Imperial College
London

PERSPECTIVE TAKING IN ROBOTS:
A FRAMEWORK AND COMPUTATIONAL MODEL

TOBIAS FISCHER

Thesis submitted for the degree of Doctor of Philosophy

Supervised by PROFESSOR YIANNIS DEMIRIS

Personal Robotics Laboratory
Department of Electrical and Electronic Engineering
Imperial College London

September 2018

COPYRIGHT DECLARATION

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution Non-Commercial No Derivatives license (CC BY-NC-ND).

Under this licence, you may copy and redistribute the material in any medium or format on the condition that; you credit the author, do not use it for commercial purposes and do not distribute modified versions of the work.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

ORIGINALITY DECLARATION

I hereby declare that this thesis and the work herein detailed, was composed and originated by myself, except where appropriately referenced and credited.

London, September 2018



Tobias Fischer

ABSTRACT

Humans are inherently social beings that benefit from their perceptional capability to embody another point of view. This thesis examines this capability, termed perspective taking, using a mixed forward/reverse engineering approach. While previous approaches were limited to known, artificial environments, the proposed approach results in a perceptional framework that can be used in unconstrained environments while at the same time detailing the mechanisms that humans use to infer the world's characteristics from another viewpoint.

First, the thesis explores a forward engineering approach by outlining the required perceptional components and implementing these components on a humanoid iCub robot. Prior to and during the perspective taking, the iCub learns the environment and recognizes its constituent objects before approximating the gaze of surrounding humans based on their head poses. Inspired by psychological studies, two separate mechanisms for the two types of perspective taking are employed, one based on line-of-sight tracing and another based on the mental rotation of the environment.

Acknowledging that human head pose is only a rough indication of a human's viewpoint, the thesis introduces a novel, automated approach for ground truth eye gaze annotation. This approach is used to collect a new dataset, which covers a wide range of camera-subject distances, head poses, and gazes. A novel gaze estimation method trained on this dataset outperforms previous methods in close distance scenarios, while going beyond previous methods and also allowing eye gaze estimation in large camera-subject distances that are commonly encountered in human-robot interactions.

Finally, the thesis proposes a computational model as an instantiation of a reverse engineering approach, with the aim of understanding the underlying mechanisms of perspective taking in humans. The model contains a set of forward models as building blocks, and an attentional component to reduce the model's response times. The model is crucial in explaining human data in congruency matching experiments and suggests that humans implement a similar attentional mechanism. Several testable predictions are put forward, including the prediction that forced early responses lead to an egocentric bias. Experimental results on the computational formalization of perspective taking also open up future possibilities of exploring links to other perceptional and cognitive mechanisms, such as active vision and autobiographical memories.

ACKNOWLEDGEMENTS

The past four years have been an incredible journey and I am grateful to my colleagues, friends and family for being part of this journey. First and foremost, I would like to thank Yiannis for his continued support and encouragement throughout the last years. All of our discussions were fruitful in many regards, progressed my research, provided me with new energy and taught me valuable lessons for the future. At the same time, I had all the academic freedom I could wish for to pursue my own goals. I look forward to continuing this great collaboration beyond my time as PhD student.

I want to thank my examiners, Professor Robert Fisher and Dr Krystian Mikolajczyk, for their thorough and valuable feedback on the thesis, which helped immensely to improve the presentation of the manuscript.

I was also very fortunate to share the lab with exceptional co-workers who also became my good friends. Hyung Jin and Max, you had been amazing mentors, thank you for sharing your knowledge and your friendship. Miguel, some people might say that I have taken your position as the lab's "grandpa" after you left. Your advice right at the beginning of my PhD have helped me a great deal and prepared me well for this role. Josh and Phil, without your help on all the hardware specifics (that I still do not quite understand) this thesis would not have been the same. Mark, Ahmed, and Antoine, thanks for many fruitful discussions and your great feedback on the thesis and my research in general. Cat, thanks for addressing my weird questions about maths and for proofreading many of my papers. I would also like to thank Fan, Theo, Martina, Yixing, Oya, Pierluigi, Vini, Urbano, Rodrigo, Ruohan, Ayse, Regina, and Dimitrios for many nice discussions, lunch breaks, squash sessions and good times outside the lab.

Moreover, it was also great having discussions and laughter with the visitors and master students in the lab: Alina, Hong Il, Jiyeoup, Young Joon, Tanichu, Vicent, and Kevin. A special thank you to Jongwon for providing the great opportunity to collaborate in visual object tracking research. Throughout the years, I have also met many nice people in the Intelligent Systems and Networks group – Rigas, Giulio, Guillermo, Juil, Mihajlo, Sara, Pamela, and Sam; I hope our paths will cross again in the future.

I would also like to acknowledge the sponsors of the research contained in this thesis. The first two years were funded by the EU project WYSIWYD,

which undoubtedly widened my research horizon. It was great to work and spend time with the other WYSIWYD researchers: Phuong, Gregoire, Clement, Ugo, Jordi, Daniel, Sock, Matej, Andreas, and Anne-Laure. Thanks to Julien for repairing the iCub over and over again. The last two years of the PhD were sponsored by two Samsung Global Research Outreach programs, which allowed me to pursue exciting research I would not have had the chance to do otherwise.

A massive thanks to Jill for all your love, support and friendship. You are a bloody ripper! You kept me going in good times and bad times, you always listened to my problems, provided great advice, taught me a great deal of idealism and so much more.

Thank you also to my friends and family all over the world, and especially to my parents. Es ist gut zu wissen, dass ihr immer hinter mir steht, mich liebt, und noch immer daran interessiert seid, an was ich arbeite – auch wenn es mir schwerfällt, dies in für Laien zu verstehendem Deutsch auszudrücken. Danke! Last but not least, a big thanks to Linde and Greggy for reading an early draft of the thesis and providing valuable feedback (and for raising such a beautiful daughter).

CONTENTS

1 INTRODUCTION 17
- 1.1 Research Questions . . . 19
- 1.2 Contributions . . . 19
- 1.3 Thesis Roadmap . . . 20

2 BACKGROUND 23
- 2.1 Perspective Taking in Robotics . . . 23
  - 2.1.1 Perspective Taking for Resolving Ambiguous Situations 24
  - 2.1.2 Perspective Taking for Language Understanding . . . 24
  - 2.1.3 Perspective Taking for Imitation Learning . . . 25
  - 2.1.4 Perspective Taking for Mental State Estimation . . . 25
  - 2.1.5 Perspective Taking for Task Understanding . . . 26
  - 2.1.6 Limitations of Previous Works . . . 26
- 2.2 Human-Robot Interaction Architectures for Robotics . . . 26
- 2.3 Perceiving Human Eye Gaze . . . 27
  - 2.3.1 Importance of Eye Gaze . . . 28
  - 2.3.2 Gaze Approximation Approaches . . . 28
  - 2.3.3 Appearance-Based Gaze Estimation Approaches . . . 29
  - 2.3.4 Dataset Collection Approaches . . . 30
  - 2.3.5 Gaze Estimation for Robotics . . . 31
  - 2.3.6 Gaze Estimation for Other Applications . . . 31
- 2.4 Perspective Taking in Humans . . . 32
  - 2.4.1 Level 1 Perspective Taking . . . 32
  - 2.4.2 Level 2 Perspective Taking . . . 33
- 2.5 On the Mechanisms of Perspective Taking . . . 33
  - 2.5.1 Automatic Perspective Taking . . . 33
  - 2.5.2 Experimental Paradigms for Perspective Taking . . . 34
  - 2.5.3 Mechanisms Underlying Level 2 Perspective Taking . . . 35
- 2.6 Computational Modeling of Perspective Taking . . . 36
- 2.7 Forward/Reverse Engineering Approaches in Other Tasks . . . 37
  - 2.7.1 Imitation Learning . . . 37
  - 2.7.2 Simultaneous Localization and Mapping . . . 38
  - 2.7.3 Object and Face Recognition . . . 38
  - 2.7.4 Visual Attention . . . 39
- 2.8 Conclusions . . . 39

3 PERSPECTIVE TAKING IN MARKERLESS ENVIRONMENTS 41
3.1 Markerless Perception . . . 43
3.1.1 Environment Mapping . . . 43
3.1.2 Object Recognition . . . 44
3.1.3 Head Pose Estimation . . . 45
3.1.4 Coordinate Transforms . . . 48
3.2 Level 1 Perspective Taking . . . 49
3.3 Level 2 Perspective Taking . . . 50
3.3.1 Level 2 Visual Perspective Taking . . . 51
3.3.2 Level 2 Spatial Perspective Taking . . . 52
3.4 Experimental Evaluation . . . 53
3.4.1 Level 1 Perspective Taking . . . 54
3.4.2 Level 1 Perspective Taking Comparison to Human Data 55
3.4.3 Head Pose Estimation . . . 55
3.4.4 Level 2 Perspective Taking . . . 57
3.5 Conclusions . . . 58

4 ICUB-HRI SOFTWARE FRAMEWORK 61
4.1 Design Principles . . . 61
4.2 Library Overview . . . 62
4.3 Knowledge Representation and Exchange . . . 63
4.4 Subsystems . . . 64
4.5 iCub-HRI Modules . . . 65
4.5.1 Perception Modules . . . 65
4.5.2 Action Modules . . . 66
4.5.3 Social Interaction Modules . . . 67
4.5.4 Tools . . . 68
4.6 Using iCub-HRI . . . 69
4.6.1 Example Usage of the Object Manipulation Subsystems 70
4.6.2 Usage within the DAC-h3 framework . . . 72
4.6.3 More Applications and Use Cases . . . 72
4.7 Conclusions . . . 73

5 GAZE ESTIMATION IN NATURAL ENVIRONMENTS 75
5.1 Architecture Overview . . . 76
5.2 Gaze Dataset Generation . . . 77
5.2.1 Eye Gaze Annotation . . . 79
5.2.2 Head Pose Annotation . . . 79
5.2.3 Coordinate Transforms . . . 80
5.2.4 Data Collection Procedure . . . 80

5.2.5 Post-Processing . . . 81
5.2.6 Annotation Accuracy . . . 81
5.3 Gaze Dataset Statistics . . . 82
5.4 Inpainting of the Eyetracking Glasses . . . 83
5.4.1 Masking the Region of the Eyetracking Glasses . . . 85
5.4.2 Semantic Inpainting . . . 85
5.5 Gaze Estimation Networks . . . 86
5.5.1 Eye Gaze Estimation . . . 87
5.5.2 Image Augmentation . . . 87
5.5.3 Training Details . . . 88
5.6 Experimental Evaluation . . . 88
5.6.1 Dataset Inpainting Evaluation . . . 88
5.6.2 Gaze Estimation Performance Evaluation . . . 90
5.6.3 Cross-Dataset Evaluation . . . 92
5.6.4 Qualitative Results and Practical Application . . . 93
5.7 Conclusions . . . 94

6 A COMPUTATIONAL MODEL FOR PERSPECTIVE TAKING 97
6.1 Motivation . . . 97
6.2 Computational Formalization . . . 99
6.2.1 Visual Perspective and Agent States . . . 99
6.2.2 Forward Model and Action Primitives . . . 100
6.2.3 Distance Metric and Control Policy . . . 102
6.2.4 Alignment Strategy . . . 102
6.2.5 Response Time and Attentional Component . . . 103
6.3 Experimental Evaluation . . . 104
6.3.1 Experimental Setup . . . 104
6.3.2 Parameter Choices . . . 105
6.4 A Model of Human Perspective Taking Mechanisms . . . 107
6.4.1 Response Time Variation with Angular Disparity . . . 108
6.4.2 Movement Congruence . . . 109
6.4.3 Posture Congruence . . . 111
6.4.4 Differences by Sex and Social Skills . . . 113
6.5 Model Predictions . . . 117
6.5.1 Forced Early Response Leads to Egocentric Bias . . . 117
6.5.2 Habituation Effects . . . 118
6.6 Conclusions . . . 121

7 CONCLUSIONS AND FUTURE WORK 123
7.1 Overview and Contributions of the Thesis . . . 123
7.2 Limitations . . . 124
7.2.1 Applying Computer Vision Methods to Robotics . . . 124
7.2.2 Object Models . . . 125
7.2.3 Extreme Head Poses . . . 126
7.2.4 A Model of Child Development . . . 126
7.2.5 Shortcut to Reverse Left/Right Judgments . . . 126
7.3 Future Directions . . . 127
7.3.1 Perspective Taking and Autobiographical Memories . . . 127
7.3.2 Taking the Perspective of Arbitrary Agents . . . 128
7.3.3 Active Vision and Perspective Taking . . . 128
7.4 Epilogue . . . 129

A ROBOTS, COMPONENTS AND SENSORS 131
A.1 Robot Operating System . . . 131
A.2 Yet Another Robot Platform . . . 131
A.3 iCub Humanoid Robot . . . 132
A.4 Pupil Labs Eyetracker . . . 133
A.5 RGB-D Cameras . . . 133
A.6 OptiTrack Motion Capture System . . . 134

B INPAINTING METHODOLOGY 135
B.1 Overall Setup . . . 135
B.2 Inpainting Network Architecture . . . 136
B.3 Training Details . . . 137

C AUTHOR'S PUBLICATIONS 139

BIBLIOGRAPHY 143

LIST OF FIGURES

Figure 1.1 Thesis roadmap 21
Figure 3.1 Overall flow of the proposed perspective taking method 42
Figure 3.2 Perspective taking setup using the iCub humanoid robot 44
Figure 3.3 Example map of a lab environment 45
Figure 3.4 Depth image normalization for head pose estimation . 48
Figure 3.5 Level one perspective taking in different scenarios . . . 54
Figure 3.6 Response time profile for line-of-sight tracing 55
Figure 3.7 Qualitative comparison of the normalized head pose algorithm and the original method 56
Figure 3.8 Horizontal error for spatial perspective taking 57
Figure 3.9 Level two perspective taking example 58
Figure 4.1 Temporal Unified Modeling Language diagram for module interaction given human speech input 68
Figure 4.2 Temporal Unified Modeling Language diagram for module interaction after hitting drive threshold 69
Figure 5.1 RT-GENE architecture overview 77
Figure 5.2 Proposed setup for recording the RT-GENE gaze dataset 78
Figure 5.3 Example images contained in the RT-GENE gaze dataset 78
Figure 5.4 3D model of the eyetracking glasses 79
Figure 5.5 Number of images per participant in RT-GENE 82
Figure 5.6 Gaze and head pose distribution in various datasets . . 83
Figure 5.7 Distance distributions in various datasets 84
Figure 5.8 Distribution of camera-to-subject distances in RT-GENE 84
Figure 5.9 Face area distributions in various datasets 86
Figure 5.10 Qualitative image inpainting results 87
Figure 5.11 Landmark extraction before and after image inpainting 89
Figure 5.12 3D gaze error on the MPII gaze dataset 91
Figure 5.13 3D gaze error on the RT-GENE gaze dataset 92
Figure 5.14 Qualitative gaze estimation results 93
Figure 6.1 Computational model's architecture overview 100
Figure 6.2 Setup for computational model experiments 104
Figure 6.3 Speed-accuracy trade-off visualization 106
Figure 6.4 Mean response time depending on the size of the attentional set 107

Figure 6.5 Response time comparison of human data and model data (angular disparity) . . . 108
Figure 6.6 Movement congruency schematic . . . 109
Figure 6.7 Response time comparison of human data and model data (movement congruence) . . . 110
Figure 6.8 Another response time comparison of human data and model data (movement congruence) . . . 110
Figure 6.9 Posture congruency schematic . . . 112
Figure 6.10 Response time comparison of human data and model data (posture congruence) . . . 113
Figure 6.11 Correlation mixing parameter / embodiment measure 115
Figure 6.12 Embodiment comparison human and model data . . . 116
Figure 6.13 Egocentric response ratio for forced early responses . . 117
Figure 6.14 Trial congruency schematic (straight body postures) . . 118
Figure 6.15 Trial congruency schematic (congruent movement) . . 120
Figure 6.16 Trial congruency schematic (incongruent movement) . 121
Figure A.1 iCub humanoid robot . . . 132
Figure A.2 Kinect v2 and Asus Xtion Pro RGB-D cameras . . . 133

LIST OF TABLES

Table 5.1 Comparison with related works on gaze estimation . . 75
Table 5.2 Comparison of gaze datasets . . . 76
Table 5.3 Face detection rate with and without inpainting . . . 90
Table 5.4 Landmark error with and without inpainting . . . 90
Table 6.1 List of action primitives . . . 101

LISTINGS

Listing 3.1 Level 1 perspective taking . . . 51
Listing 4.1 Pushing an object using iCub-HRI . . . 70
Listing 4.2 Pushing an object using KARMA (without iCub-HRI) . 71

ACRONYMS

PT	Perspective Taking
PT1	Level One Perspective Taking
PT2	Level Two Perspective Taking
RT-GENE	Real-Time Gaze Estimation in Natural Environments
HRI	Human-Robot Interaction
GAN	Generative Adversarial Network
CNN	Convolutional Neural Network
ARE	Actions Rendering Engine
IDL	Interface Description Language
UML	Unified Modeling Language
OPC	Objects Properties Collector
YARP	Yet Another Robot Platform
ROS	Robot Operating System
RGB-D	Red Green Blue-Depth (Depth Cameras)
SLAM	Simultaneous Localization And Mapping
RTAB-Map	Real-Time Appearance-Based Mapping
MTCNN	Multi-Task Cascaded Convolutional Networks

1

INTRODUCTION

In our everyday lives, we often interact with other people. Although each interaction is different and hard to predict in advance, they are usually fluid and efficient. This is because humans take many aspects into account when interacting with each other: the relationship between the interlocutors, their familiarity with the topic, and the time and location of the interaction, among many others. More specifically, humans are remarkably good at rapidly forming models of others and adapting their actions accordingly. To form these models, humans exploit the ability to take on someone else's point of view.

One particular capability of humans addressed in this thesis is perspective taking (PT), which is defined here as the ability to assume another person's visuospatial viewpoint. For example, humans often point out an object to another person if that person needs to change their point of view to perceive it. To do so, humans need to estimate the visibility of objects from another viewpoint (this is referred to as level one perspective taking, PT1). Similarly, we often provide relative spatial references from the other's perspective, e. g. "Could you please pass me the red ball on your left?" (known as level two perspective taking, PT2). These PT abilities are frequently employed in human-human interactions, with examples being joint assembly (Trafton, Schultz, Bugajska and Mintz, 2005) and collaborative wayfinding (Schwarzkopf et al., 2017), to name just two. PT therefore constitutes an essential ability in our lives and is a requirement for other fundamental abilities such as a theory of mind (Premack and Woodruff, 1978; Surtees et al., 2013*b*).

As robots are becoming part of human society, and humans prefer to interact with robots in the same way as they interact with other humans (Fong et al., 2003), this thesis examines PT abilities in robots. This has previously been shown to be crucial, as it allows robots to interact more naturally (Breazeal et al., 2006; Johnson and Demiris, 2005*a*), is required for successful cooperation (Trafton, Cassimatis, Bugajska, Brock, Mintz and Schultz, 2005), eases communication (Pandey et al., 2013) and resolves ambiguities in human-robot interactions, for instance where some objects can only be seen by one agent but not the other (Lemaignan et al., 2017).

The thesis endeavors to study PT using both forward and reverse engineering approaches. The forward engineering approach is instantiated as an artificial visual system that equips robots with PT abilities in human-robot interactions. The reverse engineering approach is instantiated as a computational model whose properties are inspired and compared to those of the human visual system. Studying PT in this bidirectional manner can uncover some of the problems inherent in the human visual system, while at the same time applying principles that have emerged from the study of the human visual system to its artificial counterpart. Both approaches are introduced below.

The artificial visual system presented in this thesis provides robots with PT abilities that work in environments that are not equipped with artificial markers without the need for prior knowledge. This is in contrast to previous PT solutions, which were only effective in constrained, artificial environments. It is shown that PT coupled with gaze estimation can be used within a software framework designed for human-robot interactions (HRIs). As large camera-subject distances are commonly encountered in HRI environments, a new gaze estimation method is introduced and validated in HRI settings.

To contribute to the discussion on the underlying mechanisms of PT in the human visual system, a computational model is presented that validates one of the accounts suggested in psychology, namely the embodied transformation account. The embodied transformation account suggests that humans employ the same body representations for physical movements and PT, with the difference being whether the movements are executed or imagined (further details are provided in Section 2.5). The computational model suggests that humans employ an attentional mechanism for PT and provides several testable predictions that can be verified in future psychological studies. The model is implemented on a simulated iCub robot to allow for a wide breadth of experiments with several dozens of replications per experiment.

The focus on the PT ability is grounded on the firm belief that PT itself is a requirement for natural human-human and human-robot interaction, and a pre-requisite for the development of other skills, such as a theory of mind (Premack and Woodruff, 1978), intention prediction (Demiris, 2007), and imitation learning (Meltzoff, 2005).

1.1 RESEARCH QUESTIONS

To summarize, the thesis aims to address the following research questions:

- How can a robot be equipped with PT abilities in markerless environments?
- How can a PT system be integrated into a cognitive architecture for HRIs?
- How can a robot accurately estimate the gaze direction of a human, taking both head pose and eye gaze into account, and can such mechanisms be learned from data?
- How can PT be modeled from a computational point of view, and how do the model's outputs compare to data from experiments with humans?

1.2 CONTRIBUTIONS

The work presented in this thesis provides the following contributions:

- It introduces the design and implementation of an **artificial visual system that is capable of solving visuospatial PT tasks in markerless scenarios**. Implementing two different mechanisms, one for each level of PT, allows a robot to reason about the spatial relationships of other agents and objects in the vicinity. This is in line with evidence from the field of experimental psychology, as will be further discussed in Section 2.4. The reasoning is done solely based on the images captured by the robot's eye cameras and an RGB-D camera mounted on top of the robot, without the need for any external sensors. The system was implemented on an iCub humanoid robot and performs visuospatial PT in real-time.
- It introduces iCub-HRI, a library for HRIs that provides the iCub robot with **components related to perception, object manipulation, and social interaction**. Since the library is modular and easily extendable, the PT system is integrated within the iCub-HRI library.
- It describes the **design of a gaze estimation method that significantly improves the accuracy of the PT system** by taking not only the head

pose, but also the eye pose into account. The gaze estimator is implemented using Convolutional Neural Networks (CNNs) that take the eye images and head pose as input.

- It proposes a novel, automated dataset collection method which addresses the problem that CNNs typically require large amounts of training data. **The novel method allows labeled data to be collected in an automated manner** as the subjects wear eyetracking glasses while being recorded. The method was used to collect a challenging dataset that is particularly suited for HRIs. This is in stark contrast to previous datasets that typically use fixation targets to annotate the gaze direction or approximate the gaze direction by the head pose. **The method prevents the CNNs from being over-fitted to users wearing eyetracking glasses by using an image inpainting method** based on Generative Adversarial Networks **to remove the eyetracking glasses** from the training images. The CNNs trained with the new dataset can be applied in a wide variety of scenarios as demonstrated in cross-dataset evaluations.

- **It proposes a novel computational model instantiating the embodied transformation account** which suggests that PT is the mental simulation of the physical movements that would be required to take the other perspective. The foundation of the proposed model is a set of action primitives that are passed through a forward model that predicts the next position given a movement. **The key finding is that the model's responses only match those of humans if an attentional component is employed**, one that favors the execution of previously employed action primitives. This allows for several predictions to be put forward for future psychological studies on PT.

Appendix C lists and describes the publications derived from this thesis.

1.3 THESIS ROADMAP

As shown in Figure 1.1, the thesis is organized over seven chapters followed by three appendices:

- **Chapter 2** provides background that is relevant for the developments reported in the thesis. Specifically, it reports on previous approaches on PT within robotics, it introduces works on appearance-based gaze

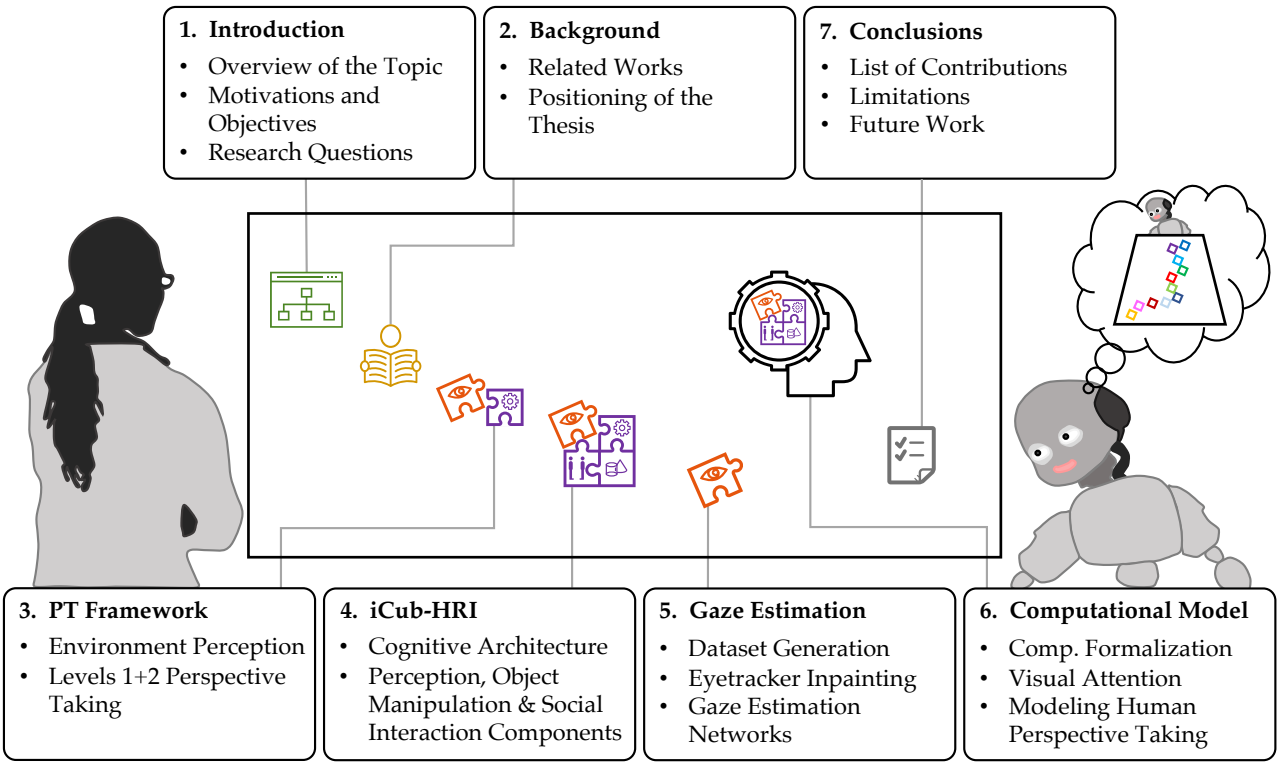


Figure 1.1: Thesis roadmap[1]. Each box indicates one chapter and the symbols indicate the relationship between the chapters.

estimation, and it provides an overview of computational modeling approaches.

- **Chapter 3** describes an artificial visual system that equips an iCub humanoid robot with the ability to perform visuospatial perspective taking in unknown environments using a single depth camera mounted above the robot, i. e. without using a motion capture system or fiducial markers. The gaze of the human is approximated using a new method for head pose estimation that relies on the depth data.

- **Chapter 4** shows the integration of the artificial visual system introduced in Chapter 3 within a cognitive architecture for the iCub to engage in a proactive, mixed-initiative exploration and manipulation of its environment.

- **Chapter 5** extends the system introduced in Chapter 3 by taking the human's eye gaze into consideration (rather than approximating the

1 The symbols contained in this figure were created by Tomas Knopp, David Carrero, Creative Stall, Irene Hoffman, Nikita Kozin, Ven Design, Artdabana@Design and Atif Arshad. They are released under the CC BY 3.0 license and are available for download on https://thenounproject.com/.

gaze with the head pose as in Chapter 3). To reach this goal, the chapter introduces an architecture that allows automatic annotation of ground truth in gaze datasets. The architecture is then used to collect a new gaze dataset, and this dataset is employed to train a deep network for gaze estimation.

- **Chapter 6** investigates possible implementations of perspective taking in the human visual system using a computational model applied to a simulated robot. The model proposes that a mental rotation of the self, also termed "embodied transformation", accounts for this ability. The computational model reproduces the reaction times of human subjects in several experiments and explains gender differences that were observed in human subjects.

- Finally, **Chapter 7** summarizes this thesis and draws conclusions based on the findings of the work reported here. The chapter highlights the importance of the research along with its limitations and discusses potential directions for future research on topics covered within this thesis.

- **Appendix A** describes the robots, components, and sensors that were used to implement the system.

- **Appendix B** provides details on the Generative Adversarial Networks that were used for the inpainting of the eyetracking glasses in Chapter 5.

- **Appendix C** contains a list of all peer-reviewed publications that resulted from the thesis and a summary of how they contribute to the thesis.

This chapter introduced the perspective taking ability that is investigated within this thesis, outlined research questions to be addressed and summarized the contributions that are provided. The next chapter will provide a detailed review of related works.

2

BACKGROUND

Various research fields inspired this thesis, including robotics, computer vision, computational modeling, and psychology. The purpose of this chapter is to review articles from these fields that are relevant to later chapters of the thesis.

The chapter is organized as follows. In Section 2.1, the issue of advanced visual perception abilities in robotics is discussed, with a particular emphasis on works that investigate perspective taking (PT) in robotics. There, it is argued that PT is a requirement for natural human-robot interaction (HRI). Section 2.2 then introduces works that embed these perception abilities into cognitive architectures for robotics. These architectures go beyond just perception and also contain components for social interactions and manipulation, and importantly their integration. Then, Section 2.3 probes the issue of estimating humans' gaze given images. It is argued that there is a research gap that needs to be bridged: combining head pose and eye gaze in HRIs with large camera-subject distances. Section 2.4 then moves the presentation to PT in humans. The distinction of level 1 and 2 PT is probed and presented along with typical experimental setups to investigate PT in humans. Section 2.5 presents an analysis of the mechanisms underlying PT. It is argued that the embodied transformation account constitutes a theory that is well supported and can be validated using computational modeling. Grounded on this finding, Section 2.6 introduces articles that present computational models of cognitive abilities, with an emphasis on articles that computationally model PT. This is followed by an overview of forward/reverse engineering approaches for other tasks in Section 2.7. Finally, Section 2.8 summarizes the chapter.

2.1 PERSPECTIVE TAKING IN ROBOTICS

This section reviews works on visual perception in robotics. A particular focus is given to works that implement aspects of PT relevant to the work presented in this thesis. The topic of joint attention, which is often considered to be a prerequisite for PT, was previously discussed by Moore et al. (2014) in human-human interactions and Nagai et al. (2003) as well as Demiris (2007)

in the robotics domain. In brief, joint attention allows the involved agents to focus on a common object, and this object then becomes subject of the PT process. Therefore, as further detailed by Moll and Meltzoff (2011*b*, who consider joint attention as level 0 PT), joint attention and perspective taking are closely linked.

2.1.1 *Perspective Taking for Resolving Ambiguous Situations*

One of the earliest works on PT for HRI is that of Trafton, Cassimatis, Bugajska, Brock, Mintz and Schultz (2005). Their robot can handle ambiguous situations where level one perspective taking (PT1) is needed; in other words, situations where the robot can see two similar objects, but one of them is occluded from the human. Kennedy et al. (2009) extend this work and show that a "like-me" simulation can solve some level two perspective taking (PT2) abilities. In a like-me simulation, the robot's reasoning capabilities from its own point of view are applied to the imagined situation of the human. Ros et al. (2010) take this idea further and resolve ambiguities by also taking an ontology of objects into account. This ontology is then used in an "I spy with my little eye" game where the robot's task is to find the correct object by asking the human partner questions.

2.1.2 *Perspective Taking for Language Understanding*

Roy et al. (2004) use a like-me simulation similar to that of Kennedy et al. (2009), but specifically target language understanding and production. Steels and Loetzsch (2009) take this idea further and present a method that allows two robots to learn a common language to describe spatial concepts. This is implemented by a turn-taking strategy whereby one of the robots describes the spatial layout of a scene to the other robot.

Lemaignan et al. (2011) and Warnier et al. (2012) extend the work by Ros et al. (2010) and integrate it with a language understanding component. While the main contribution of Lemaignan et al. (2011) is the parsing of a speech input into a symbolic representation, Warnier et al. (2012) add a temporal component that allows taking the perspective of a human based on previous as well as current object positions.

Hughes et al. (2016) and Wood et al. (2018) explore a different research direction and argue that a robot can be used to teach PT skills to autistic children. This is unusual in the sense that the robot is not the learner, as

is the case in most other works presented in this section, but instead the teacher.

2.1.3 *Perspective Taking for Imitation Learning*

Breazeal et al. (2006) use ambiguous situations in a learning scenario for humanoid robots. Their robot retains two sets of beliefs, one is for the self and one is for the other's perspective. By representing these sets probabilistically, the robot can understand the human's intent even if the demonstration is not complete, e. g. because some objects are occluded, and thus the human does not perform an action on these objects.

Similarly, Johnson and Demiris (2007) have shown that retaining a separate set of beliefs for the other's perspective can also be used to model dynamic environments. More specifically, the self predicts the effect of actions on the visual perception of the other, which is taken into account when imitating the other. This also allows the self to adapt to changing environments where the applicability of actions changes over time, for example when a third person moves objects.

2.1.4 *Perspective Taking for Mental State Estimation*

Johnson and Demiris (2005*b*) have conducted a study where an internal simulation of possible motor commands is used to gain insight into the mental state of another robot, rather than that of a human. Their robot determines the applicability of models from the other's perspective using a list of coupled inverse and forward models[1]. That is, the probabilities of models which cannot be applied by the other robot are reduced. One possible application of this work is in increasing the accuracy of action recognition (Johnson and Demiris, 2005*a*).

Winfield (2018) demonstrate that such simulation-based approaches can model a variety of experiments, ranging from imitation to narrative storytelling. Akkaladevi et al. (2016) and Devin and Alami (2016) have shown that estimating mental states of humans can be used to find the most appropriate manipulation of objects while taking the human's preferences into account.

1 A forward model predicts the next state given the current state and an action. An inverse model takes the current state and desired states as input, and outputs the required action to reach this desired state.

2.1.5 *Perspective Taking for Task Understanding*

Pandey et al. (2013) focus on human-robot interactions and teach a robot what it means to make an object visible or accessible. This needs PT abilities as the reachability of an object has to be determined from the other's perspective. One of the key contributions is that their robot can detect the effort of a human to reach an object, from "no effort needed" to "whole body effort". While Pandey et al. (2013) focus on close distance scenarios, Sisbot et al. (2007) use similar concepts in a path planning scenario and argue that the robot should take the human's path and preferences into account.

For other similar works, the reader is referred to a recent review on the importance of PT in HRIs by Lemaignan et al. (2017). This review argues that one of the most important reasons for the need for PT in robots is that humans frequently change perspectives when describing locations.

2.1.6 *Limitations of Previous Works*

The previously mentioned works all demonstrate the importance and impact of PT in robotics and more specifically HRI. However, as the environment in these works is highly constrained, these works do not meet the outlined objective of HRI in natural scenarios. Pandey et al. (2013); Ros et al. (2010); Warnier et al. (2012) and Lemaignan et al. (2011, 2017) use motion capture systems to detect humans and objects. While this provides accurate location information, motion capture systems are expensive and need precise calibration. Also, as the environment must be known in advance, the applicability of these systems is limited.

The object detection of Trafton, Cassimatis, Bugajska, Brock, Mintz and Schultz (2005) and Kennedy et al. (2009) relies on color blob segmentation, and thus can only detect a small number of objects, which have to be uniformly colored. Furthermore, their system only takes the human's body direction into account, but not the eye gaze of humans. Johnson and Demiris (2005*a*, 2007) rely on optical markers to compute the poses of the target robot and objects. Chapter 3 presents algorithms to overcome these constraints and introduces a framework that applies to natural environments.

2.2 HUMAN-ROBOT INTERACTION ARCHITECTURES FOR ROBOTICS

Having shown the importance of equipping robots with PT abilities, the discussion now moves to architectures tailored towards HRIs. The focus is

on architectures that allow the integration of various components, including those for perception and action, with the aim of providing a platform for HRI studies.

First, the concept of a robotics middleware is introduced. A middleware serves multiple purposes. It provides a means of convenient communication between distributed components, whereby one component might be responsible for object detection, another one for perspective taking, yet another for grasping and so forth. Middleware also often provides interfaces for sensors such as cameras and motor encoders. In their review article, Elkady and Sobh (2012) provide an excellent overview of robotics middleware.

Appendices A.1 and A.2 introduce the Robot Operating System (ROS) and Yet Another Robot Platform (YARP) middleware as they have been used heavily in this thesis. The low-level control of the iCub relies on the YARP middleware, while ROS has been used for implementation of most components presented within this thesis and is being adopted rapidly by more and more researchers.

Indeed, several works are introducing HRI related frameworks based on ROS. For example, Jang et al. (2015) propose a framework where modules concerned with low-level control and service logic are separated from modules concerned with social behaviors. Lane et al. (2012) present a bundle of ROS modules which allows the extension of existing projects for speech recognition, natural language understanding, and basic gesture recognition as well as gaze tracking. Krupke et al. (2017) present a toolkit which allows the evaluation of human-robot interactions in virtual reality environments and subsequent deployment on a real robot. The robot behavior toolkit (Huang and Mutlu, 2012) is based on findings within the social sciences to allow for more natural robot behavior. Finally, Sarabia et al. (2011) present a framework allowing the perception of the actions and intentions of humans and show its application in a social context where a robot imitates the dance movements of a human.

2.3 PERCEIVING HUMAN EYE GAZE

This section first highlights the importance of taking the human's eye gaze into account, followed by an overview of gaze estimation techniques within computer vision.

2.3.1 *Importance of Eye Gaze*

There is a wide range of works supporting the importance of eye gaze within human-human and human-robot interactions. One of the earliest reports on the importance of eye gaze in social interactions between humans is that by Kendon (1967) who argues that gaze serves as a regulating signal indicating when the speaker and listener change roles. Kennedy et al. (2015) show experimentally that the head pose should not be used as a proxy for eye gaze when measuring the attention of children in children-adult interactions.

Boucher et al. (2012) investigate the influence of gaze on action recognition and have shown that response times are impaired when the eyes of the person performing the action are occluded. However, the gaze is not only being directed at the other person but also at the object that is manipulated. Falck-Ytter et al. (2006) have shown that this is the case even for infants (12-month-olds).

Admoni and Scassellati (2017) show that not only the human's gaze direction is of importance, but also the robot's gaze. They provide guidelines regarding the question of where the robot should direct its gaze during HRI.

2.3.2 *Gaze Approximation Approaches*

Before moving the discussion onto gaze estimation methods that take both head and eye pose into account, some compelling works on the approximation of gaze in other ways are presented.

Chamveha et al. (2013) suggest using the walking direction of a human as a very rough approximation of the gaze direction in low-resolution and far distance scenarios. While this estimate is somewhat accurate in outdoor scenes, it fails to take head pose or eye gaze into account and does not provide gaze estimates for humans that are static.

Mukherjee and Robertson (2015) fuse two separate networks for the RGB and depth modalities. However, as the inputs are low-resolution images, the eye pose is not taken into account. Their main contribution is a probabilistic attention metric that allows both gaze estimation and interaction detection by modeling a spatial probability distribution of the gaze. If the attention of one person is focused on the other person's head and vice versa, these two people are assumed to be interacting.

Recasens et al. (2015) present a method that allows 2D gaze following in images. Their primary motivation is that objects are a strong indicator of where people tend to look, and their Convolutional Neural Network (CNN)

therefore consists of a gaze pathway and a saliency pathway. Park et al. (2013) take a similar approach, where the group members in a social scene are considered to be salient. While these saliency-based approaches work well in many situations, they are limited as gaze does not always fall on the most salient objects.

2.3.3 *Appearance-Based Gaze Estimation Approaches*

Funes-Mora and Odobez (2016) propose a gaze estimation method applied to RGB-D images. The idea is to fit a morphable face model to an image, and then estimate the gaze from the fitted parameters. However, the performance is affected if no person-specific face model is available. The method is also constrained to scenarios where the subjects are facing towards the camera and are within a close distance.

In recent years, Zhang et al. (2015) have shown that deep learning methods such as CNNs can be used to find the mapping between the eye image and the corresponding gaze angle. Their method relies on a single eye image as input. Cheng et al. (2018) have shown that it is beneficial to estimate which eye image (left/right eye image) results in better gaze estimation performance (rather than randomly choosing the eye), and then use this eye image to train a gaze estimator. In these methods, the head angle is not estimated by the CNN, but instead appended to one of the fully connected layers.

Krafka et al. (2016) have shown that using a CNN also to estimate the head angle is beneficial. Their CNN estimates the gaze by combining the left eye, right eye and face images, along with a face grid which provides the network with information about the location and size of the head within the original image. Zhang et al. (2017) take this idea further and show that the gaze direction can be directly estimated from the face image by learning a distribution that encodes the importance of the facial areas. In other words, their CNN learns that the eye region is most important for gaze estimation itself rather than explicitly providing the eye region as separate input.

An alternative approach has recently been proposed by Park et al. (2018), who introduce an intermediate "pictorial" representation rather than directly regressing the gaze from an eye image. The pictorial representation consists of iris and eyeball maps, and a lightweight CNN with relatively few parameters is trained to regress the gaze from the pictorial representation.

The reviewed methods are limited as they can only capture the gaze on a phone, tablet or laptop, rather than in a free-viewing environment. Ac-

cording to Lu et al. (2015), this is partly due to the training images, which typically contain images of subjects that face a screen. In free-viewing settings, head motion changes the eye appearance drastically, and thus methods that are trained with images from screen-based settings fail to generalize for wide head pose ranges. This prevents these methods from being used in free-viewing scenarios that are targeted in Chapter 5 of this thesis. For this reason, the following section introduces the most common approaches to collecting eye gaze datasets.

2.3.4 *Dataset Collection Approaches*

As manual labeling of the gaze is a tedious task, most gaze datasets are captured with the subject looking at pre-defined targets on a screen. In the Columbia Gaze dataset (Smith et al., 2013), the subjects are recorded with their head placed on a chin rest and asked to fixate on a dot displayed on a wall. This leads to severely limited appearances: the camera-subject distance is kept constant, and there are only a small number of possible head poses and gaze angles. The UT Multi-view dataset (Sugano et al., 2014) contains recordings of subjects with multiple cameras, which makes it possible to synthesize additional training images using virtual cameras and a 3D face model. Deng and Zhu (2017) propose a similar setup, where extreme head pose angles are contained by first displaying a head pose target, followed by an eye gaze target.

Zhang et al. (2015) present the MPII Gaze dataset where target items are displayed on a laptop screen in home environments at different times of the day, which increases the variations in appearance. Eyediap (Funes Mora et al., 2014) contains not only gaze targets on a computer screen, but also a 3D floating target which is tracked using color and depth information. The dataset with the largest number of subjects may be GazeCapture (Krafka et al., 2016). It is a crowd-sourced dataset of nearly 1500 subjects looking at gaze targets on a tablet screen. Huang et al. (2017) have taken a very similar approach and present the TabletGaze dataset.

A feature that all of these datasets have in common is that the head pose is estimated using landmark positions of the subject and a generic or subject-specific 3D head model. While these datasets are highly useful when a subject is directly facing a screen or mobile device, the camera-subject distance is relatively small, and the head pose is biased towards the screen. In comparison, datasets that favor highly accurate head pose estimation at larger distances typically do not contain eye gaze labels (Baltrusaitis et al., 2012;

Fanelli et al., 2013; Fisher, 2004; Mukherjee and Robertson, 2015). Some of these limitations are overcome by the proposed algorithms to automatically annotate the gaze as presented in Section 5.2.

Another way of obtaining annotated gaze data is by creating synthetic image patches (Lu et al., 2015; Wood et al., 2016, 2015). For example, Wood et al. (2016) propose a method to render photo-realistic images of the eye region. This has the advantage that arbitrary head poses and gazes can be created. However, the synthetic images do not look the same as real images, and thus a domain gap exists. Shrivastava et al. (2017) propose the use of Generative Adversarial Networks to refine the synthetic patches to resemble more realistic images while ensuring that the gaze direction is not affected.

2.3.5 *Gaze Estimation for Robotics*

Several works investigate gaze detection for application in robotics. Schillingmann and Nagai (2015) enable an iCub humanoid robot to detect a partner's gaze by combining head and eye features. The head pose is determined by finding landmark positions of the face and mapping them to a 3D face model (see Section 5.5 for more details). The pupil position is determined based on the contrast of the white region around the pupil and the pupil itself. However, the method is constrained to people who sit approximately one meter from the robot. Palinko et al. (2015) present a calibration-free method for gaze tracking. Their method is similar to that of Schillingmann and Nagai (2015); however the focus of their work is in tracking the gaze to find the object the human is attending to. Furthermore, by detecting when the human is looking straight at the robot, their robot can benefit from turn-taking behavior.

2.3.6 *Gaze Estimation for Other Applications*

Within this thesis, gaze estimation is used as a component within a PT framework. However, there is a wide range of other applications. Kar and Corcoran (2017) recently provided a thorough review, and thus the presentation here is limited to a few works. Parks et al. (2015) show that gaze estimation can be used to improve the fixation estimates in an eyetracking study. Thirty subjects were asked to view images containing at least one face on a computer screen. Combining the gaze estimates with saliency information significantly outperforms the individual models only focusing on gaze or saliency. Koutras and Maragos (2015) have applied Gaussian Mixture Models to sign

language videos. There, the gaze plays a significant role in detecting the change of prosody. Müller et al. (2018) present a method to detect eye contact in social interactions using both gaze and speaking behavior. Their setup relies on two cameras behind each participant so that every other participant can be seen from each participant's viewpoint. Vasudevan et al. (2018) exploit that humans gaze at objects they describe to increase the performance within the object referring task (localizing the target object in videos given a language description).

2.4 PERSPECTIVE TAKING IN HUMANS

The discussion now moves towards PT in humans. This section summarizes findings from psychology that are generally agreed upon, and briefly discusses some neurophysiological studies.

It is well documented that humans rely on two different levels of PT, which are task-specific and are likely to have different underlying mechanisms (Flavell et al., 1981; Michelon and Zacks, 2006). For both levels, we distinguish the visual and spatial dimension.

2.4.1 *Level 1 Perspective Taking*

PT1 emerges in children at around two years of age (Moll and Tomasello, 2006) and comprises the ability to identify objects which are occluded from one perspective but not the other (visual dimension), as well as the ability to infer whether an object is in front of or behind the other agent (spatial dimension). Yaniv and Shatz (1990) and Michelon and Zacks (2006) have suggested that this is achieved by tracing a line-of-sight between the other agent and the target object.

Experimentally, PT1 is frequently investigated using tasks where the perspective taker views a scene of another human that is facing a wall (Qureshi et al., 2010; Ramsey et al., 2013; Todd et al., 2017). The most straightforward task is to decide whether a dot can be seen by the other human (which is the case if the object is drawn on the wall in front of the human) or if it is occluded (object is located on the wall behind the human). This task can be extended in various ways, for example by asking the perspective taker how many objects can be seen by the other human. For PT1, Michelon and Zacks (2006) have shown that the response time increases linearly with the distance between the other human and the target object.

2.4.2 *Level 2 Perspective Taking*

PT2 is developing between three and five years of age (Moll and Meltzoff, 2011*a*) and refers to understanding *how* the object is perceived from the other perspective (rather than just understanding *what* is visible from that perspective; Michelon and Zacks, 2006).

Experimentally, one way to investigate PT2 is as follows. The perspective taker is viewing a scene containing another human situated around a table. Various objects are located on the table, and the perspective taker's task is to decide whether the target object is to the left or the right as perceived by the other human (spatial dimension, see Kessler and Rutherford, 2010 and Kessler and Wang, 2012). This experimental setup is particularly easy to replicate and has thus been used in Chapter 6. An example of visual PT2 is estimating how a numeral appears from the perspective of another human (Surtees et al., 2013*b*).

The "own-body transformation" task (Blanke et al., 2005) should also be mentioned, where the perspective taker has to decide whether the indicated hand of a human is the left or right hand. However, as this thesis focuses on multi-person interactions, this approach is not detailed further.

For PT2, some form of mental rotation seems to be employed. The precise mechanisms are still under debate and are discussed in the next section.

2.5 ON THE MECHANISMS OF PERSPECTIVE TAKING

This section introduces several accounts on the mechanisms of PT. As reviewed in Section 2.4.1, it is now well accepted that line-of-sight tracing underlies PT1. However, there is extensive debate whether PT1 is automatic (i. e. involuntarily/spontaneous), and the arguments in favor and against this proposal are discussed. This is followed by the presentation of two rivaling proposals for mechanisms underlying PT2, namely the sensorimotor interference and embodied transformation accounts.

2.5.1 *Automatic Perspective Taking*

Schurz et al. (2015) and Surtees et al. (2016) report that subjects automatically take the perspective of others' even when it is not required for the given task. Schurz et al. (2015) employ the object counting experimental paradigm introduced in Section 2.4.1 and show that the subjects' reaction time increases if the other human sees a different number of objects. While Schurz et al.

(2015) employ a PT1 task, Surtees et al. (2016) and Elekes et al. (2017) argue that participants automatically take the perspective of others in a PT2 task, but only if the subject is actively collaborating with others.

This view is currently under debate. Cole et al. (2016) modify the stimuli such that the avatar cannot see any objects in some trials. These trials show the same effect on the response times, and hence it is argued that subjects do not automatically take others' perspectives in PT1 tasks. The same effect is shown by Santiesteban et al. (2017) who replace the other human with an arrow in a PT2 task. In this case, the effect on response times does not change either, which provides evidence against automatic PT2.

2.5.2 *Experimental Paradigms Investigating Automatic Perspective Taking*

While the following two studies (Qureshi et al., 2010 and Todd et al., 2017) were conducted to contribute towards the discussion on automatic PT, they are mentioned here because of their interesting experimental paradigms, namely execution of a secondary task (Qureshi et al., 2010) and forced early responses (Todd et al., 2017). The forced early response paradigm is further investigated in Chapter 6.

Qureshi et al. (2010) have introduced a study on PT1 while executing a secondary task. The primary task was a PT1 task, while the secondary task was responding to an auditory stimulus. Qureshi et al. reported that the simultaneous execution of the secondary task overall increases the processing cost (i. e. increased reaction times), but more so when there is an inconsistency between perspectives. This led to the conclusion that the executive processes are only involved in the perspective selection (between the self and the other perspective), but not in the actual perspective calculation. In other words, the other's perspective is still (automatically) calculated despite the additional task.

Todd et al. (2017) have shown the same effect in a study where subjects were forced to give an early response. It was shown that this does not affect the automatic processing of the avatar's perspective, but the controlled (task-related) processing, which in this study was to report on the self-perspective in inconsistent trials. In other words, an early forced response leads to disruption of the correct perspective selection. Section 6.5 discusses the presented computational model's predictions when an early response is forced in a PT2 task.

2.5.3 *Mechanisms Underlying Level 2 Perspective Taking*

There is extensive debate about the mechanisms involved in level two perspective taking (Alsmith et al., 2017; Kessler and Thomson, 2010; May, 2004; Surtees et al., 2013*b*). While in general a mental self-rotation seems to be employed, Kessler and Thomson (2010) and Janczyk (2013) argued that a mental self-rotation is only invoked at angular disparities between the perspective taker and the other human that are larger than 60 degrees. For smaller angular disparities, they suggest that a visual matching process is invoked. Similarly, if the perspective taker is located directly opposite the other human, Gardner et al. (2013) propose that left and right are simply swapped. Vander Heyden et al. (2017) have shown that children indeed employ this strategy.

Concerning the mental self-rotation, various mechanisms have been proposed. The two most prominent ones are the sensorimotor interference account (May, 2004) and the embodied transformation account (Kessler and Thomson, 2010). There is particular emphasis on the embodied transformation account as the computational model presented in Chapter 6 is based on this account.

The sensorimotor interference account suggests that mental rotations are difficult due to the conflicts of spatial information that emerge when a simulated perspective is taken (May, 2004). More specifically, May (2004) suggests that objects have to be represented from the self-perspective as well as the other's perspective. For instance, when selecting the correct response in the left-right task as introduced in Section 2.4.2, the two representations compete with each other. When additional reference frames, such as the body frame and head frame, are taken into account, the sensorimotor interference account postulates that these introduce additional sources of interference.

The embodied transformation account states that humans employ egocentric encoding when taking others' perspectives, i. e. they mentally rotate the self into the target posture (Kessler and Rutherford, 2010). This process uses the body representations of the self and the corresponding forward model. In other words, the body representations used for PT are the same as the ones used for actual physical movement of the body, whereby physical movements are being executed and perspective taking emerges as an emulation of movement. It is not required to estimate the other's forward model, as the direction of alignment is from the self to the other (rather than vice versa).

Surtees et al. (2013*a*) have shown that the embodied transformation account is valid for both visual and spatial PT2 tasks. As a reminder, an ex-

ample of a visual PT2 task is estimating how a numeral appears like from the other's perspective, while an example of a spatial PT2 task is judging whether an object is to the left or right of the other. Finally, besides the evidence from psychological studies (Kessler and Thomson, 2010; Surtees et al., 2013*b*; Watanabe, 2016), Wang et al. (2016) and Gooding-Williams et al. (2017) present neurophysiological data that provides further support for the embodied transformation account. These studies suggest that a specific brain area, namely the right posterior temporoparietal junction, implements the transformation into another perspective. The interested reader is referred to Bukowski (2018) for a recent review of the neural correlates that underpin PT.

2.6 COMPUTATIONAL MODELING OF PERSPECTIVE TAKING

The discussion now moves to previous attempts at modeling perspective taking and related cognitive abilities. Schrodt et al. (2015) introduce a model that first learns to correlate visual and proprioceptive data of motion sequences. The motion patterns are assumed to be observed from four viewpoints, i. e. egocentric, left, right, and opposite viewpoints. The perspective invariance is achieved by minimizing the error between the observed motion and the closest of the motion observations at training time. However, perceiving motion (including proprioceptive information) from multiple viewpoints during training time is not a realistic assumption as proprioceptive information is only available from the self-perspective. Furthermore, as the input to the model are motion signals, merely observing others' static postures is not sufficient for perspective taking, which is contrary to human perspective taking abilities.

Ogata et al. (2009) introduced a similar model for the prediction and imitation of motion sequences. In the first step, the model learns a mapping between self-motions and movements of an object. It is assumed that a teacher imitates the robot from four different viewpoints (as in Schrodt et al., 2015, they use the egocentric, left, right, and opposite viewpoints). The model then learns "conversion modules" that transform the input so that the forward model trained from the egocentric perspective can be used to predict the other's motion. Finally, the model learns to select the conversion module that most accurately predicts the motion of the other, which also allows the imitation of the teacher. Nakajo et al. (2015) extend this work (Ogata et al., 2009) by separately representing the viewpoints and motion sequences,

which allows imitation of known actions from a viewpoint that has not been observed previously.

Gentili et al. (2015) follow a similar approach, with a focus on the imitation of another person observed from an arbitrary viewpoint. The visual input is transformed to match the self-perspective using a concatenation of rotation matrices. Subsequently, actions can be recognized and imitated using self-learned inverse models. Lopes and Santos-Victor (2005) use a similar approach to mimic the arm movements of a human.

Recently, Duran and Dale (2016) proposed a computational model where an agent considers multiple perspectives at any given time. Contextual cues, as well as processing constraints, are employed to determine the currently "salient" perspective in a probabilistic manner. Using contextual cues is an appealing approach as this allows the modeling of individual differences when taking others' perspectives. Despite the modular nature of this approach, it severely depends on rather ad-hoc assumptions about the training sequence. For example, all observations are assumed to be egocentric, followed by introducing other-centric training samples, and finally samples of virtual agents. Furthermore, it is not clear how the approach scales to a larger number of object and agent locations (currently, there are four discrete locations for the object and agent).

2.7 FORWARD/REVERSE ENGINEERING APPROACHES IN OTHER TASKS

The forward/reverse engineering approach that is taken in this thesis has been used to study several other tasks besides PT. This section presents four prominent examples, namely imitation learning, simultaneous localization and mapping, object and face recognition, and visual attention. Cox and Dean (2014) provide an excellent review that contains further examples of forward/reverse engineering approaches, with a focus on the interplay between neuroscience and computer science.

2.7.1 *Imitation Learning*

Demiris and Hayes (2002) propose an architecture for the imitation of movements. Their architecture consists of two routes: one where the motor system is only involved when reproducing the posture that is to be imitated (passive route), and another where the motor system is also involved in the perception of the posture (active route). They suggest that the passive route is particularly well suited for learning new movements. On the other hand, the

active route is better suited for imitating already known movements because the next posture can be predicted. They show that the combination of both routes shares properties with the data observed from studies with monkeys. Based on the model properties, Demiris and Hayes (2002) propose that the neurons involved in the imitation are only then active if the demonstrator's movement is performed at speeds attainable by the monkey.

2.7.2 *Simultaneous Localization and Mapping*

Milford and Wyeth (2008) show that a computational model of the hippocampus of rats can be used for Simultaneous Localization and Mapping. From an engineering perspective, their model is compelling as it allows large-scale mapping over a relatively long time frame. They found that in order to deal with ambiguous visual inputs, it is advantageous to evaluate several competing hypotheses simultaneously. Hence, so-called grid cells were introduced within their model. Later, Milford et al. (2010) showed that the rat's brain indeed has similar cells for reducing sensory uncertainty.

2.7.3 *Object and Face Recognition*

David Cox and coworkers study object and face recognition in biological and artificial visual systems. In one line of work, they investigate the responses of the visual system in the human brain when stimulated with various objects (Cox et al., 2004). They argue that contextual cues increase the object recognition performance immensely. In another line of work, these findings are then applied in an object recognition task (Pinto et al., 2008). This work shows that a computational model that implements processes similar to the primary visual cortex in primates perform favorably against state-of-the-art methods in a range of datasets. Interestingly, however, the same model fails to perform well in evaluations that use simple synthetic images. DiCarlo and Cox (2007) argue that one reason for this is that the human brain is optimized towards transforming the visual input into representations that can be used by relatively simple decision functions – rather than applying complex decision functions on "non-optimized" representations.

2.7.4 *Visual Attention*

An impressive example of the forward/reverse engineering approach is the Selective Tuning model by Tsotsos et al. (1995). Selective Tuning represents an artificial visual system that goes beyond visual perception and also addresses the question of how control should be implemented to maximize the information that is acquired at the next fixation. Tsotsos (1990) argues that the biological visual system has to perform approximations, as even "simple tasks" such as visual search are computationally intractable. Selective Tuning matches primate data in a variety of tasks and offers a wide range of testable predictions. For example, Rothenstein and Tsotsos (2014) show that the firing rate of neurons in several layers of the visual hierarchy matches that of the primate visual system. They argue that the neural modulation should first occur in higher layers of the visual system, rather than the lower layers, as is proposed in most other models (such as Desimone and Duncan, 1995; Reynolds et al., 1999; Reynolds and Heeger, 2009). For a thorough discussion, the reader is referred to Tsotsos (2011).

2.8 CONCLUSIONS

This chapter presented empirical and theoretical works that investigate PT. It was demonstrated that PT is an important component for HRI. This thesis introduces an approach to expand the applicability of PT in robotics beyond constrained, known environments that rely on artificial equipment, as is required by the majority of prior works. The chapter also highlighted the importance of eye gaze and argued that it had been omitted in many cases due to the challenges encountered by large camera-subject distances.

Since one of the aims of this thesis is to model human PT, the discussion then moved to works investigating PT in humans from psychological and neurophysiological perspectives. The most prevalent works were introduced, and it was argued that the embodied transformation account is well suited for the implementation of a computational model, which could potentially shed light on the underlying mechanisms of this theory.

Previous computational models of PT were subsequently introduced, but it was found that PT is often achieved by considering multiple viewpoints simultaneously at training time, which is not a realistic requirement for embodied agents. Finally, works that follow a similar forward/reverse engineering approach for other tasks were presented, and it was argued that they advanced their respective fields considerably.

Further background is contained within the following chapters, where a more specific conceptualization for the addressed research challenges is necessary. The following chapter starts by introducing the required components of a robotic system that allows PT in natural environments. Inspired by the reviewed literature, line-of-sight tracing was implemented for PT1, and a mental rotation process is employed for PT2.

3

ALGORITHMS AND SOFTWARE FOR PERSPECTIVE TAKING IN MARKERLESS ENVIRONMENTS

This chapter addresses the first research question:

> "How can a robot be equipped with perspective taking abilities in markerless environments?"

This chapter shows that it is feasible to equip robots with perceptional abilities for perspective taking (PT) in markerless environments using only the robot's cameras and an RGB-D camera. Several perception algorithms are in place, whose outputs are used by two separate PT mechanisms to solve level 1 and level 2 perspective taking tasks. Figure 3.1 illustrates an overview of the proposed method. The iCub humanoid is used as a robotic platform (see Appendix A.3).

The robot's perception is split into three algorithms. Firstly, a state-of-the-art visual Simultaneous Localization and Mapping (SLAM) algorithm (Labbé and Michaud, 2018) is used to map the environment, so no prior information of the environment needs to be known. Secondly, a deep learning-based algorithm is employed for real-time object recognition (Pasquale et al., 2015), such that no markers are needed. Thirdly, a new head pose estimation algorithm is proposed to approximate the gaze of the human. For this, a state-of-the-art method based on random regression forests (Fanelli et al., 2013) is extended using normalized depth images, which results in the increased robustness of the algorithm. Section 3.1 details the three different algorithms.

Grounded in the psychological studies introduced in Chapters 1 and 2 (Flavell et al., 1981; Michelon and Zacks, 2006), PT is separated into two processes. Section 3.2 describes the implementation of a line-of-sight tracing algorithm for level one perspective taking (PT1), and Section 3.3 introduces a mental rotation algorithm for level two perspective taking (PT2). Section 3.4 evaluates this perspective pipeline with several experiments. Finally, Section 3.5 summarizes and concludes this chapter.

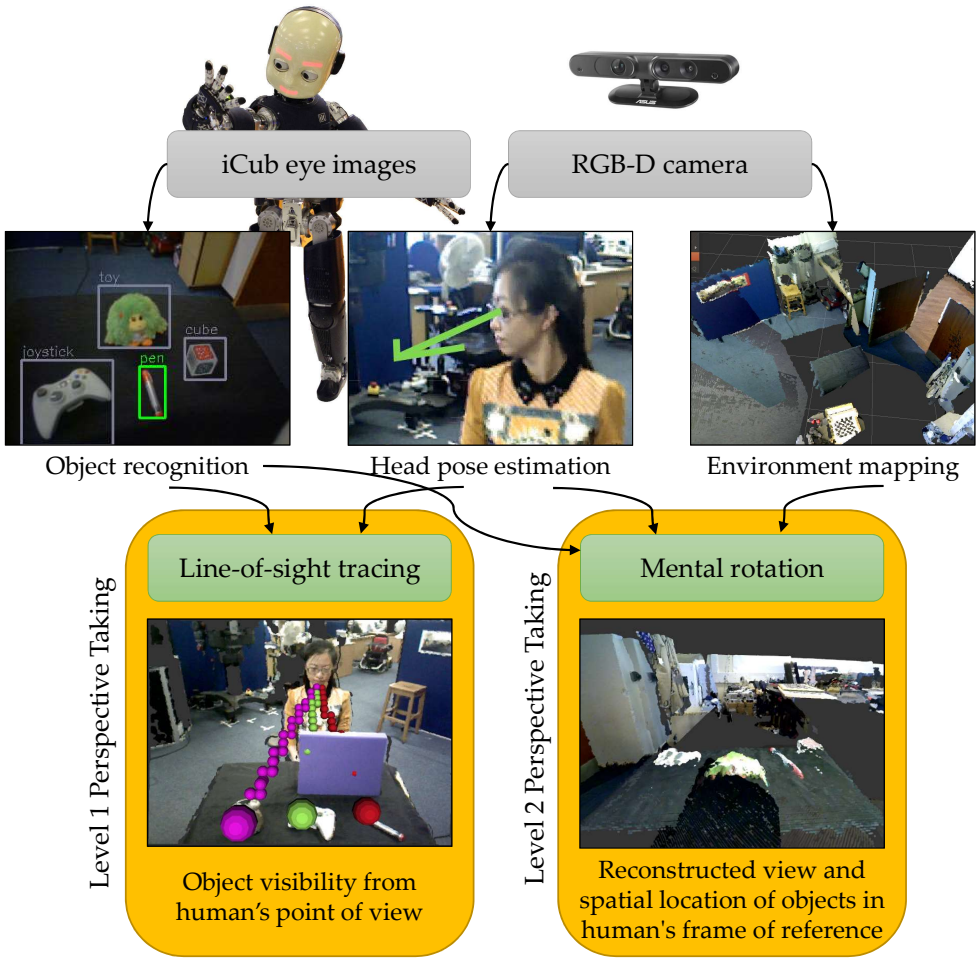


Figure 3.1: Overall flow of the proposed method[1]. The inputs to the perspective taking pipeline are images acquired from an RGB-D camera and the iCub eyes. In the first step, the robot recognizes objects, estimates the head pose of surrounding humans, and maps the environment. Two separate processes are employed for PT1 and PT2; allowing the robot to infer which objects are seen by the human, what the spatial location of these objects are in the reference frame of the human, and how the world appears from the human viewpoint.

Research from this chapter has been previously published in Fischer and Demiris (2016) and contributed to Moulin-Frier, Fischer et al. (2018), as well as Zambelli et al. (2016).

1 The figure of the iCub was originally taken by Xavier Caré (https://commons.wikimedia.org/wiki/File:ICub_Innorobo_Lyon_2014_debout.JPG, licensed under the CC BY-SA 3.0 license) and was modified to remove the background. The photo of the RGB-D camera was taken by Pierre Lecourt and is under the CC BY-NC-SA 2.0 license (originally from https://flic.kr/p/e52Lxq). The modified figure is available under the CC BY-NC-SA 4.0 license (with kind authorization from Xavier Caré to use this license).

3.1 MARKERLESS PERCEPTION OF THE ENVIRONMENT

As discussed in Section 2.1.6, previous works are constrained to environments equipped with markers. The goal here is to estimate the perceived world of humans and the surroundings of the robot, whilst not constraining the environment. To this end, only cameras mounted on the robot are used, namely the iCub eye cameras as well as a low-cost RGB-D camera.

The cloud points $p_k^C \in \mathbf{P}^C$ represent the point cloud $\mathbf{P}^C$ acquired by the RGB-D camera. The superscript C denotes that the cloud points are expressed in the reference frame $\mathbf{F}_C$ of the RGB-D camera.

Let $\boldsymbol{\Omega}^R = \{(\boldsymbol{\omega}_1^R, c_1), \ldots, (\boldsymbol{\omega}_N^R, c_N)\}$ be a set of objects. Then, $\boldsymbol{\omega}_i^R \in \mathbb{R}^3$ denotes the object location in the robot coordinate frame $\mathbf{F}_R$, N is the total number of objects perceived by the robot, and c_i contains the corresponding object class to object i.

The set $\mathbf{H}^C = \{\mathbf{h}_1^C, \ldots, \mathbf{h}_M^C\}$ contains the head poses $\mathbf{h}_j^C \in \mathbb{R}^6$ of the M humans interacting with the robot. Each $\mathbf{h}_j^C$ contains the position of the j^{th} head in the RGB-D camera coordinate frame, and the corresponding head orientation in yaw, pitch, and roll notation.

The elements o_{ij} of matrix $\mathbf{O}$ with size $N \times M$ store the perception of object i by agent j. Each $o_{ij} \in \mathbf{O}$ is a 3-tuple $(\boldsymbol{\omega}_i^{H_j}, S_i^{H_j}, L_i^{H_j})$, where $\boldsymbol{\omega}_i^{H_j}$ is the i^{th} object in reference frame $\mathbf{F}_{H_j}$ of the j^{th} human, $S_i^{H_j} \in \{\text{visible, occluded}\}$ describes whether the i^{th} object is in sight of the j^{th} human, and similarly $L_i^{H_j} \in \{\text{left, central, right}\}$ encodes the spatial location, i. e. whether an object is left, central or right as seen from the human perspective.

The following convention was used with regards to the coordinate frames. Positive values on 1) the x-axis point forwards, 2) the y-axis point to the left, and 3) the z-axis point upwards (see Figure 3.2). Distances are described in meters and angles in degrees.

3.1.1 *Environment Mapping*

Real-Time Appearance-Based Mapping (RTAB-Map; Labbé and Michaud, 2018) was chosen to map the environment, as it has two advantages over typical visual SLAM methods. Firstly, RTAB-Map can meet real-time constraints even for large maps, which is vital to allow robots to operate in complex environments. Secondly, RTAB-Map not only makes use of the RGB-D images but optionally also of the odometry and laser information provided by the mobile base of the iCub. This sensor fusion ability prevents the loss of odometry in the case of fast camera movements.

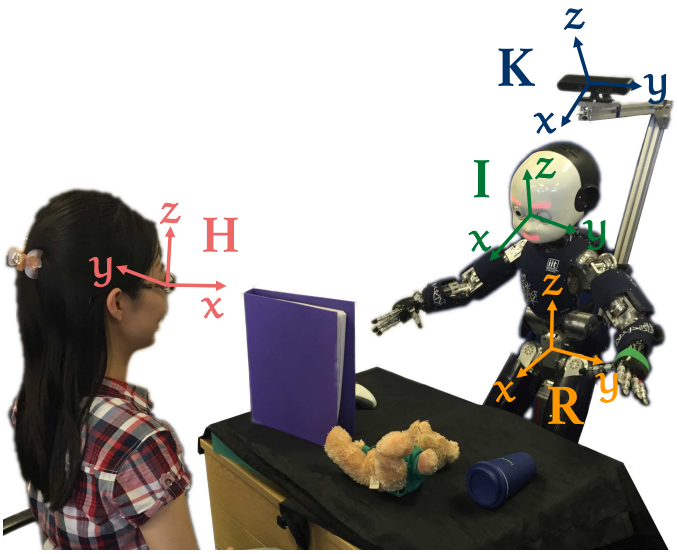


Figure 3.2: Typical setup with the iCub humanoid robot, a human interacting with the robot, and various objects placed between them. Some of the objects are occluded to the human. The figure also shows the different coordinate frames used in this chapter.

Section 3.3.1 (visual PT2) shows that the mapped environment can be used to approximate the view of humans interacting with the robot. However, the environment mapping is optional for PT1 (Section 3.2), and spatial PT2 (Section 3.3.2).

The algorithm runs online and takes the most recently captured point cloud $\mathbf{P}^C(t = t_{now})$ as input. A Bayesian filter is used to determine whether the current location has been visited before, which increases robustness on long mapping sessions. Optionally, laser scans and the estimated odometry of the wheels can be provided, which allows the algorithm to recover when two consecutive point clouds do not have enough visual words in common.

The output of the algorithm is the 3D space Π^C in the reference frame of the RGB-D camera. Π^C contains the concatenated point clouds $\mathbf{P}^C(t = t_0), \ldots, \mathbf{P}^C(t = t_{now})$, after taking the robot movements into account and removing duplicate points. Figure 3.3 shows the resulting point cloud after moving the mobile base of the iCub in a typical lab environment.

3.1.2 *Object Recognition*

The object recognition pipeline is based on the recent work by Pasquale et al. (2015) where a deep learning framework is ported to the iCub robot. The framework allows learning and classifying objects online with one shot

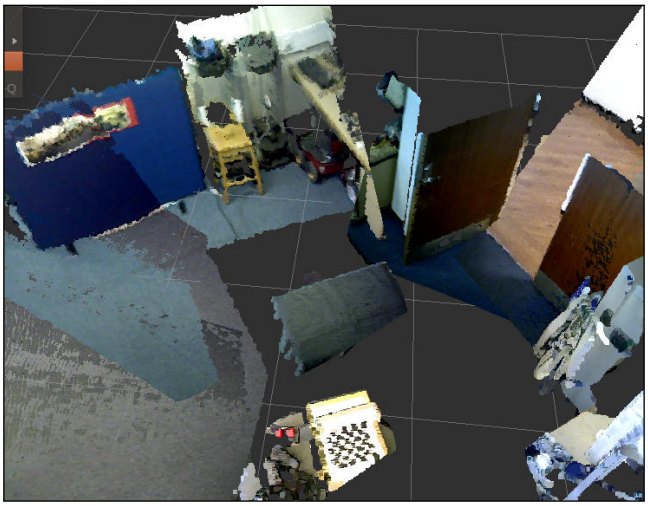


Figure 3.3: Map of the environment after turning the iCub 360 degrees on the spot in a lab environment (view from above). The robot uses this information to reason about parts of the environment that cannot be perceived at the moment but have been seen previously.

learning. The input is the camera image of the left iCub eye camera, and blobs that represent objects are extracted based on the luminosity of the image. For each object, a vector representation of the image is computed using the output of the highest layer of the deep convolutional network. This representation is somewhat invariant to changes in scale, lightness, and orientation. The classification to one of the object classes is done using a support vector machine.

Once the object class is known, the stereo vision system (Fanello et al., 2014) of the iCub is used to estimate the object location in the reference frame $\mathbf{F}_I$ of the left iCub eye. Superquadric models are used to estimate the size and pose of objects (Vezzani et al., 2017). Then, using the robot kinematics the transformation $\mathbf{T}_{I \to R}$ to the robot root reference frame $\mathbf{F}_R$ is computed, and the set $\mathbf{\Omega}^R$ is filled. Finally, the OpenCV object tracker (Kalal et al., 2012) is used to track the objects even if the robot or the human manipulate them.

3.1.3 *Head Pose Estimation*

Accurate and robust estimation of the head poses $\mathbf{H}^C$ of surrounding humans is crucial to take their perspective. If a head pose is not accurately estimated, the robot's judgments regarding the perception of the humans might be incorrect. While Chapter 5 introduces a method for eye gaze estimation, within this chapter, the head pose is used as an approximation of the gaze.

Previous works (Lemaignan et al., 2017; Pandey and Alami, 2010; Pandey et al., 2013) tackled the head pose estimation using motion capture systems. These provide accurate estimates but have the disadvantage that humans need to wear a helmet (or similar) equipped with markers, which might partially occlude the field of view and require a precise calibration. In contrast, the presented approach is based on camera images, which does not require humans to wear additional equipment but comes at the cost of information with increased noise.

3.1.3.1 *Camera-Based Head Pose Estimation*

Recent advancements in the area of computer vision allow the estimation of head poses using a single depth image acquired by an RGB-D camera in real-time (Fanelli et al., 2013). On the dataset of the authors, the position error is around 12 ± 23mm, and the error of the angles around $5.9 \pm 8.1°$. The algorithm is based on discriminative random regression forests. The forest contains many trees, and the trees are learned in a way that within each node of the tree the variance of the head pose is reduced. The output of the forest is the mean of the predictions of the individual trees. The trees also classify whether a patch given as input belongs to a head, which increases robustness when noisy depth data acquired by the RGB-D camera is used as input. If that is the case, the output of the tree is considered for the mean calculation.

A user-specific 3D morphable model is needed to train the forest. The ground truth data is generated using an iterative closest point algorithm. Fanelli et al. (2013) capture a dataset of 20 people for the training of the random forest. The subjects were sitting at a distance of 1 meter straight in front of the camera. Note that no user-specific model is required for the testing phase.

3.1.3.2 *Method for Normalizing Depth Data*

The algorithm works well on the testing set of the dataset. However, the inputs to the algorithm in the human-robot interactions investigated in this chapter differ largely from that of the dataset of Fanelli et al. (2013). The camera is not straight in front of the human, but comes with a large translational offset in the z-direction and rotation by an angle θ around the z-axis. Furthermore, the distance between the camera and the human(s) is larger than one meter. Section 3.4.3 shows that these factors severely decrease the accuracy of the algorithm.

In the proposed approach, rather than building a new training set matching a specific environment, the depth image is normalized so that the face position matches with the ones in the training images. This leads to a performance similar to that of the testing set even if the human is located far away from the training position. Note that this normalization is independent of the presented PT system, and performance improvements are expected for any application where the setup largely differs from that used by Fanelli et al. (2013).

This improvement is achieved by transforming the point cloud $\mathbf{P}^{C}$ into a new reference frame $\mathbf{F}_{A_j}$, resulting in the transformed point cloud $\mathbf{P}^{A_j} = \mathbf{T}_{C\to A_j}\mathbf{P}^{C}$. The origin of frame $\mathbf{F}_{A_j}$ is chosen to be one meter away from the head position $\mathbf{h}_j^R \in \mathbb{R}^3$, and the rotation of the coordinate axes of $\mathbf{F}_{A_j}$ coincide with that of the robot frame $\mathbf{F}_R$. Therefore, $\mathbf{F}_{A_j}$ represents a virtual camera frame that ensures that the subject's pose in this frame is similar to the poses contained in the training dataset of Fanelli et al. (2013).

To achieve this goal, the head position $\mathbf{h}_j^C \in \mathbb{R}^3$ obtained by the RGB-D skeleton tracking[2] (note that the skeleton information only includes the head position, but not orientation), is first transformed into the robot frame $\mathbf{F}_R$: $\mathbf{h}_j^R = \mathbf{T}_{C\to R}\mathbf{h}_j^C$. Then, the transformation matrix $\mathbf{T}_{C\to A_j}$ is derived as follows[3]:

$$\mathbf{T}_{C\to A_j} = \mathbf{T}_{C\to R}\mathbf{T}_{R\to A_j} = \mathbf{T}_{C\to R}\begin{pmatrix} 1 & 0 & 0 & \mathbf{h}_{j,x}^R - 1.0 \\ 0 & 1 & 0 & \mathbf{h}_{j,y}^R \\ 0 & 0 & 1 & \mathbf{h}_{j,z}^R \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.1}$$

Given $\mathbf{T}_{C\to A_j}$, the transformed point cloud $\mathbf{P}^{A_j}$ can be obtained. This is used as input to the pre-trained random forest by Fanelli et al. (2013), which outputs the head pose $\mathfrak{h}_j^{A_j} \in \mathbb{R}^6$ (with the first three elements being a refined position estimate and the last three elements being the head rotation in yaw, pitch, and roll notation) in the aligned reference frame $\mathbf{F}_{A_j}$. Finally, the refined head position $\mathfrak{h}_j^{A_j}$ is transformed back into the initial frame of reference $\mathbf{F}_C$ as follows[4]:

$$\mathfrak{h}_j^C = {\mathbf{T}_{C\to A_j}}^{-1}\mathfrak{h}_j^{A_j} \qquad \forall j \in \{1,\dots,M\}. \tag{3.2}$$

The performance improvement using the described extensions to the original head pose estimation algorithm is crucial for a markerless PT pipeline. Figure 3.4 illustrates the normalization step, and Section 3.4.3 describes the experimental results.

2 PrimeSense NiTE (http://www.openni.ru/) is used for skeleton tracking
3 $\mathbf{T}_{C\to R}$ is derived in Equation (3.3)
4 The notation is abused as only the translational components are transformed

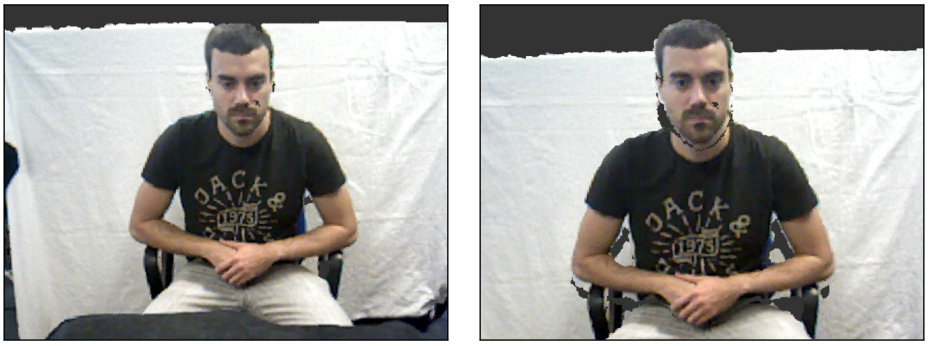


Figure 3.4: Impact of the depth image normalization. The non-normalized input on the left differs largely from the training data, as the camera is mounted on an angle and too far from the subject. The normalized image after transformation on the right is more similar to the training data (subject is facing straight and is closer), which improves the performance of the head pose estimator as shown in Section 3.4.3.

3.1.4 *Transform RGB-D Camera Frame to Robot Frame*

The transformation $\mathbf{T}_{C\to R}$ from the RGB-D camera coordinate frame $\mathbf{F}_C$ to the robot coordinate frame $\mathbf{F}_R$ is determined as follows. First, $\mathbf{T}_{C\to R}$ is initialized using the roughly known geometrical information between the RGB-D camera and the robot root. There is only one rotational component with angle θ around the y-axis[5], and three unknown translational components $x_{C\to R}$, $y_{C\to R}$, $z_{C\to R}$:

$$\mathbf{T}_{C\to R} = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) & x_{C\to R} \\ 0 & 1 & 0 & y_{C\to R} \\ \sin(\theta) & 0 & \cos(\theta) & z_{C\to R} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.3}$$

The final angle θ^* of $\mathbf{T}_{C\to R}$ is found such that the x-axis of $\mathbf{F}_R$ aligns with the floor points in $\mathbf{P}^C$ (except for a translational offset in the z-direction). Then, using ${\mathbf{T}_{C\to R}}^{-1}$, the objects $\mathbf{\Omega}^R$ in the reference frame $\mathbf{F}_R$ are transformed so that they can be visualized along the point cloud $\mathbf{P}^C$ in the RGB-D camera reference frame $\mathbf{F}_C$:

$$\mathbf{\Omega}^C = {\mathbf{T}_{C\to R}}^{-1}\mathbf{\Omega}^R. \tag{3.4}$$

The translational components are then changed step-wise such that the transformed object markers $\mathbf{\Omega}^C$ visually match with the corresponding objects in

5 The camera has been mounted such that there is no rotation around the x and z-axes, in other words there are no yaw and roll.

the point cloud, at which point the final values $x^*_{C\to R}$, $y^*_{C\to R}$ and $z^*_{C\to R}$ are found.

Instead of this manual procedure, an iterative closest point algorithm could be used to find this transformation automatically, which would also make a dynamic camera position feasible. As $\mathbf{T}_{C\to R}$ remains static in the proposed system, it was not necessary to automate the procedure.

3.2 LEVEL 1 PERSPECTIVE TAKING: WHICH OBJECTS CAN YOU SEE?

The previous section introduced the markerless perception of the environment, including the estimation of the head pose of humans and the locations of objects. This section introduces an algorithm for PT1. As a brief reminder, PT1 is the ability to know which objects are seen by others. An example is a situation where the robot is asked to grasp "the toy", but the robot recognizes two toys, leading to an ambiguous situation. However, if the robot knows that the human can see only one of the toys, the robot can infer which toy is meant.

In previous works, the robot mentally transformed its perspective to that of the other robot (Johnson and Demiris, 2005*a*, 2007) or human (Lemaignan et al., 2017; Pandey et al., 2013) to judge the visibility of objects from their perspective. This approach is not feasible in the proposed system, as it needs a very accurate representation of the objects from a non-egocentric viewpoint to recognize the objects in the mentally transformed image. Previously, this was achieved using optical markers; however, this chapter presents a markerless approach.

The line-of-sight tracing as presented in this chapter is a simpler and faster way of judging the visibility. As discussed in Section 2.4, this approach is inspired by cognitive research, which suggests that a) there are two different processes for PT (Flavell et al., 1981), and b) the process for PT1 is based on line-of-sight tracing (Michelon and Zacks, 2006; Yaniv and Shatz, 1990). Section 3.3 describes the approach used for PT2. There is one notable exception where line-of-sight tracing similar to ours is used (Pandey and Alami, 2010). However, in this work (Pandey and Alami, 2010) a motion capture system is employed to detect the objects and humans, whereas the proposed approach allows for more flexibility and does not require the environment to be known a priori.

The PT1 algorithm is based on a line-of-sight tracing approach presented by Amanatides and Woo (1987), which remains the foundation for most works on line-of-sight tracing to date (Laine and Karras, 2011). First, the

points $p_k^C \in \mathbf{P}^C$ captured by the depth camera are mapped into a grid of voxels. The voxel grid $\mathbf{V}^C$ approximates the 3D space spanned by the points p_k^C to volume items v_k^C of equal shape (here: cubes of dimension ξ^3), such that:

$$\mathbf{V}^C = \cup_{k=1}^{K} v_k^C \qquad v_a^C \cap v_b^C = \emptyset \qquad \forall a, b,\ a \neq b. \tag{3.5}$$

The coordinates of a point $p_k^C \in \mathbb{R}^3$ in 3D space are mapped to its equivalent voxel $v_k^C \in \mathbb{Z}^3$ as follows:

$$v_k^C = \lfloor p_k^C / \xi \rfloor. \tag{3.6}$$

Equation (3.6) is used to compute the coordinates of the head poses $\mathbf{H}^{V,C}$ and object locations $\mathbf{\Omega}^{V,C}$ in the voxel grid.

The line-of-sight is traced between each human $j = 1, \ldots, M$ and each object $i = 1, \ldots, N$, leading to $N \times M$ traces. The tracing is performed as follows. A trace $\boldsymbol{\tau}_{j \to i}$ starts at the nose tip $\mathbf{h}_j^{V,C}$ of the j^{th} human, and its target is the object at location $\boldsymbol{\omega}_i^{V,C}$. At each step, depending on the offset of the current voxel and the target voxel, a decision is made whether the next voxel to be traversed is one step towards the x, y, or z-direction. If the traversed voxel contains a point, the algorithm returns with the result that the object is hidden. If the traversed voxel is closer to the target voxel than a given threshold δ, the algorithm returns with the result that the object is visible to the human. Algorithm 1 contains the pseudo-code for the line-of-sight algorithm.

The result of the tracing, being either "visible" or "occluded", is stored in the elements $S_i^{H_j}$ of matrix $\mathbf{O}$ (see Section 3.1). Figure 3.5 visualizes an example trace. Interestingly, as discussed in Section 3.4.2, the execution time of the proposed approach follows qualitatively the reaction times found in humans (Michelon and Zacks, 2006).

3.3 LEVEL 2 PERSPECTIVE TAKING: WHAT DOES THE WORLD LOOK LIKE TO YOU?

This section describes how PT2 tasks are solved in the proposed system. Two tasks are differentiated: 1) estimating how the world is visually perceived by a human (visual PT2), and 2) judging whether objects are to the left, right or in front of the human (spatial PT2). The visual PT2 task is solved by transforming the concatenated point cloud $\mathbf{\Pi}^C$ (see Section 3.1.1) from the camera reference frame $\mathbf{F}_C$ in the reference frame of the j^{th} human $\mathbf{F}_{H_j}$, whose visualization leads to a reconstructed view from the human's perspective. Once the point cloud is in the reference frame of the human, a simple case differentiation is used to solve the spatial PT2 task.

Algorithm 1: Level 1 Perspective Taking

Input : Voxel grid $\mathbf{V}^C$ with leaf size ξ
Origin point $\mathbf{h}_j^{V,C}$ and target point $\boldsymbol{\omega}_i^{V,C}$
Output: Visibility of $\boldsymbol{\omega}_i^{V,C}$ from $\mathbf{h}_j^{V,C}$

direction $\leftarrow (\boldsymbol{\omega}_i^{V,C} - \mathbf{h}_j^{V,C}) / \|\boldsymbol{\omega}_i^{V,C} - \mathbf{h}_j^{V,C}\|$ `// Algorithm initialization`
$\boldsymbol{\Omega}_V^C$ $\leftarrow$ `3DtoVoxel(`$\mathbf{h}_j^{V,C}$`)`
voxel_{max} $\leftarrow$ `VoxelTo3D(`$\boldsymbol{\Omega}_V^C$`)`
for $l \in \{x, y, z\}$ **do** `// Further initialization`
  if direction[l] < 0.0 **then**
    step[l] $\leftarrow -1$; $\text{voxel}_{max}[l] \leftarrow \text{voxel}_{max}[l] - \xi/2$
  else
    step[l] $\leftarrow +1$; $\text{voxel}_{max}[l] \leftarrow \text{voxel}_{max}[l] + \xi/2$
  `// Amount of movement equaling width/height/depth of a voxel`
  $t_\delta[l] \leftarrow \xi / |\text{direction}[l]|$
  `// Distance to intersection with voxel border`
  $t_{max}[l] \leftarrow (\text{voxel}_{max}[l] - \mathbf{h}_j^{V,C}) / \text{direction}[l]$
while $\boldsymbol{\Omega}_V^C \in \mathbf{V}^C$ **do** `// Traversal while voxel remains inside voxel grid`
  if $\|\boldsymbol{\Omega}_V^C - \boldsymbol{\omega}_i^{V,C}\| < \delta$ **then** `// Current voxel is close to target voxel`
    return *visible*
  if `isOccluded(`$\boldsymbol{\Omega}_V^C$`)` **then**
    return *occluded*
  `// Decide whether to go along ray in x, y or z direction (based`
  `// on which component (x, y or z) is closest to voxel border)`
  if $t_{max}[x] \leqslant t_{max}[y]$ **and** $t_{max}[x] \leqslant t_{max}[z]$ **then**
    $t_{max}[x] \leftarrow t_{max}[x] + t_\delta[x]$
    $\boldsymbol{\Omega}_V[x] \leftarrow \boldsymbol{\Omega}_V[x] + \text{step}[x]$
  else if $t_{max}[y] \leqslant t_{max}[x]$ **and** $t_{max}[y] \leqslant t_{max}[z]$ **then**
    $t_{max}[y] \leftarrow t_{max}[y] + t_\delta[y]$
    $\boldsymbol{\Omega}_V^C[y] \leftarrow \boldsymbol{\Omega}_V^C[y] + \text{step}[y]$
  else
    $t_{max}[z] \leftarrow t_{max}[z] + t_\delta[z]$
    $\boldsymbol{\Omega}_V^C[z] \leftarrow \boldsymbol{\Omega}_V^C[z] + \text{step}[z]$

3.3.1 *Level 2 Visual Perspective Taking*

The view of the j^{th} human is estimated by converting $\mathfrak{h}_j^C \in \mathbb{R}^6$ from Euler angles in the RGB-D coordinate frame $\mathbf{F}_C$ (see Section 3.1.3) to a transformation matrix, and afterwards using this transformation matrix to convert $\boldsymbol{\Pi}^C$ in the new reference frame $\mathbf{F}_{H_j}$. First, $\mathfrak{h}_j^C$ is decomposed into its components:

$$\mathfrak{h}_j^C = (\mathfrak{h}_{j,x}^C, \mathfrak{h}_{j,y}^C, \mathfrak{h}_{j,z}^C, \mathfrak{h}_{j,\alpha}^C, \mathfrak{h}_{j,\beta}^C, \mathfrak{h}_{j,\gamma}^C). \tag{3.7}$$

The angles $\mathfrak{h}_{j,\alpha}^C$, $\mathfrak{h}_{j,\beta}^C$, and $\mathfrak{h}_{j,\gamma}^C$ are yaw, pitch and roll angles, i. e. the first rotation is described by $\mathfrak{h}_{j,\gamma}^C$ about the x-axis $\left(\mathbf{R}_{j,x}(\gamma)\right)$, the second rotation by

$\mathfrak{h}^{C}_{j,\beta}$ about the y-axis $(\mathbf{R}_{j,y}(\beta))$, and the third rotation by $\mathfrak{h}^{C}_{j,\alpha}$ about the z-axis $(\mathbf{R}_{j,z}(\alpha))$ (LaValle, 2006, p. 99). The 3x3 rotation matrix $\mathbf{R}_{C\to H_j}$ is obtained as follows:

$$\mathbf{R}_{C\to H_j} = \mathbf{R}_{j,z}(\alpha)\mathbf{R}_{j,y}(\beta)\mathbf{R}_{j,x}(\gamma). \tag{3.8}$$

Finally, the homogeneous transformation $\mathbf{T}_{C\to H^C_j}$ with rotational component $\mathbf{R}_{C\to H^C_j}$ and translational component $\mathbf{t}^C_j = (\mathfrak{h}^C_{j,x}, \mathfrak{h}^C_{j,y}, \mathfrak{h}^C_{j,z})^\top$ is calculated as:

$$\mathbf{T}_{C\to H^C_j} = \begin{pmatrix} \mathbf{R}_{C\to H^C_j} & \mathbf{t}^C_j \\ \mathbf{0} & 1 \end{pmatrix}. \tag{3.9}$$

The transformation matrix is then used to obtain a point cloud whose origin coincides with that of the nose tip of the j^{th} human:

$$\mathbf{\Pi}^{H_j} = \mathbf{T}_{C\to H^C_j}\mathbf{\Pi}^C \tag{3.10}$$

The resulting point cloud $\mathbf{\Pi}^{H_j}$ can then be visualized and contains an approximated view of what subject j is seeing. Figure 3.9 shows an example of a reconstructed view and Section 3.4.4 contains more evaluations. In comparison to earlier works (Johnson and Demiris, 2005*a*, 2007; Lemaignan et al., 2017; Pandey et al., 2013), the transformed view is solely based on the images acquired from the RGB-D camera during the environment mapping (see Section 3.1.1), rather than an a priori known virtual environment. The next section shows that the transformed cloud can also be used for spatial reasoning.

3.3.2 *Level 2 Spatial Perspective Taking*

Spatial PT2 is the ability to judge the spatial location of an object from another frame of reference. Here, the transformed point cloud $\mathbf{\Pi}^{H_j}$ is used for these judgments. Importantly, the judgments are universal, i. e. they do not depend on the frame of reference. This means that the iCub can transfer the knowledge acquired from an egocentric viewpoint ("the toy is to my left") to that of another viewpoint ("the toy is to his right") without changes in the underlying algorithm. This is not limited to spatial PT2 but might also be used for other tasks such as learning by imitation (Demiris and Johnson, 2003), as discussed further in Chapter 6 of this thesis.

This section first presents a simple spatial reasoning approach that allows the iCub to judge the object locations from the iCub's own viewpoint. Then,

the same algorithm is applied to the transformed view, allowing the robot to reason about the human's visual perception. Remember that $\boldsymbol{\omega}_i^R \in \mathbb{R}^3$ denotes the object location in the robot's reference frame $\mathbf{F}_R$, and $L_i^R \in \{\text{left}, \text{central}, \text{right}\}$ describes the spatial relationship between object i and the robot. The left/right judgments are made as follows, with $\theta = 7.5°$ and $\alpha_i^R = \arctan(\boldsymbol{\omega}_{i,x}^R / \boldsymbol{\omega}_{i,y}^R)$:

$$L_i^R = \begin{cases} \text{left} & \text{if } \alpha_i^R > +\theta \\ \text{right} & \text{if } \alpha_i^R < -\theta \\ \text{central} & \text{otherwise.} \end{cases} \tag{3.11}$$

Spatial PT2 is performed in the same way, with the only difference being the angle which is given as the input. Using Equations (3.3) and (3.9), the object location $\boldsymbol{\omega}_i^R$ is transformed in the reference frame $\mathbf{F}_{H_j}$ of human j:

$$\boldsymbol{\omega}_i^{H_j} = \mathbf{T}_{C \to H_j} \mathbf{T}_{C \to R}^{-1} \boldsymbol{\omega}_i^R. \tag{3.12}$$

Then, the angles $\alpha_i^{H_j} = \arctan(\boldsymbol{\omega}_{i,x}^{H_j} / \boldsymbol{\omega}_{i,y}^{H_j})$ are provided to the algorithm in Equation (3.11), and the return values are used to fill $L_i^{H_j}$ of the perception matrix $\mathbf{O}$. Albeit simple, this is a powerful approach to allow the iCub to take the spatial perspective of humans. The knowledge is then used by the iCub to refer to objects as seen by the human, e. g. "I want to play with the toy on your left".

3.4 EXPERIMENTAL EVALUATION

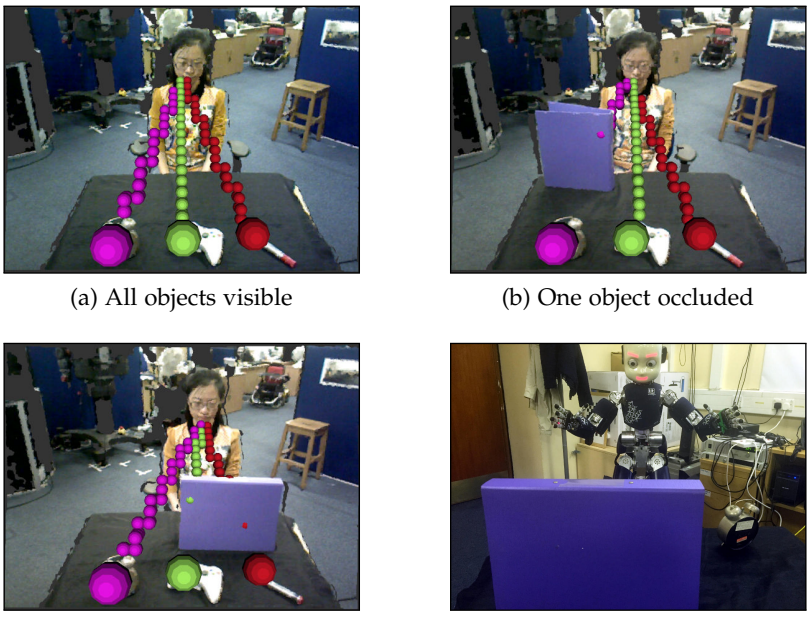
This section evaluates the methods presented in this chapter. An iCub humanoid robot mounted on an iKart mobile base is utilized, with an RGB-D camera attached $z_{C \to R} = 57.5$ cm centrally above the robot root frame $\mathbf{F}_R$, at an angle of $\theta = 26°$ downward facing (see Figure 3.2). The ASUS Xtion Pro is used as RGB-D camera, which provides RGB and depth images with a resolution of 1280×1024 and 320×240 pixels respectively (for more information, please refer to Appendix A.5). This allows the iCub to observe objects placed in front of it on a table, as well as to observe one or two humans sitting at the other end of the table. There is an accompanying video for a demonstration of the experimental results[6].

6 https://www.youtube.com/watch?v=x6EuFzWreq8

3.4.1 *Level 1 Perspective Taking Performance*

Section 3.2 proposed a line-of-sight tracing algorithm to judge which objects are seen by the humans. It was argued that this is a suitable algorithm in a markerless scenario, where object recognition in the transformed view of the human is a yet to be solved problem.

The following demonstration shows the effectiveness of this proposal in three different scenarios. In all scenarios, $N = 3$ different objects (a clock, a joystick and a pen) are placed on a table that is situated between the iCub and the subject (six subjects were evaluated). In Figure 3.5, small spheres are used to visualize the traced line-of-sight, and large spheres are used to denote the object locations. In the first case, the subject can see all objects. In the second case, one object is hidden from the subject, and the tracing stops at the barrier. Similarly, the third case shows two occluded objects. For a better comparison, Figure 3.5(d) shows the actual view of the subject. The line-of-sight tracing algorithm has successfully determined which objects can be seen, demonstrating the robustness of the proposed approach.



(a) All objects visible

(b) One object occluded

(c) Two objects occluded

(d) Subject's view for (c)

Figure 3.5: Level one perspective taking in different scenarios. The visible objects are correctly inferred in all scenarios.

3.4.2 *Level 1 Perspective Taking Comparison to Human Data*

While there is a broad agreement that the human visual system employs line-of-sight tracing to solve level 1 PT tasks (Kessler and Rutherford, 2010; Michelon and Zacks, 2006; Wang et al., 2016; Yaniv and Shatz, 1990), to my knowledge there is only a single experiment where the subject-object distance is varied and the impact on the response time is measured in a PT1 task (Michelon and Zacks, 2006). In this experiment, Michelon and Zacks (2006) have found a linear trend between distance and response time.

Figure 3.6 shows the number of voxel traversals (which serves as proxy for the response time) using the line-of-sight tracing algorithm presented in Section 3.2. The algorithm's response time is near linear ($R^2 = 0.926$), which would suggest that humans might rely on a similar strategy to solve PT1 tasks. A direct comparison with human data cannot be drawn as Michelon and Zacks (2006) contains only two data points, one for near objects (0.31m distance) and another for far objects (0.67m distance).

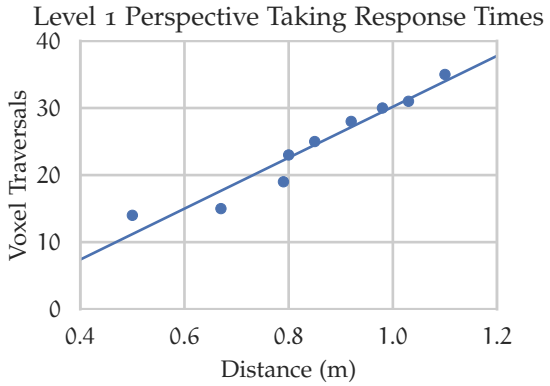


Figure 3.6: Response time profile for line-of-sight tracing. The number of voxel traversals using the proposed line-of-sight algorithm are shown in blue, along with the linear regression line ($R^2 = 0.926$). The distance between the robot and subject is varied between 0.5m and 1.1m.

3.4.3 *Head Pose Estimation with Applied Normalization*

This section evaluates the proposed head pose estimation algorithm that normalizes the input data. In Section 3.1.3, it was hypothesized that the normalization makes the algorithm more invariant to alterations in the viewpoint so that a pre-trained state-of-the-art head pose estimator can be applied in a scenario that differs largely from the training input. To validate this hypothesis, the original algorithm is compared with the extended algorithm on six

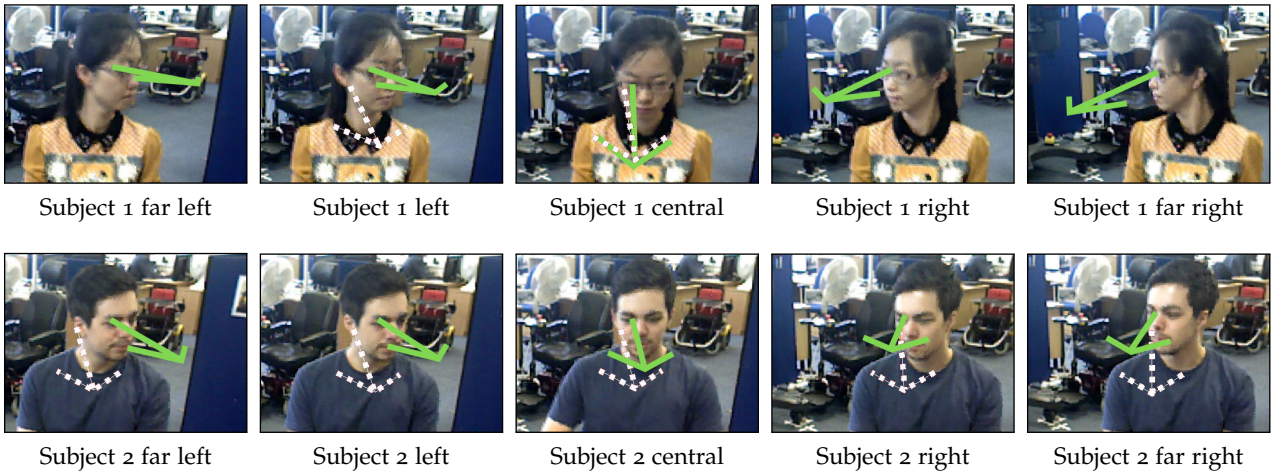


Figure 3.7: Qualitative comparison of the normalized head pose algorithm (green arrows) and the original method (white, dotted arrows). The normalized method leads to a much higher accuracy of the head pose estimation for all head rotations. For three rotations of subject 1, the normalized method resulted in accurate pose estimates whereas the original method failed to detect the head.

subjects each with five different poses, capturing a wide range of horizontal angles. All parameters were kept constant, and both methods were applied to the same input image.

A qualitative comparison is shown in Figure 3.7. Without normalizing the input, the resulting head poses were center-biased for all subjects. For the first and third subject, the original method was not able to estimate a head pose for large angles, whereas the improved method returned accurate estimates. For the sixth subject, the way the subject's hair had fallen across his face resulted in incorrect estimates for both the original and proposed methods.

A quantitative comparison was performed as follows. Six subjects focused on five points with varying distance and horizontal angle from the subject (far left / far right focal points: 1.19m from the subject at $\pm 33°$ angle; left / right focal points: 1.07m at $\pm 20°$ angle; center focal point: 1.0m distance). All focal points were level with the heads of the subjects. Figure 3.8 shows that spatial PT using the normalized algorithm is significantly more accurate (paired t-test, $p < 0.01$), allowing the robot to determine which object is being looked at with reasonably high accuracy even at large angles and distances.

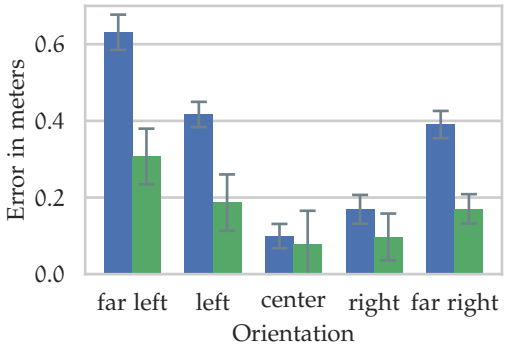


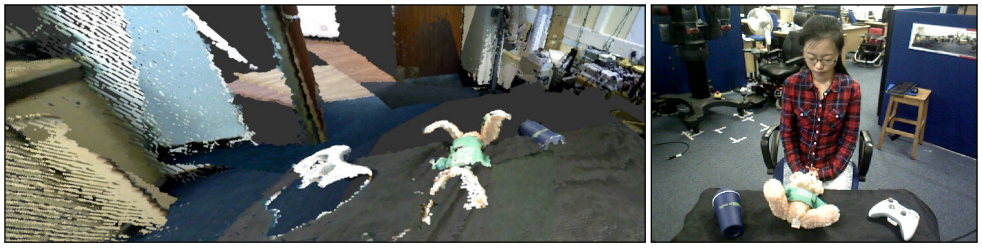
Figure 3.8: Horizontal error for spatial perspective taking, using the original (blue) and normalized (green) head pose estimation algorithms.

3.4.4 *Level 2 Perspective Taking Performance*

Section 3.3 introduced an algorithm to mentally rotate the point cloud acquired by the environment mapping in the frame of reference of the human to estimate what the world looks like to the human. Furthermore, it was shown that rotating the point cloud allows using the same spatial reasoning algorithm as from the robot's perspective. This section validates this proposal in a similar setup to the experiments on PT1. Both demonstrations were carried out with six subjects.

The first demonstration is concerned with visual PT2. Three objects are placed on a table between the iCub and the subject: a joystick to the left of the subject, a toy in the center, and a cup to the right of the subject. First, as shown in Figure 3.9, a mental transform of the point cloud in the reference frame of the human is performed. Albeit being a low-resolution approximation due to the RGB-D camera, the gist of the scene is comprehensible. Importantly, the iCub uses the mapped environment to reason about areas of the scene that cannot currently be perceived by the robot. While this works well for the overall scene, one can observe that the system so far does not build a 3D model of the objects, but rather only represents the object by the surface that can currently be perceived by the robot. Section 7.2 discusses this limitation further.

The second demonstration evaluates spatial PT2. The subject is asked to look at a specific object (while keeping the eyes straight, i. e. only moving the head) and the spatial reasoning algorithm is applied. For example, using the setup in Figure 3.5(a), the subject was asked to look at the joystick. The output of the spatial reasoning was as follows: $\mathrm{L}_{\mathrm{pen}}^{\mathrm{H_j}} = \text{left}$, $\mathrm{L}_{\mathrm{joystick}}^{\mathrm{H_j}} = \text{center}$ and $\mathrm{L}_{\mathrm{clock}}^{\mathrm{H_j}} = \text{right}$. Similarly, in the other scenarios (where the subject was



(a) Approximated view of the human (b) View from the robot

Figure 3.9: (a) Approximated view of the human using a mental transformation. The robot, while facing the human, correctly estimates that the human is looking at a table with three objects. Also, as the robot has a map of the environment, it can reason about areas of the environment that are currently perceived by the human, but not by the robot.
(b) The robot's current view.

asked to look at the pen and at the clock respectively), the spatial reasoning determined the object location from the human's view correctly. Detailed quantitative evaluations are contained in Section 5.6, where the subject's eye gaze is also taken into account.

3.5 CONCLUSIONS

This chapter introduced a novel artificial visual system that allows a robot to take the perspective of surrounding humans. The combined improvements in key parts of the visuospatial perspective taking pipeline have led to a system that works in markerless setups. The system was validated in several experiments using an iCub humanoid robot.

To estimate the head pose, a new method was proposed that normalizes the input images, so they become more similar to the images contained in the training dataset. This improves the performance in scenarios where the pre-normalized input data is dissimilar to the training data, which extends the application scenarios of the head pose estimator.

Line-of-sight tracing was employed to solve PT1 tasks, i. e. to determine whether an object is visible to the human, and it was highlighted that previous methods are not suitable for a markerless environment. For PT2, a mental perspective transformation is used to reconstruct the world from another viewpoint, whereby the robot does not have any prior information about the world and is learning the environment online. It was demonstrated that the robot can judge whether objects are to the left, right or in front of a human using the same algorithm as from an egocentric perspective.

Previous works need artificial markers and/or motion capture systems, which constrains their usability. In contrast, the proposed system can be applied to any environment, e. g. care homes where a robot might aid elderly people by describing object locations in their frame of reference ("the remote control is to your left").

One limitation is that the artificial visual system assumes normal vision of the subject. If the subject has loss of vision, the algorithms might, for example, find that an object is visible to the subject because the line-of-sight is free of obstacles, where instead the subject's vision is too low to see the object. Section 7.2 outlines more limitations of the work presented in this chapter.

The artificial visual system would also benefit from more accurate gaze estimates by taking the eye movements of the human into account. This research direction is discussed further in Chapter 5.

4

ICUB-HRI SOFTWARE FRAMEWORK

This chapter addresses the second research question:

> "How can a perspective taking system be integrated into a cognitive architecture for human-robot interactions (HRIs)?"

The methodology presented in the previous chapter allows the iCub robot to perceive its environment without the use of artificial markers, and reason about the environment from the robot's and the human's point of view using visual and spatial perspective taking (PT) abilities. However, generating complex, human-like behavior requires the integration of more components beyond PT, for example using a scalable cognitive architecture. Hence, this chapter presents the iCub-HRI library, which provides convenience wrappers for components related to perception (agent tracking, speech recognition, touch detection), object manipulation (basic and complex motor actions) and social interaction (speech synthesis, joint attention). The library is exposed as a C++ library with bindings for Java (which allow the use of iCub-HRI within MATLAB) and Python. This allows using the PT abilities introduced earlier in this chapter in a powerful, generic framework for HRI. The code is available for download on the designated GitHub repository[1] alongside extensive documentation (including class diagrams) and tutorials.

This chapter's research has been previously published in Fischer et al. (2018).

4.1 DESIGN PRINCIPLES

The following set of guidelines and design principles were adopted when coding the framework.

- *Adaptability and ease of use:* The framework should be easy to adapt by the community. Individual parts of the framework should only depend on other parts if necessary and substituting components should be easy. Furthermore, all libraries and modules should be documented appropriately.

1 https://github.com/robotology/icub-hri

- *Provision of an overall framework:* Related to the previous goal, the aim is to provide an overall framework which can work "out of the box". Hence, the proposed framework contains modules related to perception, action execution, and social interaction.
- *Extensibility:* It should be easy to extend the framework with new modules. Rather than tailoring existing modules to work with the iCub-HRI framework, it should be possible to write wrapper code for the integration.
- *Shared, centralized knowledge representation:* Each module should have access to the same knowledge database, and the contained knowledge should follow a standardized format. Within iCub-HRI, this knowledge database is called the *working memory*, and the contents are *Entities* or derivatives thereof. The working memory is the default means of communicating among modules.
- *Open software:* The code is released open source and made publicly available. All dependencies must be available as open source software too.

4.2 LIBRARY OVERVIEW

Due to the support of distributed computation within the Yet Another Robot Platform (YARP) middleware (see Appendix A.2 for more details), there are typically many modules running simultaneously when conducting research on the iCub. Data are exchanged using YARP's *Bottle* container, which can encapsulate data of arbitrary length and varying type. While this allows a high degree of flexibility, these containers are error-prone due to the requirement to parse the messages dynamically. This makes verification of compatibility and versioning when used across a large number of modules difficult (Natale et al., 2016). Thus the iCub-HRI library introduces a fixed data representations for knowledge (fully compatible with the *Bottle* container), similar to those used in Robot Operating System (ROS) messages (Quigley et al., 2009; Appendix A.1 contains more details on ROS) and the Interface Description Language (IDL) in YARP (Fitzpatrick et al., 2014). Contrary to ROS messages and IDLs, the iCub-HRI library uses the same representations across all components. Section 4.3 details the representations and their exchange that is managed in a working memory.

Subsystems describe the communication protocol with external modules. As described in Section 4.4, each subsystem connects to a host (i. e. external module) and abstracts away the communication internals. Finally, the *icubClient* class is designed with added convenience for end users in mind such that all subsystems and other higher level methods are available from within a single class.

4.3 KNOWLEDGE REPRESENTATION AND EXCHANGE

The basic representation type is an *Entity*, which is specified by an *ID* and an associated *name*. The *ID* is used when storing and retrieving the entity from the working memory. Several entities can be linked together through a *Relation*, for example 'Paul' (subject) 'holds' (verb) 'duck' (object). In the context of this thesis, relations of the following type are used to indicate whether an agent can see an object from their perspective (level one perspective taking): 'Paul' (subject) 'sees' (verb) 'duck' (object). Similarly, for level two perspective taking, the relation encodes whether the object is to the left, right, or central in front of the agent. Lallée and Verschure (2015) provide further details on relations.

Other knowledge representations inherit the basic properties and methods of *Entity* and extend them further. The *Object* class has additional properties representing the pose, size, presence, and saliency of an object (see Section 4.5.1 for details of how these properties are acquired). The *Agent* class represents a human partner, which besides all properties of an *Object* also stores the positions of all body parts and a list of beliefs. Another commonly used representation is that of a *Bodypart*, which represents a part of the robot's body. A *Bodypart* also inherits all attributes of an *Object*, and additionally contains the related joint number, tactile patch identifier, and corresponding body part of the human. Zambelli et al. (2016) have used these representations to anchor self-learned representations to those of a human interacting with the robot.

These representations must be shared across different modules (for example between perceptual modules and the more abstract reactive layer as described later in this section), and the *OPCClient* class automates the exchange of representations with the working memory of the iCub ecosystem (Objects Properties Collector; OPC, see Lallée and Verschure, 2015). The OPC is an ontology-based knowledge representation system which is grounded on the need of humans and other social animals to interact in a physical, multi-agent world (see Lallée and Verschure, 2015).

In this direction, the role of such knowledge representation should be to structure and distribute information to different modules in an asynchronous (on-demand) and centralized way. The design is inspired by the *repository pattern* known from software engineering (Evans, 2004), and its usage is similar to the centralized version control software Apache Subversion (known as SVN)[2]. For storage and retrieval, the *OPCClient* provides methods such as `checkout()` to poll representations from the shared memory, `update()` to update existing representations, and `commit()` to overwrite representations in the memory with the local version of the module. Altogether, this implementation provides a shared, centralized knowledge representation, enabling asynchronous access to the information. This follows the design principle outlined in Section 4.1.

4.4 SUBSYSTEMS

A *Subsystem* provides a wrapper around the representations used by external components and the ones used within iCub-HRI, which compares to the façade software engineering pattern (Gamma et al., 1994). This has several advantages, including that the complexity of remote procedure calls is hidden from the user and that formerly "incompatible" components can now be used within the same project. The following paragraphs provide a brief list of the most commonly used interfaces of these subsystems. The documentation on GitHub[3] contains the complete list.

The advantages outlined above are especially evident in the subsystems for the *Actions Rendering Engine* (ARE; follow up work on Pattacini et al., 2010)[4] and KARMA (Tikhanoff et al., 2015)[5] object manipulation libraries, which are typically used by the iCub community to issue high-level motor commands. If directly called, they require the provision of complex parameters. Contrary, using iCub-HRI, one merely specifies the desired action and the name of the target object, as further demonstrated in Section 4.6.1.

2 https://subversion.apache.org/

3 https://robotology.github.io/icub-hri/ → iCub-HRI libraries → Subsystems

4 The following interfaces are provided by the ARE subsystem: 1) `home()` to put the robot or a specified part in the home position, 2) `take()` to reach and grasp an object, 3) `push()` to laterally push an object, 4) `point()` to an object, 5), `expect()` to extend the hand and wait for an object, 6) `drop()` an object which is currently held, 7) `wave()` the robot's hands, 8) `look()` at an object, and 9) `track()` a moving object.

5 The following interfaces are provided by the KARMA subsystem: 1) `pushKarmaLeft()` and `pushKarmaRight()` to push an object to the left/right side with a specified target position, 2) `pushKarmaFront()` to push an object forwards, and 3) `pullKarmaBack()` to bring an object closer to the robot.

Other important subsystems are those for speech recognition and synthesis. Both are convenience wrappers for the functionality offered in the 'speech' repository of the iCub ecosystem. The speech synthesizer allows for speech production from text using a single command `say()`, with the only parameter being the sentence to be spoken, while being agnostic to the underlying synthesizer (Acapela[6], eSpeak[7], Festival[8], and SVOX Pico[9] are supported). The speech recognizer relies on the Microsoft Speech API[10], which allows recognition and extraction of words from spoken utterance given a grammar file (using the command `recogFromGrammarLoop()`).

The functionality of the different subsystems is aggregated in the *icubClient* class, which allows the use of the different subsystems from within a single class instance. A configuration file is used to specify which subsystems a module requires, such that no unnecessary resources are bound.

4.5 ICUB-HRI MODULES

The modules accompanying the iCub-HRI library can be grouped into four main areas: 1) perception, 2) action, 3) social interaction, and 4) miscellaneous tools. All modules have access to the knowledge introduced in the previous section (as they use the iCub-HRI library) and none of them is dependent on the other; i. e. one can choose which subset of modules to run for each experiment, if any.

4.5.1 *Perception Modules*

AGENT DETECTOR The *agentDetector* module is responsible for detecting and tracking the skeleton of a human partner using an RGB-D camera mounted behind the robot. It converts the joint positions detected by the RGB-D camera into the reference frame of the iCub and continuously updates the joint positions of the human partner in the working memory.

DEFAULT SPEECH RECOGNITION The *Ears* module allows for recognition of speech utterances from the human when no other module is trying to recognize speech. It takes the role of a central component to redirect the command extracted from the recognized sentence to the appropriate module,

6 http://www.acapela-group.com
7 http://espeak.sourceforge.net
8 http://www.cstr.ed.ac.uk/projects/festival/
9 https://github.com/robotology/speech/tree/master/svox-speech
10 https://msdn.microsoft.com/en-us/library/ee125663(v=vs.85).aspx

while still allowing other modules to access the speech recognition subsystem directly if needed.

OBJECT RECOGNITION The object recognition module within iCub-HRI is the same as described within Section 3.1.2 of this chapter. In short, the object regions are identified based on the luminosity of the image and features are subsequently extracted using a deep neural network. The object class is determined using a support vector machine, and the object location is found using the stereo vision system of the iCub.

SALIENCY The module *PASAR* (Mathews et al., 2012) detects the appearance and disappearance of objects, and the saliency of an object is increased proportionally to its acceleration. This also allows simple detection of pointing actions by measuring the proximity of the human's hand to each of the objects and increases the saliency with inverse proportion to the distance.

FACE AND ACTION RECOGNITION The *Synthetic Sensory Memory* module (Martinez-Hernandez et al., 2016) is used to recognize faces and actions performed on objects. It uses Gaussian Process Latent Variable Models (Damianou et al., 2011) to train classifiers for faces and actions, which can then be loaded during an interaction to perform real-time classification.

4.5.2 *Action Modules*

FACE TRACKING The face tracking module detects the face of a human based on Haar cascades implemented in OpenCV (Viola and Jones, 2001), and uses the velocity control of the iCub to follow the face. This module can be used in human-robot interaction scenarios for increased vividness of the robot.

BABBLING The *Babbling* module allows the issue of pseudo-random (sinusoids) commands to the iCub (either individual or several joints). It has been used to learn forward and inverse models for the iCub (Zambelli and Demiris, 2017), as well as to learn correspondence between the robot's body parts and that of the human (Zambelli et al., 2016). Within the scope of iCub-HRI, it is mainly used for body part learning.

4.5.3 *Social Interaction Modules*

PROACTIVE TAGGING The robot can acquire knowledge in two different ways: proactively, where a decaying drive to acquire knowledge triggers the behavior to obtain the name of an object or body part, or reactively, where the knowledge acquisition follows a human command. The demonstration described in this thesis is centered around the proactive tagging module, which makes use of several subsystems and connects directly to several other modules. It uses the speech recognition subsystem to acquire the names of entities (objects in the vicinity, partners, and body parts), the speech synthesis subsystem to enable the robot to express itself verbally (to ask for object names), and the ARE subsystem to point at objects and make them salient. Furthermore, it makes use of the functionalities provided by a number of other modules presented within the previous section, including *PASAR* to detect which object the partner is pointing to, the face recognition module to recognize the partner, and the *touchDetector* to identify which skin patch was being touched by the human. Figure 4.1 shows an overview of the interaction between the modules.

REACTIVE LAYER The reactive layer implements drive reduction mechanisms for self-regulating the robot's behavior. A drive is defined as a control loop that triggers appropriate behaviors whenever an associated internal state variable goes out of its homeostatic range. These drives present a way to self-regulate value dynamically and autonomously (Sanchez-Fibla et al., 2010). This has been shown to positively influence the acceptance of the human-robot interaction by naive users (Lallée and Verschure, 2015; Vouloutsi et al., 2014). Figure 4.2 shows the module interaction where an internal state variable goes out of the homeostatic range.

DRIVE SYSTEM In the social robotics context, two example drives allow the robot to balance knowledge acquisition and expression autonomously. The *drive for knowledge acquisition* maintains a curiosity-driven exploration of the environment by proactively requesting information from a human about the present entities (e. g. their name). The *drive for knowledge expression* regulates how the iCub expresses the acquired knowledge through synchronized speech, pointing actions and gaze. It informs the human about the robot's current state of knowledge and thus maintains the interaction.

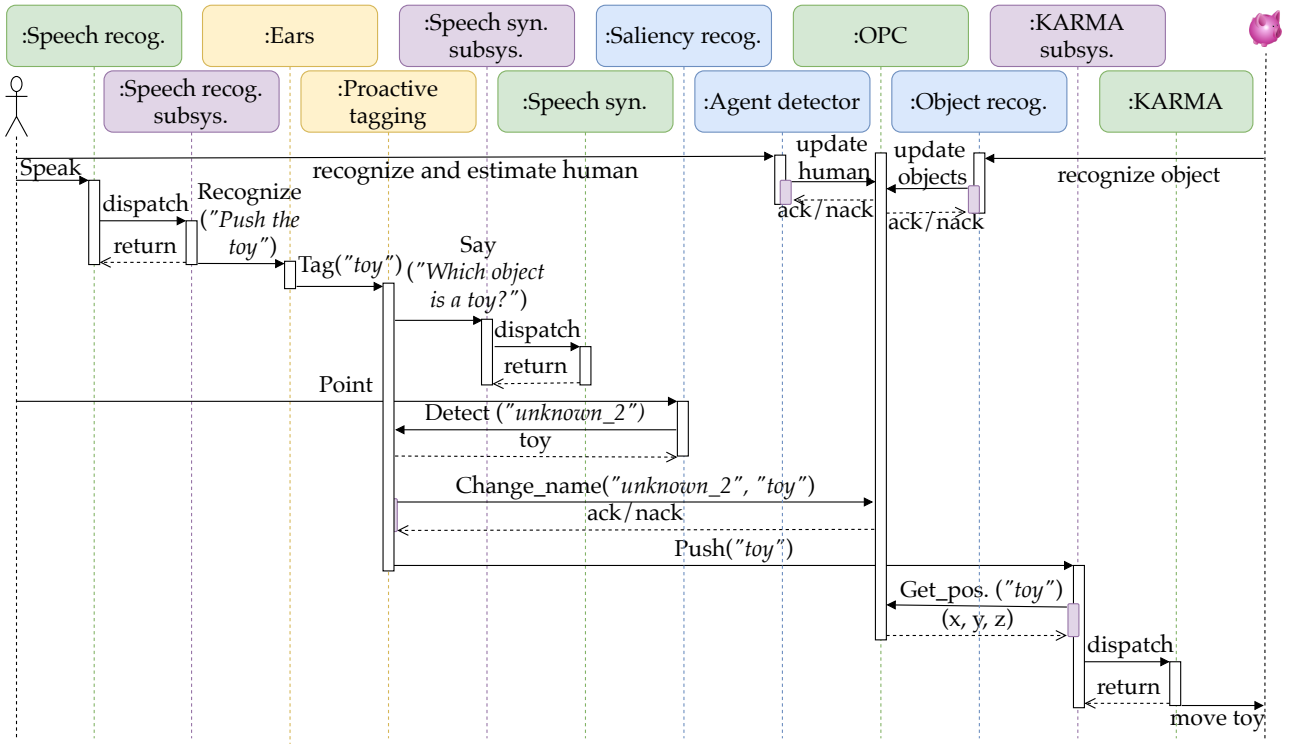


Figure 4.1: Temporal Unified Modeling Language diagram for an interaction where a human gives a speech command to the iCub to push an object which is currently unknown to the robot. The diagram depicts the modules and subsystems involved, and shows the information flow. After converting the speech command to an action plan, the robot first asks the human to indicate the desired object, and subsequently pushes that object. The agent detector and object recognition modules continuously update the knowledge database throughout the interaction, and the object name is updated after the human indicates the object by pointing to it.

4.5.4 *Tools*

Several tools provide pre-processing functionalities for the other modules or interact with other modules of the iCub ecosystem so that they can be easily used within iCub-HRI.

GUIUPDATER The *guiUpdater* translates the representations of iCub-HRI to those used within the *iCubGui*. More specifically, it allows the display of location for objects and agents stored within the working memory along with specific properties, such as their color and name.

OPCPOPULATOR The *opcPopulator* can be used to spawn new entities in the simulation and control their parameters. This allows testing new func-

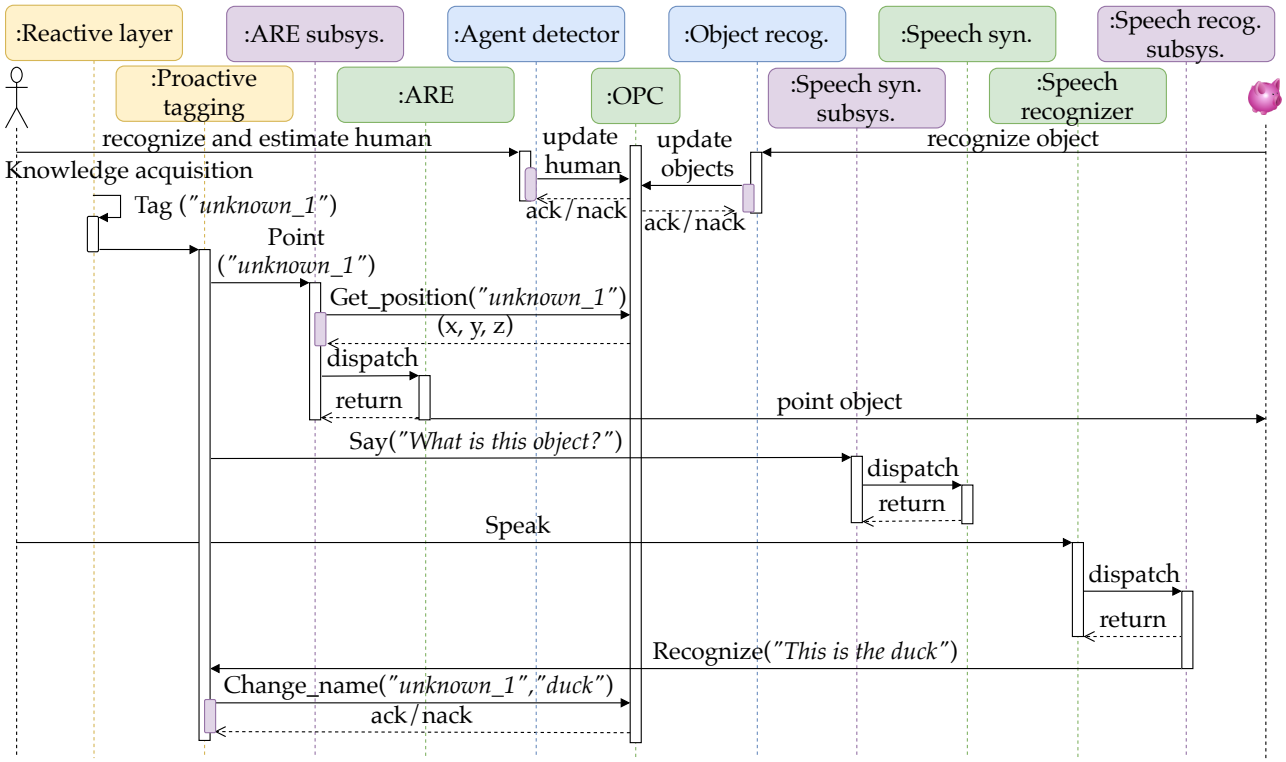


Figure 4.2: Temporal Unified Modeling Language diagram for an interaction where a drive threshold is hit. This triggers the behavior to tag an unknown object autonomously. The robot first points to the unknown object and then asks the human for the object's name. Once the human has responded, the proactive tagging module changes the object name in the OPC.

tionalities in a controlled environment, without the noise encountered when using the real robot.

4.6 USING ICUB-HRI

There is a variety of use cases for iCub-HRI. Section 4.6.1 shows the ease of use of iCub-HRI in a representative example related to the object manipulation subsystem. Subsequently, Section 4.6.2 briefly describes how an extended version of this tutorial has been used to tackle the symbol grounding problem in the DAC-h3 framework (Moulin-Frier, Fischer et al., 2018). This is followed by a description of the benefits of this library for technical and non-technical users alike in Section 4.6.3.

```
#include <cstdlib>
#include <yarp/os/all.h>
#include <icubhri/clients/icubClient.h>

int main() {
    yarp::os::Network yarp;

    icubhri::ICubClient iCub("KARMA_Simple");
    // connect to subsystems
    if(!iCub.connect()) { return -1; }

    // objectName as recognized by object recognition
    std::string objectName = "octopus";
    double targetPositionX = -0.45;

    bool ok = iCub.pushKarmaFront(objectName,
       targetPositionX);
    yInfo() << (ok ? "Success" : "Failed");

    return EXIT_SUCCESS;
}
```

Listing 4.1: Pushing an object using iCub-HRI is straightforward and requires the provision of just two parameters: the object to be pushed and the desired target position.

4.6.1 *Example Usage of the Object Manipulation Subsystems*

The GitHub repository[11] contains a range of examples, including examples of using the KARMA and ARE subsystems to manipulate objects, i. e. grasping, pushing or pulling them, in C++, Python, and MATLAB. Some examples use *yarp::sig::Vector* instances to specify the target location (important for users looking to employ iCub-HRI as a light-weight library), while others rely on the *Object* class introduced earlier (providing seamless integration with the contained object recognition module). Listing 4.1 shows an example which uses the iCub-HRI library to push an object using the *KARMA Subsystem*, while Listing 4.2 contains code directly communicating with KARMA, which is much less intuitive and likely distracts from the desired code related to the human-robot interaction.

11 https://github.com/robotology/icub-hri

```
#include <cstdlib>
#include <yarp/os/all.h>
#include <yarp/sig/all.h>

yarp::sig::Vector getPos(std::string name) {
    // Communicate with object recognition module to
    // obtain object position.
    // This is not shown for brevity.
}

int main() {
    yarp::os::Network yarp;

    yarp::os::RpcClient toKarma;
    toKarma.open("/example/toKarma");
    yarp::os::Network::connect(toKarma.getName(),
        "/karmaMotor/rpc");

    yarp::sig::Vector pos = getPos("octopus");
    double targetPositionX = -0.45;
    double radius = fabs(pos[0] - targetPositionX);

    yarp::os::Bottle cmd, reply;
    cmd.addString("push");
    cmd.addDouble(pos[0]);
    cmd.addDouble(pos[1]);
    cmd.addDouble(pos[2]);
    cmd.addDouble(-90); // angle theta
    cmd.addDouble(radius); // distance to be pushed
    toKarma.write(cmd, reply);
    bool ok = (reply.get(0).asVocab() == yarp::os::Vocab::
       encode("ack"));
    yInfo() << (ok ? "Success" : "Failed");

    return EXIT_SUCCESS;
}
```

Listing 4.2: Pushing an object communicating directly with KARMA. Besides being less readable, this code is also more error-prone as the *Bottle*'s components need to be provided with the right type and in the right order. Furthermore, many more parameters are involved.

4.6.2 *Usage within the DAC-h3 framework*

iCub-HRI has been used as the underlying framework for the DAC-h3 cognitive architecture (Moulin-Frier, Fischer et al., 2018). There, the iCub learns to solve the symbol grounding problem, acquire language capabilities, execute goal-oriented behaviors, and express a verbal narrative of the robot's experience in the world.

The work of Moulin-Frier, Fischer et al. (2018) also demonstrates that the software framework presented in this chapter can be readily used to study human-robot interaction with naive subjects. More specifically, the robot's task is to explore its environment. It asks the humans for their names and interacts with them to learn the names of objects. Once the object names are known, the iCub then interacts with a human to move objects either closer to the robot or to pass them.

4.6.3 *More Applications and Use Cases*

The central advantage of iCub-HRI is that the library bypasses the requirement for obtaining a working knowledge of the operation of an extensive range of modules during the normal operation of the iCub and of the module interaction before starting to develop one's specific application on top of these modules. Furthermore, iCub-HRI's modular subsystem architecture means that one can easily integrate applications developed on top of iCub-HRI to further abstract and accelerate the development of robotics applications.

The underlying design principles of iCub-HRI (see Section 4.1) and the high-level abstractions of the robot's basic input and output systems like speech, vision and motor control allow a widely varied range of use cases. For users with a non-technical background, iCub-HRI has the potential to reduce the learning curve for exploiting the iCub robotic platform, with potential applications such as robotic art, research into the societal effects of robotics, investigations into human-robot collaboration and human-robot interaction studies investigating the psychological effects of such an interaction. For users more familiar with the iCub, the flexibility of the library allows them to focus on the core of their applications, where iCub-HRI provides a bridge to quickly integrate these applications with the sensory, motor and affective systems of the robot. This reduces the implementation effort which leads to faster developments and allows for accelerated prototyping of embodied artificial intelligence applications.

4.7 CONCLUSIONS

This chapter introduced iCub-HRI, a software framework that integrates various components available within the iCub ecosystem and makes them easily accessible through method calls. iCub-HRI can be used in various ways, from a lightweight library up to an integrated platform for complex studies on HRI. While iCub-HRI is tailored for the iCub humanoid robot, many parts are platform independent and can be used on other robotic platforms as well.

It was shown that the PT framework introduced in Chapter 3 can be integrated within iCub-HRI, such that other modules have access to information related to the perception of the environment from the subject's point of view. The next chapter investigates the estimation of a human's gaze direction in HRI scenarios while taking both head pose and eye gaze into account.

5

REAL-TIME EYE GAZE ESTIMATION IN NATURAL ENVIRONMENTS

This chapter addresses the third research question:

> "How can a robot accurately estimate the gaze direction of a human, taking both head pose and eye gaze into account, and can such mechanisms be learned from data?"

The proposed method in this chapter is titled RT-GENE ("Real-Time Gaze Estimation in Natural Environments"). It overcomes one of the main limitations of the artificial visual system introduced in Chapter 3 – namely, taking the human's eye gaze into account. Table 5.1 shows that this goes beyond previous works as the eye gaze can be estimated even in large camera-subject distances, which are commonly encountered in human-robot interaction (HRI) scenarios. This is achieved using a novel architecture for ground truth annotations that was used to collect a new dataset, as well as an improved gaze estimator that introduces ensemble schemes to this area.

This chapter's research has been previously published in Fischer, Chang and Demiris (2018).

Table 5.1: Comparison with related works on gaze estimation

Paper	Head pose	Eye pose	Input	Estimate type	Distance	Angle	Open Source
Deep Head Pose (Mukherjee and Robertson, 2015)	x	-	RGB-D	3D vector	All mixed	All mixed	-
Eyediap (Funes-Mora and Odobez, 2016)	x	x	RGB-D	3D vector	<150cm	Frontal	Partly
MPII Gaze (Zhang et al., 2015)	x	x	RGB	2D vector	Close	Narrow	x
UnityEyes (Wood et al., 2016)	x	x	RGB	3D vector	Close	Frontal	-
Valenti et al. (2012)	x	x	RGB	3D vector	75cm	Frontal	-
Gaze Capture (Krafka et al., 2016)	x	x	RGB	Gaze on tablet	Very close	Narrow	-
YAGD (Schillingmann and Nagai, 2015)	x	x	RGB	3D vector	90-110cm	Frontal	x
Palinko et al. (2015)	x	x	RGB	3D vector	60-100cm	Frontal	-
Gazefollow (Recasens et al., 2015)	x	x	RGB	2D image pos	All mixed	All mixed	x
Chamveha et al. (2013)	-	-	RGB	1D vector	Far	All mixed	-
RT-GENE	**x**	**x**	**RGB**	**3D vector**	**50-290cm**	**Frontal**	**x**

5.1 ARCHITECTURE OVERVIEW

RT-GENE involves automatic annotation of ground truth datasets by combining a motion capture system, used for accurate detection of head pose, and mobile eyetracking glasses, used for eye gaze annotation. Figure 5.1 shows that this setup directly provides the gaze vector in an automated manner under free-viewing conditions (i. e. without specifying an explicit gaze target), which allows rapid recording of the dataset. Table 5.2 provides a comparison of various gaze datasets including RT-GENE.

Table 5.2: Comparison of gaze datasets

Dataset	RGB / RGB-D	Annotation type	#Images	Cl. Dist.	Med. Dist.	Far Dist.	Head pose annot.	Gaze annot.	Pupil annot.	Orient.
Eyediap (Funes Mora et al., 2014)	RGB-D	Gaze	Unknown	x	-	-	x	x	Partly	Frontal
MPII Gaze (Zhang et al., 2015)	RGB	Gaze	213,659	x	-	-	x	x	Partly	Frontal
CMU Multi-Pie (Gross et al., 2008)	RGB	Head pose	750,000	x	-	-	x	Partly	Partly	Frontal
BIWI (Fanelli et al., 2013)	RGB-D	Head pose	ca. 15,500	x	-	-	x	-	-	Frontal
ICT 3D Head Pose (Baltrusaitis et al., 2012)	RGB-D	Head pose	14,000	x	-	-	x	-	-	Frontal
Columbia (Smith et al., 2013)	RGB	Gaze	5,880	x	-	-	5 orient.	x	-	Frontal
Vernissage (Jayagopi et al., 2013)	RGB	Head pose	Unknown	x	-	-	x	-	-	All
Boston Uni Head Pose (La Cascia et al., 2000)	RGB	Head pose	9,000	x	-	-	x	-	-	Frontal
Oxford Surveillance (Benfold and Reid, 2011)	RGB	Head pose	1,747	-	-	x	x	-	-	All
CAVIAR (Fisher, 2004)	RGB	Head pose	15,498	-	-	x	x	-	-	All
Chamveha (Chamveha et al., 2013)	RGB	Body pose	15,000	-	-	x	-	-	-	All
Coffeebreak (Cristani et al., 2011)	RGB	Head pose	18,117	x	-	-	6 orient.	-	-	All
HIIT (Tosato et al., 2013)	RGB	Head pose	24,000	x	-	-	6 orient.	-	-	All
QMUL MultiView (Gong et al., 1998)	RGB	Head pose	6,384	x	-	-	x	-	-	Frontal
ETH Face Pose (Breitenstein et al., 2008)	Depth	Head pose	ca. 13,000	x	-	-	x	-	-	Frontal
SynthEyes (Wood et al., 2015)	RGB	Gaze	11,382	Eyes	-	-	x	x	x	Frontal
UT MultiView (Sugano et al., 2014)	RGB	Gaze	26,400	Eyes	-	-	x	x	-	Frontal
IDIAP Head Pose (Ba and Odobez, 2005)	RGB	Head pose	Unknown	x	-	-	x	-	-	Frontal
Eurecom Kinect (Min et al., 2014)	RGB-D	6 landmarks	Unknown	x	-	-	x	-	-	Frontal
Gaze Capture (Krafka et al., 2016)	RGB	Screen coords.	> 2.5M	x	-	-	-	x	-	Frontal
Rice TabletGaze (Huang et al., 2017)	RGB	Screen coords.	ca. 100,000	x	-	-	-	x	-	Frontal
Gazefollow (Recasens et al., 2015)	RGB	Screen coords.	122,143	x	x	x	-	x	-	All
RT-GENE	**RGB-D**	**Gaze**	122,531	**x**	**x**	**-**	**x**	**x**	**Partly**	**Frontal**

While the RT-GENE architecture provides accurate gaze annotations, it requires the subjects to wear eyetracking glasses that introduce the problem of unnatural subject appearance when recorded from an external camera (note that the eyetracking glasses and motion capture system are only required to capture the dataset, but not at inference time where only an RGB camera is required). Since the aim is to estimate the gaze of subjects without the use of eyetracking glasses, it is vital that the test images are not affected by

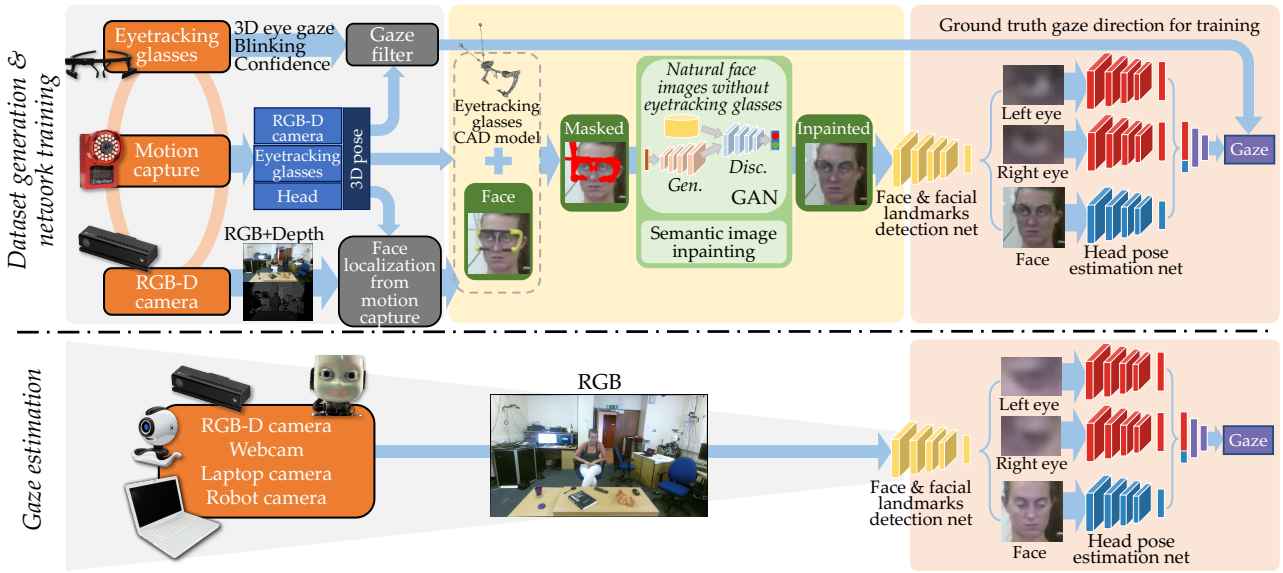


Figure 5.1: RT-GENE architecture overview. During training, a motion capture system is used to find the relative pose between mobile eyetracking glasses and an RGB-D camera (both equipped with motion capture markers), which provides the head pose of the subject. The eyetracking glasses provide labels for the eye gaze vector with respect to the head pose. A face image of the subject is extracted from the camera images, and a semantic image inpainting network is used to remove the eyetracking glasses. A landmark detection deep network extracts the positions of five facial landmarks, which are used to generate eye patch images. Finally, the proposed gaze estimation network is trained on the annotated gaze labels.

an alteration of the subjects' appearances. For this purpose, a realistic image generation method is applied in a new scenario, namely the inpainting of the area covered by the eyetracking glasses. As shown in Figure 5.1, the images with removed eyetracking glasses are then used to train a new gaze estimation framework. The experiments in Section 5.6.1 will validate that the inpainting improves the gaze estimation accuracy.

5.2 GAZE DATASET GENERATION

One of the main challenges in appearance-based gaze estimation is accurately annotating the gaze of subjects with natural appearance while allowing free movements. This chapter presents an approach that allows automatic annotation of ground truth gaze and head pose labels of subjects under free-viewing conditions and with large camera-subject distances (Figure 5.2 shows the overall setup), and presents a new dataset following this approach. The dataset was constructed using mobile eyetracking glasses and a Kinect

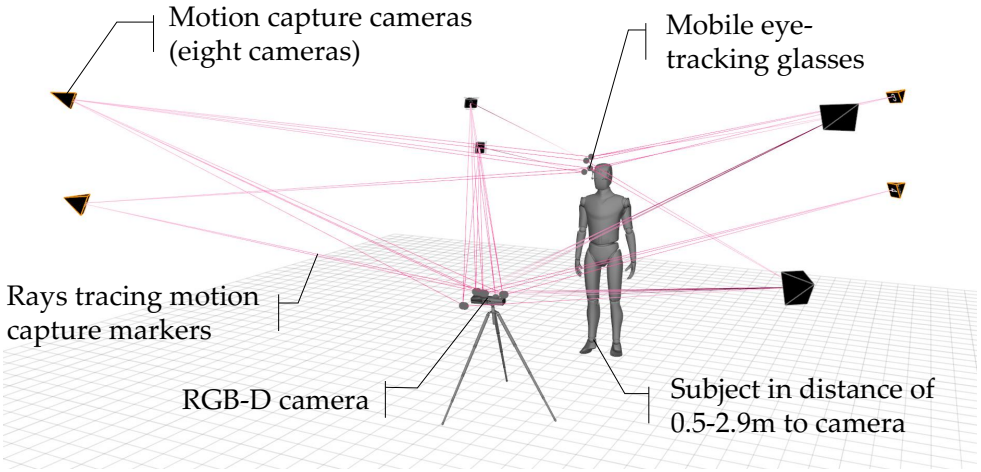


Figure 5.2: Proposed setup for recording the RT-GENE gaze dataset. An RGB-D camera records a set of images of a subject wearing Pupil Labs mobile eyetracking glasses (Kassner et al., 2014). Markers that reflect infrared light are attached to both the camera and the eyetracking glasses, in order to be captured by motion capture cameras. The setup allows accurate head pose and eye gaze annotations in an automated manner.



Figure 5.3: Some images contained in the RT-GENE dataset. The images show that the camera pose and subject position was changed for each subject.

v2 RGB-D camera, both equipped with motion capture markers, in order to precisely find their poses relative to each other. The eye gaze of the subject is annotated using the eyetracking glasses, while the Kinect v2 is used as a recording device to provide RGB and depth images (see Appendix A.5 for more details). Figure 5.3 shows some example images contained in the dataset.

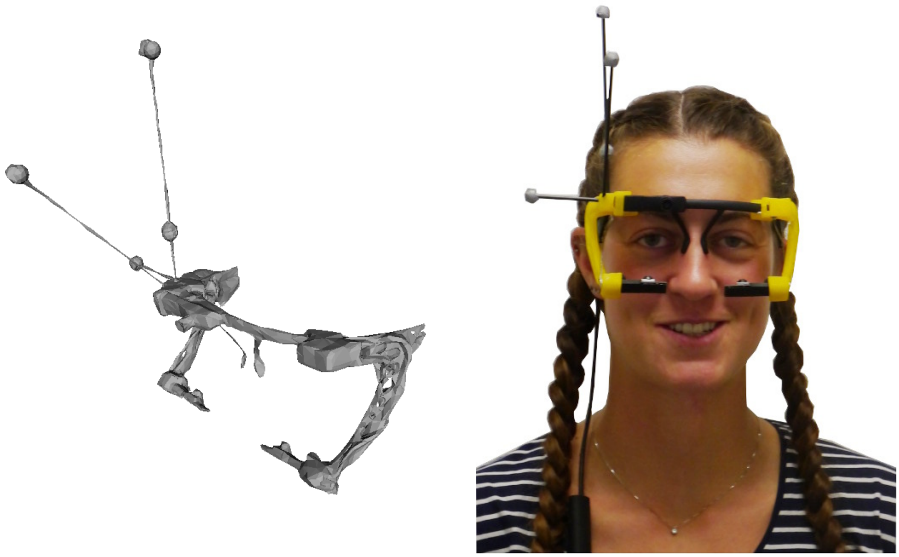


Figure 5.4: Left: 3D model of the eyetracking glasses including the motion capture markers. Right: Eyetracking glasses worn by a subject. The 3D printed yellow parts have been designed to hold the eye cameras of the eyetracking glasses in the same place for each subject.

5.2.1 *Eye Gaze Annotation*

As detailed in Appendix A.4, the gaze is annotated using a customized version of the Pupil Labs eyetracking glasses (Kassner et al., 2014), which have a very low average eye gaze error of 0.6 degrees in screen-based settings. In the proposed dataset with significantly larger distances, the angular accuracy is 2.58 ± 0.56 degrees. The headset consists of a frame with a scene camera facing away from the subject and a 3D printed holder for the eye cameras. This removes the need to adjust the eye camera placement for each subject. The customized glasses provide two crucial advantages over the original headset. Firstly, the eye cameras are mounted further from the subject, which leads to fewer occlusions of the eye area. Secondly, as described in Section 5.4, the fixed position of the holder allows the generation of a generic (as opposed to subject-specific) 3D model of the glasses, which is needed for the inpainting process. Figure 5.4 shows the generic 3D model and glasses worn by one of the subjects.

5.2.2 *Head Pose Annotation*

A commercial OptiTrack motion capture system tracks the RGB-D camera and eyetracking glasses using four markers attached to each object, with an

average position error of 1mm for each marker. This allows us to infer the pose of the eyetracking glasses with respect to the RGB-D camera, which can be used to annotate the head pose as described below. Appendix A.6 contains more information about the motion capture system.

5.2.3 *Coordinate Transforms*

The fundamental challenge in the dataset collection setup was to relate the eye gaze $\mathbf{g}^{\mathrm{E}}$ in the eyetracking reference frame $\mathbf{F}_{\mathrm{E}}$ to the visual frame of the RGB-D camera $\mathbf{F}_{\mathrm{C}}$ as expressed by the transform $\mathbf{T}_{\mathrm{E}\to\mathrm{C}}$. This transform is also used to define the head pose $\mathbf{h}^{\mathrm{C}}$ as it coincides with $\mathbf{T}_{\mathrm{C}\to\mathrm{E}}$. However, the transform $\mathbf{T}_{\mathrm{E}^*\to\mathrm{C}^*}$ provided by the motion capture system cannot be directly used, as the frames perceived by the motion capture system, $\mathbf{F}_{\mathrm{E}^*}$ and $\mathbf{F}_{\mathrm{C}^*}$, do not match the visual frames, $\mathbf{F}_{\mathrm{E}}$ and $\mathbf{F}_{\mathrm{C}}$.

Therefore, the transforms $\mathbf{T}_{\mathrm{C}\to\mathrm{C}^*}$ and $\mathbf{T}_{\mathrm{E}\to\mathrm{E}^*}$ must be found. The transform $\mathbf{T}_{\mathrm{C}\to\mathrm{C}^*}$ can be found by exploiting the property of RGB-D cameras that 3D point coordinates of an object are known in the visual frame $\mathbf{F}_{\mathrm{C}}$. If this object is equipped with markers tracked by the motion capture system, the matching coordinates in the corresponding motion capture frame $\mathbf{F}_{\mathrm{C}^*}$ can be found. By collecting a sufficiently large number of samples, the Nelder-Mead method (Nelder and Mead, 1965) can be used to find $\mathbf{T}_{\mathrm{C}\to\mathrm{C}^*}$. With the use of a 3D model of the eyetracking glasses, in which the coordinates of the four attached markers and the pose of the world camera are known, the accelerated iterative closest point algorithm (Besl and McKay, 1992) can be used to find the transform $\mathbf{T}_{\mathrm{E}\to\mathrm{E}^*}$ between the coordinates of the markers within the model and those found using the motion capture system.

Using the transforms $\mathbf{T}_{\mathrm{E}^*\to\mathrm{C}^*}$, $\mathbf{T}_{\mathrm{C}\to\mathrm{C}^*}$ and $\mathbf{T}_{\mathrm{E}\to\mathrm{E}^*}$ it is now possible to convert between any two coordinate frames. Most importantly, this allows to map the gaze vector $\mathbf{g}^{\mathrm{E}}$ to the frame of the RGB-D camera using $\mathbf{T}_{\mathrm{E}\to\mathrm{C}}$:

$$\mathbf{g}^{\mathrm{C}} = \mathbf{T}_{\mathrm{E}\to\mathrm{C}} \cdot \mathbf{g}^{\mathrm{E}}. \tag{5.1}$$

5.2.4 *Data Collection Procedure*

At the beginning of the recording procedure, the eyetracking glasses are calibrated using a printed calibration marker, which is shown to the subject in multiple positions covering the subject's field of view while keeping the head fixed. Subsequently, in the first session, subjects are recorded for 10

minutes while wearing the eyetracking glasses. The subjects were instructed to behave naturally while varying their head poses and eye gazes as much as possible and moving within the motion capture area.

In the second session, the same subjects are recorded without the eyetracking glasses for another 10 minutes, which results in unlabeled images. These images are used for the proposed inpainting method as described in Section 5.4. To increase the variability of appearances for each subject, the 3D location of the RGB-D camera, the viewing angle towards the subject and the initial subject-camera distance are changed.

5.2.5 *Post-Processing*

The timestamps of the data points are used to synchronize the recorded images of the RGB-D camera with the gaze data $\mathbf{g}^{\mathrm{E}}$ of the eyetracking glasses. The training data is filtered to only contain head poses $\mathbf{h}^{\mathrm{C}}$ between ± 37.5 degrees horizontally and ± 30 degrees vertically, which allows accurate extraction of the images of both eyes. Furthermore, a confidence threshold of 0.98 is used to filter out blinks and images where the pupil was not detected properly (see Kassner et al., 2014, for details).

5.2.6 *Annotation Accuracy*

To estimate the error caused by $\mathbf{T}_{\mathrm{E}\to\mathrm{E}^*}$, the temporal error of the motion capture system is identified. This is conducted by statically placing the eyetracking glasses in the motion capture area and measuring their orientation over an extended period (60 seconds). The standard deviation of the measured roll, pitch and yaw angles were 0.33, 0.36 and 0.54 degrees respectively.

To estimate the error for $\mathbf{T}_{\mathrm{E}^*\to\mathrm{C}^*}$, a turntable is used to rotate the eyetracking glasses by a known amount, and this is compared to the rotation measured by the motion capture system. For a rotation of 180 degrees, a mean error of 1.01 degrees was observed.

The low error values for $\mathbf{T}_{\mathrm{E}\to\mathrm{E}^*}$ and $\mathbf{T}_{\mathrm{E}^*\to\mathrm{C}^*}$ are in line with other works that have used marker-based motion capture systems to obtain ground truth annotations (see e. g. Elhayek et al., 2017 and Rogez and Schmid, 2016).

Furthermore, the RT-GENE dataset does not exhibit some errors that are problematic in traditional datasets. In datasets built using gaze targets, training data can contain wrong annotations when the subject does not precisely gaze at the target. Despite approaches taken to mitigate these effects in such datasets (e. g. in MPII Gaze, Zhang et al., 2015, the subjects are asked to press

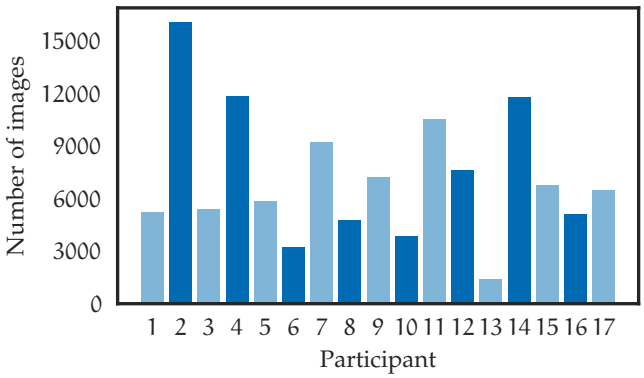


Figure 5.5: Number of images per participant in the RT-GENE dataset. Participant 13 has the fewest images (1,400), whereas participant 2 has the most images (16,000). The number of images ranges significantly as head poses outside of a specific range are filtered out as detailed in Section 5.2.5.

the space bar once the target is about to disappear), these types of errors are entirely removed from the proposed target-independent setup. Furthermore, another potential error source is due to blinking. Blinks are filtered out effectively in RT-GENE using a threshold parameter (see Section 5.2.5).

5.3 GAZE DATASET STATISTICS

The proposed RT-GENE dataset contains recordings of 17 participants (11 male, 6 female), with a total of 122,531 labeled training images (see Figure 5.5 for an illustration of the number of labeled images per participant) and 154,755 unlabeled images of the same subjects where the eyetracking glasses are not worn.

Figure 5.6 shows the head pose and gaze angle distribution across all subjects in comparison to the UT Multi-view (Sugano et al., 2014) and MPII Gaze (Zhang et al., 2015) datasets. In the RT-GENE dataset, a much higher variation is demonstrated in the gaze angle distribution, primarily due to the design of the presented setup. The free-viewing task leads to a wider spread and resembles natural eye behavior, rather than that associated with mobile device interaction or screen viewing (as in Funes Mora et al., 2014; Huang et al., 2017; Krafka et al., 2016; Zhang et al., 2015). Due to the synthesized images, the UT Multi-view dataset (Sugano et al., 2014) also covers a wide range of head pose angles; however, they are not continuous due to the fixed placing of the virtual cameras which are used to render the synthesized images.

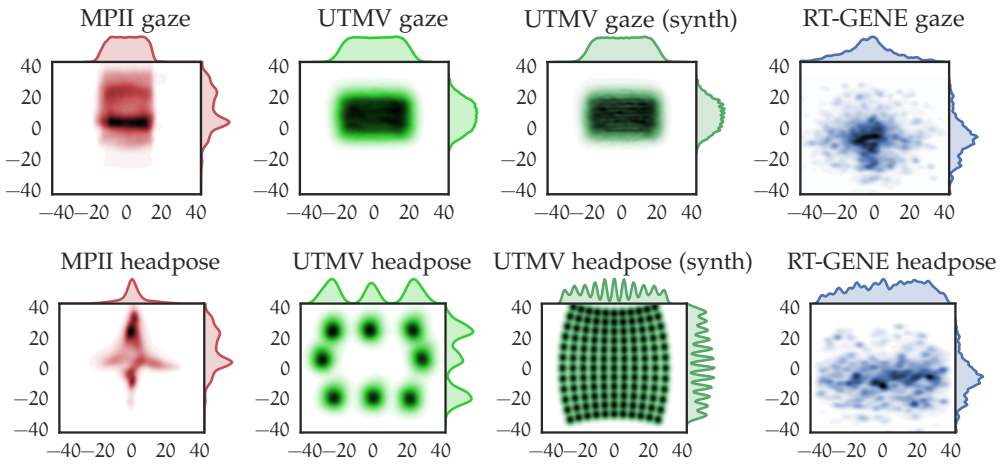


Figure 5.6: Top row: Gaze distribution of the MPII Gaze dataset (Zhang et al., 2015) (left), the UT Multi-view dataset (Sugano et al., 2014) with and without additional synthetic head poses (second from right and second from left respectively) and the proposed RT-GENE dataset (right). Bottom row: Head pose distributions, as above. The RT-GENE dataset covers a much wider range of gaze angles and head poses, which makes it more suitable for natural scenarios.

As presented in Figures 5.7 and 5.8, the camera-subject distances range between 0.5m and 2.9m, with a mean distance of 1.82m. This compares to a fixed distance of 0.6m for the UT Multi-view dataset (Sugano et al., 2014), and a narrow distribution of $0.5\text{m} \pm 0.1\text{m}$ for the MPII Gaze dataset (Zhang et al., 2015). Furthermore, the area covered by the subjects' faces is much lower in the RT-GENE dataset (mean: 100×100 px) compared to other datasets (MPII Gaze dataset mean: 485×485 px). Thus, compared to many other datasets that focus on close distance scenarios (Funes Mora et al., 2014; Huang et al., 2017; Krafka et al., 2016; Sugano et al., 2014; Zhang et al., 2015), the proposed RT-GENE dataset captures a more natural real-world setup. The RT-GENE dataset is the first to provide accurate ground truth eye gaze annotations in these settings in addition to head pose estimates.

5.4 INPAINTING OF THE EYETRACKING GLASSES

A disadvantage of using the eyetracking glasses is that they change the subject's appearance. However, when the gaze estimation framework is used in a natural setting, the subject will not be wearing the eyetracking glasses. To remove any discrepancy between training and testing data, the regions covered by the eyetracking glasses are semantically inpainted.

Image inpainting is the process of filling target regions in images by considering the image semantics. Early approaches included diffusion-based

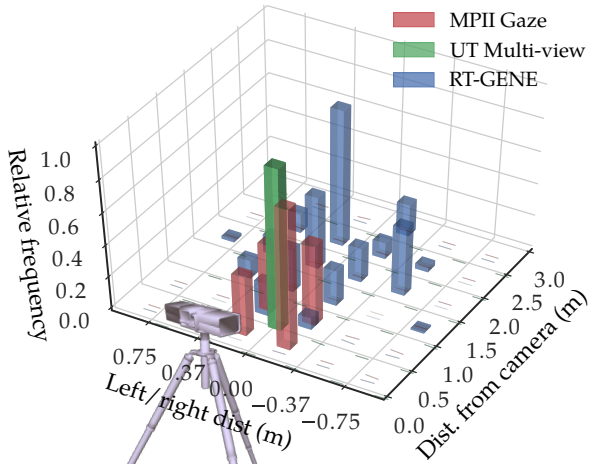


Figure 5.7: Distribution of distances between the camera and the subject for the MPII Gaze dataset (Zhang et al., 2015, red), the UT Multi-view dataset (Sugano et al., 2014, green) and the proposed RT-GENE dataset (blue). The RT-GENE dataset covers significantly more varied distances, with most camera-to-subject distances being within the range of 1.3m and 2.3m and overall distances being between 0.5m and 2.9m.

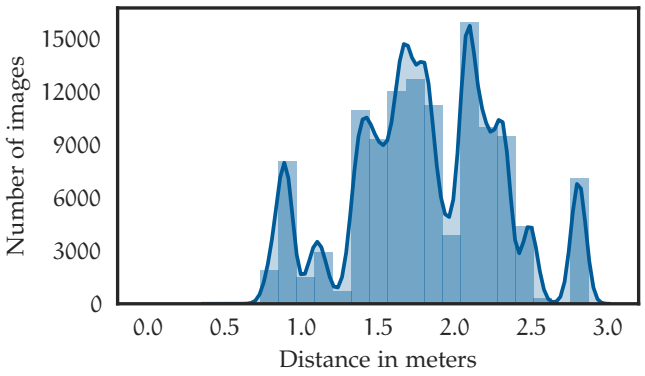


Figure 5.8: Distribution of camera-to-subject distances in the RT-GENE dataset. Figure 5.7 provides a discussion and comparison with other datasets.

texture synthesis methods (Ballester et al., 2001; Bertalmio et al., 2000; Chan and Shen, 2002), where the target area is filled by extending the surrounding textures in a coarse to fine manner. For larger regions, patch-based methods (Barnes et al., 2009; Efros and Leung, 1999; Hays and Efros, 2007; Wilczkowiak et al., 2005) that take a semantic image patch from either the input image or an image database are more successful.

Recently, semantic inpainting quality has been vastly improved through the utilization of Generative Adversarial Networks (GANs) (Iizuka et al., 2017; Pathak et al., 2016; Yeh et al., 2017). This GAN-based image inpainting approach is adopted by considering both the textural similarity to the closely surrounding area and the image semantics. To the best of my knowledge,

this thesis is the first work to make use of a semantic inpainting method for improving gaze estimation accuracy.

5.4.1 *Masking the Region of the Eyetracking Glasses*

The CAD model of the eyetracking glasses is made up of a set of $N = 2662$ vertices $\{\mathbf{v}_n\}_{n=1}^{N}$, with $\mathbf{v}_n \in \mathbb{R}^3$. To find the target region that is to be inpainted, $\mathbf{T}_{E\to C}$ is used for deriving the 3D position of each vertex in the RGB-D camera frame. For extreme head poses, the subject's head may obscure certain parts of the eyetracking glasses. Thus, masking all pixels would result in part of the image being inpainted unnecessarily. To overcome this problem, the indicator function

$$\mathbf{1}_{\mathbf{M}}(\mathbf{p}_n, \mathbf{v}_n) = \left\{0 \text{ if } \left\|\mathbf{p}_n - \mathbf{v}_n\right\| < \tau, \text{else } 1\right\} \tag{5.2}$$

is used to select vertices $\mathbf{v}_n$ of the CAD model if they are within a tolerance τ of their corresponding point $\mathbf{p}_n$ in the depth field. Each selected vertex is mapped using the camera projection matrix of the RGB-D camera into a 2D image mask $\mathbf{M} = \{m_{i,j}\}$, where each entry $m_{i,j} \in \{0, 1\}$ shows whether the pixel at location (i, j) needs to be inpainted.

5.4.2 *Semantic Inpainting*

A GAN-based image generation approach, similar to that of Yeh et al. (2017), is used to seamlessly fill the masked regions of the eyetracking glasses. There are two conditions to fulfill (Yeh et al., 2017): the inpainted result should look realistic (perceptual loss $\mathcal{L}_{\text{perception}}$) and the inpainted pixels should be well-aligned with the surrounding pixels (contextual loss $\mathcal{L}_{\text{context}}$). As shown in Figure 5.9, the resolution of the face area is larger than the 64×64px supported in Yeh et al. (2017). The proposed architecture allows the inpainting of images with resolution 224×224px[1]. This increased resolution is a crucial feature, as reducing the face image resolution for inpainting purposes could impact the gaze estimation accuracy.

Appendix B details the semantic inpainting methodology. It defines the loss functions that were used, details the network architectures and provides the parameters that were used for training the networks. Figure 5.10 shows the application of inpainting in within RT-GENE.

1 While the mean face area in the RT-GENE dataset is 100×100px as shown in Figure 5.9, the 95$^{\text{th}}$ percentile is 175×175px, which is considerably larger than the 64×64px supported by Yeh et al. (2017).

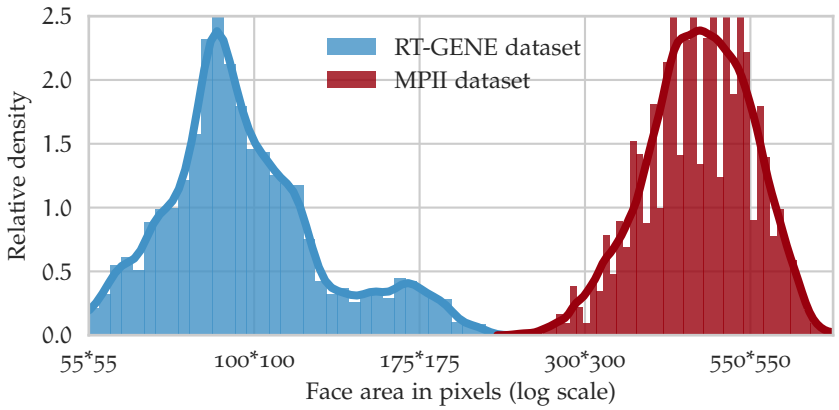


Figure 5.9: Face area distribution in the MPII Gaze dataset (Zhang et al., 2015, red) and the proposed RT-GENE dataset (blue). The resolution of the face areas in the RT-GENE dataset is much lower (mean 100×100px) than that of the MPII Gaze dataset (mean 485×485px). This is mainly due to the larger camera-subject distance (as shown in Figure 5.7).

5.5 GAZE ESTIMATION NETWORKS

As shown in Figure 5.1, the gaze estimation is performed using several deep networks. Firstly, Multi-task Cascaded Convolutional Networks (MTCNN, Zhang et al., 2016) detects the face along with the landmark points of the eyes, nose and mouth corners. Using the extracted landmarks, the face patch is rotated and scaled such that the distance between the aligned landmarks and predefined average face point positions is minimized. This process is implemented using the accelerated iterative closest point algorithm (Besl and McKay, 1992) and results in a normalized face image. The eye patches are then extracted from the normalized face images as fixed-size rectangles centered around the landmark points of the eyes.

Secondly, the head pose of the subject is found by adopting the state-of-the-art method presented by Patacchiola and Cangelosi (2017). Compared to the head pose estimation method presented in Section 3.1.3, the method by Patacchiola and Cangelosi (2017) results in similarly accurate head pose estimates while using only RGB images rather than RGB-D input (Patacchiola and Cangelosi, 2017, provide a detailed comparison). This provides more flexibility; for example, the iCub eye cameras can now be used directly for head pose and gaze estimation rather than requiring an external RGB-D camera. It also allows the use of RT-GENE on webcams and laptop cameras.



Figure 5.10: Image pairs show the original images of the subject wearing the eye-tracking glasses (left) and the corresponding inpainted images (right). The inpainted images look very similar to the subjects' appearance at testing time and are thus suited for training an appearance-based gazed estimator.

5.5.1 *Eye Gaze Estimation*

Then the eye gaze vector is estimated using the proposed network. The eye patches are fed separately to VGG-16 networks (Simonyan and Zisserman, 2015) which perform feature extraction. Each VGG-16 network is followed by a fully connected (FC) layer of size 512 after the last max-pooling layer, followed by batch normalization and ReLU activation. These layers are then concatenated, resulting in an FC layer of size 1024. This layer is followed by another FC layer of size 512. The head pose vector is appended to this FC layer, which is followed by two more FC layers of size 256 and 2 respectively. These layer sizes were determined experimentally. The outputs of the last layer are the yaw and pitch eye gaze angles. The robustness is increased using an ensemble scheme (Krizhevsky et al., 2017) where the mean of the predictions of the individual networks represents the overall prediction.

5.5.2 *Image Augmentation*

The robustness of the gaze estimator is increased by augmenting the training images in the following four ways. Firstly, to be robust against slightly off-centered eye patches due to imperfections in the landmark extraction, ten augmentations are performed by cropping the image on the sides and subsequently resizing it back to its original size. Each side is cropped by a pixel

value drawn independently from a uniform distribution $\mathcal{U}(0,5)$. Secondly, for robustness against camera blur, the image resolution was reduced to $1/2$ and $1/4$ of its original resolution, followed by a bilinear interpolation to retrieve two augmented images of the original image size. Thirdly, to cover various lighting conditions, histogram equalization is employed. Finally, color images are converted to gray-scale images so that gray-scale images can be used as input as well. This results in a total of fourteen image augmentations.

5.5.3 *Training Details*

The loss function is defined as the sum of the individual l_2 losses between the predicted and ground truth gaze vectors. The weights for the network estimating the head pose are fixed and taken from a pre-trained network (Patacchiola and Cangelosi, 2017). The weights of the VGG-16 networks are initialized using a pre-trained network on ImageNet (Simonyan and Zisserman, 2015). As weight sharing resulted in decreased performance, it is not being used. The weights of the FC layers are initialized using the Xavier initialization (Glorot and Bengio, 2010). The Adam optimizer (Diederik P. Kingma, 2015) is used with learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.95$ and a batch size of 256.

5.6 EXPERIMENTAL EVALUATION

The evaluation of the architecture presented in this chapter is split into four parts. Section 5.6.1 investigates the effectiveness of the proposed inpainting algorithm. This is followed by an evaluation of the proposed gaze estimator on several datasets including RT-GENE in Section 5.6.2. The cross-dataset evaluations in Section 5.6.3 demonstrate that the estimators do not only perform well when evaluated on the testing set of the dataset that is used to train the gaze estimator but also generalizes to other datasets. Finally, Section 5.6.4 contains some qualitative results of the gaze estimation.

5.6.1 *Dataset Inpainting Evaluation*

The first set of experiments validate the effectiveness of the proposed inpainting algorithm. The average pixel error of five facial landmark points (eyes, nose and mouth corners) was compared to manually collected ground



Figure 5.11: Left column: Original face images and landmarks found by MTCNN. One can see that the landmark positions are not precise, which is due to the dissimilarity to the images MTCNN was trained on. Middle column: Inpainted face images and refined landmarks found by MTCNN. The landmarks positions are now found accurately, which allows extracting precise eye image patches. Right column: The top image shows the extracted eye patch before inpainting, while the bottom image shows the refined extracted eye patch. In the refined image patch, the location of the eye center is estimated more accurately, and the yellow patches of the glasses have been inpainted.

truth labels on a set of 100 images per subject before and after inpainting ("original" vs. "inpainted" images). The results reported in Tables 5.3 and 5.4 confirm that all landmark estimation algorithms benefit from the inpainting, both in increased face detection rate and in lower pixel error. The performance of the proposed inpainting method is also significantly higher than a method that naively fills the area of the eyetracking glasses uniformly with the mean color. Figure 5.11 shows qualitative results for the inpainting method.

The landmark points were also extracted for images where the subject does not wear the eyetracking glasses ("natural" images). Statistical analysis using Wilcoxon signed-rank tests has shown that there is a statistical difference between the natural images and original images ($p < .01$), and between the original images and inpainted images ($p < .01$). This was also the case for the face detection rate between inpainted images and natural images ($p < .01$). Importantly, however, there was no statistical difference of the landmark extraction accuracy between the inpainted images where the face

Table 5.3: Comparison of the Constrained Local Neural Fields (CLFN; Baltrusaitis et al., 2013) and Ensemble Regression Trees (ERT; Kazemi and Sullivan, 2014) landmark detectors applied to the original images (with eyetracking glasses), images where the eyetracking glasses are filled with a uniform color (the mean color of the image), and images that were inpainted using the proposed semantic inpainting method. The face detection rate improves significantly when inpainted images are provided as input to the landmark detector. However, the detection rate does not achieve the same accuracy as in natural images. The performance of MTCNN (Zhang et al., 2016) is not reported, as it would be a biased comparison, given that MTCNN was used to extract the face patches for this comparison.

	Face detection rate (%)			
Landmark detection method	Original	Uniformly filled	Inpainted	Natural
CLFN	54.6±24.7	75.4±20.9	**93.3±10.0**	98.7±2.9
ERT	36.7±25.3	59.7±23.0	**89.5±13.9**	97.6±5.5

Table 5.4: Similar comparison as in Table 5.3 but comparing the landmark error in pixels rather than the face detection rate. The landmark error is significantly reduced for the inpainted images compared to the original images. Moreover, there is no statistical difference between the error for the inpainted and natural images.

	Landmark error (pixel)			
Landmark detection method	Original	Uniformly filled	Inpainted	Natural
CLFN	6.0±2.4	5.6±2.3	**5.3±1.8**	5.0±2.7
CLFN in-the-wild	5.8±2.3	5.3±1.8	**5.2±1.6**	5.1±3.0
ERT	6.6±2.3	5.8±1.7	**5.1±1.3**	4.9±2.4

was detected and natural images ($p = .16$), which further validates the effectiveness of the proposed inpainting method and supports the contribution of the RT-GENE dataset.

5.6.2 *Gaze Estimation Performance Evaluation*

The proposed method is evaluated on two de facto standard datasets, MPII Gaze (Zhang et al., 2015) and UT Multi-view (Sugano et al., 2014), as well as the newly proposed RT-GENE dataset[2].

2 There are no comparisons conducted using Eyediap dataset (Funes Mora et al., 2014) or the dataset of Deng and Zhu (2017) due to licensing restrictions of these datasets.

PERFORMANCE ON THE MPII DATASET

First, the performance of the newly proposed gaze estimation network is evaluated on the MPII Gaze dataset (Zhang et al., 2015). The MPII Gaze dataset uses an evaluation set containing 1500 images of the left and right eye respectively. As the proposed method employs both eyes as input, the 3000 images are directly used without considering the target eye. The previous state-of-the-art achieves an error of 4.8 ± 0.7 degrees (Zhang et al., 2017) in a leave-one-out setting. Figure 5.12 shows that the proposed method's accuracy is 4.3 ± 0.9 degrees (10.4% improvement).

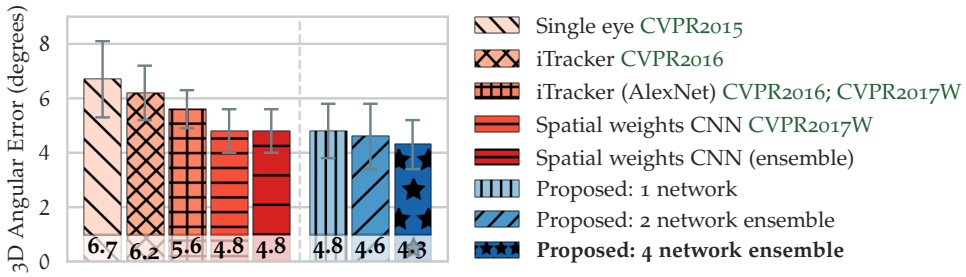


Figure 5.12: 3D gaze error on the MPII Gaze dataset. The proposed four-network ensemble performs best. A single network achieves a performance comparable to the previous state-of-the-art method (CVPR2017W by Zhang et al., 2017). However, applying an ensemble scheme to the CVPR2017W method does not improve the performance as discussed further in the text.

PERFORMANCE ON THE UT MULTI-VIEW DATASET

In evaluations on the UT Multi-view dataset (Sugano et al., 2014), the proposed method achieves a mean error of 5.1 ± 0.2 degrees, performing favorably against the previously best performing method (Zhang et al., 2015) by 13.6% (5.9 degrees error). These results demonstrate that the proposed method achieves state-of-the-art performance on two existing datasets.

PERFORMANCE ON THE RT-GENE DATASET

In the third set of experiments, the gaze estimators' performances are measured on the newly proposed RT-GENE dataset using 3-fold cross validation as shown in Figure 5.13. All methods perform worse on RT-GENE compared to the MPII Gaze and UT Multi-view datasets, which is due to the natural setting with larger appearance variations and lower resolution images due to higher camera-subject distances. Using inpainted images at training time results in higher accuracy compared to using the original images without inpainting for all algorithms including the proposed algorithm (10.5%

performance increase). For the inpainted images, the proposed gaze estimation network achieves the best performance with an error of 7.7 ± 0.3 degrees, which compares to the method of Zhang et al. (2015) with an error of 13.4 ± 1.0 degrees (42.5% improvement) and the previous state-of-the-art network (Zhang et al., 2017) with 8.7 ± 0.7 degrees error (11.5% improvement). These results demonstrate that features obtained using the proposed deep network architecture are more suitable for this dataset compared to the previous state-of-the-art.

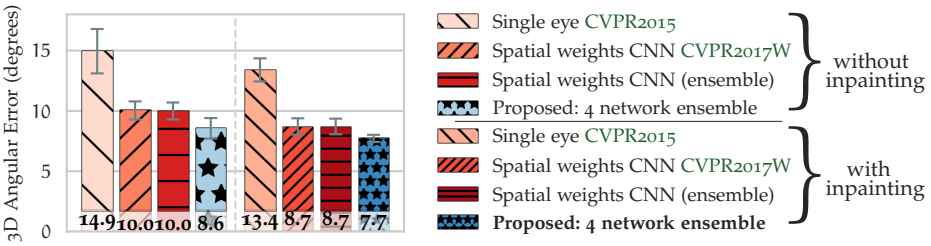


Figure 5.13: 3D gaze error on the proposed RT-GENE gaze dataset. The inpainting improves the gaze estimation accuracy for all algorithms. The proposed method performs best with an accuracy of 7.7 degrees.

ENSEMBLE SCHEME EVALUATION

Furthermore, ensemble schemes were found to be particularly effective in the RT-GENE architecture. For a fair comparison, the ensemble scheme was also applied to the state-of-the-art method by Zhang et al. (2017). However, there was no performance improvement over the single network (see Figure 5.12). This is due to the spatial weights scheme that leads to similar weights in the intermediate layers of the different networks. This, in turn, results in similar gaze predictions of the individual networks, and therefore an ensemble scheme does not improve the accuracy for this particular method.

5.6.3 *Cross-Dataset Evaluation*

To further validate whether the RT-GENE dataset can be applied in a variety of settings, the proposed ensemble network was trained on samples from the RT-GENE dataset (all subjects included) and tested on the MPII Gaze dataset (Zhang et al., 2015). This is challenging, as the facial appearance and image resolution is very different as shown in Figures 5.7, 5.9 and 5.14. This resulted in an accuracy of 7.7 ± 1.3 degrees, which outperforms the currently best performing method in a similar cross-dataset evaluation (Wood et al., 2016, 9.9 degrees error, 22.4% improvement). The proposed method also out-

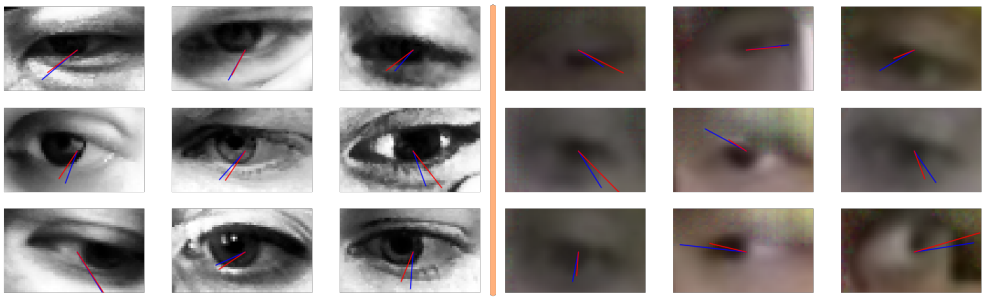


Figure 5.14: Sample estimates (red) and ground truth annotations (blue) using the proposed method on the dataset (Zhang et al., 2015) (left) and the proposed RT-GENE dataset (right). The RT-GENE dataset is more challenging, as images are blurrier due to the higher subject-camera distance and show a higher variation in head pose and gaze angles.

performs the method of Shrivastava et al. (2017, 7.9 degrees error), which uses unlabeled images of the MPII Gaze dataset at training time, while the proposed method uses none.

To investigate whether the reported improvements are due to the proposed gaze estimator or the new RT-GENE dataset, an ensemble network was trained on the UT Multi-view dataset (all subjects and head poses included) instead of RT-GENE as above and this ensemble was then evaluated on the MPII Gaze dataset. Therefore, the same estimator is used to train the ensemble, and the only difference is the employed dataset for training (UT Multi-view vs. RT-GENE). For the ensemble trained on UT Multi-view, the angular error is 8.9 ± 1.5 degrees, compared to an error of 7.7 ± 1.3 degrees for the ensemble trained on RT-GENE. This confirms that while the proposed gaze estimator leads to an improved performance overall, the RT-GENE dataset is of importance for the generalization capability of the ensemble networks.

5.6.4 *Qualitative Results and Practical Application*

Figure 5.14 shows some qualitative results of the proposed method applied to the MPII Gaze and RT-GENE datasets. The framework can be used for real-time gaze estimation using any RGB or RGB-D camera such as Kinect, webcam, laptop and the iCub eye cameras. The framework runs at 25.3 fps with a latency of 0.12s using an Intel i7-6900K@3.2GHz with a Nvidia 1070 and 64GB RAM.

The best results in practical settings are achieved when the proposed framework is trained using multiple datasets, e. g. the RT-GENE and MPII datasets. One reason is that in this case samples where the subject is in very

close distance to the camera (MPII dataset) are merged with samples where the distance is much larger (RT-GENE dataset). This allows application of the framework in an even wider range of distances compared to just using samples from the RT-GENE dataset for training. Furthermore, the performance of the framework is usually harmed for subjects that wear eyeglasses to correct their vision, as the RT-GENE dataset does not contain subjects wearing eyeglasses. This problem is avoided as the MPII dataset contains some subjects that wear eyeglasses, and thus the gaze estimator can learn to extract features and estimate the gaze accurately even for these subjects.

5.7 CONCLUSIONS

This chapter proposed RT-GENE, a novel approach for ground truth gaze estimation in natural settings. A new challenging dataset was collected using this approach, and it was demonstrated that the dataset covers a wider range of camera-subject distances, head poses and gazes compared to previous in-the-wild datasets. It was also demonstrated that semantic inpainting using GAN can be used to overcome the appearance alteration caused by the eyetracking glasses during training. The proposed deep convolutional network achieved state-of-the-art gaze estimation performance on the MPII Gaze dataset (10.4% improvement), UT Multi-view (13.6% improvement), the proposed RT-GENE dataset (11.5% improvement), and in cross-dataset evaluation (22.4% improvement). Overall, RT-GENE allows for accurate gaze estimation in scenarios where the head pose of the subject previously approximated the gaze. The RT-GENE dataset and code are available for download: https://www.imperial.ac.uk/personal-robotics/software/.

The proposed inpainting method could be applied to bridge the gap between training and testing in settings where wearable sensors are attached to a human (e. g. Electroencephalography (EEG), Electromyography (EMG) and Inertial Measurement Unit (IMU) sensors). Another interesting scenario could be the inpainting of augmented/virtual reality devices. One requirement of the inpainting is the masking of the area to be inpainted, which requires precise pose estimates. As augmented/virtual reality devices provide pose estimates using external sensors (e. g. HTC Vive[3], Oculus Rift[4]) or SLAM (e. g. Microsoft Hololens[5]), the proposed inpainting method could be easily applied.

3 https://www.vive.com/
4 https://www.oculus.com/rift/
5 https://www.microsoft.com/en-us/hololens/

Future works will investigate gaze estimation in situations where the eyes of the participant cannot be seen by the camera, e. g. for extreme head poses or when the subject is facing away from the camera (in this instance, the eyes cannot be seen by the camera). As the RT-GENE dataset collection method allows annotation of gaze even in these diverse conditions, it would be interesting to explore algorithms which can handle these challenging situations. One hypothesis is that the saliency information of the scene could prove useful in this context, as humans are inclined to look at salient objects. The framework could then be used for intention recognition, where the inferred 3D gaze position could be used as a cue. These research directions are discussed further in Section 7.2.3.

6

EMBODIED TRANSFORMATION AS COMPUTATIONAL MODEL FOR PERSPECTIVE TAKING

This chapter addresses the last research question:

> "How can perspective taking be modeled from a computational point of view, and how do the model's outputs compare to data from experiments with humans?"

Chapter 3 introduced an artificial visual system that allows a robot to take the perspective of a human and used this ability to make well-informed decisions given a human's speech command. This chapter investigates the mechanisms that underlie perspective taking in humans by developing a computational model and comparing the model responses with human responses. Importantly, the computational model also provides testable predictions that can be experimentally validated in future psychological studies with humans.

This chapter is an extended version of the research that has been previously published in Fischer and Demiris (2018).

6.1 MOTIVATION

The socio-cognitive skills of the human brain are the product of prolonged childhood development (Heckman, 2006; Herrmann et al., 2007), where fundamental skills such as a theory of mind (Premack and Woodruff, 1978; Surtees et al., 2013*b*) and a capacity for perspective taking (Kessler and Thomson, 2010; May, 2004; Michelon and Zacks, 2006) are developed. Possessing a theory of mind implies being aware that other people's visual and mental states differ from one's own (Frith and Frith, 1999). This requires an understanding of how the physical space is perceived from the viewpoint of another person (Flavell, 1977), which is referred to as perspective taking (Michelon and Zacks, 2006; Salatas and Flavell, 1976) as discussed in Chapter 1 of the thesis. Together these skills are used to analyze and infer the intentions of others (Blakemore and Decety, 2001).

One hypothesis, known as the embodied transformation account (Kessler and Thomson, 2010, see Section 2.5.3), suggests that perspective taking (PT)

is the mental simulation of the physical rotation or translation necessary to acquire another perspective. Moreover, the body representations used for the mental simulation are identical to the ones used for physical movement. In other words, PT emerges as a variation of physical movements that are simulated rather than executed.

Several works have endorsed this hypothesis and provided support in terms of psychological (Kessler and Thomson, 2010; Kessler and Wang, 2012; Surtees et al., 2013*b*; Watanabe, 2016; Yu and Zacks, 2017) and neurophysiological (Gooding-Williams et al., 2017; Wang et al., 2016) data. However, the hypothesis has not yet been investigated using a computational model which represents a concrete implementation of the hypothesis. This chapter introduces such a model and shows an implementation of the model on a simulated robot platform in order to systematically test this hypothesis computationally.

The chapter advocates that PT is governed by a competition process for visual attention that selects action primitives that should be passed through the forward model. The forward model predicts the agent's state given the current state and a (mentally simulated) action primitive that acts as a motor input (Demiris and Hayes, 2002). As the forward model is recurrently executed, an internal (mental) representation of the simulated state is required. The model explains the response times of humans in PT experiments that contain the following variations: 1) the angular disparity between the self-agent and target agent, 2) the body posture of the self-agent, and 3) the body posture of the target agent. On account of the computational formalization, the model provides further explanation of the precise mechanisms of embodied simulation and generates predictions to be tested in further experimental psychology studies. Specifically, the predictions state that forced early responses lead to an egocentric bias and that habituation effects occur when the imagined movement direction of the previous trial matches that of the current trial.

The chapter is structured as follows. Section 6.2 provides a formalization of perspective taking using two simulated iCub robots. Then, Section 6.3 details the experimental setup and justifies the parameter choices. Section 6.4 discusses the proposed computational model as a model for human PT mechanisms. It provides qualitative and quantitative comparisons to a variety of experiments that were conducted with human subjects. The model offers several testable predictions that are presented in Section 6.5. Finally, Section 6.6 concludes this chapter.

6.2 COMPUTATIONAL FORMALIZATION OF PERSPECTIVE TAKING

This section formalizes visual PT as an embodied simulation of physical movements using a set of action primitives that are passed through a forward model. It also introduces a visual attentional mechanism that improves computational efficiency by favoring the execution of previously employed action primitives. This will be shown to be essential to explain the response times of humans in PT tasks in Section 6.4.

An overview of how computational formalization of perspective taking is achieved is as follows (refer to Figure 6.1). The self-agent and target agent are represented by states that contain the torso pose, head pose and eye gaze of the respective agents (Section 6.2.1). A forward model outputs a state estimate given the current state and an action primitive to be executed (Section 6.2.2). The embodied perspective taking process is implemented so that the self-agent mentally aligns its perspective with that of the target agent. This is implemented by executing the forward model for all action primitives and choosing the primitive that results in the lowest expected distance between the self-agent and target agent (Section 6.2.3). The distance is a weighted average with two terms: one where the torso, head and eye angles of the self-agent and target agent have to match individually, and another where only the final gaze direction is considered (Section 6.2.4). The model's response time corresponds to the number of forward model passes. An attentional component reduces the response time by selecting a subset of action primitives to be passed through the forward model, rather than executing all action primitives (Section 6.2.5).

6.2.1 *Visual Perspective and Agent States*

The two agents, namely the self-agent and target agent, are represented by the states $\mathbf{z}(k)$ and $\hat{\mathbf{z}}(k)$ respectively. In Chapter 3 the state was approximated by the head pose, and Chapter 5 extended it by the eye gaze of the agent. Within this chapter, the agent embodiment is taken further and the torso pose is also considered, thus head pose, eye gaze and torso pose are collectively considered as the perspective of an agent. The incorporation of the torso pose in what is termed perspective is required as the torso pose impacts the response times of the model, which is further detailed in Section 6.2.4 and experimentally shown in Sections 6.4.2 to 6.4.4.

More formally, an agent state $\mathbf{z}(k) = \{\mathbf{z}_0(k), ..., \mathbf{z}_{N-1}(k)\}$ is composed of N joint states. As the planar problem is considered, each $\mathbf{z}_n(k)$ contains two

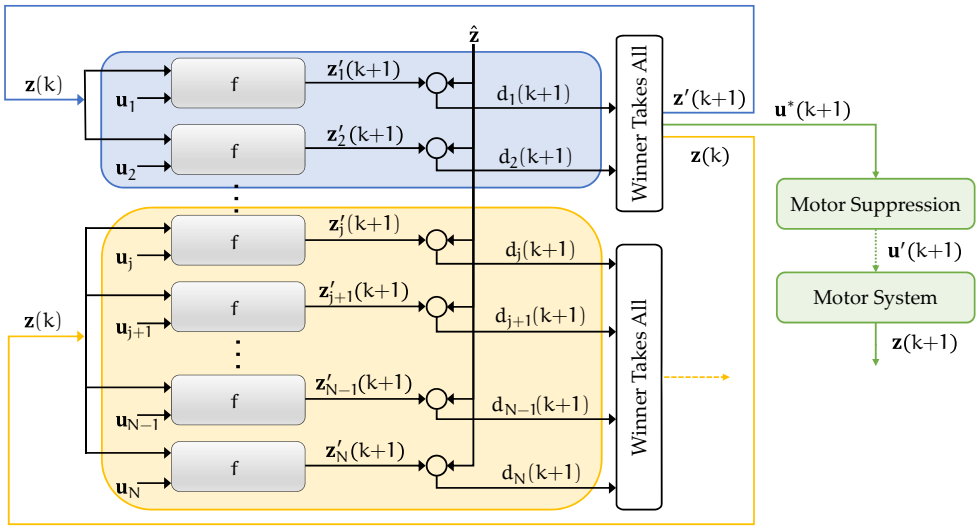


Figure 6.1: One instantiation of the computational model. The forward model f is used to provide state estimates $\mathbf{z}'_i(k+1)$ for the self-agent given the current state $\mathbf{z}(k)$ and an action primitive $\mathbf{u}_i$. In the first phase of this model instance, $\mathbf{u}_1$ and $\mathbf{u}_2$ are contained in an attentional set (blue box) and fed into the forward models. The state estimates are compared to the target agent's state $\hat{\mathbf{z}}$, resulting in the distances $d_1(k+1)$ and $d_2(k+1)$. If none of these distances are smaller than $d(k)$, then a second phase ensues (orange box). The primitive $\mathbf{u}^*(k+1)$ that results in the lowest distance is mentally executed.

translational components $x_n(k)$ and $y_n(k)$, and one rotational component $\theta_n(k)$. All components are relative to the parent joint $n-1$ and joints are defined for the torso (root joint, $n=0$), head ($n=1$), and eyes ($n=2$). As the self-agent perceives the world from an egocentric perspective, all components of the torso's joint state are 0: $\mathbf{z}_0 = \mathbf{0}$. Furthermore, for simplicity, the head and eye joints remain in a fixed position with respect to the torso and can only rotate in place, thus $x_1 = y_1 = 0$ and $x_2 = y_2 = 0$.

6.2.2 *Forward Model and Action Primitives*

The architecture makes extensive use of the forward model $f(\mathbf{z}(k), \mathbf{u}_i(k))$, which provides the (simulated) predicted state $\mathbf{z}'_i(k+1)$ given the current state $\mathbf{z}(k)$ of the self-agent and a motor input $\mathbf{u}_i(k)$ at iteration k. The forward model is known thanks to the predefined kinematic model of the robot provided by the simulator. The motor inputs $\{\mathbf{u}_i \mid \mathbf{u}_i \in \mathbf{U}\}$ are implemented as action primitives (move forward, move left, rotate torso left, etc.) of 0.1 units of translation or 10 degrees of rotation. A full list of all action primitives can be found in Table 6.1.

Table 6.1: This table contains a list of all action primitives, along with their impact on the state of the robot. While the architecture has been shown to be robust against the precise choice of action primitives, within the thesis the following primitives have been chosen. Primitive 1: no movement; primitives 2–5: translational movement in either direction; primitives 6–9: combinations of the translational movements; primitives 10–15: rotational movement of each individual joint in either direction; primitives 16–19: combinations of the rotational movements; primitives 20 and 21: some combinations of translational and rotational movements that may frequently occur.

Action primitive name	Index	Δx_0	Δy_0	$\Delta\theta_0$	$\Delta\theta_1$	$\Delta\theta_2$
No move	1	0	0	0	0	0
Move forward	2	$+0.1$	0	0	0	0
Move backward	3	-0.1	0	0	0	0
Move left	4	0	$+0.1$	0	0	0
Move right	5	0	-0.1	0	0	0
Move forward left	6	$+0.1\sqrt{2}/2$	$+0.1\sqrt{2}/2$	0	0	0
Move forward right	7	$+0.1\sqrt{2}/2$	$-0.1\sqrt{2}/2$	0	0	0
Move backward left	8	$-0.1\sqrt{2}/2$	$+0.1\sqrt{2}/2$	0	0	0
Move backward right	9	$-0.1\sqrt{2}/2$	$-0.1\sqrt{2}/2$	0	0	0
Rotate torso left	10	0	0	$+10$	0	0
Rotate torso right	11	0	0	-10	0	0
Rotate head left	12	0	0	0	$+10$	0
Rotate head right	13	0	0	0	-10	0
Rotate eyes left	14	0	0	0	0	$+10$
Rotate eyes right	15	0	0	0	0	-10
Rotate torso left without head	16	0	0	$+10$	-10	0
Rotate torso right without head	17	0	0	-10	$+10$	0
Rotate torso and head left	18	0	0	$+10$	$+10$	0
Rotate torso and head right	19	0	0	-10	-10	0
Move forward left rotate right	20	$+0.1\sqrt{2}/2$	$+0.1\sqrt{2}/2$	-10	0	0
Move forward right rotate left	21	$+0.1\sqrt{2}/2$	$-0.1\sqrt{2}/2$	$+10$	0	0

The goal of the self-agent is to mentally adopt the visual perspective of the target agent, $\hat{\mathbf{z}}$, which is assumed to be static and known (the obtention of the target agent's perspective is discussed in Chapter 5). One could argue that knowing $\hat{\mathbf{z}}$ is all what it takes to solve the PT task, however this is not the case – knowing $\hat{\mathbf{z}}$ means that the gaze direction (along with the configuration of the other joints) is known, however only the mental adoption of this viewpoint allows the self-agent to infer how the world is perceived from this perspective.

The motor inputs are inhibited from being sent to the motor system, which results in a feed-forward control system. Hence, this suggests that there is a visuospatial memory representation of the mentally transformed self, which

is updated over time in a simulation loop. In other words, the predicted output $\mathbf{z}'(k+1)$ is used as input $\mathbf{z}(k+1)$ at iteration $k+1$.

6.2.3 *Distance Metric and Control Policy*

The distance metric $d\big(\mathbf{z}(k), \hat{\mathbf{z}}\big)$ is defined so that:

$$d\big(\mathbf{z}(k), \hat{\mathbf{z}}\big) = d_S\big(\mathbf{z}(k), \hat{\mathbf{z}}\big) + d_\theta\big(\mathbf{z}(k), \hat{\mathbf{z}}\big), \tag{6.1}$$

where d_S is the Euclidian distance between the translational components of the root joints of the self-agent ($\mathbf{z}_0(k)$) and target agent ($\hat{\mathbf{z}}_0$), and d_θ is a measure of the angular disparity between the agents (further details are provided in Section 6.2.4).

The aim is to find a control policy $\mathbf{u}^*(k) = \boldsymbol{\pi}\big(\mathbf{z}(k), f\big)$ that minimizes d, such that $d\big(\mathbf{z}(k), \hat{\mathbf{z}}\big) < d\big(\mathbf{z}(k-1), \hat{\mathbf{z}}\big), \forall k < k_{goal}$ and $d\big(\mathbf{z}(k_{goal}), \hat{\mathbf{z}}\big) < \epsilon$, where ϵ is a distance threshold acting as termination criterion, and k_{goal} is the iteration (point in time) where the estimated distance falls below this threshold. The control policy $\boldsymbol{\pi}$ is chosen such that the optimal action primitive $\mathbf{u}^*(k)$ is found by executing the forward model f for all action primitives and choosing the primitive $\mathbf{u}_i$ that results in the lowest expected distance:

$$\mathbf{u}^*(k+1) = \arg\min_{\mathbf{u}_i} d\Big(f\big(\mathbf{z}(k), \mathbf{u}_i\big), \hat{\mathbf{z}}\Big), \tag{6.2}$$

and stop the process once $d\big(\mathbf{z}(k_{goal}), \hat{\mathbf{z}}\big) < \epsilon$.

6.2.4 *Alignment Strategy*

In the visuospatial perspective taking literature, it thus far remains unclear to which frame of reference the self-agent aligns the perspective to, and whether each joint is individually matched, or a combination of the reference frames is matched (Alsmith et al., 2017). For instance, several body configurations enable an agent to look in the same direction – the direction of the torso does not impact the perceived environment as long as the head and eyes remain in the same configuration. The question is thus whether the perspective taker aligns the perspective such that the torso, head and eye angles all match $\big(\mathbf{z}(k_{goal}) \approx \hat{\mathbf{z}}\big)$, or so that only the final gaze direction is

considered. Hence, the weighted average with mixing parameter $0 \leqslant \omega \leqslant 1$ is introduced such that:

$$d_\theta\big(\mathbf{z}(k), \hat{\mathbf{z}}\big) = (1-\omega)\, d_I\big(\mathbf{z}(k), \hat{\mathbf{z}}\big) + \omega\, d_\Sigma\big(\mathbf{z}(k), \hat{\mathbf{z}}\big), \text{with} \tag{6.3}$$

$$d_I\big(\mathbf{z}(k), \hat{\mathbf{z}}\big) = \sum_{n=0}^{N-1} \Big|\theta_n(k) - \hat{\theta}_n(k)\Big|, \text{and} \tag{6.4}$$

$$d_\Sigma\big(\mathbf{z}(k), \hat{\mathbf{z}}\big) = \left|\sum_{n=0}^{N-1} \theta_n(k) - \sum_{n=0}^{N-1} \hat{\theta}_n(k)\right| \tag{6.5}$$

while ensuring that all angle differences are in the interval $[-\pi, \pi]$. Therefore, an agent with $\omega = 0$ matches each joint individually to those of the target agent (note that this is task irrelevant in PT tasks), whereby an agent with $\omega = 1$ considers only the final gaze direction.

6.2.5 *Response Time and Attentional Component*

This section introduces an attentional component to reduce the response time of the proposed model. The response time C is defined as the number of forward passes that are executed. In other words, for each action primitive that is considered, the response time increases by 1.

To reduce the response time, an attentional component selects a subset $\mathbf{A}(k) \subsetneq \mathbf{U}$ of action primitives to be passed through the forward model at iteration k (rather than passing all action primitives $u_i \in \mathbf{U}$). The selection is governed such that the action primitive $u^*(k-1)$ that was executed at the previous iteration becomes one element of $\mathbf{A}(k)$. The other elements are selected randomly[1].

The action primitives that are not included in the attentional set $\big(u_j \notin \mathbf{A}(k)\big)$ are only executed if no suitable action primitive is found within $\mathbf{A}(k)$ (i. e. none of the initially executed primitives reduced d). Therefore, the number of forward passes per iteration equals the number of attentional components $|\mathbf{A}|$ in case a suitable action primitive is found, or the total number of action primitives $|\mathbf{U}|$ if no suitable action primitive is found within $\mathbf{A}(k)$.

1 Two additional comparisons for other scheduling mechanisms were performed. In one comparison, the other elements are based on the similarity of the action primitives $\mathbf{u}_i$. For example, the primitives 'rotate torso left' and 'rotate head left' have a high similarity, while 'move forward' and 'move backward' have low similarity. In the other comparison, a round robin scheduling was used. For both comparisons, no significant differences where found. Other knowledge-based scheduling mechanisms that might reduce the model's response time further will be investigated in future works.

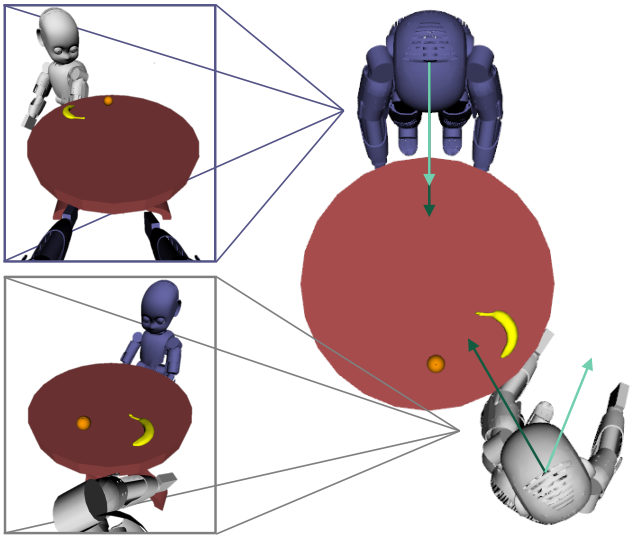


Figure 6.2: Experimental setup. The task of the blue robot is to take the perspective of the gray robot, and to decide whether one of the objects (e. g. the orange) is to the left or right as perceived by the gray robot. The images on the top and bottom left show the scene as perceived by the blue and gray robot respectively. Bright green arrows indicate the torso's pose, and dark green arrows the head pose.

6.3 EXPERIMENTAL EVALUATION

This section investigates the properties of the computational model applied to two simulated iCub humanoid robots in a visual PT task. Simulated rather than physical robots were more suitable for this study, and are chosen for the following reasons. Firstly, using simulated robots allows a large-scale systematic evaluation of the model. The following comparisons contain variations in the experimental setup (such as the angular disparity and the postures of both agents) and contain ablation studies that require variations in the parameter values. Taken together, this requires many trials to achieve statistical significance, which is hard to achieve on the physical robot. Secondly, the simulation is repeatable and replicable. Given the parameter values and the random seed that was used, the same results can be obtained repeatedly.

6.3.1 *Experimental Setup*

As shown in Figure 6.2, the two robots are placed in a scenario where two objects are positioned on a table top. The task of one of the robots (self-agent, blue) is to mentally adopt the perspective of the other static robot

(target agent, gray) and to decide whether an object is to the left or right of the target agent. The left/right judgments are made based on the angle to the object, akin to Equation (3.11). A typical setup from experiments with humans (Kessler and Thomson, 2010) is replicated as closely as possible so that comparisons can be drawn. That is, the perspective difference between the two robots is varied, and so are the body postures of the robots. The perspective difference (angular disparity) is varied according to the human data that the model is compared with, and is detailed within each section. The response time serves as the primary evaluation metric.

6.3.2 *Parameter Choices*

The parameter values were experimentally determined as follows.

6.3.2.1 *Mixing Parameter*

The mixing parameter ω (defining whether each joint is matched individually or only the final gaze direction is considered, see Section 6.2.4) was drawn from two random distributions depending on the number of male and female subjects in the respective experiment: $\omega_{\text{males}} \sim \mathcal{N}(0.95, 0.029)$ for males and $\omega_{\text{females}} \sim \mathcal{N}(0.91, 0.065)$ for females[2]. These distributions are chosen based on comparisons with human response times and are further discussed and justified in Section 6.4.4.

6.3.2.2 *Speed-Accuracy Trade-Off*

The parameter ϵ governs a speed-accuracy trade-off. The larger ϵ, the larger the remaining distance between the two agents and the earlier the mental simulation is stopped, but the higher the chance of an incorrect response.

Figure 6.3 visualizes the response times and accuracies with ϵ as the independent variable. The data originates from 24 model instances (corresponding to 24 subjects in Michelon and Zacks, 2006), and each of the model instances solves PT tasks with angular disparities $\theta \in \{45, 90, 135, 180\}$, whereby four trials were executed per angular disparity.

In the following experiments, ϵ was set such that the model's error rate is below 1.0% ($\epsilon = 1.5$), which is in line with human data where error rates are between 1.0% and 5.0% (see e. g. Kessler and Thomson, 2010; Michelon and Zacks, 2006).

2 A threshold was put in place such that $0 \leqslant \omega \leqslant 1$.

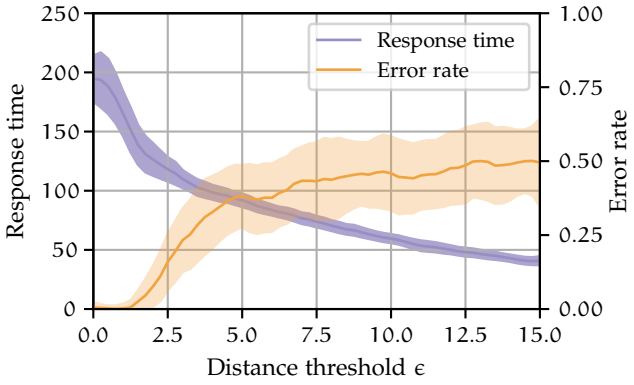


Figure 6.3: This figure depicts the speed-accuracy trade-off. The response times and error rates are shown with ϵ as the independent variable, whereby ϵ is the termination criterion as described in Section 6.2.3. While larger ϵ reduce the response time, the error rate increases and approaches the level of chance (0.5). The response time is the mean response time of all responses where the angular disparity was larger than zero (as for zero degrees angular disparity no mental rotation is necessary, and thus the accuracy approaches 1) and the shaded area depicts the standard deviation for the 24 model instances.

6.3.2.3 *Size of Attentional Set*

Figure 6.4 shows how the mean response time varies depending on the number of models contained within $\mathbf{A}$. There were 24 model instances $\forall |\mathbf{A}| \in \{1, \ldots, 18\}$. The angular disparities were varied between 0 and 180 degrees in 45-degree steps, and four trials were executed per angular disparity. The figure also shows the response time when the attentional component was disabled (depicted as $|\mathbf{A}| = 0$).

One can observe that there is a minimum for $|\mathbf{A}| = 4$ with a mean response time of $\overline{C}_4 = 118.0$. The mean response times for three and five attentional models are very similar ($\overline{C}_3 = 119.6$ and $\overline{C}_5 = 119.4$ respectively), while the response times are increasing for $|\mathbf{A}| < 3$ and $|\mathbf{A}| > 5$. Without using the attentional component, the mean response time is $\overline{C}_0 = 197.8$, and this response time is also approached when all action primitives are contained in the attentional set ($|\mathbf{U}| = |\mathbf{A}| = 18$), which is equivalent to not using the attentional component ($\overline{C}_{18} \approx \overline{C}_0$).

All further results in this chapter are reported for $|\mathbf{A}| = 4$ attentional components, except when it is explicitly stated that the attentional component was disabled.

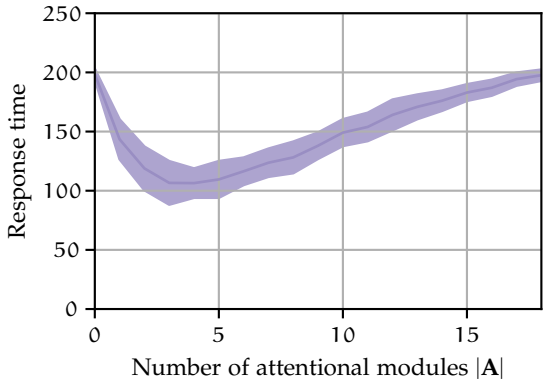


Figure 6.4: This figure shows the mean response time depending on the size of the attentional set **A**. The minimum mean response time is achieved for $|\mathbf{A}| = 4$, while the response time is increasing for less or more than 4 attentional modules. The shaded area depicts the standard deviation for the 24 model instances.

6.4 A MODEL OF HUMAN PERSPECTIVE TAKING MECHANISMS

This section discusses the proposed computational model as a model for human level two perspective taking (PT2) mechanisms. Humans are targeted as many studies suggest that other primates are not capable of PT2 (see e. g. Anderson et al., 1996; Karg et al., 2016; Kummer et al., 1996). The model suggests that humans simulate physical movements when taking the perspective of others.

The model is validated by comparing the response time profiles obtained using the model with those from humans in psychology studies. There, the human subjects are typically presented with stimuli on a photograph (Michelon and Zacks, 2006), screen (Kessler and Rutherford, 2010; Kessler and Thomson, 2010; Kessler and Wang, 2012) or virtual reality environment (Deroualle et al., 2015; Kockler et al., 2010) where another human (or avatar) is also sitting at a table. Two or more objects are located on the table, and the subject's task is to respond as quickly and accurately as possible whether the target object is to the other human's left or right.

Note that there are significant variations in the response times of humans depending on the specific setup (see e. g. Kessler and Rutherford, 2010, where the impact of the response modality is investigated). Also, while the model's response time only takes the time of the mental simulation into account (the response time equals the processing time), the human's response time includes the time to perceive the stimulus, the mental simulation, the decision time, the motor execution time and so forth. Therefore, the follow-

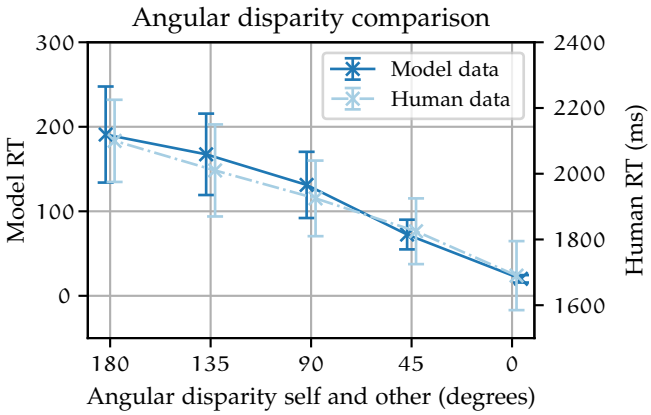


Figure 6.5: The response times of the computational model (dark, solid lines) are compared to experimental data in humans (bright, dashed lines, data from Michelon and Zacks, 2006). For both human and model data is was found that there is a linear relationship between angular disparity and response times.

ing comparisons focus on the response time profiles rather than comparing the absolute values.

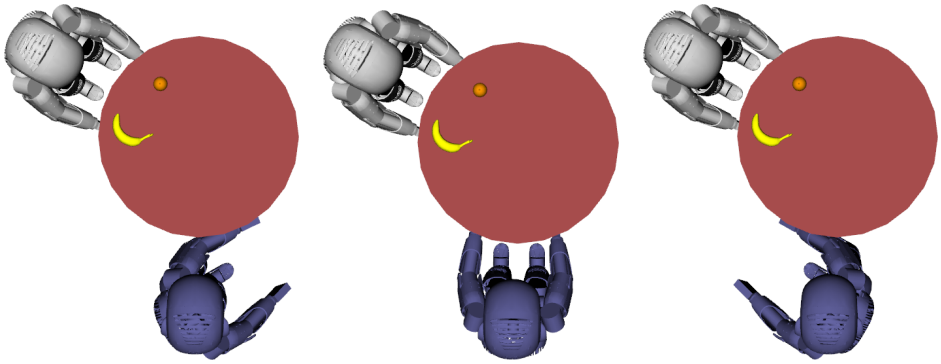
Furthermore, there are substantial differences across individuals in humans. Most psychological studies report the standard error rather than the standard deviation[3]. The following figures show the standard error for human data while showing the standard deviation for model data. This decision was made as the model's standard error is typically close to zero, while still showing considerate variations from the mean. The model's response times have been scaled so that model and human data can be easily compared.

All experiments were replicated so that the number of model instances matches the number of subjects in the experiments with humans (details are provided within the respective sections). For each model instance, the mixing parameter ω was randomly drawn from a Gaussian distribution as further described in Section 6.4.4. For each trial, the models that are initially contained in the attentional set is randomly chosen.

6.4.1 *Response Time Variation with Angular Disparity*

The response times of humans in PT tasks grow linearly with the angular disparity between the self and the target agent (Kessler and Thomson, 2010;

3 The standard error measures how far the sample mean of the data is from the population mean, while the standard deviation measures the variability of the sample data from the sample mean.



(a) Congruent movement (b) Straight (c) Incongruent movement

Figure 6.6: Movement congruency schematic. The required mental rotation is clockwise for all examples. In (a), the self-agent's (blue robot) torso is also rotated clockwise, which is considered as congruent. This is opposed to (c), where the self-agent's torso is rotated counterclockwise and hence the movement direction and torso rotation are incongruent. As shown in (b), a straight body posture is used as baseline.

Michelon and Zacks, 2006; Surtees et al., 2013*b*). Michelon and Zacks (2006) were the first to show this effect by varying the angular disparity between the self and other between 0 degrees and 180 degrees in 45-degree steps. The first comparison, shown in Figure 6.5, validates that the model response time also grows linearly with the angular disparity (t-test, $p < 0.01$) and that there is a good qualitative agreement between model and human data for all angular disparities $(\theta \in \{0, 45, 90, 135, 180\})$.

There is an interesting observation to make: the model's standard deviation (across the 24 model instances with 4 trials per angular disparity per model) increases with the angular disparity. This has various reasons, with the predominant reason being the random choice of the models contained in the attentional set. Indeed, the standard deviation is not significantly different for different angular disparities if the attentional component is not used.

6.4.2 *Movement Congruence*

The following sections investigate whether the model also matches human data when introducing several experimental variations. The first variation follows Experiment 1 in Kessler and Thomson (2010) and is concerned with the movement congruence. In human data (Figure 6.7, right), body postures that are congruent with the required movement direction (torso pose and movement direction are both clockwise or both counterclockwise; see Fig-

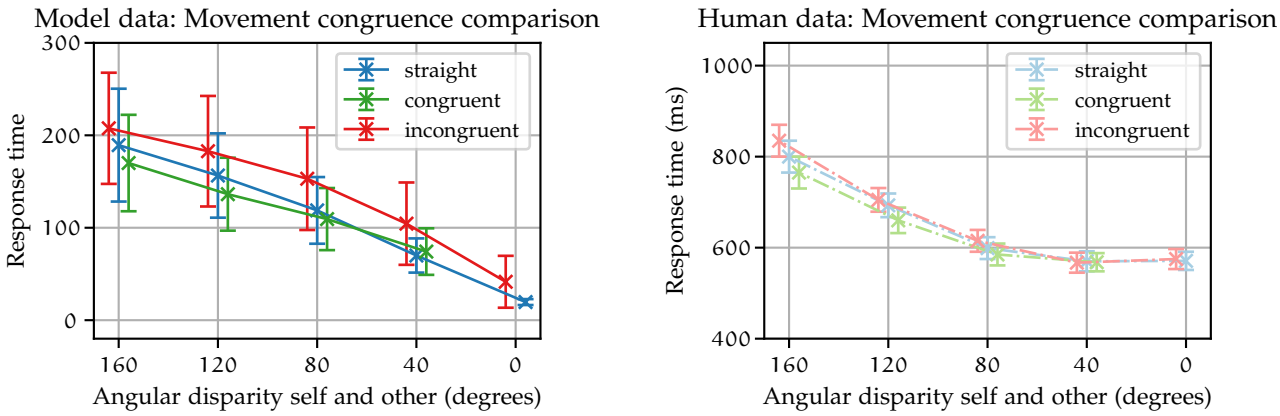


Figure 6.7: The response times with respect to movement congruence of the computational model (left) are compared to experimental data in humans (right, data from Kessler and Thomson, 2010, Experiment 1). In both the model and human data, the movements where the self-agent's posture is aligned with the movement direction lead to faster response times than movements where the self-agent starts with a straight posture, and incongruent movements are the slowest. These differences are more pronounced for larger angular disparities in both human and model data. Note that for 0 degrees angular disparity (i. e. judging the egocentric perspective), there is no congruent case. As the target agent's torso posture is straight, a torso rotation of the self-agent to either side leads to an incongruent movement (arguably, the straight body posture could be considered congruent).

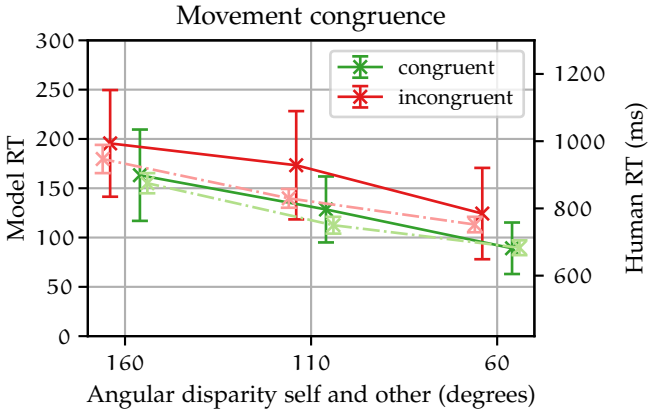


Figure 6.8: Comparing movement congruence as in Figure 6.7, but using human data from another study (Kessler and Rutherford, 2010, Experiment 1). Dark, solid lines correspond to model data, whereas the bright, dashed lines indicate human data.

ure 6.6 for a schematic) lower the response time compared to straight body postures, while incongruent postures result in increased response times. Furthermore, the differences between congruent and incongruent body postures are more pronounced for larger angular disparities.

Figure 6.7 (left) shows the model's response times[4]. As for the human data, there are significant differences between straight, congruent and incongruent postures (t-test, $p < 0.01$). However, the t-test was significant even for 0 degrees, which is contrary to the data reported by Kessler and Thomson (2010).

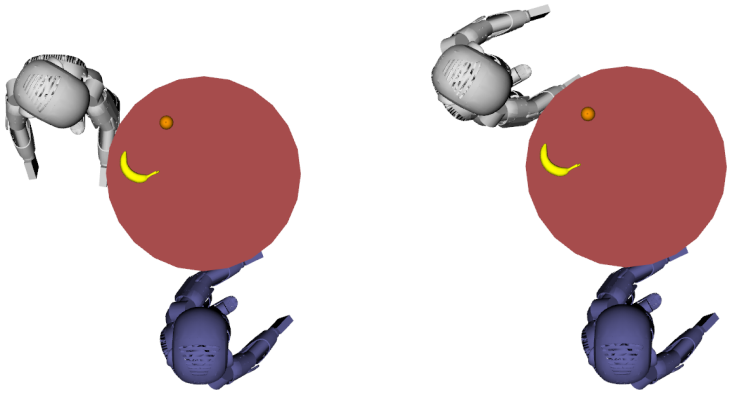
There is another notable difference for the 0-degree disparity case: while the model data suggests that the response time should be lower compared to 40 degrees angular disparity, the human data indicate that the response times for 0 and 40 degrees do not significantly differ. As discussed in Section 2.5.3, Janczyk (2013) argue that this might be due to the usage of a visual matching strategy for small angular disparities. Also note that the response times for 0 and 45 degrees differ significantly in the human data collected by Michelon and Zacks (2006, see Figure 6.5). Furthermore, the human experimental data contained in the next sections do not contain data for 0 degrees, such that this effect currently cannot be examined further.

Figure 6.8 shows another comparison with other subjects and a slightly different experimental setup (with different angular disparities, following the experimental setup in Kessler and Rutherford, 2010, Experiment 1). There is a good qualitative match between human and model data for congruent movements, while for incongruent movements there is a slight discrepancy for 110 degrees angular disparity.

6.4.3 *Posture Congruence*

Kessler and Thomson (2010) also investigate the impact of the target agent's body posture (Experiment 4 in Kessler and Thomson, 2010). While in the previous section movement congruence indicated whether the self-agent's body posture was congruent with the movement direction towards the target agent, posture congruence in this section indicates whether the self-agent's body posture and the target agent's body posture match (see Figure 6.9). In other words, if the self-agent's torso is rotated clockwise and the target agent's torso is also rotated clockwise, then the posture is congruent

4 To replicate the experimental setup in Kessler and Thomson, 2010 (Experiment 1), 24 model instances were created. There were 12 trials per angular disparity $(\theta \in \{0, 40, 80, 120, 160\})$ for each model instance.



(a) Congruent posture (b) Incongruent posture

Figure 6.9: Posture congruency schematic. In (a), both the self-agent's (blue robot) and target agent's (gray robot) torso are rotated clockwise, which is considered as congruent posture. This is opposed to (b), where the self-agent's torso is rotated clockwise while the target agent's torso is rotated counterclockwise (incongruent posture).

(and similarly for counterclockwise rotations). In human data, the impact is smaller compared to that of the movement congruence, and there is only a small effect for 120 and 160 degrees angular disparity (but not for 40 and 80 degrees).

In the computational model, the mixing parameter ω controls the impact of the target agent's posture on the response times. The smaller ω, the higher the impact as each joint state is individually matched. On the other hand, a large ω leads to a small impact as the combination of all states is matched. For this experiment, the mixing parameter was randomly drawn from $\omega_{\text{females}} \sim \mathcal{N}(0.91, 0.065)$ for 12 model instances (corresponding to the 12 female subjects in Kessler and Thomson, 2010) and $\omega_{\text{males}} \sim \mathcal{N}(0.95, 0.029)$ for the remaining 12 instances (corresponding to the 12 male subjects in Kessler and Thomson, 2010). The choice of the mean and variance for these distributions is justified in the next section. There were 16 trials per angular disparity $\left(\theta \in \{40, 80, 120, 160\}\right)$ for each model instance.

Figure 6.10 compares the model and human response times for the body posture congruence. As there is a good qualitative match for the response times, the model suggests that the cost function for the angular disparity is heavily biased towards d_Σ in humans, i. e. ω is close to 1. In other words, the model indicates that joint states are not matched individually, but instead as a combination of all states. The next section discusses this indication in more detail.

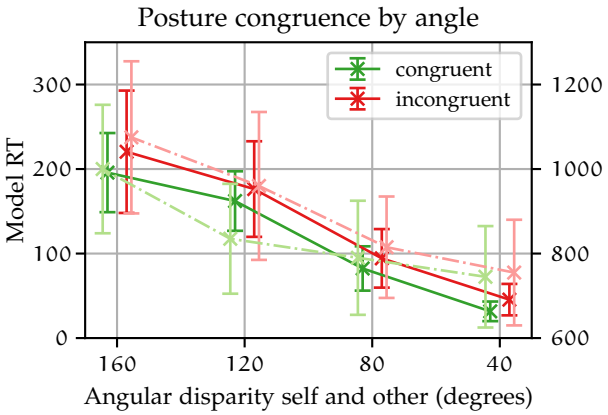


Figure 6.10: The response times with respect to posture congruence of the computational model (dark, solid lines) are compared to experimental data in humans (bright, dashed lines, data from Kessler and Thomson, 2010, Experiment 4). There is a good qualitative match of the model and human data for both congruent as well as incongruent postures.

6.4.4 *Differences by Sex and Social Skills*

Kessler and Wang (2012) have investigated whether there are differences in the perspective taking strategy employed by males and females, and depending on their social skills score. They suggest that there are different groups of perspective takers: "embodiers" (typically female, highly socially skilled individuals) that employ a mental rotation strategy and "systemizers" (typically male individuals with low social skills) who use alternative strategies.

6.4.4.1 *Embodiment Measure*

Kessler and Wang (2012) introduced the embodiment measure E to analyze the perspective taking strategy that was employed by different individuals and groups. They propose that E measures the "proportion of the body schema that is mentally transformed". Within this thesis, the measure is used to compare model and human responses.

The measure E is defined as the (average) difference of the z-scores for incongruent ($M = 0$) and congruent ($M = 1$) movements at $\theta = 120$ degrees and $\theta = 160$ degrees of angular disparity[5]:

$$E = \frac{(z_{120,0} - z_{120,1}) + (z_{160,0} - z_{160,1})}{2}. \tag{6.6}$$

5 Samples with $\theta = 40$ and $\theta = 80$ are not used as a visual matching strategy might have been employed at these small angles.

The z-score measures how many standard deviations σ_C a response time $C_{\theta,M}$ is from the overall mean μ_C, and is defined as[6]:

$$z_{\theta,M} = \frac{C_{\theta,M} - \mu_C}{\sigma_C}. \tag{6.7}$$

In the study conducted by Kessler and Wang (2012), most subjects (81 out of 96) had a positive embodiment measure. That is, response times of congruent movements were faster than these of incongruent movements. Kessler and Wang (2012) have shown that there is a significant relationship between sex and embodiment as well as a social skills score and embodiment.

6.4.4.2 *Link Between Mixing Parameter and Embodiment Measure*

From the computational model's perspective, the hypothesis is that the mixing parameter ω has a negative correlation with the embodiment measure. This directly follows the definition in Equation (6.3), where a higher ω leads to a lower proportion of the body schema that is mentally transformed and vice versa. The hypothesis was tested as follows. The model was instantiated 96 times (as many model instantiations as subjects in Kessler and Wang, 2012), with $\omega_{\text{females}} \sim \mathcal{N}(0.91, 0.065)$ for 51 instances and $\omega_{\text{males}} \sim \mathcal{N}(0.95, 0.029)$ for the remaining 45 instances. The means and standard deviations for the Gaussian distributions were chosen so to closely follow the human data as described in Section 6.4.4.3. In Figure 6.11 it is shown that there is a strong negative correlation between ω and the embodiment measure E ($R^2 = 0.697$).

6.4.4.3 *Comparison of Embodiment Measure for Human and Model Data*

The parameters of the normal distribution for the mixing parameter ω were chosen so that the embodiment scores E in humans are resembled as closely as possible. This is achieved qualitatively by choosing the mean and standard deviation so that model and human instances have similar embodiment scores as shown in Figure 6.12. This resulted in $\omega_{\text{females}} \sim \mathcal{N}(0.91, 0.065)$ for 51 instances and $\omega_{\text{males}} \sim \mathcal{N}(0.95, 0.029)$ for the remaining 45 instances.

It was then validated whether the embodiment measures of the human and model data were drawn from different distributions using two-sided Kolmogorov-Smirnov tests. The validation was performed for three groups: females, males, and the combined data. The statistical tests have shown that there are differences between human and model data for males ($p = .048$) and the combined group ($p = .01$), but not for females ($p = .089$).

6 Following Kessler and Wang (2012), all angles (including $\theta = 40$ and $\theta = 80$) are used to calculate the overall mean μ_C and standard deviation σ_C.

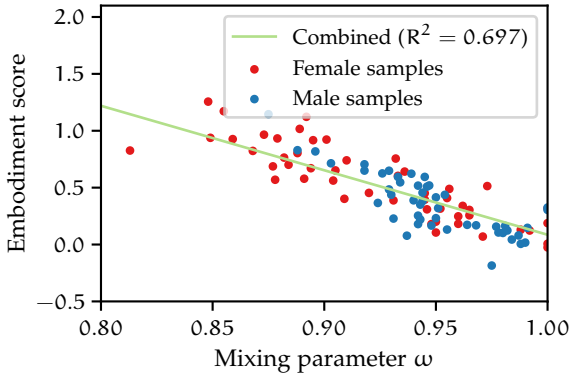


Figure 6.11: This figure visualizes the negative correlation between the mixing parameter ω and the corresponding embodiment measure E in the computational model. As detailed in the main text, the "female" and "male" samples have different underlying distributions. The green line indicates the linear regression line.

However, as discussed by Kessler and Wang (2012), 15 out of the 96 human subjects (most of them male) might not have employed a self-rotation strategy, i. e. their embodiment score is zero or negative. As all instances of the computational model implement a self-rotation strategy, another validation was performed where human subjects that used alternative strategies were excluded. In other words, the model data was compared to the 81 human subjects that employed a self-rotation strategy ($E \geqslant 0$) rather than comparing to all human subjects. Then, the null hypothesis that human and model samples are drawn from the same continuous distribution cannot be rejected ($p = .484$ for female data, $p = .921$ for male data, and $p = .374$ for combined data).

The statistical evaluations whether human and model samples are drawn from the same distribution are interesting for multiple reasons. Firstly, they suggest that some human subjects (with near-zero or sub-zero embodiment scores) indeed employed a strategy that does not rely on self-rotations. Secondly, the results suggest that the embodiment measures of human subjects with a positive embodiment score and the embodiment measures of the model instances might be drawn from the same distribution, which would indicate that these human subjects have used an embodied transformation strategy. Thirdly, it seems that males and females indeed differ in the way they take perspectives of other people. Although the differences in the distributions are small, some females seem to align themselves more thoroughly compared to males.

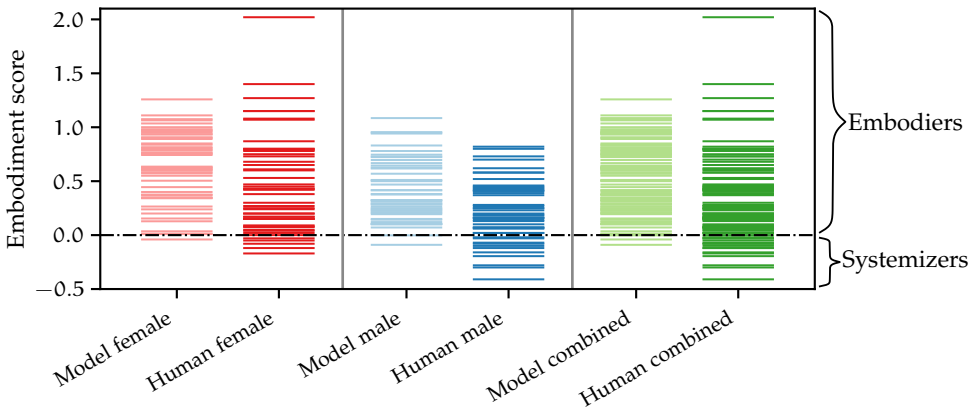


Figure 6.12: This figure qualitatively compares model and human embodiment measures in three groups: males, females and combined. One observation is that while there are some model instances with a near-zero or sub-zero embodiment score (the horizontal black line indicates the null-line), there are significantly more human subjects with such scores (so called "systemizers" that use an alternative strategy rather than a mental rotation). Also, there is one female human subject with a very high embodiment score of 2.02, while there do not seem to be any outliers contained in the model data.

6.4.4.4 *Importance of Attentional Component*

The results obtained from the computational model also provide strong support that the attentional component is crucial to obtain results that are comparable to humans. The embodiment measures significantly differ when the attentional component is not used ($p < .001$). This is because there is only a slight correlation between the mixing parameter ω and the embodiment measure E in that case. The smallest embodiment measure that was observed is $E = 0.45$ for $\omega = 0.993$, which is significantly higher compared to the embodiment measures in humans[7].

More generally, this shows that while the response times for congruent and incongruent movements differ even if the attentional component is not used, the differences between congruent and incongruent response times of the model is then dissimilar to the differences observed in humans. This finding suggests that humans employ an attentional mechanism similar to that formalized in the computational model.

7 As in Sections 6.4.4.2 and 6.4.4.3, 96 model instances were created and the experimental setup was the same as in these sections.

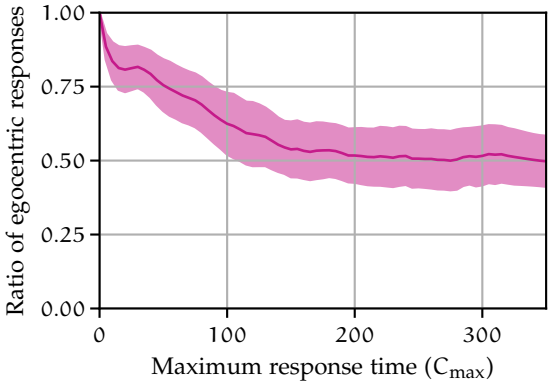


Figure 6.13: This figure shows ratio of egocentric responses when forced early responses are used (stopping the perspective taking process using C_{max}).

6.5 MODEL PREDICTIONS

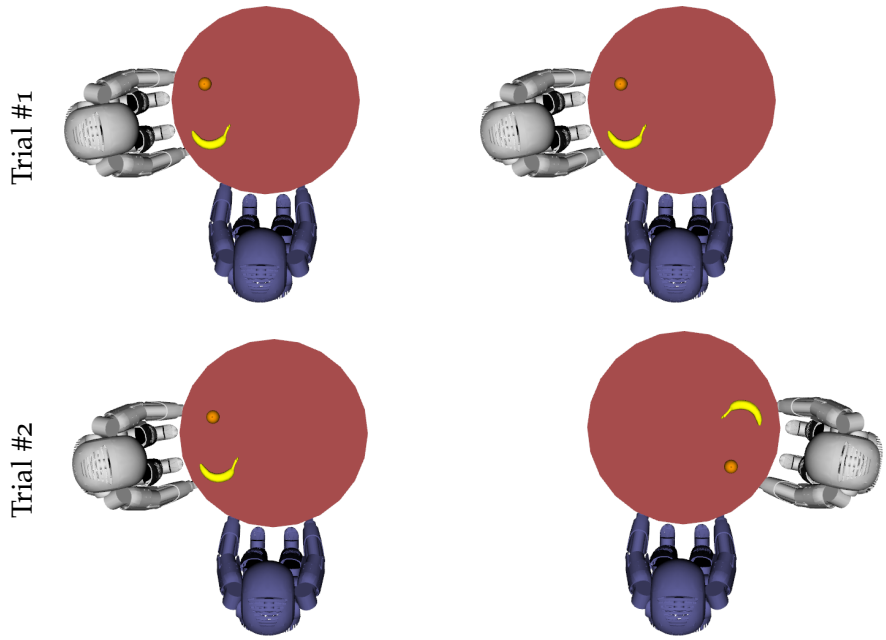
The model offers the following testable predictions. To the best of my knowledge, no studies have investigated the following aspects or suggested these predictions.

6.5.1 *Forced Early Response Leads to Egocentric Bias*

The first model prediction is concerned with imposing time pressure on model responses. While the model typically responds as soon as the distance to the target agent is below the threshold ϵ, in these experiments the response is forced once the response time hits an upper threshold C_{max}[8]. Remember that the response time C represents the number of forward passes and corresponds to the response time of the model (see Section 6.2.5). Therefore, the mental simulation process might be interrupted before the embodied transformation process is completed, i. e. before the distance between the self-agent and target agent drops below the distance threshold ϵ (see Section 6.2.3).

As the embodied transformation account hypothesizes that the self-agent mentally translates and rotates into the other's point of view, not giving any time for the mental rotation process should lead to a response which is fully compatible with the self-agent's own perspective (egocentric response). This is shown in Figure 6.13, where $C_{max} = 0$ leads to 100% egocentric responses. Then, with increasing C_{max} the ratio of egocentric responses slowly reduces,

8 Besides varying the threshold C_{max}, the experimental setup used within this section is the same as in Section 6.4.1.



(a) Congruent trials (b) Incongruent trials

Figure 6.14: Trial congruency schematic (straight body postures of both agents). In (a), the required direction of the mental rotation for trials #1 and #2 remains the same (both clockwise, congruent trials), while the direction changes from clockwise to counterclockwise in (b) (incongruent trials).

up until $C_{max} \approx 200$ when it approaches chance level (i. e. the responses do not depend on the egocentric view but instead only on the perspective of the target agent). Note that $C_{max} = 200$ coincides roughly with the mean response time of the model in the same setup for 180-degrees angular disparity (see Figure 6.5).

6.5.2 *Habituation Effects*

The second model prediction investigates whether there are habituation effects that should be taken into consideration when designing and evaluating experiments with humans. For the following experiments, the model is set up such that consecutive trials are dependent in the sense that the attentional set **A** remains the same across trials. In other words, the modules that are contained in the attentional set at the end of the previous trial are matching those of the attentional set at the beginning of the current trial. This is compared to a baseline where the trails are independent.

6.5.2.1 *Straight Body Postures*

The first set of experiments assumes straight postures of both the self and target agents. Then, the required mental movement directions of the previous and current trials are altered. Congruent trials are those where the required movement directions match (i. e. clockwise follows clockwise, or counterclockwise follows counterclockwise), and incongruent trials are those where clockwise follows counterclockwise or counterclockwise follows clockwise (see Figure 6.14)[9].

A Mann-Whitney U test is performed with the null hypothesis that the distributions of congruent and incongruent trials are equal (the response times are equal), and the alternative hypothesis that congruent trails have lower response times than incongruent trials (one-sided hypothesis). The null hypothesis can be rejected when comparing congruent and incongruent trials ($p = .006$). This result suggests that two consecutive trials that require the same movement direction are faster compared to trials that require different movement directions. As expected, the null hypothesis cannot be rejected for the baseline comparison ($p = .58$).

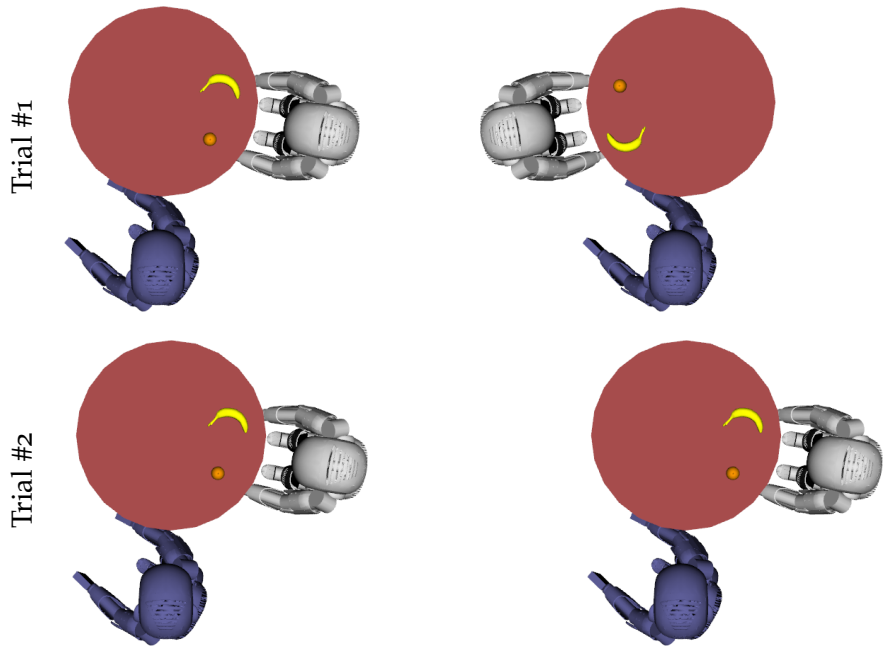
6.5.2.2 *Congruent and Incongruent Body Postures*

The second set of experiments investigates whether these results can be extended to trials where the self-agent's body posture is not straight. Now, there are two types of congruencies: movement congruency (the self-agent's body posture is compatible with the required movement direction) and trial congruency (the previous trial's movement direction is compatible with the current trial's movement direction, as above). Importantly, however, the self-agent's posture is not changed across two trials as the physical movement would change the modules in the attentional set.

Therefore, two tests can be performed. The first test compares two trials, one with compatible trial congruency and another incompatible one. For the first test, both trials have congruent movements (see Figure 6.15 for a schematic). As above, a Mann-Whitney U test with a one-sided null hypothesis is performed. The null hypothesis that both trials have equal response times can be rejected ($p < .001$). This is not the case for the baseline comparison ($p = .51$).

The second test is very similar to the first one, with the difference that both trials have incongruent movements (rather than congruent movements as above; see Figure 6.16 for a schematic). The null hypothesis of equal response

9 For this prediction, 24 models were instantiated. The angular disparity was set to ± 80 degrees, and 12 trials were performed per angular disparity and model instance.



(a) Congruent trials (b) Incongruent trials

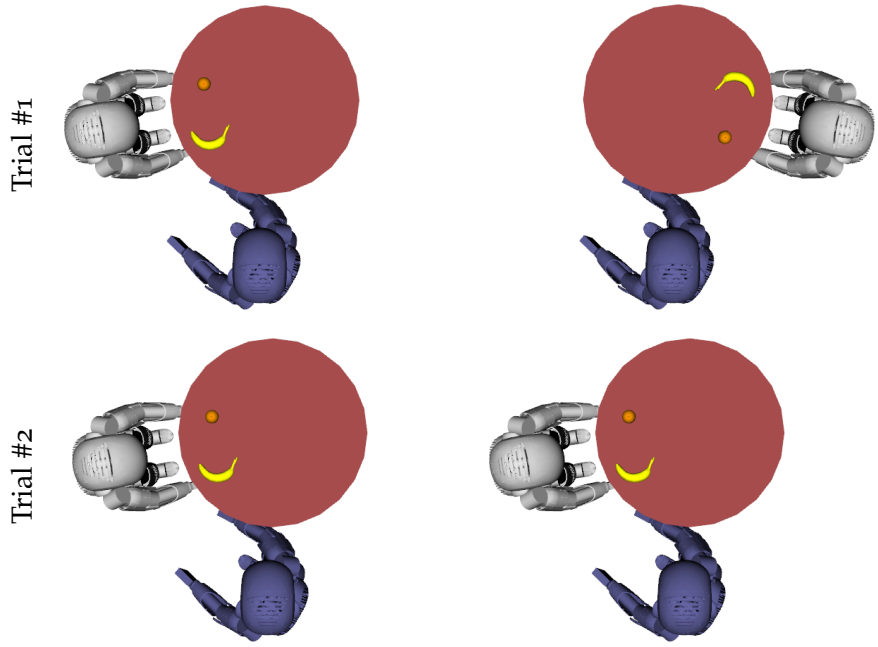
Figure 6.15: Trial congruency schematic (congruent movement direction). While the required mental rotation and the torso rotation are counterclockwise for trial #2 in both (a) and (b), the required movement direction in the previous trial (trial #1) is congruent with trial #2 in (a) (congruent trials) but not in (b) (incongruent trials).

times can be rejected ($p < .001$), while this is not the case for the baseline comparison ($p = .33$).

One can further compare the baseline trials with the trials where the attentional modules remain across trials. There are statistically significant differences for congruent trials, both for congruent movements and incongruent movements ($p < .001$ for both). However, there are no statistical differences for incongruent trials, neither for congruent movements nor incongruent movements ($p = .25$ and $p = .82$ respectively).

6.5.2.3 *Implications for Experimental Design*

The results presented in this section hypothesize that there are habituation effects across trials. The habituation effects occur when the required movement direction of two consecutive trials remains the same (Section 6.5.2.1), regardless of the movement congruency (Section 6.5.2.2). Therefore, the human trials should be pseudo-randomized with respect to the trial congruence, and statistical evaluation should take the trial congruence into account.



(a) Congruent trials (b) Incongruent trials

Figure 6.16: Trial congruency schematic (incongruent movement direction). In both (a) and (b), the required mental rotation (clockwise) and the torso rotation (counterclockwise) are incongruent for trial #2. However, the required movement direction in the previous trial (trial #1) is congruent with trial #2 in (a) (congruent trials) but not in (b) (incongruent trials).

6.6 CONCLUSIONS

This chapter presented a computational model for PT that contains a set of action primitives that are passed through a forward model as building blocks. An attentional component that introduces competition between multiple action primitives was employed to reduce the model's response time. It was shown that the model's response time is similar to those of the human visual system only if this attentional component is employed. It was therefore argued that humans implement an attentional mechanism similar to that of the proposed model.

The model also proposes the following testable predictions. The model suggests that there should be a bias towards the egocentric perspective for early forced responses, and a habituation effect with respect to the mental movement direction of the previous stimulus.

This chapter has investigated a wide breadth of experiments that would not be replicable on the physical iCub robot. However, future works on the physical robot could shed light on complementary questions to those addressed in this chapter. For example, one could investigate how the noise

contained in the estimates for the pose of the robot, the state of the human, and the positions of the object will impact the results.

A rough estimate of the impact can be obtained using the simulated results. As discussed in Section 6.3.2.2, the distance threshold ϵ was set to 1.5 to achieve an error rate below 1%. Let us further assume that the primary source of noise is the perception of the human's state (SLAM and motor encoders can be used to obtain accurate estimates of the robot's state, and the object positions can be accurately obtained using RGB-D cameras). As $\omega \approx 1$, the gaze direction of the human is the main component of the distance between self-agent and target agent, and the distance originating from comparisons of individual joints is negligible (see Section 6.2.4). Then, $\epsilon = 1.5$ is equivalent to a gaze estimation error of 15 degrees, which is well above the errors of the gaze estimation method presented in Chapter 5. In summary, these estimates suggest that the model's validity is not constrained to the simulated setup presented in this chapter, but also applies to physical robots.

Further future work will discuss the developmental aspects of PT by learning the forward model and will show that an accurate forward model is needed for PT. Thus, Section 7.2.4 will discuss the suggestion that children are typically ego-centric as their developing forward models are only sufficiently accurate to overcome small perspective differences.

Section 7.2.5 will discuss whether there is an embodied process that accounts for the often observed heuristic of swapping left and right if the target agent's perspective is directly opposite the self-agents perspective (180 degrees angular disparity).

7

CONCLUSIONS AND FUTURE WORK

The purpose of this chapter is threefold. It first summarizes the contributions that were presented in this thesis, followed by a discussion of the limitations of the work. Finally, this chapter presents work in progress and future research directions that might emerge from this thesis.

7.1 OVERVIEW AND CONTRIBUTIONS OF THE THESIS

The main contribution of this thesis is the study of perspective taking using a mixed forward/reverse engineering approach, which led to an artificial visual system that is implementable on a robotic system and a computational model to study the human visual system. The thesis has thus addressed and advanced research in various domains.

First, the perceptional components required to endow a robot with perspective taking abilities in markerless environments were detailed and implemented on the iCub humanoid robot. It was shown that it is of advantage to have separate mechanisms implementing the two levels of perspective taking – fast line-of-sight tracing for level one perspective taking and another more elaborate mental rotation process for level two perspective taking. One limitation of this initial approach was the approximation of the gaze direction of the human with their head pose, which was argued is not suitable for human-robot interactions (HRIs).

To address this issue, a novel architecture for gaze estimation with large camera-subject distances as commonly encountered in HRIs was designed. It was shown that a major factor contributing to the weak performance of previous gaze estimators in HRIs is the lack of a large, labeled dataset in these scenarios, which is due to the difficulty of obtaining ground truth annotations. Therefore, the dataset collection was tackled from a new perspective. Eyetracking glasses were used to obtain the ground truth gaze direction. Semantic image inpainting was subsequently applied to overcome the appearance alteration caused by the eyetracking glasses. It was then shown that this generalizes well to other scenarios such as laptop viewing.

Moreover, the forward engineering approach described above is just one way of studying the perspective taking ability. The final contribution of this

thesis was a reverse engineering approach that investigated the embodied transformation account that stems from the psychology area. This was a difficult challenge because the level of analysis within psychology is of a different focus, and the terminology differs considerably. The work reported here advocates that perspective taking is governed by a competition process for visual attention. It was subsequently argued that humans implement an attentional mechanism similar to that of the proposed model. This offers a new perspective on the data obtained from behavioral experiments. The model provides the following testable predictions. Firstly, it predicts a habituation effect between different trials that depends on the congruence of two subsequent trials. Secondly, it suggests that a forced early response leads to a bias towards the own perspective.

These contributions are not only theoretical but can be readily applied in practical settings as the work emerging from this thesis has been made available in open-source developer kits to other researchers and the general public. I hope this opens up future directions on perspective taking in various research areas, including robotics, computer vision, computational cognition, and computational neuroscience. The research itself is developed using multiple open-source libraries, and the work in this thesis extended many of them as highlighted in Appendix A.

7.2 LIMITATIONS

The research in this thesis is only a small step on a long journey. This section presents the limitations of the research, and the following section discusses future avenues which might be taken following the presented work in the thesis.

7.2.1 *Applying Computer Vision Methods to Robotics*

The presented work lies at the intersection between computer vision and robotics, and several state-of-the-art algorithms for the visual perception of the world surrounding the robot are used and proposed. However, it is well known that applying computer vision algorithms to robots is challenging. This is due to different hardware, diverging datasets and real-time constraints to name a few reasons (see Sünderhauf et al., 2018, for more insights).

The proposed artificial visual system for perspective taking (PT) in HRI that was presented in Chapters 3 and 5 is no different, and also suffers from

some limitations. For example, the object recognition pipeline currently only performs well on uniformly colored backgrounds and fails to segment the objects properly when the background has varying texture.

Moreover, humans are perceived using an additional RGB-D camera that is mounted on the robot's mobile base. It would be desirable to omit this camera, which would lead to a system that only uses the sensors of the humanoid robot (as opposed to the "augmented" robot with sensors mounted on the mobile base). However, the robot's cameras only allow to either observe the object locations or the human's gaze at any one time. As PT requires the observation of object locations and human's gaze, omitting the RGB-D camera would require a strategy to attend these locations and integrate the acquired information (for example by utilizing a working memory). The decision whether to use additional sensors is a typical trade-off on whether one is looking for a close link to the human mechanisms (where no additional sensors outside of the humanoid robot should be used) or increased accuracy and ease of implementation.

7.2.2 *Object Models*

Another limitation is that the proposed system is currently not capable of modeling individual objects in 3D. More specifically, the object model only contains the surface that is perceivable from the robot's current point of view. Therefore, when mentally rotating the environment, so that it is aligned with the human's perspective, the visual models of the objects are incomplete as the surface that is perceived by the human is not contained within the model (unless the object is flat). This limits the applicability of the system for visual level two perspective taking but does not impact level one perspective taking or spatial level two perspective taking.

This limitation could be overcome by allowing the robot to manipulate objects so that they are perceived from multiple viewpoints and fusing the perceptions to obtain dense and accurate 3D representations of the objects. Alternatively, the 3D representation could be obtained by changing the robot's location so that the object is perceived from multiple viewpoints, as recently proposed by Florence et al. (2018). One could also imagine applying recent methods that can reconstruct a 3D representation of objects given a single 2D image (see e. g. the methods of Yan et al., 2016 and Wu et al., 2016).

7.2.3 *Extreme Head Poses*

While the proposed gaze estimation method presented in Chapter 5 considerably enlarges the applicability of eye gaze estimation methods, it fails to estimate the gaze for extreme head poses. One could imagine the most extreme case where the subject does not even face the camera. In future work, the method could be extended so the gaze can be estimated even in these situations. This will require the integration of an appropriate face detection method, such as the one proposed by Marin-Jimenez et al. (2014). Initial steps were made towards integrating saliency information for gaze estimation when the eyes are not visible, similar to the proposed work by Recasens et al. (2015, see Section 2.3 for more details on this work). Another appealing way to pursue this research direction would be in enabling the gaze estimator to output a probability distribution, where extreme head poses would lead to a relatively flat distribution that corresponds to a high uncertainty about the precise gaze location.

7.2.4 *A Model of Child Development*

The computational model introduced in Chapter 6 also has some limitations. One of the assumptions is that the forward model is known. As we experience situations like those in the PT experiments that are investigated in Chapter 6 in our everyday lives, this is a reasonable assumption for adults. However, the motor system of infants and toddlers is relatively immature, and hence a known forward model cannot be assumed (Demiris and Meltzoff, 2008). Instead, learning the forward model using training samples acquired by issuing random movements (i. e. motor babbling) that change the robot's state would be analogous to the maturement of the child's motor system. The computational model could then be used to investigate the development of perspective taking skills subject to the development of the internal forward model.

7.2.5 *Shortcut to Reverse Left/Right Judgments*

Furthermore, the model does not capture the shortcut to reverse left/right judgments for the other's perspective if the other is directly opposite (see Gardner et al., 2013, and Section 2.5.3 for more details). While it seems straightforward to include such a shortcut, the question arises whether there is yet another component that switches between the shortcut strategy and the mental rotation strategy.

7.3 FUTURE DIRECTIONS

Some future directions that directly emerge from the shortcomings of the presented research were outlined above. This section describes some more general and broader research questions that are related to the presented research in this thesis and could be addressed in the future.

7.3.1 *Relating Perspective Taking and Autobiographical Memories*

While not presented in this thesis, I have recently investigated an autobiographical memory for robots that allows organization of multi-modal data so that it can be remembered, relived and augmented over time (Petit, Fischer and Demiris, 2016*a*; 2016*b*). It would be interesting to investigate PT in these autobiographical memories, which comes with several difficulties. For example, the state of the objects and other agents will be generally more uncertain compared to situations that happen in the present. Furthermore, there is yet another "agent" involved: the "past self" (besides the "current self"; Libby and Eibach, 2002), which may or may not use the same forward model as the "current self".

It also remains an open question whether the other's perspective is inferred while retrieving the memory, or whether the other's perspective is stored as memory along with the self-perspective[1]. The former would require additional computation but avoids information duplication, while the latter would imply that duplicated information is stored, but the other's perspective is readily available without requiring an embodied transformation. This representation might change over time, as typically older memories are more likely to be recalled from a third person perspective (Libby and Eibach, 2002).

The question which perspective is used to retrieve a memory is closely related to the concept of autonoetic consciousness, which studies the human's ability to mentally travel in time either into the past or future. Sutin and Robins (2008) argue that the employed perspective can change the feelings and thoughts of the person who is retrieving the memory, which ultimately impacts the way we evaluate the memory and whether we would act again in the same way.

1 Another alternative one could envision is a representation that is not tied to an agent but to an arbitrary reference point.

7.3.2 *Taking the Perspective of Arbitrary Agents*

This thesis endowed robots with the ability to take the perspective of humans[2], which relied on algorithms to estimate the head and eye poses of humans. However, the robots fail to take the perspective of agents that are not humans, as there is no knowledge of how these agents perceive the world. This is in contrast to the skills humans possess. It has been shown that humans can imagine the viewpoint as seen from even inanimate objects such as arrows or lamps (Schurz et al., 2015). Equipping robots with such skills would be interesting.

However, learning the associated "natural" viewpoint direction for each object class individually does not scale. Instead, a more compelling research avenue to explore is to find correspondences between the human body and the object of interest automatically. Indeed, I have participated in works that find correspondences on the abstraction level of kinematic structures (Chang, Fischer, Petit, Zambelli and Demiris, 2016, 2018). There, it has been shown that correspondence matches between humans and robots (or other objects such as lamps) can be found. Future works will investigate whether the similarity metrics between objects that are defined in these works can predict the response time differences of humans that are tasked to take the perspective of non-human agents. For example, one could investigate whether it is faster to take the perspective of the iCub robot (with a similar body structure to a human) or that of a lamp (which is dissimilar to a human body).

7.3.3 *Active Vision and Perspective Taking*

Humans are active agents, in the sense that they constantly move around and change their pose to extract more information from their environment depending on the task. Within the computer vision and robotics areas, manipulating the camera's viewpoint to achieve these goals is called "active vision" (see Bajcsy et al., 2018, for an overview and excellent review of recent works).

I believe that active vision and perspective taking are two related problems and wish to explore this relationship in detail. Specifically, one could argue that the emergence of PT requires not only forward model learning as discussed in the previous section but also an understanding of a means to

2 The problem discussed in this section also applies to Chapter 6 where the perspective of another robot is taken.

achieve a task-specific goal. Precisely this task understanding is also investigated in the active vision literature.

Another interpretation of active vision within PT is the following. Chapter 6 introduced a distance metric that is minimized, which contained the distance to the target agent. One could argue that this is a particular case of a more general cost function that captures the additional information being gathered when changing the viewpoint. This, in turn, is a prominent research topic within active vision.

7.4 EPILOGUE

Just like humans, robots should be able to understand the intentions of others and act accordingly. To make some steps towards this goal, this thesis has drawn inspiration from computer vision, robotics, computational modeling, and psychology. The thesis advanced the former areas directly and suggested promising research avenues in the latter. While the thesis aimed at a specific cognitive function, namely perspective taking, I believe that this interdisciplinary approach, where one discipline informs another, opens exciting new opportunities overall.

A

ROBOTS, COMPONENTS AND SENSORS

This thesis features several robots, components and sensors, which are described in this appendix for reference.

A.1 ROBOT OPERATING SYSTEM

Throughout this thesis, the Robot Operating System (ROS; Quigley et al., 2009) was used to communicate between different processes (within ROS called *nodes*) in a standardized manner. The advantage of ROS over other robotic middleware like Yet Another Robot Platform (see Appendix A.2) is an extensive collection of tools that allow rapid development of new applications, an example being the *RViz* 3D visualization tool. ROS is well integrated with most robots and sensors and allows for the fast integration of new hardware. Hence, it is the most widely used robotics middleware at the time of writing this thesis.

A.2 YET ANOTHER ROBOT PLATFORM

While ROS was used to develop most of the proposed algorithms, Yet Another Robot Platform (YARP; Fitzpatrick et al., 2006) was used to control the iCub humanoid robot (Appendix A.3). Similarly to ROS, YARP provides a collection of libraries for robotics, and a means to communicate between the different processes. YARP is mostly written in C++ but exposes bindings to many programming languages including Python.

YARP allows communication of processes that are written in ROS. Work carried out for the thesis contributed several additions to streamline this intercommunication, with the main addition being the ability to visualize YARP processes and their interaction with ROS processes using the standard ROS tools (namely *rqt_graph*). Also, a bug was resolved that prohibited resetting the iCub simulation in ROS.

A.3 ICUB HUMANOID ROBOT

Figure A.1 depicts the iCub humanoid robot that was used to conduct the experiments in this thesis. The iCub is equipped with multiple sensors: encoders in all its 53 joints, force/torque sensors, tactile sensors integrated into the artificial skin, and eye cameras (Metta et al., 2010). They allow for a coherent understanding of body configuration, motor capabilities and the environment as well as an ability to show facial expressions, which makes it an ideal platform for studies of human-robot interaction and cognition. The iCub uses YARP as underlying robotic middleware, such that the additions that were provided for YARP as discussed above are directly applicable when working with the iCub.



Figure A.1: iCub humanoid robot[1]

1 This figure was originally taken by Xavier Caré (https://commons.wikimedia.org/wiki/File:ICub_Innorobo_Lyon_2014_debout.JPG) and was modified to remove the background. The original and modified figures are available under the CC BY-SA 4.0 license.

A.4 PUPIL LABS EYETRACKER

The Pupil Labs eyetracker (Kassner et al., 2014) was used within the dataset collection framework described in Chapter 5 of this thesis. They have several advantages over other mobile eyetracking glasses, one being the relatively low price and another one being the open source software which allowed us to write an interface between the programming interface of the Pupil Labs glasses and ROS. Code was written to fix some issues within the Pupil Labs software, which is now part of the official code repository.

As described in Section 5.2.1 and shown in Figure 5.4, the hardware of the glasses was customized so that the eye cameras are at the same position for all subjects. This modification is credited to Joshua Elsdon.

A.5 RGB-D CAMERAS

Two different RGB-D cameras were used in the experiments of this thesis. For the work reported in Chapters 3 and 4, the Asus Xtion Pro was used (Figure A.2, left), which relies on structured light information to estimate the depth of objects. Color images are provided at 1280x1024 resolution and depth images at 320x240 resolution. The depth information is provided for distances between 0.8 to 3.5 meters.



Figure A.2: Asus Xtion Pro[2] (left) and Kinect v2[3] (right) RGB-D cameras

The Kinect v2 (Figure A.2, right) is a time of flight camera that is able to capture color images at 1920x1080 resolution and depth images at 512x424 resolution. It has a wide field of view (70 degrees horizontally and 60 degrees vertically) and can provide depth information for distances between 0.5 and

2 This figure is by Pierre Lecourt and is under the CC BY-NC-SA 2.0 license (originally from https://flic.kr/p/e52Lxq)

3 This figure is by Evan-Amos and is in the public domain (originally from https://commons.wikimedia.org/wiki/File:Xbox-One-Kinect.jpg)

4.5 meters. These improved specifications are the reason for using the Kinect v2 to collect the dataset as described in Chapter 5.

A.6 OPTITRACK MOTION CAPTURE SYSTEM

In Chapter 5, an OptiTrack motion capture system was used to record the ground truth pose of the eyetracking glasses worn by subjects (OptiTrack, 2018). The motion capture system consists of eight Flex 3 cameras, each capturing images at 640x480 resolution and a 100 frames per second sampling rate. It allows tracking of the pose of objects with very high accuracy.

B

INPAINTING METHODOLOGY

This appendix details the semantic inpainting method based on Generative Adversarial Networks (GANs) that was used in Chapter 5. The appendix is split into three parts. First, the overall setup is provided along with the training objectives for both the generator and discriminator. This is followed by the detailed network architecture that was used to implement the GANs. Finally, the training details are provided. The GAN implementation is credited to Dr Hyung Jin Chang.

B.1 OVERALL SETUP

Separate inpainting networks are trained for each subject i. Let D_i denote a discriminator that takes as input an image $\mathbf{x}_i \in \mathbf{R}^d$ ($d = 224 \times 224 \times 3$) of subject i from the dataset where the eyetracking glasses are not worn, and outputs a scalar representing the probability of input $\mathbf{x}_i$ being a real sample. Let G_i denote the generator that takes as input a latent random variable $\mathbf{z}_i \in \mathbf{R}^z$ ($z = 100$) sampled from a uniform noise distribution $p_{\text{noise}} = \mathcal{U}(-1, 1)$ and outputs a synthesized image $G_i(\mathbf{z}_i) \in \mathbf{R}^d$. Ideally, $D_i(\mathbf{x}_i) = 1$ when $\mathbf{x}_i$ is from a real dataset p_i of subject i and $D_i(\mathbf{x}_i) = 0$ when $\mathbf{x}_i$ is generated from G_i. In the rest of the appendix, the subscript i is omitted for clarity.

The training is performed using a least squares loss function (Mao et al., 2017), which has been shown to more stable and better performing, while having a smaller chance of mode collapsing compared to other methods (Mao et al., 2017; Zhu et al., 2017). The training objective for the discriminator of the GAN is

$$\min_D \mathcal{L}_{GAN}(D) = \mathbf{E}_{\mathbf{x}\sim p}\left[\left(D(\mathbf{x}) - 1\right)^2\right] + \mathbf{E}_{\mathbf{z}\sim p_{\text{noise}}}\left[\left(D\left(G(\mathbf{z})\right)\right)^2\right] \tag{B.1}$$

and for the generator

$$\min_G \mathcal{L}_{GAN}(G) = \mathbf{E}_{\mathbf{z}\sim p_{\text{noise}}}\left[\left(D\left(G(\mathbf{z})\right) - 1\right)^2\right]. \tag{B.2}$$

In particular, $\mathcal{L}_{GAN}(G)$ measures the realism of images generated by G, which is considered as perceptual loss:

$$\mathcal{L}_{\text{perception}}(\mathbf{z}) = \Big[D\big(G(\mathbf{z})\big) - 1\Big]^2. \tag{B.3}$$

The contextual loss is measured based on the difference between the real image $\mathbf{x}$ and the generated image $G(\mathbf{z})$ of non-masked regions as follows:

$$\mathcal{L}_{\text{context}}\left(\mathbf{z}|\mathbf{M},\mathbf{x}\right) = \left|\mathbf{M}'\odot\mathbf{x} - \mathbf{M}'\odot G(\mathbf{z})\right|, \tag{B.4}$$

where $\odot$ is the element-wise product and $\mathbf{M}'$ is the complement of $\mathbf{M}$ (i. e. to define the region that should not be inpainted).

The latent $\mathbf{z}$ variable controls the images produced by $G(\mathbf{z})$. Thus, generating the best image for inpainting is equivalent to finding the best $\hat{\mathbf{z}}$ value which minimizes a combination of the perceptual and contextual losses:

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \Big(\lambda\,\mathcal{L}_{\text{perception}}(\mathbf{z}) + \mathcal{L}_{\text{context}}(\mathbf{z}|\mathbf{M},\mathbf{x})\Big) \tag{B.5}$$

where λ is a weighting parameter. After finding $\hat{\mathbf{z}}$, the inpainted image can be generated by:

$$\mathbf{x}_{\text{inpainted}} = \mathbf{M}'\odot\mathbf{x} + \mathbf{M}\odot G(\hat{\mathbf{z}}). \tag{B.6}$$

Poisson blending (Pérez et al., 2003) is then applied to $\mathbf{x}_{\text{inpainted}}$ in order to generate the final inpainted images with seamless boundaries between inpainted and not inpainted regions.

B.2 INPAINTING NETWORK ARCHITECTURE

In order to obtain high quality images, hyperparameter tuning was performed. The generator's architecture is **z**-dense(25088)-(256)5d2s-(128)5d2s-(64)5d2s-(32)5d2s-(3)5d2s-**x**, where "(128)5c2s/(128)5d2s" denotes a convolution/deconvolution layer with 128 output feature maps and kernel size 5 with stride 2. All internal activations use SeLU (Klambauer et al., 2017) while the output layer uses tanh activation function. The discriminator's architecture is **x**-(16)5c2s-(32)5c2s-(64)5c2s-(128)5c2s-(256)5c2s-(512)5c2s-dense(1). LeakyReLU (Maas et al., 2013) is used for all internal activations ($\alpha = 0.2$) and a sigmoid activation is used for the output layer. The same architecture is used for all subjects.

B.3 TRAINING DETAILS

The Adam optimizer (Diederik P. Kingma, 2015) is used to train G and D, with a learning rate of 0.00005, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a batch size of 128 for 100 epochs. The Xavier weight initialization (Glorot and Bengio, 2010) is used for all layers. To find $\hat{\mathbf{z}}$, all values in $\mathbf{z}$ are constrained to be within $[-1, 1]$ (as suggested by Yeh et al., 2017). The network is trained for $1,000$ iterations and the weighting parameter λ is set to 0.1.

C

AUTHOR'S PUBLICATIONS

This appendix contains references and short summaries of all peer-reviewed publications and contributions made during the course of this PhD. Listed first are the publications that form the main work of the thesis, with the relevant chapters in which they are contained:

Fischer, T. and Demiris, Y. (2016), Markerless Perspective Taking for Humanoid Robots in Unconstrained Environments, *in* 'IEEE International Conference on Robotics and Automation', pp. 3309–3316. doi: 10.1109/ICRA.2016.7487504.

- Presents a framework for perspective taking implemented on the iCub humanoid robot. Two separate mechanisms are implemented corresponding to the two different levels of perspective taking, and the framework does not require any markers or prior knowledge of the environment.
- Chapter 3 is based on this article.

Fischer, T., Puigbo, J.-Y., Camilleri, D., Nguyen, P., Moulin-Frier, C., Lallée, S., Metta, G., Prescott, T. J., Demiris, Y. and Verschure, P. F. M. J. (2018), 'iCub-HRI: A software framework for complex human robot interaction scenarios on the iCub humanoid robot', *Frontiers in Robotics and AI* **5**(22), 1–9. doi: 10.3389/frobt.2018.00022.

- Introduces the iCub-HRI library that provides components related to perception, object manipulation and social interaction for the iCub humanoid robot.
- Chapter 4 is based on this article.

Fischer, T., Chang, H. J. and Demiris, Y. (2018), RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments, *in* 'European Conference on Computer Vision', pp. 339–357. doi: 10.1007/978-3-030-01249-6_21.

- Describes the design of a real-time gaze estimation framework, and presents a new dataset containing accurate gaze annotations of 17 participants. The framework is particularly suited for large camera-subject distances as commonly encountered in human-robot interactions.
- Chapter 5 is based on this article.

Fischer, T. and Demiris, Y. (2018), A Computational Model for Embodied Visual Perspective Taking: From Physical Movements to Mental Simulation, *in* 'IEEE Conference on Computer Vision and Pattern Recognition Workshop on Vision Meets Cognition'. Available from https://hdl.handle.net/10044/1/60434.

- Describes the design of a computational model for perspective taking, and puts forward the proposal that a visual attention mechanism explains the response times reported in human visual perspective taking experiments.
- Parts of Chapter 6 are based on this article.

I also co-authored the following publications, though they do not form the main work of the thesis:

Moulin-Frier, C.*, **Fischer, T.*** (contributed equally), Petit, M., Pointeau, G., Puigbo, J.-Y., Pattacini, U., Low, S. C., Camilleri, D., Nguyen, P., Hoffmann, M., Chang, H. J., Zambelli, M., Mealier, A.-L., Damianou, A., Metta, G., Prescott, T. J., Demiris, Y., Dominey, P. F. and Verschure, P. F. M. J. (2018), 'DAC-h3: A Proactive Robot Cognitive Architecture to Acquire and Express Knowledge About the World and the Self', *IEEE Transactions on Cognitive and Developmental Robotics* **10**(4), 1005–1022. doi: 10.1109/TCDS.2017.2754143.

- Presents a cognitive architecture that allows the iCub humanoid robot to engage in a proactive, mixed-initiative exploration and manipulation of the environment. Human-robot interactions experiments show that the cognitive architecture can be used with naive users.

Petit, M.*, **Fischer, T.*** (contributed equally) and Demiris, Y. (2016), 'Lifelong Augmentation of Multi-Modal Streaming Autobiographical Memories', *IEEE Transactions on Cognitive and Developmental Robotics* **8**(3), 201–213. doi: 10.1109/TAMD.2015.2507439.

- Provides a principled framework for the cumulative organization of sensorimotor and interaction data in an autobiographical memory. The framework allows processing and augmenting of these data as the processing and reasoning abilities of the iCub humanoid robot develop and further interactions with humans take place.

Petit, M.*, **Fischer, T.*** (contributed equally) and Demiris, Y. (2016), Towards the Emergence of Procedural Memories from Lifelong Multi-Modal Streaming Memories for Cognitive Robots, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Machine Learn-

ing Methods for High-Level Cognitive Capabilities in Robotics'. Available from https://hdl.handle.net/10044/1/40206.

- Extends the autobiographical memory framework introduced above with a reasoning algorithm that generalizes the robots' understanding of actions by finding the point of commonalities with previous actions.

Chang, H. J., **Fischer, T.,** Petit, M., Zambelli, M. and Demiris, Y. (2016), Kinematic Structure Correspondences via Hypergraph Matching, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 4216–4225. doi: 10.1109/CVPR.2016.457.

- Presents a framework that builds up kinematic structure correspondence matches across heterogeneous objects captured with different sensors. For example, the framework allows the iCub humanoid robot to find correspondences between a human captured using an RGB-D camera and the robot's arm recorded using the robot's eye cameras.

Chang, H. J., **Fischer, T.,** Petit, M., Zambelli, M. and Demiris, Y. (2018), 'Learning Kinematic Structure Correspondences Using Multi-Order Similarities', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(12), 2920–2934. doi: 10.1109/TPAMI.2017.2777486.

- This journal article extends the conference paper above with more rigorous experimental analyses, more comparisons to other methods, and a dataset containing more sequences.

Zambelli, M., **Fischer, T.,** Petit, M., Chang, H. J., Cully, A. and Demiris, Y. (2016), Towards Anchoring Self-Learned Representations to Those of Other Agents, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Bio-inspired Social Robot Learning in Home Scenarios'. Available from https://hdl.handle.net/10044/1/40970.

- Proposes a developmental framework that allows an iCub humanoid robot to anchor representations autonomously learned by the robot into the perspective of other agents. This represents a step towards the emergence of a mirror neuron-like system. The perspective taking algorithms presented within this thesis are a critical component for this developmental framework.

Choi, J., Chang, H. J., **Fischer, T.,** Yun, S., Jeong, J., Lee, K., Demiris, Y. and Choi, J. Y. (2018), Context-aware Deep Feature Compression for High-speed Visual Tracking, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 479–488. doi: 10.1109/CVPR.2018.00057.

- Presents a real-time object tracking framework that compresses deep features using auto-encoders that are adapted for the specific tracking scene.

Kristan et al. (2018), The sixth Visual Object Tracking VOT2018 challenge results, *in* 'European Conference on Computer Vision Workshops' (to appear). Available from http://prints.vicos.si/publications/files/365.

- Evaluates the object tracking framework described above in a large scale comparison with over 50 other recent object trackers.

Choi, J., Chang, H. J., Yun, S., **Fischer, T.,** Demiris, Y. and Choi, J. Y. (2017), Attentional Correlation Filter Network for Adaptive Visual Tracking, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 4807–4816. doi: 10.1109/CVPR.2017.513.

- Introduces an object tracking framework with high robustness and computational speed which is achieved by choosing a subset of the associated correlation filters using an attentional mechanism.

Nguyen, P. D. H., **Fischer, T.,** Chang, H. J., Pattacini, U., Demiris, Y. and Metta, G. (2018), Transferring Visuomotor Learning from Simulation to the Real World for Manipulation Tasks in a Humanoid Robot, *in* 'IEEE/RSJ Conference on Intelligent Robots and Systems', pp. 6667–6674. Available from https://goo.gl/AvBEmR.

- Introduces a method that allows transferring calibration data from simulation to the real world for eye-hand coordination of the iCub robot. It also describes a calibrator that can automatically compensate for the systematic error contained in the real robot's joint measurements.

The following non-peer-reviewed meeting abstract also resulted from the work on this thesis:

Fischer, T. and Demiris, Y. (2017), Perspective Mechanisms for Facilitating Joint Actions in Human-Robot Collaborations, *in* 'Joint Action Meeting'. Available from https://goo.gl/c6ewBR.

- Describes preliminary research towards learning the forward model that is contained in the computational model presented in Chapter 6. The main hypothesis is that accurate forward models are needed for perspective taking, which predicts that children are typically more egocentric as their forward model still matures.

BIBLIOGRAPHY

Admoni, H. and Scassellati, B. (2017), 'Social Eye Gaze in Human-Robot Interaction: A Review', *Journal of Human-Robot Interaction* **6**(1), 25–63. doi: 10.5898/JHRI.6.1.Admoni (page 28).

Akkaladevi, S. C., Plasch, M., Pichler, A. and Rinner, B. (2016), Human Robot Collaboration to Reach a Common Goal in an Assembly Process, *in* 'European Starting AI Researcher Symposium', pp. 3–14. doi: 10.3233/978-1-61499-682-8-3 (page 25).

Alsmith, A. J., Ferrè, E. R. and Longo, M. R. (2017), 'Dissociating contributions of head and torso to spatial reference frames: The misalignment paradigm', *Consciousness and Cognition* **53**, 105–114. doi: 10.1016/j.concog.2017.06.005 (pages 35, 102).

Amanatides, J. and Woo, A. (1987), 'A Fast Voxel Traversal Algorithm for Ray Tracing', *Eurographics* **87**(3), 3–10. Retrieved from `https://diglib.eg.org/handle/10.2312/egtp19871000` (page 49).

Anderson, J. R., Montant, M. and Schmitt, D. (1996), 'Rhesus monkeys fail to use gaze direction as an experimenter-given cue in an object-choice task', *Behavioural Processes* **37**(1), 47–55. doi: 10.1016/0376-6357(95)00074-7 (page 107).

Ba, S. and Odobez, J.-M. (2005), Evaluation of Multiple Cues Head Pose Estimation Algorithms in Natural Environments, *in* 'International Conference on Multimedia & Expo', pp. 1330–1333. doi: 10.1109/ICME.2005.1521675 (page 76).

Bajcsy, R., Aloimonos, Y. and Tsotsos, J. K. (2018), 'Revisiting active perception', *Autonomous Robots* **42**(2), 177–196. doi: 10.1007/s10514-017-9615-3 (page 128).

Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G. and Verdera, J. (2001), 'Filling-in by joint interpolation of vector fields and gray levels', *IEEE Transactions on Image Processing* **10**(8), 1200–1211. doi: 10.1109/83.935036 (page 84).

Baltrusaitis, T., Robinson, P. and Morency, L. P. (2012), 3D Constrained Local Model for Rigid and Non-Rigid Facial Tracking, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 2610–2617. doi: 10.1109/CVPR.2012.6247980 (pages 30, 76).

Baltrusaitis, T., Robinson, P. and Morency, L.-P. (2013), Constrained Local Neural Fields for Robust Facial Landmark Detection in the Wild, *in* 'IEEE International Conference on Computer Vision Workshops', pp. 354–361. doi: 10.1109/ICCVW.2013.54 (page 90).

Barnes, C., Shechtman, E., Finkelstein, A. and Goldman, D. B. (2009), 'PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing', *ACM Transactions on Graphics* **28**(3), 24:1–24:11. doi: 10.1145/1531326.1531330 (page 84).

Benfold, B. and Reid, I. (2011), Stable Multi-Target Tracking in Real-Time Surveillance Video, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 3457–3464. doi: 10.1109/CVPR.2011.5995667 (page 76).

Bertalmio, M., Sapiro, G., Caselles, V. and Ballester, C. (2000), Image inpainting, *in* 'Annual Conference on Computer Graphics and Interactive Techniques', pp. 417–424. doi: 10.1145/344779.344972 (page 84).

Besl, P. J. and McKay, N. D. (1992), 'A method for registration of 3-D shapes', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(2), 239–256. doi: 10.1109/34.121791 (pages 80, 86).

Blakemore, S.-J. and Decety, J. (2001), 'From the Perception of Action to the Understanding of Intention', *Nature Reviews Neuroscience* **2**, 561–567. doi: 10.1038/35086023 (page 97).

Blanke, O., Mohr, C., Michel, C. M., Pascual-Leone, A., Brugger, P., Seeck, M., Landis, T. and Thut, G. (2005), 'Linking Out-of-Body Experience and Self Processing to Mental Own-Body Imagery at the Temporoparietal Junction', *Journal of Neuroscience* **25**(3), 550–557. doi: 10.1523/JNEUROSCI.2612-04.2005 (page 33).

Boucher, J. D., Pattacini, U., Lelong, A., Bailly, G., Elisei, F., Fagel, S., Dominey, P. F. and Ventre-Dominey, J. (2012), 'I reach faster when I see you look: Gaze effects in human-human and human-robot face-to-face cooperation', *Frontiers in Neurorobotics* **6**(3), 1–11. doi: 10.3389/fnbot.2012.00003 (page 28).

Breazeal, C., Berlin, M., Brooks, A., Gray, J. and Thomaz, A. L. (2006), 'Using perspective taking to learn from ambiguous demonstrations', *Robotics and Autonomous Systems* **54**(5), 385–393. doi: 10.1016/j.robot.2006.02.004 (pages 17, 25).

Breitenstein, M. D., Kuettel, D., Weise, T., van Gool, L. and Pfister, H. (2008), Real-Time Face Pose Estimation from Single Range Images, *in* 'IEEE Conference on Computer Vision and Pattern Recognition'. doi: 10.1109/CVPR.2008.4587807 (page 76).

Bukowski, H. (2018), 'The Neural Correlates of Visual Perspective Taking: a Critical Review', *Current Behavioral Neuroscience Reports* **5**, 189–197. doi: 10.1007/s40473-018-0157-6 (page 36).

Chamveha, I., Sugano, Y., Sugimura, D., Siriteerakul, T., Okabe, T., Sato, Y. and Sugimoto, A. (2013), 'Head direction estimation from low resolution images with scene adaptation', *Computer Vision and Image Understanding* **117**(10), 1502–1511. doi: 10.1016/j.cviu.2013.06.005 (pages 28, 75, 76).

Chan, T. F. and Shen, J. (2002), 'Mathematical Models for Local Nontexture Inpaintings', *SIAM Journal on Applied Mathematics* **62**(3), 1019–1043. doi: 10.1137/S0036139900368844 (page 84).

Chang, H. J., **Fischer, T.**, Petit, M., Zambelli, M. and Demiris, Y. (2016), Kinematic Structure Correspondences via Hypergraph Matching, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 4216–4225. doi: 10.1109/CVPR.2016.457 (page 128).

Chang, H. J., **Fischer, T.**, Petit, M., Zambelli, M. and Demiris, Y. (2018), 'Learning Kinematic Structure Correspondences Using Multi-Order Similarities', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(12), 2920–2934. doi: 10.1109/TPAMI.2017.2777486 (page 128).

Cheng, Y., Lu, F. and Zhang, X. (2018), Appearance-Based Gaze Estimation via Evaluation-Guided Asymmetric Regression, *in* 'European Conference on Computer Vision', pp. 105–121. doi: 10.1007/978-3-030-01264-9_7 (page 29).

Cole, G. G., Atkinson, M., Le, A. T. D. and Smith, D. T. (2016), 'Do humans spontaneously take the perspective of others?', *Acta Psychologica* **164**, 165–168. doi: 10.1016/j.actpsy.2016.01.007 (page 34).

Cox, D. D. and Dean, T. (2014), 'Neural Networks and Neuroscience-Inspired Computer Vision', *Current Biology* **24**(18), R921–R929. doi: 10.1016/j.cub.2014.08.026 (page 37).

Cox, D., Meyers, E. and Sinha, P. (2004), 'Contextually Evoked Object-Specific Responses in Human Visual Cortex', *Science* **304**(5667), 115–117. doi: 10.1126/science.1093110 (page 38).

Cristani, M., Bazzani, L., Paggetti, G., Fossati, A., Tosato, D., Bue, A. D., Menegaz, G. and Murino, V. (2011), Social interaction discovery by statistical analysis of F-formations, *in* 'British Machine Vision Conference', pp. 23.1–23.12. doi: 10.5244/C.25.23 (page 76).

Damianou, A., Titsias, M. K. and Lawrence, N. D. (2011), Variational Gaussian Process Dynamical Systems, *in* 'Advances in Neural Information Processing Systems', pp. 2510–2518. arXiv: 1107.4985 (page 66).

Demiris, Y. (2007), 'Prediction of intent in robotics and multi-agent systems', *Cognitive Processing* **8**(3), 151–158. doi: 10.1007/s10339-007-0168-9 (pages 18, 23).

Demiris, Y. and Hayes, G. (2002), Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model, *in* 'Imitation in Animals and Artifacts', MIT Press, pp. 327–361. Retrieved from https://www.inf.ed.ac.uk/publications/online/0254.pdf (pages 37, 38, 98).

Demiris, Y. and Johnson, M. (2003), 'Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning', *Connection Science* **15**(4), 231–243. doi: 10.1080/09540090310001655129 (page 52).

Demiris, Y. and Meltzoff, A. (2008), 'The Robot in the Crib: A Developmental Analysis of Imitation Skills in Infants and Robots', *Infant and Child Development* **17**(1), 43–53. doi: 10.1002/icd.543 (page 126).

Deng, H. and Zhu, W. (2017), Monocular Free-Head 3D Gaze Tracking with Deep Learning and Geometry Constraints, *in* 'IEEE International Conference on Computer Vision', pp. 3162–3171. doi: 10.1109/ICCV.2017.341 (pages 30, 90).

Deroualle, D., Borel, L., Devèze, A. and Lopez, C. (2015), 'Changing perspective: The role of vestibular signals', *Neuropsychologia* **79**, 175–185. doi: 10.1016/j.neuropsychologia.2015.08.022 (page 107).

Desimone, R. and Duncan, J. (1995), 'Neural Mechanisms of Selective Visual', *Annual Review of Neuroscience* **18**(1), 193–222. doi: 10.1146/annurev.ne.18.030195.001205 (page 39).

Devin, S. and Alami, R. (2016), An Implemented Theory of Mind to Improve Human-Robot Shared Plans Execution, *in* 'International Conference on Human-Robot Interaction', pp. 319–326. doi: 10.1109/HRI.2016.7451768 (page 25).

DiCarlo, J. J. and Cox, D. D. (2007), 'Untangling invariant object recognition', *Trends in Cognitive Sciences* **11**(8), 333–341. doi: 10.1016/j.tics.2007.06.010 (page 38).

Diederik P. Kingma, J. B. (2015), Adam: A Method for Stochastic Optimization, *in* 'International Conference on Learning Representations'. arXiv: 1412.6980 (pages 88, 137).

Duran, N. and Dale, R. (2016), 'Toward Integrative Dynamic Models for Adaptive Perspective Taking', *Topics in Cognitive Science* **8**(4), 761–779. doi: 10.1111/tops.12219 (page 37).

Efros, A. and Leung, T. (1999), Texture Synthesis by Non-parametric Sampling, *in* 'IEEE International Conference on Computer Vision', pp. 1033–1038. doi: 10.1109/ICCV.1999.790383 (page 84).

Elekes, F., Varga, M. and Király, I. (2017), 'Level-2 perspectives computed quickly and spontaneously: Evidence from eight- to 9.5-year-old children', *British Journal of Developmental Psychology* **35**(4), 609–622. doi: 10.1111/bjdp.12201 (page 34).

Elhayek, A., de Aguiar, E., Jain, A., Thompson, J., Pishchulin, L., Andriluka, M., Bregler, C., Schiele, B. and Theobalt, C. (2017), 'MARCOnI—ConvNet-Based MARker-Less Motion Capture in Outdoor and Indoor Scenes', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(3), 501–514. doi: 10.1109/TPAMI.2016.2557779 (page 81).

Elkady, A. and Sobh, T. (2012), 'Robotics Middleware: A Comprehensive Literature Survey and Attribute-Based Bibliography', *Journal of Robotics* **2012**(959013), 1–15. doi: 10.1155/2012/959013 (page 27).

Evans, E. (2004), *Domain-driven design: tackling complexity in the heart of software*, Addison-Wesley Professional. (page 64).

Falck-Ytter, T., Gredebäck, G. and von Hofsten, C. (2006), 'Infants predict other people's action goals', *Nature Neuroscience* **9**(7), 878–879. doi: 10.1038/nn1729 (page 28).

Fanelli, G., Dantone, M., Gall, J., Fossati, A. and Van Gool, L. (2013), 'Random Forests for Real Time 3D Face Analysis', *International Journal of Computer Vision* **101**(3), 437–458. doi: 10.1007/s11263-012-0549-0 (pages 31, 41, 46, 47, 76).

Fanello, S., Pattacini, U., Gori, I., Tikhanoff, V., Randazzo, M., Roncone, A., Odone, F. and Metta, G. (2014), 3D Stereo Estimation and Fully Automated Learning of Eye-Hand Coordination in Humanoid Robots, *in* 'IEEE-RAS International Conference on Humanoid Robots', pp. 1028–1035. doi: 10.1109/HUMANOIDS.2014.7041491 (page 45).

Fischer, T., Chang, H. J. and Demiris, Y. (2018), RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments, *in* 'European Conference on Computer Vision', pp. 339–357. doi: 10.1007/978-3-030-01249-6_21 (page 75).

Fischer, T. and Demiris, Y. (2016), Markerless Perspective Taking for Humanoid Robots in Unconstrained Environments, *in* 'IEEE International Conference on Robotics and Automation', pp. 3309–3316. doi: 10.1109/ICRA.2016.7487504 (page 42).

Fischer, T. and Demiris, Y. (2018), A Computational Model for Embodied Visual Perspective Taking: From Physical Movements to Mental Simulation, *in* 'IEEE Conference on Computer Vision and Pattern Recognition Workshop on Vision Meets Cognition'. Retrieved from http://hdl.handle.net/10044/1/60434 (page 97).

Fischer, T., Puigbo, J.-Y., Camilleri, D., Nguyen, P., Moulin-Frier, C., Lallée, S., Metta, G., Prescott, T. J., Demiris, Y. and Verschure, P. F. M. J. (2018), 'iCub-HRI: A software framework for complex human-robot interaction scenarios on the iCub humanoid robot', *Frontiers in Robotics and AI* **5**(22), 1–9. doi: 10.1109/frobt.2018.00022 (page 61).

Fisher, R. B. (2004), The PETS04 surveillance ground-truth data sets, *in* 'International Workshop on Performance Evaluation of Tracking and Surveillance'. Retrieved from http://homepages.inf.ed.ac.uk/rbf/CAVIAR/PAPERS/pets04.pdf (pages 31, 76).

Fitzpatrick, P., Ceseracciu, E., Domenichelli, D. E., Paikan, A., Metta, G. and Natale, L. (2014), 'A middle way for robotics middleware', *Journal of Software Engineering for Robotics* **5**(2), 42–49. doi: 10.6092/JOSER_2014_05_02_p42 (page 62).

Fitzpatrick, P., Metta, G. and Natale, L. (2006), 'YARP: Yet Another Robot Platform', *International Journal of Advanced Robotic Systems* **3**(1), 43–48. doi: 10.5772/5761 (page 131).

Flavell, J. H. (1977), The development of knowledge about visual perception, *in* 'Nebraska Symposium on Motivation', Vol. 25, pp. 43–76. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/753993 (page 97).

Flavell, J. H., Everett, B. A., Croft, K. and Flavell, E. R. (1981), 'Young Children's Knowledge About Visual Perception: Further Evidence for the Level 1-Level 2 Distinction', *Developmental Psychology* **17**(1), 99–103. doi: 10.1037/0012-1649.17.1.99 (pages 32, 41, 49).

Florence, P. R., Manuelli, L. and Tedrake, R. (2018), Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation, *in* 'Conference on Robot Learning', pp. 373–385. arXiv: 1806.08756 (page 125).

Fong, T., Nourbakhsh, I. and Dautenhahn, K. (2003), 'A survey of socially interactive robots', *Robotics and Autonomous Systems* **42**(3-4), 143–166. doi: 10.1016/S0921-8890(02)00372-X (page 17).

Frith, C. D. and Frith, U. (1999), 'Interacting minds - a biological basis', *Science* **286**, 1692–1695. doi: 10.1126/science.286.5445.1692 (page 97).

Funes Mora, K. A., Monay, F. and Odobez, J.-M. (2014), EYEDIAP: A Database for the Development and Evaluation of Gaze Estimation Algorithms from RGB and RGB-D Cameras, *in* 'ACM Symposium on Eye Tracking Research and Applications', pp. 255–258. doi: 10.1145/2578153.2578190 (pages 30, 76, 82, 83, 90).

Funes-Mora, K. A. and Odobez, J. M. (2016), 'Gaze Estimation in the 3D Space Using RGB-D Sensors: Towards Head-Pose and User Invariance', *International Journal of Computer Vision* **118**, 194–216. doi: 10.1007/s11263-015-0863-4 (pages 29, 75).

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994), *Design Patterns: Elements of Reusable Object Oriented Software*, Addison-Wesley. (page 64).

Gardner, M. R., Brazier, M., Edmonds, C. J. and Gronholm, P. C. (2013), 'Strategy modulates spatial perspective-taking: evidence for dissociable disembodied and embodied routes.', *Frontiers in Human Neuroscience* **7**(457), 1–8. doi: 10.3389/fnhum.2013.00457 (pages 35, 126).

Gentili, R. J., Oh, H., Huang, D.-W., Katz, G. E., Miller, R. H. and Reggia, J. A. (2015), 'A Neural Architecture for Performing Actual and Mentally Simulated Movements During Self-Intended and Observed Bimanual Arm Reaching Movements', *International Journal of Social Robotics* **7**(3), 371–392. doi: 10.1007/s12369-014-0276-5 (page 37).

Glorot, X. and Bengio, Y. (2010), Understanding the difficulty of training deep feedforward neural networks, *in* 'International Conference on Artificial Intelligence and Statistics', pp. 249–256. Retrieved from http://proceedings.mlr.press/v9/glorot10a.html (pages 88, 137).

Gong, S., Ong, E.-J. and Mckenna, S. (1998), Learning to associate faces across views in vector space of similarities to prototypes, *in* 'British Machine Vision Conference', pp. 54–63. doi: 10.5244/C.12.6 (page 76).

Gooding-Williams, G., Wang, H. and Kessler, K. (2017), 'THETA-Rhythm Makes the World Go Round: Dissociative Effects of TMS Theta Versus Alpha Entrainment of Right pTPJ on Embodied Perspective Transformations', *Brain Topography* **30**(5), 561–564. doi: 10.1007/s10548-017-0557-z (pages 36, 98).

Gross, R., Matthews, I., Cohn, J., Kanade, T. and Baker, S. (2008), Multi-PIE, *in* 'IEEE International Conference on Automatic Face & Gesture Recognition'. doi: 10.1109/AFGR.2008.4813399 (page 76).

Hays, J. and Efros, A. A. (2007), 'Scene Completion Using Millions of Photographs', *ACM Transactions on Graphics* **26**(3), 4:1–4:7. doi: 10.1145/1276377.1276382 (page 84).

Heckman, J. J. (2006), 'Skill Formation and the Economics of Investing in Disadvantaged Children', *Science* **312**(5782), 1900–1902. doi: 10.1126/science.1128898 (page 97).

Herrmann, E., Call, J., Hernandez-Lloreda, M. V., Hare, B. and Tomasello, M. (2007), 'Humans have evolved specialised skills of social cognition: The cultural intelligence hypothesis', *Science* **317**(5843), 1360–1366. doi: 10.1126/science.1146282 (page 97).

Huang, C.-M. and Mutlu, B. (2012), Robot behavior toolkit, *in* 'ACM/IEEE International Conference on Human-Robot Interaction', pp. 25–32. doi: 10.1145/2157689.2157694 (page 27).

Huang, Q., Veeraraghavan, A. and Sabharwal, A. (2017), 'TabletGaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets', *Machine Vision and Applications* **28**(5-6), 445–461. doi: 10.1007/s00138-017-0852-4 (pages 30, 76, 82, 83).

Hughes, D. E., Vasquez, E. and Nicsinger, E. (2016), Improving perspective taking and empathy in children with autism spectrum disorder, *in* 'IEEE International Conference on Serious Games and Applications for Health'. doi: 10.1109/SeGAH.2016.7586232 (page 24).

Iizuka, S., Simo-Serra, E. and Ishikawa, H. (2017), 'Globally and locally consistent image completion', *ACM Transactions on Graphics* **36**(4), 107:1–107:14. doi: 10.1145/3072959.3073659 (page 84).

Janczyk, M. (2013), 'Level 2 perspective taking entails two processes: Evidence from PRP experiments', *Journal of Experimental Psychology: Learning, Memory, and Cognition* **39**(6), 1878–1887. doi: 10.1037/a0033336 (pages 35, 111).

Jang, M., Kim, J. and Ahn, B.-K. (2015), A software framework design for social human-robot interaction, *in* 'International Conference on Ubiquitous Robots and Ambient Intelligence', pp. 411–412. doi: 10.1109/URAI.2015.7358887 (page 27).

Jayagopi, D. B., Sheiki, S., Klotz, D., Wienke, J., Odobez, J.-M., Wrede, S., Khalidov, V., Nyugen, L., Wrede, B. and Gatica-Perez, D. (2013), The Vernissage Corpus: A Conversational Human-Robot-Interaction Dataset, *in* 'ACM/IEEE International Conference on Human-Robot Interaction', pp. 149–150. doi: 10.1109/HRI.2013.6483545 (page 76).

Johnson, M. and Demiris, Y. (2005*a*), 'Perceptual Perspective Taking and Action Recognition', *International Journal of Advanced Robotic Systems* **2**(4), 301–308. doi: 10.5772/5775 (pages 17, 25, 26, 49, 52).

Johnson, M. and Demiris, Y. (2005*b*), Perspective Taking Through Simulation, *in* 'Towards Autonomous Robotic Systems Conference', pp. 119–126. Retrieved from http://hdl.handle.net/10044/1/12692 (page 25).

Johnson, M. and Demiris, Y. (2007), Visuo-Cognitive Perspective Taking for Action Recognition, *in* 'International Symposium on Imitation in Ani-

mals and Artifacts', pp. 262–269. Retrieved from http://hdl.handle.net/10044/1/12622 (pages 25, 26, 49, 52).

Kalal, Z., Mikolajczyk, K. and Matas, J. (2012), 'Tracking-Learning-Detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(7), 1409–1422. doi: 10.1109/TPAMI.2011.239 (page 45).

Kar, A. and Corcoran, P. (2017), 'A Review and Analysis of Eye-Gaze Estimation Systems, Algorithms and Performance Evaluation Methods in Consumer Platforms', *IEEE Access* **5**, 16495–16519. doi: 10.1109/ACCESS.2017.2735633 (page 31).

Karg, K., Schmelz, M., Call, J. and Tomasello, M. (2016), 'Differing views: Can chimpanzees do Level 2 perspective-taking?', *Animal Cognition* **19**(3), 555–564. doi: 10.1007/s10071-016-0956-7 (page 107).

Kassner, M., Patera, W. and Bulling, A. (2014), Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction, *in* 'ACM International Joint Conference on Pervasive and Ubiquitous Computing', pp. 1151–1160. doi: 10.1145/2638728.2641695 (pages 78, 79, 81, 133).

Kazemi, V. and Sullivan, J. (2014), One millisecond face alignment with an ensemble of regression trees, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 1867–1874. doi: 10.1109/CVPR.2014.241 (page 90).

Kendon, A. (1967), 'Some functions of gaze-direction in social interaction', *Acta Psychologica* **26**(1), 22–63. doi: 10.1016/0001-6918(67)90005-4 (page 28).

Kennedy, J., Baxter, P. and Belpaeme, T. (2015), Head Pose Estimation is an Inadequate Replacement for Eye Gaze in Child-Robot Interaction, *in* 'ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts', pp. 35–36. doi: 10.1145/2701973.2701988 (page 28).

Kennedy, W. G., Bugajska, M. D., Harrison, A. M. and Trafton, J. G. (2009), '"Like-Me" Simulation as an Effective and Cognitively Plausible Basis for Social Robotics', *International Journal of Social Robotics* **1**(2), 181–194. doi: 10.1007/s12369-009-0014-6 (pages 24, 26).

Kessler, K. and Rutherford, H. (2010), 'The two forms of visuo-spatial perspective taking are differently embodied and subserve

different spatial prepositions', *Frontiers in Psychology* **1**(213), 1–12. doi: 10.3389/fpsyg.2010.00213 (pages 33, 35, 55, 107, 110, 111).

Kessler, K. and Thomson, L. A. (2010), 'The embodied nature of spatial perspective taking: Embodied transformation versus sensorimotor interference', *Cognition* **114**(1), 72–88. doi: 10.1016/j.cognition.2009.08.015 (pages 35, 36, 97, 98, 105, 107, 108, 109, 110, 111, 112, 113).

Kessler, K. and Wang, H. (2012), 'Spatial Perspective Taking is an Embodied Process, but Not for Everyone in the Same Way: Differences Predicted by Sex and Social Skills Score', *Spatial Cognition and Computation* **12**, 133–158. doi: 10.1080/13875868.2011.634533 (pages 33, 98, 107, 113, 114, 115).

Klambauer, G., Unterthiner, T., Mayr, A. and Hochreiter, S. (2017), Self-Normalizing Neural Networks, *in* 'Advances in Neural Information Processing Systems', pp. 971–980. arXiv: 1706.02515 (page 136).

Kockler, H., Scheef, L., Tepest, R., David, N., Bewernick, B. H., Newen, A., Schild, H. H., May, M. and Vogeley, K. (2010), 'Visuospatial perspective taking in a dynamic environment: Perceiving moving objects from a first-person-perspective induces a disposition to act', *Consciousness and Cognition* **19**(3), 690–701. doi: 10.1016/j.concog.2010.03.003 (page 107).

Koutras, P. and Maragos, P. (2015), Estimation of eye gaze direction angles based on active appearance models, *in* 'IEEE International Conference on Image Processing', pp. 2424–2428. doi: 10.1109/ICIP.2015.7351237 (page 31).

Krafka, K., Khosla, A., Kellnhofer, P. and Kannan, H. (2016), Eye Tracking for Everyone, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 2176–2184. doi: 10.1109/CVPR.2016.239 (pages 29, 30, 75, 76, 82, 83, 91).

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2017), 'ImageNet classification with deep convolutional neural networks', *Communications of the ACM* **60**(6), 84–90. doi: 10.1145/3065386 (page 87).

Krupke, D., Starke, S., Einig, L., Steinicke, F. and Zhang, J. (2017), Prototyping of Immersive HRI Scenarios, *in* 'International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines', pp. 537–544. doi: 10.1142/9789813231047_0065 (page 27).

Kummer, H., Anzenberger, G. and Hemelrijk, C. K. (1996), 'Hiding and perspective taking in long-tailed macaques (Macaca fascicularis)', *Journal*

of Comparative Psychology **110**(1), 97–102. doi: 10.1037/0735-7036.110.1.97 (page 107).

La Cascia, M., Sclaroff, S. and Athitsos, V. (2000), 'Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Textured-Mapped 3D Models', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(4), 322–336. doi: 10.1109/34.845375 (page 76).

Labbé, M. and Michaud, F. (2018), 'RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation', *Journal of Field Robotics* (to appear). doi: 10.1002/rob.21831 (pages 41, 43).

Laine, S. and Karras, T. (2011), 'Efficient Sparse Voxel Octrees', *IEEE Transactions on Visualization and Computer Graphics* **17**(8), 1048–1059. doi: 10.1109/TVCG.2010.240 (page 49).

Lallée, S. and Verschure, P. (2015), 'How? Why? What? Where? When? Who? Grounding Ontology in the Actions of a Situated Social Agent', *Robotics* **4**(2), 169–193. doi: 10.3390/robotics4020169 (pages 63, 67).

Lane, I., Prasad, V., Sinha, G., Umuhoza, A., Luo, S., Chandrashekaran, A. and Raux, A. (2012), HRItk: The Human-robot Interaction ToolKit Rapid Development of Speech-centric Interactive Systems in ROS, *in* 'NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data', pp. 41–44. Retrieved from `http://www.aclweb.org/anthology/W12-1817` (page 27).

LaValle, S. M. (2006), *Planning Algorithms*, Cambridge University Press, Cambridge, U.K. (page 52).

Lemaignan, S., Ros, R., Alami, R. and Beetz, M. (2011), What are you talking about? Grounding dialogue in a perspective-aware robotic architecture, *in* 'IEEE International Symposium on Robot and Human Interactive Communication', pp. 107–112. doi: 10.1109/ROMAN.2011.6005249 (pages 24, 26).

Lemaignan, S., Warnier, M., Sisbot, E. A., Clodic, A. and Alami, R. (2017), 'Artificial cognition for social human-robot interaction: An implementation', *Artificial Intelligence* **247**, 45–69. doi: 10.1016/j.artint.2016.07.002 (pages 17, 26, 46, 49, 52).

Libby, L. K. and Eibach, R. P. (2002), 'Looking Back in Time: Self-Concept Change Affects Visual Perspective in Autobiographical

Memory', *Journal of Personality and Social Psychology* **82**(2), 167–179. doi: 10.1037/0022-3514.82.2.167 (page 127).

Lopes, M. and Santos-Victor, J. (2005), 'Visual Learning by Imitation With Motor Representations', *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* **35**(3), 438–449. doi: 10.1109/TSMCB.2005.846654 (page 37).

Lu, F., Sugano, Y., Okabe, T. and Sato, Y. (2015), 'Gaze Estimation From Eye Appearance: A Head Pose-Free Method via Eye Image Synthesis', *IEEE Transactions on Image Processing* **24**(11), 3680–3693. doi: 10.1109/TIP.2015.2445295 (pages 30, 31).

Maas, A. L., Hannun, A. Y. and Ng, A. Y. (2013), Rectifier nonlinearities improve neural network acoustic models, *in* 'International Conference on Machine Learning Workshop on Deep Learning for Audio, Speech and Language Processing'. Retrieved from `https://sites.google.com/site/deeplearningicml2013/relu_hybrid_icml2013_final.pdf` (page 136).

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z. and Smolley, S. P. (2017), Least Squares Generative Adversarial Networks, *in* 'IEEE International Conference on Computer Vision', pp. 2813–2821. doi: 10.1109/ICCV.2017.304 (page 135).

Marin-Jimenez, M. J., Zisserman, A., Eichner, M. and Ferrari, V. (2014), 'Detecting People Looking at Each Other in Videos', *International Journal of Computer Vision* **106**(3), 282–296. doi: 10.1007/s11263-013-0655-7 (page 126).

Martinez-Hernandez, U., Damianou, A., Camilleri, D., Boorman, L. W., Lawrence, N. and Prescott, T. J. (2016), An integrated probabilistic framework for robot perception, learning and memory, *in* 'IEEE International Conference on Robotics and Biomimetics', pp. 1796–1801. doi: 10.1109/ROBIO.2016.7866589 (page 66).

Mathews, Z., Bermúdez i Badia, S. and Verschure, P. F. (2012), 'PASAR: An integrated model of prediction, anticipation, sensation, attention and response for artificial sensorimotor systems', *Information Sciences* **186**(1), 1–19. doi: 10.1016/j.ins.2011.09.042 (page 66).

May, M. (2004), 'Imaginal perspective switches in remembered environments: Transformation versus interference accounts', *Cognitive Psychology* **48**(2), 163–206. doi: 10.1016/S0010-0285(03)00127-0 (pages 35, 97).

Meltzoff, A. N. (2005), Imitation and Other Minds: The "Like Me" Hypothesis, *in* 'Perspectives on Imitation: From Neuroscience to Social Science', Vol. 2, MIT Press, pp. 55–77. Retrieved from https://www.researchgate.net/publication/237231822_1_Imitation_and_Other_Minds_The_Like_Me_Hypothesis (page 18).

Metta, G., Natale, L., Nori, F., Sandini, G., Vernon, D., Fadiga, L., von Hofsten, C., Rosander, K., Lopes, M., Santos-Victor, J., Bernardino, A. and Montesano, L. (2010), 'The iCub humanoid robot: An open-systems platform for research in cognitive development', *Neural Networks* **23**(8-9), 1125–1134. doi: 10.1016/j.neunet.2010.08.010 (page 132).

Michelon, P. and Zacks, J. M. (2006), 'Two kinds of visual perspective taking', *Perception & Psychophysics* **68**(2), 327–337. doi: 10.3758/BF03193680 (pages 32, 33, 41, 49, 50, 55, 97, 105, 107, 108, 109, 111).

Milford, M. and Wyeth, G. (2008), 'Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System', *IEEE Transactions on Robotics* **24**(5), 1038–1053. doi: 10.1109/TRO.2008.2004520 (page 38).

Milford, M. J., Wiles, J. and Wyeth, G. F. (2010), 'Solving Navigational Uncertainty Using Grid Cells on Robots', *PLoS Computational Biology* **6**(11). doi: 10.1371/journal.pcbi.1000995 (page 38).

Min, R., Kose, N. and Dugelay, J.-L. (2014), 'KinectFaceDB: a Kinect Face Database for Face Recognition', *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **44**(11), 1534–1548. doi: 10.1109/TSMC.2014.2331215 (page 76).

Moll, H. and Meltzoff, A. N. (2011*a*), 'How does it look? Level 2 perspective-taking at 36 months of age', *Child Development* **82**(2), 661–73. doi: 10.1111/j.1467-8624.2010.01571.x (page 33).

Moll, H. and Meltzoff, A. N. (2011*b*), Perspective-Taking and its Foundation in Joint Attention, *in* J. Roessler, H. Lerman and N. Eilan, eds, 'Perception, Causation, and Objectivity', Oxford University Press, Oxford, UK, chapter 16, pp. 287–304. (page 24).

Moll, H. and Tomasello, M. (2006), 'Level 1 perspective-taking at 24 months of age', *British Journal of Developmental Psychology* **24**(3), 603–613. doi: 10.1348/026151005X55370 (page 32).

Moore, C., Dunham, P. J. and Dunham, P. (2014), *Joint attention: Its origins and role in development*, Psychology Press. (page 23).

Moulin-Frier, C., **Fischer, T.**, Petit, M., Pointeau, G. G., Puigbo, J.-Y., Pattacini, U., Low, S. C., Camilleri, D., Nguyen, P., Hoffmann, M., Chang, H. J., Zambelli, M., Mealier, A.-L., Damianou, A., Metta, G., Prescott, T. J., Demiris, Y., Dominey, P. F. and Verschure, P. F. M. J. (2018), 'DAC-h3: A Proactive Robot Cognitive Architecture to Acquire and Express Knowledge About the World and the Self', *IEEE Transactions on Cognitive and Developmental Systems* **10**(4), 1005–1022. doi: 10.1109/TCDS.2017.2754143 (pages 42, 69, 72).

Mukherjee, S. S. and Robertson, N. M. (2015), 'Deep Head Pose: Gaze-Direction Estimation in Multimodal Video', *IEEE Transactions on Multimedia* **17**(11), 2094–2107. doi: 10.1109/TMM.2015.2482819 (pages 28, 31, 75).

Müller, P., Huang, M. X., Zhang, X. and Bulling, A. (2018), Robust Eye Contact Detection in Natural Multi-Person Interactions Using Gaze and Speaking Behaviour, *in* 'ACM Symposium on Eye Tracking Research & Applications', pp. 31:1–31:10. doi: 10.1145/3204493.3204549 (page 32).

Nagai, Y., Hosoda, K., Morita, A. and Asada, M. (2003), 'A constructive model for the development of joint attention', *Connection Science* **15**(4), 211–229. doi: 10.1080/09540090310001655101 (page 23).

Nakajo, R., Murata, S., Arie, H. and Ogata, T. (2015), Acquisition of viewpoint representation in imitative learning from own sensory-motor experiences, *in* 'Joint International Conference on Development and Learning and Epigenetic Robotics', pp. 326–331. doi: 10.1109/DEVLRN.2015.7346166 (page 36).

Natale, L., Paikan, A., Randazzo, M. and Domenichelli, D. E. (2016), 'The iCub Software Architecture: Evolution and Lessons Learned', *Frontiers in Robotics and AI* **3**(24), 1–21. doi: 10.3389/frobt.2016.00024 (page 62).

Nelder, J. A. and Mead, R. (1965), 'A simplex method for function minimization', *The Computer Journal* **7**(4), 308–313. (page 80).

Ogata, T., Yokoya, R., Tani, J., Komatani, K. and Okuno, H. G. (2009), Prediction and imitation of other's motions by reusing own forward-inverse model in robots, *in* 'IEEE International Conference on Robotics and Automation', pp. 4144–4149. doi: 10.1109/ROBOT.2009.5152363 (page 36).

OptiTrack (2018), 'Flex 3'. http://optitrack.com/products/flex-3/ (accessed on: 2018-07-01) (pages 79, 134).

Palinko, O., Rea, F., Sandini, G. and Sciutti, A. (2015), Eye Gaze Tracking for a Humanoid Robot, *in* 'IEEE-RAS International Conference on Humanoid Robots', pp. 318–324. doi: 10.1109/HUMANOIDS.2015.7363561 (pages 31, 75).

Pandey, A. K. and Alami, R. (2010), Mightability Maps: A Perceptual Level Decisional Framework for Co-operative and Competitive Human-Robot Interaction, *in* 'International Conference on Intelligent Robots and Systems', pp. 5842–5848. doi: 10.1109/IROS.2010.5651503 (pages 46, 49).

Pandey, A. K., Ali, M. and Alami, R. (2013), 'Towards a Task-Aware Proactive Sociable Robot Based on Multi-state Perspective-Taking', *International Journal of Social Robotics* **5**(2), 215–236. doi: 10.1007/s12369-013-0181-3 (pages 17, 26, 46, 49, 52).

Park, H. S., Jain, E. and Sheikh, Y. (2013), Predicting Primary Gaze Behavior Using Social Saliency Fields, *in* 'IEEE International Conference on Computer Vision', pp. 3503–3510. doi: 10.1109/ICCV.2013.435 (page 29).

Park, S., Spurr, A. and Hilliges, O. (2018), Deep Pictorial Gaze Estimation, *in* 'European Conference on Computer Vision', pp. 741–757. doi: 10.1007/978-3-030-01261-8_44 (page 29).

Parks, D., Borji, A. and Itti, L. (2015), 'Augmented saliency model using automatic 3D head pose detection and learned gaze following in natural scenes', *Vision Research* **116**, 113–126. doi: 10.1016/j.visres.2014.10.027 (page 31).

Pasquale, G., Ciliberto, C., Odone, F., Rosasco, L. and Natale, L. (2015), Teaching iCub to recognize objects using deep Convolutional Neural Networks, *in* 'Workshop on Machine Learning for Interactive Systems', pp. 21–25. Retrieved from `http://proceedings.mlr.press/v43/pasquale15.pdf` (pages 41, 44).

Patacchiola, M. and Cangelosi, A. (2017), 'Head pose estimation in the wild using Convolutional Neural Networks and adaptive gradient methods', *Pattern Recognition* **71**, 132–143. doi: 10.1016/j.patcog.2017.06.009 (pages 86, 88).

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T. and Efros, A. A. (2016), Context Encoders: Feature Learning by Inpainting, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 2536–2544. doi: 10.1109/CVPR.2016.278 (page 84).

Pattacini, U., Nori, F., Natale, L., Metta, G. and Sandini, G. (2010), An experimental evaluation of a novel minimum-jerk Cartesian controller for humanoid robots, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems', pp. 1668–1674. doi: 10.1109/IROS.2010.5650851 (page 64).

Pérez, P., Gangnet, M. and Blake, A. (2003), 'Poisson image editing', *ACM Transactions on Graphics* **22**(3), 313–318. doi: 10.1145/882262.882269 (page 136).

Petit, M., **Fischer, T.** and Demiris, Y. (2016*a*), 'Lifelong Augmentation of Multi-Modal Streaming Autobiographical Memories', *IEEE Transactions on Cognitive and Developmental Systems* **8**(3), 201–213. doi: 10.1109/TAMD.2015.2507439 (page 127).

Petit, M., **Fischer, T.** and Demiris, Y. (2016*b*), Towards the Emergence of Procedural Memories from Lifelong Multi-Modal Streaming Memories for Cognitive Robots, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Machine Learning Methods for High-Level Cognitive Capabilities in Robotics'. Retrieved from http://hdl.handle.net/10044/1/40206 (page 127).

Pinto, N., Cox, D. D. and DiCarlo, J. J. (2008), 'Why is Real-World Visual Object Recognition Hard?', *PLoS Computational Biology* **4**(1), e27. doi: 10.1371/journal.pcbi.0040027 (page 38).

Premack, D. and Woodruff, G. (1978), 'Does the chimpanzee have a theory of mind?', *Behavioral and Brain Sciences* **4**, 515–526. doi: 10.1017/S0140525X00076512 (pages 17, 18, 97).

Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A. Y. (2009), ROS: an open-source Robot Operating System, *in* 'IEEE International Conference on Robotics and Automation Workshop on Open Source Software'. Retrieved from http://www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf (pages 62, 131).

Qureshi, A. W., Apperly, I. A. and Samson, D. (2010), 'Executive function is necessary for perspective selection, not Level-1 visual perspective calculation: Evidence from a dual-task study of adults', *Cognition* **117**(2), 230–236. doi: 10.1016/j.cognition.2010.08.003 (pages 32, 34).

Ramsey, R., Hansen, P., Apperly, I. A. and Samson, D. (2013), 'Seeing It My Way or Your Way: Frontoparietal Brain Areas Sustain Viewpoint-

independent Perspective Selection Processes', *Journal of Cognitive Neuroscience* **25**(5), 670–684. doi: 10.1162/jocn_a_00345 (page 32).

Recasens, A., Khosla, A., Vondrick, C. and Torralba, A. (2015), Where are they looking?, *in* 'Advances in Neural Information Processing Systems', pp. 199–207. doi: 10.1038/scientificamerican0700-38 (pages 28, 75, 76, 126).

Reynolds, J. H., Chelazzi, L. and Desimone, R. (1999), 'Competitive Mechanisms Subserve Attention in Macaque Areas V2 and V4', *Journal of Neuroscience* **19**(5), 1736–1753. doi: 10.1523/JNEUROSCI.19-05-01736.1999 (page 39).

Reynolds, J. H. and Heeger, D. J. (2009), 'The Normalization Model of Attention', *Neuron* **61**(2), 168–185. doi: 10.1016/j.neuron.2009.01.002 (page 39).

Rogez, G. and Schmid, C. (2016), MoCap-guided Data Augmentation for 3D Pose Estimation in the Wild, *in* 'Advances in Neural Information Processing Systems', pp. 3108–3116. arXiv: 1607.02046 (page 81).

Ros, R., Lemaignan, S., Sisbot, E. A., Alami, R., Steinwender, J., Hamann, K. and Warneken, F. (2010), Which One? Grounding the Referent Based on Efficient Human-Robot Interaction, *in* 'International Symposium in Robot and Human Interactive Communication', pp. 570–575. doi: 10.1109/ROMAN.2010.5598719 (pages 24, 26).

Rothenstein, A. L. and Tsotsos, J. K. (2014), 'Attentional Modulation and Selection – An Integrated Approach', *PLoS ONE* **9**(6), e99681. doi: 10.1371/journal.pone.0099681 (page 39).

Roy, D., Hsiao, K. Y. and Mavridis, N. (2004), 'Mental Imagery for a Conversational Robot', *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **34**(3), 1374–1383. doi: 10.1109/TSMCB.2004.823327 (page 24).

Salatas, H. and Flavell, J. H. (1976), 'Perspective taking: The development of two components of knowledge', *Child Development* **47**(1), 103–109. doi: 10.2307/1128288 (page 97).

Sanchez-Fibla, M., Bernardet, U., Wasserman, E., Pelc, T., Mintz, M., Jackson, J. C., Lansink, C., Pennartz, C. and Verschure, P. F. M. J. (2010), 'Allostatic control for robot behavior regulation: a comparative rodent-robot study', *Advances in Complex Systems* **13**(3), 377–403. doi: 10.1142/S0219525910002621 (page 67).

Santiesteban, I., Kaur, S., Bird, G. and Catmur, C. (2017), 'Attentional processes, not implicit mentalizing, mediate performance in a perspective-taking task: Evidence from stimulation of the temporoparietal junction', *NeuroImage* **155**, 305–311. doi: 10.1016/j.neuroimage.2017.04.055 (page 34).

Sarabia, M., Ros, R. and Demiris, Y. (2011), Towards an open-source social middleware for humanoid robots, *in* 'IEEE-RAS International Conference on Humanoid Robots', pp. 670–675. doi: 10.1109/Humanoids.2011.6100883 (page 27).

Schillingmann, L. and Nagai, Y. (2015), Yet Another Gaze Detector: An Embodied Calibration Free System for the iCub Robot, *in* 'IEEE-RAS International Conference on Humanoid Robots', pp. 8–13. doi: 10.1109/HUMANOIDS.2015.7363515 (pages 31, 75).

Schrodt, F., Layher, G., Neumann, H. and Butz, M. V. (2015), 'Embodied learning of a generative neural model for biological motion perception and inference', *Frontiers in Computational Neuroscience* **9**(79), 1–20. doi: 10.3389/fncom.2015.00079 (page 36).

Schurz, M., Kronbichler, M., Weissengruber, S., Surtees, A., Samson, D. and Perner, J. (2015), 'Clarifying the role of theory of mind areas during visual perspective taking: Issues of spontaneity and domain-specificity', *NeuroImage* **117**, 386–396. doi: 10.1016/j.neuroimage.2015.04.031 (pages 33, 128).

Schwarzkopf, S., Büchner, S. J., Hölscher, C. and Konieczny, L. (2017), 'Perspective tracking in the real world: Gaze angle analysis in a collaborative wayfinding task', *Spatial Cognition & Computation* **17**(1-2), 143–162. doi: 10.1080/13875868.2016.1226841 (page 17).

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W. and Webb, R. (2017), Learning from Simulated and Unsupervised Images through Adversarial Training, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 2242–2251. doi: 10.1109/CVPR.2017.241 (pages 31, 93).

Simonyan, K. and Zisserman, A. (2015), Very Deep Convolutional Networks for Large-Scale Image Recognition, *in* 'International Conference on Learning Representations'. arXiv: 1409.1556 (pages 87, 88).

Sisbot, E. A., Marin-Urias, L. F., Alami, R. and Sim, T. (2007), 'A Human Aware Mobile Robot Motion Planner', *IEEE Transactions on Robotics* **23**(5), 874–883. doi: 10.1109/TRO.2007.904911 (page 26).

Smith, B. A., Yin, Q., Feiner, S. K. and Nayar, S. K. (2013), Gaze Locking: Passive Eye Contact Detection for Human-Object Interaction, *in* 'ACM Symposium on User Interface Software and Technology', pp. 271–280. doi: 10.1145/2501988.2501994 (pages 30, 76).

Steels, L. and Loetzsch, M. (2009), Perspective Alignment in Spatial Language, *in* 'Spatial Language and Dialogue', Oxford University Press, pp. 70–88. doi: 10.1093/acprof:oso/9780199554201.003.0006 (page 24).

Sugano, Y., Matsushita, Y. and Sato, Y. (2014), Learning-by-Synthesis for Appearance-based 3D Gaze Estimation, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 1821–1828. doi: 10.1109/CVPR.2014.235 (pages 30, 76, 82, 83, 84, 90, 91).

Sünderhauf, N., Brock, O., Scheirer, W., Hadsell, R., Fox, D., Leitner, J., Upcroft, B., Abbeel, P., Burgard, W., Milford, M. and Corke, P. (2018), 'The limits and potentials of deep learning for robotics', *The International Journal of Robotics Research* **37**(4-5), 405–420. doi: 10.1177/0278364918770733 (page 124).

Surtees, A. D. R., Apperly, I. A. and Samson, D. (2016), 'I've got your number: Spontaneous perspective-taking in an interactive task', *Cognition* **150**, 43–52. doi: 10.1016/j.cognition.2016.01.014 (pages 33, 34).

Surtees, A., Apperly, I. A. and Samson, D. (2013*a*), 'Similarities and differences in visual and spatial perspective-taking processes', *Cognition* **129**(2), 426–438. doi: 10.1016/j.cognition.2013.06.008 (page 35).

Surtees, A., Apperly, I. A. and Samson, D. (2013*b*), 'The use of embodied self-rotation for visual and spatial perspective-taking', *Frontiers in Human Neuroscience* **7**(698), 1–12. doi: 10.3389/fnhum.2013.00698 (pages 17, 33, 35, 36, 97, 98, 109).

Sutin, A. R. and Robins, R. W. (2008), 'When the "I" looks at the "Me": Autobiographical memory, visual perspective, and the self', *Consciousness and Cognition* **17**(4), 1386–1397. doi: 10.1016/j.concog.2008.09.001 (page 127).

Tikhanoff, V., Pattacini, U., Natale, L. and Metta, G. (2015), Exploring affordances and tool use on the iCub, *in* 'IEEE-RAS International Conference on Humanoid Robots', pp. 130–137. doi: 10.1109/HUMANOIDS.2013.7029967 (page 64).

Todd, A. R., Cameron, C. D. and Simpson, A. J. (2017), 'Dissociating processes underlying level-1 visual perspective taking in adults', *Cognition* **159**, 97–101. doi: 10.1016/j.cognition.2016.11.010 (pages 32, 34).

Tosato, D., Spera, M., Cristani, M. and Murino, V. (2013), 'Characterizing Humans on Riemannian Manifolds', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(8), 1972–1984. doi: 10.1109/TPAMI.2012.263 (page 76).

Trafton, J. G., Cassimatis, N. L., Bugajska, M. D., Brock, D. P., Mintz, F. E. and Schultz, A. C. (2005), 'Enabling Effective Human-Robot Interaction Using Perspective-Taking in Robots', *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **35**(4), 460–470. doi: 10.1109/TSMCA.2005.850592 (pages 17, 24, 26).

Trafton, J. G., Schultz, A. C., Bugajska, M. D. and Mintz, F. (2005), Perspective-taking with Robots: Experiments and models, *in* 'International Workshop on Robots and Human Interactive Communication Perspective-taking', pp. 580–584. doi: 10.1109/ROMAN.2005.1513842 (page 17).

Tsotsos, J. K. (1990), 'Analyzing vision at the complexity level', *Behavioral and Brain Sciences* **13**(3), 423–445. doi: 10.1017/S0140525X00079577 (page 39).

Tsotsos, J. K. (2011), *A Computational Perspective on Visual Attention*, MIT Press, Cambridge, MA, USA. (page 39).

Tsotsos, J. K., Culhane, S. M., Kei Wai, W. Y., Lai, Y., Davis, N. and Nuflo, F. (1995), 'Modeling visual attention via selective tuning', *Artificial Intelligence* **78**(1-2), 507–545. doi: 10.1016/0004-3702(95)00025-9 (page 39).

Valenti, R., Sebe, N. and Gevers, T. (2012), 'Combining Head Pose and Eye Location Information for Gaze Estimation', *IEEE Transactions on Image Processing* **21**(2), 802–15. doi: 10.1109/TIP.2011.2162740 (page 75).

Vander Heyden, K. M., Huizinga, M., Raijmakers, M. E. and Jolles, J. (2017), 'Children's representations of another person's spatial perspective: Different strategies for different viewpoints?', *Journal of Experimental Child Psychology* **153**, 57–73. doi: 10.1016/j.jecp.2016.09.001 (page 35).

Vasudevan, A. B., Dai, D. and Van Gool, L. (2018), Object Referring in Videos with Language and Human Gaze, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 4129–4138. doi: 10.1109/CVPR.2018.00434 (page 32).

Vezzani, G., Pattacini, U. and Natale, L. (2017), A Grasping Approach Based on Superquadric Models, *in* 'IEEE International Conference on Robotics and Automation', pp. 1579–1586. doi: 10.1109/ICRA.2017.7989187 (page 45).

Viola, P. and Jones, M. (2001), Rapid object detection using a boosted cascade of simple features, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. I–511–I–518. doi: 10.1109/CVPR.2001.990517 (page 66).

Vouloutsi, V., Grechuta, K., Lallée, S. and Verschure, P. F. M. J. (2014), The Influence of Behavioral Complexity on Robot Perception, *in* 'Conference on Biomimetic and Biohybrid Systems', pp. 332–343. doi: 10.1007/978-3-319-09435-9_29 (page 67).

Wang, H., Callaghan, E., Gooding-Williams, G., McAllister, C. and Kessler, K. (2016), 'Rhythm makes the world go round: An MEG-TMS study on the role of right TPJ theta oscillations in embodied perspective taking', *Cortex* **75**, 68–81. doi: 10.1016/j.cortex.2015.11.011 (pages 36, 55, 98).

Warnier, M., Guitton, J., Lemaignan, S. and Alami, R. (2012), When the robot puts itself in your shoes. Managing and exploiting human and robot beliefs., *in* 'IEEE International Symposium on Robot and Human Interactive Communication', pp. 948–954. doi: 10.1109/ROMAN.2012.6343872 (pages 24, 26).

Watanabe, M. (2016), 'Developmental changes in the embodied self of spatial perspective taking', *British Journal of Developmental Psychology* **34**(2), 212–225. doi: 10.1111/bjdp.12126 (pages 36, 98).

Wilczkowiak, M., Brostow, G. J., Tordoff, B. and Cipolla, R. (2005), Hole Filling Through Photomontage, *in* 'British Machine Vision Conference', pp. 492–501. Retrieved from `http://www.bmva.org/bmvc/2005/papers/55/paper.pdf` (page 84).

Winfield, A. F. T. (2018), 'Experiments in Artificial Theory of Mind: From Safety to Story-Telling', *Frontiers in Robotics and AI* **5**(75), 1–13. doi: 10.3389/frobt.2018.00075 (page 25).

Wood, E., Baltrušaitis, T., Morency, L.-P., Robinson, P. and Bulling, A. (2016), Learning an appearance-based gaze estimator from one million synthesised images, *in* 'ACM Symposium on Eye Tracking Research & Applications', pp. 131–138. doi: 10.1145/2857491.2857492 (pages 31, 75, 92).

Wood, E., Baltrušaitis, T., Zhang, X., Sugano, Y., Robinson, P. and Bulling, A. (2015), Rendering of Eyes for Eye-Shape Registration and Gaze Estimation, *in* 'IEEE International Conference on Computer Vision', pp. 3756–3764. doi: 10.1109/ICCV.2015.428 (pages 31, 76).

Wood, L. J., Robins, B., Lakatos, G., Syrdal, D. S., Zaraki, A. and Dautenhahn, K. (2018), Piloting Scenarios for Children with Autism to Learn About Visual Perspective Taking, *in* 'Towards Autonomous Robotic Systems Conference', pp. 260–270. doi: 10.1007/978-3-319-96728-8 (page 24).

Wu, J., Xue, T., Lim, J. J., Tian, Y., Tenenbaum, J. B., Torralba, A. and Freeman, W. T. (2016), Single Image 3D Interpreter Network, *in* 'European Conference on Computer Vision', pp. 365–382. doi: 10.1007/978-3-319-46466-4_22 (page 125).

Yan, X., Yang, J., Yumer, E., Guo, Y. and Lee, H. (2016), Perspective Transformer Nets: Learning Single-View 3D Object Reconstruction without 3D Supervision, *in* 'Advances in Neural Information Processing Systems', pp. 1696–1704. arXiv: 1612.00814 (page 125).

Yaniv, I. and Shatz, M. (1990), 'Heuristics of Reasoning and Analogy in Children's Visual Perspective Taking', *Child Development* **61**(5), 1491–1501. doi: 10.1111/j.1467-8624.1990.tb02877.x (pages 32, 49, 55).

Yeh, R. A., Chen, C., Lim, T. Y., Schwing, A. G., Hasegawa-Johnson, M. and Do, M. N. (2017), Semantic Image Inpainting with Deep Generative Models, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 6882–6890. doi: 10.1109/CVPR.2017.728 (pages 84, 85, 137).

Yu, A. B. and Zacks, J. M. (2017), 'Transformations and representations supporting spatial perspective taking', *Spatial Cognition and Computation* **17**(4), 304–337. doi: 10.1080/13875868.2017.1322596 (page 98).

Zambelli, M. and Demiris, Y. (2017), 'Online Multimodal Ensemble Learning using Self-learnt Sensorimotor Representations', *IEEE Transactions on Cognitive and Developmental Systems* **9**(2), 113–126. doi: 10.1109/TCDS.2016.2624705 (page 66).

Zambelli, M., **Fischer, T.**, Petit, M., Chang, H. J., Cully, A. and Demiris, Y. (2016), Towards Anchoring Self-Learned Representations to Those of Other Agents, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Bio-inspired Social Robot Learning in Home Scenarios'. Retrieved from http://hdl.handle.net/10044/1/40970 (pages 42, 63, 66).

Zhang, K., Zhang, Z., Li, Z. and Qiao, Y. (2016), 'Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks', *IEEE Signal Processing Letters* **23**(10), 1499–1503. doi: 10.1109/LSP.2016.2603342 (pages 86, 90).

Zhang, X., Sugano, Y., Fritz, M. and Bulling, A. (2015), Appearance-Based Gaze Estimation in the Wild, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 4511–4520. doi: 10.1109/CVPR.2015.7299081 (pages 29, 30, 75, 76, 81, 82, 83, 84, 86, 90, 91, 92, 93).

Zhang, X., Sugano, Y., Fritz, M. and Bulling, A. (2017), It's Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation, *in* 'IEEE Conference on Computer Vision and Pattern Recognition Workshops', pp. 2299–2308. doi: 10.1109/CVPRW.2017.284 (pages 29, 91, 92).

Zhu, J.-Y., Park, T., Isola, P. and Efros, A. A. (2017), Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks, *in* 'IEEE International Conference on Computer Vision', pp. 2242–2251. doi: 10.1109/ICCV.2017.244 (page 135).