Predicting Human Behavior in Smart Environments: Theory and Application to Gaze Prediction

Dissertation
to obtain the academic degree of
Doktor-Ingenieur (Dr.-Ing.)
of the Faculty of Computer Science and Electrical Engineering
at the University of Rostock

submitted by
M.Sc Redwan Abdo Abdullah Mohammed
born on 22.05.1979 in Lahj, Yemen
from Rostock

Rostock, 2015

urn:nbn:de:gbv:28-diss2017-0029-6

First Reviewer: Prof. Dr. Sc. techn. Oliver Staadt, Visual Computing Group, Institute of Computer Science, University of Rostock

Second Reviewer: Prof. Dr.-Ing. Thomas Kirste, Mobile Multimedia Information Systems Group, Institute of Computer Science, University of Rostock

External Reviewer: Prof. Dr. Henning Sprekeler, Modelling of Cognitive Processes Group, Institute of Software Engineering and Theoretical Computer Science, Technical University Berlin

External Reviewer: Dr. rer. nat. Lars Schwabe, Data Insight Lab, Lufthansa Industry Solutions

Date of dissertation submission: 30 October 2015

Date of dissertation defence: 28 November 2016

Abstract

Predicting human behavior is desirable in many application scenarios in smart environments. Gaze represents one of the essential cues, which is important to understand these behaviors. In this thesis, we consider eye movements and the spatial location of visual attention in different behavioral context as a model system. Behavioral eye movements data in a different context is presented together with predictive models of visual saliency. The existing models for eye movements do not take contextual factors into account. This addressed using a systematic machine-learning approach, where user profiles for eye movements behaviors are learned from data. Machine learning models and the analysis of behavioral data show the limitations of current predictive models describing human eye movements behaviors and reveal the influences of task on gaze selection. The analysis furthermore demonstrates the relative importance given to the individual visual features, and it shows that simple predictive "one-fits-all"-models will not work for eye movements prediction. This part of the work used model-based systematic data analysis.

Human studies have shown that eye movements behaviors are mostly effected by the task at hand. For that, human vision has to learn how to move the eyes to the relevant information. In this part of the work a theoretical innovation is presented, which goes beyond pure data analysis. The thesis proposed the modeling of eye movements as a Markov Decision Processes (MDPs). Then it use Inverse Reinforcement Learning (IRL) paradigm to infer the reward function. The examined IRL approaches used information about the possible eye movement positions. We found that it is possible to automatically extract reward function based on effective features from user eye movement behaviors using IRL. We found that the reward function was able to extract expert behavior information that fulfill to predict eye movements behaviors. By using a new inverse reinforcement learning paradigm that constructs the parameters of the learning model to best match the observed human behavior, the connection between model and empirical data is obtained. The application of this method to the empirical data shows that this model can be used in eye movement predictions, and in human behavior modeling in general.

Keywords: human behavior models, visual attention, gaze prediction, normative models.

Zusammenfassung

Die Vorhersage menschlichen Verhaltens in intelligenten Umgebungen ist für viele Anwendungsszenarien wünschenswert. Einer der wichtigsten Hinweise für die Vorhersage menschlichen Verhaltens ist die Blickrichtung des Menschen. In dieser Arbeit betrachten wir Augenbewegungen und die räumliche Verteilung der visuellen Aufmerksamkeit in verschiedenen Kontexten als Modellsystem. Die Augenbewegungsdaten von Menschen in verschiedenen Kontexten werden in dieser Arbeit zusammen mit Vorhersagemodellen der visuellen Aufmerksamkeit präsentiert. Bereits existierenden Modelle für die Vorhersage der Augenbewegungen berücksichtigen keine Kontextfaktoren. Um die Kontextfaktoren zu berücksichtigen, nutzen wir einen systematischen Ansatz des Maschinellen Lernens, wo Nutzerprofile für die Augenbewegungen gelernt werden. Modelle des maschinellen Lernens und die Analyse der Verhaltensdaten zeigen die Grenzen aktueller Vorhersagemodelle zur Beschreibung menschlicher Augenbewegungen und zeigen den Einfluss der jeweiligen Aufgabe. Die Analysen in dieser Arbeit zeigen darüber hinaus die relative Bedeutung spezifischer visueller Merkmale. Auwird gezeigt, dass es für die Vorhersage von Augenbewegungen kein einfaches Modell für alle Daten gibt („one-fits-all"-Modell). Dieser Teil der Arbeit verwendet stark modell-gestützte systematische Datenanalysen.

Studien am Menschen haben gezeigt, dass Augenbewegungen am meisten durch die jeweilige Aufgabe beeinflusst sind. Um dies zu erreichen, musste die menschliche visuelle Wahrnehmung lernen, den Blick auf die jeweils relevante Information zu lenken. In diesem Teil der Arbeit wird eine theoretische Innovation präsentiert, die über reine Datenanalyse hinausgeht. Es wird vorgeschlagen, die Modellierung der Augenbewegungen als Markov Entscheidungsproblem zu verstehen. Dann wird inverses Belohnungslernen („Inverse Reinforcement Learning") angewendet, um auf Grundlage von beobachtetem Verhalten (den Augenbewegungen) eine Belohnungsfunktion abzuleiten. Eine solche Belohnungsfunktion wird dann als Teil eines prädiktiven Modells verwendet werden, um Augenbewegungen vorherzusagen. Dies zeigt, dass das Modell in der Vorhersage genutzt werden kann und damit auch der gesamte neue in der Modellierung menschlichen Verhaltens generell.

Acknowledgments

This dissertation could not have been possible without the support of many people. First and foremost, I would like to thank my supervisor Professor Oliver Staadt for his excellent and valuable guidance, patience, support and assistance over the last two years. This thesis would not have been possible without his support. I also would like to thank Professor Thomas Kirste for his support as a second supervisor of my thesis. Thank you for helping me shape and finishing this work.

Very special thanks go to my third supervisor Dr. Lars Schwabe. I am grateful to him for the help, encouragement, suggestions and valuable ideas during the development of this work.

I would like to thank also my colleagues at the Visual Computing Group for making the work environment rewarding. Furthermore, I would like to express my thank my former MuSAMA colleagues for the incredible time I had in the graduate college. I am especially thankful to Kristina Yordanova, Alexander Steiniger, Michael Zaki, Rene Leistikow and Christian Scheel for being not only colleagues but such awesome friends.

I would like to express my sincere gratitude to my parents and my brothers; They have been generous with their encouragement during my studies; I am indebted to them forever. I would like to dedicate this masterpiece to my wife Rasha and my daughter Rahaf for their encouragement and love that gave me the strength to do this dissertation.

Finally, I would like to acknowledge that this work funded by the German Research Foundation.

Contents

List of Figures

Chapter 1

Introduction and Motivation

1.1 Motivation

Predicting user behavior is desirable in many application scenarios in smart environments. This includes the use of user models, which take into account internal states of the user such as the focus of attention. The motivation for my work is to build systems, which can infer and predict the attention/intention of users based on signals collected from various sensors so that smart environments can react in a proactive manner in order to assist the user. However, the signals delivered by sensors in such smart environments are usually not informative enough to simply read of the internal states of users as, for example, one can read of the body temperature from a thermometer. Prior knowledge about how humans reason, decide, and act needs to be employed in order to disambiguate the signals and infer attention/intentions with a limited amount of data.

The general problem is to best predict the actions of users in smart environments in collaborative scenarios such as a smart meeting room or a situation room, where they have to jointly make decisions based upon incomplete and unreliable information under time constraints. One approach, which is also pursued in MuSAMA[1], is to move beyond the recognition of activities to the recognition of internal states of users such as intentions, current goals, or the focus of attention, because based on recognized internal states the activity prediction may be easier.

In smart environments users are often interrupted, manage very large quantities of information, and they switch between the contents of different displays, for example on tiled Large High-Resolution Displays (LHRDs). In order to address interaction in a more realistic manner, it is essential to investigate the processes that govern for example human attentional processes in that settings. Attention plays a fundamental role in interaction and task execution. Attention helps us to reason, decide, act and communicate with our environments that

[1] `http://www.informatik.uni-rostock.de/musama_homepage.html`

offer us a massive amount of stimuli. Selective attention exists for all senses, because of the need to deal with this massive amount of sensory data. It enables people to extract the relevant information at an early processing stage. In this thesis, I consider eye movements and the spatial location of visual attention as the model system.

Although attention recognition is certainly a desirable property of any computing system interacting with humans, it may come as a surprise that it is largely absent from the most existing systems [91, 208, 165]. Why is this the case? I argue that current visual attention recognition systems are using cognitive models, which are not grounded enough in empirical research. Therefor, in my thesis I address this shortcoming. More specifically, in my thesis I conduct work on two converging lines of research: *First*, I explore recent findings from cognitive social neuroscience and decision making, because researchers in these fields are investigating how humans visual attention system work and how humans form attention/intentions and recognize the intentions of others. Then, I exploit existing models in these fields. I used this models in real-world settings, where I applied various methods of computing saliency maps for predicting eye movements in smart environments in different scenarios and determining (by combinatorial exploration) which features are relevant as a function of the context. *Second*, I develop models for predicting eye movements in the smart environment. Here I used machine learning technique and a normative theory. In order to constrain my work, I focus on a few selected application scenarios, but my aim is the use of these methods and tools generalize to other applications.

Eye movements and the visual field locations during fixation periods are often considered as an informative observable, but eye movements are at best an indirect measure of attention. It is known for a long time that task-demands affect the patterns of eye movements [213]. Hence, properly predicting eye movements is still a challenging task, in particular for more natural scenarios [22] such as those encountered in ubiquitous computing.

The notion of a saliency map has been helpful in visual attention research: Here, certain locations in the visual field are determined as "salient" if they are – in statistical terms – outliers relative to the surrounding visual field locations. Computational modeling of the visual system was quite successful in the sense of predicting saliency maps based on image properties, which closely match the experimentally measurable maps of eye movements and fixation periods [97]. Such saliency maps reflect bottom-up attentional processes, in other words, the attraction of attention by external cues.

The existing computational models of visual saliency rely only on 2D scene features and need to be extended to the 3D world. These models do not consider task demands or the user's internal state. Also, the limitation to use these models in real world application need to be investigated in order to make them

more robust to noise, and illumination changes. Here we[2] argue that based on information processing principles where saliency models derivable from, combined with learning mechanism or the theory of decision making, a notion of optimal processing could be predicted.

1.2 Problem

This thesis addressed the problem of predicting human gaze behavior in smart environments. Internal states of users such as visual attention, intentions and the cognitive load are important for predictions, but can not measure them directly. Attentional models can find interest regions that attract our attention. The existing models for eye movements not satisfying, in particular for more natural scenarios such as those encountered in smart environments. Also, the existing models for eye movements rely only on low-level features and do not take contextual factors or top-down into account.

Therefor the goals of this thesis is to: (1) empirically investigates how different visual features are relevant for predicting human eye movements in different behavioral context in smart environments; and (2) introduce a theory of normative modeling as a paradigm for human behavior prediction, and used it to predict user eye movement behavior in interaction scenario in smart environments.

To achieve (1), the work uses a systematic machine-learning approach, where user profiles for eye movements are learned from data in different contexts, and determining by combinatorial exploration which features are relevant for behavioral context.

To achieve (2), the work proposes the modeling of eye movements using principles from normative theory. The approach taken here is to formulate eye movements as a Markov decision process (MDP) problem, but with the use of Inverse Reinforcement Learning (IRL) to infer the reward function.

1.3 Contribution of the Thesis

The contributions of this thesis are summarized below:

1. We determine by combinatorial exploration which features are relevant for eye movements prediction in different behavioral contexts.

   We investigate meeting scenarios in terms of how relevant different features are for eye movements prediction in different behavioral contexts.

[2]Throughout the thesis, the personal pronoun ”we” is used for simplicity in the sentence structure, and not as an indication that the work was completed by multiple persons.

We used a machine-learning approach to find out which features are important in meeting scenarios. The details described in Chapter 4. The main result of this study is that the prediction differed according to the type of features we selected. As a consequence, simple predictive "one-fits-all"-models will not work for eye movements prediction. This finding points towards including context information about the scene and situation into the computation of saliency maps as important towards developing models of eye movements, which operate well under natural conditions such as those encountered in smart environments settings. This work is published in [135].

2. We investigate, how relevant depth features are for eye movement prediction.

   (a) We analyze the scene dependency in saliency map in luminance and depth images features in natural scenes

   Here we explore, for the first time, statistical properties of saliency in natural luminance and depth images, and it is described in Chapter 5. We first analyze the dependency between luminance and depth images features in natural scenes using information-theoretic measures. We did this using a database of natural images and depth images. Then we measure the scene-dependency in saliency map in luminance and depth images features in natural scenes. We find that certain oriented filter responses convey more information about relevant depth features than other oriented filters. Also, we find that saliency in depth images is bimodally distributed with highly salient locations corresponding to low salient 2D image locations. The results published in [131, 136].

   (b) We develop a system called BatGaze system to measure depth features in the center of gaze

   In Chapter 5 we present the BatGaze system, which we have built to measure depth at the center of gaze in free-viewing scenarios. We argue that it will become a tool for mapping the visual environment of free viewing humans in an unprecedented way. The rationale for building such a system is to inform computational vision research about these features, so that generative models of visual signals could be learned. We have described in depth the technical aspects of this system, the software we have developed, and the analysis procedures. In addition, we have also performed an experimental validation. This work is published in [130].

   (c) We use machine learning techniques to learn models based on depth features

We used machine learning techniques to train a bottom-up, top-down model of saliency based on 2D and depth features/cues. More detail about this work described in Chapter 5. Briefly, we find that the depth information improves prediction and hence it should be included in predictive models. This work is published in [132].

3. We explore how well existing bottom-up visual saliency models perform compare to human eye movements behavior in real world scenarios (i.e., in the interaction scenario with tiled Large High-Resolution Displays).

   Here we investigate the effects of bezels LHRDs on human eye movements and on saliency algorithm predictions. Our results presented in Chapter 6. The results published as a short paper in [133].

4. We propose a new model using a normative approach for eye movements predictions on LHRD.

   In Chapter 7 we present our approach of modeling eye movements on tiled Large High-Resolution Displays (LHRD) using inverse reinforcement learning. We have examined two different inverse reinforcement learning algorithms. The presented approach used information about the possible eye movement positions. We found that it is possible to automatically extract reward function based on effective features from user eye movement behaviors using IRL. The results published in [134].

1.4 Outline of the Thesis

The thesis is structured as follows.

Chapter 2 outlines the background of my work and reviews the current work on modeling human eye movements.

Chapter 3 introduces specific approaches to modeling human behavior, which is necessary to position my conceptual and empirical contributions.

Chapter 4 introduces the use of machine learning for predicting eye movements in smart environments based on context.

Chapter 5 practically illustrates how relevant are depth features for predicting human eye movements.

Chapter 6 illustrates the effects of interior bezels of tiled-displays on saliency algorithms prediction and human eye movements Behaviors.

Chapter 7 introduces our proposed model for predicting eye movements via inverse reinforcement learning in the interaction scenario with the LHRD.

Finally, **Chapter 8** presents the thesis conclusion and the future work.

1.5 Publications Resulting from this Dissertation

Portions of this thesis have previously appeared as conference publications:
Peer reviewed publications

1. Redwan Abdo A. Mohammed and Oliver Staadt. *Learning Eye Movements Strategies on Tiled Large High-Resolution Displays using Inverse Reinforcement Learning.* In Proceedings of the IJCNN 2015, Killarney, Ireland, July 2015, published in IEEE Xplore Digital Library. Available from: doi:10.1109/IJCNN.2015.7280675. The details presented in Chapter 7.

2. Redwan Abdo A. Mohammed and Oliver Staadt. *Effects of Interior Bezels of Tiled Large High-Resolution Displays on Saliency Prediction and Human Eye Movement Behavior.* In Yuki Hashimoto, Torsten Kuhlen, Ferran Argelaguet, Takayuki Hoshi, and Marc Erich Latoschik, editors, ICAT-EGVE 2014 - Posters and Demos. The Eurographics Association, 2014. Available from: doi:10.2312/ve.20141369. The results presented in Chapter 6.

3. Redwan Abdo A. Mohammed, Lars Schwabe and Oliver Staadt. *Towards Context-Dependence Eye Movements Prediction in Smart Meeting Rooms.* In Proceedings of the 24th International Conference on Artificial Neural Networks (ICANN 2014) Springer LNCS, Hamburg, Germany, September 2014. Available from: doi:10.1007/978-3-319-11179-7_32. The details presented in Chapter 4

4. Redwan Abdo A. Mohammed, Lars Schwabe and Oliver Staadt. *Gaze Location Prediction with Depth Features as Auxiliary Information.* In Proceedings of the 16th International Conference on Human-Computer Interaction (HCII 2014) Springer LNCS, Crete, Greece, June 2014. Available from: doi:10.1007/978-3-319-07230-2_28. The details presented in Chapter 5.

5. Redwan Abdo A. Mohammed, S. Mohammed and Lars Schwabe. *BatGaze: A New Tool to Measure Depth Features at the Center of Gaze During Free Viewing.* In Proceedings of the 2012 International Conference Brain Informatics pages 85-96 Springer LNCS 7670, Macau, China, Dec 2012. Available from: doi:10.1007/978-3-642-35139-6_9. The details presented in Chapter 5.

6. Redwan Abdo A.Mohammed, S. Mohammed and Lars Schwabe. *A Brain Informatics Approach to Explain the Oblique Effect via Depth Statistics.* In Proceedings of the 2012 International Conference Brain Informatics

pages 97-106 Springer LNCS 7670, Macau, China, Dec 2012. Available from: doi:10.1007/978-3-642-35139-6_10. The details presented in Chapter 5.

7. Lars Schwabe and Redwan Abdo A. Mohammed. *Scene-Dependence of Saliency Maps of Natural Luminance and Depth Images*. In Fifth Baltic Conference "Human Computer Interaction", 2011, pages 29-36, Riga, Latvia, 2011. Available from: `http://basoti.uni-rostock.de/index.php?id=1149`. The details presented in Chapter 5.

Chapter 2

Background and Related Work

This thesis address the general problem of predicting human gaze behavior in order adapt smart environments to the goals of humans and their anticipated actions. Human behavior prediction is a rather young research field with many connections to other related fields (Sec. 2.1). In this Chapter, we introduce a few key concepts (Sec. 2.2) that serve as the background for our work, namely concepts from cognitive and social neuroscience, decision theory, and probabilistic machine learning. Cognitive science investigates the basis of human decision making and thus should be considered as a potentially valuable source of information and inspiration for this thesis. It is related to decision and probability theory, because current cognitive neuroscience uses them extensively to explain human behavior as being optimal (or rational) under certain constrains. Of course, these concepts are also of relevance to our methods for analyzing the data from our experiments. We describe in greater detail the work on modeling eye movements in Sec. 2.3, because this is the domain of human behavior that we selected to explore in greater detail in the rest of this thesis. Then, we recapitulate the state-of-the-art in related fields briefly in Sec 2.4.

2.1 Position of this Thesis

This thesis is about using predictive user models in smart environments. This includes the use of user models, which take into account internal states of the user such as the focus of attention. The latter is widely acknowledged as an important factor to be considered in the design of the human-centric application. Besides the need for visual attention models that are more robust to noise, and environmental changes. With this viewing direction, this work investigates visual attention modeling in term of the relevant of visual features for gaze prediction in different behavioral context. Different models were investigated; Also, this thesis presents a new method for modeling human eye movements using inverse reinforcement learning.

By the nature of my work, I need to connect to a few and currently still largely distinct fields of research, namely i) *cognitive and social neuroscience*, i.e., both their empirical branches and the theories developed within these fields, which are largely rooted in ii) *decision theory*. In Artificial Intelligence (AI) the discipline of iii) *cognitive modeling* became an established field, where cognitive architectures are developed and applied. As compared to the mainly theoretical and minimalistic models from decision theory the cognitive architectures are usually much more complex and aimed at accounting for many aspects of human cognition, whereas decision theory is more focused on decision making itself. The field of iv) *machine learning* provides the methodological toolbox for both the user modeling and the integration of these models into smart environments.

2.2 Key Concepts

2.2.1 Cognitive and Social Neuroscience

The study of how a human behaves is becoming important as the understanding grows that much of human information processing and behavior appears in social interaction. The magnificent amount of work done in recent years in cognitive neuroscience, and social psychology has yielded new anticipation into the processes involved in attention and intention understanding and task sharing. Exploring the relationship between perception and action understanding became serious due to the discovery of mirror neurons that fire when animals execute actions and when they observe the same actions done by other individuals.

2.2.1.1 Social Neuroscience

Classical cognitive neuroscience and social psychology have formed a new discipline field called social neuroscience.

In last years, classical cognitive neuroscience has much improved our understanding of how the brain processes information that we perceive, such as: color, shape, smells, and motion. Also, it advanced our knowledge about how our brain enables us to perform higher-order cognitive operations as short- and long-term memory tasks, speech generation and recognition, and the executive functions involved in planning. Such approaches govern by the implicit hypothesis that understanding one brain is adequate for realizing the behavior of all humans [180]. Obviously, such methods ignore the fact that humans are natural social rather than individualists. In fact, the social environment which surrounding the brain affects its basic actions.

In general, social neuroscience devoted to understanding the complex interactions between social factors and their influence on behavior. Furthermore,

studying the cognitive processes underlying these behaviors (see also [148] and [180]).

Currently, many researchers are interested in understanding the nature of human social interaction and its relation to human decision-making, in order to determine the neural mechanisms underlying these complex social skills (see for example [63] [124] [181]).

2.2.1.2 Theory of Mind

First introduced by Premack and Woodruff [154], while discussing whether chimpanzees have the ability to attribute mental states of other in term of their desires, intentions, and beliefs. In the same year the philosopher Daniel Dennett [52] suggested that the most stringent test for the presence of theory of mind would be to see whether someone can predict someone else's actions on the basis of that person's false belief. Later Wimmer and Perner [210] developed the false-belief paradigm to test and understand another person's wrong belief.

This paradigm was widely used to examine children's mentalizing abilities. In this paradigm, subjects observe how an actor put an object into a location x and then noticed that in the absence of the actor the object was moved from x to position y. Then subjects had to find out where the actor will look for the object [210]. Many studies funded that children age four and older start to correctly attribute false beliefs to others and provide a valid demonstration when asked.

The study of our capacity to reason about other people's minds become the focus of cognitive neuroscience research. Because of the development of modern imaging techniques, theory-of-mind studies have shown different area of brain network involved in theory of mind, which are: the posterior superior temporal sulcus (STS) extending into the temporoparietal junctions (TPJ), the medial prefrontal cortex (mPFC), and sometimes also the temporal poles (TP). A graphical drawing of the mentalizing brain network is illustrated in blue in figure 2.1 taken from [180] (see also [28]). Also, the mental states have been studied in decision-making task [124] and game theoretical paradigms [69, 125]. In [69, 125] subjects in MRI scanner played strategy games against someone sitting outside the scanning room and against a computer. The brain areas activation compared between computer and human conditions. These studies have found involvement of the medial prefrontal lobe.

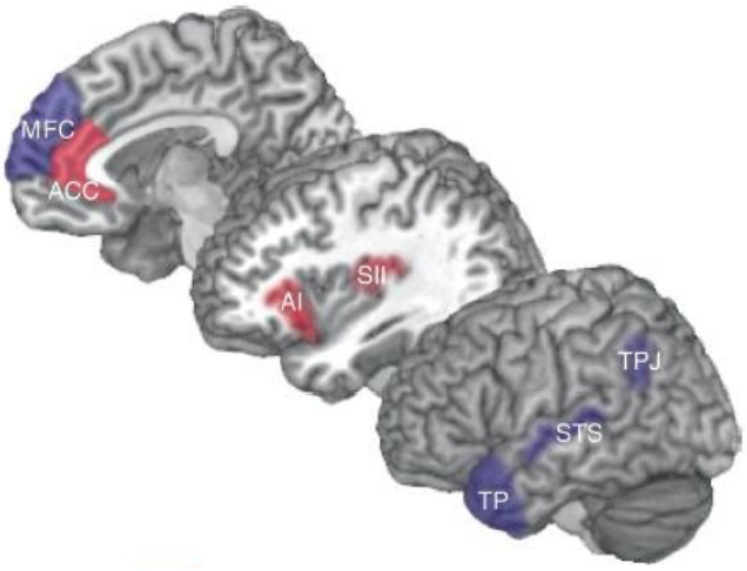


Figure 2.1: Brain networks involved in understanding others. graphical representation of the brain areas typically involved in theory of mind (blue) and empathy (red) tasks (from [180]).

2.2.1.3 Simulation theory

Theory of mind enables one to understand the mental states that cause others' behavior. So that it allows one to explain and predict the observed actions produced by others [37, 205, 180, 77]. Simulation theory is one of the accounts given by psychologists for the mechanism underlying this ability [37].

Breazeal et al. [31] argued that by simulating other individual's actions and the stimuli they are facing with using our own behavioral and stimulus manipulation mechanisms. We can predict the behaviors and mental states of others based on the behaviors and mental states that they would possess if they were in the other's situation.

Simulation theory approaches recently used in the field of understanding other people's intention by simulate other people movements using our motor program, and feelings by simulating their feelings using our affective motor programs [51, 104], taking into accounts the specific role of mirror neurons in the understanding of others motor actions and action-related intentions. In case of feelings understanding, Iacoboni et al. [36] and Dapretto et al. [49] suggested a motor theory of empathy. According to the broader role of mirror neurons in social cognition.Where their role is not only to action understanding

but also to understand the emotions and minds of others.

2.2.1.4 Mirror neurons

Mirror neurons where discovered in the frontal and parietal cortex of the monkey brain [163], which activated not only when monkey execute actions, but also when they observe the same actions done by other individuals. In human, it is possible to measure the activity of single neurons [93]. However, the mirror system applies to the brain imaging data where a brain region is considered to be part of a mirror system. The discovery of mirror neurons has a huge impact on social cognitive neuroscience. The existence of mirror neurons can extend our ability to understand other people's goals and intentions. Recently, it was suggested that mirror neurons may represent the basis for imitation.

Since the discovery of mirror neurons, several studies have demonstrated a generation of motor actions and similar common coding of the perception in the human brain using imaging techniques. In these studies, people were scanned while they watched movies depicting short motor actions. The observed activation was then compared to that observed when the scanned subjects performed the same motor action themselves.

Currently, researchers are debating about the exact function of the mirror neuron system and its role in social cognition. It has been proposed that besides action recognition [70]. The mirror neuron system might play a general role in understanding other people's intentions and goals [65], understanding of the functioning of imitation [66] and emotions [49], as well as to other theory of mind such as simulation theory [71].

2.2.2 Decision Making Theory

Decision making theory models individual decision makers who have to choose between a set of options by processing actions, plans or strategies [152]. This section presents the decisional framework and introduces the notation used to model decision problems.

Situations with uncertainty are the most interesting from a decision theoretic point of view. An example of situations with uncertainty, consider buying a lottery ticket. In this situation, we can not choose to buy the winning ticket. In this situation, agent's actions depend on random or nondeterministic occurrences in the environment. Models of decision making under this uncertainty would then be represented as an action that gives the agent a certain probability of winning. Another sort of uncertainty could arise when the outcomes of the agent's decisions depend on the actual state of the world, which he has only partial information. In uncertain situations, maximization of expected payoffs is the most widely accepted framework to solve this decision problem.

2.2.2.1 The Basic Elements of a Decision

The basic elements of a decision problem are [59, 147]:

(1) the choices space, which represent the alternatives available to the decision maker;

(2) the state space, which represent the state of nature which is not controlled by the decision maker;

(3) the outcome or the payoff that needed to compare each combination of decision choices and state of nature;

(4) Utility evaluation that represent the quantification of a decision maker preferences.

2.2.2.2 The Rational Choice

The rational choice theory states that rational agents always make reasonable and logical decisions. These decisions provide the rational agent with the greatest benefit or satisfaction, given the set of options available, and are also in their highest self-interest.

The rational decision makers are assumed to have ranking over the set of choices, which usually represented as a choice function. This function reflects the way the agents would choose between pairs or sets of options.

In Situations where there is no uncertainty, usually it is clear how to reconstruct the agent's preferences over outcomes. From the rational perspective, given that each action yields a certain outcome, about which the agent has some preferences, the rational agent will choose the actions that yield a most preferred outcome. The rational agent also assumes to the account of the utility functions and probability distributions in determining preferences and interactions between individuals [59, 147].

2.2.2.3 Probability vs. Utility

In decision research, in a situation where we deal with uncertainty, we will need to make use of probabilities, and utility. We will, therefore, review the basic concepts of probabilities and utility theory.

Probability Theory Probability theory presents a consistent framework for the quantification and manipulation of uncertainty. In situations where events are uncertain, a probability measures the likelihood that a particular event (or set of events) occurs [27] (see Sec. 2.2.3.2 for more details).

Utility Theory Utility theory provides a consistent framework for the judgment of alternative choices made by individuals. The notion of utility is introduced to quantify preferences among various choices that a decision maker may

faced. Utility theory based on the assumption that any decision made on the basis of the utility maximization principle, where the best choice is the one that provides the highest utility(score) to the decision maker. Also, in application utility theory is used to represent preferences among potential (or obtained) outcomes of a decision.

Yates [215] pointed out that, there are two ways to relate preferences to the objective values of the outcome. The first one is called value function. This function represents the increase in the strength of the decision maker's preferences as a function of the outcomes' objective value. This function produces various outcomes outlined on a scale of higher values and lower values, where the higher values are called higher preference. The second way is called a utility function, where the assumption is that preference reflects both the value of the outcome and the feelings of the decision maker about risk. For example, the uncertainty about whether the outcome will occur or not.

In all situations, the utility function U used to measure the utility that the decision maker gets from selecting a specific choice. This function is a mathematical representation of the decision maker's system of references. For example $U(x) > U(y)$, where the choice x is preferred over choice y or $U(x) = U(y)$, where both choices preferred equally. In this case, the utility function represents the utility of the choice and used to derive a numerical score for each choice. In this case, the utilities (scores) assigned to different choices are comparable.

Irrespective of the type of utility function, according to [215] utility theory make three fundamental assumptions:

- Connectivity: that assume that the decision maker can judge his or her preferences (or indifference) when faced with two choices.
- Transitivity: They assume that preferences among multiple choices is transitive. For example: if any three choices x, y, z such that x favored over y, and y is favored over z, it is concluded that x is favored over z.
- Summation: Which assume that the preference for a sequence of choices is greater than the preferences for any of its parts [215].

2.2.2.4 Matching vs. Maximization Strategies

Decision-making theory concern on studying different behavioral strategies such as maximizing and matching strategies in various behavioral circumstances.

Matching Strategy Some experiments show that animals and humans often exhibit matching behavior in a variety of decision-making tasks [188, 89]. Herrnstein et al. [88] studied this phenomenon and expanded it into a general principle of choice that he named the matching law.

The matching law states that fraction choices made to any option is proportional to the amount of past income (i.e., total reward) earned from that option or

$$\frac{I_k}{\sum I} = \frac{C_k}{\sum C},$$

where I_k and C_k represent the total amount of income and total choices on option obtained on option k, respectively, and the summations are over all available options.

There are several decision-making models proposed to reproduce the matching behavior (see for example ([188, 89]). Sugrue et al. [188] argued that, in order to match behavior to income for particular behaviors, first the animals must integrate the earned rewards, and then, the brain must maintain an appropriate representation of the reward value of competing alternatives. Sugrue et al. [188] studied matching in the context of visually based eye movement behavior. They used Macaca monkeys to carry out a dynamic version of a conventional matching task in which saccadic eye movements to a pair of competing visual targets are rewarded at different rates. They found that a simple model based on reward history could duplicate this behavior. That neurons in the parietal cortex represent the relative value of competing actions predicted by this model (see [188] for more details).

Maximization Strategy Examples of the maximizing strategy can be seen in the stochastic process of solving Markov decision process. It is considered that subjects attempt to choose a behavioral policy that will maximize the amount of reward under a given environmental condition. Also, there are many developed frameworks for representing decision-making situations with a goal of reward maximization. or slightly more generally, optimization of a given cost function (see section 2.2.2.5 for more details). Another example of the maximizing strategy can be seen in expected payoff maximization in game theory [150].

2.2.2.5 Decision-Theoretic Models

There are many developed frameworks for representing decision-making situations with the goal of representing the factors that influence the optimal decision. Almost of these frameworks describe behaviors as a sequence of interactions with a stochastic process that maximize expected utility. We introduce these normative decision-theoretic frameworks to use it to predict user behavior later in this thesis.

2.2.2.6 Markov Decision Processes (MDPs)

One typical framework for representing decision-making and planning is the Markov Decision Processes (MDPs), which represent the decision processes in term of states, actions, rewards associated with those states, and transition probabilities.

Definition 1. A Markov decision process (MDP) is a tuple, $M = (S, A, T, \gamma, R)$ where

- S is the set of possible states.
- A is the set of possible actions.
- $T : S \times A \times S \longrightarrow [0, 1]$ is a transition probability function
- γ is a discount factor, controls the comparative worth of reward at various points in the future.
- $R : S \times A \longrightarrow \mathbb{R}$ is a reward function, with absolute value bounded by Rmax.

MDP works under the assumption that the agent interacts with the world in discrete time steps. The state S_t at timestep t is generated from the probability function based on S_{t-1} and A_{t-1}. The discount factor $1 \geq \gamma \leq 0$, make the future rewards are worthless than the current reward. MDPs allow states and actions to have discrete or continuous values. In this chapter, we consider only discrete spaces.

Choosing the actions that maximize the expected discounted sum of rewards is the goal. This could be solved by defining a policy for action selection as $(\pi(s) \rightarrow A)$, which represents a mapping from states to actions. For a broader overview of MDPs refer to [155].

Optimal policies We can solve the MDPs by finding a policy $(\pi(s) \rightarrow A)$ determining which actions to take in specific states in order to achieve a goal which maximizes the expected discounted sum of rewards $E\left[\sum_{t=0}^{\infty} \gamma^t R^t \mid \pi\right]$ [21].

Theorem 1. *The optimal policy can be computed by solving the Bellman equation,*

$$\pi(s) = \arg\max_a \left\{ R(s,a) + \gamma \sum_{s'} \Pr(s' \mid s, a) V(s') \right\} \tag{2.1}$$

$$V^*(s) = \max_a \left\{ R(s,a) + \gamma \sum \Pr(s\prime \mid s, \pi(s)) V^*(s') \right\}. \tag{2.2}$$

For the optimal value function $V^*(s)$ we can define the optimal action value function $Q^*(s,a)$:

$$V^*(s) = \max_a \{R(s,a) + Q^*(s,a)\}$$

$$Q^*(s,a) = \gamma \sum \Pr(s' \mid s,a) V^*(s').$$

The Bellman equations can be solved recursively using dynamic programming by updating the$V^*(s)$ values and policies $\pi(s)$ iteratively. The value iteration algorithm [21] repeatedly updates $V(s)$ by developing its formula to be in terms of $V^*(s)$ terms. The policy iteration algorithms uses equation 2.1 to compute a policy and then iteratively uses the updates of equation 2.2 until it converge [155].

Even though MDPs can be used to model a large number of simplistic tasks, in modeling real world tasks, MDPs assume that the agent has the complete knowledge about the state of the world at all times, but this assumption not realistic in many tasks.

2.2.2.7 Partially Observable Markov Decision Processes (POMDPs)

As mentioned above, MDPs assume that the agent has the complete knowledge about the state of the world every time step. In real-world tasks, this assumption is unrealistic. For example in situations where the agent's sensors are limited and noisy. Which means that the agent only perceives part of the world and because of sensors noise the perceived information is just a projection of the real world state.

The POMDPs framework extend the MDPs to settings with uncertain States [56]. A POMDPs models an agent decision process in which the full state S may only partially be known.

Definition 2. A partially noticeable Markov decision process (POMDP) is a tuple, (S, A, O, T, Ω, R), where

- S is the state space.
- A is a set of actions.
- O is a set of observations.
- T is a set of conditional transition probabilities.
- Ω is a set of conditional observation probabilities.
- $R : S \times A \longrightarrow \mathbb{R}$ is the reward function.

The state S_t at timestep t is generated from the probability function based on S_{t-1} and A_{t-1}, but the observation variable, O_t, distributed according to the state is observed before the next action (A_t) is selected.

In POMDPs sitting the state s is not completely observable so that the agent has to estimate, based on the observations O, a posterior distribution over all possible states. This posterior distribution is known as the belief state or information state. It is difficult to develop exact algorithms for solving POMDPs that is the problem. The most common algorithms just approximate an optimal solution [38, 167].

2.2.2.8 Game Theory

Game theory, which was originally developed in economics has come to provide a very effective quantitative framework for studying how different sources of information, social knowledge, and economic incentives impact optimal strategies for social interaction. In general, game theory in the context of economic decision-making is based on the assumption, that people can predict other people's actions when they understand their motivations, preferences, and beliefs (for a similar argument, see [180]).

Game theory aims to help us understand situations in which decision makers interact [150, 74]. Game theory studies strategic decision making, when many rational decision makers determine the outcome of a decision situation.

In a situation with strategic interaction, each agent - similar to decision theory- chooses among various actions, plans or strategies. The main difference from decision theoretic is that uncertainty in strategic interaction come from choices of other agents and the outcomes in games are determined by a combination of the choices of all agents. The game theory usually divided into cooperative and non-cooperative branches. In cooperative games, communication among players is allowed but it's not allowed in the non-cooperative game. In the strategic game, it is assumed that the agents are rational, which mean that they choose in order to maximize their expected payoffs. The main difficulty is now to specify what are the expected payoffs of an action when its outcome depends on the actions of others.

In situations with incomplete information, the agent might not be able to expect the choices of others, Because they are uncertain about each other's preferences. In such situations each agent forms expectations about the others' decisions before making his own. But each agent also knows that the other will do the same. Which mean that an agent's expectations about the others' actions take into account the fact that the others choose on the basis of what they think he will do. Expected payoff maximization in game theory provides a variety of solution concepts, such as iterated elimination of dominated strategies and Nash equilibrium. For a broader overview of game theory the reader is referred

to [150, 74].

Definition 3. A strategic game G is a tuple $(I, S_i, X, \pi, \succeq)$ such that:

- I is a finite set of agents.
- S_i is a finite set of strategies or actions for each i. A strategy profile $\sigma \in \pi_{i \in I^{S_i}}$ denotes a vector of strategies, one for each agent in I.
- X is a finite set of outcomes.
- $\pi : \Pi_{i \in I^{S_i}} \rightarrow X$ is an outcome function that assigns to every strategy profile $\sigma \in \Pi_{i \in I^{S_i}}$ an outcome $x \in X$.
- $\succeq_i$ is a reflexive, transitive and total preference relation on X.

The definition of the outcome function captures the idea that outcomes determined by the choices of all agents [150].

2.2.3 Probabilistic Machine Learning

Machine learning is the study of methods of getting computers to act by learning from experience. Machine Learning methods are suitable in situations where people are unable to present accurate specifications for desired program behavior; instead the examples of target behavior are available. In the past decade, machine learning was successful in situations such as automated steering of automobiles, effective web search, speech recognition, handwriting recognition and quickly better understanding of the human genome. Also, machine learning have been applied in situations where the task is changing over time or across different users, where it is difficult to anticipate exactly how the program should behave. For example predicting user browsing behavior on the world-wide web, refining information retrieval queries and filtering news articles [27, 185, 139]. In this section, I discuss an important concept from machine learning.

2.2.3.1 Learning Problems

The range of learning problems is large. There are several ways an algorithm can model a problem based on its interaction with the experience or environment [27, 185, 139]. There are few examples of algorithms and problem types that machine learning algorithms can categorized to:

Classification The problem of constructing a model of a process from samples of the process's input and output - where the output is one of a discrete set - called classification. Binary classification is the most studied problem in machine learning. In a simple form, we can reduce it to the argument: Given a pattern x selected from a domain X , we want to find out the value of an associated binary random variable $y \in \{+1, -1\}$ [27].

In this thesis, I use a learning approach to train a classifier directly from human eye tracking data. We used a support vector classifier (see section 2.2.3.7) to learn the difference between positive and negative examples.

Regression In Supervised learning when the output is one or more real numbers, it is called regression. In general the regression algorithm creates a model $\tilde{f}$ that accurately approximates a target function f.The algorithm is given a set of training data set $X = \{(\overrightarrow{x_1}, y_1), (\overrightarrow{x_2}, y_2), ..., (\overrightarrow{x_n}, y_n)\}$ of samples of input and output from f. When the model $\tilde{f}$ is learned, it can be used to predict to predict outputs $\tilde{f}(\overrightarrow{x_q})$ for any unlabeled query $\overrightarrow{x_q}$ [27].

Structured Estimation Structured prediction problems go beyond simple multiclass estimation. It deals with hidden variable discovery. This problem appears in many problems where multiple decisions must be weighed against each other to find a globally satisfactory and consistent solution. With the assumption that the labels y have some additional structure that can used in the estimation process. For examples, when attempting to classify web pages, y might be a path in an ontology and when attempting to match objects y might be a permutation. Each of those tasks has its properties in terms of the set of labels y that we might consider admissible, or how to search this space. Max-margin training of structured models as HMMs and PCFGs has become popular for this type of problem in recent years (For a broader overview refer to [185, 139]).

Novelty Detection This problem describes the issue of determining unusual observations given a set of past measurements. It is one of the basic requirements of a good classification system. Novelty detection is a difficult problem in machine learning. A commonly accepted notion is that unusual events occur rarely (see [185, 139] for more details).

2.2.3.2 Probability Theory

Probability theory presents a consistent framework for the quantification and manipulation of uncertainty. In situations where events are uncertain, a probability measures the likelihood that a particular event (or set of events) occurs [27].

The probability of an event defines as the fraction of times that event occurs out of the total number of trials. Here we will introduce the basic concepts of probability theory by considering random variables, X takes the values x_i where $i = 1, ..., M$, and Y takes the values y_j where$j = 1, ..., L$. Take into account a total of N trials where we sample both of the variables X and Y. Let n_{ij} represent the number of trials in which $X = x_i$ and $Y = y_j$.

The probability distribution of a random variables, such as X and Y, is denoted by $P(X)$ and $P(Y)$. The random variable X can be either continuous or discrete. Although, both cases are described, here we focus on the discrete case.

A joint probability measure is written as $P(X, Y) = P(Y \mid X)P(X)$. This joint probability represents the possibility space and can be used to determine other probability measure:

- The marginal probability measure denoted by: $P(X) = \sum_Y P(X, Y)$ which obtained by marginalizing the other variable in this case Y.
- The conditional probability measure denoted by: $P(Y = y_j \mid X = x_i)$.
- If we consider the symmetry property of the joint probability function $P(X, Y) = P(Y, X)$ we can obtain the Bayes' theorem:

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)},$$

  which plays a primary role in machine learning and decision making.

In the Bayes' theorem $P(Y)$ called the prior probability and $P(X \mid Y)$ called the likelihood probability. Then we multiply the prior probability by the likelihood to obtain the posterior probability $P(Y \mid X)$.

If X and Y are said to be independent, the joint distribution of two independent variables factorizes as the product of the marginals, $P(X, Y) = P(X)P(Y)$ [27].

2.2.3.3 The General Setting of the Learning Problem

The learning problem can be described as the problem of minimizing the risk functional based on empirical data. Vapnik [198] described the general learning problem as follows:

Let the probability measure $P(z)$ defined on the space Z. Consider the set of functions $Q(z, \alpha)$, $\alpha \in \Lambda$.

The goal is: to minimize the risk functional

$$R(\alpha) = \int Q(z, a)\, dP(z), \quad \alpha \in \Lambda \tag{2.3}$$

if probability measure $P(A)$ is unknown but an i.i.d. sample

$$z_1, ..., z_l \tag{2.4}$$

is given. Where z describes a pair (x, y) and Q is the specific loss function.

2.2.3.4 Empirical Risk Minimization Principle

In order to minimize the functional risk 2.3 for the probability measure $P(z)$, usually the expected risk function is replaced by the empirical risk function [198].

$$R(\alpha) = \frac{1}{l} \sum_{i=1}^{l} Q(z, \alpha) \tag{2.5}$$

contracted on data set presented in 2.4.

This principle called the empirical risk minimization induction principle. Which approximate the function $Q(z, \alpha_0)$ to minimizes risk of 2.3 by using 2.5.

In order to specify the regression problem, one could introduce an $n + 1$ dimensional variable $z = (x, y) = (x_1, ..., x_n, y)$. Then by use the loss function, the empirical risk will be:

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i=1}^{l} (y_i - f(x, \alpha))^2 \tag{2.6}$$

which we want to minimize, to find the regression estimate by using, for example, the least square method [198].

2.2.3.5 Generative vs. Discriminative Learning

In probability and statistics, a generative classifier learns a model of joint probability $P(x, y)$ over observation x and label y sequences. The prediction of the generative model make use of Bayes theory to calculate $P(x \mid y)$, and picking the most likely y [139].

Discriminative classifiers are a class of models used in machine learning for modeling the dependence of an unobserved variable y on an observed variable x. In other words it model the posterior $P(y \mid x)$ directly, or learn direct map from input x to the class labels [139].

Application specific details prescribe the convenience of selecting a discriminative versus generative model.There are different reason for using discriminative rather than the generative model. For example, Vapnik [198] articulated that: "one should solve the classification problem directly and never solve a more general problem as an intermediate step (such as modeling $P(x \mid y)$)".

2.2.3.6 Supervised Learning

In supervised learning, the task is to infer a function from supervised training examples. In supervised learning, each example is a pair of the form (x_i, y_i) consisting of an input object x_i usually an n-dimensional vector and each output value y_i is a scalar. A supervised learning algorithm takes a set of training examples as input and produces an inferred function as output. If the output is discrete, the inferred function called a classifier, and it called a regression function if the output is continuous. The prediction of the inferred function should be the correct value for any valid input object. Therefore, the learning algorithm requires generalizing from the training data to unseen situations in a reasonable way [27].

In the next section, we discuss the linear support machine as an example of the supervised learning algorithm, which we used later in the thesis to learn eye movements models.

2.2.3.7 Support Vector Machines (SVM)

Here we discuss standard SVM problem (See [45, 33] for more details). We first consider a linear machines trained on separable data, where we are given the labeled training data (x_i, y_i), $i = 1, .., l$, $x_i \in \mathbb{R}$, $y_i \in \{1, -1\}$. Our aim is to define the "margin" of a separating hyperplane. Assume we have a separating hyperplane that separates the positive from the negative examples, where all the points x that lay on the hyperplane fulfilling: $w.x+b = 0$, where w represent the normal vector to the hyperplane, $\frac{|b|}{\|w\|}$ is the orthogonal distance from the hyperplane to the origin and $\| w \|$ is the Euclidean norm of w. In our case, the support vector algorithms look for the separating hyperplane with the largest margin.

Assume that the training data satisfy the following constraints:

$$x_i.w + b \geq +1 \ , y_i = +1 \tag{2.7}$$

$$x_i.w + b \leq -1 \ , y_i = -1 \tag{2.8}$$

which could be combined into the following inequalities:

$$y_i(x_i.w + b) - 1 \geq 0 \quad \forall i \tag{2.9}$$

The points for which the Equation 2.7 holds lie on the hyperplane $H_1 : x_i.w+ b = 1$ with normal w and orthogonal distance from the origin $\mid 1 - b \mid / \| w \|$. Similarly, the points for make the Equation 2.8 holds lie on the hyperplane $H_2 : x_i.w + b = -1$, with normal w and orthogonal distance from the origin $\mid -1 - b \mid / \| w \|$. While H_1 and H_2 are parallel and they have the same

normal. One can find the pair of hyperplanes which gives the maximum margin by minimizing $\| w \|^2$, subject to constraints (2.9).

By introducing positive Lagrange multipliers $\alpha i, i = 1, \cdots, l$, for each inequality constraints in (2.9). One can write as :

we solved the standard SVM problem

$$L_P = \frac{1}{2} \| w \|^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i.w + b) + \sum_{i=1}^{l} \alpha_i \tag{2.10}$$

We must now minimize LP with respect to w, b. Requiring that, the gradient of L_P with respect to w and b be small given the conditions:

$$w = \sum_i \alpha_i y_i x_i \tag{2.11}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.12}$$

The solution of 2.10 take form of:

$$L_D = \sum_i \alpha_i - \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \tag{2.13}$$

In the separable case (linear case) Support vector training, therefore, amounts to maximizing L_D with respect to the α_i, subject to constraints (2.12) and positivity of the α_i, with solution given by (2.11).

If we applied the above algorithm to separable data, will find no feasible solution. Because the objective function (i.e., the dual Lagrangian) will be growing arbitrarily large. Cortes and Vapnik [45] extend the idea to handle non-separable data by introducing positive slack variables ξ_i, $i = 1, \cdots, l$ in the constraints of Equations. 2.7 and 2.8. The dual Lagrangian become:

Maximize:

$$L_D = \sum_i \alpha - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i.x_j \tag{2.14}$$

subject to:

$$0 \leq \alpha \geq C$$

$$\sum_i \alpha_i y_i = 0$$

The solution is given by:

$$w = \sum_{i=1}^{Ns} \alpha_i y_i x_i. \tag{2.15}$$

where NS is the number of support vectors. So the only difference from the optimal hyperplane case is that the αi now have an upper bound of C [45].

In general the SVM have some important properties: It has a unique solution for the constructed optimization problem, and the learning process is quite fast. Also, It obtains a set of support vectors together with constructing the decision rule. Also, it is possible to implement a new set of decision functions by changing only one function (i.e., the kernel function).

2.2.3.8 Controlling the Generalization of Learning Machines

The theory for controlling the generalization of a learning machine described in details in [198]. Such theory dedicated to forming an induction principle for minimizing the risk functional, where the size of the training set should be considered. The goal is to identify methods that are appropriate for a given sample size.

The empirical risk principle applies to a large sample size. Another principle called the principle of structural risk minimization (SRM), intended to minimize the risk functional with respect to both empirical risk and VC-dimension of the set of functions. The concept of VC-dimension is based on relevant properties of the growth-function [198].

Let S represents the set of functions $Q(z, \alpha)$, $\alpha \in \Lambda$, be provided with structure: so that S is composed of the nested subsets of functions $S_k = \{Q(z, \alpha),\ \alpha \in \Lambda_k\}$ so that

$$S_1 \subset S_2 \subset ... \subset S_n \subset ... \tag{2.16}$$

and $S^* = \cup_k S_k$.

An admissible structure should satisfy the following three properties:

1. The set S is everywhere dense in S^*.
2. The VC-dimension $h_k$of each set S_k of functions is finite.
3. Any element S_k of the structure contains totally bounded functions $0 \leq Q(s, \alpha) \leq B_{k,\alpha} \in \Lambda_k$.

In general the SRM principle suggests a tradeoff between the quality of the approximation and the complexity of the approximating function (For a broader overview of SRM principle refer to [198]).

2.2.3.9 Why Can Support Vectors Machines Generalize?

The generalization ability of the support vectors networks is based on the factors described in the previous section.

According to the theory of controlling the generalization of the learning processes [198], to ensure a high percentage of generalization of the learning machine. First one has to construct a structure $S_1 \subset S_2 \subset ... \subset S_n$ on the set of decision functions $S_k = \{Q(z, \alpha), \alpha \in \Lambda_k\}$. and then select a convenient element S_k of the structure and a function $Q((z, \alpha_l^k) \in S_k$ which minimizes the bound. The bound can be formulated simply as follow

$$R(\alpha_l^k) \leq R_{emp}(\alpha_l^k) + \Omega(\frac{l}{h_k}) \tag{2.17}$$

where the estimation of the risk represented in the first term, and the confidence interval for this estimation represented in the second term.

In support vector methods, we can control both properties. Where In the separable case one obtains the unique solution that minimizes the empirical risk using a margin separating hyperplane with the maximal margin (which is a subset of the smallest VC dimension). In the general case, it obtains the unique solution when one chooses the value of the trade-off parameter C [198].

2.3 Computational Visual Attention Models

Information manipulation and interpolation is one key problem in perception. The rich streams of visual information continuously enter our visual system from the surrounding environment. Processing this much information in real time can be very expensive to our brain. Thus, with the help of a clever mechanism, our brain makes decisions on which information will be selected for further processing. This mechanism called selective attention.

Attention is a general concept covering all important factors that influence selection mechanisms. There are two types of processing visual attention have been suggested in many visual attention studies, which are: scene-driven bottom-up and expectation-driven top-down.

Saliency maps identify important regions of a scene that seem to an observer as an out-layer relative to their neighboring parts. Saliency map models are often considered in the context of bottom-up computations. Computational models of visual saliency are widely used to predict gaze locations. Usually, these models vary in details but they have a similar structure.

In this section, we first present the common structure to the most visual saliency models in section 2.3.2. We then review some important existing computational attention systems 2.3.3. We then introduce a set of applications of visual attention models in section 2.3.5.

2.3.1 Eye Movements and Visual Attention

The inhomogeneity of the retina is the most important property of human eye. The central part of the retina known as the fovea has a high-resolution central and a low-resolution periphery. Also, the fovea represents a very small region of the retina, with an angular diameter between 0.3°and 2° [186]. Because to this property the eyes move in order to obtain a detailed view of the whole scene. There exist numerous types of eye movements. The saccadic eye movements were the most studied ones. The goal of saccades is to shift the fovea onto a given target to obtain high resolution samples. When we exploring a given scene, we shift our fovea to a set of targets, creating the so called scan path.

Psychophysical studies of humans eye movements have demonstrated how these saccades are generated. One of the earliest studies was made by Yarbus in 1967. Yarbus [214] hypothesis was that fixation duration and gaze patterns vary according to the current cognitive goals or the task being performed by the subject. In his study, the subjects were asked to watch the same scene with different conditions such as "find out the material circumstances of the family", "What are the ages of the people?", or simply to freely examine the scene. Eye movements differed considerably see Figure 2.2.

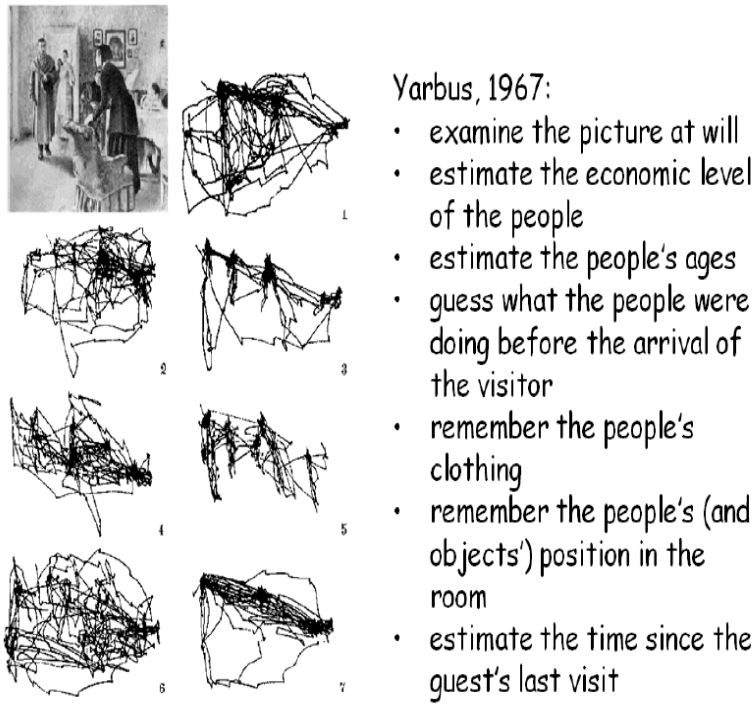


Figure 2.2: Eye movements paths of subjects whilst scanning a picture with different questions (from [214]) .

2.3.2 General Structure

In this section, we illustrate the common structure for the most computational visual saliency models. The basis of many attention models goes back to Feature Integration Theory introduced by Treisman & Gelade [196]. The first algorithmic model for a computational architecture of visual attention was introduced by Koch and Ullman [107]. The main concept is that different features are extracted in parallel, and their conspicuities are collected in a saliency map (see Figure 2.3).

More specifically, the following steps are included in the models processing:

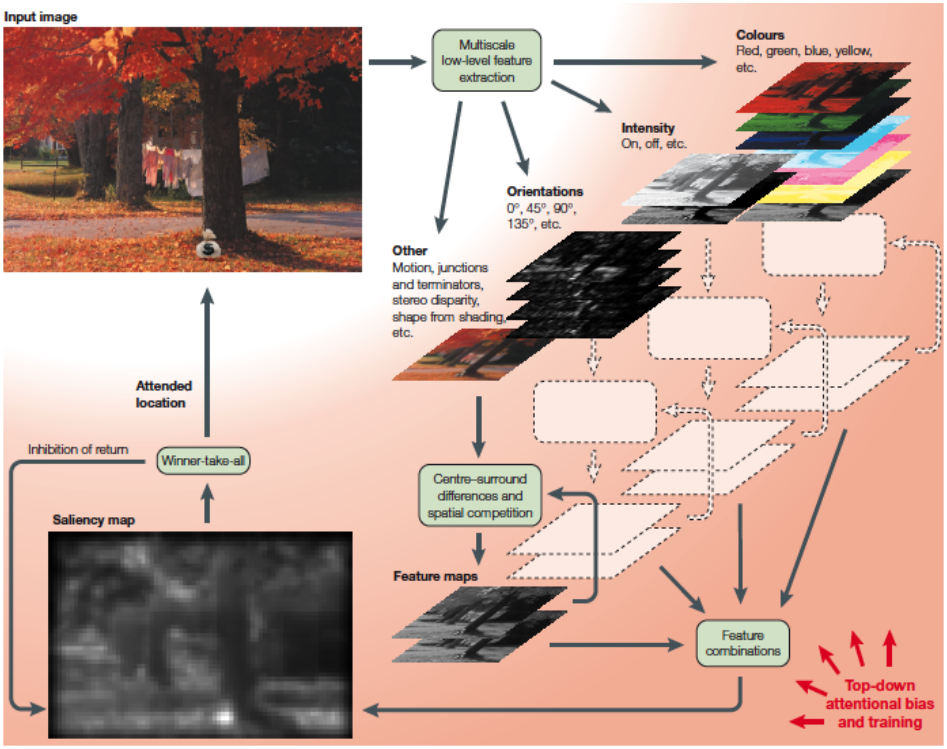


Figure 2.3: The general structure of the bottom-up attention model (from [97]).

First visual input is decomposed into a set of topographic feature maps based on one or several image pyramids. Different pre-attentive feature detection mechanisms (sensitive to color, intensity, orientation and so on), which operate in parallel over the entire visual scene. Each feature map is discomposed into several features types (such as r, g, b maps of color).

Followed by a center-surround mechanism or differences of Gaussian which extracts extract local spatial discontinuities for each features types. This op-

eration computes the average value of the center region and compares it with the average value of a surrounding region. Then the features maps are summed together to generate the conspicuity maps. Finally, the conspicuity maps are normalized and combined together to form the saliency map.

The saliency map gives the saliency for each region of a scene. But some application interested in the trajectory of image regions. This could be represented through the interplay between a winner-take-all network, which finds the point of highest saliency at any given time, and inhibition-of-return, which suppresses the last attended location from the saliency map. So that that attention can focus on the next most salient location [98, 97].

Most models consider only bottom-up mechanism described above and ignore the important significance of top-down cues. Including top-down knowledge also possible by modulating the weight of the conspicuity maps before they are combined with other top-down knowledge such as context information [97].

In the next section, we discuss specific examples of computational models of visual saliency.

2.3.3 Overview of Existing Computational Models

The basis of many attention models goes back to Treisman & Gelade's [196]. They stated that visual features are combined to direct attention. Then, this informal concept was formalized as a saliency map by Koch and Ullman [107]. Later, Itti et al. [98] proposed the first computer implementation of this model, where the visual input is decomposed into a set of different feature maps (color, intensity, and orientation). Different spatial locations compete within each map, and then maps are combined into a master saliency map.

This idea behind saliency maps that used in other studies, where it was extended and further developed. For example, Mahadevan and Vasconcelos [72] presented a discriminant formulation of center-surround saliency for static images. Indeed, one can view their work as a normative approach, because they first formulate the saliency map computation as a problem, and then derive their algorithm as the solution to this problem. More specifically, they consider saliency as a decision-making task informed by natural image statistics. The outcome of their work is an automatic selection of the important features. This improves the original Itti & Koch model, where the features selection and combination was done in a heuristic way. This was later also extended to dynamic scenes and movies using dynamic textures [121]. However, the original Itti & Koch model was also improved recently using graphs to compute saliency [85]. This shows that the concept of the computational saliency maps is still very fruitful and can guide research in predicting eye movements.

These saliency-based models are all based on low-level image features. De-

spite this limitation, they often predict gaze well, but mid- and high-level features also affect gaze. Therefore, Judd et al. [101] pursued a machine learning approach: They learned gaze points based on measured eye movements using a linear SVM. They report better predictions than Itti & Koch on 1003 images observed by 15 subjects [101].

All these previous works predicted gaze using only information from the images. It is known since the early days of eye movement research that the task-demands affect the patterns of eye movements [214]. A more recent study re-investigated and confirmed this: Hayhoe et al. [86] proposed that there is a strong relationship between eye movements and visual cognition when dealing with complex tasks. Subjects performing a visually-guided task were found to direct the majority of fixations toward task-relevant locations.

2.3.3.1 Itti and Koch

The Itti and Koch model was inspired by biological concepts from cognitive science and based on a bottom-up computational model [98]. We introduced the computational architecture of visual attention proposed by Koch and Ullman [107] in Sec. 2.3.2. Itti et al. [98] proposed the first computer implementation of this model. This model has been the basis for later models and is a standard benchmark for comparison. An input image is subsampled into a Gaussian pyramid, and each pyramid level is decomposed into channels for color, intensity, and local orientations. From these channels, center-surround *feature maps* for different features are constructed and normalized. Finally, conspicuity maps are linearly combined once more to generate the saliency map.

2.3.3.2 Torralba Saliency (T-Saliency)

This model combines sensory evidence with prior constraints. Prior knowledge (e.g., scene context or gist) and sensory information (e.g., target features) are combined according to Bayes' rule. The presented architecture for attention guidance consists of three parallel modules extracting different information: bottom-up saliency, object-centered features, and contextual modulation of attention. The focus was on showing how to introduce global scene factors to model the contextual modulation of local saliency. The proposed model learns the relationship between global scene features and local object properties (identity, location, and image scale). The drawback of this method is that it need a priori assumptions about features that contribute to salience (See [195] for more detail).

2.3.3.3 Graph-Based Visual Saliency (GBVS)

This model is based on a probabilistic framework in which a graph denotes the conditional independence structure between random variables. This model treats eye movements as a time series. Since there are hidden variables influencing the generation of eye movements, a Hidden Markov Models (HMM) approach was been incorporated. In this model, feature maps are extracted at multiple spatial scales. Then, a fully-connected graph over all grid locations of each feature map is built. Weights between two nodes are assigned proportionally to the similarity of feature values and their spatial distance. The resulting graphs are treated as Markov chains by normalizing the weights of the outbound edges of each node to one and by defining an equivalence relation between nodes and states, as well as between edge weights and transition probabilities. The activation maps are finally normalized to emphasize conspicuous detail, and then combined into a single overall map. Graphical models could be viewed as a generalized version of Bayesian models. That enables them to model more complicated attention mechanisms. The disadvantages lie in model complexity, especially when it comes to training and readability (See [85] for more detail).

2.3.4 Performance Measures

To better understand the relationship between a viewer's fixation locations and the predictions of the saliency models, we have to evaluate it quantitatively by comparing it with eye movement data. We used the following four performance measures that are widely used in the state of the art of visual attention literature, to evaluate the performance saliency models. Because the evaluation measures for attention modeling can be classified into point-based and region-based, we used four performance measures to deal with this perspective.

2.3.4.1 Kullback-Leibler (KL) Divergence

KL divergence measures the distance between distributions of saliency values between human and random eye positions [29, 97]. We used KL because it is sensitive to differences between histograms, where other measures essentially calculate the rightward shift between two histograms. Furthermore, KL is invariant to reparameterizations, such that applying any continuous monotonic non-linearity to estimated saliency map values. Let $i = 1 \ldots N$ be N human eye positions in the experimental session. For a given saliency model, the estimated saliency map is sampled at the human saccade $X_{i,human}$ and at a random point $X_{i,random}$. First, the saliency magnitude at the sampled locations is normalized to the range [0,1]. Then, a histogram of these values in $q = 10$ bins across all eye positions is calculated. $\Pr\left(X_{human}\left(i\right)\right)$ and $\Pr\left(X_{random}\left(i\right)\right)$ are the sub-

sets of points in bin i for salient and random points, respectively. Finally, the difference between these histograms was measured using KL divergence

$$KL\left(X_{human}; X_{random}\right) = \sum_{i}^{q} \Pr\left(X_{human}\left(i\right)\right) \log\left(\frac{\Pr\left(X_{Human}\left(i\right)\right)}{\Pr\left(X_{random}\left(i\right)\right)}\right). \quad (2.18)$$

Models show higher KL divergence, are better in predicting human fixations, because usually human gaze towards the regions with the highest model responses and avoiding the low model responses regions.

2.3.4.2 Area Under Curve (AUC)

AUC is the area under the *Receiver Operating Characteristic* (ROC) curve [79]. ROC is used for evaluating a binary classifier system with a variable threshold. Using this measure, the model's ESM is treated as a binary classifier on every pixel in the image. Considering Pixels with larger saliency values than a threshold are classified as *fixated* while the rest of the pixels are classified as *non-fixated.* Human fixations are then utilized as ground truth. Via changing the threshold, the ROC curve is drawn as the false positive rate vs. true positive rate and the area under this curve indicates how well the saliency map predicts actual human eye fixations [32]. Perfect prediction corresponds to a score of one. This measure has the desired characteristic of transformation invariance: the area under the ROC curve does not differ when applying any monotonically increasing function to the saliency measure (See [92] for further details about ROC calculation).

2.3.4.3 Linear Correlation Coefficient (CC)

This error measure is widely used to compare the relationship between two images for applications such as disparity measurement, object recognition, and image registration [99, 156]. Clearly the linear correlation coefficient measures the strength of a linear relationship between two variables:

$$CC\left(G, S\right) = \frac{\sum_{x,y}\left(G\left(x,y\right) - \mu_G\right).\left(S\left(x,y\right) - \mu_S\right)}{\sqrt{\sigma_G^2.\sigma_S^2}}, \quad (2.19)$$

Where G and S represent the fixation map (a map with 1's at fixation locations, usually convolved with a Gaussian kernel) and the estimated saliency map, respectively. μ and σ represent the mean and the variance of the values in these maps. An attractive advantage of CC is the capacity to compare two variables by providing a single scalar value between -1 and 1. When the correlation is close to 1 there is essentially a perfectly linear relationship between the two variables.

2.3.4.4 Mean Squared Error (MSE)

In statistical modeling, the MSE is representing the difference between the actual observations and the observation values predicted by the model [94]. This measure is widely used to compare the various image compression techniques. We used to measure the difference the between actual human fixation map and the saliency map. The mean-squared error (MSE) between two maps $G(x, y)$ and $S(x, y)$ is:

$$MSE(G, S) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \left[G\left(n, m\right) - \left(S(n, m)\right] \right]^2 \tag{2.20}$$

Where G and S represent the fixation map and the saliency map, respectively.

2.3.5 Applications of Visual Attention Models

By using computational attention models, we can build a smart application. Also, we can improve technical systems in many field such as computer vision, robotics, human-computer interaction and computer graphics. In this section, we discuss several application scenarios for attention models.

2.3.5.1 Computer Vision

Image segmentation and detecting regions of interest are important methods in computer vision. In automatic segmentation of images, we need first to set the starting points for segmentation (seeds) and then choosing the similarity criterion to segment regions. Achanta et al. [5] present an approach where the saliency regions of the attention system work as expected candidates for the seeds and the homogeneity criterion is adapted according to the features that discriminate the region to be segmented from its surroundings.

Visual saliency model can be used to enhance object recognition. One example of a combination of an attentional front-end with a biological object recognizer is presented in [129]. The biologically object recognizer HMAX [162] focuses on simulating processes in human cortex, and it is able only to recognize simple artificial objects such as circles or rectangles. Miau et al. [129] used support vector machine algorithm to detect objects in natural images. A similar system proposed by Walther and Koch [203], where the visual attention model combined with an object recognizer based on SIFT features [120] and it found that the recognition results are improved.

Another interesting application scenario presented in [151] is the use of saliency algorithms in the application of image compression. Image compression algorithms could be enhanced by compressing not interesting regions more than

regions that are attended. In [96] saliency model is used for video compression. Gue et al. [84] used multiresolution spatiotemporal saliency model in image and video compression.

2.3.5.2 Computer Graphics

Model of saliency can help in automatic image cropping techniques. Santella et al. [169] presented an interactive method for image cropping given information about gaze location, provided by eye tracking.

Suh et al. [189] and Chin et al. [40] used visual saliency models to identify important image areas and built automatic image cropping systems that require no user input. These systems identify important image content and compute the best crop for any given size or aspect ratio. Which could be used in many applications such as automatic snapshot re-composition, adaptive documents and thumbnailing.

Grbli et al. [78] used saliency data to identify regions of intensest which help to determine the level of details appropriate to stylize and abstract photographs to make them more understandable.

In content aware media, where we have a variable of platform and display sizes. Images and videos should be changed to fit the aspect ratio of that platform. In Holtzman et al. [90] used saliency map in the re-targeting methods as a cost function to define which pixels are the least important to prioritize them to be removed before the important pixels. The same idea was suggested by [75].

2.3.5.3 Robotics

Directing a camera to interesting scene regions and/or zooming these regions are of interest for active vision research. The goal is to acquire data that is as suitable as possible for the current task and to reduce the processing complexity by actively guiding the sensors (usually the camera) to reasonable regions. In these scenarios using attentional models that highlight most salient locations in the video streams are essential. [128] the active vision system NAVIS is presented that uses an attention system to guide the gaze. It is evaluated on a fixed stereo camera head as well as on a mobile robot with a monocular camera head. Other approaches that use attention systems to direct the gaze of an active vision system are described in [42].

Another application scenario of an attention system is in the process of robot localization. The robot had to determine its positions by manipulating its sensor date. Usually using laser scanner fail in outdoor environments. Attentional mechanisms can facilitate the search of landmarks during operation by selecting interesting regions in the sensor data. By concentrating on these

regions and comparing the candidates with trained landmarks, the most probable location can be determined. Siagian and Itti [177] used salient objects in the environments for localization landmarks.

2.3.5.4 Design and Marketing

Models of saliency can be useful in Market research. Companies are interested in knowing how the consumers view with their websites, advertisement or images.

Eye tracking sensors allows the companies to see how the consumers view their websites and advertisements. Models of saliency could be used to predict where people look and reduce the need for using eye tracking devices.

2.4 State of the Art in Related Fields

2.4.1 Cognitive Architectures and Models

A cognitive architecture refers to a theory that specifies the underlying structures for an intelligent system. The goal cognitive architecture is to summarize different aspects of a cognitive agent that does not change over time in a comprehensive computer model, which includes:

- the short-term and long-term memories that reserve information about the agent's beliefs, goals, and knowledge.
- the representation of components that are enclosed in these memories and their structuring into expansive mental structures;
- The functional operations that work on these structures, and its content of the performance mechanisms that use them and the learning mechanisms that change them.

Different cognitive architectures can differ in the specific assumptions they make about these aspects. Also, different architectures can make different commitments about how to characterize and obtain knowledge and beliefs. Newell [142] has argued that we should seek to unify many findings into a single theoretical framework and refine that theory, instead of conduct micro-studies that address only one issue at a time. In his claiming "You can't play 20 questions with nature and win" [142].

Many researchers have proposed and studied cognitive architectures over the past three decades. In the next subsection, I review the most important cognitive architectures.

Examples of Cognitive Architectures

ACT-R

ACT-R (Adaptive Control of Thought—Rational) [12, 11] the most recent cognitive architecture developed by John Robert Anderson, which concerned primarily with understanding how people organize knowledge and produce intelligent behavior. ACT-R has continuous development since the late 1998s. ACT-R distinguish between declarative and procedural representation. Procedural knowledge is a set of all productions, and declarative knowledge is represented in the form of chunks or memory.

ACT-R is structured into a set of modules; each works with a different type of information. The most common modules are sensory modules for visual processing, a declarative module for long-term declarative knowledge, motor modules for action and the goals represented by an intentional module. A short-term memory buffer holds a declarative relational structure associated with each module. The processing of the modules coordinates by a long-term production memory. The activation of each declarative chunk reflects its past usage, and it impacts its retrieval from long-term memory, given that each production has an expected cost and probability of success.

On every cycle, ACT finds out matching productions versus the contents of short-term memory. After that, the system selects the production with the highest utility and executes its actions.

Learning in ACT-R involves creating new facts and productions in addition to updating base activations and utilities associated with these structures.

SOAR

SOAR [111, 112, 143] is a cognitive architecture that has continuous development since the early 1980s. Soar is based on a production system, where the long-term procedural knowledge in Soar takes the form of production rules, which can be described as operators in a problem space. These operators describe simple actions that modify the internal state of the agent or generate primitive external actions, and also describe more abstract activities. Recently separate episodic and semantic memories have been added to Soar to represent these long-term knowledge [111].

Problem solving in Soar formulated as a search through a problem space for a goal state. The basic processing cycle consists of a decision cycle that repeatedly proposes, selects, and applies operators of the current problem space to a problem state, together with a decision procedure. Learning in Soar has multiple mechanisms for different types of knowledge: for example chunking and reinforcement learning need procedural knowledge, but episodic and semantic learning need their corresponding types of declarative knowledge [112].

ICARUS

ICARUS [113, 114] is cognitive architecture works with two distinct forms of knowledge. The concept knowledge encodes classes of environmental situations in terms of other precepts, and skills knowledge determine how to reach goals by decomposing them into ordered subgoals. The performance element first infers all beliefs, which is implicit in its concepts and its perceptions of the environment, then finds applicable path downward through the skill hierarchy to execute. The problem solving occurs when Icarus can find no applicable path whereas learning creates new skills based on traces of successful problem solving.

PRODIGY

PRODIGY [35] encodes two kinds of long-term structures, which are domain operators and control rules. Domain operators describe the effects of actions and the control rules specify when the system should select, prefer, or reject a given operator, binding, state, or goal. The structures of the short-term memories include representations of states and contents of a goal stack. Problem solving use means-ends analysis. Means-ends analysis repeatedly selects an operator to reduce differences between the current goal and state until it finds a sequence that achieves the top-level goal. Learning involves an explanation-based module that analyzes the problem-solving traces and creates a new selection, rejection, and preference rules to reduce search on future tasks. Other modules control search by analogy with earlier solutions, learn operator descriptions from experimentation, and learn to improve the quality of solutions.

CLARION

CLARION [190] used an explicit and implicit form to represents both action-centered and non-action knowledge. By using multi-layer neural networks in the implicit representation and using symbolic production rules in the explicit representation. Corresponding short-term memories carry activations on nodes in addition to the symbolic elements that the architecture matches against long-term structures. Problem solving involves: first passing sensory information into the implicit layer to produce alternative high-value actions, and to the explicit layer to propose actions using the production rules. Then the CLARION architecture selects the candidate that has the highest expected value. Learning involves weight correction by using a combination of back propagation and reinforcement learning to estimate value functions in the implicit system. Also, construction of production rules by extraction from the implicit layer, error-driven revision, and instantiation of rule templates.

2.4.2 Smart Meeting Rooms

The process of human-human interaction during meetings and their technological support have been a subject of research for a long time. In 1987, there was a project called CoLab [187] at Xerox PARC in Palo Alto, California. The focus was to make meetings more effective and to provide the opportunity for research on how computer tools affect the meeting process. Two types of tools were developed, namely to support the group interaction and to provide parallel access to shared objects. A first tool allowed for brainstorming, the collective preparation of a presentation, and for the organization of the meeting agenda. A second tool facilitated the organization and evaluation of arguments for proposals.

In 2001, the NEEM project [60, 61] at the University of Colorado was concerned with improving distributed multimedia meetings. A major novel aspect of the NEEM project was the use of intelligent artificial agents as meeting participants. Goals of this project included the enhancement of distributed group interaction understanding, and the creation and testing of prototype distributed meeting environments. Thus, this system helped in the organization of meetings, in collecting and organizing information, and in facilitating the social interaction between the meeting participants. Some of these projects focused more on collecting meeting data (AMI meeting corpus [126], NIST meeting room pilot corpus [73]), and others on modeling human behavior in meetings (ICSI project [137]).

2.4.3 Human Social Interaction

In human social interaction, meetings are important life activities. It is the place where a group of people comes together, share information, engage in discussions, and make decisions. There have been many improvements in technology-oriented tools to make meetings more efficient. For example, browsing elements of interest within a recorded meeting [206]. Also the usage of tools that allowed parallel access to shared objects [61] or create abstractive summaries [106] (see Sec. 2.4.2). Regarding the social aspects of meetings requires the analysis of different nonverbal communication cues. For example, recognizing meeting activities cues [127] or recognition of roles in meetings [62]. That can lead to the design of efficient tools for computer-enhanced human-to-human interactions (see Sec. 2.4.4).

2.4.4 Activity and Intention Recognition Systems

Work on activity and intention recognition has been done for more than thirty years. Bratman [30, 43] described an important aspect of intentions, which is future-directedness. They even argue that an agent needs to have a course

of actions available to achieve something in the future. Cohen et al. [43] investigate principles dominating the rational balance among an agent's beliefs, goals, actions, and intentions. Those principles provide specifications for artificial agents and approximate a theory of human action. By making precise conditions under which an agent can drop his goals, i.e., by stipulating how the agent is committed to his intentions, the formalism captures some important properties of intention. Specifically, the formalism provides analysis for Bratman's three characteristic functional roles played by intentions [30].

Currently, meeting scene analysis has emerged as research area focusing on peoples' interaction. Several approaches have been made to achieve automatic recognition of group actions in meetings and use statistical methods. For example using Hidden Markov Models (HMMs) [127], layered-HMM [217], coupled-HMM [20], and dynamic Bayesian networks [53].

Helaoui et al. [87] describe the usage of Markov logic as a declarative framework for recognizing interleaved and concurrent activities incorporating both inputs from pervasive light-weight sensor technology and common-sense background knowledge. In particular, they evaluate its capability to learn statistical-temporal models from training data also to integrate these models with the background knowledge to enhance the overall recognition accuracy.

Miquel RamÃrez et al. [158] extend the model-based approach to plan recognition to the Partially Observable Markov Decision Process (POMDP) setting, where states are partially observable, and the actions are stochastic. The task is to indicate a probability distribution over the possible goals of an agent. The POMDP model is shared between agent and observer except for the true goal of the agent that is hidden to the observer. The last approach using POMDP is of special interest for this thesis, because many normative approaches to decision making and planning use POMDPs.

2.4.5 Models of Decision Making and Planing

Bratman [30] have been concerned with the role intentions play in directing rational decision making and guiding future actions, and then Rao et al. [159] proposed an agent model, which models this in a decision-making framework using a symbolic reasoning. However, one important aspect to consider in models of decision making and planning is how agents learn to decide. Reinforcement learning (RL) is a natural framework for that.

Going beyond the classical RL setting, Ng et al. [145] argue that the reward function from RL must be considered as an unknown when examining the animal and human behavior. They proposed algorithms to solve the problem of this inverse reinforcement learning (IRL), i.e., of constructing a reward function given observed optimal behavior. This reward function, which cannot be observed directly, can be considered as part of the internal state of a user,

similar to the state of the attentional system, or the current goal state.

The proper way of including such uncertainties into RL is Bayesian RL (BRL). Baker et al. [17] argue that action understanding is much like visual perception, they characterize vision as inverse graphics and action understanding as inverse planning or IRL. They propose a framework based on Bayesian inverse planning for modeling human action understanding. The underlying assumption here is that agents are rational, and they deal with uncertainties in the optimal way, which is the Bayesian approach.

Most previous decision-making models (i.e., classical RL, IRL, and Bayesian RL) all assume that the agents are learning the parameters of a model. However, the structure of a model may be even more important for human agents than the values of their parameters. Recently, Acuna et al. [6] formulated the problem of structure learning in sequential decision tasks using BRL, and demonstrate qualitative differences in the behavior of optimal learning agents between parameter and structure learning. Thus, a full normative model of decision making in learning agents needs to by a BRL of structure learning. When an agent is observed (by another agent or a smart room), then the reward function should be estimated based on the observed behaviour. This can be considered as a combination of IRL, BRL, and online structure learning.

2.4.6 Interaction with Large High-Resolution Displays

The problems of interacting with Large High-Resolution Displays (LHRDs) have been investigated in HCI. Shoemaker et al. [176] introduced interaction technique that makes use of a perspective projection applied to a shadow representation of a user. This system was designed to facilitate manipulation over large distances. Bezerianos et al. [24, 23] presented how current software does not support users on managing large amounts of dynamic visual information on large displays and they proposed a set of tasks that are relevant to wall display interaction. Tan et al. [193] investigated how LHRD effect performance in spatial orientation tasks. Czerwinski et al. [47] studied the effects of a larger field of view on user performance. Jota et al. [100] tested four ray pointing variants on a wall display with varying viewing angles. Lehmann et al. [115] introduced bimanual interaction techniques that enable users to manipulate virtual content with the suitable accuracy. Grudin et al. [81] proposed that the increase in available information has increased our requirement to split our digital worlds into different places, so that multiple monitors can be used in this case.

2.4.7 Bezel Effects on Tiled-Display Walls

The effects of interior bezels on tiled displays have been considered previously. Earlier work on desktops using multiple screens has discussed the effect of Bezels

in viewing and work practices [192, 81]. Bi et al. [25] studied the effects of tiled-displays interior bezels on visual search, straight-tunnel steering, and target selection tasks. They showed that interior bezels do not affect visual search time or error rate, but, splitting objects across bezels is detrimental to search accuracy. Also, interior bezels are disturbing to straight-tunnel steering, but not to target selection. Wallace et al. [202] investigated how the presence and width of interior bezels impact visual search performance across tiled displays. They found that the presence or width of interior bezels did not reveal any negative effects on a person's ability to perform the visual search across tiled displays. Another studies [18, 164] show that a large high-resolution display affords a number of advantages and disadvantageous: these bezels improved user performance for task switching or viewing large documents and increased ability to spatially position applications and shortcuts for quick access and recall, on the other hand, bezels distort images and documents confusing users.

2.4.8 Models of Eye Movements and Visual Attention

Models of saliency are used to predict fixation locations. Almost of these models [98, 121] are a bottom-up model where different low-level features such as color, intensity, orientation, texture, and motion are derived from the image at multiple scales. A saliency map then is determined for each of the features and combined together to generate a master saliency map, which indicates the saliency of each pixel. This idea of saliency maps was extended and further developed in other studies. For example, Gao and Vasconcelos [72] proposed a discriminant formulation of center-surround saliency for static images. The outcome of their work is an automatic selection of the important features. This improves the original Itti & Koch model, where feature selection and combination was done in a heuristic way. This was later extended to dynamic scenes and movies using dynamic textures [121]. Torralba et al. [195] used Bayesian approach to combines sensory evidence with prior constraints. Where prior knowledge (e.g., scene context or gist) and sensory information (e.g., target features) are combined according to Bayes' rule. The drawback of this method is that it need a priori assumptions about features that contribute to saliency prediction.

Harel et al. [85] improved the original Itti & Koch model using graphs to compute saliency. This model treats eye movements as a time series. Since there are hidden variables influencing the generation of eye movements, a Hidden Markov Models (HMM) approach was been incorporated. Graphical models enable to model more complicated attention mechanisms. The disadvantages of this approach lie in model complexity, especially when it comes to training and readability. Judd et al. [101] used a data-driven approach to learn classifier based on eye movements data and various visual features as inputs.

Most models of saliency rely only on low-level 2D scene features such as color, orientation, contrast, and intensity. Unfortunately, it is unclear which of the feature channels are most important in generating predictions. Furthermore what other features should be included? Although some studied investigated that [109, 16], this is still not fully answered. Kootstra et al. [109] introduced local symmetry as a measure of saliency. They found that the symmetry models better match the human data than the contrast model. Overall there is still need for further research to determine which features are most relevant in which contexts. Thus, this thesis address this problem by investigating which features are relevant for eye movements prediction in different behavioral contexts (See Chapter 4). Baddeley et al. [16] used a Bayesian system to explore whether high-frequency edges affect human eye movement behaviors. They found that the characteristics of fixated locations were dominated by high-frequency edges. But, simple questions such as "Do humans look more often to high contrast edges due to depth gaps than to edges due to texture borders?" have not been addressed yet. Therefore, this thesis explores how relevant depth features are for eye movement prediction (See Chapter 5).

In order to have a model able to obtain the contribution of different features (such as: contrast, luminance, color, edges, etc.), in various spatial scales, a potentially very large number of parameters requires to be identified [97, 16]. Baddeley et al. [16] found that standard maximum likelihood system identification techniques fail to give good predictions (i.e., the models overfits the dataset, fitting both the signal and noise, and consequently fails to generalize to new dataset). Therefore, we used a linear Support Vector Machine (SVM) [45, 33] approach to finding out the contribution of different features in different contexts. We used models with linear kernels because it is faster to compute and the resulting weights of features are easier to understand. This approach does not need a priori assumptions about features that contribute to saliency. Other methods used previously to constrain high dimensional mappings (such as: principal component analysis, singular value decomposition, and Fourier-based techniques) essentially bias the recognized model to distributed, which is difficult to interpret the solutions even if the problem is simple.

Other interested questions revolve around how top-down cues such as the affect of different tasks and internal states of the observers, influence the computation of visual saliencies? and how humans might select the next gaze location. Previous research has suggested that human eye movement behavior is consistent with decision-making mechanisms for fixation selection that attempt to maximize reward [168, 140]. Also, there is a gap to fill between models performance in real world scenarios and human performance. In this thesis, we investigating how existing predictive gaze models perform in real world scenarios compare to human eye movements behaviors (i.e., in the interaction scenario with tiled Large High-Resolution Displays). Then, the connec-

tion between model and empirical data is made, by using IRL paradigm that constructs the parameters of the learning model to best match the observed human behavior.

Chapter 3

Concept of Predictive User Modeling

In Chapter 2 we introduced key concepts and reviewed the state-of-the-art in related fields. We also reviewed the current work on modeling eye movements. In this Chapter, we present specific approaches to modeling human behavior, but now from a more abstract point of view. The goal of this Chapter is to present conceptually important approaches.

More specifically, we start with considering human behavior prediction as yet another prediction task that could be solved using inductive learning (Sec. 3.2). Then, we emphasize the notion of "normative models" as compared to "descriptive models" (Sec. 3.3). This distinction is common in biology, in particular in neuroscience, and to some extend also in physics. Descriptive models rephrase observations in a more compact form. They may or may not refer to causes and effects. In short: Descriptive models describe the observations, and the models (their structure and/or parameters) are usually fit to data. Normative models can even be formulated without given data. In short: Normative models state how things should be, given some a priori assumptions that may or may not be true. Then, comparing predictions of normative models with data can give insights into **why** humans behave as they do as compared to only treating human behavior prediction as yet another prediction task.

In Secs. 3.4 and 3.5 we introduce Reinforcement Learning (RL) as a paradigm for human behavior prediction. While RL has been used to train adaptive smart meeting rooms, our contribution is to highlight to this community that RL is a valuable paradigm from which human behavior models can be deduced. One innovation in this Chapter is to propose Inverse Reinforcement Learning (IRL) as a promising approach.

3.1 A Concise Overview of the Main Concepts

This thesis addressed the problem of predicting human gaze behavior in smart environments. As a results of the discussion in Sec. 2.4.8, we decide to investigate these problems by:

1. The thesis uses systematic machine-learning approach, where user profiles for eye movements are learned from data in different context, and determining by combinatorial exploration which features are relevant for behavioral context. We used a linear Support Vector Machine (SVM) [45, 33] approach to finding out the contribution of different features in different contexts. This approach does not need a priori assumptions about features that contribute to saliency models. In this approach, the program is fed labeled training data (x_i, y_i), $i = 1, .., l$, $x_i \in \mathbb{R}$, $y_i \in \{1, -1\}$, and tries to learn the unknown model parameters that underlies it (see Sec. 2.2.3.7 for a broader overview of SVM). Figure 3.1 illustrates how to use machine learning approach for gaze location prediction, where user profiles for eye movements are learned from user data.

2. The thesis proposes the modeling of eye movements using normative models. The prediction of these models are based on decision-making theory. Previous research has suggested that human eye movement behavior is consistent with decision-making mechanisms for fixation selection that attempt to maximize reward [168, 140]. Our approach formulated eye movements as a Markov Decision Process (MDP) problem, with the use of Inverse Reinforcement Learning (IRL) to infer the reward function. Figure 3.2 illustrates how to use Inverse Reinforcement Learning for eye movements prediction. Given an exact model of the environment and the measurement of the agent's behavior over time. Instead of predefining the reward function, we seek to identify it from human eye movements behavior.

In the following sections, we discuss the use of machine learning to support user modeling, and we discuss the problems of human behavior modeling in decision making processes.

3.2 Predicting User Behavior with Inductive Learning

Inductive learning methods have been widely used in human behavior modeling. There are a plenty of algorithms exist that can learn the structure from a given data set and some prior information about the originality of the data

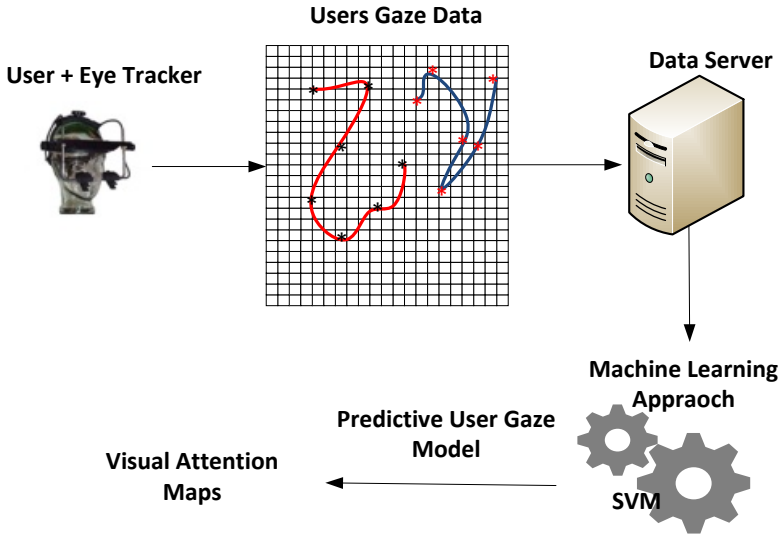


Figure 3.1: Illustration of how to use machine learning approach for gaze location prediction, where user profiles for eye movements are learned from user data in different context.

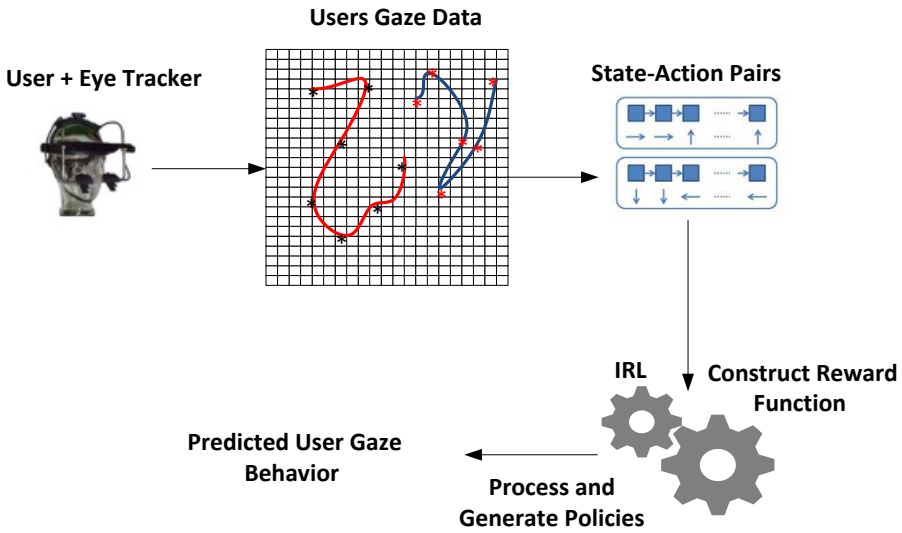


Figure 3.2: Illustration of the use of Inverse Reinforcement Learning for eye movements prediction. Given an exact model of the environment and the measurement of the agent's behavior over time. Instead of predefining the reward function, we seek to identify it from human eye movements behavior.

by minimizing an error criterion. Early machine learning researchers in user modeling focused on user's behavior modeling without seeking to describe the internal processes that produce behavior. Another line of research focusing modeling agent's cognitive processes i.e. models of the internal processes that underlie the user behavior for example [10, 41, 44, 95, 149, 122]. Feature Based Modeling (FBM) [204], an example of user modeling based on feature-value machine learning, has shown that it is possible within reasonable computational constraints to produce models of users' competencies with high predictive accuracy. Some previous works have examined aspects of user modeling based on FBM for example ([173, 26, 9, 110]). Modeling the internal operations of the cognitive system is challenging because the precise mechanisms of how our cognitive system operates is still not well understood. In addition to the inability to observe internal cognitive operations.

In situations where the user repeatedly performs a task that involves selecting among various predefined options appear appropriate for using standard machine learning methods to frame a model of the user. One example of such a task are the human eye movement strategies that have been chosen as the model system for this thesis. In such situations, we consider understanding where people look as straightforward standard classification learning tasks. The visual information available to the users influence them to move their eyes to the most salient locations, so the gaze positions and the stimulus can serve as the training data for a learning algorithm. The algorithm will create a model of a user's eye movements that can then be used to predict the user's behavior on future problems.

In this thesis, we use a learning approach to train a classifier directly from human eye tracking data (see chapters 4 and 5). We use a linear Support Vector Machine (SVM) [45, 33] to find out which features are relevant to different behavior context. We used models with linear kernels because it performed well for our specific task. Linear models are also faster to compute, and the resulting weights of features are easier to understand (see Sec. 2.2.3.7 for more details about the SVM).

Nevertheless, user modeling brings a set of challenges for machine learning applications especially for our considered scenarios where eye movements are known to be a very dynamic modeling task. In the following subsections, we address key concepts and some of the key challenges.

3.2.1 The Need for Large Data Sets

In user behavior modeling situations, it is reasonable that learning algorithms require many training examples to be accurate. To study user eye movement behaviors in smart environments, one needs a set of ground truth data of user fixations in a different context. Many eye movement datasets have been col-

lected using eye tracking experiments. These data sets differ according to the types & numbers of images presented to the users, the task demand was given to the users and also the number of users in each experiment.

In this thesis, we collected a database of eye tracking data in smart environment for three different scenarios:

Eye Movements In Meeting Scenarios: The First scenario is a meeting scenario, where we collected a database of eye tracking data in a meeting room in two scenarios (giving a talk vs. listening). In which three people were involved, each of them makes a presentation. At the beginning the three participants enter the room, one of them goes to the stage to give a talk for four minutes where he was wearing a head-mounted eye tracker and the other go to their respective seat. After the presentation is over, the presenter repeats his talk without wearing the eye tracker, but one of the audience was wearing the head-mounted eye tracker. The same procedure was repeated for the other participants. During these meetings, people had natural behaviors (For more details refer to Chapter 4).

Eye Movements Data On LHRD: The second scenario is an interaction scenario with the tiled Large High-Resolution Displays (LHRDs). We collected a database of eye tracking data for the user when performing a free viewing task with a LHRD and with DLP TV with a 67 inch screen diagonal. Eight users participated in this study. The goal of this study was to find out how well visual saliency algorithms perform with LHRD and to investigate the effects of tiled display (interior horizontal and vertical) bezels on human eye movements and on saliency algorithms predictions (For more details refer to Chapter 6).

3D Eye Movements Data: The last scenario is a free viewing task with 3D natural scenes where we collected a database of eye tracking data on natural scenes where we also have depth information, because all other published database never included depth information. The rational for investigating depth images is that they may reveal the saliency that matters because when interacting with the environment, we evolved by interacting with objects in a three-dimensional (3D) world. This dataset used to improve the computation of saliency maps, by using luminance and depth image features (For more details refer to Chapter 5).

3.2.2 The Need for Labeled Data

A major concern in using machine learning for user modeling tasks is that the supervised machine learning approaches used require explicitly labeled data. Also finding the correct labels may not be easily visible from a simple inspection

of the user's behavior. We consider again the example of eye movement data. Even though the class of salient locations is well defined by the set of fixations (i.e., the positions where saccades land), the selection of non-fixed locations is not easy. While generating random locations from a uniform distribution from regions that were not fixated is not effective. Because, it has been found that users fixate more on the central part of the display rather than in the periphery [194]. This could be due to the photographer's bias of keeping objects at the center of the image. Therefore, it has been proposed that negative examples should also be drawn from this specific distribution of human fixation locations [105]. To avoid this effect, we used the approach proposed in [105], which suggest that the negative samples should be collected from the same locations such as the positives, but with the image data taken from different images. To represent fixations and non-fixed locations accordingly, for each location, we can get a square patch and saved the pixel values in a feature vector x_i together with a label $y_i \in [1, -1]$, indicating fixation or background. From each image, we chose 200 positively labeled pixels randomly from the top 40% salient locations of the human ground truth saliency map and 200 negatively labeled pixels from the bottom 60% salient locations. We noted that expanding the number of examples collected per image more than 200 did not improve the performance of the learned model. In order to have examples that were strongly positive and strongly negative, we collected examples from the top 40% and bottom 60%; we avoided samples on the boundary between the two. In order to have zero mean and unit variance we normalized the features of our training set and used the same normalization parameters to normalize our test data.

3.2.3 Drift Correction

Learning from user behavior in a very dynamic modeling task is not easy, such as characterizing a user eye movement behavior, which is very likely to change over time. Another concern in learning from eye movement data is the accuracy of the gaze position measurements. The exciting eye tracker system produces errors around 0.5 degrees of visual angle. This affects the learning procedure, where even similar regions can seem uncorrelated when misaligned by this amount. Therefore, we took great care to minimize and control measurement errors. We follow the approach proposed in [195, 101]. Eye movements smaller than predefined criteria were considered drift within a fixation.

3.3 Predicting User Behavior with Normative Theories

Normative decision theory is occupied with identifying the best decision to take by selecting between different choices. A normative model of user behavior shall not only *describe* how humans behave and the role cognition plays in accounting for such behavior. Instead, the model shall *explain* why humans behave as they do and the role cognition plays in account for it.

To illustrate this distinction between describing and explaining cognition consider game theory (see Sec. 2.2.2.8): A plain descriptive approach would simply collect a lot of data about how humans decide in scenarios, which can be formalized as a game. Then, a descriptive model (e.g., a statistical regression model) would account for how the properties of a particular game affect the decisions of humans, where accounting for the data is the main goal. Such descriptive models could even have predictive power. However, game theory does *not* start from actual data, but from so-called first principles. It first states desirable properties of the outcome of games, it then predicts the outcome of games, and only afterward it is checked if the predicted outcomes match the data. Interestingly, it turns out that humans often make sub-optimal decisions (see matching vs. maximization in Section 2.2.2.4). It is currently not clear if these sub-optimal decisions may turn out to be optimal when considered from an alternative perspective or within a richer definition of the task the humans are performing as decide.

The existing decision-making and planning frameworks provides the necessary formalisms for representing user observed behavior. There are many developed frameworks for representing decision-making situations with the goal of representing the factors that influence the optimal decision. Almost of these frameworks describe behaviors as a sequence of interactions with a stochastic process that maximize expected utility (see Sec. 2.2.2.5). In the sections that follow, the discussion will be about a few common use of these frameworks.

3.3.1 The Underlying Normative Agent-Environment Architecture

Here we refine from the standard Reinforcement Learning (RL) notation by explicitly distinguishing between the states $\mathcal{S}$ of the environment and the internal states $\mathcal{Z}$ of the agent. A presented in Figure 3.3, At each time an agent A in some external environment E receive an observation and makes an action. The agent also has internal states $\mathcal{Z}$ of the internal environment and the agent selects action $a \in \mathcal{A}$ using it is policy π denoting the conditional probability of selecting action $a \in \mathcal{A}$ if the agent is in the internal state $z \in \mathcal{Z}$. Also, there is a space of reward function R for every agent, which an agent is optimizing.

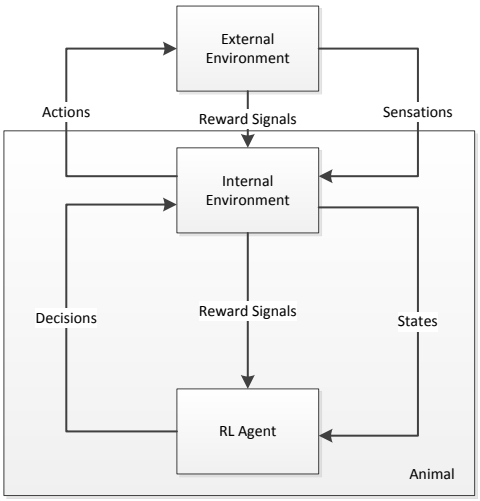


Figure 3.3: Normative Agent- Environment interaction architecture.

3.3.2 The Environment Model for a Single Agent

We use a setting of defining an environment for a single agent that is very similar to the setting used in RL. We consider a set of states $\mathcal{S}$, where the states $s \in \mathcal{S}$ denote *states of the environment*. The environment changes states only in response to an agent's action, and these state transitions are modeled probabilistically.

Definition 4. Let $\mathcal{S}$ be a set of states of the environment and $\mathcal{A}$ be a set of actions. Then, we define a *model of the environment of an agent* as a tupel $(\mathcal{S}, \mathcal{A}, P)$, where $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \to [0, 1]$ is a function with $P(s, a)(s') = P(s' \mid s, a)$ denoting the conditional probability of transitioning to state $s' \in \mathcal{S}$ if the agent has taken action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$.

Note that here we have not introduced any explicit notion of time yet but refer only to state transitions. In other words, a model of the environment for an agent is nothing but a probabilistic state transition system. The randomness expressed by P shall model the randomness in the environment.

3.3.3 The Agent Model

From the perspective of RL an agent is represented by its policy, which is a probabilistic action selection modeled as a conditional probability $P(A = a \mid S = s)$ to select the action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. Here we refine from the standard RL

notation by explicitly distinguishing between the states $\mathcal{S}$ of the environment and the internal states $\mathcal{Z}$ of the agent, but otherwise we define an agent model as usual by referring to a policy π.

Definition 5. Let $\mathcal{Z}$ be a set of internal states of an agent and $\mathcal{A}$ be a set of actions. Then, we define a *model of an agent* as tupel $(\mathcal{Z}, \mathcal{A}, \pi)$, where $\pi : \mathcal{Z} \to \mathcal{A} \to [0, 1]$ is a function with $\pi(z)(a) = P(a \mid z)$ denoting the conditional probability of selecting action $a \in \mathcal{A}$ if the agent is in the internal state $z \in \mathcal{Z}$.

Note that the set of internal states $\mathcal{Z}$ can indeed be completely disjoint from the set of states $\mathcal{S}$ of the environment, $\mathcal{Z} \cap \mathcal{S} = \emptyset$, but this is not required. For example, one is free to define the environment's and the agent's states such that $\mathcal{Z} \subset \mathcal{S}$. The randomness expressed by π shall model the randomness in the action selection of the agent.

Simply considering an agent as a "black box" with a some internal but otherwise not further specified states $\mathcal{Z}$ is too abstract as a model for an agent's cognitive system. Thus, these states are first decomposed into a product

$$\mathcal{Z} = \prod_{i \in C} \mathcal{Z}_i$$

over the sets of states $\mathcal{Z}_i$ of the individual components C of the agents cognitive system. In the following I consider these components individually, namely the *perceptual system* with states $\mathcal{Z}_p$, the attentional system with states $\mathcal{Z}_a$, memory with states $\mathcal{Z}_m$, a subsystem for reasoning with states $\mathcal{Z}_r$, and the decision-making component with states $\mathcal{Z}_d$.

3.3.4 Examples of Environment and Agent Models

3.3.4.1 Change Blindness

Change blindness usually defined as a surprising perceptual phenomenon that happens when the observer of the visual stimulus does not observe a change introduced to the visual environment [80, 184]. The phenomenon of change blindness has greatly contributed to our understanding of visual attention, visual memory, and awareness (for more details review, see [54]). Grimes et al. [80] noted that observers failed to detect large changes introduced to photos during an eye movement, where 50% percent of the observers failed to notice when two cowboys sitting on a bench exchanged heads.

Other studies found that these effects are even stronger when the changes are unexpected. For example, Simons et al. [184] found that most observers do not notice if an actor in a scene is changed during a changing in camera position, even if the actor is replaced by another person.

Also, It has been found that change blindness phenomenon is closely related to visual attention in humans and is traditionally referred to attentional failures [161]. Rensink et al. [161] studied the changes introduced during eye movements. In Rensink's flicker task [161], the original and changed image alternate repeatedly and separated by a blank screen, until observers detect the change. They found that the visual changes in meaningful details of a visual scene called 'central interests' are more easy-to-detect than changes in insignificant details called 'marginal interests'.

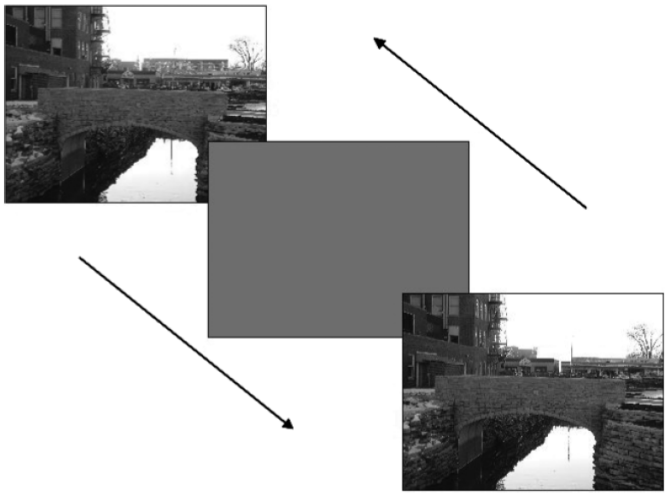


Figure 3.4: Schematic illustration of flicker paradigm used in Rensink's task (from [161]).

3.3.4.2 Probability Matching

To illustrate probability matching, we consider a simple selection task, where players are asked to predict which of two events will take place, given that the two events have different probabilities of occurring. One outcome appears with a higher probability than the other. For example, event $E1$ could occur with a probability of $P(E1) = .75$ while event $E2$ occurs with $p(E2) = 1 - p(E1) = .25$. Assuming that the sequence of events is random, choosing the most probable event E1 is the best strategy in terms of expected payoffs, the average accuracy of 75% . This strategy is called maximizing. However, it has been found that many people match their choice probabilities to the proper outcome probabilities i.e. respond 75% of the time to the highest probability choice. In the example above predicting E1 in 75% of the trials and E2 in 25% of the trials. Because it would yield lower expected payoffs, this phenomenon, called probability matching. There are plenty of approaches tries to explain this choice anomaly (for more details review, see [58, 201, 57]). Almost of these

approaches bind probability matching to cognitive limitations [207]. West et al. [207] argued that the default processing strategy of most participants is a nonnormative cognitive shortcut.

Recently Wolford et al. [211] proposed that probability matching occurs because people do not take into account the randomness of the sequence and attempt to be more successful than the optimal maximizing strategy. Goodnow et al. [76] showed that if the task was framed as gambling instead of problem solving people were more likely to maximize. Wolford et al. [211] found that if the alternation rate was higher than chance people maximized more strongly. Gaissmaier et al. [68] recently argued that probability matching may be "smart strategy", i.e., an adaptive response to the environments where the outcomes potentially follow predictable patterns.

3.3.4.3 Reinforcement Learning(RL)

The problem of behavior learning through trial-and-error interactions with a dynamic environment known as reinforcement learning [102]. The agent learns how to map situations to actions. In RL the agent must find which actions yield the most reward by trying them, instead of telling the agent which action to take. Reinforcement learning is differed than supervised learning where we learn from examples provided by the external supervisor. One major difference is that a reinforcement-learner must be able to learn from its experience and must explicitly explore its environment. Also, there is no presentation of input/ output pairs in Reinforcement learning setting [191, 102]. The agent can select an action and observe the state change and may receive a reward.

The reinforcement learning framework The reinforcement learning is considered to be the problem of learning from interaction to achieve a goal. The agent interact with the environment. On each step of interaction, the agent can select an action, a, and observe the state change of the environment and may receive a reward r. The agent's behavior should select actions that tend to increase the long-run sum of values of the reward signal. It can learn to do this over time by systematic trial and error, guided by a wide variety of algorithms. At each time step, the agent's job is to find a policy π, which is a mapping from state representations to probabilities of selecting each possible action, that maximize the sum of the reward signals it receives over the long run [191]. Figure 3.5 shows the agent-environment interaction in RL.

The reinforcement learning framework is abstract and very flexible, which allows it to be applied to many different problems in different ways [191].

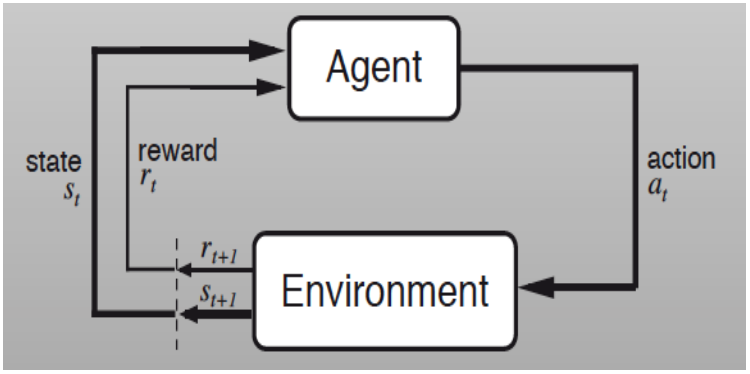


Figure 3.5: The reinforcement learning framework (from [191]).

3.3.5 The Problem Setting of Human Behavior Modeling

On the one hand, humans are surprisingly predictable. For example, models of fluid dynamics are successful in accounting for the dynamics of large crowds, i.e., the models of the individual agents are very simplistic. On the other hand, predicting the behavior of a single human agent even in simple decision-making tasks remains a challenge. We have reviewed that human agents in such tasks often behave in an apparently suboptimal manner: the follow a matching strategy as compared to a maximization, which is clearly optimal once the agent has learned the model of the environment. What is the cause of this intrinsic randomness? Is it imperfection? Or is there a rational explanation for behaving in that apparently suboptimal manner?

It has recently been postulated that this randomness is due to the fact that in real-world scenarios humans not only learn the parameterization of models of the environment, but they also learn the structure and causal dependencies of such model [6]. Acuna et al. [6] formulated the problem of structure learning in sequential decision tasks using Bayesian Reinforcement Learning (BRL) and demonstrate qualitative differences in the behavior of optimal learning agents between parameter and structure learning. As the first step in the normative modeling of users, which take into account how agents deal with uncertainty. In [6] the task was designed, where users shall infer the *structure* of a model environment in a sequential decision-making task. The model was so simple that a closed-form solution for iterative BRL could be derived because the prior distributions were conjugated priors.

The proper way of including such uncertainties into RL is Bayesian RL (BRL). Baker et al. [17] argue that action understanding is much like visual

perception: While action understanding is a kind of "inverse planning", or Inverse Reinforcements Learning (IRL), and vision is a kind of "inverse graphics". They propose a framework based on Bayesian inverse planning for modeling human action understanding. The underlying assumption here is that agents are rational, and they deal with uncertainties in the optimal way, which is the Bayesian approach.

However, another important factor not considered in many other cognitive models is that agents usually have a limited access to the state of the world. For humans, many cognitive resources are spent on organizing the sensory signals into a meaningful interpretation upon which decision making is then based. Attention is probably the most important such organizing principle. Thus, for building more realistic models of higher cognitive function, the complexity of the perceptual apparatus needs to be taken into account.

3.4 Predicting User Behaviour with Reinforcement Learning

In Sec. 3.3.4.3 we have referred to RL as an example of the agent and environment model, but we have not employed a key concept of RL, namely rewards. They were introduced as part of the examples (see Sec. 3.3.4.3). This shows that the proposed model has been generic enough to allow for introducing concepts such as rewards as part of the state definitions. One important aspect to consider in models of decision making and planing is how agents learn to decide. Reinforcement learning (RL) is a natural framework for that. The reinforcement learning framework is a considerable abstraction of the problem of goal-directed learning from the interaction. Moreover, in the setting of the normative model for a multi-agent world, we explicitly use the notion of a utility function, and the normativity of the model rests on this utility function.

In this section, we return to Reinforcement-Learning (RL) within the setting of a single agent interacting with its environment. We present key concepts from RL, namely the state value function for human behavior modeling in a smart environment.

3.4.1 The Problem Setting of Reinforcement Learning

An agent is interacted with its environment via perception and action in the standard reinforcement-learning model, as illustrated in Figure 3.5. More specifically, at each time step, $t = 0, 1, 2, 3, ..$ the agent receives indication of the current state,s, of the environment, $s_t \in S$, Where S is the set of possible states, the agent then chooses an action $a_t \in A(s_t)$ where $A(s_t)$ is the set of

action available in the state s_t. As consequence of the agent action, it receives a numerical reward, $r_{t+1} \in \mathbb{R}$, and find itself in a new state S_{t+1}.

Beyond the agent and the environment, a reinforcement learning system consist of the following sub-elements: a policy, a value function, a reward function, and, optionally, a model of the environment [191].

Reinforcement learning models with fully observable state usually concerns solving tasks formulated as Markov Decision Processes (MDPs) problems with delayed reinforcement are well modeled as Markov decision processes (See Sec.2.2.2.6 for more details about MDPs).

3.4.2 Value-Function based RL and Policy Search Methods

Reinforcement learning algorithms can be differentiated in one of two classes: (1) Policy search methods: those that learn a controller without learning a model [209]. (2) Value-function based methods: Learn a model and use it to derive a controller.

Algorithms in this first class are called policy search algorithms, where it learn policies directly without modeling a value function. Direct policy search methods learn parameters for a policy, a way of acting in a particular task. Agents using policy search methods, first explore policy space by adjusting parameters of the policy and then evaluating candidate policies by the performance of one or more trajectories resulting from the policy. So that, using policy search algorithms require an objective function that evaluates whole paths, such objective function not need to be built from a reward function, that provides feedback for each step that can be added to evaluate a trajectory.

On the other hand, value-based methods learn to estimate a value function for each situation that the agent could find itself in. The agent is then able to take the action that it believes will give it the most value in the long run. Over time, the learned value function estimated the true value of each state. Therefore, the agent will build an accurate measure of how valuable each action.

Both value-based and policy search methods can find the optimal policy. However, only value-based methods can provide an estimate of the goodness of choosing a given action from a given state. In this thesis we are interested more on value-based RL method, so in the following we consider this method only.

Formally, for MDPs the value of state s when following policy π, denoted $V^\pi(s)$ can define as

$$V^\pi(s) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right\} \tag{3.1}$$

where $E_\pi\{\}$ indicates the expected value under a policy π at any t time step.

Similarly, we define $Q^\pi(s,a)$ which is the value of taking action a in state s under a policy π as:

$$Q^\pi(s,a) = E_\pi\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\} \tag{3.2}$$

The optimal value of a state, which is the the expected discounted sum of rewards that the agent will gain if it start in that state and execute the optimal policy is denoted by: $V^*(s) = \max_a\{R(s,a) + \gamma\sum \Pr(s\prime \mid s,\pi(s))V^*(s')\}, \forall s \in S$, the optimal policy as: $\pi^*(s) = \arg\max_a\{R(s,a) + \gamma\sum_{s'}\Pr(s' \mid s,a)V^*(s')\}$.

3.4.3 Using the State Value Function to Guide Assistance

Some RL methods are interested in estimating value functions of states or of state-action pairs. Such methods, for example, Temporal Difference (TD) algorithms, use value function to assign a value to each state that is an estimate of the amount of reward expected over the future after that state is visited. Thus, the state value function determines how good it is for the agent to be in a certain state or how good to execute a particular action in a specific state. In other words, the state value function defines how valuable for the agent to be in a certain state.

In Barto et. al. [19] the RL system that use the value functions as "adaptive critic", which refer to the component that estimates values for evaluating ongoing behavior.

Moreover, it is possible to generalize state value function learned for some set of environments to a new but similar environment. The Idea of using the learned state value function in a new environment has been investigated for a long time in Markov Decision Process (MDP) and reinforcement learning [191].

The problem here is how to translate the learned behavior obtained for one domain to another. Many researchers investigated how to supply a learned behavior with some initial simple behavior [144, 183, 174, 55, 13]. Selfridge et al. [174] showed that it is possible to speed up the learning behavior on a task if the learner has first learned on a simpler variation of the task. They argued that the adaption to the new task would be faster using a policy trained on a related task compare to learning from scratch. Russell et al. [13] explored the process of eliminating features to reduce the effective state space in hierarchical reinforcement learning. They found that state value function learned in subroutines had been successfully transferred in a hierarchical reinforcement learning framework. Guestrin et al. [83] employed linear programming to construct value functions for classes of similar agents. Based on the presumption that transition function and reward are similar with all agents of a class. Then

such class-based value functions are placed into agents in a new environment which have a different number of objects. They found that, Although no learning is conducted in the new environment, the previously learned value functions performed better than a baseline hand-coded strategy.

Reward shaping [55, 123, 144] allows to modify a learning behavior by adding in artificial rewards to the environmental rewards. In this setting, to ensure that unintended behaviors are not established, it requires a priori appropriate knowledge about the environment to guide the learner. Ng et al. [144] argued that, While it is well understood how to add this type of guidance to a learner, it would prefer to allow the agent to learn faster by training on different tasks.

In RL setting, rewards come from the agent's environment. Recently, Ng et al. [145] argued that the reward function from RL must be considered as an unknown when examining the animal and human behavior. This reward function, which cannot be observed directly, can be considered as part of the internal state of a user, similar to the state of the attentional system, or the current goal state. In this thesis, we proposed to use value function based on learned reward function in an application with the smart environments. The state value function can be used so that the application, on for example the powerwall, can be adapted based on this state-value function, the policy of the user, and the potential actions that can be taken.

3.5 Predicting User Behaviour with Inverse Reinforcement Learning Modeling

In user behavior modeling, the task often is less well defined and the goal is to learn a policy, which determines which actions to take in specific states to achieve a goal. In situations, where no direct data set exist which could be used to learn a policy in a supervised way. The selection of actions can also depend on the decisions and actions of others, especially when the possibility of communication with other agents are available. Thus, using a normative framework based IRL for modeling sequential decision-making task may be helpful in these situations. Further, this paradigm is useful for learning how to process a new task based on already learned tasks; where the transferred behavior could be used to finish a new task more quickly.

In this section, we present key concepts of using Inverse Reinforcement Learning (IRL) in modeling and explaining user observed behavior.

3.5.1 Learning from Demonstrated Behavior

Learning from demonstration is a powerful method of earning skill in humans [171, 141]. Many Researchers have pursued to develop computational methods

to learn by observing humans performing a task (See [171] for a survey of this work). Precocious control approaches used to solve the problem of imitation learning by modeling the near-optimal policy for future execution in similar situations. But, It is known that it is more demanding for predicting long-term decision-making behavior, but this approaches satisfy for predicting short-term stimulus-response behavior. More recent approaches, seek to learn an underlying reward function to explain human behavior rather than directly learning the policy. The problem of finding the underlying reward function of the demonstrator from its behavior is known as the Inverse Optimal Control problem or Inverse reinforcement learning(IRL) [145], which was originally formulated by Kalman [103].

3.5.2 The Problem Setting of Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) [145] describes the problem of recovering an agent's reward function from demonstrated behavior. In an IRL setting, we assume the setting of the Markov Decision Process (MDP) (Section 2.2.2.5), the algorithm is presented with M/R, together with expert demonstrations $D = \{\zeta_1, ..., \zeta_N\}$, where $\zeta_i = \{(s_{i,0}, a_{i,0}), ..., (s_{i,T}, a_{i,T})\}$ (i.e., its trajectory or path, ζ, of states si and actions ai). In combination with features of the form $f : S \longrightarrow \mathbb{R}$ that can be used to represent the unknown reward R. Ng & Russell [145] formulate IRL as the reconstruct of reward weights, θ, which make the demonstrated behavior optimal.

Definition 6. Inverse Reinforcement Learning (IRL) problem is defined in [145] as follows:

> **Given 1)** Measurement of the agent's behavior over time, in a variety of circumstances **2)** Measurements of the sensory information inputs to the agent; **3)** an exact model of the environment.
>
> **Determine** the reward function that an agent is optimizing.

Remark 1. Ng & Russell [145] There are many solutions of R including R=0, That may make the demonstrated behavior optimal.

Toward optimal reward function, one can turn to evolutionary optimization [182] to generate reward functions if there is an effective way to evaluate the appropriateness of the resulting behavior. One advantage of the evolutionary approach is that, among the many possible reward functions that generate good behavior, it can identify ones that provide helpful but not too distracting hints that can speed up the learning process.

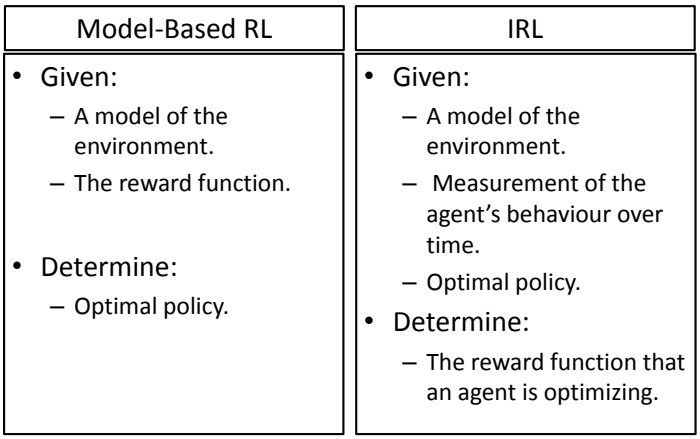


Figure 3.6: Comparison between RL and IRL.

3.5.3 Learning the Reward Function from Demonstrated Behavior

Reconstructing the agent's reward functions is an ill- posed problem. Given an exact model of the environment and the measurement of the agent's behavior over time, the goal of IRL is to determine a reward function that can justify the demonstrated behavior. Ng & Russell [145] argue that the reward function from RL must be considered as an unknown when examining animal and human behavior; They presents methods to solve the problem of the inverse reinforcement learning (IRL). Abbeel & Ng et al. [4] propose a strategy of matching feature expectations between an observed policy and a learner's behavior; they show that this matching is essential to achieving the same performance as the agent if the agent were solving an MDP with a reward function linear in those features. Other researchers provide further development to ameliorate the original algorithms suggested by Ng et al. [145] and [4]. For example, Ramachandran and Amir [157] explains how to combine prior knowledge and evidence from the expert's actions to infer a probability distribution over the space of reward functions. Ziebart et al. [219] developed a probabilistic approach based on the principle of maximum entropy. Levine et al. [117] used Gaussian processes (GPs) to learn the reward as a nonlinear function while determining the relevance of each feature to the expert's policy.

3.5.4 Feature Matching Optimal Policy Mixtures

Abbeel & Ng [4] introduce a novel approach based on Inverse Reinforcement Learning (IRL) [145]. They propose a strategy of matching feature expectations between an observed policy and a learner's behavior; they show that this matching is essential to achieving the same performance as the agent if the agent were solving an MDP with a reward function linear in those features. They represent the reward function by a linear combination of m feature functions fi with weights θi, which maps the features of each state, $f_{s_j} \in \mathbb{R}^m$, to a state reward value. Hence, the reward function is defined by:

$$reward(s,a) = \sum_{i=1}^{m} \theta_i^\top f_i(s,a) = \theta^\top f(s,a),$$

where $\theta \in \mathbb{R}^m$ and $f(s,a) \in \mathbb{R}^m$. The features functions f_i are bounded and mapped from $S \times A$ into R.

For a given trajectory $\zeta_i = \{(s_{i,0}, a_{i,0}), ..., (s_{i,T}, a_{i,T})\}$ the feature counts are given by $\tilde{f}_i^\zeta = \sum_{t=1}^{H} \gamma^t f_i(s_t, a_t)$. The feature count f_i^π when following policy π can be defined by

$$f_i^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t f_i(s_t, a_t)^t \mid \pi, \zeta\right]$$

While the reward function is given by a linear combination of features f_i, the expected value function of a policy π can be defined as

$V_\theta^\pi(s) = \sum_{i=1}^{m} \theta_i f_i^\pi(s) = \theta^\top f^\pi(s)$ where $f^\pi \in \mathbb{R}^{\ m}$ and holding entries of the single feature counts $f_i^\pi(s)$.

The reward value along a trajectory represented by the sum of the state features along the path. Therefore, the agent observes single paths, and has an expected feature count, $\tilde{f} = \frac{1}{m}\sum_i f_i$, based on many (m) demonstrated trajectories.

Abbeel & Ng [4] showed that this representation is enough to achieve the same performance as the agent were solving an MDP with a reward function linear in those features (Equation 3.3).

$$\sum_{path\zeta i} P(\zeta i) f_{\zeta i} = \tilde{f} \tag{3.3}$$

3.5.5 Maximum Entropy Inverse Reinforcement Learning Method (Max Entropy IRL)

The Maximum Entropy Inverse Reinforcement method [219] reduces learning to the problem of recovering a reward function; that makes the behavior influenced by a near-optimal policy that closely imitate demonstrated behavior.

It is a probabilistic approach based on the principle of maximum entropy. A maximum entropy IRL formulation finds a distribution P over all trajectories. With respect to the eye movements behavior, it models the distribution over all possible eye movement paths of length T starting from state s for a given image as:

$$P(\zeta_i \mid \theta) = \frac{1}{Z(\theta)} \exp(\theta^\top f_{\zeta_i}) = \frac{1}{Z(\theta)} \exp \sum_{s_t \in \zeta_i} \theta_{a_t}^\top f(s_t), \forall \zeta \in D \tag{3.4}$$

where $D = \{\zeta_1, ..., \zeta_N\}$, $\zeta_i = \{(s_{i,0}, a_{i,0}), ..., (s_{i,T}, a_{i,T})\}$
and $r_\theta(s_t, a_t) = \theta_{a_t}^\top f(s_t)$, so

$$P(\zeta_i \mid \theta) = \frac{1}{Z(\theta)} \exp \sum_{s_t \in \zeta_i} r_\theta(s_t, a_t), \forall \zeta \in D \tag{3.5}$$

where $r\theta(st, at)$ is the reward function, θ are the model parameters and $Z(\theta)$ is the partition function, for paths of length T starting with state s. The reward function $r_\theta(s_t, a_t) = \theta_{a_t}^\top f(s_t)$ is the product between a feature vector $f(s_t)$ extracted at image location s_t and a vector of weights corresponding to action a_t.

Maximum Entropy IRL finds the weights θ that maximize the likelihood of the demonstrated trajectories under the maximum entropy distribution.

$$\theta^* = \arg\max_\theta L(\theta) = \arg\max_\theta \sum_{examples} \log P(\zeta \mid \theta, T) \tag{3.6}$$

This maximization problem can be solved using gradient-based optimization methods, and it is convex for deterministic MDPs. Expected state visitation frequencies, D_{si}, can be expressed by the gradient, which is the difference between expected empirical feature counts and the learner's expected feature counts.

$$\nabla L(\theta) = \tilde{f} - \sum_\zeta P(\zeta \mid \theta, T) f_\zeta = \tilde{f} - \sum_{si} D_{si} f_{si} \tag{3.7}$$

To deal with the exponential growth of paths with the MDP's time horizon. MaxEntropy IRL algorithm used a technique similar to the forward-backward algorithm for Conditional Random Fields to compute the expected state Frequency (See [219] for more details).

Efficient State Frequency Calculations

For optimization, the gradient can easily computed (Equation 3.7), given the expected state frequencies. The expected state frequencies can be computed using straight-forward approach based on enumerating each possible path.

[0] **Backward pass**
1. Set $Z_{Steminal} = 1$
2. Recursively compute for N iterations
$Z_{a_{i,j}} = \sum_k P\left(s_k \mid s_i, a_{i,i}\right) e^{reward(s_i|\theta)} Z_{sk}$
$Z_{s_i} = \sum_{a_{i,j}} Z_{a_{i,j}} + 1 s_i = s_{terminal}$

Local action probability computation

3. $P(a_{i,j} \mid s_i) = \frac{Z_{a_{i,j}}}{Z_{s_i}}$

Forward pass

4. Set $D_{s_{i,t}} = P(s_i = s_{initial})$

5. Recursively compute for $t = 1$ to N

$D_{s_{k,t+1}} = \sum_{s_i} \sum_{a_{i,j}} D_{s_i,t} P(a_{i,j} \mid s_i) P(s_k \mid a_{i,j}, s_i)$

6. $D_{s_i} = \sum_t D_{s_{i,t}}$

3.5.6 The Feature Construction for IRL (FIRL)

The feature construction for Inverse Reinforcement Learning method [116] constructs reward features from a large collection of component features, by building logical conjunctions of those component features that are relevant to the example policy. The algorithm repetitively builds both the features and the reward function. Each iteration consists of two step formulation:

An optimization step computes a reward function $R^{(i)}$ of the i^{th} iteration using the current set of features $\Phi^{(i-1)}$ beginning with an empty feature set $\Phi^{(0)}$, and a fitting step determines a new set of features $\Phi^{(i)}$.

The objective of the FIRL optimization step is to identify areas where the current features are not enough, and must be able to step outside of the constraints of these features, and learn a reward function $R^{(i)}$ that best fits the last feature hypothesis $\Phi^{(i-1)}$. The reward function $R^{(i)}$ is constructed using a constrained quadratic programming solver, using constraints that maintain $R^{(i)}$ stable with D.

The fitting step generates a new feature hypothesis $\Phi^{(i)}$ by inspecting the reward function $R^{(i)}$ to better captures the variation in the reward function. See [116] for more detail.

Optimization Step

For i^{th} optimization step, FIRL compute a reward function $R^{(i)}$ using the examples D and the current feature set $\Phi^{(i-1)}$. If the optimal policy under the reward is consistent with the examples D then the reward function is chosen, and so

that it minimizes the sum of squared errors between $R^{(i)}$ and its projection onto the linear basis of features $\Phi^{(i-1)}$.

Formally, let $T_{R\to\Phi}$ be a $\mid \Phi^{(i-1)} \mid$ by $\mid S \mid$ matrix for which $T_{R\to\Phi}(\tilde{\mathrm{A}}, s) = \mid \tilde{\mathrm{A}} \mid^{-1}$ if $s \in \tilde{\mathrm{A}}$, and 0 otherwise, and let let $T_{\Phi\to R}$ be a $\mid S \mid$ by $\mid \Phi^{(i-1)} \mid$ matrix for which $T_{R\to\Phi}(s, \tilde{\mathrm{A}}) = 1$ if $s \in \tilde{\mathrm{A}}$, and 0 otherwise. So that, $T_{\Phi\to R}T_{R\to\Phi}R$ is a vector where the reward in each state is the average over all rewards in the feature that state belongs to. Letting π^R denote the optimal policy under$R$, the reward optimization problem can be expressed as:

$$\min_R \|T_{\Phi\to R}T_{R\to\Phi}R\|^2$$

.

$$s.t. \qquad \pi^R(s) = a \quad \forall(s,a) \in D \tag{3.8}$$

the constraint (3.8) is not convex, which make it difficult to solve this optimization problem. We can equivalently express in terms of the value function corresponding to R as:

$$V(s) = R(S,a) + \gamma \sum_{s'} P_{sas'}V(s') \qquad \forall(s,a) \in D$$

$$V(s) = \max_a R(S,a) + \gamma \sum P_{sas'}V(s') \qquad \forall s \in S \tag{3.9}$$

These constraints are also not convex, by using a pseudo value function that bounds the value function, we can construct a convex relaxation. By replacing (3.9) with the linear constraint:

$$V(s) \geq R(S,a) + \gamma \sum_{s'} P_{sas'}V(s') \qquad \forall s \notin D$$

where $P_{sas'}$ is the transition probabilities.

All of the constraints in the final optimization are sparse. While both $T_{R\to\Phi}$ and $T_{\Phi\to R}$ are sparse, and contain $\mid S \mid\mid A \mid$ non-zero entries. By introducing a new set of variables R defined as $R = T_{R\to\Phi}R$, FIRL make the optimization fully sparse, yielding the sparse objective $\|T_{\Phi\to R}T_{R\to\Phi}R\|^2$. To that end, it construct a sparse matrix N, where each row$k$ of N corresponds to a pair of features $\tilde{\mathrm{A}}_{k1}$ and $\tilde{\mathrm{A}}_{k2}$ (for a total of K rows).

By normalizing the two objectives by the number of entries, we get the following sparse quadratic program:

$$\min_{R,R_\Phi,V} \frac{1}{\mid S \mid\mid A \mid}\|T_{\Phi\to R}T_{R\to\Phi}R\|_2^2 + \frac{w_N}{K} \parallel NR_\Phi \parallel_1$$

$$s.t. \quad R_\Phi = T_{R\to\Phi}R$$

$$V(s) = R(S,a) + \gamma \sum_{s'} P_{sas'} V(s') \qquad \forall (s,a) \in D$$

$$V(s) \geq R(S,a) + \gamma \sum_{s'} P_{sas'} V(s') \qquad \forall s \notin D$$

$$V(s) \geq R(S,a) + \gamma \sum_{s'} P_{sas'} V(s') + \epsilon \qquad \forall s \notin D, (s,a) \in D$$

where in the implementation, this weight w_N was set to 10^{-5}.

This program can be solved efficiently with any quadratic programming solver (See [116] for more detail).

Fitting Step

Once the reward function$R^{(i)}$ for the current feature set $\Phi^{(i-1)}$ is computed, FIRL formulate a new feature hypothesis $\Phi^{(i)}$ that is better able to represent this reward function. The goal of the fitting step is to build a set of features that gives greater resolution in regions where the old features are not good enough, and lower resolution in regions where the old features are unnecessarily fine. FIRL obtain $\Phi^{(i)}$ by constructing a regression tree for $R^{(i)}$ over the state-space S, using the standard intra-cluster variance splitting criterion (See [116] for more detail).

3.5.7 Using the Reward Function to Predict Behaviour

IRL approaches interested in recovering a reward function, which can explain observed behavior via the corresponding optimal policy. Ng et al. [144, 145] argued that if the reward function for the target behavior is recognized, the space of behavior preserving transformations to this reward function is well understood. Inverse reinforcement learning applied to different problems such as modeling goal-directed trajectories of pedestrians [220], helicopter control [2], robot navigation across different environments [108], parking lot navigation [3], routing preferences of drivers [219], learning strategies in table tennis [138] and user simulation in spoken dialog management systems [39].

Once the reward function is constructed, the IRL method (e.g., the feature construction method [4]) learn a mixed policy, or an ensemble of policies with a certain probability whose feature expectation, on average, mimics the expert behavior. For example, the smart room observes an agent acting in the environment and then determines the reward function. The learned function will then be used to predict behavior. Such behavior form can be transferred to unknown environments by changing the environment. Also, once several behavior styles are learned, one can create a variety of styles that combines between them.

In this thesis we focus on eye movements, but the idea of learning a reward function for an agent could also lead to better user models in general: Since the state value function of classical RL and BRL is defined in terms of the reward function (and a model of the environment, if available), a smart room could annotate various states with the value of the corresponding value function of a user. This is valuable information for designing proactive smart rooms.

Chapter 4

Gaze Locations Prediction Based on Context

For the empirical studies in this thesis we aimed at a balance between well-controlled experimental conditions and more natural settings, where humans behave as in everyday situations. Our first study, presented in this Chapter, is well-controlled in the sense that we focus on a specific and repeatable scenario: giving and listening to a presentation. However, we allow the subjects in this experiment to move freely. We equipped the subjects with a mobile eye tracker, recorded their eye movements and then analyzed them with respect to the presence of visual features at the center of gaze. This first study already exemplifies our general approach to empirical studies: While more classical psychology or neuroscience studies would have characterized in great detail the features at the gaze locations with descriptive statistics, we directly employ predictive models as the prime analysis method and deduce insights from their prediction performance.

The main result of this first study is that eye movement prediction depends on the context (giving a presentation vs listening to a presentation). This may not come as a surprise, but it already shows simple predictive "one-fits-all"-models will not work for eye movements prediction. Thus, even though eye movements appear to be simple compared to the full repertoire of human behavior, they are still a major challenge for predictive models.

This chapter is organized as follows: First, we describe the material and methods including the eye tracking experiment (Sec. 4.2) and the features we extracted from our dataset (Sec. 4.2.5). Then, we present the results of our analysis, we first compare the predictions for the individual features in both scenarios (giving a talk and listening) and then we present the results from features combined (Sec. 4.3).

The results of this chapter have previously appeared as conference publication [135].

4.1 Introduction

In human social interaction, meetings are important life activities. It is the place where a group of people comes together, share information, engage in discussions, and make decisions. Understanding human behaviors that are presented during meetings are important. Among these behaviors, gaze represents one of the important cues. Taking into account the social aspects of meetings requires the analysis of different nonverbal communication cues, for example, recognizing meeting activities cues [127] or recognition of roles in meetings [62]. There have been many improvements in technology-oriented tools to make meetings more efficient. For example, browsing elements of interest within a recorded meeting [206], or the usage of tools that allowed parallel access to shared objects [61] or create abstractive summaries [106].

Models of saliency are used to predict gaze locations. Most models of saliency [98, 121] are biologically inspired and based on a bottom-up computational model. These models does *not* take into account contextual factors or the goal of a user in a visual task. These computational models are all based on low-level image features. Although the saliency-based models were quite successful in the sense of predicting saliency maps, the models have limited use, as they frequently do not match actual human saccades from eye-tracking data [101]. It was noted that combining all features produces the best eye fixation predictions [101]. However, it is known for a long time that task-demands affect the patterns of eye movements [214], but neither saliency models nor the data-driven approach takes that into account.

Figure 4.1 illustrates how saliency maps can be used within a so-called smart lab (Figure 4.1a): Various sensors may extract the gaze direction of users in a room. But even if the gaze direction is similar, as for user A and B (Figure 4.1b), the saliency maps can still differ due to different task demands. For example, while user A aims at following the presentation, user B's task might be to spot spelling mistakes in slides, which shall make different visual field locations salient. Moreover, saliency maps yield richer information than just gaze direction, because they label the whole visual field of a user. This is valuable information for estimating the internal states of users such as in, for example, intention recognition, to adapt visual interfaces, or to place important information.

In this work, we present the results of experimental study to improve the prediction of saliency maps in smart meeting rooms. More specifically, we investigate meeting scenarios in terms of their context-dependence saliency based on different image features. We used data-driven approach to derive models that describe the features that play a role in these scenarios. We found that the prediction differs according to the type of features we selected. Most interestingly, we found that models trained on the face features perform better than models

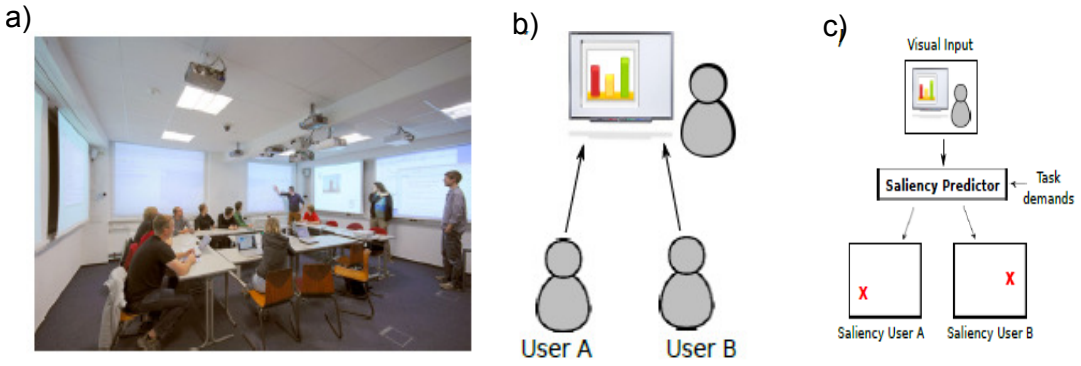


Figure 4.1: Application scenario for using saliency maps in smart environments. **a)** A typical scenario in the smart meeting room. **b)** First abstraction with a speaker in front of a presentation screen and two users looking at that screen. **c)** Illustration of how to use a saliency predictor, which computes a saliency map, in such a setting: Users A and B have approximately the same visual input, but depending on their task demands, different locations in their visual field are rendered as most salient (red crosses).

trained on other features in the ”giving a talk -speaker- ” scenario, but in the listening scenario the models trained on competing saliency features from Itti and Koch perform better than models trained on other features. The investigation of context in analyzing group interactions is a prominent approach, as the same nonverbal behavior can have a different interpretation depending on the context. For eye movement prediction, the task context (what are the people or the group doing) affect the gaze of people. The knowledge of these contexts can improve the gaze prediction. However, here we investigate which features are important in each circumstance in meeting scenarios (giving a talk and listening) in term of predicting eye movements. Thus, we hypothesize that saliency maps respecting this will ultimately outperform saliency maps computed only on the basis of 2D pixel images.

4.2 Material and Methods

We collected eye movements data in two scenarios (giving a talk and listening). One independent variable in the experiments is the task context. We hypothesize that, based on the behavior context, different visual features will make different contributions into gaze location predictions. Thus, predictive models respecting this will ultimately outperform "one-fits-all"- saliency map models.

4.2.1 Measuring Gaze Locations

We used an iView X HED 4 Eye Tracking System (SMI) to record eye position. The system reports gaze positions with a sampling rate of 50 Hz and a reported accuracy of 0.5°-1°. We used the default lens ($f = 3.6$ mm) for the scene camera which provides a viewing angles of 31° horizontally and 22° vertically. The eye tracker' scene camera has a resolution 752 × 480 pixels.

4.2.2 Visual Stimulus

We designed ten slides presentation (which contain: text, charts, graphs, images, equations, etc.). The presentation duration was for four minutes. We employed Microsoft PowerPoint to present the slides on a projector display.

4.2.3 Participants

Three participants took part in this study (Three males, 24-40 years). The participants were with normal vision and no history of neurological problems. All of them were researchers in the institute of computer science.

4.2.4 Eye Tracking Experiment

We collected a database of eye tracking data in a meeting room in two scenarios (giving a talk vs. listening) in which three people were involved, each of them supposed to make a presentation. At the beginning of the experiment, one of the participant goes to the stage to give a talk for four minutes where he was wearing a head-mounted eye tracker and the other go to their respective seat. After the presentation is over, the presenter repeats his talk without wearing the eye tracker but one of the audience was wearing the head-mounted eye tracker. The same procedure was repeated for the other participants. During these meetings, people had natural behaviors. We generate a saliency map of the locations fixated by the viewer in each frame. Also, we convolve a Gaussian low pass filter, with circular boundary conditions with parameters values similar to [199, 195], across the user's fixation locations in order to obtain a continuous saliency map of an image from the eye tracking data of a user. Figure 4.2 shows examples from the data collection in the giving a talk vs. listening scenarios recorded with our setup.

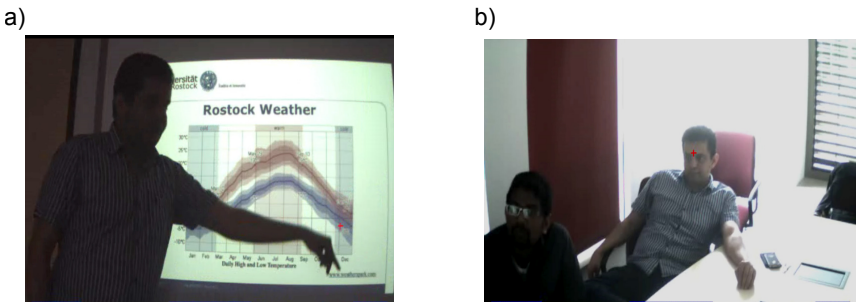


Figure 4.2: Examples from the data collection in different scenarios (giving a talk vs. listening) recorded with our setup. **a)** Frame from the scene camera of the eye tracker and the corresponding gaze point (red cross) of an audience in giving a talk scenario. **b)** Frame from the scene camera of the eye tracker and the corresponding gaze point (red cross) of the speaker in the listening scenario.

4.2.5 Features of Luminance Image

For each image frame in the dataset, we compute a number of low-, mid- and high- level features (See Figure 4.3 for an example) for every pixel within the image and used them as input to the SVM algorithms similar to [101]. We used the local energy of the steerable pyramid subbands (S-Features) in four orientations and three scales [179]. We also include features used in the Torralba saliency model (T-Saliency) [195]. In addition to, the intensity, orientation and color contrast channels as calculated by Itti and Koch saliency method [98]. Also, we used a horizon line detector from mid-level gist features, because most objects rest on the surface, besides a feature that indicates the distance to the center for each pixel. Furthermore, we used the Viola Jones face detector [200] to detect face features in the image frames and include it as an example of high level features.

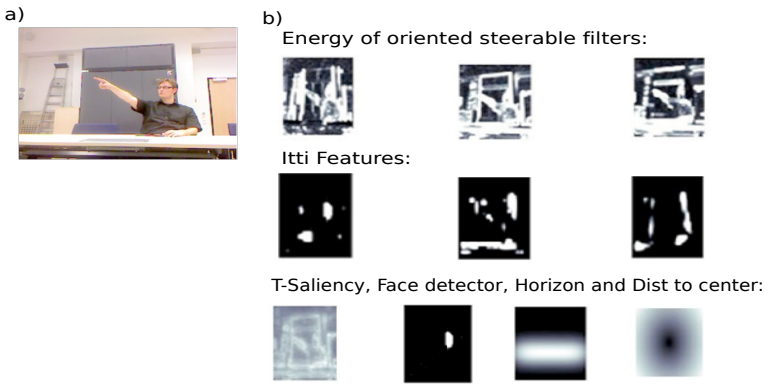


Figure 4.3: Features. **a)** A sample image (top left) and **b)** different low-, mid- and high-level features we used in our analysis.

4.2.6 Classifiers for Predicting Gaze Locations

Opposed to previous biologically inspired bottom-up computational visual saliency model, we use a machine learning approach to train a classifier from human eye tracking data. We use a linear Support Vector Machine (SVM) to find out which features are informative. We used models with linear kernels because it performed well for our specific task. Linear models are also faster to compute, and the resulting weights of features are easier to understand (see Sec. 2.2.3.7 for more details about the SVM). We split our dataset into training images and testing images in order to train and test our model.

In order to have zero mean and unit variance we normalized the features of our training set and used the same normalization parameters to normalize our test samples. We predict the saliency per pixel using a particularly trained model, for each image in our dataset. The continuous saliency map, which represents how each pixel is salient, represented by the values of $w^T x + b$ (here w and b represent the learned parameters and x is the feature vector). Then the saliency map was thresholded at 40% percent of the image for binary saliency maps.

4.2.7 Error Measure

We used the Kullback–Leibler (KL) divergence to measure the distance between distributions of saliency values at human vs. random eye positions (see Sec. 2.3.4 for more details). Models show higher KL divergence, are better in predicting human fixations, because usually human gaze towards the regions with the highest model responses and avoiding the low model responses regions.

4.3 Results : Gaze Location Prediction in Meeting Scenarios (Giving a Talk vs. Listening)

We measured the performance of saliency models using KL divergence (see Section 4.2.7). The results of the performance of different features models averaged over all testing frames are shown in Figure 4.4. For each frame, we predict the saliency per pixel using a specific trained model. We can see that the prediction differ according to the type of features we selected in both scenarios (giving a talk and listening scenarios) (see Figure 4.4). Furthermore, the context dependence shows up: In the listening scenario, models trained on competing saliency features from Itti and Koch perform better than the models trained on other features (see Figure 4.4 red bars). This is not a surprise but expected, because the presentation slides may contain many colored figures, images or text that are more relevant for the audience. In the giving a talk scenario, models trained on the face features perform better than the models

trained on other single features (see Figure 4.4 blue bars). This may be due to the fact that in the giving talk scenario the speaker intends to look on faces to indicate whom they address and secure the listeners attention.

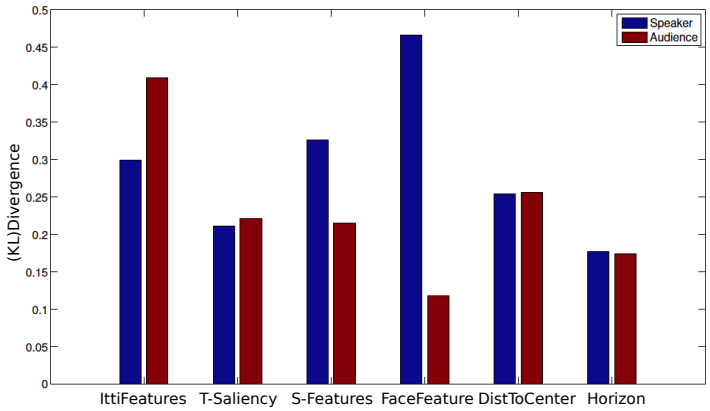


Figure 4.4: The KL divergence describing the performance of different SVMs models trained on a set of features individually in two scenarios (speaker vs. audience), averaged over all subjects.

Interestingly the model trained on Itti and face features combined outperforms models trained on other features combined in both scenarios (see Figures 4.5 and 4.6). This may be due to while listeners turn their gaze toward speakers to show their attentiveness and find suitable time windows to interact, speakers also find that time windows to gaze his / her presentation slides.

Finally, the overall summary of our analysis is shown in Figures 4.5 and 4.6. We can see the KL divergence matrices describing the performances of different SVMs models averaged over all testing images in the ”giving a talk-speaker-” scenario (Fig.4.5) and ”listening -audience-” scenario (Fig. 4.6). The KL divergence matrices are symmetric with respect to the main diagonal. The main diagonals show the performance of SVMs models trained on individual features. The lower/ upper triangular parts of the matrices show the performance for SVMs models trained on pairs of features combined. The models performance matrices for all subjects are presented in Appendix C.1.

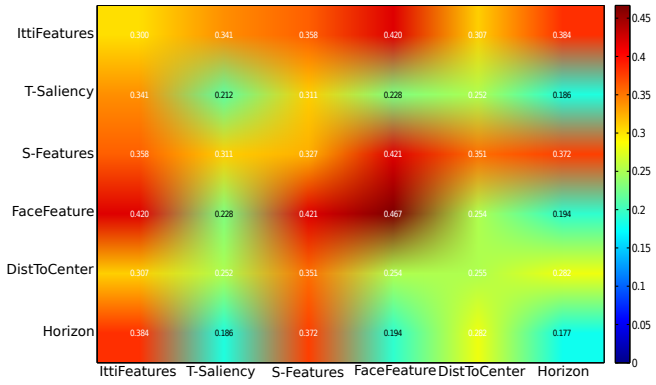


Figure 4.5: The KL divergence matrix describing the performance of different SVMs models trained on a set of features individually and pairs of features combined together, in the "giving a talk-speaker-" scenario, averaged over all subjects. The main diagonal shows the performances of the models trained on individual features. The lower/ upper triangular parts of the matrix show the performances of the models trained on pairs of features combined.

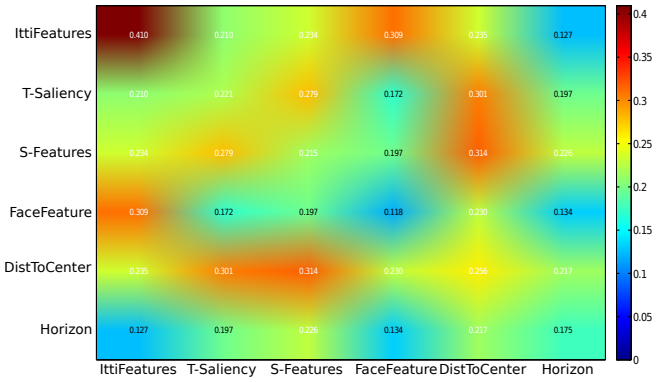


Figure 4.6: The KL divergence matrix describing the performances of different SVMs models trained on a set of features individually and pairs of features combined together, in the "listening -audience-" scenario, averaged over all subjects. The main diagonal shows the performances of the models trained on individual features. The lower/ upper triangular parts of the matrix show the performances of the models trained on pairs of features combined.

4.4 Conclusion

We have examined the prediction of gaze locations in meeting scenarios using different low, middle and high-level visual features. We trained a linear SVM to find out which features are descriptive in various scenarios. We concluded that the prediction differed according to the type of features we selected. Most interestingly, we found that models trained on the face features perform better than models trained on other features in the "giving a talk" scenario. But in the listening scenario the models trained on competing saliency features from Itti and Koch performs better than models trained on other features. This finding points towards including context information about the scene and situation into the computation of saliency maps as an important step towards developing models of eye movements, which operate well in the interactive environments settings. Our results suggest that context dependent saliency maps could become an integral part of any user model in ubiquitous computing settings, where users are experiencing a much richer visual environment than in the desktop computing settings. Our work is an important step towards building generative models for recognition of gaze in meetings, which explicitly take the context information into account. Most importantly, we suggest that a new series of analysis can even empirically measure the corresponding prior probabilities over latent variables in such models.

Chapter 5

Gaze Locations Prediction Based on Depth Features

We have demonstrated that eye movements depend on the behavioral context (Chapter 4), namely by the relative importance given to the individual visual features. Given that predicting eye movements in free-viewing scenarios are far from perfect, this raises the question: Can we do better in eye movement prediction when using more or other features compared to the ones that are commonly used? More specifically, will existing models perform better when using depth features as well? If so, how relevant are depth features? If they improve predictions this would suggest that humans consider depth as a relevant cue for eye movements, and existing models need to extended to include depth as a feature.

In this chapter, we start by first characterizing the statistical properties of depth images in natural scenes, because it informs us about the surrounded environment to which our visual system has been adapted, which form the a priori assumptions the humans in the experiments will likely use as well (Sec. 5.2-5.5). Then, we present a system that we have build to measure depth at the center of gaze in free-viewing scenarios (Sec. 5.6). We then conducted two studies, where we explored as to whether depth features are relevant in eye movement prediction (Sec. 5.7), namely in a free-viewing scenario with subjects walking freely (see Sec. 5.8.1), and in a scenario with fixed head-position (Sec. 5.8.2). We find that in both settings the depth information improves prediction and hence it should be included in predictive models.

The results of this chapter have previously appeared as conference publications [136], [130] and [131].

5.1 Introduction

While machine vision is a mature field with many industrial applications, artificial vision systems still fall short in terms of generalization when compared to the human visual system. The human visual system may be slow and built with sluggish components, but it works well under various lighting conditions and in many contexts. This is probably due to the vast amount of prior knowledge humans bring into interpreting visual scenes. Most computational vision researchers, who aims at reverse engineering the principles behind biological visual systems, adopted the hypothesis that the environmental signals shape biological visual systems [178, 218]. In other words, according to this approach the goal is not build a biologically inspired vision system, but to engineer the learning mechanisms of biologically inspired vision systems and then let them learn based on natural signals.

Much work has been invested into the statistical modeling of natural luminance images [34, 64, 178, 218, 197]. The rational behind many such approaches is that latent variables in generative probabilistic models of these luminance images will eventually correspond to meaningful scene descriptions in terms of, for example, properties of surfaces, objects. etc. Thus, once a structure for a probabilistic model of luminance images has been set up, the remaining task is to perform a model selection given natural images using, for example, Maximum Likelihood learning.

It was shown, however, that eye movements are far from a random sampling. It was even suggested that the statistics of natural images differ at the center of gaze when compared to random sampling [160]. Thus, taking into account eye movements is essential for shaping artificial vision systems via natural images. Another line of research has investigated the depth structure of natural scenes using range sensors [153, 212]. This depth structure is not directly accessible to the human vision system and needs to be inferred using stereo vision or other depth cues. Some statistical aspects of depth images as well as the relation between depth and luminance images have been investigated before [212], but the statistical properties of depth images at the center of gaze during free viewing are not clear. For example, simple questions such as "Do humans look more often to high contrast edges due to depth gaps than to edges due to texture borders?" have not been addressed yet.

We argue that characterizing the statistical properties of luminance and depths images at the center of gaze during free viewing is an important step in characterizing natural visual stimuli in order to learn, for example, better generative models of visual signals, which artificial visual systems can then invert to perform human-level visual computations.

5.2 Why Investigate Natural Stimuli to Understand the Human Brain?

Understanding how the brain is processing complex visual signals is a challenging problem in vision science. The investigation of natural images in terms of their statistical properties is a prominent approach in vision research, because it informs us about the environment to which our visual system has been adapted during evolution and ontogenesis. Therefore, many researchers turned to investigate biological vision systems in order to reverse engineer them and implement their principles into artificial vision systems. An important approach for developing a theory of vision is to characterize the visual environment in statistical terms, because this may provide objective yard sticks for evaluating natural vision systems using measures such as, for example, the information transmission rates achieved by natural vision systems. Then, with added "normative" assumptions about the potential goal of visual processing such as redundancy reduction, optimal coding, or optimal statistical inference predictions about the organization of natural vision systems can be derived.

5.3 Material and Methods

5.3.1 Description of the 2D/ 3D Natural Scenes Datasets

Our first analysis is based on a collection of images obtained originally from Stanford University (see Fig. 5.1) [170]. The total number of images in this dataset is 400. The 2D color pixel images were recorded with a high resolution 1704×2272 pixel (width $\times$ height), but the depth images with a resolution of 305×55 pixel (width $\times$ height). All images were inspected manually by us and then labeled as either "forest scene", "city scene", or "landscape scene". Only 12 images were labeled as landscape scenes, and we did not include them in our analysis, because they have fewer 3D structures compare to the city and forest scenes. 80 scenes were labeled as city scenes, and the remaining ones as forest scenes. Therefore, we compared only forest and city scenes.

Figure 5.1: Examples from the image collection (from [170]). **a)** Pixel images of city scenes. **b)** RGB luminance image. **c)** Depth map (yellow is closest, followed by red and then blue).

5.3.2 Features in the Luminance Images

The luminance images were first transformed into gray scale images. Then, each gray scale image is linearly decomposed into a set of edge feature responses to Gabor filters with different orientations. Gabor filters [67] are widely used in image processing for feature extraction and texture analysis [1]. The Gabor function is given by

$$G(x,y) = \exp\left(-\frac{\hat{x}^2 + \gamma^2\hat{y}^2}{2\sigma^2}\right)\cos\left(2\pi\frac{\hat{x}}{\lambda} - \psi\right)$$

with $\hat{x} = x\cos\theta + y\sin\theta$ and $\hat{y} = -x\sin\theta$.

We used orientations $\theta = \{0°, 15°, \ldots, , 165°\}$, but only one spatial frequency $\lambda = 6.1$ (and the standard deviation of the Gaussian $\sigma = 3.4$) and two spatial phases $\psi \in \{0, \pi/2\}$ and we set the spatial aspect ratio $\gamma = 1$, as recommended in [50] and [178]. Within each image we subtracted the mean from the filter responses to each orientation, and normalized the responses to the interval between -1 and 1. Figure 5.2 shows the histogram of such normalized responses for selected orientations.

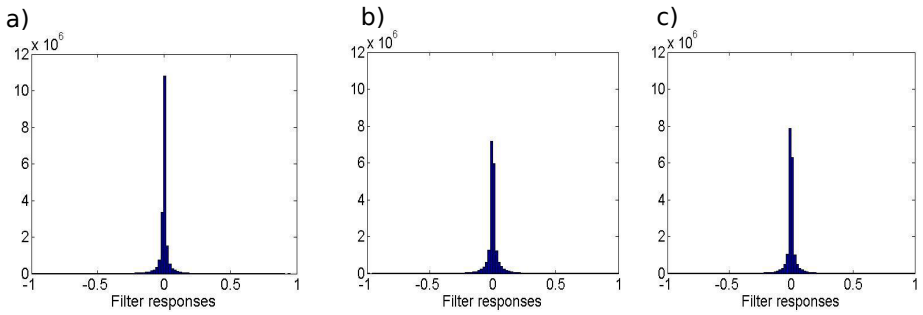


Figure 5.2: Histograms of the Gabor filter responses with three different orientations. **a)** Histogram of the Gabor filter responses in the vertical, **b)** oblique, and **c)** horizontal orientation.

5.3.3 Features in the Depth Images

Gap Discontinuity

A gap discontinuity in the underlying 3D structure is a significant depth difference in a small neighborhood. We measure gap discontinuity μ_{GD} by computing the maximum difference in depth between the depth of a pixel in the depth image and the depth at its eight neighboring pixels. Here, we considered the methods presented in [216]; μ_{GD} for a point (x, y) is defined as:

$$\mu_{GD}\,(x, y) = \max\{\ |\ z(x, y) - z(x + i, y + j)\ |\ : -1 \leq i, j \leq 1\}\,, \qquad (5.1)$$

where $z\,(x, y)$ represents a depth value. This quantity is then thresholded to generate a binary gap discontinuity map. In our analysis, we have empirically chosen a threshold $\mu_{GD}\,(x, y) > T_d$ where $T_d = 0.5$. Fig. 5.3b shows an illustration of a gap discontinuity map.

Surface Orientation Discontinuity

An orientation discontinuity is present when two surfaces meet with significantly different 3D orientations. Orientation discontinuity was measured using surface normal analysis. Here, we considered the methods presented in [7, 216]. The orientation discontinuity measure μ_{OD} is computed as the maximum angular difference between adjacent unit surfaces normal. First, a three-dimensional point cloud was constructed from the x, y, z coordinates for each pixel in a depth image. Then, each pixel is represented by a pixel patch $P_{(x,y,z)}$ compiled from the eight neighboring points in the point cloud. Finally, the unit surfaces normal are computed for each patch $P_{(x,y,z)}$ using Singular Value Decomposition (SVD).

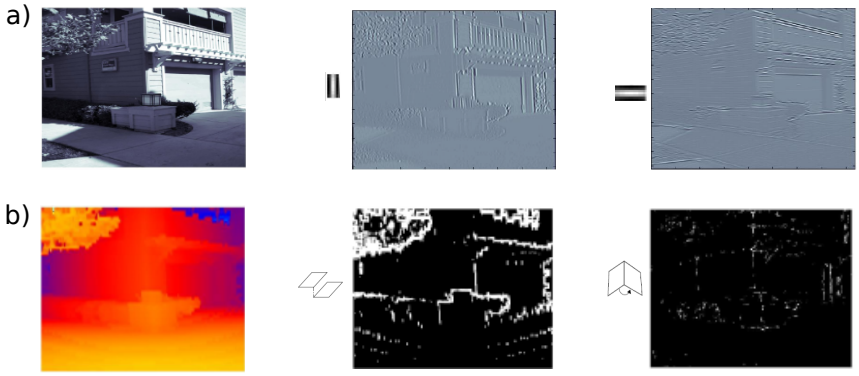


Figure 5.3: Examples for features in luminance and depth images. **a)** A gray-scale image convolved with two Gabor filters selective for the same spatial frequency, but different orientation. **b)** A depth map (left, where yellow is closest, followed by red and then blue) decomposed into its discontinuity maps: gap discontinuity map (middle) and orientation discontinuity map (right).

More specifically, for an image patch $P_{(x,y,z)}$ the orientation discontinuity is defined as

$$\mu_{OD}\left(P_{(x,y,z)}\right) = \max\left\{\alpha\left(n\left(P_{(x,y,z)}\right), n\left(P_{(x+i,\,y+j,\,\mathrm{z+k})}\right)\right) \text{ for } -1{\leq}i,j,k{\leq}1\right\}, \tag{5.2}$$

where $n\left(P_{(x,y,z)}\right)$ is a function, which computes the unit surface normal of a patch $P_{(x,y,z)}$ in 3D coordinates using Singular Value Decomposition (SVD), α is a function computing the angle between adjacent unit surfaces normal. It is given by

$$\alpha\left(P_1, P_2\right) = \arccos\left(n\left(P_1\right) \cdot n\left(P_2\right)\right). \tag{5.3}$$

max is function to compute the maximum angular difference between adjacent unit surfaces normal. This measure is also thresholded, but based on two criteria, namely i) an *angular criterion*: the maximum angular difference between adjacent unit surfaces normal should be more than a threshold $T_{\theta 1}$ and less than $T_{\theta 2}$, and ii) a *distance-based criterion*: the maximum difference in depth between a point and its eight neighbor's μ_{GD} should be less than a threshold T_d.

In our analysis, we have empirically chosen $T_{\theta 1} = 20°$, $T_{\theta 2} = 160°$ and $T_d = 0.5$, respectively. These values work with our specific data set where $0 \leq \mathit{depth\,values} \leq 80\,m$. Fig. 5.3b shows an illustration of an orientation discontinuity map.

5.3.4 Analysis Methods

Spatial Correlations

The correlation between pixels is probably the simplest statistical characterization of images. It reveals how spatial dependencies in images fall off with distance. The luminance and depth values are each given by a single number. Based on these numbers we estimated the correlation coefficient as a function of the distance between any two pixels, i.e.,

$$\mathrm{corr}\,(d) := \mathrm{corr}(X_1,\, X_2) = \frac{\mathrm{cov}(X_1,\, X_2)}{\sqrt{\mathrm{var}\,(X_1)\,\mathrm{var}\,(X_2)}}, \tag{5.4}$$

where X_1 and X_2 are two random variables representing two gray-scale/depth values of two pixels separated by d pixel. Here,

$$\begin{aligned} \mathrm{cov}\,[X_1,\, X_2] &= E\,[X_1 X_2] - E\,[X_1]\,E\,[X_2] && (5.5)\\ \mathrm{var}\,[X] &= E\left[X^2\right] - E\,[X]^2 && (5.6) \end{aligned}$$

are the covariance and variance, respectively. $E\,[\cdot]$ denotes the expectation, which we estimated by the sample mean.

Mutual Information

Mutual information (MI) typically measures the amount of information that one variable contains about another [46]. It is a graded quantification of the statistical dependencies between two random variables beyond second order. We used MI as a dependency measure between luminance and depth images. The luminance images are first linearly decomposed into a set of edge feature responses to Gabor filters in different orientations (see Figure 5.3a). For the depth images we computed gap and orientation discontinuities (see Figure 5.3b). Then, we estimated the MI between the discretized filter responses for each oriented filter and the binary discontinuity feature by sampling the Gabor filter responses and the gap discontinuity feature at the corresponding image location.

More specifically, the responses of Gabor filters at orientation θ were computed for all luminance images $I_1 ... I_n$. This orientation response vector is denoted by $X_\theta = [X_\theta\,(I_{i=1}), ..., X_\theta\,(I_{i=n})]$, and the discontinuity maps (combined gap and orientation discontinuity maps) of corresponding depth images are denoted by $Y = [Y\,(3D_{i=1}), ..., Y\,(3D_{i=n})]$, where $Y\,(3D_i) \in \{0, 1\}$.

The dependency between all luminance responses for orientation θ and the depth discontinuity maps is measured by the MI between X_θ and Y

$$MI_\theta\,(X_\theta; Y) = \sum_{x,y} \Pr_{X_\theta, Y}\,(x, y) \log\left(\frac{\Pr_{X_\theta, Y}\,(x, y)}{\Pr_{X_\theta}\,(x)\,\Pr_Y\,(y)}\right), \tag{5.7}$$

where $\Pr_{X_\theta,Y}(x,y)$ is the joint probability distributions calculated using a joint histogram, and $\Pr_{X_\theta}(x)$ and $\Pr_Y(y)$ are the marginal probabilities. The θ subscript emphasizes the fact that the MI is a function of orientation.

Local Standard Deviation as a Feature

We separated each image, i.e., both the luminance and the depth image, into non-overlapping small patches of size 6×6 pixel. Given that the resolution of these images is 305×55 pixel, a pixel-wise square patch corresponds to a vertically elongated rectangular patch in visual space. Therefore, we report all results for distances in pixel, which cannot be related directly to visual space, but still allows for a fair comparison between 2D and depth images, because the 2D images were resized and aligned to the depth images. Then, we computed the *standard deviation* of the 6×6 gray-scale values for each patch as the value of a local feature. The same was done for the depth image. This yielded, for each image I, the values $f^{2D}(\mathbf{x}; I)$ and $f^{depth}(\mathbf{x}; I)$ as the local feature for the visual field location $\mathbf{x}$.

The z-score to Quantify Saliency

Within an image I we consider a location $\mathbf{x}_1$ as more salient than another location $\mathbf{x}_2$ in terms of the 2D image feature when $f^{2D}(\mathbf{x}_1; I) > f^{2D}(\mathbf{x}_2; I)$; the same applies to the depth images. In order to abstract from the absolute values of these features, we consider the corresponding z-scores of these features and not the values of the features itself. The z-score at a location $\mathbf{x}$ in an image i is given by

$$z(\mathbf{x};\, I) = \frac{f(\mathbf{x};\, I) - \mu(I)}{\sigma(I)},$$

where $\mu(I)$ and $\sigma(I)$ are the mean and standard deviation of the feature *within* the i-th image, i.e., $f(\cdot)$ could be either $f^{2D}(\cdot)$ or $f^{depth}(\cdot)$. This way, saliency is defined relative to an image not in absolute terms.

5.4 Statistical Analysis of Registered Luminance and Depth Images

Some vision scientists turned to investigating the statistical structures of natural images in order to obtain statistical models of them for example [34, 64, 178, 218, 197]. A lot of feature statistics of natural images are discussed in [94]. Simoncelli et al. [178] explored sensory neural behaviors by investigating the efficient coding hypothesis and its role in the environmental statistics and neural responses. They were interested in testing models of visual processing in the

biological aspects. They presented in their work some of natural image statistics like the intensity statistics and spatial correlation. Schwartz & Simoncelli [172] proposed a nonlinear decomposition to build models of sensory control directly from the natural signals properties. They found that some sensory filters responses to natural signals are not statistically independent. The model they proposed deals with physiologically observed nonlinearities. Rothkopf et al. [166] showed that when learning basis function of a sparse generative models the distribution of orientations when fitting Gabor functions to the obtained basis functions shows an asymmetry, which may serve to explain the dominance of horizontally and vertically oriented filters (the oblique effect) in the center of the visual field with increasingly meridional directions in the periphery.

Most such studies focused on characterizing natural luminance images. Here we go beyond the analysis of the 2D pixel images by incorporating depth images. Those cannot be sensed directly by our visual system but need to be inferred. We employ information-theoretic measures to quantify the dependence between the oriented filter responses to luminance images and the features computed from corresponding depth images. We will arrive at an alternative explanation of the oblique effects, namely that it is rooted in the information from luminance images about depth features. Our approach incorporates the luminance images and depth images, similar to a few pioneering studies [153, 212]. We also find an asymmetry between the orientations of the oriented filters, but compared to other optimal coding theories our explanation of this asymmetry is different: In our interpretation it emerges, because we think of the distribution of oriented filters as being optimized to encode information about the depth features as not as a code for the optimal reconstruction of the luminance images. This may be important for image transmission, or the energy-efficient transmission of information in nervous systems, but our results suggests that another optimality criteria for information-based "normative" approaches to understand natural vision systems shall be taken into account, namely the faithful representation of relevant features, where here we consider properties of depth images as relevant.

This section is organized as follows: First, we describe the material and methods including the image material we used in this analysis (Sec. 5.3). Then, we present the results of our analysis, where we first compare the spatial correlations of the luminance and depth images for different types of scenes (Sec. 5.4.1) and then the dependency for responses of oriented filters and depth features as quantified by the mutual information.

5.4.1 Spatial Correlations in Luminance and Depth Images

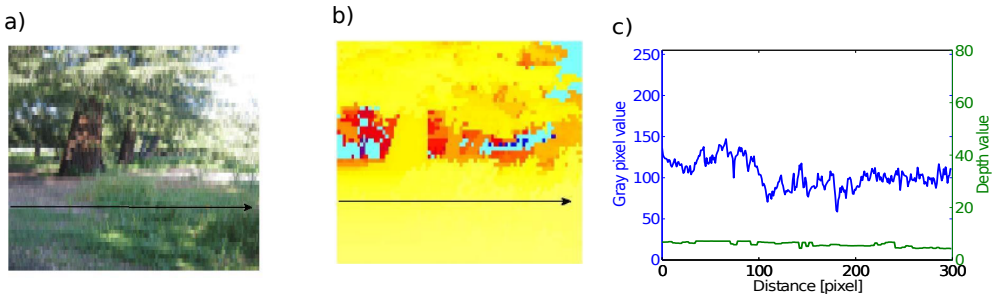


Figure 5.4: Illustration to compare the changes in luminance values and depth values. **a, b)** Color image and depth image (for depth image, yellow is closest, followed by red and then blue for the depth image) of an example scene. **c)** Gray-scale represents the average values of the color channels ((R+G+B)/3) and depth values of the pixels along the black arrow in panels a, b.

The luminance and depth images clearly differ in terms of their spatial correlations, which is illustrated in Figure 5.4a-c and summarized in Figure 5.5. Consider the example luminance image and the corresponding depth image: While the gray-scale values of the pixels in the luminance image along a horizontal line (see arrow) is variable (Figure 5.4c, blue line) the corresponding depths are almost constant (Figure 5.4c, green line). This suggests that the 3D environment is spatially more homogeneous than it appears from the luminance images.

The spatial correlation over many images reveals a scene dependence: The correlations in the luminance and depth city scenes are more extended than in forest scenes (Figure 5.5, green lines vs. blue lines). This is due to the presence of many spatially extended surfaces in the city scenes such as walls, streets, etc, while there are many depth discontinuities in forest scenes such as due to trees. On the other hand, the correlation in the depth images in the city scenes are more extended than in the corresponding luminance images (green dashed vs. green solid line). The same is true for forest scenes (blue dashed vs. blue solid line), which means that the 3D environment is generally more homogeneous than evident from the luminance images.

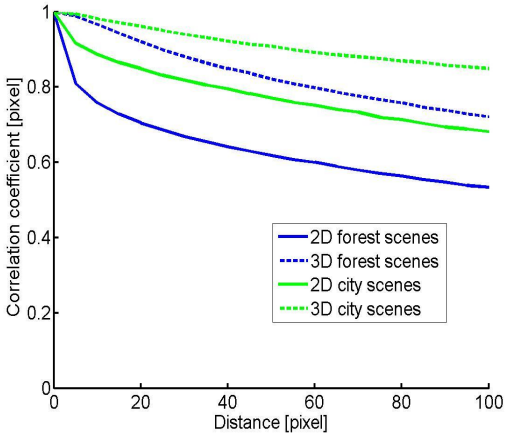


Figure 5.5: Spatial correlation as a function of distance measured in pixel. The pixel pairs selected for estimating this function were selected randomly from all possibly pixel pairs in an image with the corresponding distance in pixels.

5.4.2 Scene-Dependence of the Information about Depth Features in Luminance Images

The results of the mutual information analysis for *gap discontinuities* are shown in Fig. 5.6a,b for two different thresholds. The general pattern of the orientation-dependence does not differ much between the two threshold values we selected (Fig. 5.6a vs. Fig. 5.6b), but a scene-dependence shows up: In forest scenes the responses of the vertically oriented filters are much more informative about the gap discontinuities than for other oriented filters. This is not a surprise but expected, because the forest scenes have many trees with vertically oriented trunks, which are present in both the luminance and the depth images. When considering the *orientation discontinuities* separately (Fig. 5.6c), we do not find such a strong scene-dependence, but again a dominance of vertically oriented filter responses, i.e., they are more informative about orientation discontinuities than responses of other oriented filters. It is interesting to note that the information of the responses in horizontally oriented filters is lower in forest scenes compared to oblique and vertically oriented filters (in particular when compared to city scenes), which is probably due to the presence tree branches and trunks, but almost no horizontal gap discontinuities. Finally, the overall summary of our analysis is shown in Fig. 5.6d, where we computed the MI between the filter responses and a "joint" depth feature, i.e., the presence of either a gap or an orientation discontinuity.

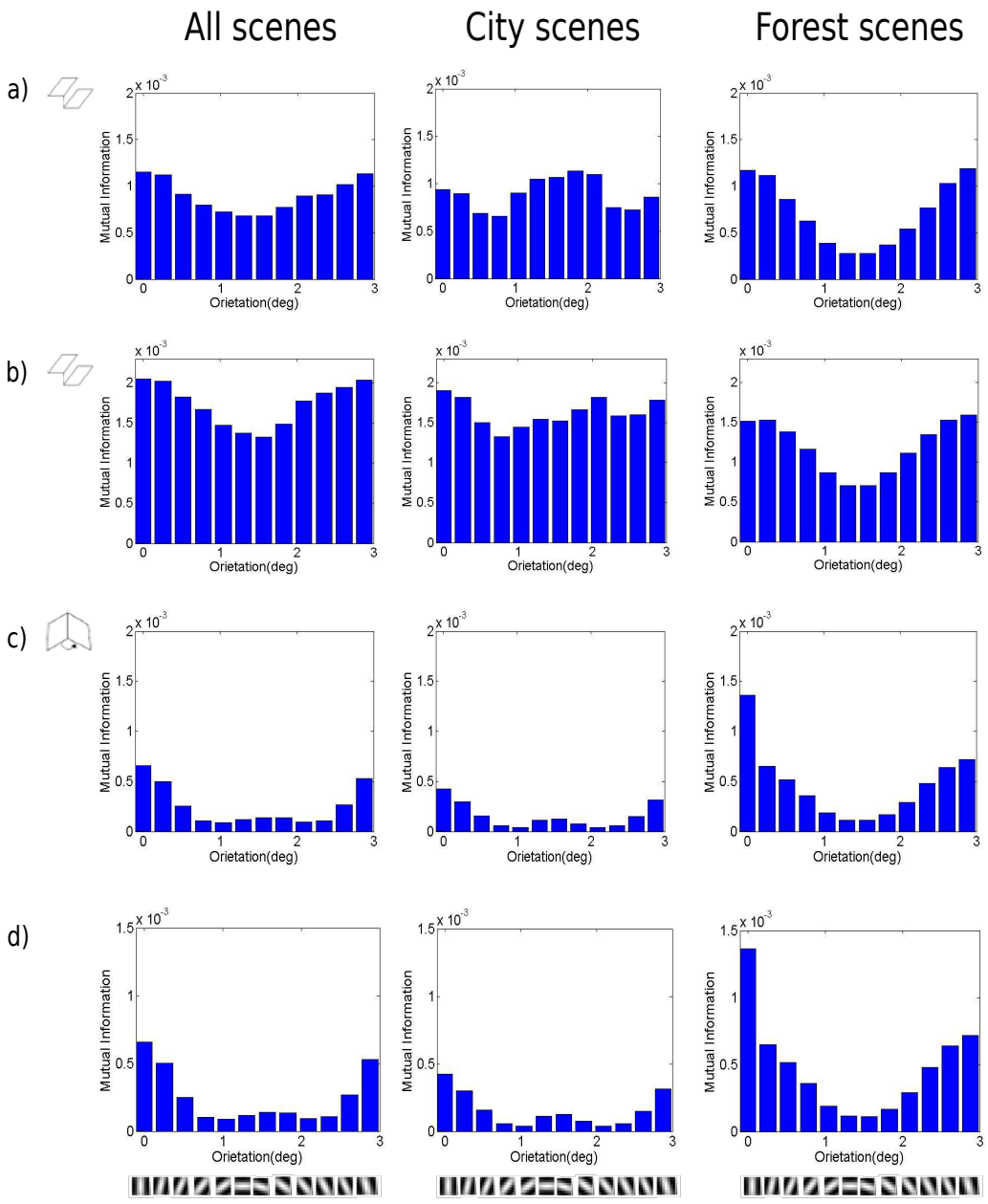


Figure 5.6: Mutual information between oriented filter responses and orientation filter responses and 3D gap discontinuities with different thresholds T_d. **a)** T_d=0.5 m and **b)** T_d=0.1 m. **c)** Information about orientation discontinuities. **d)** Information about a "joint" discontinuity, i.e., either a gap or an orientation discontinuity.

Our approach, which at this point does *not* use an explicit statistical model, can be viewed as a hybrid of the efficient coding approach and the use of generative statistical models: We determine, using information-theoretic measures, the potentially informative features in the luminance images and argue that those should be encoded most reliable and robust, but we do not postulate any particular neural code. Thus, while our results suggest an asymmetry in orientation processing as evident in the oblique effect, we cannot yet predict how exactly that should be reflected in the early human visual system. We explicitly refer to the properties of depth images, and it may be tempting to integrate such information directly into generative statistical models of the visual scenes. However, we intentionally did not formulate such models, because the whole idea of vision being "inverse graphics" may serve as a good guidance for computer vision, but it is still only a hypothesis as to whether biological visual systems implement such models, or if they follow other strategies. We argue that our results suggest that future natural image analysis may revitalize and refine the pioneering studies of depth and the corresponding luminance images [153, 212], because this way important constraints for any theory of visual processing can be obtained.

5.5 Scene-Dependence of Saliency Maps of Natural Luminance and Depth Images

From our analysis of the dependency between luminance and depth images features in natural scenes using mutual information. We found that the dependencies differed according to the type of visual environments. In this section we consider the dependency of saliency map in natural luminance and depth images. The notion of a saliency map has been turned out helpful in visual attention research: Here, certain locations in the visual field are determined as "salient" if they are – in statistical terms – outliers relative to the surrounding visual field locations. Computational modeling of the visual system was quite successful in the sense of predicting saliency maps based on image properties, which closely match the experimentally measurable maps of eye movements and fixation periods [98]. Here we report the results of a first experimental study to further improve the computation of saliency maps, i.e., to make them ultimately more predictive for eye movements. More specifically, we investigate a collection of natural scenes in terms of their saliency based on the two-dimensional (2D) pixel images and the corresponding depth images. The rational for investigating depth images is that they may reveal the "saliency that matters", because when interacting with the environment we evolved by interacting with objects in a three-dimensional (3D) world. Thus, we hypothesize that saliency maps respecting this will ultimately outperform saliency maps computed only

on the basis of 2D pixel images in terms of predicting eye movements.

Our analysis is based on a collection of images obtained from Stanford University which described in Section 5.3. All analyses were repeated by taking random subsets of size 80 from the forest scenes (as we had 80+ forest scenes). The results reported here were not affected by this difference in the sample sizes.

5.5.1 Distribution of Saliency in Natural and Depth Images

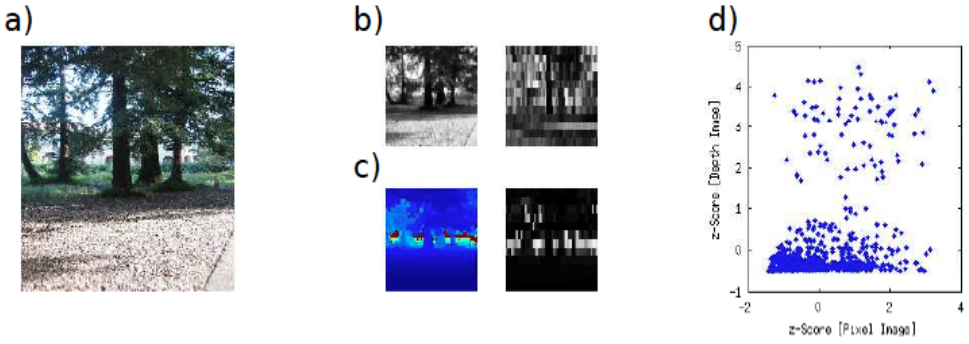


Figure 5.7: Example of saliency in a 2D and depth image. **a)** Color image of a scene. **b)** Resized image in gray-scale and corresponding saliencies based on the standard-deviation feature (see Section 5.3). Shown are the z-scores (white=high z-score, black=low z-score). **c)** Same as b) but for the depth image (but here blue is closest, followed by yellow and then red). **d)** Scatter plot of the z-scores in b,c with each point corresponding to an image patch. Saliencies in 2D appear to be unimodal, depth saliency is clearly bimodal.

To illustrate the computation of saliency maps we computed the saliency based on the standard-deviation feature for an example image shown in Figure 5.7a. Compare the z-scores shown Figure 5.7b,c (right panels). They are shown using the same color-scale. It is obvious that the luminance image has much more intermediate z-scores than the depth image, which has mainly small values with some high values at the location of the depth discontinuities at the tree trunks. This difference is also prominent in the scatter plot shown in Figure 5.7d, where the 2D saliencies appear to be unimodal but the saliencies for the depth image are bimodal, i.e., with either low or high values.

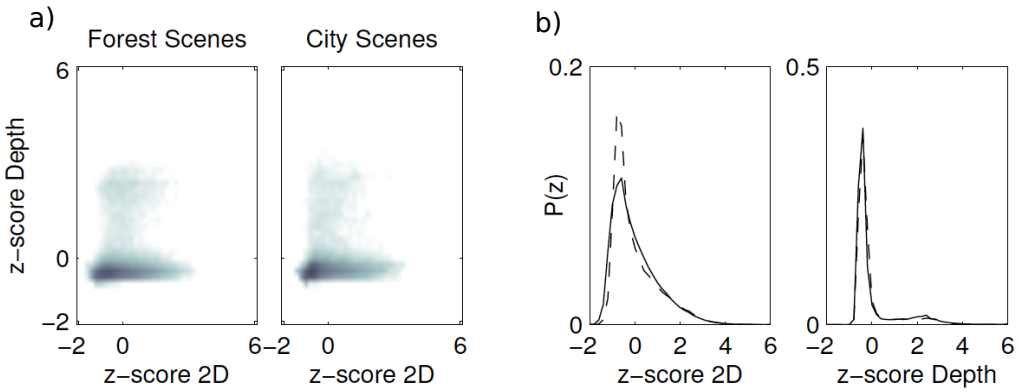


Figure 5.8: Joint and marginal distributions for the 2D and depth saliency. **a)** Joint probability distribution for the 2D and depth saliency computed for 80 forest and city scenes, estimated using a two-dimensional histogram. Both panels use the same (logarithmic) color-scale. Black corresponds to high probabilities. **b)** Marginal distributions for the forest (solid line) and city scenes (dashed lines).

We also performed this analysis for 80 forest and city scenes. The resulting joint distribution for the saliency in the 2D and depth images is shown in Figure 5.8a. Interestingly, the locations with high saliency in the depth image correspond to low salient locations in the 2D image. If the "saliency that matters" for our interaction with the 3D environment are the salient locations in the depth images, then they are not spotted by our (certainly very simplistic) 2D saliency algorithm. The marginal distributions are shown in 5.8b and reveal that 2D saliency is distributed unimodal, whereas depth saliency is bimodal. By visually comparing the joint distribution for forest with city scenes, it appears as if depth saliency is "more bimodal" in forest than in city scenes, but future studies need to explore this further. It is obvious that these two types of scenes differ in terms of their depth structure, but our first analysis could not make that distinction as clear as expected. We hypothesize that this is due to the (intentionally) rather simple feature we used and the lack of local "center surround interactions", i.e., a cross-talk between neighboring image locations.

In the next section, we will introduce the BatGaze system, that we have built to measure depth at the center of gaze in free-viewing scenarios. This enable us to use more and different features (both 2D and depth features/cues) for the computation of saliency maps, and systematically compare the predictions with experimentally measured eye movements.

5.6 BatGaze : A new tool to Measure Depth Features at the Center of Gaze During Free Viewing

In order to measure the depth features in the center of gaze we developed the BatGaze system, which combines an eye tracker (Figure 5.9a) with the lightweight depth sensor Xtion Pro Live from Asus (Figure 5.9b). The eye tracker is equipped with a camera to record the eye movements and a scene camera. After calibration, the gaze points are given in coordinates of this scene camera. The depth sensor was mounted next to the eye tracker's scene camera (Figure 5.9c), and it records depth images and RGB luminance images. The image streams of the Asus camera are already aligned to each other. We then developed procedures and tools to align them to the scene camera in order to obtain proper coordinates of the gaze point matched in space and time to the image streams from the Asus camera. Here we describe the details of the hardware setup, the software and processing, and the alignment in space and time.

5.6.1 Hardware Setup

5.6.1.1 The Mobile SMI Eye Tracker

SensoMotoric Instruments (SMI, smivision.com) offers to researchers state-of-the-art eye tracking systems [175]. It was our choice for recording the gaze data, because of its easy access to the raw data (gaze location, pupil position, pupil diameter, etc.). Our analysis were all done offline, but the eye tracker also gives online access to this data. The eye tracker uses two cameras (Figure 5.9a): The first is used to track the pupil and the second camera records the scene view. The gaze position is reported with a sampling rate of 50 Hz and a reported accuracy of $0.5° - 1°$. The scene camera comes with three lenses ($8, 6$ and $3.6\,mm$). The default $3.6\,mm$ lens provides a viewing angle of $\pm 31°$ horizontally and $\pm 22°$ vertically. The scene camera resolution is 752×480. We used the $3.6\,mm$ lens to record indoor scenes, where the observed objects are within $3\,m$ distance. Then, to avoid parallax error, we calibrated in a distance within $1 - 1.5\,m$. We used a calibration with 5 points so that the SMI recording software can compute the gaze location in scene camera coordinates from the recorded pupil images.

5.6.1.2 The Asus Xtion Depth Sensor

Depth sensing technology is now widely applied in video games and computer vision applications. As a side effect, new applications such as markerless full

body tracking become available to many researchers via low price consumer devices such as the Microsoft Kinect camera. Among the various sensors available the choice between different brands has to be made by respecting their specifications and the requirements of our BatGaze system. Options available to us were: the Asus Xtion Pro/Pro Live sensor, the Microsoft Kinect, and a time-of-flight (TOF) camera from PMDTec (pmdtec.com). We selected the the Asus Xtion Pro Live (Figure 5.9b) for two reasons: First, the camera does not need an external power supply as it is powered via USB, unlike the Kinect or the PMD TOF, which demand for external power supply. This makes the Asus Xtion Pro Live much more mobile and portable. Second, the Asus camera is much smaller than the Kinect and TOF and also weights less ($170\,g$), which makes it easier to mount it onto the head of a subject. The Asus Xtion Pro Live has three sensors: an infrared (IR) emitter with IR receiver to sense depth via the structured light principles and an RGB camera. The camera supports registration of depth and RGB frames in hardware and synchronized audio recording. It is most suitable for indoor environments. The camera has an effective depth sensing distance between $0.8\,m$ and $3.5\,m$ while the lenses effective angle is 58Âº horizontally and 45Âº vertically, which satisfies most computer vision application requirements. Asus released it with a complete SDK, which includes the OpenNI APIs[1].

5.6.1.3 Combining the SMI Eye Tracker with the Asus Xtion Depth Sensor

We first removed the base of the depth camera and mounted it on the front upper part of the helmet of the eye-tracker. We adjusted its position so that the RGB lenses of both cameras align vertically as much as possible. Then, we fixed the depth camera on the helmet using a tape and ensured that during free viewing the depth camera will not be moved or shacked. This is a very important part of our system setup. Any shifting in the depth camera position during an experiment will affect the alignment and registration process. To ease the movement of the subject during the experiment, we built two shelves to be carried on the back: one for a laptop connected to eye-tracker and the other for another laptop connected to Xtion camera. Before starting the calibration of the eye-tracker camera we checked the captured views from both cameras. If necessary, we readjusted the cameras' position to record the same view. The next step is the calibration of the eye-tracker camera using SMI's iViewX software. The depth camera does not need any calibration, but we usually ensure uniform light conditions.

[1] https://www.asus.com/us/3D-Sensor/Xtion_PRO_LIVE/

Figure 5.9: Illustration of the BatGaze hardware setup. **a)** Eye tracker from SMI (smivision.com). The field of view is not occluded as the eye tracking camera and the corresponding scene camera are mounted out of sight from the subject. **b)** Asus Xtion Pro Live camera, which captures depths images using the structured-light principle as well as RGB images. **c)** Our setup with a depth camera (here: the predecessor of the Asus Xtion Pro Live, which only recorded depth but no RGB images) mounted on the mobile SMI eye tracker.

5.6.2 Software Setup: Recording Software and Processing Tool chain

Asus ships the camera with the NiViewer tool, which records from all sensors of the camera. It can be configures via a configuration file. The recorded streams are saved in a custom file format (*.oni), which is accessible to OpenNI software. In a previous version of the BatGaze system we used the predecessor of the Asus camera and developed a custom recording software, but with the new Asus Xtion Pro Live it turned out that the NiViewer software is sufficient for our needs. We always recorded RGB images with a resolution of 640x480 at 25 fps. We developed a custom player for oni-files (OniPlayer), which can read, process and render scenes and depth frames from oni-files. Most importantly, it converts images into a custom binary format for further processing using MATLAB. Finally we also developed a custom software called (XtionRecorder) that can stream directly from the Xtion camera for online processing.

5.6.2.1 Temporal Synchronization

The depth camera delivers the depth map and the RGB frames already synchronized with timestamps. The scene camera of the eye tracker also delivers RGB frames as well as gaze locations, both with timestamps. All synchronization was done offline. When both cameras were recording, we generated two special events in time, which were recorded by both cameras: a “clapper board” at the beginning and end of recording. More specifically, we did the alignment using the timestamps of both cameras, where both are given in microseconds.

For depth camera, let

- $\mathbf{Z}$ be a three-dimensional matrix of n depth frames from the Asus camera; $\mathbf{Z}$ has size of $640 \times 480 \times n$ (width $\times$ height $\times$ frames),
- $\mathbf{R}$ be a matrix of n RGB frames from the Asus camera; $\mathbf{R}$ has a size of $640 \times 480 \times 3 \times n$ (width $\times$ height $\times$ RGB channels $\times$ frames),
- T_Z be set a function T_Z : Frames $\rightarrow$ Timestamps to obtain the timestamps for each depth frame,

and for eye-tracker camera, let

- $\mathbf{S}$ be a matrix of m frames from the scene camera; S has size of $752 \times 480 \times 3 \times m$ (width $\times$ height $\times$ RGB channels $\times$ frames),
- $\mathbf{G}$ be a matrix of m gaze points from the SMI system, one for each frame from the scene camera; $\mathbf{G}$ has size of $2 \times m$ (x/y gaze point position $\times$ frames), and
- T_S be a function T_S : Frames $\rightarrow$ Timestamps to obtain the timestamps for each frame.

The recording of depth data is started always some seconds later than the recording with the eye-tracker camera, and stopped always first, so that we have $n < m$. For each recording we identified reference frames i_Z^{ref} and i_S^{ref} for the depth and scene camera, respectively, by manually inspecting the frames around the first "clapper board" event. Then, a frame from the scene camera, i_S, was assigned to a frame from the depth camera, i_Z, where the difference in timestamps was smallest, i.e.,

$$\begin{aligned} i_Z\left(i_S\right): &= argmin_i \left|t_Z\left(i\right) - t_S\left(i_S\right)\right| \\ t_Z\left(i\right) &= T_Z\left(i\right) - T_Z\left(i_Z^{ref}\right) \\ t_S\left(i\right) &= T_S\left(i\right) - T_S\left(i_S^{ref}\right) \end{aligned}$$

Then, we generated a new pair of aligned streams with equal length. The results of this temporal alignment were double-checked with the software "Kinovea" (kinovea.org), which supports frame-by-frame inspection of videos. Finally, we double-checked temporal alignment by inspecting the alignment of the second "clapper board" event at the end of the recording. Failures of alignment for this second event would be indicative of technical problems with the timestamps from either the SMI or Asus system.

5.6.2.2 Spatial Registration of Images

After the frames have been aligned temporally, they are also aligned spatially. We aligned each pair of frames using a transformation obtained from a pair of frames in the beginning of the recording, i.e., we assume that the spatial relation of the two cameras does not change in the course of a recording. The geometrical aligning of images is termed image registration, and many algorithms are available for that. We registered the scene frames of both cameras using a simple registration of two 2D images.

If (x, y) is a pixel in the eye trackers scene camera and (x', y') is a pixel in the Asus scene camera, we can write:

$$\begin{bmatrix} r_1 & r_2 & t_1 \\ r_3 & r_4 & t_2 \\ s_1 & s_2 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

where

$$\begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

is a rotation matrix, $[t_1\, t_2]^T$ is a translation vector, and $[s_1\, s_2]^T$ accounts for scaling/shrinking. MATLAB already offers a solution to this transformation problem (2D image registration), which is based on manually identifying pairs of matching points in the two images. The resulting transformations are then applied to the gaze positions, which are in the coordinates of the scene camera form the eye tracker, in order obtain their coordinates in the Asus scene camera. Figure 5.10 shows an example of a spatial registration. (See Appendix B.2 for more details about the workflow of the BatGaze system).

5.6.3 Experimental Validation

Here we report the experimental validation of the BatGaze system. We explicitly instructed subjects, who were freely walking around a table with boxes on top of the table, to direct their gaze to either the edges or the surfaces of the boxes ("look at edges" vs. "look at surfaces"). The rational for these instructions was to collect ground truth data: We expected that an analysis of the structure of the depth images at the center of gaze will uncover a higher probability of inspecting edges in the edge condition as compared to surfaces, and vice versa in the surface condition.

5.6.3.1 Participants

Three participants took part in this study (one female, two male, 24-40 years). The participants were with normal vision and no history of neurological prob-

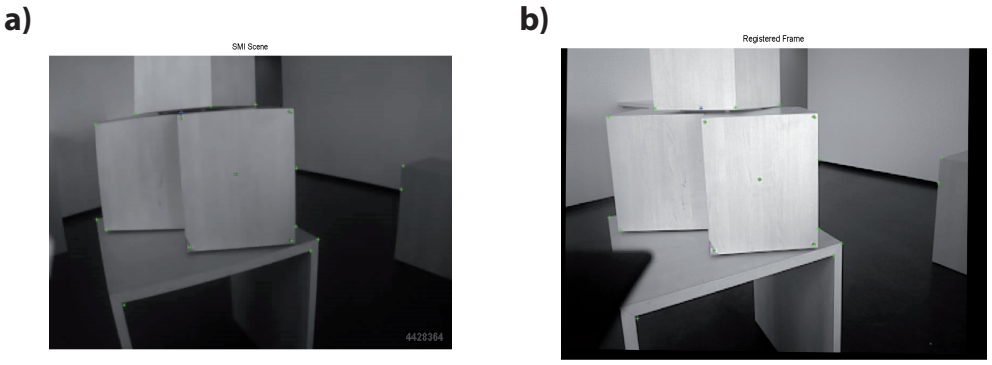


Figure 5.10: Example of a spatial registration. a) Frame from the scene camera of the eye tracker, after registration to the image from the Asus scene camera. b) Corresponding image from the Asus scene camera. The small green dots are the identified points for matching the images.

lems. All of them were daily computer users. They participated in the main experimental validation of the BatGaze system.

5.6.3.2 Experimental Design

The three subjects executed the two task conditions "look to edges" and "look to surfaces", while they were freely walking around the table with boxes on top of the table in a big hall. A total of three boxes were assembled on top of the table. The eye tracker was calibrated before each experiment using a 5-point calibration target. While the subjects were performing the task we recorded the gaze positions, the scene frames and the depth frames on one computer. The overall duration of a single recording was $80s$ on average. Subjects were given verbal instructions. In the first condition ("look to edges"), subjects were instructed to look only to the edges of the boxes. In the second condition ("look to surfaces"), subjects were instructed to look only to surfaces. The three subjects participated multiple times in each conditions.

5.6.3.3 Results: Depth Features at the Center of Gaze

The recorded data was analyzed by computing the probability of finding a gap discontinuity in a neighborhood of 25, 49 and 81 pixel around the gaze location. This was done for both conditions. Figure 5.11 shows the estimated probabilities and confirms, as expected, that the probability of finding a gap discontinuity in the "look at edges" condition is higher than in the "look at surfaces" condition. Also note that the probability for a gap discontinuity increases with increasing neighborhood size while is remains largely constant

in the surface condition. This is due to the fact that the surfaces of the boxes in our study was rather large compared to the largest neighborhood, and that the subjects presumably looked at the center of the surfaces. These results validate that the BatGaze system is working as anticipated. Future work can now address the accuracy of the whole system, which will be only limited by the accuracy of the eye tracker.

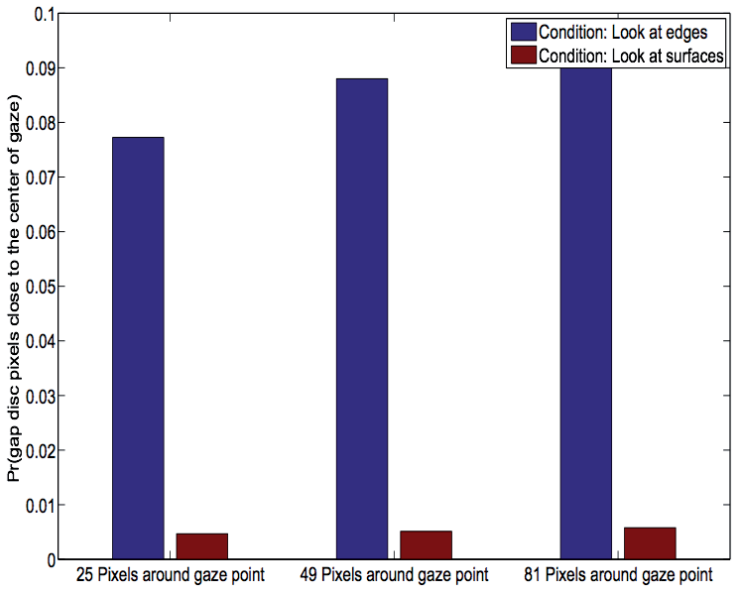


Figure 5.11: Bar plot for the depth features around the gaze point in the two experimental conditions**.** Shown are the probabilities of finding gap discontinuity around gaze point.

5.6.3.4 Results: Luminance Features at the Center of Gaze

In order to characterize the features in the luminance images at the center of gaze we first selected 500 consecutive frames from the middle of a block for each condition. Then we transformed each frame into gray-scale images. Each gray-scale image is linearly decomposed into a set of edge feature responses to the Gabor filters with different orientations. In this analysis, however, we used only the responses to horizontal and vertical filters.

We then compiled histograms for the responses to these Gabor filters from the pixels around the gaze point in each condition. Figure 5.12a,b show these histograms and reveal that the probability of horizontal or vertical edges being present, i.e., non-zero filter responses, around the center of gaze is much higher in the "look at edges" compared to the "look at surfaces" condition. This hold true for both horizontal (red bars) and vertical edges (blue bars).

In order to further highlight this difference between conditions, we also generated "combined histograms", where we did not distinguish between the orientations of the Gabor filters. Figure 5.13 shows clearly that in the "look at edges" condition the non-zero filter responses are much more frequent for all sizes of the neighborhood.

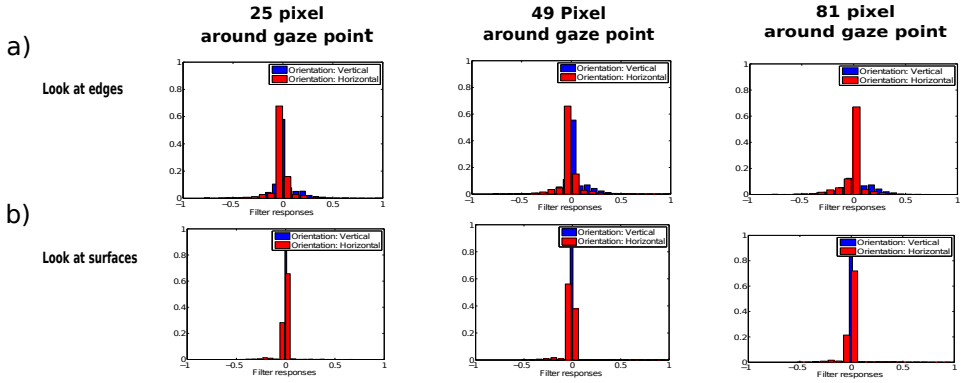


Figure 5.12: Normalized histograms for the Gabor features in the vertical and horizontal directions at the center of gaze in the two experimental conditions. **a)** Condition "look at edges". **b)** Condition "look at surfaces".

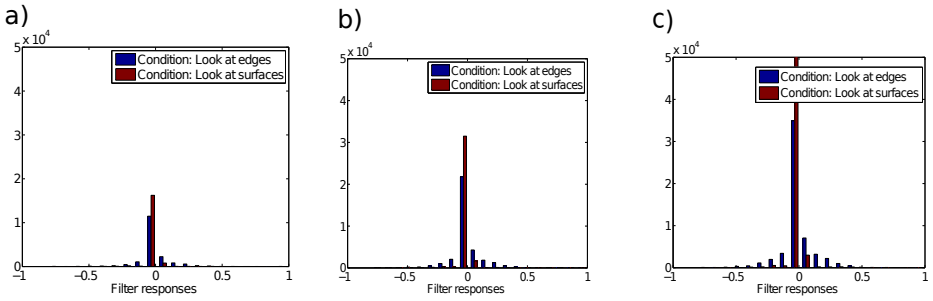


Figure 5.13: Normalized histograms for the combined Gabor features (vertical and horizontal) in the experimental conditions. **a-c)** Different neighborhoods (25, 49 and 81 pixel).

5.7 Gaze Location Prediction with Depth Features

In this section we present the results of a first experimental study to improve the computation of saliency maps, by using luminance and depth images features.

More specifically, we have recorded the center of gaze of users when they were viewing natural scenes. We first examined the statistical characterization of depth features in natural scenes at the center of gaze. We then examined the presence of depth features around gaze locations. We used machine learning to train a bottom-up, top-down model of saliency based on 2D and depth features/cues.

5.7.1 Eye Tracking Experiments

Because of the limitation to use our BatGaze system to collect eye movements data from environments where we have a rich visual information. We have also recorded the center of gaze of users when they were viewing natural scenes in stationary setting.

The rational for investigating depth features for gaze location prediction is that they may reveal the "saliency that matters", because when interacting with the environment we evolved by interacting with objects in a three-dimensional (3D) world. Thus, we hypothesize that saliency maps respecting this will ultimately outperform saliency maps computed only on the basis of 2D pixel images.

5.7.1.1 Stimulus Material

Forty images selected form our 2D/3D natural scenes dataset (see Sec. 5.3.1) were presented to five subjects. The 2D color pixel images were recorded with a resolution of 1704×2272 pixels, but the depth images with a resolution of 305×55 pixels. They where 40 images from "forest scene", "city scene", and "landscape scene".

5.7.1.2 Measuring Gaze Locations

We used an iView X HED 4 Eye Tracking System (SMI) to record eye position. The system reports gaze positions with a sampling rate of 50 Hz and a reported accuracy of 0.5°-1°. We used the default lens ($f = 3.6$ mm) for the scene camera which provides a viewing angles of 31° horizontally and 22° vertically. The eye tracker scene camera has a resolution 752×480 pixels.

5.7.1.3 Participants

Five participants took part in this study (Five males, 18-40 years). Three of the viewers were researchers in the institute of computer science and the others were naive viewers.

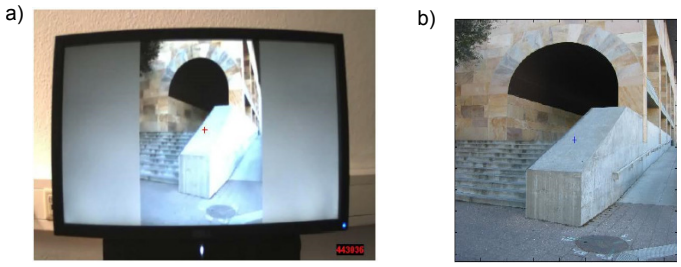


Figure 5.14: Example of a gaze registration. **a)** Frame from the scene camera of the eye tracker and the corresponding gaze point (Red cross) . **b)** Registered gaze point (Blue cross) on the corresponding high resolution image.

5.7.1.4 Experiment Design

Each subject carried out a 9-point calibration procedure before the start of the experiment. The stimuli presented to the viewers in similar presentations order on a computer screen of resolution $1280x1024$. All viewers sit at a distance of approximately $1.5\,m$ from the computer screen in a dark room, this corresponds to a distance where the subjects could comfortably view the display, and used a chin rest combined with a bite bar to stabilize their head. An mobile eye tracker recorded their gaze path on a separate computer as they viewed each image at full resolution for ten seconds separated by two seconds of viewing a gray screen. The scene camera of the eye tracker delivers RGB frames as well as gaze locations, both with time stamps (Figure 5.14 a), Also we recorded information about which and when each image have been presented to the viewer.

Our analysis were all done offline. First we aligned the frames temporally to the high resolution images using the information we recorded about when each image have been presented to the viewer. Then we used normalized Cross-Correlation [118] to register each part of interest in each frame to the corresponding high resolution image. Using the transformation obtained to register each gaze point to the high resolution image (Figure 5.14 b), we generated a saliency map of the locations fixated by each viewer. Also, we convolve a Gaussian filter similar to [199, 195] across the user's fixation locations in order to obtain a continuous saliency map of an image from the eye tracking data of a user.

5.7.2 Features Used for Machine Learning

Different low-level features were collected. For example: the intensity, orientation and color contrast channels as calculated by Itti and Koch's saliency method [98]. Also, each gray-scale image is linearly decomposed into a set of edge feature responses to Gabor filters with different orientations. We used

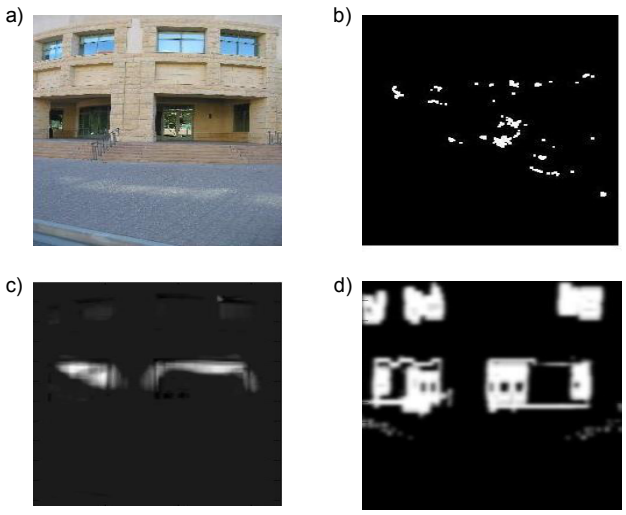


Figure 5.15: Examples for features in luminance and depth images. **a)** Natural scene. **b)** Fixation map recorded with our stationary setup . **c)** Itti & Koch features. **d)** Depth discontinuity features.

orientations $\theta = \{0°, 15°, \ldots, 165°\}$, but only one frequency and two spatial phases. Within each image we subtracted the mean from the filter responses to each orientation, and normalized the responses to the interval between -1 and 1 (Fig. 5.3 (a)). We used Gabor filters responses to compare the performance with the 3D features.

We also extracted a gap discontinuity map and orientation discontinuity map for each depth image using methods presented in the previous chapter (see Sec. 5.3.3) and combine them together to generate the gap depth features.

5.7.3 Classifiers for Predicting Gaze Locations

Similar to the previous chapter, we use a machine learning approach to train a classifier from human eye tracking data. We use a linear Support Vector Machine (SVM) to find out how depth features are informative compare to other features. Again we split our dataset into training images and testing images in order to train and test our model. We selected randomly 200 positively labeled pixels from the top 40% salient locations for each image, and 200 negatively labeled pixels from the bottom 60% salient locations.In order to have zero mean and unit variance we normalized the features of our training set and used the same normalization parameters to normalize our test samples. Finally, we predict the saliency per pixel using a particular trained model, for each image in our dataset.

5.7.4 Error Measure

Again we used the Kullback–Leibler (KL) divergence to measure the distance between distributions of saliency values at human vs. random eye positions (see Sec. 2.3.4 for more details). Models show higher KL divergence, are better in predicting human fixations, because usually human gaze towards the regions with the highest model responses and avoiding the low model responses regions.

5.8 Results

5.8.1 Depth Features at the Center of Gaze.

For each depth image we extracted square image patches around the subject's center of gaze. We also extracted image patches selected at random positions.

5.8.1.1 Depth Values around Gaze

We first compared the distribution of depth values of patches in the center of gaze to that expected from random sampling. It is clear that, the distribution of depth values of patches at the center of gaze statistically differ than from random sampling. Figure 5.16 (a) shows that the normalized histogram of the random sampling from 40 scenes, averaged over all subjects, differ than the distribution of patches in the center of gaze (see Figure 5.16 (b)) (with P-value = 1.091e-016 of the two-side Kolmogorov–Smirnov (K-S) test with significance level of 0.05).

Figure 5.16(c) shows that the normalized histogram of patches in the center of gaze over 40 scenes averaged over all subjects in the first three seconds of viewing the scenes differ than the last seven seconds (see Figure 5.16(d)) (with P-value = 8.6504e-065 of the two-side Kolmogorov–Smirnov (K-S) test with significance level of 0.05).

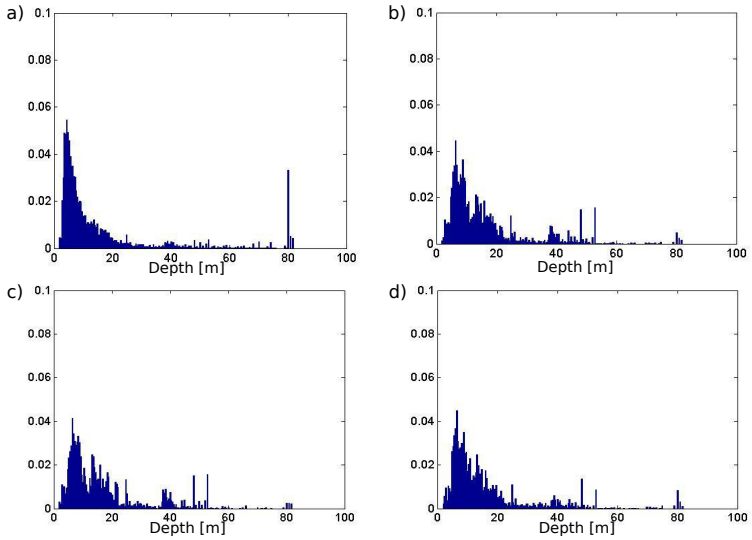


Figure 5.16: **a)** Normalized histogram of depth values of random sampling over 40 scenes, averaged over all subjects. **b)** Normalized histogram of depth at gaze locations, averaged over all subjects. **c)** Normalized histogram of patches in the center of gaze over 40 scene for each subject in the first three seconds of viewing the scenes, averaged over all subjects. **d)** Normalized histogram of patches in the center of gaze over 40 scenes in the last seven seconds of viewing the scenes, averaged over all subjects.

5.8.1.2 Depth Features around Gaze

Before we used depth features as new information for predicting gaze locations. We examined the presence of depth features around the center of gaze locations. The result of the distribution of depth features in a different neighborhoods around the gaze location averaged over all subjects are shown in Figure 5.17(a) and the distribution of depth features around gaze for individual subjects are shown in Figure 5.17(b). It is clear that the presence of depth features around gaze locations are high. This suggest that saliency maps models respecting this will ultimately outperform saliency maps computed only on the basis of 2D pixel images in terms of predicting eye movements.

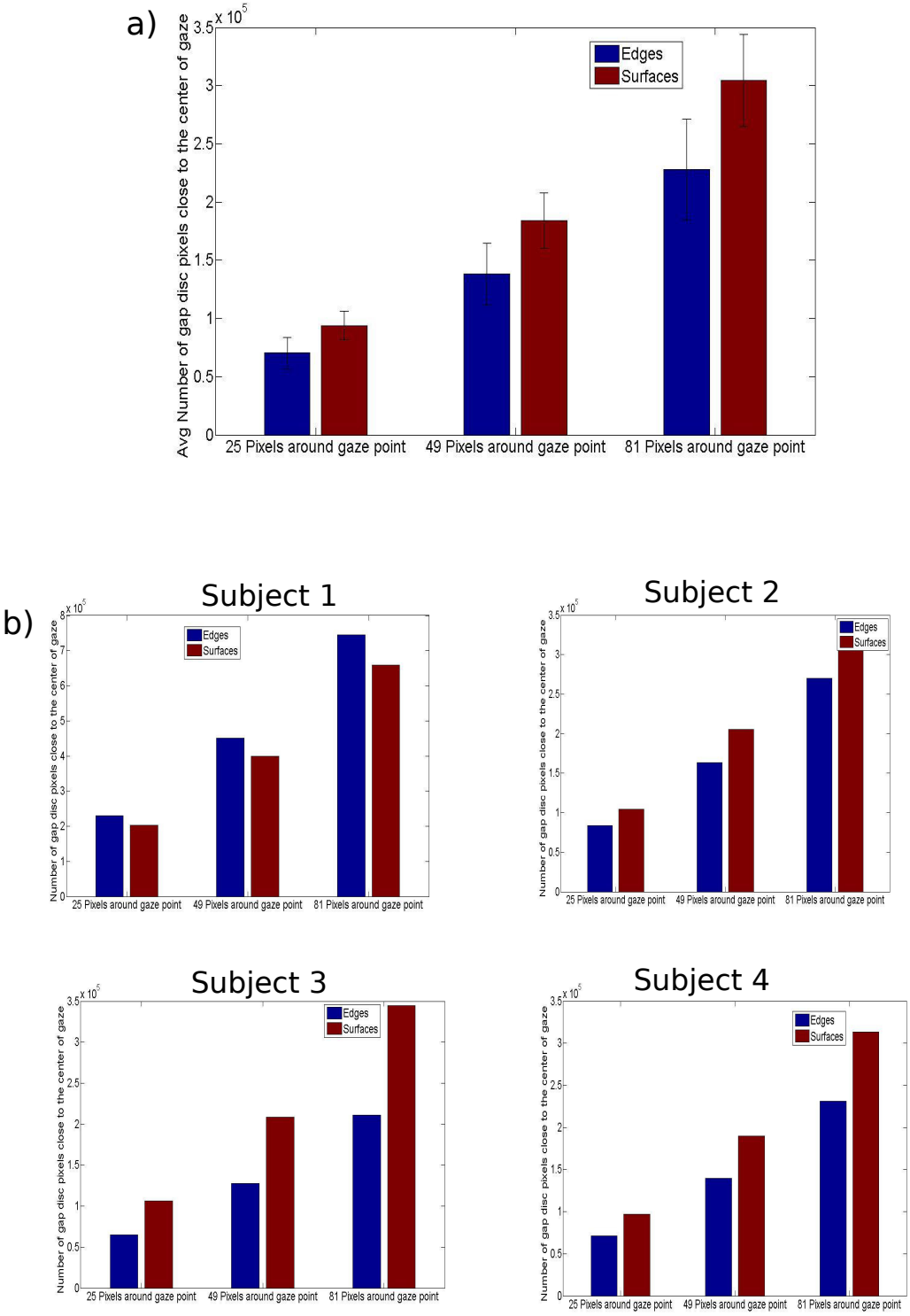


Figure 5.17: The presence of depth features in a different neighborhoods around the gaze points. **a)** Bar plot for the presence of depth features in a different neighborhoods around the gaze points, averaged over all subjects. **b)** Bar plot for for the presence of depth features in a different neighborhoods around the gaze points for individual subjects.

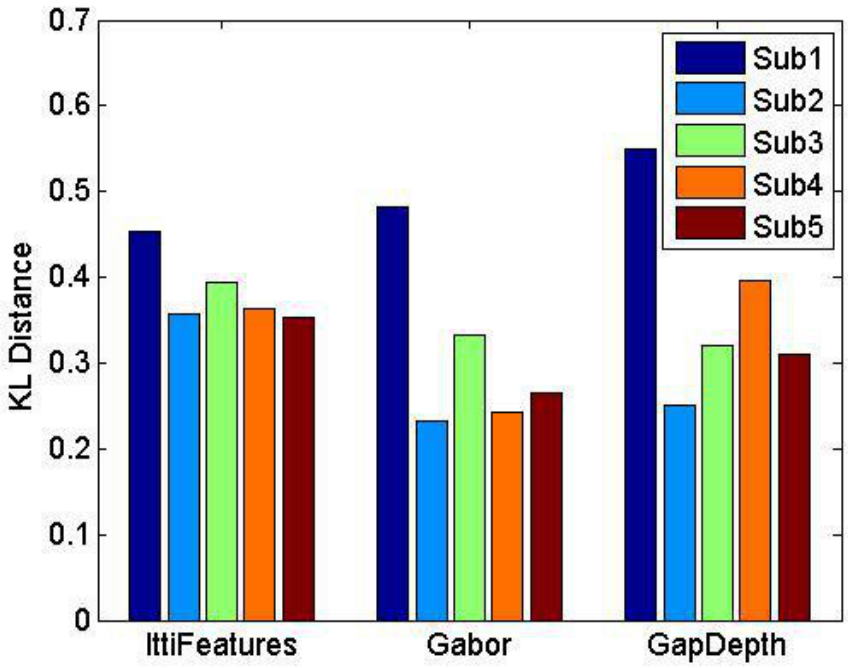


Figure 5.18: The KL divergence describing the performance of different SVMs trained on each feature individually, for individual subject.

5.8.2 Gaze Location Prediction when Viewing Photos of Natural Scenes

We measured the performance of saliency prediction using KL divergence (see Section 5.7.4). Figure 5.18 describing the performance of different features models for each subject averaged over all testing images. We predict the saliency for each image using a specific trained model. We can see that the prediction differ according to the type of features we selected. While the model trained on competing saliency features from Itti and Koch perform better than the models trained on other individual features (i.e. only Gabor or only depth features). The averaged result over all subjects shows this finding (see the diagonal of Figure 5.19).

Interestingly the models trained on Itti & Koch combined with depth features outperform models trained on other individual features (i.e., only Gabor or only depth features), or trained on combination of these features. (see Figure 5.19). It is interesting to note that, depth features combined with luminance features improve the prediction of gaze locations.

Finally, the overall summary of our analysis is shown in Figure 5.19 where we computed the KL performance for SVMs trained with different individual

features and combined together, averaged over all subjects. We perform the statistical test (t-test2) for all pairs of features (i.e., KL_Itti vs. KL_Gabor, KL_Itti vs KL_GapDepth and KL_Gabor vs KL_GapDepth) with significance level of 0.05 the corresponding P-values were (0.3740, 0.9240 and 0.4488) respectively.

In Figure 5.19, we see the KL divergence matrix describing the performance of different SVMs models averaged over all subjects. The KL divergence matrix are symmetric with respect to the main diagonal. The main diagonal shows the performance for SVMs models trained on individual features. The lower/ upper triangular parts of the matrix show the performance for SVMs models trained on pairs of features combined. The models performance matrices for all subjects are presented in Appendix C.2.

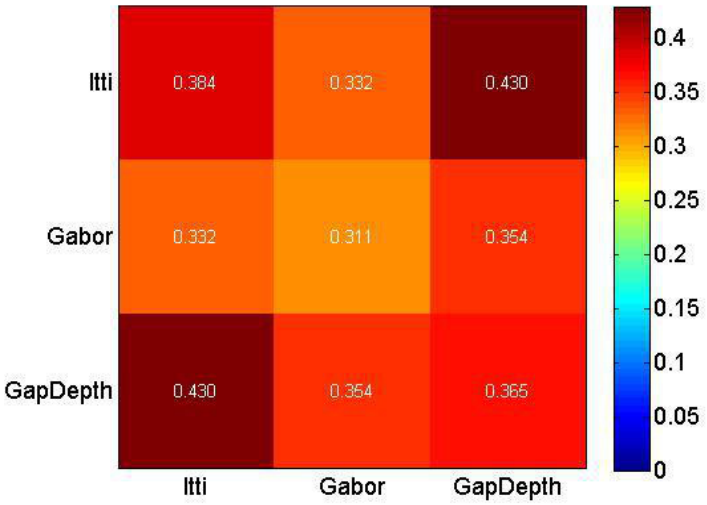


Figure 5.19: The KL divergence matrix describing the performance of different SVMs models trained on set of features individually and pairs of features combined, averaged over all subjects. The main diagonal shows the performance of the models trained on individual features. The lower/ upper triangular parts of the matrix show the performance of the models trained on pairs of features combined.

5.9 Conclusion

In this Chapter, we first have analyzed the dependency between luminance and depth images features in natural scenes using mutual information. We find that the dependencies differ according to the type of visual environments. Most interestingly, we find that response of vertical filters carry most information about

3D discontinuities. This can explain the preferred processing of vertical orientations, but not because the corresponding orientations are more frequent in the luminance images, but because they are more informative about ecologically relevant depth features. This is in contrast to other efficient coding hypotheses. More specifically, such hypotheses state that the visual system shall encode the stimuli from the sensory periphery most efficiently. This could be done by transforming these stimuli into neural representations, which are less redundant such as factorial codes. Other hypotheses about the visual system state that it implements statistical models of the environment. In that way of thinking vision is "inverse graphics", but finding and learning proper statistical models and inverting them is a current research topic.

Also, we have analyzed the saliency in 2D pixel and depth images using a rather crude and highly simplistic feature: the local standard deviation of pixels. We find that saliency in depth images is bimodally distributed with highly salient locations corresponding to low salient 2D image locations. We also found differences between scenes in the spatial correlation functions and a tendency for saliency being more bimodal in forest than in city scenes. It is obvious that these two types of scenes differ in terms of their depth structure, but our first analysis could not make that distinction as clear as expected. We argue that this is due to the (intentionally) rather simple feature we used and the lack of local "center surround interactions", i.e., a cross-talk between neighboring image locations.

Furthermore, we have analyzed the statistical of depth features in natural scenes at the center of gaze. We found that the distribution of depth values of patches at the center of gaze differ than from random sampling. Most interestingly, we found that the presence of depth features around gaze locations were high. This finding points us towards including depth cues into the computation of saliency maps as a promising approach to improve their plausibility.

Then, we used machine learning to train a bottom-up, top-down model of saliency based on 2D and depth features. We found that models trained on Itti & Koch and depth features combined outperformed models trained on other individual or combination of these features. As a consequence, we find that, the depth information improves prediction and hence it should be included in predictive models..

Our approach, of using joint luminance and depth features is an important step towards developing models of eye movements, which operate well under natural conditions such as those encountered in HCI settings.

Chapter 6

Eye Movements Prediction for Wall-Sized Displays with Bezels

In chapters 4 and 5 we have demonstrated how relevant different features are for eye movements prediction in different behavioral contexts. To fill the gap between models performance in real world scenarios and human behaviors. In this chapter, we investigate how existing predictive gaze models perform in the interaction scenario with wall-sized displays compare to human eye movements behaviors. Wall-sized displays with bezels are now frequently used in various application domains. It has been recognized that interior bezels bring a new set of interaction challenges from fundamental selection, manipulation to task management [146]. But the performances of visual saliency algorithms on this type of wall-sized display compare to human viewing behavior, have not been studied yet. Given that, the bezels between the individual smaller displays are striking visual features that cause high-contrast borders, this suggests the questions: does bottom-up saliency predictive models will consider them and adapt the predictions accordingly? Furthermore, is human viewing behavior really that much affected by the bezels?

In this Chapter we report two studies that measure the effects of bezels on human eye movements (Sec. 6.2) and on saliency algorithm predictions (Sec. 6.3). Subjects observe natural images on two different display systems, with and without interior bezels. In short: we find that: (i) the effect of interior bezels on the subjects' gaze decreases after a short period of time. While eye movement patterns of images presented on LHRDs vary (especially for the images presented at the beginning of the experiments), eye movement patterns of images presented on a single-screen display without interior bezels focus on the most salient locations in the images. (ii) The interior bezels of tiled displays affect the results of saliency prediction algorithms. This shows that the investigated predictive models don't work well for eye movements prediction on tiled LHRD. One explanation is that some feature channels used in these

model are less important for the prediction, and hence predictive models should have inhibition mechanism (or scales) for the computation of features that are less important in some scenarios.

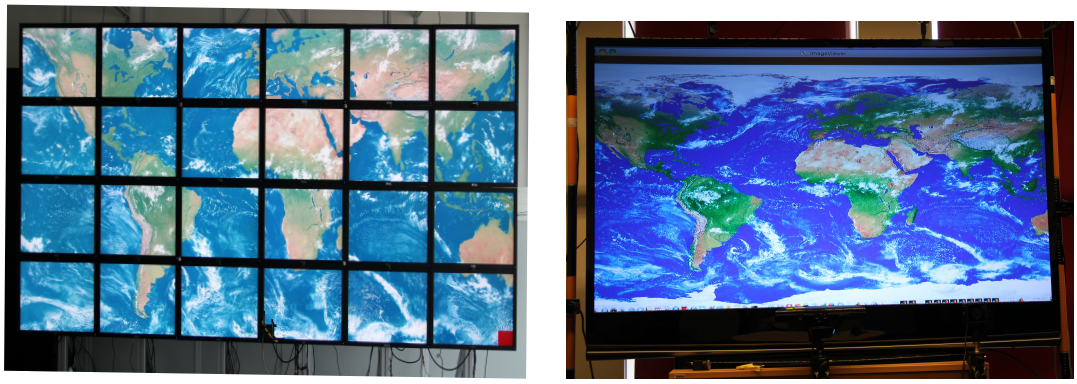
6.1 Introduction

Large high-resolution displays (LHRD) are widely used in various application domains, such as automotive design, geospatial imaging, scientific visualization, telepresence, and astronomy [146]. With a larger capacity for visual information, these display environments provide users with a significantly larger display surface area compared with desktop displays. Combined with high pixel density, this facilitates collaborative interaction among multiple persons [29] and prompts physical navigation, thus, improving performance in navigation tasks [18].

However, it has been recognized that tiled-display systems bring a new set of interaction challenges, from fundamental selection, manipulation to task management [146]. Previous work identified interior bezels as a possible limitation of LHRDs based on tiled LCD panels. Interior bezels cause visual discontinuities of displayed images as well as cursor trajectory. Several studies investigated the effects of interior bezels on tiled displays [25, 164, 192, 202, 18, 81], but the effects of interior bezels on human eye movements during free viewing, and on saliency algorithms predictions, have not been studied yet.

Information about the user's gaze and visual attention can improve the interaction with LHRDs. Existing computational models of the visual attention are able to predict saliency maps based on image properties that closely match the experimentally measurable maps of eye movements and fixation periods. Such saliency maps reflect bottom-up attentional processes, in other words, the attraction of attention by external cues. Predictions of such models are desirable in many HCI application scenarios such as the design of web pages, adaptive user interfaces, interactive visualization, video compression, or attention management systems [165, 91].

In natural images, including both landscapes and man-made environments, vertical and horizontal orientations are more frequent than diagonals [166, 136]. Considering that most saliency algorithms work on 2D image features (e.g., orientation features) and that vertical and horizontal orientations have a strong influence on perception in the human visual system, it is necessary to characterize the effects of the interior bezels on human eye movement and saliency prediction algorithms. Primary results presented in poster abstract [14] suggested that further investigation on the bezels effects on the subjects eye movements for free-viewing tasks with images presented in different orders and on saliency algorithm predictions is necessary.



(a) (b)

Figure 6.1: a) LHRD with 24 LCD panels. b) Single-screen DLP TV with 67 inch display diagonal.

In this chapter, we present the results of two studies to quantify the effects of interior bezels in LHRDs on human eye movements and image saliency algorithms. In the first experiment, we recorded the subjects' center of gaze and eye movement when they were looking at natural images on a single-screen display and a tiled LHRD (see Figure 6.1). In the second experiment, we compared the performance of different state-of-the-art saliency prediction algorithms with and without the presence of interior bezels.

6.2 Influence of Interior Bezels on Human Eye Movements

We conducted a user study in order to explore the effects of interior bezels on human eye movements when performing a free viewing task with a LHRD. We compared the results with the patterns of eye movements of single-screen display.

We have studied the following two hypotheses:

H1: The subjects' eye movement patterns are affected by the type of the display.

H2: The subjects will get used to the interior bezels after a short time, so the effects of interior bezels will vanish over time.

6.2.1 Material and Methods

6.2.1.1 Experiment Setup

In this section, we explain the details of our experiments. One independent variable in the two experiments is the type of display. We used the following display systems: (i) A tiled LHRD comprising 24 LCD panels with a resolution of 1900 $\times$ 1200 pixels each, with a combined resolution of approximately 55 megapixles. The width of a pair of interior bezels from two neighboring panels is 4.8 cm. The LHRD wall has dimensions of 378 cm (W) $\times$ 164 cm (H) and (ii) a DLP TV with a 67 inch screen diagonal and Full HD resolution of 1920 $\times$ 1080 pixels. The two displays are depicted in Figure 6.1.

6.2.1.2 Measuring Gaze Locations

We used an iView X HED 4 Eye Tracking System (SMI) to record eye position. The system reports gaze positions with a sampling rate of 50 Hz and a reported accuracy of 0.5°-1°. We used the default lens ($f = 3.6$ mm) for the scene camera which provides a viewing angles of 31° horizontally and 22° vertically. The eye tracker scene camera has a resolution 752 $\times$ 480 pixels.

6.2.1.3 Visual Stimulus

We selected 20 images from the *Microsoft Salient Object Dataset* [119] and from the *York University Eye Fixation Dataset* [32]. The images resolution was different in the range of 700 - 1200 $\times$ 600 - 800 pixels. Each image was presented to eight subjects with the two types of displays described above. We employed Vrui toolkit[1] to present the stimulus on the displays. Each image was shown for ten seconds.

6.2.1.4 Participants

Eight participants took part in this study (one female, seven male, 18-40 years). The participants were students with normal vision and no history of neurological problems. All of them were daily computer users and two of them had work experience with tiled-monitor displays. They participated in the main eye tracking experiment.

6.2.1.5 Eye Tracking Experiment

For our study, we used a within-subjects design. Each subject carried out a 9-point calibration procedure before the start of the experiment. We split the subjects into two groups. The stimuli presented to the second group was

[1] `http://idav.ucdavis.edu/okreylos/ResDev/Vrui/`

similar to what presented to the first group but with different presentations order. Subjects sat at a distance of approximately 460 cm from the tiled display wall and 160 cm from the single DLP display. This corresponds to a distance where the subjects could comfortably view the entire display, using a chin rest to fix their head position. Each subject performed two experiments, with the presentation order of the experiments counter-balanced across the participants. Throughout the experiments, the subjects' right eye position was recorded.

Based on the eye-tracking data, we generated saliency maps of the locations fixated by the subjects for each frame. We filtered the subjects's fixation locations using a Gaussian kernel to obtain a continuous saliency map.

6.2.1.6 Bezels Features in Luminance Images

The color images captured by the eye tracker's scene camera were first transformed into gray-scale images. Then, each gray-scale image was linearly decomposed into a set of edge-feature responses to Gabor filters. We used Gabor filters with only vertical and horizontal orientations, $\theta = \{0°, 90°\}$, with only one frequency and two spatial phases. Within each image we subtracted the mean from the filter responses to each orientation and normalized the responses to the interval $[-1, 1]$. These filter responses were used later to characterize the patches in the luminance images at the center of gaze (see Subsection 6.2.2.3).

6.2.2 Results

6.2.2.1 Participants Eye Movement Behaviors on the LHRD vs. on the Single Display

We first compared the distribution of eye movements of individual subjects, when they performed the free viewing task on images presented on the LHRD, to that presented on a single display. We performed the two-sample t-test between the distribution of eye movements on the LHRD and the single DLP display, across all images and subjects, using a significance level of 0.05. We observed no significant difference between eye movements and displays types. We also carried out a qualitative analysis. We found that the distribution of eye movements of users on the LHRD doesn't differ from that on the single display. It is important to note that the eye movement patterns for the first images presented on the LHRD differ from the ones on the single display (see Figure 6.2). This suggests that further investigation on the subjects' eye movements for free-viewing tasks with images presented in different orders is necessary.

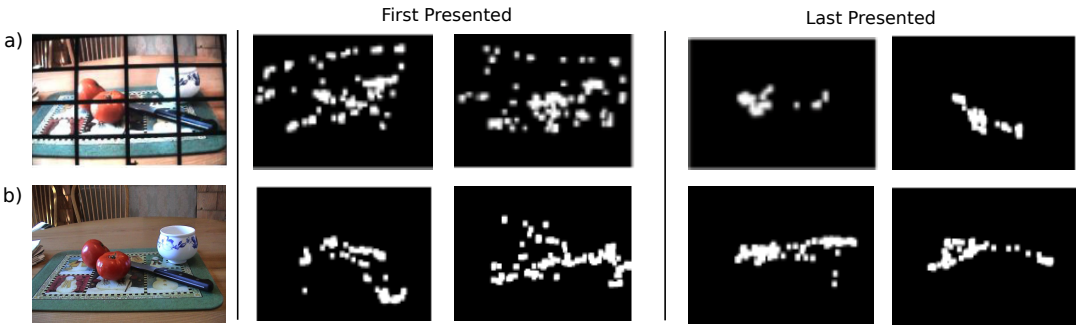


Figure 6.2: The distribution of eye movements of individual subjects, on images presented in different presentation orders on **a)** the tiled LHRD and **b)** the single DLP display.

6.2.2.2 Participants Eye Movement Behaviors on Images Presented in Different Time Slide (First vs. Last Presented)

We also compared the distribution of eye movements of individual subjects with different presentation orders. Figure 6.3 depicts an example of eye movement patterns of individual subjects, with images presented with different presentation orders on the LHRD. We can see that the distribution of eye movements for images presented in varying presentation order is different. While the eye movement patterns of images presented to the subjects at the beginning of the experiment are distributed across the whole scene, the eye movement patterns of images presented after a while are focused on the most salient locations in the image.

A two-sample t-test between the distribution of eye movements of the first five images presented to the subjects and final 15 images shows a significant difference ($p < 0.05$).

Figure 6.2 shows an example of eye movement patterns of individual subjects, on images presented with different presentation order on the LHRD and on the single DLP display. We can see that, the distribution of eye movements of the first image presented on the LHRD differ than on the first image presented on the single display. While the eye movement patterns of the image from LHRD are variable (Figure 6.2(a), First Presented) the corresponding eye movements patterns of the image presented on single DLP display are focused on the most salient locations in the images (Figure 6.2(b) First Presented). But when the same images presented to other subjects in different time slide(i.e. Last Presented), the eye movement patterns are focused on the most salient locations in the image, for both displays (i.e. LHRD and on the single DLP display, Figure 6.2(a,b) Last Presented).

We repeated the two-sample t-test between the distribution of eye movements on the first five images presented to the subjects on the LHRD and on

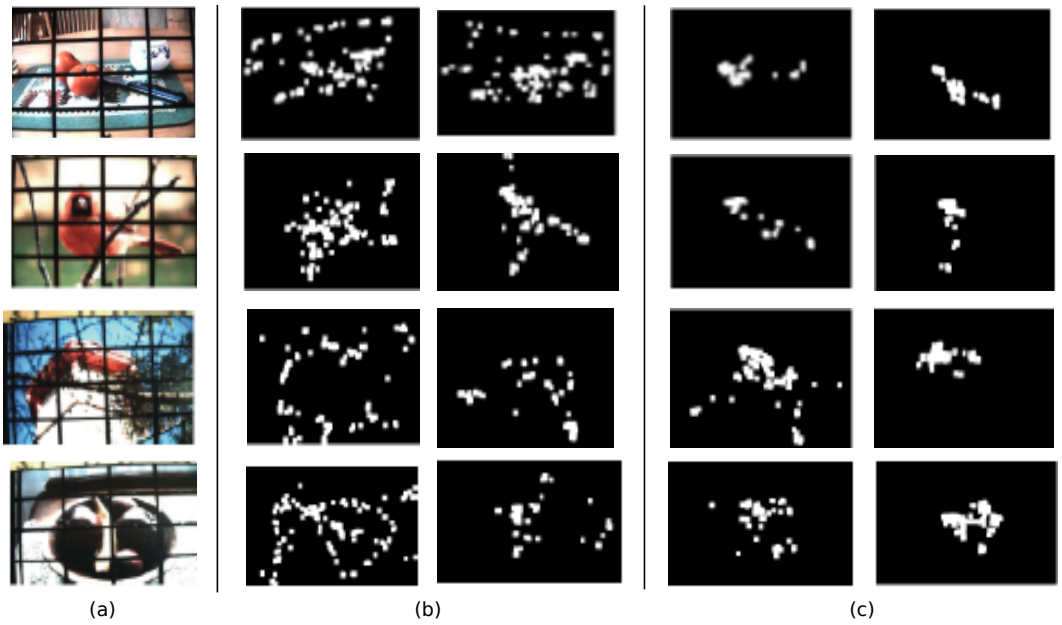


Figure 6.3: The distribution of eye movements of individual subjects, on images presented in different presentation orders on the tiled LHRD. **a)** Examples Images presented on LHRD. **b)** The eye movement patterns when the images presented in the beginning of the experiments. **c)** The eye movement patterns when the images presented after a short time from the beginning of the experiments.

the single DLP display. It reveals a significant difference ($p < 0.05$). But there was no significant difference between the eye movement patterns on both displays over the last 15 images.

6.2.2.3 Bezels Features at the Center of Gaze

We examined the presence of the interior horizontal and vertical bezels around gaze locations. We first extracted the responses to Gabor filters for each frame using the method presented in subsection 6.2.1.6. We then compiled histograms from the responses to these Gabor filters form the pixels around the gaze point in each condition. Figure 6.4 show these histograms and reveal that the probability of horizontal or vertical edges being present, (i.e. non-zero filter responses), around the center of gaze are more frequent in the first five images compared to the final 15 images. This hold true for the LHRD (red bars in Figure 6.4).

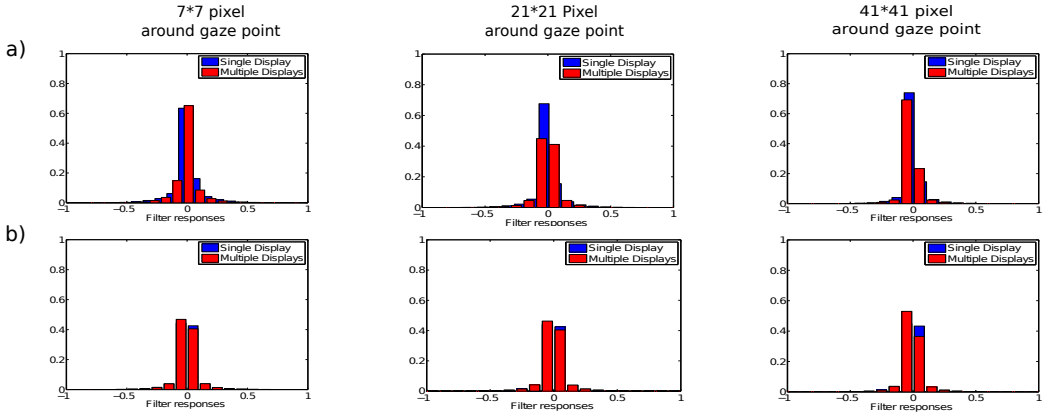


Figure 6.4: Normalized histograms for the combined Gabor features (vertical and horizontal directions) at the center of gaze in the two experimental conditions, with different neighborhoods. **a)** The first five images. **b)** The last 15 images.

6.3 Influence of Interior Bezels on Visual Saliency Models Predictions

The objective of this analysis is to investigate how interior bezels affect saliency algorithms performances. We compare how well visual saliency models perform when we have the bezels of tiled displays in the image compared to when they are not there. Given that most saliency algorithms work on the 2D image features (including orientation features), we hypothesize that interior bezels are detrimental to the saliency algorithm predictions. To be specific, the major hypotheses in this study is:

H3: The presence of interior bezels affects saliency algorithm performance on tiled displays.

6.3.1 Computational Visual Saliency Models

To assess the performance of saliency algorithms under the presence of LHRD interior bezels, we selected three visual attention models that differ in terms of their mechanism of determining saliency.

Itti and Koch The Itti and Koch model was inspired by biological concepts from cognitive science and based on a bottom-up computational model [98]. This model has been the basis for later models and is a standard benchmark for comparison (See Sec. 2.3.3 for more detail).

Graph-Based Visual Saliency (GBVS) This model is based on a probabilistic framework in which a graph denotes the conditional independence structure between random variables. This model treats eye movements as a time series. Since there are hidden variables influencing the generation of eye movements, a Hidden Markov Models (HMM) approach was been incorporated [85] (See Sec. 2.3.3 for more detail).

Torralba Saliency (T-Saliency) This model combines sensory evidence with prior constraints. Prior knowledge (e.g., scene context or gist) and sensory information (e.g., target features) are combined according to Bayes' rule. The proposed architecture for attention guidance consists of three parallel modules extracting different information: bottom-up saliency, object-centered features, and contextual modulation of attention [195] (See Sec. 2.3.3 for more detail).

6.3.2 Error Measures

To better understand the relationship between a viewer's fixation locations and the predictions of the saliency models, we have to evaluate it quantitatively by comparing it with eye movement data. We used four performance measures that are widely used in the state of the art of visual attention literature, to evaluate the performance saliency models. Because the evaluation measures for attention modeling can be classified into point-based and region-based, we used four performance measures to deal with this perspective (for more details about these measures, see Sec. 2.3.4).

6.3.3 Results

6.3.3.1 Comparing Visual Saliency Models Predictions

To evaluate the saliency models performances, we have to compare it with eye movement data. In this analysis, we use the same eye movement data described in the previous section (see section 4.2 for more detail).

We compared how well the classic Itti and Koch, GBVS and Torralba saliency models perform when we have the bezels of tiled displays in the image compared to when they are not there. We used subjects gaze locations to test and validate the predictions of attention locations by each of the three models. Using each model, we generated a saliency map for each image in our test images. Figure 6.5 (a) shows an image presented on the tiled LHRD and the saliency maps generated by the Itti and Koch, GBVS, and Torralba saliency models. Figure 6.5 (b) shows an image presented on the single-panel DLP display and the resulting saliency maps. We can observe that the bezels of the LHRD affect the predictions of saliency algorithms, and this influence differs

according to the contrast properties of the images. It is important to note that this influence was less pronounced with the GBVS model predictor.

We carried out statistical tests using a two-sample t-test between the three model predictions and the two types of displays. The statistical test revealed a significant difference ($p < 0.05$).

We measured the performance of the saliency models using different methods (see Section 4.2.7). We computed the distance between human fixation maps and the saliency maps generated by Itti and Koch, GBVS and Torralba using KL divergence and the area under ROC curve. The results of the performance of different models averaged over all users and all images are shown in Figure 6.6(a and b). We can see that the performance of visual saliency models is better for the images on the single-panel DLP display compared with the LHRD. Also we can see that, GBVS performs better than Itti and Koch and Torralba with both display settings. The models performance results using linear correlation coefficient and mean square error methods are presented in Appendix C.3.

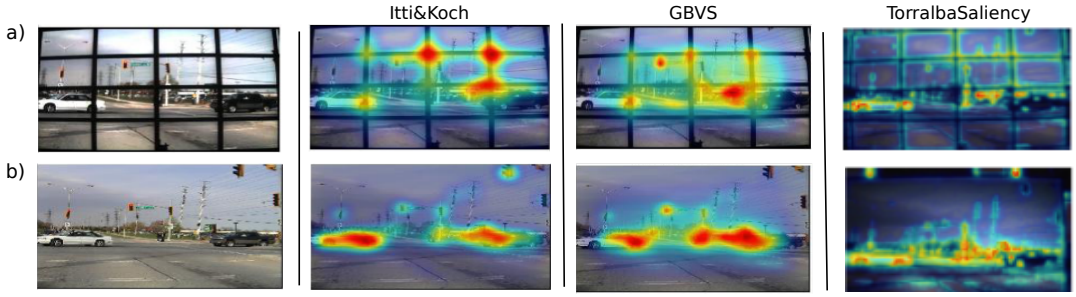


Figure 6.5: Sample saliency heat maps predicted by Itti and Koch, GBVS and Torralba saliency models, superimposed on images from the tiled display wall and the single display. **a)** Image presented on multi display walls along with the saliency maps generated by Itti and Koch, GBVS and Torralba saliency models. **b)** The original image presented on single display along with the saliency maps generated by Itti and Koch, GBVS and Torralba saliency models.

6.4 Conclusions

We have investigated the effects of tiled display (interior horizontal and vertical) bezels on human eye movements and saliency prediction algorithms. We conducted two experiments for two types of display systems (i.e., single-panel display and multi-tile LHRD). We first examined human eye movement behavior for the two types of displays. We then examined saliency algorithm performance with and without the presence of interior bezels.

We conclude the first study with our results with respect to our two hypotheses:

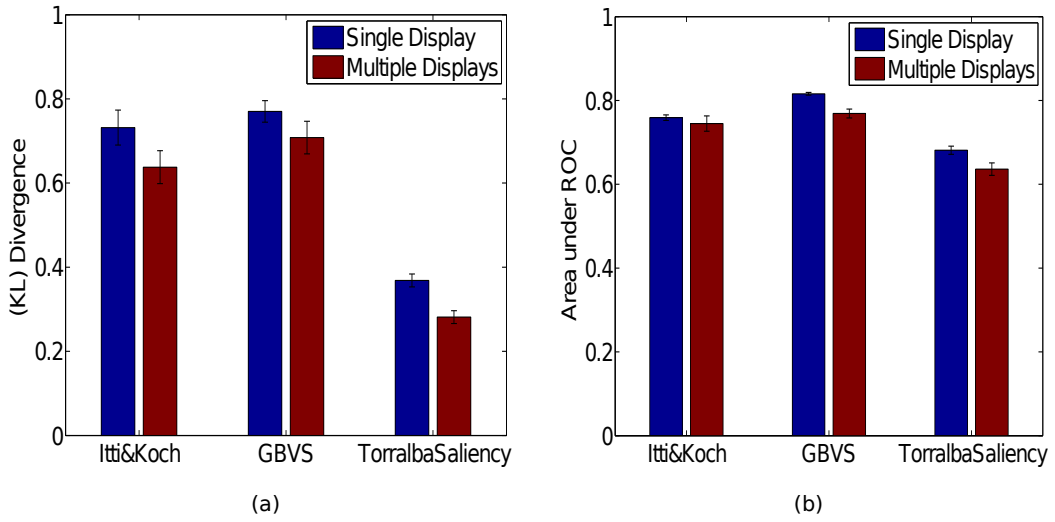


Figure 6.6: The performance of saliency models, averaged over all subjects. **a)** The KL divergence describing the performance of Itti and Koch, GBVS and Torralba saliency models in two scenarios (single DLP display vs. multiple LHRD). **b)** The Area under ROC describing the performance of Itti and Koch, GBVS and Torralba saliency models, in two scenarios (single DLP display vs. multiple LHRD).

H1: The subjects' eye movement patterns are affected by the type of the display This hypothesis was not confirmed. Results showed that there is no significant difference between the distribution of eye movements on the LHRD and on the single display.

H2: The subjects will get used to the interior bezels after a short time, so the effects of interior bezels will vanish over time This hypothesis was confirmed. Results showed that the difference of eye movement patterns for images presented on the LHRD for varying presentation orders is significant. While the eye movement patterns of images presented to the viewers at the beginning of the experiment were distributed across the whole scene, the eye movement patterns of images presented later were focused on the most salient locations in the image.

In the second study, we hypothesis that:

H3: The presence of interior bezels affects saliency algorithm performance on tiled displays. This hypothesis was confirmed. Our experiments indicate that the presence of interior bezels affects saliency algorithm performance on tiled displays. The results show a significant difference between the

model predictions and the display system types, and this influence differs according to the saliency algorithm we used and the contrast properties of the images presented on the tiled displays.

Also, the results show that the GBVS performs better for predicting attention location compared with Itti and Koch and Torralba. Since the visual properties present in an image generate a visual saliency map that explicitly marks regions that are different from their surround based on color, intensity, contrast, and edge orientation, we can assume that models which ignore edge feature in vertical and horizontal orientation will perform better with multi-tiled LHRDs. Therefore, we suggest to use saliency-prediction methods directly on the original (i.e., bezel-free) image data.

The limitations of this work are, here we considered static natural scenes and fixed head positions, our results have shown that there is no significant difference between eye movement patterns and display types. We suggest more investigation with dynamic scenes and free head movements for future work. Also by combining mobile eye tracking with a head tracking system we suggest repeating the experiments with dynamic 3D scenes.

Chapter 7

Predicting Eye Movements Strategies with Inverse Reinforcement Learning

In this Chapter we return to the distinction between descriptive and normative models that we introduced in the theoretical Chapter 3. We recapitulate IRL and devise a simple experimental setting so that it becomes applicable to predicting eye movements. We have examined two different inverse reinforcement learning algorithms. The presented approach used information about the possible eye movement positions. We showed that it is possible to automatically extract reward function based on effective features from user eye movement behaviors using IRL. The learned reward function was able to extract expert behavior information that fulfill to predict eye movements. Thus, this study serves as a proof-of-concepts for using IRL in eye movement predictions, and in human behavior modeling in general.

This chapter is organized as follows: First, we introduce the motivation of using IRL for modeling eye movement behaviors. Then, we describe the material and methods including the eye tracking experiment and the features we extracted from our dataset (Sec. 7.2 and 7.3). Then, we present the results of our analysis, we first compared the reward feature weights for each of the two methods, and then compared the performances of the two algorithms to the user behaviors which runs separately on each eye movements trajectory (Sec. 7.4).

The results of this chapter have previously appeared as conference publication in [134].

7.1 Introduction

Truly gaze represents one of the essential cues, which is important to understand the behaviors that are exhibited during human-computer interaction. Many researchers have considered the problem of predicting human eye movements. Some of them describe eye movements using visual saliency perspective [98, 121]. Other models eye movements as the future information gain using the reward driving approach [168, 140]. This is an important information to be considered in the interaction with the tiled Large High-Resolution Displays (LHRD) [82, 8], and desirable in many application scenarios such as commercials adaptive user interfaces, interactive visualization, or attention management systems, to adapt visual interfaces, or to place important information.

In a Markov Decision Process (MDP) framework, an agent is represented by its policy π, which is a probabilistic action selection modeled as a conditional probability $P(A = a \mid S = s)$ to select the action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. The agent can select an action and observe the state change and may receive a reward. A policy determines which actions to take in specific states in order to achieve a goal. One can find an optimal policy using optimal control techniques [191]. In an MDP, we want to find a policy that maximizes the expected reward. Thus the reward encodes the goal of the task. The accomplishment of a goal by an agent usually can be described by a sequence of actions that agent has selected by interacting with the environment. The selection of actions can also be depend on the decisions and actions of others especially when the possibility of communication with other agents are available. Using supervised learning we can learn a policy directly from demonstrations [15], such approaches usually have limited generalization abilities because they are limited to the demonstrated scenarios. As they do not consider the underlying dynamics, they cannot be applied in a task with changing dynamics. In the interaction with the tiled Large High-Resolution Displays (LHRD), the eye movements change as the dynamics of the environment changes.

Given an exact model of the environment and the measurement of the agent's behavior over time. Instead of predefining the reward function, we seek to identify it from human eye movements behavior. Finding a reward function by learning it from an expert demonstration is referred to Inverse Reinforcement Learning (IRL) or inverse optimal control [145]. Ng et al. [145] argue that the reward function from RL must be considered as an unknown when examining the animal and human behavior. They present methods to solve the problem of the inverse reinforcement learning (IRL). This reward function, which cannot be observed directly, can be considered as part of the internal state of a user, similar to the state of the attentional system, or the current goal state. Inverse optimal control applied to different problems such as modeling goal-directed trajectories of pedestrians [220], helicopter control [2], robot navigation across

Figure 7.1: A user wearing an eye tracker viewing an image on a 24-panel tiled display wall.

different environments [108], parking lot navigation [3], routing preferences of drivers [219], learning strategies in table tennis [138] and user simulation in spoken dialog management systems [39].

Figure 7.2 illustrates the considered scenario we used in our lab (Figure 7.1): A user is viewing an image on tiled display walls. At time point t, he decides to look at tiled display Nr.9, but at time point t+1, he decides to look at tiled display Nr.12. In this scenario, various sensors can detect the gaze direction of users in a room. In the interaction with the tiled Large High-Resolution Displays even if the dynamics of the environment not changes, the eye movements behavior changes(Figure 7.2).

In this work, we learn the reward function from demonstrated scenarios and use this reward function to explain the observed behavior. Thus, here we do not introduce new IRL methods for solving IRL problem, but we aim to use available methods on modeling human eye movements behavior during the interaction with tiled LHRD. Since modeling the dynamics of eye movements is highly challenging, We rely on a maximum entropy IRL formulation [219] and feature construction Inverse Reinforcement Learning method [116] to model the distribution of all possible eye movement trajectories, across all the images in the dataset.

7.2 Modeling Human Eye Movements Strategies

To use IRL, we need to represent the problem as a Markov decision problem (MDP). A Markov Decision Process is described by a tuple $M = (S, A, T, \gamma, R)$ (See Sec.2.2.2.6 for more details). In an IRL setting, the algorithm is presented with $M \setminus R$, together with expert demonstrations $D = \{\zeta_1, ..., \zeta_N\}$, where $\zeta_i = \{(s_{i,0}, a_{i,0}), ..., (s_{i,T}, a_{i,T})\}$ (i.e., its trajectory or path, ζ, of states si and actions ai). In combination with features of the form $f : S \longrightarrow \mathbb{R}$ that can be used to represent the unknown reward R.

We represent the reward function by a linear combination of m feature

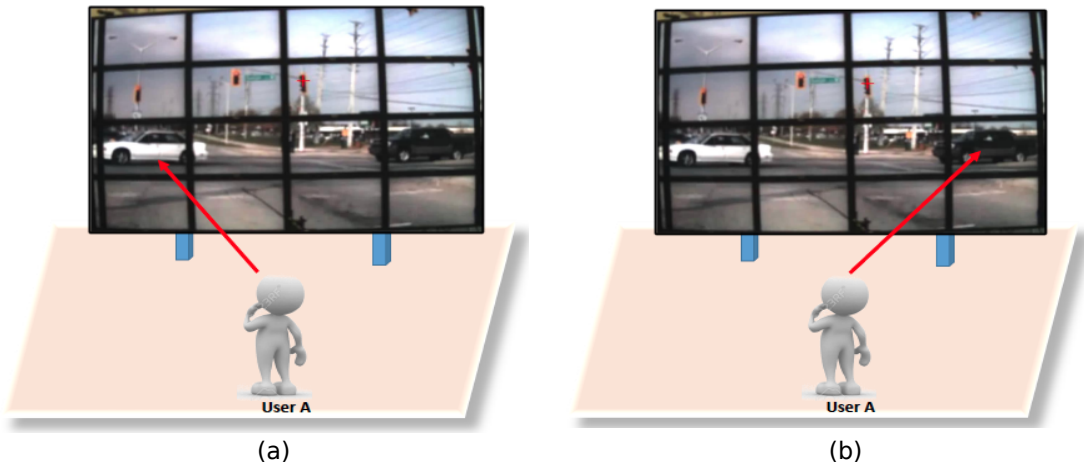


Figure 7.2: Considered scenario: A user viewing an image on tiled display walls consisting of 24-panel LCD. **a)** At time point t, he has decide to look at tiled display s9. **b)** but at time point t+1, he has decide to look at tiled display s12.

functions f_i with weights θi, which maps the features of each state, $f_{s_j} \in \mathbb{R}^m$, to a state reward value. Hence, the reward function is defined by:

$$reward(s,a) = \sum_{i=1}^{m} \theta_i^\top f_i(s,a) = \theta^\top f(s,a),$$

where $\theta \in \mathbb{R}^k$ and $f(s,a) \in \mathbb{R}^k$. The features functions f_i are bounded and mapped from $S \times A$ into R.

In this work, we construct the reward functions from human eye movements behavior and use this reward function to predict eye movements strategies on the tiled LHRD.

7.2.1 Learning the Reward Function

Going beyond the classical RL setting. Ng et al. [145] proposed algorithms to solve the problem of the inverse reinforcement learning (IRL), i.e., of extracting a reward function given observed optimal behavior. Abbeel & Ng [4] introduce a novel approach based on Inverse Reinforcement Learning (IRL). They suggest a procedure of coordinating feature expectations between an observed behavior and a learner's behavior (See Equation 1). In there work, they showed that this representation is enough to achieve the same performance as the agent were solving an MDP with a reward function linear in those features.

Many researches provided further development in order to improve to the original algorithms suggested by [145], For example [117, 219, 157, 116].

To compute the reward functions, we used two different methods. The first method based on the principle of maximum entropy [219], while the second algorithm is the feature construction for Inverse Reinforcement Learning [116].

Maximum Entropy Inverse Reinforcement Learning Method (Max Entropy IRL): The Maximum Entropy Inverse Reinforcement method [219] reduces learning to the problem of recovering a reward function; that makes the behavior influenced by a near-optimal policy that closely imitate demonstrated behavior. It is a probabilistic approach based on the principle of maximum entropy. A maximum entropy IRL formulation finds a distribution over all trajectories. With respect to the eye movements behavior, it model the distribution over all possible eye movement paths of length T starting from state s for a given image (See Sec.3.5.5for more detail).

Feature Construction Inverse Reinforcement Learning Method (FIRL): The feature construction for Inverse Reinforcement Learning method [116] builds reward features from a large collection of essential features, by building logical conjunctions of those component features that are relevant to the example policy. The algorithm repetitively builds both the features and the reward function. Each iteration consists of two step formulation: an optimization step computes a reward function $R^{(i)}$ of the i^{th} iteration using the current set of features beginning with an empty feature set, and a fitting step determines a new set of features (See Sec.3.5.6 for more detail).

7.2.2 Computational Model for Representing Eye Movements Strategies

The key concepts we build upon for modeling eye movements are Markov decision processes (MDPs). Our framework is based on learning the reward function that can explain observed eye movements behavior via the corresponding optimal policy π^* [145]. By employing policies that operate over long time horizons, we learn the reward function directly from large amounts of human eye movement data. We cast the problem as Inverse Reinforcement Learning (IRL), where we aim to construct the reward function that stimulate human eye movements behavior recorded from human subjects performing a free viewing task. Our learned model can imitate useful eye movement's strategies on LHRD. We now need to identify the states, actions and reward features of our framework.

States In MDPs state $St = i \in S = [1, 2, ..., N]$ represents the state of the world at time t. In this paper, it represents the state of the tiled LHRD at time t and indicates that the target is located at tiled display i. Concretely, we present each image on tiled display walls consisting of 16-panel LCD (see Figure 7.3), and assume that the location of the important object's center is inside one of those tiled locations. For this work, we chose to tile the frame from eye tracker scene camera with a 4×4 LCD grid, meaning the eye movements could be located at any of 16 locations, and each location represent one of the tiled display.

Figure 7.3: A 4×4 tiled LCD grid used to present each image, forming the basis of the hypotheses that are entertained about the possible eye movements in the tiled LCD.

Actions action $At = i \in A = [1, 2, ..., N]$, is random variable that represents the action taken by the agent at time t, where actions model eye movements on tiled displays. In this work, we consider action space equals to state space. We then encode all scanpaths in this discrete (state, action) space.

Reward features In order to determine the reward function, we suppose that the reward function is given by a linear combination of observable features. We choose the features as a combination of the state information.

- We used information of the intensity, orientation and color channels as computed by Itti and Koch's saliency method [98].
- We also include histograms of color features (Ho- Color): which represents the probabilities of the red, green and blue channels as features. This probability computed using color histograms of the image filtered with a median filter at six different scales.

Figure 7.4: **a)** Eye movements trajectories for three users viewing an image on LHRD. **b)** Eye movements path predicted using Itti model.

- Histogram of Oriented Gradients (HoG) features [48], we extracted 7 HoG descriptors with different grid arrangements at each tiled and concatenate them a with the output of our feature detector.
- We used the local energy of the steerable pyramid subbands in four orientations and three scales [179].
- We also included a distance to the center as a prior feature which denotes the distance to the center for each pixel in the image, we used this feature because when humans take pictures, they spontaneously place the area of interest near the center of the image.

All features are scaled to lie in an interval of [0 1].

7.3 Experiments and Evaluations

7.3.1 Eye Tracking Experiments

In order to extract basic eye movements strategic elements, we recorded eye movements of users participated with free viewing task experiments on LHRD. We used tiled display walls consisting of 24-panel LCD. The display surface is subdivided into 24-panels by three horizontal and five vertical black plastic bezels. Each interior bezel is 4.8 cm wide and represents the plastic bezels of physical tiled-monitor displays. In this study, we consider only 16- Panel LCD because of the eye tracker scene camera limitation. For this study, we used the eye-tracking experiment described in Chapter 6 (see Sec. 6.2.1 for more details), but here we consider only the eye movements data with tiled LHRD.

7.3.1.1 Visual Stimulus

We used the visual stimulus described in Sec.6.2.1.3. They were 20 images, each was presented to eight subjects. We employed Vrui toolkit[1] to present the stimulus on the LHRD displays. Each image was shown for ten seconds.

7.3.1.2 Experiment Design

Each subject carried out a 9-point calibration procedure before the start of the experiment. Subjects sat at a distance of approximately 460 cm from the tiled display wall. This corresponds to a distance where the subjects could comfortably view the entire display, using a chin rest to fix their head position. Based on eye tracking data, we first generated scanpaths "gaze locations" on each image, for each user. Toward obtaining the same number of gaze locations on each image with all users, we selected 300 gaze locations on each image, after eliminating the out of range gaze locations. Then we generated eye movements trajectories in term of (state, action) for each image, where we cover each image frame by 4×4 grid for the state space(i.e. 4×4 tiled displays). We then encode all scanpaths in this discrete (state, action) space.

7.3.2 Performance Evaluation

We evaluated the performance of each algorithm using the Expected Value Difference (EVD) score, which measures of how optimal the learned policy π compare to the expert optimal policies π^*. Using this score, we compute the optimal policy under each learned reward, and subtract this value from the expected sum under the true policy. Where π^* is the agent's ground truth policy function, and π is the policy with the learned reward function. The expected value difference estimates the difference in the performance between the agent's optimal policy and the policy induced by the learned reward function. We first estimated the expected value difference between the true and learned policy functions of each individual trajectory and calculated the expected average difference of all trajectories in the behavior dataset.

7.4 Result

We carried out a leave-one-out testing scheme to evaluate the reward functions. We computed the reward feature weights for each of the two methods. We also evaluated the performance of each method in predicting eye movement scanpaths compares to the user demonstrated behaviors.

[1] `http://idav.ucdavis.edu/okreylos/ResDev/Vrui/`

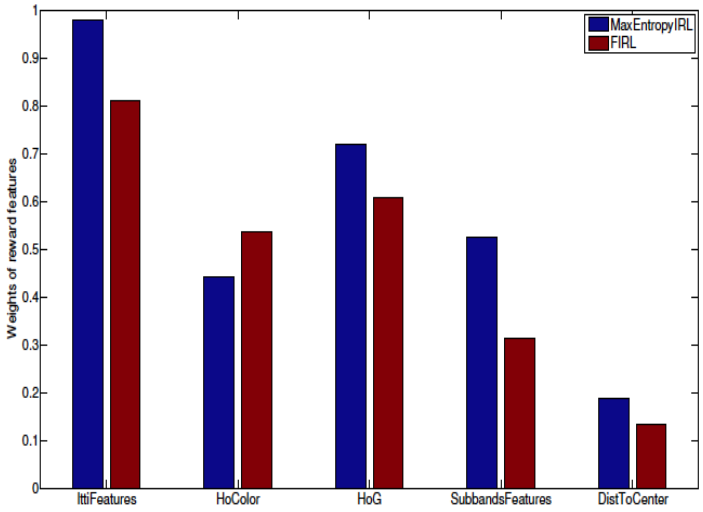


Figure 7.5: The resulting reward weights of the individual features features for Algorithm 1 (MaxEntropyIRL) and Algorithm 2 (FIRL), respectively.

7.4.1 Individual Reward Features

Figure 7.5 shows the calculated reward weights for the individual features. We compared the learned weights for the individual features of the reconstructed reward functions obtained by both algorithms. We found that, the two algorithms revealed similar weights for the most important features. While the Itti and HoG features show highest positive reward weights for both algorithms, the distance to center feature had only a small positive rewards weight and almost no influence. It is important to note that, the reward weights of the histograms of color features with FIRL algorithm shows higher positive reward weights compare to Max entropy IRL algorithm.

7.4.2 Comparison of the tested IRL methods

We compared the performances of the maximum entropy Inverse Reinforcement (MaxEntropyIRL) and the feature construction Inverse Reinforcement Learning (FIRL) algorithms to the expert behaviors which runs separately on each eye movements trajectory. Fig. 7.6 displayed the performance scores for the learning reward functions from the demonstrated data over 20 images instances. We can see that, when the size of the behavior data is small, the clustering performances of both MaxEntropyIRL and FIRL were not performing well. However, as we increased the size of the data, both MaxEntropyIRL and FIRL achieved better EVD results. In addition, we can see that the maximum entropy IRL learned a reward function that more precisely imitates the policy of the expert

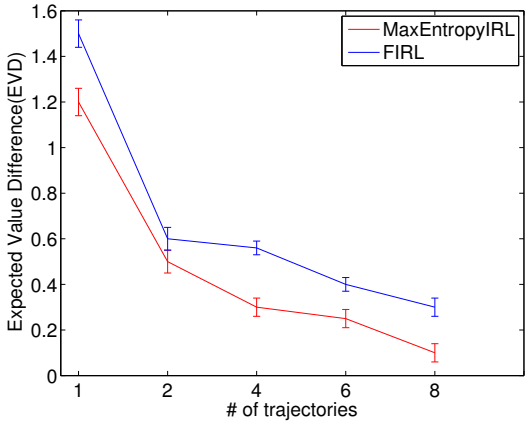


Figure 7.6: Evaluation of both methods with human demonstrations. Maximum entropy IRL learned a reward function that more precisely imitates the policy of the expert behavior.

behavior.

7.5 Conclusion

In this work, we modeled eye movements on tiled LHRD as an MDP. We collected eye movements of users participated with free viewing task experiments on LHRD. We have examined two different inverse reinforcement learning algorithms. The presented approach used information about the possible eye movement positions. We showed that it is possible to reconstruct reward function based on effective features from user eye movement behaviors using IRL. The reward function was able to extract expert behavior information that fulfill to predict eye movements. The findings of the IRL methods we used support each other and demonstrate that they are all suitable for the challenging task context presented in this paper. This is important information for estimating the internal states of users, and desirable in many application scenarios such as commercials adaptive user interfaces, interactive visualization, or attention management systems, to adapt visual interfaces, or to place important information.

In the current model, we models eye movements on tiled LHRD as an MDP, we assumed that the agent has complete information about the environment. Also, we did not account for noisy sensory information, where POMDPs could be useful to use.

Chapter 8

Conclusions and Future Work

This thesis addressed the problem of predicting human gaze behavior in smart environments. For that, it approached two questions: (1) how different visual features are relevant for predicting human eye movements in different behavioral context in smart environments; and (2) how humans might select the next gaze location. Previous research has suggested that human eye movement behavior is consistent with decision-making mechanisms for fixation selection that attempt to maximize reward [168, 140].

This thesis investigates these problems by:

1. It uses systematic machine-learning approach, where user profiles for eye movements are learned from data in different context, and determining by combinatorial exploration which features are relevant for behavioral context.

2. The thesis proposes the modeling of eye movements using decision-making mechanisms. For that, our approach formulated eye movements as a Markov Decision Process (MDP) problem, with the use of Inverse Reinforcement Learning (IRL) to infer the reward function.

This chapter gives a summary of the results acquired in this thesis, accompanied by some suggestions for future research.

8.1 Summary

Throughout the thesis, the predictive gaze models were evaluated by comparing it with human eye movement data in different contexts. This section presents our main results:

8.1.1 Context Dependence of Human Gaze Prediction

In our first contribution, we investigated how relevant different features are for gaze locations prediction in different behavioral contexts. We studied the dependencies between the behavior context and the visual features selection in meeting scenarios (giving a presentation vs. listening to a presentation). We used a linear SVM to find out which features are descriptive in each scenarios. The main result of this study is that gaze location prediction depends on the context. The prediction differed according to the type of features we selected. As a consequence, simple predictive "one-fits-all"-models will not work for eye movements prediction. This finding points towards including context information about the scene and situation into the computation of saliency maps as important towards developing models of eye movements.

8.1.2 Relevance of Depth Features for Gaze Prediction

Most models of bottom-up attention rely on features from 2D images. Our second contribution was an investigation of how relevant depth features for eye movements prediction.

- We first analyzed scene dependency in saliency map prediction in 2D images and depth images. From our analysis of the dependency between luminance and depth images features in natural scenes using mutual information. We found that the dependencies differed according to the type of visual environments. Moreover, we found that saliency in depth images bimodally distributed with highly salient locations corresponding to low salient 2D image locations. As a consequence, low-saliency locations in luminance images can be highly salient in depth-images. This first characterization of joint luminance and depth saliency is an important first step towards developing models of eye movements, which operate well under natural conditions such as those encountered in HCI in ubiquitous computing settings.

- We have also presented a new system, the BatGaze system, which we used to measure luminance and depth features at the center of gaze in a free-viewing scenarios. The rationale for building such a system is to inform computational vision research about these features, so that generative models of visual signals could be learned. Collecting such depth information will also help to improve models for predicting eye movements, which are currently based only on features obtained from luminance images even though the human visual system certainly uses top-down post-recognition information to guide eye movements.

- We explored as to whether depth features were relevant to eye movement prediction. We used machine learning techniques to train a bottom-up, top-down model of saliency based on 2D and depth features/cues. We found that the distribution of depth values at the center of gaze differ than from random sampling. We used machine learning techniques to train a bottom-up, top-down model of saliency based on 2D and depth features/cues. We found that the depth information improves prediction and hence it should be included in predictive models.

8.1.3 Performance of the Predictive Gaze Models in Real World Scenarios

The third contribution was investigating how existing predictive gaze models perform in real world scenarios compare to human eye movements behaviors (i.e., in the interaction scenario with tiled Large High-Resolution Displays). We conducted two studies, where we explored how good the saliency algorithms perform on two different types of wall-sized displays compare to human eye movements behaviors. We found that the presence of interior bezels affected the performance of saliency prediction algorithms. But the effect of interior bezels on the subjects' gaze decreases after a short period of time. While eye movement patterns of images presented on LHRDs vary (especially for the images presented at the beginning of the experiments), eye movement patterns of images presented on a single-screen display without interior bezels focus on the most salient locations in the images. This shows that the investigated predictive models don't work well for eye movements prediction in this scenario. This due to that some feature channels used in these model are less important for the prediction in this scenario. Therefore, predictive models should have inhibition mechanism (or scales) for the computation of features that are less important in some scenarios.

8.1.4 Predicting Eye Movements on LHRD using IRL

The fourth contribution was applying IRL on eye movement data in an interaction scenario with the tiled LHRDs. We modeled eye movements on tiled LHRD as an MDPs. We collected eye movements of users participated with free viewing task experiments on LHRD. We have examined two different inverse reinforcement learning algorithms. The presented approach used information about the possible eye movement positions. We proved that, it is plausible to extract reward function based on effective features from user eye movement behaviors using IRL. The learned reward function was able to extract expert behavior information that fulfill to predict eye movements. Thus, this study serves as a proof-of-concepts for using IRL in eye movement predictions, and

in human behavior modeling in general.

8.2 Future Work

Our approach, of using joint luminance and depth features is an important step towards developing models of eye movements, which operate well in the 3D world. So that work on extending exciting 2D saliency model to work with the 3D word is still open.

In the smart meeting room, investigation of which feature makes a good prediction is important. In the "listening scenario", in our work, we found that models trained on the color, intensity and orientation features from Itti and Koch performs better than models trained on other features. When we examined our eye movements data, we found that there were a large amount of fixations on text. We think include text detector could improve the prediction.

Another point comes to my mind finally when I was doing a presentation, where I included some text from another language (i.e. French language). One of the listeners asked me after the presentation do you speak French. The question here does include text from another language will effect the gaze of people especially if there are native in that language?.

From our investigation of using existing models of visual saliency in real-world setting. We have investigated the effects of tiled display (interior horizontal and vertical) bezels on saliency prediction algorithms and human eye movements. We found that the presence of interior bezels affects the performance of saliency prediction algorithms. It is important to make these models more robust to the real world scenarios. Also with static natural scenes and fixed head positions, our results have shown that there is no significant difference between eye movement patterns and display types. We suggest more investigation with dynamic scenes and free head movements for future work. Also by combining mobile eye tracking with a head tracking system we will repeat the experiments with dynamic 3D scenes.

This thesis modeled eye movements on tiled LHRD as an MDP, assuming the agent has perfect knowledge about its environment. In the current model, we did not account for noisy sensory information and incomplete knowledge. Where, POMDPs could be useful to use. In future work, we will model the task using an MDP, PoMDPs assume that the agent cannot completely observe its environment. Also to simplify the problem presented in this work, we predicted eye fixation locations on LHRD using IRL where each tiled display represented by one cell grid. In future work, we will predict eye fixation locations using multiple cells on each tiled display.

As applications for this work, we plan to propose a visual saliency mechanism for rendering a scene on the tiled LHRD. The idea here to use saliency

maps generated from bottom up saliency models, along with an adaptive strategy, and applied it the tiled LHRD. The proposed system can present a high-resolution rendition of the image in the most salient locations and a severely reduced the resolution in the other locations to save computing resources without the observer noticing artifacts. Furthermore, the state value function can be used so that the application on the LHRD can be adapted based on this state-value function, the policy of the user and the potential actions that can be taken.

Bibliography

[1] Til Aach, Andre Kaup, and Rudolf Mester. On texture analysis: Local energy transforms versus quadrature filters. *Signal Processing*, 45(2):173 – 181, 1995.

[2] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *Int. J. Rob. Res.*, 29(13):1608–1639, November 2010.

[3] Pieter Abbeel, Dmitri Dolgov, Andrew Ng, and Sebastian Thrun. Apprenticeship learning for motion planning, with application to parking lot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-08)*, Nice, France, September 2008. IEEE.

[4] Pieter Abbeel and Andrew Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 1–, New York, NY, USA, 2004. ACM.

[5] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Suesstrunk. Salient region detection and segmentation. In Antonios Gasteratos, Markus Vincze, and JohnK. Tsotsos, editors, *Computer Vision Systems*, volume 5008 of *Lecture Notes in Computer Science*, pages 66–75. Springer Berlin Heidelberg, 2008.

[6] Daniel Acuna and Paul R. Schrater. Structure learning in human sequential decision-making. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Leon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1–8. MIT Press, 2008.

[7] Xiaoyi Jiang Adam Hoover, Gillian Jean-Baptiste. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:673–689, 1996.

[8] B.A. Ahlborn, O. Kreylos, B. Hamann, and O. Staadt. A foveal inset for large display environments. In *Virtual Reality Conference, 2006*, pages 281–282, March 2006.

[9] N.H. Amato and C.P. Tsang. Student modelling in a keyboard scale tutoring system. In ChristopherJ. Barter and MichaelJ. Brooks, editors, *AI '88*, volume 406 of *Lecture Notes in Computer Science*, pages 108–123. Springer Berlin Heidelberg, 1990.

[10] J. R. Anderson, C. F. Boyle, A. T. Corbett, and M. W. Lewis. Cognitive modelling and intelligent tutoring. In *Artificial Intelligence 42, 7-49*, 1990.

[11] John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *PSYCHOLOGICAL REVIEW*, 111:1036–1060, 2004.

[12] John R. Anderson and Christian Lebiere. *The Atomic Components of Thought*. Lea, June 1998.

[13] David Andre and Stuart J. Russell. State abstraction for programmable reinforcement learning agents. In *In Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 119–125. AAAI Press, 2002.

[14] Anonymous. Poster abstract. 2014.

[15] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robot. Auton. Syst.*, 57(5):469–483, May 2009.

[16] Roland J. Baddeley and Benjamin W. Tatler. High frequency edges (but not contrast) predict where we fixate: A bayesian system identification analysis. *Vision Research*, 46(18):2824 – 2833, 2006.

[17] C. L. Baker, R. Saxe, and J. B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 2009.

[18] Robert Ball and Chris North. Effects of tiled high-resolution displays on basic visualization and navigation tasks. In *In Proc. Ext. Abstracts CHI 2005, ACM Press*, pages 1196–1199, 2005.

[19] Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. In Joachim Diederich, editor, *Artificial Neural Networks*, chapter Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems, pages 81–93. IEEE Press, Piscataway, NJ, USA, 1990.

[20] Sumit Basu, Tanzeem Choudhury, Brian Clarkson, and Alex (Sandy) Pentland. Learning human interactions with the influence model. Technical report, MIT MEDIA LABORATORY TECHNICAL NOTE, 2001.

[21] R. Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, 6:679Ð684, 1957.

[22] Torsten Betz, Tim C Kietzmann, and Peter König. Investigating task-dependent top-down effects on overt visual attention. *Journal of Vision*, 10:1–14, 2010.

[23] Anastasia Bezerianos. Using alternative views for layout, comparison and context switching tasks in wall displays. In *Proceedings of the 19th Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces*, OZCHI '07, pages 303–310, New York, NY, USA, 2007. ACM.

[24] Anastasia Bezerianos, Pierre Dragicevic, and Ravin Balakrishnan. Mnemonic rendering: An image-based approach for exposing hidden changes in dynamic displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 159–168, New York, NY, USA, 2006. ACM.

[25] Xiaojun Bi, Seok hyung Bae, and Ravin Balakrishnan. Effects of interior bezels of tiled-monitor large displays on visual search, tunnel steering, and target selection. In *In Proceedings of CHI*, pages 65–74, 2010.

[26] Daniel Billsus and Michael J. Pazzani. A hybrid user model for news story classification. In *Proceedings of the Seventh International Conference on User Modeling*, UM '99, pages 99–108, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc.

[27] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[28] R J R Blair. Responding to the emotions of others: dissociating forms of empathy through the study of typical and psychiatric populations. *Consciousness and cognition*, 14(4):698–718, December 2005.

[29] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):185–207, 2013.

[30] M. Bratman. *Intention, plans, and practical reason*. The David Hume series of philosophy and cognitive sciences reissues. Center for the Study of Language and Information, 1987.

[31] Cynthia Breazeal, Daphna Buchsbaum, Jesse Gray, David Gatenby, and Bruce Blumberg. Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots. *Artificial Life*, 11:1–2, 2005.

[32] Neil Bruce and John Tsotsos. Saliency based on information maximization. In Y. Weiss, B. Schölkopf, and J.C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 155–162. Tsotsos:2005.

[33] ChristopherJ.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[34] G J Burton and I R Moorhead. Color and spatial structure in natural scenes. *Applied Optics*, 26(1):157–170, 1987.

[35] Jaime Carbonell, Oren Etzioni, Yolanda Gil, Robert Joseph, Craig Knoblock, Steve Minton, and Manuela Veloso. Prodigy: An integrated architecture for planning and learning. *SIGART Bull.*, 2(4):51–55, July 1991.

[36] Laurie Carr, Marco Iacoboni, Marie-Charlotte Dubeau, John C Mazziotta, and Gian Luigi Lenzi. Neural mechanisms of empathy in humans: a relay from neural systems for imitation to limbic areas. *Proceedings of the National Academy of Sciences of the United States of America*, 100(9):5497–502, May 2003.

[37] Peter Carruthers, Peter K Smith, and University of Sheffield. Hang Seng Centre for Cognitive Studies. *Theories of theories of mind*. Cambridge Univ Press, 1996.

[38] Anthony Rocco Cassandra. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Providence, RI, USA, 1998. AAI9830418.

[39] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefevre, and Olivier Pietquin. User simulation in dialogue systems using inverse reinforcement learning. In *INTERSPEECH*, pages 1025–1028. ISCA, 2011.

[40] Li-Qun Chen, Xing Xie, Xin Fan, Wei-Ying Ma, Hong-Jiang Zhang, and He-Qin Zhou. A visual attention model for adapting images on small displays. *Multimedia Systems*, 9(4):353–364, 2003.

[41] William J. Clancey. *Knowledge-Based Tutoring: The GUIDON Program*. The MIT Press, 1987.

[42] J.J. Clark and Nicola J. Ferrier. Modal control of an attentive vision system. In *Computer Vision., Second International Conference on*, pages 514–523, Dec 1988.

[43] Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artif. Intell.*, 42(2-3):213–261, March 1990.

[44] Albert T. Corbett, John R. Anderson, and Alison T. O'Brian. The predictive validity of student modelling in the act programming tutor. In *AI-Ed 93, Edinburgh*, 1993, pp. 457-464.

[45] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[46] Thomas M. Cover and Joy A. Thomas. *Elements of information theory.* Wiley-Interscience; 2 edition (July 18, 2006), 2006.

[47] Mary Czerwinski, Desney S. Tan, and George G. Robertson. Women take a wider view. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 195–202, New York, NY, USA, 2002. ACM.

[48] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

[49] Mirella Dapretto, Mari S Davies, Jennifer H Pfeifer, Ashley a Scott, Marian Sigman, Susan Y Bookheimer, and Marco Iacoboni. Understanding emotions in others: mirror neuron dysfunction in children with autism spectrum disorders. *Nature neuroscience*, 9(1):28–30, January 2006.

[50] J.G. Daugman. Uncertainty relations for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America*, 2:1160–1169., 1985.

[51] F. de Vignemont and T. Singer. The empathic brain: How, when and why? *Trends in Cognitive Sciences*, 10(10):435–441, 2006.

[52] Daniel Dennett. Beliefs about beliefs. *Behavior Brain Sciense*, 1:568 Ð 570, 1978.

[53] Alfred Dielmann and Steve Renals. Dynamic bayesian networks for meeting structuring, 2004.

[54] Simons DJ and Rensink RA. Change blindness: past, present, and future. *Trends Cogn Sci. Jan;9(1):16-20.*, 9(1):16 –20, 2005.

[55] Marco Dorigo and Marco Colombetti. Robot shaping: Developing situated agents through learning. Technical report, 1993.

[56] A. Drake. *Observation of a Markov process through a noisy channel. PhD thesis,*. PhD thesis, Massachusetts Institute of Technology, 1962.

[57] Fantino E and Esfandiari A. Probability matching: Encouraging optimal responding in humans. *Canadian Journal of Experimental Psychology*, 56:58Ð63, 2002.

[58] Ward Edwards. Probability learning in 1000 trials. *Journal of Experimental Psychology*, 62:385Ð 394, 1961.

[59] Hillel J. Einhorn and Robin M. Hogarth. Behavioral decision theory: Processes of judgment and choice. In David E. Bell, Howard Raiffa, and Amos Tversky, editors, *Decision Making*, pages 113–146. Cambridge University Press, 1988. Cambridge Books Online.

[60] Clarence Ellis, Paulo Barthelmess, Bo Quan, and Jacques Wainer. Neem: An agent based meeting augmentation system, 2001.

[61] Clarence (Skip) Ellis and Paulo Barthelmess. The neem dream. In *Proceedings of the 2003 conference on Diversity in computing*, TAPIA '03, pages 23–29, New York, NY, USA, 2003. ACM.

[62] Sarah Favre, Hugues Salamin, Alessandro Vinciarelli, Dilek Hakkani T, and N. P. Garg. Role recognition for meeting participants: an approach based on lexical information and social network analysis. In *ACM International Conference on Multimedia*, 10 2008.

[63] Lesley K Fellows. The cognitive neuroscience of human decision making: a review and conceptual framework. *Behavioral and cognitive neuroscience reviews*, 3(3):159–172, 2004.

[64] David J. Field. Relations between the statistics of natural images and the response properties of cortical cells. 1987.

[65] L. Fogassi, P. F. Ferrari, B. Gesierich, S. Rozzi, F. Chersi, and G. Rizzolatti. Parietal lobe: From action organization to intention understanding. *Science*, 308(5722):662–666, 2005.

[66] V. Gallese G. Rizzolatti, L. Fogassi. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nat Rev Neurosci*, 2(9):661–670, September 2001.

[67] D. Gabor. Theory of communication. part 1: The analysis of information. *Electrical Engineers - Part III: Radio and Communication Engineering, Journal of the Institution of*, 93(26):429–441, November 1946.

[68] Wolfgang Gaissmaier and Lael J. Schooler. The smart potential behind probability matching. *Cognition*, 109(3):416 – 422, 2008.

[69] Helen L. Gallagher, Anthony I. Jack, Andreas Roepstorff, and Christopher D. Frith. Imaging the intentional stance in a competitive game. *NeuroImage*, 16(3, Part A):814 – 821, 2002.

[70] Vittorio Gallese, Luciano Fadiga, Leonardo Fogassi, and Giacomo Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119(2):593–609, 1996.

[71] Vittorio Gallese and Alvin Goldman. Mirror neurons and the simulation theory of mind reading. *Trends in Cognitive Science*, 2(12):493Ð501, 1998.

[72] D. Gao and N. Vasconcelos. Discriminant saliency for visual recognition from cluttered scenes. In *NIPS*, 2004.

[73] John S. Garofolo, Christophe Laprun, Martial Michel, Vincent M. Stanford, and Elham Tabassi. The nist meeting room pilot corpus. In *LREC*. European Language Resources Association, 2004.

[74] Robert Gibbons. *A Primer in Game Theory*. Financial Times Prent. (1. Juni 1992), 1992.

[75] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1915–1926, Oct 2012.

[76] J. J. Goodnow. Determinants of choice-distribution in two-choice situations. *American Journal of Psychology*, 68:106:116, 1955.

[77] Alison Gopnik and Henry M. Wellman. The theory theory. In Lawrence A. Hirschfeld and Susan A. Gelman, editors, *Mapping the mind*, pages 257–293. Cambridge University Press, 1994. Cambridge Books Online.

[78] S. Grabli, F. Durand, and F.X. Sillion. Density measure for line-drawing simplification. In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pages 309–318, Oct 2004.

[79] David M. Green and John A. Swets. *Signal Detection Theory and Psychophysics*. Wiley, New York, 1966.

[80] John A. Grimes. *On the failure to detect changes in scenes across saccades*. Oxford University Press (1996), 1996.

[81] Jonathan Grudin. Partitioning digital worlds: Focal and peripheral awareness in multiple monitor use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 458–465, New York, NY, USA, 2001. ACM.

[82] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. Foveated 3d graphics. *ACM Trans. Graph.*, 31(6):164:1–164:10, November 2012.

[83] Carlos Guestrin, Daphne Koller, Chris Gearhart, and Neal Kanodia. Generalizing plans to new environments in relational mdps. In *In International Joint Conference on Artificial Intelligence (IJCAI-03*, pages 1003–1010, 2003.

[84] Chenlei Guo and Liming Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *Image Processing, IEEE Transactions on*, 19(1):185–198, Jan 2010.

[85] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in Neural Information Processing Systems 19*, pages 545–552. MIT Press, 2007.

[86] M. Hayhoe and D. Ballard. Eye movements in natural behavior,. *Trends in Cognitive Science*, 9:188–194, 2005.

[87] R. Helaoui, M. Niepert, and H. Stuckenschmidt. Recognizing interleaved and concurrent activities: A statistical-relational approach. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 1 –9, march 2011.

[88] R. J. Herrnstein. Relative and absolute strength of response as a function of frequency of reinforcement1,2. *Journal of the Experimental Analysis of Behavior*, 4(3):267–272, 1961.

[89] Richard Herrnstein. *The Matching Law: Papers on Psychology and Economics*. Harvard University Press,Edited Volume, 1997.

[90] Michal Holtzman-gazit, Lihi Zelnik-manor, and Irad Yavneh. Salient edges: A multi scale approach. In *In: ECCV, Workshop on Vision for Cognitive Tasks*, 2010.

[91] Eric Horvitz, Carl Kadie, Tim Paek, and David Hovel. Models of attention in computing and communication: From principles to applications, 2003.

[92] Xiaodi Hou, Jonathan Harel, and Christof Koch. Image signature: Highlighting sparse salient regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):194–201, 2012.

[93] W. D. Hutchison, K. D. Davis, A. M. Lozano, R. R. Tasker, and J. O. Dostrovsky. Pain-related neurons in the human cingulate cortex. *Nature Neuroscience*, 2:403 – 405, 1999.

[94] Aapo Hyvaerinen, Jarmo Hurri, and Patrik O. Hoyer. *Natural Image Statistics – A probabilistic approach to early computational vision.* Springer-Verlag, London, UK, 2009.

[95] Mitsuru Ikeda, Yasuyuki Kono, and Riichiro Mizoguchi. Nonmonotonic model inference: A formalization of student modeling. In *In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence: IJCAI'93*, pages 467–473. Morgan Kaufmann, 1993.

[96] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *Image Processing, IEEE Transactions on*, 13(10):1304–1318, Oct 2004.

[97] L Itti and C Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, March 2001.

[98] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1254 –1259, nov 1998.

[99] Timothee Jost, Nabil Ouerhani, Roman von Wartburg, Rene Mueri, and Heinz Huegli. Assessing the contribution of color in visual attention. *Computer Vision and Image Understanding*, 100(1Ð2):107 – 123, 2005. Special Issue on Attention and Performance in Computer Vision.

[100] Ricardo Jota, Miguel A. Nacenta, Joaquim A. Jorge, Sheelagh Carpendale, and Saul Greenberg. A comparison of ray pointing techniques for very large displays. In *Proceedings of Graphics Interface 2010*, GI '10, pages 269–276, Toronto, Ont., Canada, Canada, 2010. Canadian Information Processing Society.

[101] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2106–2113, Sept 2009.

[102] Leslie Pack Kaelbling, Michael L. Littman, and Andrew P. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[103] R. Kalman. When is a linear control system optimal? *Trans. ASME, J. Basic Engrg.*, 86:51Ð60, 1964.

[104] Christian Keysers and Valeria Gazzola. Towards a unifying neural theory of social cognition. In G. Anders, S.and Ende, M. Junghofer, J. Kissler, and D. Wildgruber, editors, *Understanding Emotions*, volume 156 of *Progress in Brain Research*, pages 379 – 401. Elsevier, 2006.

[105] W. Kienzle, FA. Wichmann, B. Schölkopf, and MO. Franz. A nonparametric approach to bottom-up visual saliency. In B Schölkopf, J Platt, and T Hofmann, editors, *Advances in Neural Information Processing Systems 19*, pages 689–696, Cambridge, MA, USA, September 2007. Max-Planck-Gesellschaft, MIT Press.

[106] Thomas Kleinbauer, Stephanie Becker, and Tilman Becker. Combining multiple information layers for the automatic generation of indicative meeting abstracts. In *In: Proc. of ENLG 2007*, 2007.

[107] C. Koch and S. Ullman. Shifts in selective visual attention: Towards the underlying neural circuitr. *Human Neurobiolog*, 4:219–227, 1985.

[108] J. Zico Kolter and Andrew Y. Ng. The stanford littledog: A learning and rapid replanning approach to quadruped locomotion. *I. J. Robotic Res.*, 30(2):150–174, 2011.

[109] G. Kootstra, A. Nederveen, and B. de Boer. Paying attention to symmetry. In *Proceedings of the British Machine Vision Conference*, pages 111.1–111.10. BMVA Press, 2008. doi:10.5244/C.22.111.

[110] Mark A. P. Kuzmycz and Geoffrey I. Webb. Modelling elementary subtraction: Intelligent warfare against bugs. In *Proceedings of the Fourth Australian Society for Computers in Learning in Tertiary Education Conference, Launceston, pp. 367-376*, 1991.

[111] John E. Laird. Extending the soar cognitive architecture. In *In: Proceedings of the First Conference on Artificial General Intelligence*. Springer, 2008.

[112] John E. Laird, Allen Newell, and Paul S. Rosenbloom. Soar: An architecture for general intelligence. *Artif. Intell.*, 33(1):1–64, September 1987.

[113] Pat Langley and Kirstin Cummings. Hierarchical skills and cognitive architectures. In *Proceedings of the Twenty-Sixth Annual Conference of the Cognitive Science Society (pp. 779Ð 784*, pages 779–784, 2004.

[114] Pat Langley, John E. Laird, and Seth Rogers. Cognitive architectures: Research issues and challenges. *Cogn. Syst. Res.*, 10(2):141–160, June 2009.

[115] Anke Lehmann and Oliver Staadt. Distance-aware bimanual interaction for large high-resolution displays. In *7th International Joint Conference, VISIGRAPP 2012, Rome, Italy*, pages 24–26, February 2012.

[116] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Feature construction for inverse reinforcement learning. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1342–1350. Curran Associates, Inc., 2010.

[117] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with gaussian processes. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 19–27. 2011.

[118] J. P. Lewis. Fast normalized cross-correlation, 1995.

[119] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2):353–367, February 2011.

[120] DavidG. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[121] Vijay Mahadevan and Nuno Vasconcelos. Spatiotemporal saliency in dynamic scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):171–177, January 2010.

[122] Joil Martin and Kurt WanLehn. Ola e: Progress toward a multi-activity, bayesian student modeller. In *In Proceedings of AI-Ed 93, Edinburgh, pp. 410-417*, 1993.

[123] Maja J Mataric. Reward functions for accelerated learning. In *In Proceedings of the Eleventh International Conference on Machine Learning*, pages 181–189. Morgan Kaufmann, 1994.

[124] K McCabe and T Singer. *Brain signatures of social decision making*, pages 103–122. Strüngmann Forum Reports. MIT Press, Cambridge, Massachusetts, 2008.

[125] Kevin McCabe, Daniel Houser, Lee Ryan, Vernon Smith, and Theodore Trouard. A functional imaging study of cooperation in two-person reciprocal exchange. *Proceedings of the National Academy of Sciences*, 98(20):11832–11835, 2001.

[126] I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. The ami meeting corpus. In *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research. L.P.J.J. Noldus, F. Grieco, L.W.S. Loijens and P.H. Zimmerman (Eds.), Wageningen: Noldus Information Technology*, 2005.

[127] L. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang. Automatic analysis of multimodal group actions in meetings. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):305 –317, march 2005.

[128] B. Mertsching, M. Bollmann, R. Hoischen, and S. Schmalz. *The Neural Active Vision System Navis.* Handbook of Computer Vision and Applications Vol. 3 (Systems and Applications), B. Jahne, H. Haullecke, and P. GeiI3ler, eds., pp. 543-568,, 1999.

[129] Florence Miau, Constantine S. Papageorgiou, and Laurent Itti. Neuromorphic algorithms for computer vision and attention, 2001.

[130] Redwan Abdo A. Mohammed, Samah Abdulfatah Mohammed, and Lars Schwabe. Batgaze: A new tool to measure depth features at the center of gaze during free viewing. In FabioMassimo Zanzotto, Shusaku Tsumoto, Niels Taatgen, and Yiyu Yao, editors, *Brain Informatics*, volume 7670 of *Lecture Notes in Computer Science*, pages 85–96. Springer Berlin Heidelberg, 2012.

[131] Redwan Abdo A. Mohammed and Lars Schwabe. Scene-dependence of saliency maps of natural luminance and depth images. In *Fifth Baltic Conference "Human - Computer Interaction"*. (to appear), 2011.

[132] Redwan Abdo A. Mohammed, Lars Schwabe, and Oliver Staadt. Gaze location prediction with depth features as auxiliary information. In Masaaki Kurosu, editor, *Human-Computer Interaction. Advanced Interaction Modalities and Techniques*, volume 8511 of *Lecture Notes in Computer Science*, pages 281–292. Springer International Publishing, 2014.

[133] Redwan Abdo A. Mohammed and Oliver Staadt. Effects of Interior Bezels of Tiled Large High-Resolution Displays on Saliency Prediction and Human Eye Movement Behavior. In Yuki Hashimoto, Torsten Kuhlen,

Ferran Argelaguet, Takayuki Hoshi, and Marc Erich Latoschik, editors, *ICAT-EGVE 2014 - Posters and Demos*. The Eurographics Association, 2014.

[134] Redwan Abdo A. Mohammed and Oliver Staadt. Learning eye movements strategies on tiled large high-resolution displays using inverse reinforcement learning. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–7, July 2015.

[135] Redwan AbdoA. Mohammed, Lars Schwabe, and Oliver Staadt. Towards context-dependence eye movements prediction in smart meeting rooms. In Stefan Wermter, Cornelius Weber, Duch, Timo Honkela, Petia Koprinkova Hristova, Sven Magg, Guenther Palm, and AlessandroE.P. Villa, editors, *Artificial Neural Networks and Machine Learning Ð ICANN 2014*, volume 8681 of *Lecture Notes in Computer Science*, pages 249–256. Springer International Publishing, 2014.

[136] RedwanAbdoA. Mohammed and Lars Schwabe. A brain informatics approach to explain the oblique effect via depth statistics. In FabioMassimo Zanzotto, Shusaku Tsumoto, Niels Taatgen, and Yiyu Yao, editors, *Brain Informatics*, volume 7670 of *Lecture Notes in Computer Science*, pages 97–106. Springer Berlin Heidelberg, 2012.

[137] N. Morgan, D. Baron, S. Bhagat, H. Carvey, R. Dhillon, J. Edwards, D. Gelbart, A. Janin, A. Krupski, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. Meetings about meetings: research at icsi on speech in multiparty conversations. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 4, pages IV – 740–3 vol.4, april 2003.

[138] Katharina Muelling, Abdeslam Boularias, Betty Mohler, Bernhard Schölkopf, and Jan Peters. Learning strategies in table tennis using inverse reinforcement learning. *Biol. Cybern.*, 108(5):603–619, October 2014.

[139] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive computation and machine learning series. Mit Press, 2012.

[140] Vidhya Navalpakkama, Christof Kocha, Antonio Rangelc, and Pietro Peronab. Optimal reward harvesting in complex perceptual environments. *Proceedings of the National Academy of Sciences*, 107, no. 11:5232–5237, 2010.

[141] Chrystopher L. Nehaniv and Kerstin Dautenhahn. *Imitation and Social Learning in Robots, Humans and Animals Behavioural, Social and Communicative Dimensions*. Cambridge Univ Press, 2007.

[142] Allen Newell. You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. Chase, editor, *Visual information processing*, pages 283–308. Academic Press, New York, 1973.

[143] Allen Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA, USA, 1990.

[144] Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *In Proceedings of the Sixteenth International Conference on Machine Learning*, pages 278–287. Morgan Kaufmann, 1999.

[145] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[146] Tao Ni, Greg S. Schmidt, Oliver G. Staadt, Mark A. Livingston, Robert Ball, and Richard May. A survey of large high-resolution display technologies, techniques, and applications. In *Proceedings of the IEEE Conference on Virtual Reality*, VR '06, pages 223–236, Washington, DC, USA, 2006. IEEE Computer Society.

[147] D. Warner North. A tutorial introduction to decision theory. *IEEE Transactions on Systems Science and Cybernetics*, 1968.

[148] K. N. Ochsner and M. D. Lieberman. The emergence of social cognitive neuroscience. *Am Psychol*, 56(9):717–734, September 2001.

[149] S. Ohlsson and N. Bee. Strategy variability. a challenge to models of procedural learning. In *Proceedings of the International Conference of the Learning Sciences, Charlottesville, VA., pp. 351-356*, 1991.

[150] Martin J. Osborne. *An Introduction to Game Theory*. Oxford University Press;, 2003.

[151] Nabil Ouerhani. *Visual Attention: From Bio-Inspired Modeling to Real-Time Implementation*. PhD thesis, Institut de Microtechnique, Universit de Neuchate, Switzerland, 2003.

[152] Giovanni Parmigiani and Lurdes Inoue. *Decision Theory: Principles and Approaches*. Wiley; 1 edition, 2009.

[153] Brian Potetz and Tai Sing Lee. Statistical Correlations Between 2D Images and 3D Structures in Natural Scenes. *Journal of Optical Society of America, A*, 7(20):1292–1303, 2003.

[154] Guy Premack, David; Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526, Dec 1978.

[155] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

[156] Umesh Rajashekar, Alan C. Bovik, and Lawrence K. Cormack. Visual search in noise: Revealing the influence of structural cues by gaze-contingent classification image analysis. *Journal of Vision*, 6(4):7, 2006.

[157] Deepak Ramachandran. Bayesian inverse reinforcement learning. In *in 20th Int. Joint Conf. Artificial Intelligence*, 2007.

[158] Miquel Ramirez and Hector Geffner. Goal recognition over pomdps: inferring the intention of a pomdp agent. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2009–2014. AAAI Press, 2011.

[159] Anand S. Rao and Michael P. Georgeff. Bdi agents: From theory to practice. In *IN PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS (ICMAS-95*, pages 312–319, 1995.

[160] P. Reinagel and A. M. Zador. Natural scene statistics at the centre of gaze. *Network (Bristol, England)*, 10(4):341–350, November 1999.

[161] R. A. Rensink, J. K. O'Regan, and James J. Clark. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8:368 –373, 1997.

[162] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999.

[163] Giacomo Rizzolatti and Laila Craighero. The mirror-neuron system. *Annual Review of Neuroscience*, 27:169–192, July 2004.

[164] G. Robertson, M. Czerwinski, P. Baudisch, B. Meyers, D. Robbins, G. Smith, and D. Tan. The large-display user experience. *Computer Graphics and Applications, IEEE*, 25(4):44–51, July 2005.

[165] Claudia Roda. *Human Attention in Digital Environments*. Cambridge University Press, Cambridge, UK, 2011.

[166] C.A. Rothkopf, T.H. Weisswange, and J. Triesch. Learning independent causes in natural images explains the spacevariant oblique effect. In *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*, pages 1 –6, june 2009.

[167] Nicholas Roy. Finding Approximate POMDP solutions Through Belief Compression. *Journal of Artificial Intelligence Research*, 23, 2000.

[168] Albert Ali Salah, Ethem Alpaydin, and Lale Akarun. A selective attention-based method for visual pattern recognition with application to handwritten digit recognition and face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):420–425, 2002.

[169] Anthony Santella, Maneesh Agrawala, Doug DeCarlo, David Salesin, and Michael Cohen. Gaze-based interaction for semi-automatic photo cropping. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 771–780, New York, NY, USA, 2006. ACM.

[170] A. Saxena and H Sung. Learning depth from single monocular images. In *In NIPS 18*. MIT Press, 2005.

[171] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Philosophical Transaction of the Royal Society of London*, Series B, Biological Sciences(1431):537–547, 2003.

[172] O Schwartz and E P Simoncelli. Natural signal statistics and sensory gain control. *Nature Neuroscience*, 4(8):819–825, 2001.

[173] Richard B. Segal and Jeffrey O. Kephart. Mailcat: An intelligent assistant for organizing e-mail. In *Proceedings of the Third Annual Conference on Autonomous Agents*, AGENTS '99, pages 276–282, New York, NY, USA, 1999. ACM.

[174] Oliver G. Selfridge, Richard S. Sutton, and Andrew G. Barto. Training and tracking in robotics. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'85, pages 670–672, San Francisco, CA, USA, 1985. Morgan Kaufmann Publishers Inc.

[175] SensoMotoric Instruments. *SMI iView X system manual*, 2.7 edition, March 2011.

[176] Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. Shadow reaching: A new perspective on interaction for large displays. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 53–56, New York, NY, USA, 2007. ACM.

[177] C. Siagian and L. Itti. Biologically inspired mobile robot vision localization. *Robotics, IEEE Transactions on*, 25(4):861–873, Aug 2009.

[178] E. P. Simoncelli and B. A. Olshausen. Natural image statistics and neural representation. *Annu Rev Neurosci*, 24(1):1193–1216, 2001.

[179] Eero P Simoncelli and William T Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *IEEE INTL CONF ON IMAGE PROCESSING*, pages 444–447. IEEE Signal Processing Society, 1995.

[180] Tania Singer. Understanding others: brain mechanisms of theory of mind and empathy. In Paul W Glimcher, Colin F Camerer, Ernst Fehr, and Russell A Poldrack, editors, *Neuroeconomics: decision making and the brain*, pages 251–268. Elsevier, Amsterdam, 2008.

[181] Tania Singer and Ernst Fehr. The neuroeconomics of mind reading and empathy. IZA Discussion Papers 1647, Institute for the Study of Labor (IZA), 2005.

[182] Satinder P. Singh, Richard L. Lewis, Andrew G. Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE T. Autonomous Mental Development*, 2(2):70–82, 2010.

[183] Satinder Pal Singh. Transfer of learning by composing solutions of elemental sequential tasks. In *Machine Learning*, pages 323–339, 1992.

[184] D.J. Sismons and D.T. Levin. Failure to detect changes to people in a real world interaction. *Psychonomic Bulletin Review*, 5:644:649, 1998.

[185] Alexander Johannes Smola and S.V.N. Vishwanathan. *Introduction to Machine Learning*. Cambridge University Press, 2008.

[186] R. Snowden, P. Thompson, and T. Troscianko. *Basic vision: An introduction to visual perception*. Oxford: Oxford Univ. Press, 2006.

[187] Mark Stefik, Gregg Foster, Daniel G. Bobrow, Kenneth Kahn, Stan Lanning, and Lucy Suchman. Beyond the chalkboard: computer support for collaboration and problem solving in meetings. *Commun. ACM*, 30(1):32–47, January 1987.

[188] L. P. Sugrue, G. S. Corrado, and W. T. Newsome. Matching behavior and the representation of value in the parietal cortex. *Science*, 304:1782–1787, 2004.

[189] Bongwon Suh, Haibin Ling, Benjamin B. Bederson, and David W. Jacobs. Automatic thumbnail cropping and its effectiveness. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 95–104, New York, NY, USA, 2003. ACM.

[190] R. Sun, E. Merrill, and T. Peterson. From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science*, 25(2):203–244, 2001.

[191] R. Sutton and A. Barto. *Reinforcement Learning: An Intro- duction*. MIT Press, Cambridge, 1998.

[192] Desney S. Tan and Mary Czerwinski. Effects of visual separation and physical discontinuities when distributing information across multiple displays. In *In Proceedings of Interact 2003*, pages 252–255, 2003.

[193] Desney S. Tan, Darren Gergle, Peter Scupelli, and Randy Pausch. With similar visual angles, larger displays improve spatial performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 217–224, New York, NY, USA, 2003. ACM.

[194] Benjamin W. Tatler, Roland J. Baddeley, and Iain D. Gilchrist. Visual correlates of fixation selection: Effects of scale and time. *Vision*, Volume 45, Issue 5:643Ð659, 2005.

[195] Antonio Torralba. Modeling global scene factors in attention. *JOSA - A*, 20:1407–1418, 2003.

[196] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

[197] J H Van Hateren and A Van Der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society B Biological Sciences*, 265(1394):359–366, 1998.

[198] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience; 1 edition, 1998.

[199] B. M. Velichkovsky, Marc Pomplun, Johannes Rieser, and Helge Ritter. *Attention and Communication: Eye-Movement-Based Research Paradigms*, volume 116 of *Visual Attention and Cognition*, pages 125–154. Elsevier, 1996.

[200] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.

[201] N Vulkan. An economists perspective on probability matching. *Journal of Economic Surveys*, 14:101:118, 2000.

[202] James R. Wallace, Daniel Vogel, and Edward Lank. Effect of bezel presence and width on visual search. In *Proceedings of The International Symposium on Pervasive Displays*, PerDis '14, pages 118:118–118:123, New York, NY, USA, 2014. ACM.

[203] Dirk Walther and Christof Koch. Modeling attention to salient proto-objects, 2006.

[204] G. I. Webb. Feature based modelling: A methodology for producing coherent, consistent, dynamically changing models of agents competency. In P. Brna, S. Ohlsson, and H. Pain, editors, *Proceedings of the 1993 World Conference on Artificial Intelligence in Education (AI-ED'93)*, pages 497–504, Charlottesville, VA, 1993. AACE.

[205] H.M. Wellman. *The Child's Theory of Mind*. Learning, development, and conceptual change. Mit Press, 1990.

[206] Pierre Wellner, Mike Flynn, and Mael Guillemot. Browsing recorded meetings with ferret. In *In In Proceedings of MLMI04*, pages 12–21. Springer-Verlag, 2004.

[207] RichardF. West and KeithE. Stanovich. Is probability matching smart? associations between probabilistic choices and cognitive ability. *Memory & Cognition*, 31(2):243–251, 2003.

[208] Christopher D. Wickens, John D. Lee, Yili Liu, and Sallie E. Gordon Becker. *An Introduction to Human Factors Engineering*. Pearson Prentice Hall, Upper Saddle River, NJ, 2004.

[209] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, pages 229–256, 1992.

[210] Heinz Wimmer and Josef Perner. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1):103 – 128, 1983.

[211] G. Wolford, S. Newman, M. B. Miller, and G. Wig. Searching for patterns in random sequences. *Canadian Journal of Experimental Psychology*, 58:221:228, 2004.

[212] Z. Yang and D. Purves. Image/source statistics of surfaces in natural scenes. *Network: Computation in Neural Systems*, 14:371Ð390, 2003.

[213] A. L. Yarbus. *Eye movements and vision*. Plenum Press, New York, 1967.

[214] A.L. Yarbus. Eye-movements and vision. *Plenum Press, New York,*, 1967.

[215] J. Frank Yates. *Judgment and Decision Making*. Prentice Hall College Div, January 1990.

[216] N. Yokoya and M.D. Levine. Range image segmentation based on differential geometry: a hybrid approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(6):643 –649, jun 1989.

[217] Dong Zhang, Daniel Gatica-perez, Samy Bengio, Iain Mccowan, and Guillaume Lathoud. Modeling individual and group actions in meetings: a two-layer hmm framework. In *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Workshop on Event Mining in Video (CVPREVENT), Washington DC*, 2004.

[218] Song Chun Zhu and David Mumford. Prior learning and gibbs reaction-diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:1236–1250, November 1997.

[219] Brian Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 1433–1438. AAAI Press, 2008.

[220] Brian D. Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J. Andrew (Drew) Bagnell, Martial Hebert, Anind Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. In *Proc. IROS 2009*, October 2009.

Appendix A

Nomenclature

Acronyms

ACRONYM	DESCRIPTION
SVM	Support Vector Machines
RL	Reinforcement Learning
IRL	Inverse Reinforcement Learning
MDPs	Markov Decision Processes
POMDPs	Partially Observable Markov Decision Processes
LHRD	Large High-Resolution Display
DLP TV	Digital Light Processing TV
LCD	Liquid-Crystal Display
2D	Intensity image or luminance image
3D	Range image or depth map
MI	Mutual Information
KL	Kullback-Leibler divergence
AUR	Area Under Curve
CC	Correlation Coefficient
MSE	Mean Squared Error
EVD	Expected Value Difference
HMMs	Hidden Markov Models

Table A.1: Acronyms descriptions.

Mathematical Notations

SYMBOL	DESCRIPTION
$corr$	correlation coefficient
cov	covariance function
var	variance function
$i(x, y)$	Gray scale pixel value
$d(x, y)$	Depth value
$\sigma(x, y)$	Local standard deviation
$H_{x,y}$	Local entropy
$g_\theta(x, y)$	Gabor filter response
$\mu_{GD}(x, y)$	Gap discontinuity
$\mu_{OD}(x, y)$	Surface orientation discontinuity

Table A.2: Mathematical Notation.

Appendix B

Operations Details

B.1 Gabor Filters

The two-dimensional Gabor function consists of a complex sinusoidal plane wave of some frequencies and orientations, modulated by a 2D Gaussian function[50], and hence defined as follows:

$$g_{\lambda\theta\psi\sigma\gamma}(x,y) = \exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right).\cos\left(2\pi\frac{x'}{\lambda}+\psi\right) \tag{B.1}$$

where

$$x' = x\cos(\theta) + y\sin(\theta)$$

$$y' = y\cos(\theta) - x\sin(\theta)$$

In this equation, λ represents the wavelength of the sinusoidal wave, θ denotes the orientation of the normal to the parallel lines of the Gabor function in degrees, ψ is the phase offset in degrees, γ is the spatial aspect ratio.

The spatial frequency of the sinusoidal wave is defined as $f = \frac{1}{\lambda}$. The ratio $\frac{\sigma}{\lambda}$ determines the spatial frequency bandwidth. The half-response spatial frequency bandwidth b (in octaves) and the ratio $\frac{\sigma}{\lambda}$ are related as it follows:

$$b = \log_2 \frac{\frac{\sigma}{\lambda}\pi + \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda}\pi - \sqrt{\frac{\ln 2}{2}}} \tag{B.2}$$

$$\frac{\sigma}{\lambda} = \frac{1}{\pi}\sqrt{\frac{\ln 2}{2}\frac{2^b+1}{2^b-1}} \tag{B.3}$$

We can process any Image $I(x,y)$ by a Gabor filter $g(x,y)$, the result is the convolution of the image and the Gabor function, i.e., $r(x,y) = g(x,y) * I(x,y)$, where $*$ denotes the two dimensional convolution [50]. .

Choice of the filter parameters: We use orientation separation angles of 15°, so that θ values are given as follow:

$$\theta = \{0°, 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150°, 165°, 180°\}$$

As recommended in [50] and other studies [178], we used only one spatial frequency $\lambda = 6.1$ (and the standard deviation of the Gaussian $\sigma = 3.4$) and two spatial phases $\psi \in \{0, \pi/2\}$ and we set the spatial aspect ratio $\gamma = 1$.

B.2 Overview of the Processing Workflow of the BatGaze System

Figure B.1 shows BatGaze system under the second version after recording and saving to disk. This process also have five tasks to run over them.

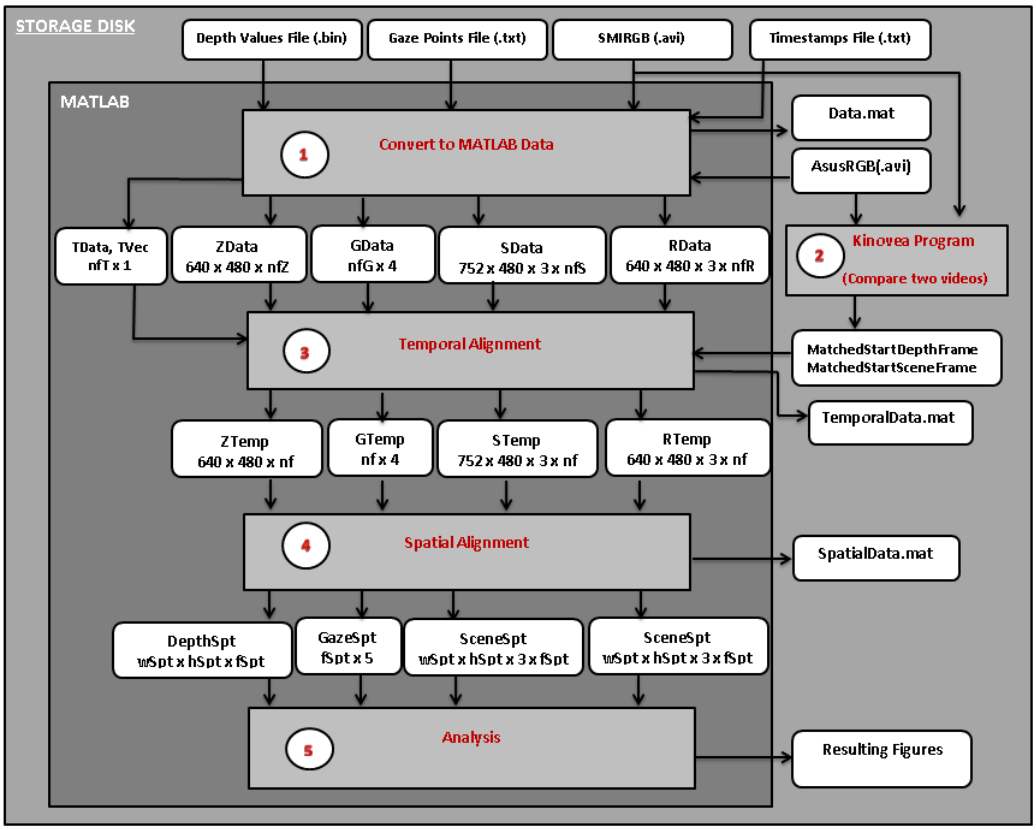


Figure B.1: The BatGaze system workflow.

First task (Data Conversion):

The files are read by MATLAB and then data are converted to MATLAB-workspace variables. For example, the scene video file from SMI will be read and converted to 4D matrix (752 x 480 x 3 x no. of frames). The binary file of the depth values, which is in a vector form, will be converted to 3D matrix (640 x 480 x no. of frames). But this conversion depends on the recording duration which should be given in seconds. Wrong entry of the recording duration is checked with the length of the depth vector and the image resolution and then an error of mismatching number of frames is returned. Artifacts correction of the data is done in this task. To perform the next task, the depth images are transformed to gray-scale and a video of them is generated and saved to disk. In each task and for memory issues, the data under processing is saved to disk in the end of the task and MATLAB workspace can be cleared or if desired can go on to the next task. At any time one can start from any intermediate task and process the proper data without the need to start from the beginning. In table B.1, we summarize the conversion from raw data to MATLAB based data. The files are read by MATLAB and then data are converted to MATLAB-workspace variables.

After conversion to MATLAB workspace variable, the data are saved to disk.

Second task (Frames Comparing):

Here we used an external program called "Kinovea[1]" (kinovea.org) to compare the frames of the two videos; the SMI scene video and the generated Asus depth video. As the frames are almost exceeding 1000 frames so we just moved to compare the event-frames in the beginning or in the end of both videos. We decide here on the matched frames (MatchedStartDepthFrame, Matched-StartSceneFrame) and supply their values to the next task. This way we get the proper position of the "event" frames, and we could check the timestamp method. Here the external program "Kinovea" used to compare the frames of the two videos; the SMI scene video and the Asus scene video.

Third task (Temporal Alignment):

We interpolated the timestamps here to find the matching frames and examined the values from task two as a double check. We then trimmed the unmatched frames from both data images to get in the final equal umber of frames where each frame in the depth images is aligned temporarily to the corresponding frame in the SMI scene images. Gaze data vector also trimmed consequently.

[1]http://www.kinovea.org/

Here, all the data are trimmed to aligned number of frames, that is nf. All data variables preserve its size.

Forth task (Spatial Alignment):

The temporal-aligned matrices of images are now registered spatially using MATLAB registration tools. Transformation is performed on the raw depth images as targets, not the gray-scale images, and on the scene images as references. Using the transformation function the new positions of the gaze points are computed. After the 2D spatial registration, the parts of images that are mostly registered, are cropped and a new size of images results. Here the whole data set is aligned temporarily and spatially and the depth value of each gaze point is accurately computed. We can generate a movie of both scene views in this task.

We here applied the registration on the two RGB images, by using either the monomodal or multimodal registration method. The resulting transformation is performed on both frames types: raw depth images and RGB images. Here we can opt to correct the fisheye effect of eye-tracker frames and/or to correct the shadow problem of the Asus depth camera. The process can include some image processing such as cropping the most aligned parts of the scenes. For each subtask here, the gaze data positions are computed and new gaze data results. Then the whole data set is aligned temporarily and spatially and the depth value of each gaze point is accurately computed. We can generate a movie of both scene views in this task.

Fifth task (Analysis):

Resulting data from previous step comes in here. In each task the output is saved to disk so one can later proceed from any task by loading those saved data. It helps in dealing with memory problems.

Table B.1: Description of the conversion of files to MATLAB variables.

File	Description
Depth Values File (.bin)	This is the binary file that saves the depth frames. It contains one vector of unsigned short data types for each depth value. It is then converted to matrix of ($640 \times 480 \times nfZ$) representing the depth resolution used and nfZ is the number of depth frames.
Asus RGB (.avi)	This is the video file that saves the RGB frames from Asus XtionPRO Live camera. MATLAB reads it and converts it to matrix of ($640 \times 480 \times 3 \times nfR$) with the resolution used and nfR is the number of scene frames in unsigned 8 bit integers. nfR doesn't always equal nfZ.
Timestamps File(.txt)	This is the text file that saves the depth frames timestamps. It contains one vector of unsigned short data types for each depth frame. It is then converted to two vectors, each of (nfT) entry. nfT is the number of depth frames and always equals nfZ. First vector $TData$ contains the timestamps, and the second $Tvec$ contains the system time format of those timestamps. The recording time is also included in this file.
SMIRGB (.avi)	This is the video file that saves the RGB frames from eye-tracker camera. MATLAB reads it and converts it to matrix of ($752 \times 480 \times 3 \times nfS$) with its resolution. nfS is the number of scene frames in unsigned 8 bit integers. nfS d differs than nfZ.
Gaze Data (.txt)	This is the text file that saves the gaze data. It contains one cell of arrays where each array has its own type. Its size is of ($nfG \times 4$) , where nfG is the number of sampled gaze data which is always twice the size of nfS , and sometimes more. The first column contains the timestamps. The second and third columns contain the gaze data in pixels in x and y direction. Finally the forth column contains the frames counter. The recording time is also included in this file.

Appendix C

Experiments and Analysis Results Details

C.1 Models Performance for the Individual Subjects in Meeting Scenarios

In chapter 4, we examined the prediction of eye movements in meeting scenarios using different low, middle and high-level visual features. We trained a linear SVM to find out which features are descriptive in two scenarios (giving a talk vs. listening to a talk). We measured the performance of saliency models using KL divergence. Here we present the results of the performance of different models for the individual subject. Again, the context dependence shows up with all individual subjects: In the listening scenario, models trained on competing saliency features from Itti and Koch perform better than the models trained on other features (see Figure C.1a). In the giving a talk scenario, models trained on the face features perform better than the models trained on other single features (see Figure C.1b).

Figures C.2 and C.3 show the KL divergence matrices describing the performances of different SVMs models, for individual subject, in the "listening -audience-" scenario and "giving a talk-speaker-" scenario.

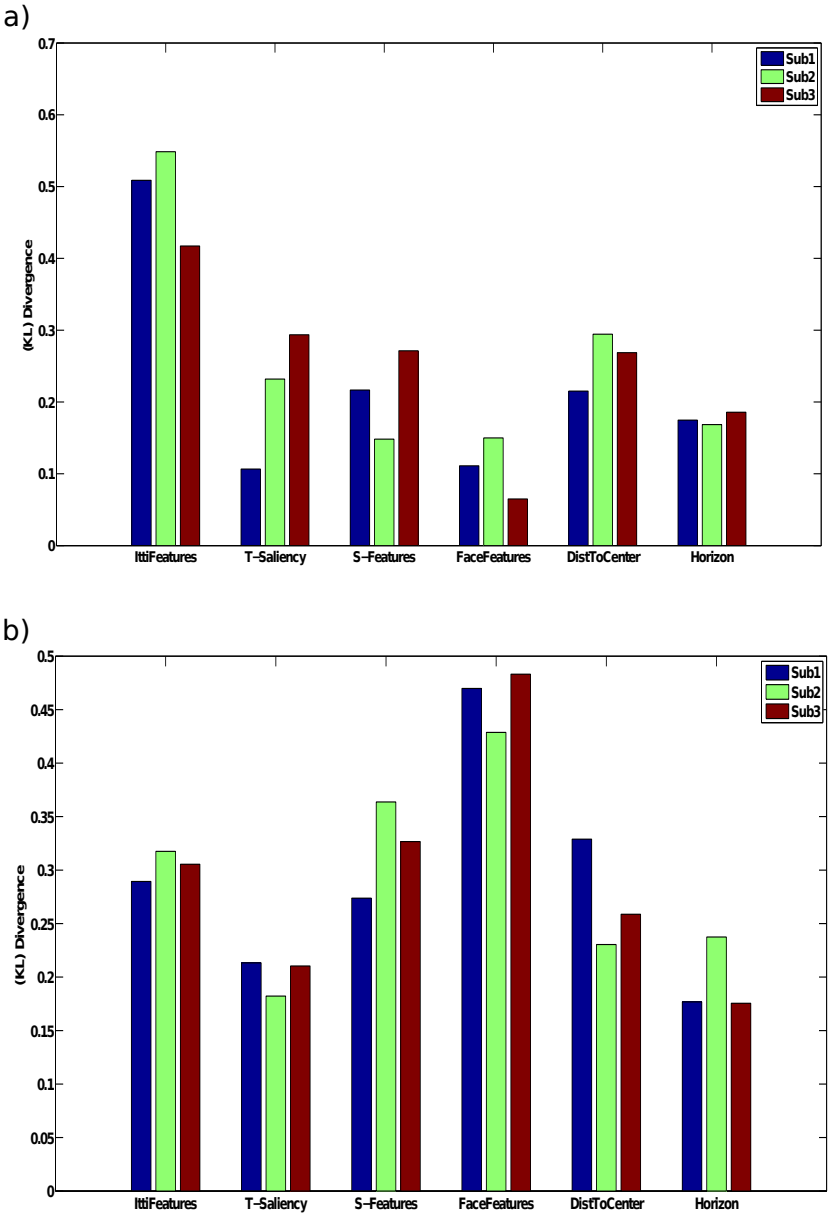


Figure C.1: The KL divergence describing the performance of different SVMs trained on each feature individually, for individual subject, in two scenarios. a) In the listening scenarios. b) In the given talk scenarios.

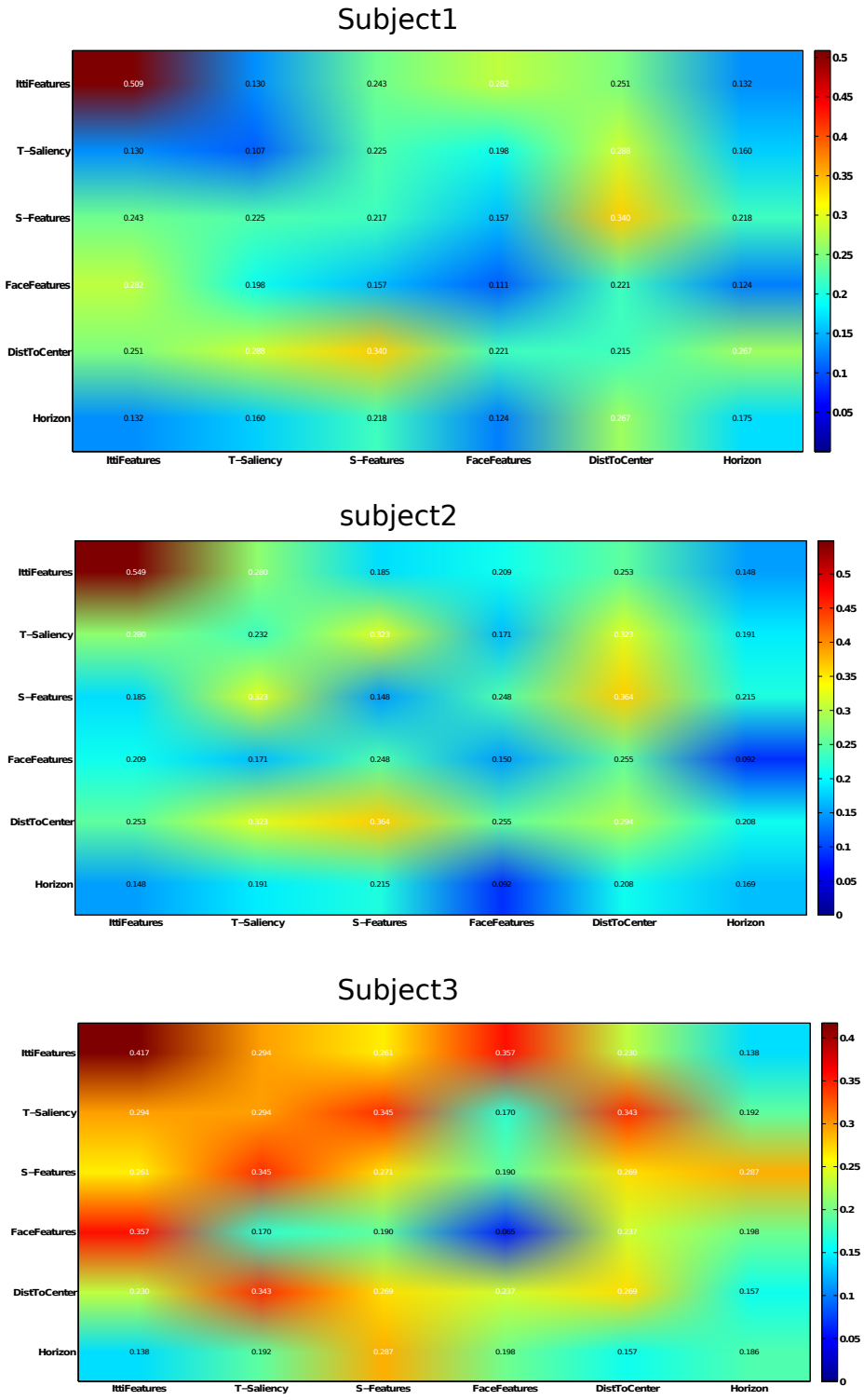


Figure C.2: The KL divergence matrix describing the performances of different SVMs models trained on a set of features individually and pairs of features combined together, in the "listening -audience-" scenario, for individual subject. The main diagonal shows the performances of the models trained on individual features. The lower/ upper triangular parts of the matrix show the performances of the models trained on pairs of features combined.

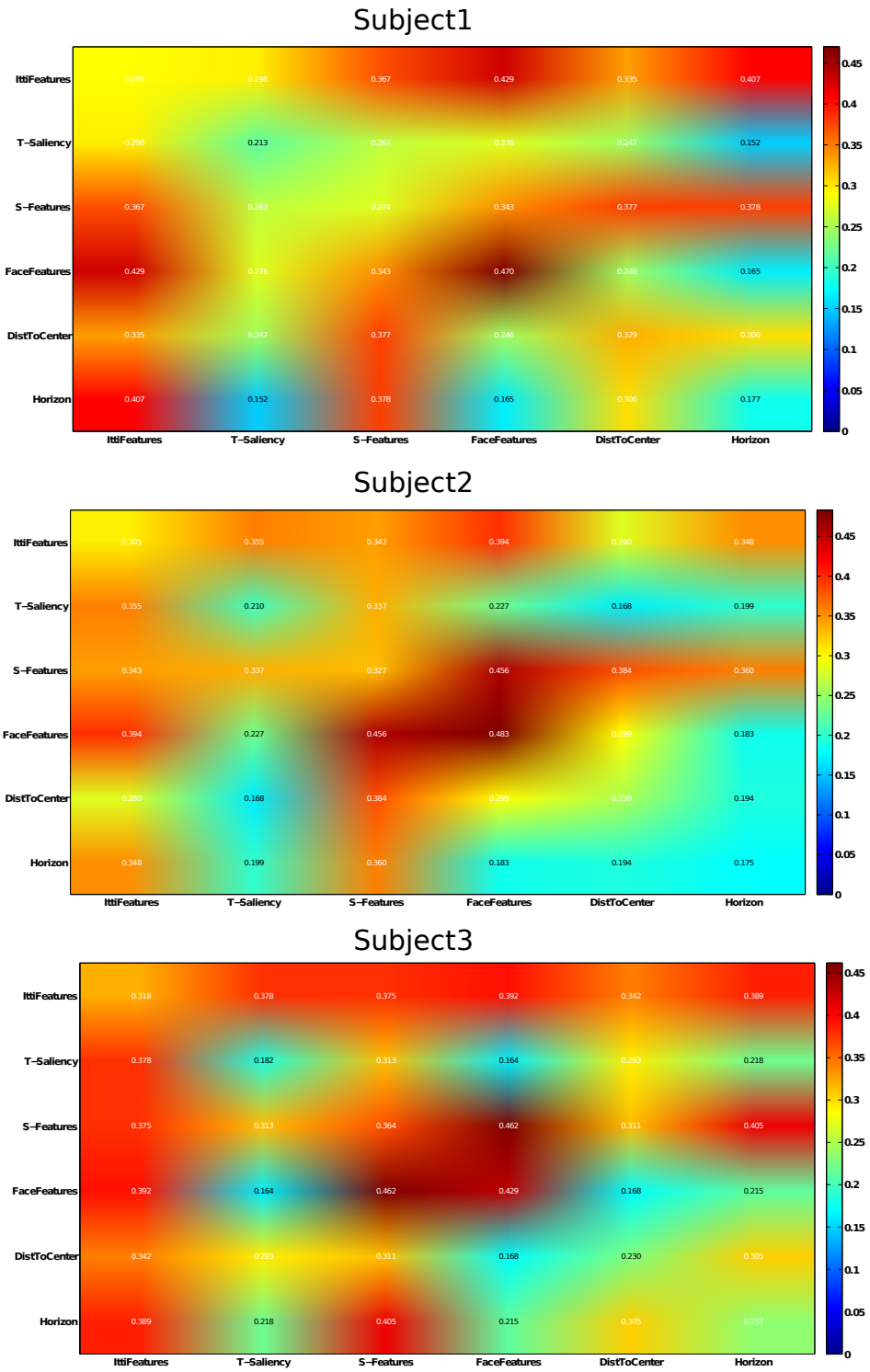


Figure C.3: The KL divergence matrix describing the performances of different SVMs models trained on a set of features individually and pairs of features combined together, in the "giving a talk-speaker-" scenario, for individual subject. The main diagonal shows the performances of the models trained on individual features. The lower/ upper triangular parts of the matrix show the performances of the models trained on pairs of features combined.

C.2 Models Performance when using Depth Features for the Individual Subjects

In chapter 5, we investigate, how relevant depth features are for eye movement prediction. Here we present the results of the performance of different features models for the individual subjects. Figure 5.19 compare KL performance for SVMs trained with different individual features and combined together.

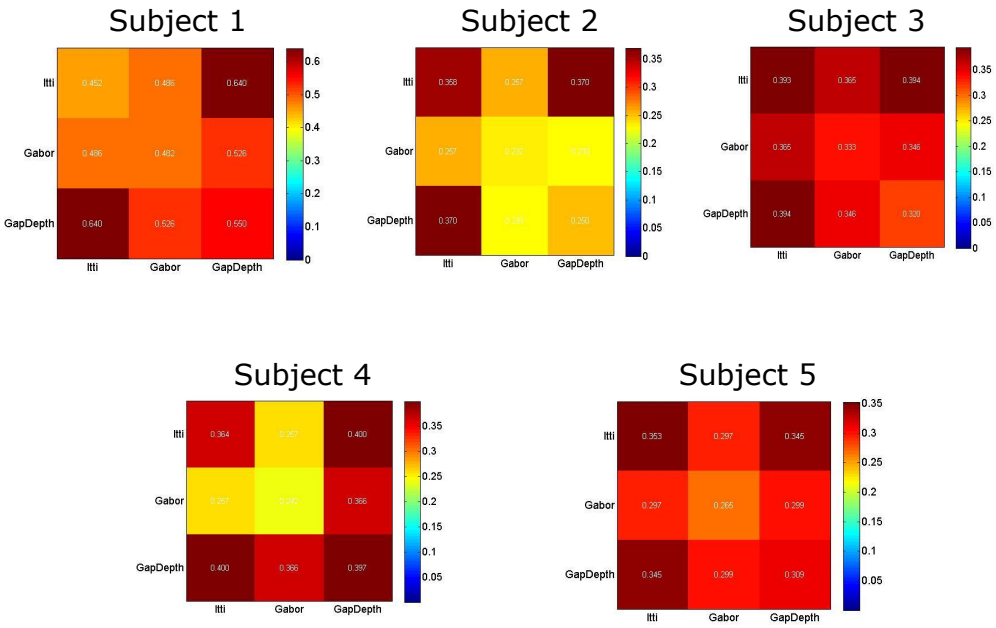


Figure C.4: The KL divergence matrix describing the performance of different SVMs models trained on set of features individually and pairs of features combined, for the individual subjects. The main diagonal shows the performance of the models trained on individual features. The lower/ upper triangular parts of the matrix show the performance of the models trained on pairs of features combined.

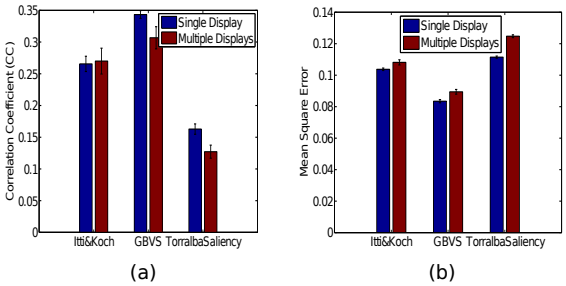


Figure C.5: **a)** The correlation coefficient used to compare the relationship between the predictions of Itti and Koch, GBVS and Torralba saliency models and the user fixation maps, in two scenarios (single display vs. multi tiled displays), averaged over all subjects. **b)** The mean square error between the predictions of Itti and Koch, GBVS and Torralba saliency models and the user fixation maps, in two scenarios (single display vs. multiple LHRD), averaged over all subjects.

C.3 Comparing Visual Saliency Models Predictions

In chapter 6, we investigate the effects of bezels on human eye movements and on saliency algorithm predictions. Here we compared the relationship between human fixation maps and the saliency maps generated by Itti and Koch, GBVS and Torralba using the correlation coefficient. In Figure C.5(a) we can see the correlation between human fixation maps and the prediction maps averaged across all users and all images. We can see that the correlations between the human fixation maps and the saliency maps from the single DLP display are higher than in multi tiled displays images. On the other hand, the correlation between the human fixation maps and the GBVS predictions are higher than in the Itti and Koch model and Torralba model, in both displays, which means that, the GBVS model provides overall better performance than Itti and Koch model and Torralba model. These results were confirmed when we used the cumulative squared error between human fixation maps and the saliency maps (see Figure C.5(b)).

C.4 Examples of the Experiments Data

Examples of images that were shown in the experiments from the *Microsoft Salient Object Dataset* [119] and from the *York University Eye Fixation Dataset* [32] (see chapters 6 and 7):

Figure C.6: Examples of the images that were shown in the experiments.

Examples of forest scenes from Stanford University 2D/3D Dataset [170], used in the analysis of registered luminance and depth Images (see chapter 5).

Figure C.7: Examples of outdoor forest scenes.

Examples of in city scenes from Stanford University 2D/3D Dataset [170], used in the analysis of registered luminance and depth Images (see chapter 5).

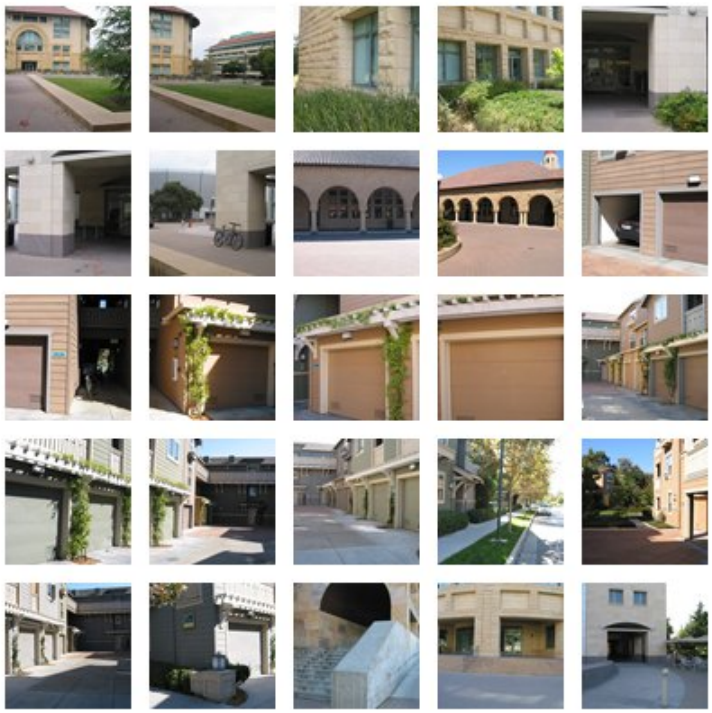


Figure C.8: Examples of outdoor in city scenes.

Theses

1. Although ubiquitous computing and seamless human-computer-interaction (HCI) require systems, which are human-centric in the sense of taking into account a user's internal states such as intentions, current goals, or the focus of attention. It comes as a surprise that it is largely absent from the most existing systems. This thesis address this shortcoming by exploring the problem of predicting human gaze behavior in order adapt smart environments to the goals of humans and their anticipated actions.

2. In smart environments users are often interrupted, manage very large quantities of information, and they switch between the contexts of different displays and tasks. In these settings, selective attention plays a fundamental role in interaction and task execution.

3. Computational modeling of the visual system was quite successful in the sense of predicting saliency maps based on image properties. However, accurately predicting eye movements remains a challenging problem in real world scenarios. The investigated predictive gaze models don't work well for eye movements prediction on the tiled Large High Resolution Displays (LHRDs).

4. The existing predictive models for eye movements do not take contextual factors into account, where it rely only on low-level 2D scene features such as color, orientation, contrast, and intensity, which is most important. However, what other features should be included? Overall, there is a need for investigating which features are most relevant in which settings.

5. These issues can be handled using a data-driven approach, where user profiles for eye movements behaviors are learned from data in different behavioral context. The machine learning model proposed by this thesis represents both different low-, mid- and high- level features and depth features together with different user behavioral context. That allows to determine the relevance of different features in different situations.

6. Characterizing the statistical properties of luminance and depths images at the center of gaze allows for better understanding of how relevance

depth features are for gaze location prediction. Thus, predictive models respecting this will ultimately outperform saliency maps computed only on the basis of 2D pixel images.

7. Machine learning models and the analysis of behavioral data show the limitations of current predictive models describing human eye movements and reveal the influences of task on gaze selection. Additionally, the relevance of different features was vary among different behavioral contexts.

8. Normative modeling of user behavior allows not only *to describe* how humans behave. But also, the model *explain* why humans behave as they do, and the role cognition plays in an account for it. Normative models are suitable for modeling user behavior, even when there is no direct data set exist that could be used to learn a policy in a supervised way. Also, in situations when the selection of actions depend on the decisions and actions of others especially when the possibility of communication with other agents are available.

9. Inverse reinforcement learning paradigm allows to construct the parameters of the learning model to best match the observed human gaze behavior. Thus, the connection between model and empirical data is made.

10. Inverse reinforcement learning models enable to automatically extract the reward function based on effective features from user eye movement behaviors. The learned reward function was able to obtain user behavior information that fulfill to predict eye movements.