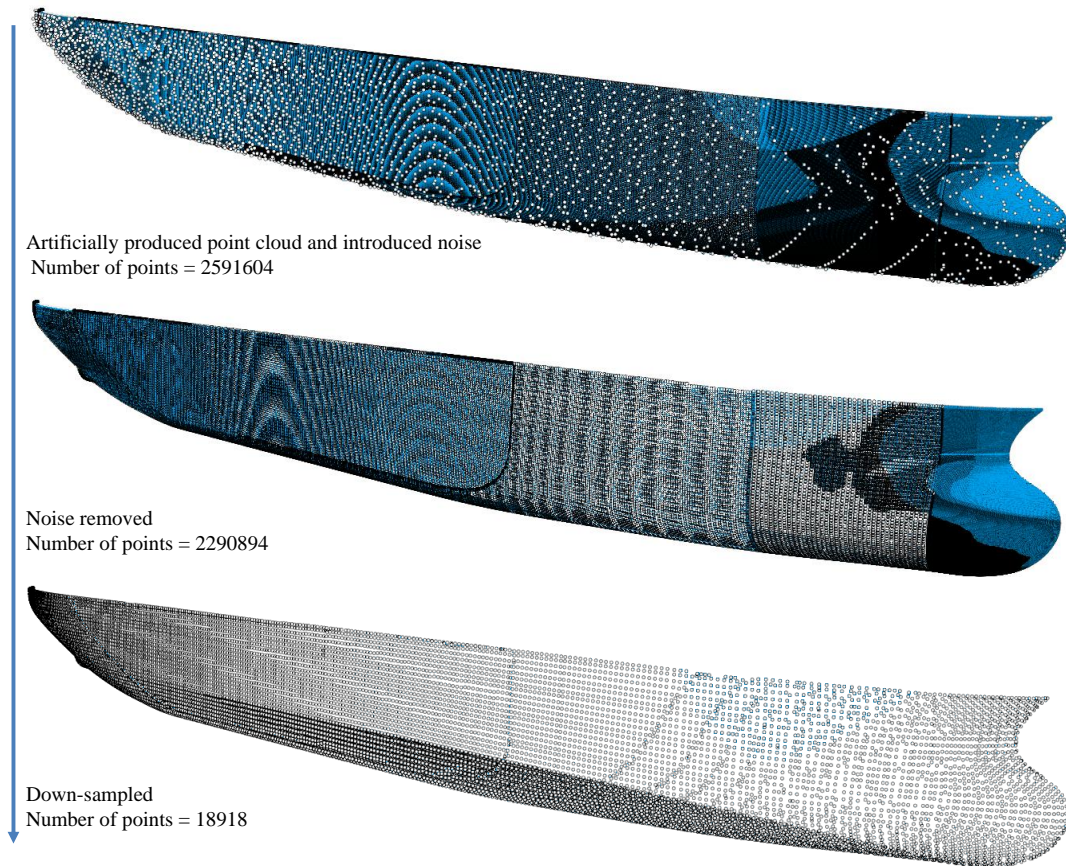


Figure 8.1: The pre-processing stage of the framework.

In this illustration two triangular surface reconstruction approaches are considered. The ball pivoting and Poisson methods as shown in Figure 8.2(a) and (b) respectively. The ball pivoting and Poisson are the two distinct reconstruction approaches, as the ball pivoting interpolates most of the input points and Poisson approximates the points. The ball pivoting does not require normal estimation while the Poisson is even very sensitive to normal orientation. Ball pivoting preserves sharp edges while the Poisson smooths out. Poisson is robust to noise while ball pivoting is sensitive to noise. Even though they have a contradicting performances, ball pivoting and Poisson approaches are found to be suitable for the ship hull form reconstruction out of the implemented triangular surface reconstruction approaches.

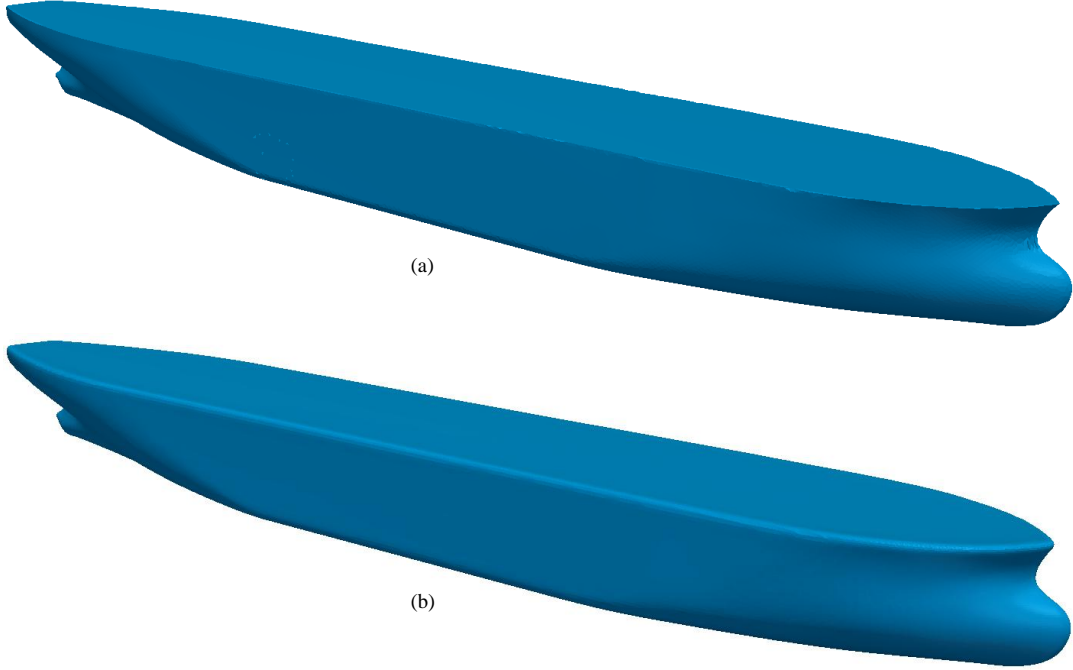


Figure 8.2: The reconstruction stage of the framework. a) reconstructed using ball pivoting method, b) reconstructed using Poisson method.

Once a consistent triangular surface is reconstructed from an unorganized point cloud, it can be used e.g. for hydrodynamic computations of the ship. In this case, the hull form reconstructed using Poisson method is further processed towards hydrodynamic computation. The hull form is divided into different regions, first divided into wet and dry regions and then the wet region is further divided into three parts (i.e. bow, middle and stern regions) as shown in Figure 8.3 (a). The region identification algorithm is developed to enable the definition of mesh refinement variables based on different regions. Once the hull form regions are identified, they are exported to colored STL file format. The colored STL file format is imported to *OpenFOAM's SnappyHexMesh* to generate a mesh as shown in Figure 8.3 (b). As shown in the Figure 8.3 (b) and (c) the mesh is generated with different refinement values for different regions with smooth transition between different regions.

The second illustration is made based on the scanned model ship. The ship is scanned from different directions and the raw point clouds are transformed into the same coordinate system (registered) as shown in Figure 8.4. After the point cloud patches from different measurement views are registered, one side of the hull is separated using centerline symmetry for further processing. Due to incapability of the measurement device and environments, the acquired point cloud has missing data specially at the flat of side of the hull form. To solve this problem, up-sampling is used to fill the missing data using prior knowledge (i.e. flat of side is a plane). After the missing data is filled, the points are regularized to fit the NURBS surface to the point cloud.

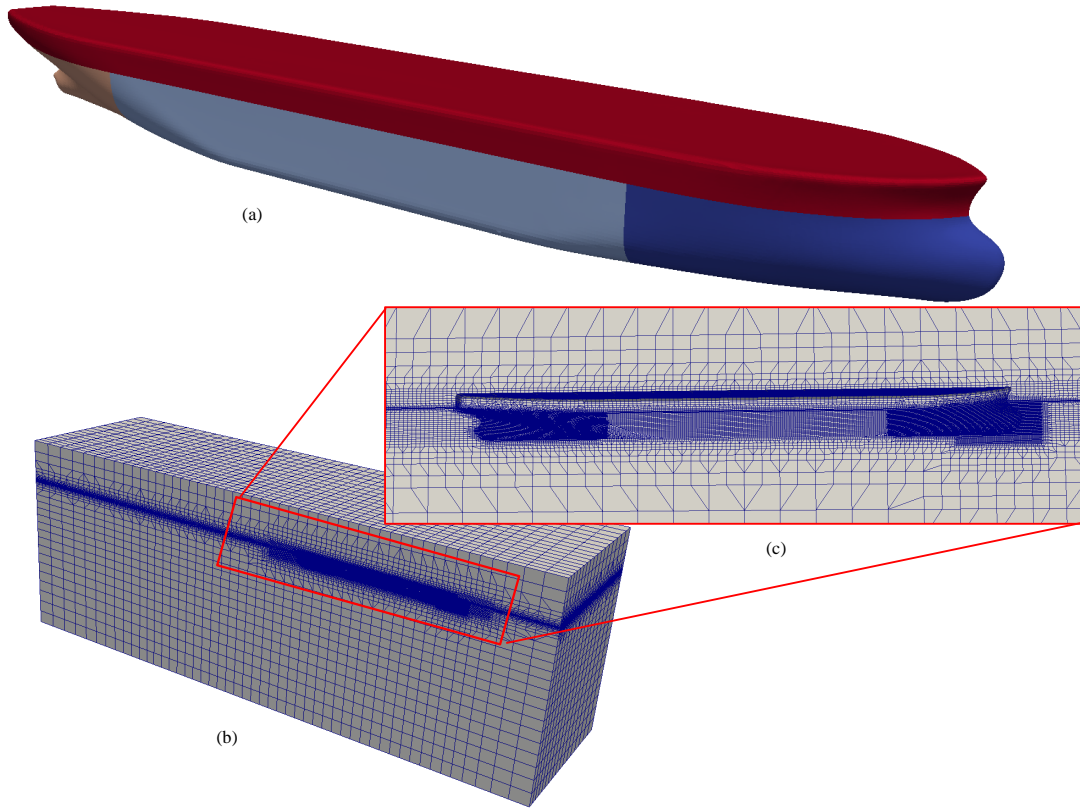


Figure 8.3: Hull form with region identified and mesh generated based on it. a) hull form divided into four regions b) mesh generated with different resolution to different region, c) zoomed mesh around the hull form.

Usually the produced NURBS patches have geometrical and topological inconsistencies that have to be repaired in order to integrate the geometry into numerical simulation. Therefore the NURBS patches are repaired and triangulated as shown in Figure 8.4. In general the developed framework consists of many algorithms which can be combined together to build a specific workflow to provide the best suitable solution for specific applications.

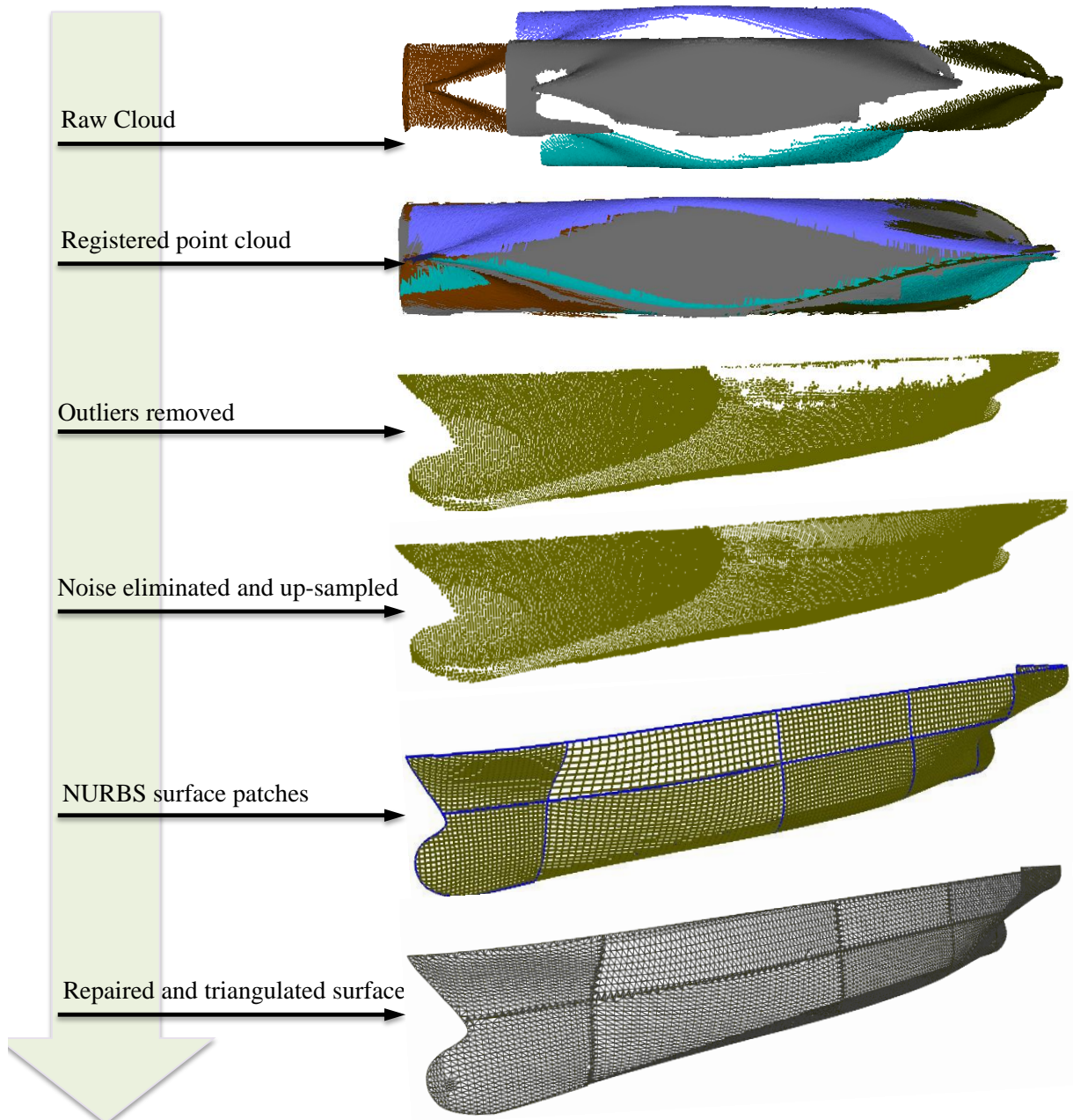


Figure 8.4: Hull form surface reconstruction from an unorganized point cloud. Direct NURBS surface reconstruction and preparation for downstream application.

This page intentionally left blank.

CHAPTER 9

DISCUSSION AND CONCLUSION

9.1 General Discussion

RE involves several strategies, technologies and mathematical numerics from data acquisition systems to reconstructed models. These make the RE methods computationally expensive, time consuming and more over error-prone. In addition most RE approaches are developed for general purposes and in most cases are not efficient to solve specific problems. In order to achieve the required level of quality with a reasonable associated time and resources, the development of an application specific RE method is indispensable. This thesis proposes three hull form specific RE techniques such as curves network, direct surface fitting and triangular surface reconstruction. The techniques are systematically implemented so that they deliver an improved level of quality. The implementations do not only include mathematical reconstruction techniques but also the integration of data, algorithm, spatial and scene knowledge. Neither RE nor FE techniques are error free rather they usually produce a CAD model which fails to meet the input requirements of downstream applications. As a result the CAD data repairing, region identification and domain preparation approaches are developed to prepare the reconstructed or a newly designed ship hull form for downstream applications specifically towards hydrodynamic computations. Both the RE techniques and CAD data repairing, region identification and domain preparation are integrated into a framework aiming mainly to bridge the acquired point cloud data to the downstream applications. Before proceeding to the development of the framework, the following important factors are carefully studied:

- characteristics of input data;
- traditional hull form design procedures;
- existing surface reconstruction techniques capabilities and their suitability to a hull form reconstruction application;
- knowledge about a hull form shape;
- geometric requirements of downstream applications.

The developed framework lays its foundation on these factors, which help to develop the best suited techniques for the intended application.

9.2 Detail Discussion and Conclusion

The framework consists of five distinct components or subsystems presented in chapters 3 to 7. This section discusses the proposed hull form specific techniques and draws the conclusions as follow.

- The proposed curves network hull form RE technique offers the possibility to reconstruct ship lines plans from an unorganized point cloud. With a receipt of consistently aligned point cloud from the various patches of the 3D point cloud data views, the develop methodology provides an efficient and time saving reconstruction strategy. It extracts the transversal sections, waterline sections, centerline and deck curves from an unorganized point cloud

using either B-Spline curves interpolation or approximation. This method has two main reasons to be preferred for a hull form RE. The first reason is the fact that a curves network (lines plan) hull form representation is the state of the art in maritime industry and the end users (i.e. naval architects) are well experienced in this regard. The second reason is that the method is technically efficient and time saving as it quickly reduces a large size of an input unorganized point cloud to a manageable size. This is suitable for a large scale objects like the ship hull forms. The efficiency and time saving characteristics of the curves network RE technique are attributed to its capability to handle noisy and large data sets; unnecessary to compute the local properties (i.e. surface normal and curvature) and parametrization (i.e. triangulation and surface patches) based on a point cloud which are always computationally expensive and error-prone, and ability to approximate or interpolate the curves network semi-automatically. In general, the curves network hull form RE approach enables to align the hull form surface reconstruction from an unorganized point cloud into the traditionally well developed lines plan based surface construction. The well developed NURBS surface generation approaches based on the curves network in the CAD systems also make the curves network hull form RE technique preferable and easier to be used.

- The direct surface reconstruction method fits a NURBS surface to an unorganized point cloud. It is computationally efficient and time saving as it avoids the prior parameterization such as triangulation and NURBS patch networks. However, the direct NURBS surface fitting requires a thorough point cloud pre-processing including outliers removal, noise elimination, data down-sampling and more importantly regularization. Therefore, the developed technique incorporates these pre-processing functionalities. Having the segmented and regularized point cloud, it is easy to fit standard NURBS surface patches. The framework also includes post-processing techniques which trim unnecessary surfaces and resolve topological problems.
- Compare to the curves network and direct surface fitting reconstruction methods, the triangular surface reconstruction approach is computationally expensive in terms of a computer memory and processing time as it includes complicated procedures of a refinement, parametrization and maintainability of the continuity between networks of triangular surface patches. However, there are cases where the triangular surface reconstruction approach is preferable. For instance most mesh generators and computational analyses usually offer the functionality to operate based on triangular surface representations. In this case, using curves network or direct NURBS surface fitting is not efficient as the process is in a sense of repeating itself and comes back to where it is started but with deviations or inaccuracies far in excess of the original scanned point cloud. On the other hand, the triangular surface reconstruction is the most employed method with many techniques proposed over the last decades. There exist various triangular surface reconstruction approaches which range from those capable to reconstruct simple shapes to those able to handle complex shapes and data imperfections. They are also different in surface representations (i.e. explicit and implicit), the fitting approaches (i.e. interpolation and approximation) and the nature of the input data (i.e. only points, unoriented normal and points, oriented points). However the main difference lays in their capability to handle the different aspect of the point cloud imperfections. In this work six previously proposed triangular surface reconstruction approaches are studied and evaluated against quality, capability and usability criteria. The evaluation considers an ideal reconstruction techniques, for instance the best technique does not require normals, robust to noisy data, ables to manage missing and large data, is capable to reconstruct a complex shape, preserves sharp features like knuckles while guaranteeing topology and smoothness, produces both interpolating and approximating surfaces, reconstructs a geometrically accurate model, and is user friendly and time saving. However, none of the reconstruction techniques considered in this study satisfies all the criteria. This reveals

the strong and weak sides of the considered methods which opens the door to work toward the improvement of their capability, quality and usability for the specific application. The evaluation results help to identify the requirements on the input data, the working principle, the driving input parameters, and the quality of the output. These provide a knowhow about which additional techniques should be integrated in order to improve the quality, capability and usability. Based on these evaluation results: the triangular surface reconstruction approach which consists of the pre-processing, surface reconstruction, post-processing and knowledge integration is developed. In addition the requirements of scene, spatial, data and algorithm knowledge are investigated for each mathematical techniques and possible prior knowledge are incorporated to recover as much information of the shape as possible. The basic knowledge required by the users is identified for each technique implemented for the reconstruction process. In addition the way to enforce a hull form related knowledge such as centerline, flat of side, and flat of bottom is devised. In general the knowledge based reconstruction approach improves the capability, quality and usability of the involved techniques. Among the six triangular surface reconstruction methods implemented in the developed framework, the ball pivoting and Poisson methods are found to be suitable for the ship hull form reconstruction. In general, the developed strategy provides different alternatives to reconstruct a hull form, not only alternatives but also the way to improve the output results.

- The target outputs of the RE part of the developed framework are triangulated and parametric surfaces. These representations usually require further post-processing due to different input requirements imposed by the downstream applications. The adaptation of the CAD model to the requirements of the downstream applications is known to be tedious, time and resource consuming. As a result, the ship hull form specific CAD data processing strategies toward downstream application are developed to reduce the amount of time and cost associated with the CAD data processing. The strategy includes surface healing, regions identification and domain preparation. Surface healing consists of two parts: parametric and triangular surface repairing. The parametric surface healing reads geometries from IGES file format and automatically repairs commonly found topological and geometrical inconsistencies. It is partially tolerance driven and consists of two main parts: shape fixing and sewing. Shape fixing aims to resolve inconsistencies in entities while sewing aims to maintain continuity between entities. The triangular surface healing repairs the CAD model read from the STL file format. The sources of the geometries can be the output of the RE part or any possible FE sources. The algorithm is tested against several test cases in order to check the consistency and robustness. The achieved result reveals that the developed surface healing strategy substantially decreases the time, and therefore cost required to repair the surface inconsistencies.
- Ensuring a consistent, watertight and continuous CAD model representation usually enables an automatic mesh generation but does not guarantee a usable mesh. Each downstream applications (i.e. mesh generators, CFD, FEA, etc.) has their own specific input requirements based on specific applications. This makes the attempt to develop a common CAD model adapting methodology rather complicated and bulky. Furthermore such an attempt can be inevitably incomplete as particular cases and new applications arise too frequently. However, it is possible to develop the region identification and domain preparation strategies for a specific downstream application. In this work, different ship hull form region identification methods are proposed from which the data range method is implemented for automatic purposes. Based on the repaired triangular surface, region identification and domain preparation strategies are developed for two volume mesh generators such as *Numeca HEXPRESS* and *snappyHexMesh*.

In general the framework offers different ship hull form specific reconstruction approaches with

pre-and post-processing functionalities including knowledge integration. These make the developed framework effective and time saving. The surface healing, region identification, and domain preparation strategies of the developed framework are applied to ensure the consistency of the CAD model constitutes (i.e. geometry, topology and tolerance). It does not only provide a consistent output representation but also saves the associated time and resources.

9.3 Limitations

The developed framework presents different solution alternatives to the RE and geometric processing problems so that they provide a consistent output for downstream applications. Even though efforts have been made to recover the shape of hull forms as accurate as possible, the framework still needs improvements when it comes to preserving features such as holes, knuckles, boundaries and high curvatures. These challenges are tackled using different RE techniques including knowledge integration, however they have their own shortcomings. For instance, the curves network RE technique is unable to preserve holes. In addition, the curves network and direct surface fitting RE approaches are poor in preserving knuckles and sharp edges. The triangular surface RE techniques considered in this thesis have also their own specific limitations. Alpha shapes, ball pivoting and greedy projection triangulation interpolate most of the points and are able to preserve features but too sensitive to point cloud inconsistencies such as noisy, varying and missing data. On the other hand the triangular surface reconstruction methods categorized under the implicit surface representation such as grid projection, marching cubes and Poisson methods approximate the points and are robust to point cloud inconsistencies but poor in preserving features.

The developed framework considers a registered point cloud as an input, however a large scale object (e.g. hull form) requires measurement from different view points. Therefore, the integration of robust registration algorithms is inevitable.

9.4 Further Work

Despite many contributions to the hull form RE and CAD data processing, the proposed framework has a broad avenue for further works. The future work will focus on equipping the framework with the integration of existing methodologies, potential improvements and new developments.

There are many RE approaches proposed over the last decade claiming to reconstruct objects. These approaches should be further studied, evaluated and implemented to the specific application of the framework. This includes registration, robust normal estimation, feature sensitive reconstruction approaches, smoothing and fairing algorithms.

Potential improvements are also desirable to upgrade the algorithms included in the framework so that they are more robust to the variation of an input data quality, and also capable to preserve hull form features. The curves network and direct surface fitting RE techniques should be improved with feature sensitive reconstruction algorithms and knowledge based RE approaches. Direct surface reconstruction is poor in handling closed and occlude surfaces. Therefore, methods should be investigated to solve this problem.

Generally the geometric based reconstruction approaches categorized under an implicit surface representation have provable reconstruction guarantee, while those under an explicit surface representation can produce accurate interpolating and approximating surfaces with sharp features. Therefore, the combination of these techniques might lead to an ideal solution.

The integration of knowledge in the RE process requires to know the data feature, object properties and scene characteristics. In addition, it also requires step by step visual observation to enforce knowledge. Therefore, a graphical user interface for the developed framework is desirable.

This page intentionally left blank.

BIBLIOGRAPHY

- [1] National Shipbuilding Research Program (NSRP) ASE. Ship check data capture follow-on project(NSRP ASE 05-01), final report, march. Internet, 2007.
- [2] H. J. Koelman. Application of a photogrammetry-based system to measure and re-engineer ship hulls and ship parts: An industrial practices-based report. *Computer-Aided Design*, 42(8):731–743, 2010.
- [3] M. Krause, M. Hübler, F. Roland, and M. Müller. Efficient retrofitting: How planning tools and reverse engineering methodologies can improve repair shipyards’ performance. In *Transport Research Arena, Paris*, 2014.
- [4] ARCHIE-WeSt. CFD simulations to investigate the effect of retrofitting technologies to improve the energy efficiency of bulk carrier. Internet, 2014.
- [5] F. Menna, E. Nocerino, and A. Scamardella. Reverse engineering and 3D modelling for digital documentation of maritime heritage. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume Xxxviii-5W16, ISPRS Trento, Italy*, pages 245–252, 2011.
- [6] F. Menna, E. Nocerino, S. Del Pizzo, S. Ackermann, and A. Scamardella. Underwater photogrammetry for 3D modeling of floating objects: The case study of a 19-foot motor boat. In Enrico Rizzuto and Carlos Guedes Soares, editors, *Sustainable Maritime Transportation and Exploitation of Sea Resources*. Taylor and Francis Group, London, 2011.
- [7] T. kanzok, F. Süß, L. Linsen, and P. Rosenthal. Efficient removal of inconsistencies in large multi-scan point clouds. In *21th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen, 2013*, pages 120–129, 2013.
- [8] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, and C. T. Silva. State of the Art in Surface Reconstruction from Point Clouds. In Sylvain Lefebvre and Michela Spagnuolo, editors, *Eurographics 2014 - State of the Art Reports*. The Eurographics Association, 2014. doi: 10.2312/egst.20141040.
- [9] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer. Parameterization-free projection for geometry reconstruction. *ACM Transactions on Graphics*, 26(3):22, 2007.
- [10] P. Mullen, F. De Goes, M. Desbrun, D. Cohen-Steiner, and P. Alliez. Signing the unsigned: Robust surface reconstruction from raw pointsets. In *Computer Graphics Forum (Proc. of the Symposium on Geometry Processing)*, 2010.
- [11] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing*, pages 61–70, 2006.
- [12] A. C. Öztireli, G. Guennebaud, and M. Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer Graphics Forum*, 2009.
- [13] M. Kazhdan. Reconstruction of solid models from oriented point sets. In *Eurographics Symposium on Geometry Processing*, 2005.
- [14] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. Voronoi-based variational reconstruction of unoriented point sets. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, 2007.
- [15] A. Hornung and L. Kobbelt. Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.
- [16] T. K. Dey, X. Ge, Q. Que, I. Safa, L. Wang, and Y. Wang. Feature-preserving reconstruction of singular surfaces. *Computer Graphics Forum*, 31(5), 2012.
- [17] A. Tagliasacch, H. Zhang, and D. Cohen-Or. Curve skeleton extraction from incomplete point cloud. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2009.
- [18] R. Schnabel, P. Degener, and R. Klein. Completion and reconstruction with primitive shapes. *Computer Graphics Forum (Proc. of Eurographics)*, 28(2):503–512, 2009.
- [19] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas. Discovering structural regularity in 3D geometry. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2008.

- [20] Y. Li, X. Wu, Y. Chrysatho, A. Shar, D. Cohen-O, and N. J. Mitra. Globfit: Consistently fitting primitives by discovering global relations. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2011.
- [21] Y. Lipman, D. Cohen-Or, and D. Levin. Error bounds and optimal neighborhoods for mls approximation. In *Eurographics Symposium on Geometry Processing*, 2006.
- [22] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011.
- [23] R. B. Fisher. Applying knowledge to reverse engineering problems. In *Geometric Modeling and Processing 2002 Proceedings*, pages 149–155, 2002.
- [24] R. B. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Institut für Informatik der Technischen Universität München, 2009.
- [25] S. Arya and D. M. Mount. Algorithms for fast vector quantization. In *Proceedings of Data Compression Conference*, pages 381–390, 1993.
- [26] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, February 5-8*, pages 381–390, 2009.
- [27] J.-F. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert. Scale selection for classification of point-sampled 3D surfaces. In *Proceedings of the 5th International Conference on 3D Digital Imaging and Modeling (3DIM), Ottawa, Ontario, Canada, June 13-16*, pages 285–292, 2005.
- [28] N. J. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, 2003.
- [29] A. Nurunnabi, G. West, and D. Belton. Robust outlier detection and saliency features estimation in point cloud data. In *2013 International Conference on Computer and Robot, May 28-31*, 2013.
- [30] D. M. Hawkins. *Identification of Outliers*. Springer Netherlands, Chapman and Hall, London, 1980.
- [31] T. Johnson, I. Kwok, and R. T. Ng. Fast computation of 2-dimensional depth contours. In *Proc. KDD*, 1998.
- [32] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [33] E. M. Knorr., R. T. Ng, and V. Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal Springer-Verlag*, 8:237–253, 2000.
- [34] M. M. Breunig, H. Kreigel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proc. ACM SIGMOD 2000 Int. Conf. On Management of Data, Dallas*, 2000.
- [35] S. Sotoodeh. Outlier detection in laser scanner point clouds. In *IAPRS Volume XXXVI, Part 5, Dresden 25-27 September*, 2006.
- [36] H. Houshiar, D. Borrmann, J. Elseberg, and A. Nüchter. Panorama based point cloud reduction and registration. In *Advanced Robotics (ICAR), 2013 16th International Conference on 25-29 Nov.*, 2013.
- [37] C. Moenning and N. A. Dodgson. A new point cloud simplification algorithm. In *Proceedings 3rd IASTED Conference on Visualization, Imaging and Image Processing*, 2003.
- [38] Y. Wang, J. Tang, Q. Rao, and C. Yuan. High-efficient point cloud simplification based on the improved k-means clustering algorithm. *Journal of Computational Information Systems*, 11(2):433–440, 2015.
- [39] C. Zhangwen and D. Feipeng. 3D point cloud simplification algorithm based on fuzzy entropy iteration. *Acta Optica Sinica*, 33(8):1–7, 2013.
- [40] B.-Q. Shi, J. Liang, and Q. Liu. Adaptive simplification of point cloud using k-means clustering. *Computer-Aided Design*, 43(8):910–922, 2011.
- [41] W. Ru, Z. Meng-Quan, and X. Yu-Hua. Reduction algorithm for 3D scattered points cloud data based on clustering plane feature. *Computer Engineering*, 37(10):249–254, 2011.
- [42] K. H. Lee, H. Woo, and T. Suk. Data reduction methods for reverse engineering. *Journal of Advanced Manufacturing Technology*, 17(10):735–743, 2001.
- [43] B. Belkens, V. Spruyt, R. Berkvens, and M. Weyn. A survey of rigid 3D point cloud registration algorithms. In *the Fourth International Conference on Ambient Computing, Applications, Services and Technologies*, 2014.
- [44] S. Marden and J. Guivant. Improving the performance of icp for real-time applications using an approximate nearest neighbour search. In *Proceedings of Australasian Conference on Robotics and Automation, Victoria University of Wellington, New Zealand, 3-5 Dec.*, 2012.
- [45] P. J. Besl and N. D. Mckay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

- [46] S. Fantoni, U. Castellani, and A. Fusiello. Accurate and automatic alignment of range surfaces. In *IEEE, Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Oct.*, 2012.
- [47] A. V. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Proceedings of Robotics: Science and Systems, Seattle*, 2009.
- [48] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. In *IEEE Transactions on Medical Imaging, Aug.*, 1999.
- [49] S. Rusinkiewicz and M. Levoy. Qsplat: a multiresolution point rendering system for large meshes. In *Proceedings of ACM Siggraph*, 2000.
- [50] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Surface splatting. In *Proceedings of ACM Siggraph*, 2001.
- [51] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, G. Turk, and H. Seidel. Multi-level partition of unity implicits. *ACM Trans Graph*, 22(3):463–470, 2003.
- [52] R. Kolluri. Provably good moving least squares. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005.
- [53] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, 2007.
- [54] B. Alexandre and M. Renaud. Fast and robust normal estimation for point clouds with sharp features. *Comp. Graph. Forum*, 31(5):1765–1774, August 2012. ISSN 0167-7055.
- [55] T. K. Dey, Gang Li, and J. Sun. Normal estimation for point clouds: a comparison study for a voronoi based method. In *Proceedings of Eurographics/IEEE VGTC Symposium on Point-Based Graphics*, 2005.
- [56] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *ACM Siggraph 1992*, 1992.
- [57] C. Lange and K. Polthier. Anisotropic smoothing of point sets. *Computer Aided Geometric Design*, 22(7):680–692, 2005.
- [58] M. Yoon, Y. Lee, S. Lee, I. Ivriissimtzis, and H. Seidel. Surface and normal ensembles for surface reconstruction. *Computer Aided Geometric Design*, 39(5):408–420, 2007.
- [59] M. Pauly, R. Keiser, L. P. Kobbelt, and M. Gross. Shape modeling with point-sampled geometry. *ACM Transactions on Graphics*, 22(3):641–650, 2003.
- [60] G. Guennebaud and M. Gross. Algebraic point set surfaces. *ACM Transactions on Graphics*, 26(3):23, 2007.
- [61] N. Amenta and M. Bern. Surface reconstruction by voronoi filtering. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, 1998.
- [62] T. K. Dey and S. Goswami. Provable surface reconstruction from noisy samples. *Computational Geometry: Theory and Applications*, 35(1):124–141, 2006.
- [63] D. Ouyang and H. Feng. On the normal vector estimation for point cloud data from smooth surfaces. *Computer-Aided Design*, 37(1):1071–1079, 2005.
- [64] B. Li, R. Schnabel, R. Klein, Z. Cheng, G. Dang, and S. Jin. Robust normal estimation for point clouds with sharp features. *Computers and Graphics*, 34(2):94–106, 2010.
- [65] K. Demarsin. *Extraction of closed feature lines from point clouds based on graph theory*. PhD thesis, Katholieke Universiteit Leuven - Faculteit Ingenieurswetenschappen, 2009.
- [66] A. Meyer and P. Marin. Segmentation of 3D triangulated data points using edges constructed with a c^1 discontinuous surface fitting. *Computer-Aided Design*, 36(13):1327–1336, 2004.
- [67] M. Garland, A. Willmott, and P. S. Heckbert. Hierarchical face clustering on polygonal surfaces. In *Proceedings of the Symposium on Interactive 3D Graphics*, 2001.
- [68] M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, 2006.
- [69] M. Vanco and G. Brunnet. Direct segmentation of algebraic models for reverse engineering. *Computing*, 72(1-2):207–220, 2004.
- [70] P. Benkő, R. R. Martin, and T. Varady. Algorithms for reverse engineering boundary representation models. *Computer-Aided Design*, 33(11):839–851, 2001.
- [71] J. Huang and C.-H. Menq. Automatic data segmentation for geometric feature extraction from unorganized 3D coordinate points. *IEEE Transactions on Robotics and Automation*, 17(3):268–279, 2001.

- [72] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, 1981.
- [73] S. P. Lim and H. Haron. Surface reconstruction techniques: a review. *Artificial Intelligence Review*, 42(1):59–78, 2014.
- [74] H.-K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. In *Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision, Vancouver, BC*, pages 194–201, 2001.
- [75] S. Ilic. *Implicit Meshes: Unifying implicit and explicit surface representations for 3D reconstruction and tracking*. PhD thesis, Ecole Polytechnique Federale De Lausanne, 2005.
- [76] M. Bolitho, M. Kazhdan, R. Burns, and H. Hoppe. Parallel poisson surface reconstruction. In *International Symposium on Visual Computing*, pages 678–689, 2009.
- [77] K. Zhou, M. Gong, X. Huang, and B. Guo. Data-parallel octrees for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):669–681, 2011.
- [78] H. Xie, K. T. McDonnell, and H. Qin. Surface reconstruction of noisy and defective data sets. In *IEEE Visualization, October 10–15, Austin, Texas, USA*, pages 259–266, 2004.
- [79] P. Z. Wen, X. J. Wu, Y. Zhu, and X. W. Peng. LS-RBF network based 3D surface reconstruction method. In *IEEE, Control and Decision Conference, 2009. CCDC '09. Chinese*, pages 5785–5789, 2009.
- [80] J. Barhak and A. Fischer. Parameterization and reconstruction from 3D scattered points based on neural network and PDE techniques. *IEEE Transactions on Visualization and Computer Graphics*, 7(1):1–16, 2001.
- [81] M. G. Bolitho. *The Reconstruction of Large Three-dimensional Meshes*. PhD thesis, Johns Hopkins University, 2010.
- [82] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.
- [83] N. Amenta, S. C. Ravi, and K. Kollur. The power crust. In *Eurographics Symposium on Geometry Processing*, pages 249–260, 2001.
- [84] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.
- [85] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *Proceedings of the Conference on Visualization*, pages 21–28, 2001.
- [86] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *SIGGRAPH Comput. Graph.*, 26(2):71–78, July 1992. ISSN 0097-8930.
- [87] G. Petrova J. Manson and S. Schaefer. Streaming surface reconstruction using wavelets. *Computer Graphics Forum*, 27(5):1411–1420, 2008.
- [88] G. E. Farin. *Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide*. Academic Press, 2 edition edition, 1990.
- [89] H. Nowacki, M. I. G. Bloor, and B. Oleksiewicz. *Computational Geometry for Ships*. World Scientific Pub Co Inc, Singapore, 1. ed. edition, 1995.
- [90] K. J. Versprille. *Computer-Aided Design Applications of the Rational B-spline Approximation Form*. PhD thesis, Syracuse, NY, USA, 1975.
- [91] H. Qin and D. Terzopoulos. Triangular NURBS and their dynamic generalizations. 14(4):325–347, 1997.
- [92] B. F. Gregorski, B. Hamann, and K. I. Joy. Reconstruction of B-spline surfaces from scattered data points. In *Proceedings of Computer Graphics International*, page 163, 2000.
- [93] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. In *SIGGRAPH 96 Conference Proceedings. ACM SIGGRAPH, Addison Wesley*, pages 173–205, 1996.
- [94] M. Bertram, X. Tricoche, and H. Hagen. Adaptive smooth scattered-data approximation for large-scale terrain visualization. In *EUROGRAPHICS - IEEE TCVG Symposium on Visualization*, 2003.
- [95] X. Shi, T. Wang, P. Wu, and F. Liu. Reconstruction of convergent G1 smooth B-spline surfaces. *Computer Aided Geometric Design*, 21(9):893–913, 2004.
- [96] N. E. L. Narváez, E. A. L. Narváez, and J. W. Branch. Automatic construction of NURBS surfaces from unorganized points. *Dyna, Year 78*, 78(116):133–141, 2011.
- [97] N. E. L. Narváez, E. A. L. Narváez, and J. W. Branch. Simple method for constructing nurbs surfaces from unorganized points. In *Proceedings of the 19th International Meshing Roundtable*, pp.

- 161-175, 2010.
- [98] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, and et al. The digital michelangelo project: 3D scanning of large statues. In *Proceeding SIGGRAPH'00 Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 131–144, 2000.
 - [99] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H. Seidel. Multi-level partition of unity implicits. In *Proceeding SIGGRAPH '03 ACM SIGGRAPH 2003 Papers*, number 22, pages 463–470, 2003.
 - [100] B. Liao, C. Xiao, L. Jin, and H. Fu. Efficient feature-preserving local projection operator for geometry reconstruction. *Computer-Aided Design*, 45(5):861–874, 2011.
 - [101] P. Labatut, J.-P. Pons, and R. Keriven. Robust and efficient surface reconstruction from range data. *Computer Graphics Forum*, 0(0):1–15, 1981.
 - [102] M. Vanco, B. Hamann, and G. Brunnett. Surface reconstruction from unorganized point data with quadrics. *Computer Graphics Forum*, 27(6):1593–1606, 2008.
 - [103] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, IT-D(4), july 1983. ISSN 0730-0301.
 - [104] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. In *Manuscript UIUCDCS-R-92-1734, Dept. Comput. Sci., Univ. Illinois, Urbana-Champaign, IL*, 1992.
 - [105] X. Xu and K. Harada. Automatic surface reconstruction with alpha-shape method. *The Visual Computer*, 19(7-8):431–443, 2003.
 - [106] F. Cazals, J. Giesen, M. Pauly, and A. Zomorodian. Conformal alpha shapes. In *Proceedings of the Second Eurographics / IEEE VGTC Conference on Point-Based Graphics*, SPBG'05, pages 55–61, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association. ISBN 3-905673-20-7.
 - [107] Z. C. Marton, R. B. Rusu, and M. Beetz. On fast surface reconstruction methods for large and noisy datasets. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17 2009.
 - [108] M. Gopi and S. Krishnan. A fast and efficient projection-based approach for surface reconstruction. In *SIBGRAPI 2002: Proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing*, Kobe, Japan, 2002.
 - [109] R. Li, L. Liu, L. Phan, S. Abeysinghe, C. Grimm, and T. Ju. Polygonizing extremal surfaces with manifold guarantees. In *Proceedings of the 14th ACM Symposium on Solid and Physical Modeling*, SPM '10, pages 189–194, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-984-8.
 - [110] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. *ACM Trans. Graph.*, 21(3):339–346, July 2002. ISSN 0730-0301.
 - [111] M. Kazhdan, A. Klein, K. Dalal, and H. Hoppe. Unconstrained isosurface extraction on arbitrary octrees. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 125–133, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. ISBN 978-3-905673-46-3.
 - [112] J. Digne, J.-M. Morel, C.-M. Souzani, and C. Lartigue. Scale space meshing of raw data point sets. *Computer Graphics Forum*, page 1630–1642, 2011.
 - [113] J. Digne. An analysis and implementation of a parallel ball pivoting algorithm. In *Image Processing On Line*, pages 149–168, 2014.
 - [114] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3): 29:1–29:13, July 2013. ISSN 0730-0301.
 - [115] T. Weyrich, M. Pauly, R. Keiser, S. Heinzle, S. Scandella, and M. Gross. Post-processing of scanned 3D surface data. In *Proceedings of the First Eurographics Conference on Point-Based Graphics*, SPBG'04, pages 85–94, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association. ISBN 3-905673-09-6.
 - [116] X. Li and X. Li. Filling the holes of 3D body scan line point cloud. In *Advanced Computer Control (ICACC), 2010 2nd International Conference*, Vol. 4, pages 334–338, Shenyang, 2010. IEEE. ISBN 3-905673-09-6.
 - [117] C. Wang and P. Hua. A hole-filling algorithm for triangular meshes in engineering. *International Journal for Computational Methods in Engineering Science and Mechanics*, 14(5):465–471, 2013.
 - [118] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 209–216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0-89791-896-7.

- [119] Q. H. Truong. *Knowledge-based 3D point clouds processing*. PhD thesis, Universit  te de Bourgogne, 2014.
- [120] J.-Y. L. and W.-D. Ueng. Reconstruction of surfaces of revolution from measured points. *Computers in Industry*, 41(2):147–161, 2000.
- [121] Y.-K. Lai, S.-M. Hu, and H. Pottmann. Surface fitting based on a feature sensitive parametrization. *Computer-Aided Design*, 38(7):800–807, 2006.
- [122] A. Karniel, Y. Beisky, and Y. Reich. Decomposing the problem of constrained surface fitting in reverse engineering. *Computer-Aided Design*, 37(4):399–417, 2005.
- [123] P. Benko, G. K  s, T. V  rady, L. Andor, and R. Martin. Constrained fitting in reverse engineering. *Computer Aided Geometric Design*, 19(3):173–205, 2002.
- [124] T. V  rady, R. R. Martin, and J. Cox. Reverse engineering of geometric models an introduction. *Computer-Aided Design*, 29(4):255–268, 1997.
- [125] B. Liao, C. Xiao, L. Jin, and H. Fu. Efficient feature-preserving local projection operator for geometry reconstruction. *Computer-Aided Design*, 45:861–874, 2013.
- [126] A. Durupt, S. Remy, G. Ducellier, and B. Eynard. From a 3D point cloud to an engineering CAD model: a knowledge product-based approach for reverse engineering. *Virtual and Physical Prototyping*, 3(2):51–59, 2008.
- [127] A. Durupt, B. Remy, and G. Ducellier. KBRE: Knowledge based reverse engineering for mechanical components. *Computer-Aided Design and Applications*, 7(2):279–289, 2010.
- [128] M.-I. Ouamer-Ali, F. Laroche, A. Bernard, and S. Remy. Toward a methodological knowledge based approach for partial automation of reverse engineering. *Procedia CIRP*, 21(24):270–275, 2014.
- [129] I. Nonaka and N. Konno. The concept of ‘ba’: Building a foundation for knowledge creation. *California Management Review*, 40(3):40–54, 1998.
- [130] S. Grimm, P. Hitzler, and A. Abecker. Knowledge representation and ontologies – logic, ontologies and semantic web languages. In *Springer Berlin Heidelberg*, pages 149–155, 2007.
- [131] O. Busaryev, T. K. Dey, and J. A. Levine. Repairing and meshing imperfect shapes with delaunay refinement. In *Symposium on Solid and Physical Modeling*, 2009.
- [132] R. T. Farouki. Closing the gap between CAD model and downstream application. *SIAM News*, 32(5):303–319, 1999.
- [133] M. W. Beall, J. Walsh, and M. S. Shephard. Accessing CAD geometry for mesh generation. In *12th International Meshing Roundtable, Sandia National Laboratories, SAND-2003-3030P*, 2003.
- [134] C. M. Hoffmann. Geometric and solid modeling: an introduction. In *the Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, July*, 1989.
- [135] A. A. Mezentsev and T. Woehler. Methods and algorithms of automated CAD repair for incremental surface meshing. In *Proceedings, 8th International Meshing Roundtable, South Lake Tahoe, CA, U.S.A., October*, pages 299–309, 1999.
- [136] G. Butlin and C. Stops. CAD data repair. In *Proceedings of the 5th International Meshing Roundtable, Sandia National Laboratories, October*, 1996.
- [137] D. Wang, O. Hassan, K. Morgan, and N. Weatherill. Enhanced remeshing from STL with applications to surface grid generation. *Commun. Numer. Meth. Engng*, 23:227–239, 2007.
- [138] V. Francois, J. C. Cuilliere, and M. Gueury. Automatic meshing and remeshing in the simultaneous engineering context. *Research in Engineering Design*, 11:55–66, 1999.
- [139] R. Bronsart, G. Knieling, and M. Zimmermann. Automatic generation of a panel-based representation of ship hulls for wave resistance calculations. In *Proceedings, PRADS 2004, Schiffbautechnische Gesellschaft, Hamburg*, 2004.
- [140] R. B. Rusu and S. Cousins. 3D is here: Point cloud library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*.
- [141] D.-J. Yoo. Three-dimensional surface reconstruction of human bone using a B-spline based interpolation approach. *Computer Aided Geometric Design*, 43(8):934–947, 2011.
- [142] J. G. Hayes and J. Halliday. The least-squares fitting of cubic Spline surfaces to general data sets. *J. Inst. MathsApplics*, 14(1):89–103, 1974.
- [143] N. Amenta, M. Bern, and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *Proceeding SIGGRAPH ’98 Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, number 26, pages 415–421, 1998.
- [144] C. Xiao, W. Zheng, Y. Miao, Y. Zhao, and Q. Peng. A unified method for appearance and geometry completion of point set surfaces. *The Visual Computer*, 23(6):433–443, 2007.

- [145] I. K. Park, I. D. Yun, and S. U. Lee. Constructing NURBS surface model from scattered and unorganized range data. In *Proceedings. Second International Conference on 3D Digital Imaging and Modeling*, pages 312–320, 1999.
- [146] V. Weiss, L. Andor, G. Renner, and T. Varady. Advanced surface fitting techniques. *Computer Aided Geometric Design*, 19(1):19–42, 2002.
- [147] H. Park. An approximate lofting approach for B-spline surface fitting to functional surfaces. *International Journal of Advanced Manufacturing Technology*, 18(7):474–482, 2001.
- [148] W.-D. Ueng, J.-Y. Lai, and J.-L. Doong. Sweep-surface reconstruction from three dimensional measured data. *Computer-Aided Design*, 30(10):791–806, 1998.
- [149] X. Shi, T. Wang, and P. Yu. A practical construction of G1 smooth biquintic B-spline surfaces over arbitrary topology. *Computer-Aided Design*, 36(5):413–424, 2004.
- [150] J. Wang, Z. Yub, W. Zhangc, M. Weid, C. Tana, N. Dai, and X. Zhange. Robust reconstruction of 2D curves from scattered noisy point data. *Computer-Aided Design*, 50:27–40, 2014.
- [151] I. Budak, D. Vukelić, D. Bračun, J. Hodolić, and M. Soković. Pre-processing of point-data from contact and optical 3D digitization sensors. *Sensors*, 12(1):1100–1126, 2012.
- [152] M. Bole. Regenerating hull design definition from poor surface definitions and other geometric representations. In *COMPIT 2014, Redworth, UK*, pages 173–205, May 2014.
- [153] F. L. Pérez, J. A. Clemente, J. A. Suárez, and J. M. González. Parametric generation, modeling, and fairing of simple hull lines with the use of nonuniform rational B-spline surfaces. *Journal of Ship Research*, 52(1):1–15, 2008.
- [154] F. Menna and S. Troisi. Low cost reverse engineering techniques for 3D modelling of propellers. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVIII, Part 5 Commission V Symposium, Newcastle upon Tyne, UK.*, 2010.
- [155] F. Menna and S. Troisi. Photogrammetric 3D modelling of a boat’s hull. In *Proceedings of the Optical 3D Measurement Techniques VIII, at ETH Zurich.*, 2007.
- [156] S. Gerbino, F. Renno, and S. Papa. Two reverse engineering methods for the reconstruction of an high speed craft surface: A comparison. In *International Design Conference-Design, May 18-21, 2004*.
- [157] Desta M. Edessa and R. Bronsart. A contribution to curves network based ship hull form reverse engineering. *International Shipbuilding Progress*, 62(1-2):17–42, 2015.
- [158] L. Piegl and W. Tiller. *The NURBS Book*. Springer, New York, 2. ed. edition, 1997.
- [159] C.-K. Shene. Introduction to computing with geometry notes. Internet, 2014.
- [160] Robert McNeel and Associates. Rhinoceros homepage. Software, 2015.
- [161] Point Cloud Library. Point cloud library homepage. Internet, 2015. URL <http://pointclouds.org/>.
- [162] DAVID 3D Solutions GbR. David laserscanner homepage. Internet, 2015. URL <http://www.david-3d.com/>.
- [163] M. Eck and H. Hoppe. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’96*, pages 325–334, New York, NY, USA, 1996. ACM. ISBN 0-89791-746-4. doi: 10.1145/237170.237271.
- [164] A. Yvart, S. Hahmann, and G.-P. Bonneau. Smooth adaptive fitting of 3D models using hierarchical triangular splines. In *In Proceedings of the International Conference on Shape Modeling and Applications (SMI)*, pages 13–22, 2005.
- [165] A. S. Wen, S. M. H. Shamsuddin, and Y. Samian. An optimized ship hull fitting approach using nurbs. In *Proceedings of the Postgraduate Annual Research Seminar*, pages 33–37, 2006.
- [166] W. Y. Chang. Surface reconstruction from points. In *CSD CSE Technical Report CS2008-0922*. February 2007.
- [167] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, and C. T. Silva. A benchmark for surface reconstruction. *ACM Trans. Graph.*, 32(2):1–17, April 2013. ISSN 0730-0301.
- [168] WS Atkins Consultants and members of the NSC. *Best Practice Guidelines For Marine Applications of Computational Fluid Dynamics*. MARNET-CFD, 2002.
- [169] F. Stern, J. Yang, Z. Wang, H. Sadat-Hosseini, M. Mousaviraad, S. Bhushan, and T. Xing. Computational ship hydrodynamics: Nowadays and way forward. *International Shipbuilding Progress*, 60:pp. 3–105, 2013.
- [170] ITTC2011. The specialist committee on computational fluid dynamics. In *Proceedings of 26th International Towing Tank Conference. Rio de Janeiro, Brazil*, 2011.

- [171] T. J. Tautges. The common geometry module (CGM): A generic extensible geometry interface. In *Proceedings, 9th International Meshing Round Table*, 2000.
- [172] S. Merazzi, E. A. Gerteisen, and A. Mezentsev. A generic cad mesh interface. In *Proceedings, 9th International Meshing Round Table*, 2000.
- [173] J. P. Steinbrenner, N. J. Wyman, and J. R. Chawner. Fast surface meshing on imperfect CAD models. In *Proceedings, 9th International Meshing Round Table*, 2000.
- [174] N. A. Petersson and K. K. Chand. Detecting translation errors in CAD surfaces and preparing geometries for mesh generation. In *Proceedings, 10th International Meshing Roundtable*, pages pp.363–371. Sandia National Laboratories, 2001.
- [175] N. A. Petersson. A software demonstration of 'rap': preparing cad geometries for overlapping grid generation. In *Proceedings of the 8th International Conference on Numerical Grid Generation in Computational Field Simulations, UCRL-JC-147260*, 2002.
- [176] C. Piret, J.-F. Remacle, and E. Marchandise. Mesh and CAD repair based on parameterizations with radial basis functions. In *Proceedings of the 20th International Meshing Roundtable 2012, Part 6*, pages pp. 419–436, 2012. doi: DOI:10.1007/978-3-642-24734-723.
- [177] F. Hétroy, S. Rey, C. Andújar, P. Brunet, and Á. Vinacua. Mesh repair with topology control. Technical report, Rapport de recherche, INRIA, numéro 6535, France, 2008.
- [178] S. Bischoff and L. Kobbelt. Structure preserving cad model repair. In *Comput. Graph. Forum*, volume 24, pages 527–536, 2005.
- [179] P. S. Patel, D. L. Marcum, and M. G. Remotigue. Automatic CAD model topology generation. *International Journal for Numerical Methods in Fluids*, 52:pp. 823–841, 2006.
- [180] C. Abt, E. Bergmann, and S. Harries. Domain preparation for RANSE simulations about hull and appendage assemblies. *Journal for Research in Ship Technology. Ocean Engineering and Related Subjects*, 59(3):24–36, 2012.
- [181] R. Bronsart and G. Knieling. Automated knuckle detection in ship hull forms. *Ship Technology Research*, 53(1):3–10, 2007.
- [182] Desta M. Edessa, kleinsorge Lutz, and R. Bronsart. Automatic pre-mesh cad data repairing. *International Journal of Mechanical Engineering and Applications*, 1(1), 2013.
- [183] Desta M. Edessa, Kleinsorge L., and B. Bronsart. A contribution to automatic ship hull form cad data repairing. In *Proceedings of the 12th International Symposium on Practical Design of Ships and Other Floating Structures (PRADS)*, volume 1, pages 437–443, Changwon City, South Korea, 2013.
- [184] S. Krüger, H. Vorhölter, T. Rung, and M. Manzke. Introduction of RANS-CFD into the initial design process. In *8th Int. Conf. Computer Applications and Information Technology in the Maritime Industries*, Budapest, Hungary, 2009.
- [185] Desta M. Edessa, kleinsorge Lutz, and R. Bronsart. A tool for ship hull surface healing and domain preparation for downstream applications. *International Journal of Computer Aided Engineering and Technology*, In Press.
- [186] Desta M. Edessa, Kleinsorge L., and B. Bronsart. Assuring quality ship hull form representations for downstream applications. In *Proceedings of 2nd International Conference on Maritime Technology and Engineering (MARTECH)*, pages 317–324, Lisbon, Portugal, 2014.
- [187] O. Moctar, V. Shigunov, and T. Zorn. Duisburg test case: Post-panamax container ship for benchmarking. *Ship Technology Research*, 59(3):50–64, 2012.
- [188] L. Kleinsorge, N. Kornev, R. Bronsart, and B. Hannker. OPENFOAM's CFD workbench and its integration into the initial design process. In *Proceedings of "International Conference on Computer Applications in Shipbuilding (ICCAS)*, volume 1, pages 1–8, Budapest, Hungary, 2011.

Declaration of Authorship

I, Desta Milkessa Edessa, declare that this thesis titled, 'A Framework for Hull Form Reverse Engineering and Geometry Integration into Numerical Simulations' and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Acknowledgements

Next to none, I thank the Almighty God for His blessing, intellectual insight and power through this work.

I would like to express my special appreciation and thank to my supervisor Professor Dr.-Ing. Robert Bronsart, you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on my career have been priceless. The encouragements, inspirations, suggestions and comments that I have received from you, are particularly appreciable.

This dissertation would have not been possible without the funding from "Landes-graduiertenförderung" at University of Rostock. I gratefully acknowledge the funding source that made my PhD work possible.

I would like to extend my special acknowledgment to all colleagues at the chair, for their support, comments and more importantly for making the work environment attractive, enjoyable and lovable. I would like to say thank you, specially Mr. Lutz Kleinsorge for his immense help during my research work.

A special thanks to my family. My grandmother and grandfather deserve special mention for raising me with great ambition and encouragement, which helped me to come thus far. My mother's care, love and prayer have special place. The assistance, love and encouragements of my friends and all well-wishers should be always remembered. Thank you.

Desta Milkessa Edessa

University of Rostock

Rostock, 2016