Abstract

Robot Self-Modeling

Justin Wildrick Hart

2014

Traditionally, models of a robot's kinematics and sensors have been provided by designers through manual processes. Such models are used for sensorimotor tasks, such as manipulation and stereo vision. However, these techniques often yield static models based on one-time calibrations or ideal engineering drawings; models that often fail to represent the actual hardware, or in which individual unimodal models, such as those describing kinematics and vision, may disagree with each other.

Humans, on the other hand, are not so limited. One of the earliest forms of self-knowledge learned during infancy is knowledge of the body and senses. Infants learn about their bodies and senses through the experience of using them in conjunction with each other. Inspired by this early form of self-awareness, the research presented in this thesis attempts to enable robots to learn unified models of themselves through data sampled during operation. In the presented experiments, an upper torso humanoid robot, Nico, creates a highly-accurate self-representation through data sampled by its sensors while it operates. The power of this model is demonstrated through a novel robot vision task in which the robot infers the visual perspective representing reflections in a mirror by watching its own motion reflected therein.

In order to construct this self-model, the robot first infers the kinematic parameters describing its arm. This is first demonstrated using an external motion capture system, then implemented in the robot's stereo vision system. In a process inspired by infant development, the robot then mutually refines its kinematic and stereo vi-

sion calibrations, using its kinematic structure as the invariant against which the system is calibrated. The product of this procedure is a very precise mutual calibration between these two, traditionally separate, models, producing a single, unified self-model.

The robot then uses this self-model to perform a unique vision task. Knowledge of its body and senses enable the robot to infer the position of a mirror placed in its environment. From this, an estimate of the visual perspective describing reflections in the mirror is computed, which is subsequently refined over the expected position of images of the robot's end-effector as reflected in the mirror, and their real-world, imaged counterparts. The computed visual perspective enables the robot to use the mirror as an instrument for spacial reasoning, by viewing the world from its perspective. This test utilizes knowledge that the robot has inferred about itself through experience, and approximates tests of mirror use that are used as a benchmark of self-awareness in human infants and animals.

Robot Self-Modeling

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Justin Wildrick Hart

Dissertation Director: Brian Scassellati

December 2014

Contents

List of Figures

Acknowledgements

When I came for my first visit to Yale, Brian Scassellati and I discussed the sorts of ambitious robotics projects that we wanted to pursue together. The ideas that we went on to explore together have shaped my research career and led to the questions that I intend to pursue as I move forward as an investigator. What eventually became this thesis, began as Scaz and I sitting in his living room discussing projects on the development of self-awareness. He has helped me to develop both as a researcher and a professional, and has been a great mentor and friend.

My committee is a truly remarkable group of people. Chad Jenkins has been a friendly face at conferences and a source of guidance ever since we first played air hockey at ICDL in 2008. I met Aaron Dollar when he was a postdoc at NEMS in 2008, and drove to NEMS with him in 2010, with some of his first students at Yale. Steven Zucker doesn't merely use mathematics to describe phenomena, he uses it as a lens through which to see and understand the entire world. I like to think that he has helped me to see the world in the same way.

The Social Robotics Laboratory has brought me an amazing group of colleagues. The students who welcomed me were Chris Crick, Marek Doniec, Kevin Gold, Elizabeth Kim, and Frederick Shic. The first thing that anybody will tell you about Chris is that he does interesting things. If you're reading this and have done a few laps over the Yale campus in a small plane, there's a good chance that you did so with Chris. Marek helped me get my feet wet programming Nico and collaborated with me on the design for Nico's hand. For years, I had excited students who had seen a video of one of Kevin's experiments come up to me saying, "I got the ball." I have since disassembled the ball. Eli and I were ridiculously productive working together. Scaz considered having us collaborate on our theses, had they not gone in such different directions. Fred is now a professor at Yale. When he sees something in my blind spot and says, "do this," I benefit from doing whatever it is he tells me to do.

The next couple of years brought Dan Leyzberg, Henny Admoni, and Bradley Hayes, followed by Corina Grigore, Alex Litoiu, Aditi Ramachandran, and Jean Zheng as students. It also brought Cindy Bethel and David Feil-Seifer as postdocs. Dan took the photo of me and Nico chosen by almost every media outlet to represent my work. I've known Henny since she was an undergraduate at Wesleyan, doing research with our lab. She has the most disciplined and organized approach to graduate school of anyone that I've met. Brad has helped me do everything that I needed to be at Yale to accomplish, while I have been thousands of miles away at my postdoc. Prior to that, he taught me how to play hockey. I have worn the suit that Corina helped me pick out to every interview and talk since. Alex and Aditi picked up what little machining I had time to teach them quickly, helping to machine Dragonbot parts during my final summer at Yale. Jean let me borrow the LaTeXtemplate that I used to write this thesis. If there are formatting errors, I know that Yale has accepted a thesis with similar errors before. I clearly remember the first time I met Cindy, and thinking it was cool that she studied the same types of Human-Robot Interaction that we study on humanoids, except using search and rescue robots. David Feil-Seifer was one of the greatest movie night hosts to ever grace our lab and helped me to get my feet wet with ROS, which I now use extensively. Larissa Hall came on in my last year at Yale to help us with the new duties brought on by managing our NSF Expedition. She's managed, with almost no time to prepare, to help me make brilliant visuals on every occasion that I've asked.

I would also like to thank the many undergraduate students who came through the lab during my time here, especially those who I've had the pleasure of collaborating with and mentoring, including, but not limited to: Eleanor Avrunin, Wilma Bainbridge, Kenny Castenada, Ashley Douglas, Gabriel Fernandez, David Golub, Jason Kim, Justin Kosslyn, Graham Radman, Elaine Short, Sam Spaulding, and, Michelle Vu. Also, students from the other robotics labs at Yale deserve my thanks, but in particular Joe Belter, who can design and fabricate anything, and has helped me do so on many an occasion.

A few other professors and staff deserve a mention. Drew McDermott and I have always managed to have engaging conversations about artificial intelligence. Vladimir Rokhlin just knows how to fix things. Julie Dorsey, Joan Feigenbaum, and Holly Rushmeyer have always taken the time to give me words of encouragement. Nick Bernardo and Dave Johnson taught me almost everything I know about metalwork, and have helped me to fabricate absolutely incredible machines.

The "Thursday Night Dinner Group" are the very first friends I made at Yale, and have been with me ever since. Stephen Eckel and Rachel Kramer, Chris Milan and Patricia Rios Milan, Tom Morgan and Katie Jozwicki, Karen Paczkowski and Christopher Thissen, Nicholas Ruozzi and Blair Kenney, Shannon Stewart and Paul Vierthaler, Mychal Thigpen and Tracelyn Hairston, and Edwin Van Bibber-Orr. Andrea Januta, Aaron Johnson, Jared Harwayne-Gidansky, Derek Park, and Randy Stein also deserve thanks for keeping me company on many a late night in the lab. I'd like to thank the many other friends outside of work who shaped my time at Yale, and the many friends back home who have given me constant and steadfast encouragement.

My postdoc advisor, Elizabeth Croft, and my new labmates at the University of British Columbia also deserve credit for their support as I have balanced my duties in the lab with preparations for my defense and graduation from Yale.

Finally my parents Jay and Jackie Hart, and my brother Chad Hart, as well as an extended family too large to list here, have always been there to encourage and help me. I could not have done this without them.

I could not possibly list here everyone who I would like to thank for their help and support - colleagues from other universities, friends from my time at Yale, and the community I left back in Virginia to pursue this goal. You have my most sincere gratitude.

Support for this work has been generously provided by the National Science Foundation under contracts #0534610, #0835767, #0968538, #1117801, CAREER award #0238334, and Expedition in Computing #1139078, as well as grants from Microsoft Research, the Sloan Foundation, the DARPA Computer Science Futures II Program, the Air Force Office of Scientific Research, and the National Geospatial Intelligence Agency. Software grants were also provided by QNX Software Systems Ltd. Hardware grants were provided by Ugobe Inc.

Chapter 1

Introduction

In robotics, the need to reason about a machine's physical structure and sensors is inescapable. These provide the robot with the means for it to interact with its environment as well as the objects and the other agents in it. Traditionally, however, the thinking about the robot's senses and structure is done by scientists and engineers, either through calibration processes or in the form of hand-coded models that are developed when the machine is designed. These models are then used in black-box subprograms, vision and inverse kinematic routines that are not generally dealt with directly by the robot's cognitive model. Though robots have been programmed to reason and learn about the tasks that they perform, they rely on their designers to do all of the thinking about their physical manifestations. They learn about the world that they operate in, but know nothing about themselves.

People, on the other hand, learn about their physical and sensory capabilities through first-hand experience. The humans that many of these robots attempt to emulate do not start out with hand-calibrated kinematic and visual models. To infants learning to grasp objects, the kinematic and sensory capabilities of their bodies are something to be learned. While robots are provided with this information a priori by skilled engineers, humans learn about their bodies and senses through the experience of using them in concert with each other. Through their senses, infants perceive the effects of their actions, the form and structure of their bodies, and the objects, environments, and other agents that they are able to interact with. Their bodies allow infants to act upon their selves and their surroundings, in turn providing stimuli to their senses. Infants waving their arms in front of their faces or reaching toward objects and people learn about their eyes and arms alongside the things that they interact with. This learning process has been a matter of extensive study by developmental psychologists. The understanding of the body, the senses, and their relationship to each other that is learned through this process is one of the earliest forms of self-awareness to develop during infancy. Developmental theorists refer to this knowledge of the body and senses as knowledge of the ecological self [3]. The ecological self is the combination of the physical self as an object that can be controlled by one's mind and the senses that take in information about this body and other physical objects.

1.1 Self-awareness in humans

Despite the seeming disadvantage of not starting with knowledge of the body and senses, infants have a flexibility that exists in no robotic system. Learning about the body alongside the senses, infants flexibly learn a wider repertoire of behaviors than any robot ever has. Rather than the, still limited, library of motor tasks such as grasps and pushes that are learned by modern robots, an infant can learn to grasp and kick and throw, and when presented with a novel situation they can reason about new physical maneuvers that allow them to navigate that situation. They can reason about pushing with the back of the hand while holding a drink or stretching to reach a screw in a hard-to-reach spot while working on an automobile. The infant mind and body eventually become capable of learning or even creating entirely new motions and skills, such as the bicycle kick in soccer, or beautiful acrobatic displays, such as those in gymnastic competitions.

The seemingly random behaviors of infants - kicking their legs and putting their fists into their mouths - eventually become these complex acts, and help to tune models of their ecological selves [3]. One consequence of learning about the ecological self through the interaction of the body and the senses with each other is that they are calibrated to each other [3]. The ecological self is a cohesive model of the body and the senses, learned by using and witnessing them with respect to each other.

The unified nature of the ecological self allows sensory information to be combined and interpreted as a whole. The centralized impression of the body, its pose, and the mapping of senses along this structure, allowing for registration of the senses with respect to this structure and its current pose is often referred to as the body schema [4]. The body schema allows for the senses to be interpreted with respect to each other by way of this central model. For example, the map of the tactile sense can be interpreted with respect to the pose of the body, allowing one to perform such acts as reaching in the dark for a light switch or fumbling in a drawer for a pair of scissors. When one does so, they know where their hand is with respect to the pose of their body, where the tactile sensation is felt on their hand, and, consequently, know where the object is in space. Moreover, through the combined interpretation availed by the body schema, they perceive the pose of the tactile stimulus with respect to their visual field. As such, in this example they have a good idea of where they would see their hand under the tabletop if the desk were made of glass.

It is known that children are born with part of this sense of the ecological self intact. Rochat and Hespos [5] demonstrated that the rooting reflex is not triggered in newborns when they touch themselves on the cheek, suggesting that they are able to differentiate between their own touch and that of others. A number of hypotheses not rooted in knowledge of their kinematic and tactile structures could be supported by this evidence, such as the possibility that this differentiation is performed through interpreting the correlation in the onset of the tactile sensation as experienced at the fingertip and on the cheek. In another experiment, however, Rochat, Blass, and Hoffmeyer [6] demonstrated that neonates open their mouths in preparation to receive their fists when exhibiting fist-sucking behavior. The observation that an infant is aware that they are about to insert their fist into their mouth, which cannot be directly observed in the visual field, is strongly suggestive that infants use knowledge of the relationship between the kinematics of the arm and mouth as a route to experiencing the tactile sensation of sucking on their fists. This is in contrast with a prior hypothesis, that infants experience tactile pleasure once the hand is in the mouth, and so choose to leave it their after incidentally inserting the hand into the mouth. Together, these experiments support the hypothesis that neonates are born with a degree of knowledge of their bodies and senses.

This knowledge of the ecological self and the body schema adapts as the body changes. As children grow, their knowledge of their self grows with them. They continue to walk despite growing longer legs, can still reach objects with their longer arms, and continue to perceive their world with aging eyes whose optical properties have changed over time. This adaptation continues throughout our lives and on both long any short time scales. A person can identify when they are injured and change their strategy for interacting with the world; not using a broken arm, or walking differently to accommodate a strained muscle. When they heal, they adapt their behaviors again to their physically-well selves. When they use tools, the body schema adapts as well, incorporating the tool into knowledge of the body. In an experiment demonstrating this short-term adaptation, it was shown that the tactile experience of touching an object with an L-shaped tool, such as a hex key, is experienced at the tip of the tool, rather than only as the tool shifting in the hand [7].

The merging of visual and tactile information, especially as it relates to self-other discrimination and body localization has also been studied extensively by psychologists. In the rubber-hand illusion [8], study participants place their hand under an obstruction on a table with a rubber hand placed on top. A stimulus is presented to study participants in the form of simultaneous strokes of paint brushes both on each participant's real hand, which is hidden from view, and the rubber hand, which is in their visual field. These participants have been demonstrated both to "feel" the stimulus as displaced onto the rubber hand and to experience a an altered sense of proprioception. Perhaps the most widely-known series of experiments demonstrating sensorimotor adaptation are those involving participants wearing glasses that shift or invert the visual field. Pioneered by George Stratton [9] in 1897, these experiments have demonstrated sensorimotor adaptation in a number of settings[10, 11, 12] (note that [13] provides a good overview).

More recently, Volcic, Fantoni, Caudek, Assad, and Domini [14] demonstrated rapid adaptation of the perceptual system to changes in the perceived length of the arm in humans. In this study, the experimenters manipulated study participants' perceptions of the length of their arms. They found that scaling perceptions of the arm's length also scaled perceptions of distances perceived visually and through tactile stimuli. These findings support the hypothesis that the perception of distance is rooted in an understanding of the size of the body, is tuned to the range at which a person is able to perform grasps, and that the neurological mechanisms for these perceptions are able to rapidly recalibrate to each other [14]. This manipulation also provides support for ideas regarding the body schema and the ecological self by demonstrating that changes to the knowledge of the structure of one's body influences one's spatial perception.

The concept of the ecological self casts the knowledge that people and animals learn about their bodies and senses into a framework of self-awareness. The development of self-awareness in humans and animals has been studied extensively by psychologists. Bertenthal and Fischer sought to document the development of self-awareness from $6 - 24$ months, relating the development of self-recognition to the concept of object permanence [15]. Their study built upon the earlier work of Gordon Gallup [16], in which he developed the so-called "mirror-test," which has become the classical test of self-awareness in humans and animals.

The mirror test [16] and similar tests (Povinelli [17] provides a good discussion) attempt to evaluate whether an animal is able to recognize itself in a mirror. In these tests, a mirror is introduced into an animal's enclosure. The animal is given time to acclimate to and learn about the mirror. The animal is then discreetly marked with an odorless, non-tactile dye. If upon encountering the mirror the animal produces a self-directed behavior, such as inspecting the mark on their own forehead, then the animal is considered to be self-aware. In doing so, the test identifies whether the animal has sufficient knowledge of their body to recognize their own appearance and to identify that this appearance has changed.

Bertenthal and Fischer's [15] study builds upon Gallup's [16] and several other studies of self-awareness [18, 19, 20, 21, 22, 17, 23] to construct a timeline of the development of these capabilities in infants. To Gallup's test, Bertenthal and Fischer add tasks related to spatial reasoning. In one test, the "hat task," a special vest holds a hat over an infant study participant's head to observe whether they look up to investigate the hat. Bertenthal and Fischer found that infants are able to perform simple spatial reasoning tasks involving mirrors at as early as 8 months, well prior to passing the mirror test, which occurs at 18 months.

Similarly, studies of animal cognition involving spatial reasoning tasks with mirrors suggest that the ability to use a mirror for spatial reasoning is a more primitive skill than that tested in the mirror test. Examples of such tests include ones in which animals are tasked with grasping food pellets which are only visible as reflected in mirrors. The pellets are situated in places where these animals can fit their arms (such as through holes in a sheet of plywood), but cannot fit their heads [24, 25]. These tests identify whether animals are capable of making the spatial inferences necessary to perform tasks using mirrors. These results, combined with the developmental timeline established by Bertenthal and Fischer [15], suggest that spatial reasoning using mirrors may not only be a more-primitive skill, but also a precursor to the mirror self-recognition capability tested in the mirror test [16].

The self-monitoring processes that humans possess, learning and adapting their knowledge of their bodies and senses throughout their lives, provides them with powerful capabilities that are not present in robots. They adapt to changes in their bodies and to the information provided to them through their senses. They can identify when they are hurt, change their behavior accordingly, and seek help as needed. They can use this flexibility in clever ways by adapting to tool use, learning new skills, and reasoning about unforeseen situations. This primitive form of self-awareness, knowledge of body and the senses, gives humans a flexibility that no existing machine possesses.

1.2 What is to be gained by emulating this process in robots?

What if a robot could reason about its body and senses in the same way that a human can? What if robots could learn and adapt their self-representations to compensate for inaccuracies, or to adapt to changes as they sustain damage or simply experience wear and tear through prolonged use? What happens when models of the body and the senses are tightly calibrated to each other, such that reasoning can be done between senses with respect to the robot's body? What if robots learned about their ecological selves in the ways that humans do, and reasoned about their senses through a centralized body schema?

The work in this thesis concentrates on early forms of self-awareness that develop during infancy regarding awareness of the body and senses. Bringing together the observations of Rochat [3] and Bertenthal and Fischer [15] we arrive at a timeline that indicates that infants start are born with a primitive sense of how their bodies and senses work that gives rise to motor and cognitive capabilities.[1] Psychologists studying both animal behavior and human development use variations of the mirror test to provide perspective onto self-awareness competency. Though Bertenthal and Fischer were attempting to place mirror reasoning capabilities in context with object permanence, this thesis uses their developmental timeline to identify cognitive milestones towards self-aware systems.

Others have investigated matters related to self-awareness in artificially intelligent systems, focusing on different problems. The study of metacognition seeks to understand introspection in terms of higher-level processes that monitor and reason

[1] As Rochat describes it it, the "ecological self," the physical self learned through the senses, and "self-efficacy," what can be accomplished with one's body [3].

about lower-level processes [26]. Theory of mind describes the ability to attribute mental states to both oneself and to others, and can form the basis for social reasoning and communication [27]. In robotics and artificial intelligence, numerous systems have been developed to help understand and replicate these processes. Systems have been developed to attempt to understand mirror self-recognition [28, 29, 30, 31], the development of aspects of the ecological self [32], the body schema [33, 34, 35, 36], and social reasoning processes involving self-reflection [37, 38, 39]. While robots running such systems cannot definitively prove that a specific hypothesis explains a natural phenomenon, they can be used as existence proofs to demonstrate the plausibility of a hypothesis, and can be used to further the technological state of the art by providing new capabilities to robots and other systems.

This dissertation focuses on learning about the body and senses as a unified self-model. Studying self-knowledge in a framework of understanding the body and the senses makes calibration a central theme in this thesis. The problem of calibrating a robot's kinematics to its vision system arises in cases where the two must be used together. In systems where representations of a machine's kinematic and perceptual hardware are developed separately; camera model parameters coming from calibration processes, with kinematic models provided by manual processes, there may be disagreements between these separate models. These separate models may not even be described in the same bases - with scales, origins, and orientations entirely in disagreement. In their calibration paper, Pradeep, Konolige, and Berger [40] provide a good overview of work on this topic.

In contrast to humans, a traditionally-designed robot has no way to modify these self-representations. The kinematic models and camera calibrations that these robots are provided with are handled by subprograms that perform tasks such as stereo vision and motion planning. With no means to self-calibrate these models, such robots may be able to identify that an attempted action has failed, but have no way to modify faulty calibration data or an underling faulty model that led to the failure. Such robots may incorrectly attribute failures to incorrect strategies for performing tasks such as manipulation tasks, when these failures are not due to faulty strategies but to inaccuracies in the underlying visual and kinematic models that they operate over. A truly robust model of a robot's sensors and kinematics must not be so limited, as these calibrations represent the framework upon which all sensorimotor behavior is built.

A robot which is able to refine and correct these models during operation, however, has the potential to arrive at both an accurate self-representation and a matching correct policy for the task, rule, or behavior that it is trying to accomplish. This is because the policy is developed with respect to an accurately calibrated self-model. The system developed in this thesis is a step in that direction, allowing the robot to develop and refine a self-model based on its observations of its own actions.

Multiple areas of current research may be impacted by the self-modeling methods developed in this thesis. For instance, systems developed for fault detection, diagnosis, and recovery which use model-based diagnosis [41, 42, 43] monitor sensor data for compliance with a model of the correct operation of a device. In the case of the system developed in this dissertation, the system is updated through retraining in order to accommodate changes to the system.[2] Another possibility is to develop a classifier based on differences between inferred and optimal models for the purposes of fault detection. Similarly, the methods developed in this thesis could also be used to assure that machines retain accurate calibrations throughout their lifespans by refining calibrations during operation. The self-representations used in modern machines are often calibrated by expert roboticists, but self-calibrating machinery could

[2] This is demonstrated through the system's adaptation to a tool mounted in its gripper.

also be an enabling technology as robotics attempts to enter more domains where operators will not be expected to be experts. There is also current interest in the use of tactile sensors in robotics [44]. Interpreting their input with respect to a body-schema could enable systems to identify where touched objects are in space. The focus of this dissertation is on a kinesthetic-visual self-model, emulating one of the earliest, most-primitive forms of self-awareness possessed by infants. It is our hope that this will serve as a starting-point to the study of many forms of self-knowledge through the use of and implemented into robotic systems.

1.3 The kinesthetic-visual self-model

Inspired by the developmental process of self-discovery that infants experience, the goal of robot self-modeling is to build robots which learn about themselves - their hardware, sensors, and capabilities - through data sampled during operation. The intention of self-modeling is to replace the models that are provided to robots through manual processes with methods that allow robots to learn these models continually and online.

Self-modeling provides us with a concrete milestone toward the ultimate goal of constructing self-aware robotic systems. The inspiration for this thesis came partly from discussions of what it would take to construct a system capable of passing the commonly-accepted test of self-awareness, the "mirror test." In Chapter 2, we propose an architecture which, upon completion, may allow for a robot to pass this test.

In this thesis, a humanoid robot infers a unified model of its body and senses, combining kinematic and sensory aspects that are traditionally modeled separately. This model will be referred to as the robot's kinesthetic-visual self-model, or simply, "self-model." This self-model captures the information about the robot's kinematics and camera calibration that is necessary for sensorimotor tasks such as motion planning and stereo reconstruction. Conceptually, any robot with sensors that interact with its motor state can run a form of the self-modeling outlined in this thesis. For the purposes of this body of work, self-modeling algorithms were developed on a humanoid upper-torso named Nico. Nico has 23 degrees of freedom and is designed to match the form and kinematic structure of an 18-month-old human child at the fiftieth percentile [45]. The robot's hardware will be further described in Chapter 3.

Using a kinematic inference technique developed in Chapter 6, in Chapter 7 the robot will determine how its arm moves by watching the arm's motion in the visual field. Because the robot derives the kinematic representation of its arm from data sampled by its vision system, the estimate it obtains is represented in the same mathematical basis as that in which it performs stereo vision. More simply-put, the model which describes the motion of the robot's arm will be stated in a manner that is compatible with the description that the robot uses to see, allowing the robot to make inferences between the two. The robot will arrive at a self-model that combines the kinematics of its arm and the data perceived by its vision system. This self-model is analogous to the body-schema [4]. The tight coupling between the visual and kinematic systems will produce a model that is predictive of the position of the hand in the visual field, both in 2D and in 3D. Taking advantage of this will allow us to develop a process of simultaneous refinement of the kinematic and visual parameters of the self-model which is analogous to the learning process by which the ecological self is discovered by infants, as theorized by Rochat in [3], in Chapter 8. In Chapter 9, we will demonstrate the power of the self-model to reason about objects in the robot's environment. The robot will use its self-model to infer the visual perspective that produces the images that the robot sees reflected in a mirror, enabling it to accurately determine the position of objects reflected therein through the use of its stereo vision system. This is analogous to the sorts of spatial reasoning tasks that appear to be a precursor to mirror self-recognition.

1.4 Summary

In typical practice, when a robot is designed and constructed, everything that it knows about itself is provided by engineers. In a sense, robots often learn and reason about their environment and objects in that environment, but all of the thinking about the robot itself has been done by engineers ahead of time. This stands in contrast to people and many animals, which learn about their selves through first-hand experience, using this knowledge of their selves to reason about the world. As a result, humans and animals possess a number of capabilities that are beyond the reach of modern robotics technologies. This dissertation seeks to lay a groundwork for constructing systems which learn and reason about themselves through experience.

The self-modeling process detailed in this dissertation enables a humanoid robot to develop and maintain a high degree of accuracy in the calibration of its self-representations with respect to the state of the art, and gives rise to unique capabilities. Robotic self-modeling is a technique by which robots learn about their hardware and sensors through the first-hand experience of using them in concert with each other. It is a starting point upon which research into self-awareness in robotic systems can be built.

This dissertation will discuss the details of the construction and evaluation of a software suite with self-modeling capabilities for a humanoid robot. The remainder of this document will proceed as follows:

- Chapter 2, The mirror-test as a target for self-aware systems

The mirror-test [16] has become the de facto standard by which an animal is judged to be self-aware. Therefore, it provides both a recognizable milestone in the development of self-aware artificially-intelligent systems and an interesting target for investigations into their development. This chapter discusses a planned architecture by which mirror self-recognition by a robot may be accomplished and the present work's place within that architecture.

- Chapter 3, Test platform

  The work in this dissertation centers around a system developed to allow a humanoid robot to learn about its self-representations, rather than have them provided a priori by an engineer during design. It is helpful, therefore, to develop an early understanding of the hardware that is being used, so as to provide grounding for the reader's understanding of the models and techniques developed in this thesis by relating it to the real hardware upon which it is implemented.

- Chapter 4, Background: Homogeneous representations

  It is likely that many readers will be unfamiliar with the geometric conventions used in the representations and methods developed in this thesis. This chapter provides a brief overview of homogeneous coordinates and projective geometry in order to sufficiently familiarize or reacquaint the reader with these topics prior to developing the rest of the material.

- Chapter 5, Background: Computer vision

  The remainder of the thesis will make heavy use of techniques from 3D computer vision. As such, this chapter provides a brief overview of the necessary computer vision background required to develop the rest of the thesis.

- Chapter 6, Kinematic inference

  As a starting point in self-modeling capabilities, this chapter discusses the construction and evaluation of a system that allows for the robot to infer its kinematics based on data sampled by an external 3D metrology system. The robot estimates the parameters of the kinematic model describing its arm and the performance of this model is evaluated.

- Chapter 7, Integrating kinematics and computer vision

  In this chapter we construct the first complete version of the robot's kinematic-visual self-model by enabling the robot to infer its kinematic structure from data sampled by its stereo vision system. In doing so, we derive the inferred kinematic chain from measurements that agree with the stereo vision bases. As a consequence, the two systems are mutually-calibrated.

- Chapter 8, Simultaneous refinement of kinematics and vision

  One consequence of having the kinematic and visual representations calibrated to each other is the ability to project the robot's predicted end-effector position into the visual field of the robot's stereo cameras. Projecting the robot's kinematic predictions into 2D allows us to use the robot's kinematic structure as a visual calibration target. Optimization allows us to mutually refine the two models against each other, arriving at a superior calibration for both kinematic and visual aspects of the robot's self-model.

- Chapter 9, Inferring the visual perspective describing reflections in a mirror

  An important component of self-understanding is the ability to situate oneself in the environment. A component of this is understanding different views and perspectives. In this chapter, the robot will not only infer a visual perspective different from its own, but it will also use self-knowledge in order to infer this perspective. By watching the reflection of its arm moving in a mirror, the robot will infer the position and orientation of the mirror. It will go on to compute the visual perspective representing images of 3D scene geometry as reflected in the mirror.

- Chapter 10, Toward self-aware robotic systems

  This chapter will discuss and summarize the content of this thesis, as well as discuss current research directions, potential applications of techniques developed in this thesis, and future work.

Chapter 2

The mirror-test as a target for self-aware systems

The mirror test [16] has long been of interest to the psychology and artificial intelligence communities because of interest in understanding self-awareness and developing self-aware systems. As described in the introduction, the mirror test involves discreetly marking a subject with dye or makeup and observing their reaction to this mark when witnessing it in a mirror. If the animal inspects the mark on their own forehead in a self-directed gesture, then it is considered to be self-aware. The test has now been performed in many variations and on many animals (see [46] or [47] for reviews). To date, only a few non-human species pass these tests, including some primates [47], elephants [48], dolphins [49], orcas [50], and European magpies [51]. Humans pass this test by around 18 months of age [15].

What it means to pass the mirror test has been a subject of debate among psychologists. Gallup's account [52] of how this is accomplished relies on the notion that an agent recognizes the physical manifestation of their self as reflected in the mirror (what Rochat [3] would call the "Ecological Self"). Recognizing a difference between their expected appearance and their current appearance in the mirror, an animal that passes the mirror test investigates this difference on themselves. In Gallup's view [52], the mirror test is evidence of "mind," the ability to observe one's own metal states. Epstein, Lanza, and Skinner [53] famously offered a counterpoint to the idea that the presence "self-awareness" and "self-concept" are tested by the mirror test by conditioning pigeons to peck at blue dots. When presented with a blue dot painted on the body, obscured from direct view by a bib such that it could only be observed as reflected in a mirror, the pigeons pecked at the dot on their body, rather than its reflection in the mirror.[1] Mitchell [54] discusses an alternate hypothesis to Gallup's for how self-recognition is accomplished, in which the contingency of one's motor state to changes witnessed in the visual field is identified. The hypothesis proceeds from the idea that agents readily observe the motor states of others, allowing them to perform such actions as to imitate them. In Mitchell's view [54], mirror self-recognition arises from identifying that the motor state of the mirror-image matches one's own kinesthetic state.

A variety of tests have been developed to test both the development of the mirror self-recognition capability as well as the presence of various aspects of the ability to use mirrors and self-recognize. Instrumental mirror use as a tool for spatial reasoning, for instance, has been tested both in animals [24, 25] and in the developing mind of the human child [15]. In one such test [24], a marmoset sits on a shelf in an enclosure with food pellets placed under the shelf. There is a gap between the shelf and the wall of the cage such that the marmoset can reach its arm through the gap to reach the food pellets, but cannot look directly at them, as its head will

[1] Of course, because the pigeons were trained to peck at blue dots, recognition of this as different from their prior appearance is not implied. Rather, this implies an ability to reason about the spatial transformation imposed by the mirror, combined with either knowledge that the dot now lies on the body or the ability to determine the position of the dot in space relative to the body. To compare this to other tests, it is reminiscent of instrumental mirror use, as discussed later in this chapter.

not fit. Observations of marmosets in this study support the hypothesis that they are capable of instrumental mirror use in order to perform the spatial reasoning necessary to reach the food pellets. This sort of instrumental mirror use occurs in several animals that are unable to pass the mirror test [16].

Instrumental mirror use also occurs in humans prior to the ability to pass the mirror test[2] [15]. As discussed in the introduction, Bertenthal and Fischer [15] performed a series of tests in which they constructed a developmental timeline for a number of skills regarding interactions with mirrors. In this timeline, they placed spatial reasoning tasks as emerging prior to the ability to pass the rouge test, emerging at around 8 and 18 months, respectively. Toddlers were able to pass a social task, being able to verbally identify themselves as the subject reflected in a mirror in response to the question, "Who's that?" at around 24 months. This timeline, combined with the aforementioned animal tests, helps to identify the relative complexity of these tasks and to place the emergence of self-reasoning capabilities into a developmental context.

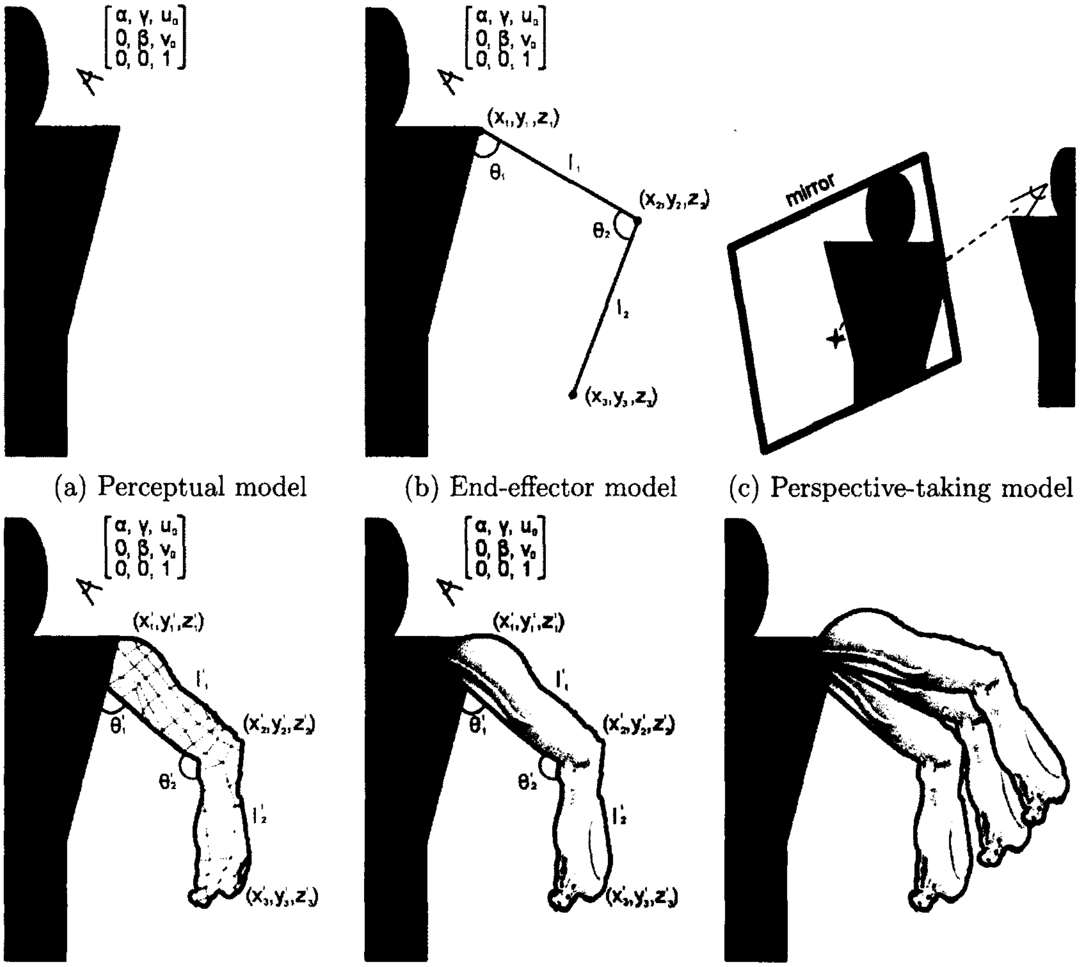
Because of interest in this test, in understanding self-awareness, and in developing self-aware machinery, several robots have been programmed to perform some variety of mirror test. In one series of experiments, Michel, Gold, and Scassellati [28] and Gold and Scassellati [30] solved a task of image segmentation, classifying pixels as either belonging to the robot ("self") or not ("other"), based on temporal correlations between changes in the visual field and the initiation of motor activity. This system is unable to pass the mirror test, however, because it does not model the visual appearance of the robot and therefore cannot identify changes in the robot's appearance. Takeno, Inaba and Suzuki [29] observed a robot when imitating its re-

[2] Also called the rouge test when performed on infants because in the classical setup [15] the child's mother marks their face with rouge makeup.

flected image in a mirror, versus when imitating another robot, to determine that the robot could distinguish between the two using a neural network architecture. The mirror behavior in this task, however, is based on flashing LEDs, and the robot performing this task has no way of interpreting visual self-image in the way that the mirror test requires. More recently, a small, wheeled robot with a human-like face was programmed to respond differently to images of itself (using first-person pronouns) rather than other objects, by using object recognition techniques [31]. In a later expansion on this, the robot differentiates itself from other robots of the same model by randomly generating flashes of an LED mounted in its "nose," and checking to see if the flashes are the same (reflected) or different (randomly generated by another robot). To date, no robot has passed the full mirror test as designed by Gallup [16].

2.1 An architecture for mirror self-recognition

The overall goal of this project is to explore the concept of self-awareness in robotic systems by emulating forms of self-awareness that develop in humans. Because it is the commonly-recognized marker of self-awareness in animals, the mirror test provides us with a concrete, recognizable milestone to work toward in this pursuit. As a first step in understanding self-awareness in robotic systems, we outline an architecture that, conceptually, could allow a robot to pass the mirror test. The proposed architecture is composed of six models describing different forms of self-knowledge that we believe are sufficient to accomplish this task. Conceptually illustrated in Figure 2.1, they are the *perceptual model*, the *end-effector model*, the *perspective-taking model*, the *structural model*, the *appearance model*, and the *functional model*. These are learned by the robot through observation, allowing for refinement and



(a) Perceptual model (b) End-effector model (c) Perspective-taking model

(d) Structural model (e) Appearance model (f) Functional model

Figure 2.1: Diagrams describing the six basic components of the proposed architecture for mirror self-recognition.

change over time. We propose that in a sufficiently advanced system, this process of self-observation will enable the system to pass the mirror test, as the system will detect differences between its expected appearance and its current one.

This dissertation will cover work on the first three of these models, the perceptual, end-effector, and perspective-taking model. At the close of this thesis, the robot will possess a model of how its arm moves and how its vision system works, with the ability to relate the two to each other. It will be able to use self-knowledge in order to infer the visual perspective of a mirror, enabling it to perform tasks reminiscent of the spatial reasoning tasks that infants and animals have performed using mirrors [24, 15], which appear to be a precursor to passing the mirror test. The comparable ability in humans appears at 8 months, much earlier than the 18 months at which humans pass the mirror test. In terms of the mirror test, the ability to infer the visual perspective of a mirror and relate it to the current position and pose of the body with respect to that mirror provides the means by which the robot could eventually compare its expected appearance, as projected into the mirror, with its actual appearance, as reflected in it.

2.1.1 The perceptual model

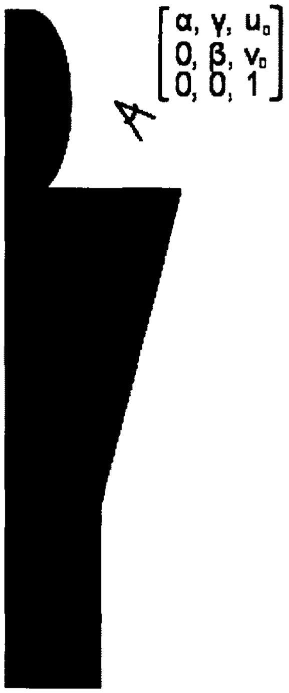


Figure 2.2: The perceptual model describes the robot's stereo vision system, capturing standard stereo vision parameters.

The perceptual model, Figure 2.2, describes the robot's vision system using the common *pinhole camera model* [55]. It is capable of both reconstructing a 3D point in space, given 2D coordinates in both of the robot's stereo cameras, and projecting a known 3D point to its corresponding 2D coordinates in each camera.

2.1.2 The end-effector model

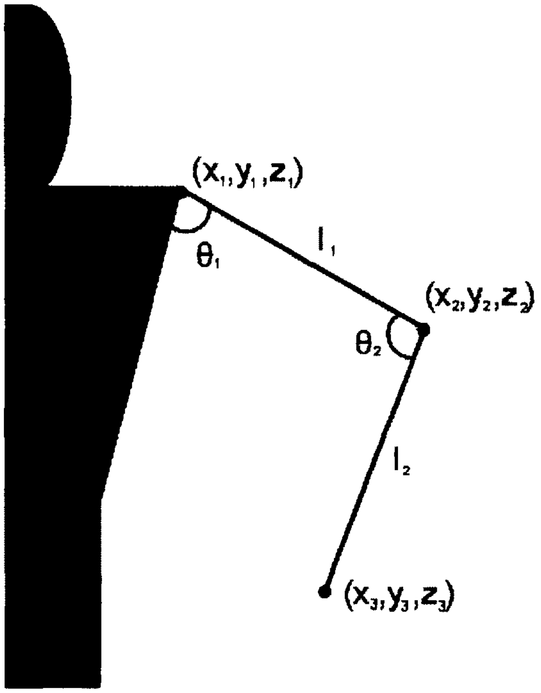


Figure 2.3: The end-effector model is intended to describe the the robot's kinematics.

The end-effector model, Figure 2.3, describes the motion of the robot's end-effectors through space. In this dissertation, the robot's kinematics are modeled using the Denavit-Hartenberg convention [1], though it is conceptually possible to use any number of kinematic modeling conventions in this system. In Chapter 6, we will introduce the methods used by the robot in order to infer its kinematics based on observations of the arm's motion.

An important feature of the system is that the perceptual and the end-effector models are calibrated to each other. Because the end-effector model is inferred from data sampled by the stereo vision system, the samples used to reconstruct the robot's kinematic chains are expressed in the native coordinate system of the stereo vision system. The mounting of the robot's cameras with respect to its frame is known, as in Figure 2.3. In Chapter 7, this basic level of competency will be accomplished. In Chapter 8, we will demonstrate how the systems are able to refine each-other's calibrations by minimizing the difference between the expected positions of the robot's end-effectors in each camera and their observed positions, utilizing models and methods that we developed in prior work [56]. This is reminiscent of developmental accounts of how children learn about their bodies and senses by using them with respect to each other, one of the earliest forms of self-awareness to develop in infancy.

The mutual calibration between the perceptual and end-effector models produces properties reminiscent of the body schema, in which senses can be interpreted with respect to the body's current pose as related through kinesthesis. As such, interesting computations can be performed, such as computing the position of the 2D image of the end-effector in the visual field.

2.1.3 The perspective-taking model

The perspective-taking model, Figure 2.4, is an extension of the perceptual model. It allows the robot to model sensors from an external point of view. One could imagine social functions of this model, such as representing the visual perspectives of other actors in an interaction. In this dissertation, it will be used to allow the robot to take the perspective of a mirror in its current visual landscape. This will represent a milestone towards the development of a robot which is able to pass the mirror test, the ability to represent and interpret the visual perspective of the mirror. It will be developed in Chapter 9.

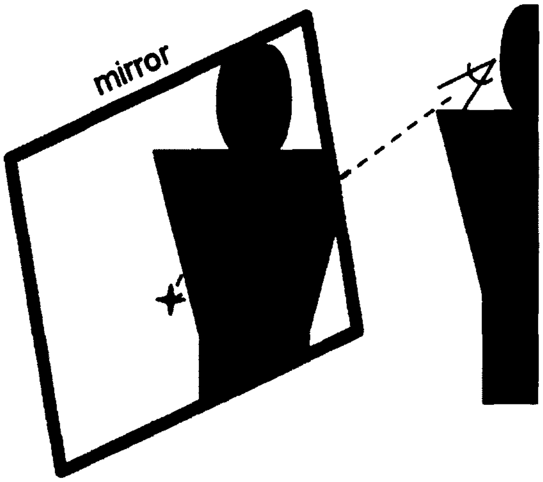


Figure 2.4: The perspective-taking model enables the robot to take on a different visual perspective. In this dissertation, it will be used to create a virtual camera describing the robot's spatial relationship to reflections that it sees in a mirror.

2.1.4 The structural model

The remaining three models - the structural, appearance, and functional models - are reserved for future work.

As currently planned, the structural model is intended to represent the robot's rigid, external 3D structure, as shown in Figure 2.5. It is intended to be computed by automatically by choosing 3D points along the robot's frame and computing a model of their position using the techniques from the end-effector model. This collection of points would then be used to approximate the robot's surface geometry. Once implemented, it is hoped that the structural model will capture the 3D shape of the robot's body parts, such as the 3D structure of its hand, the surfaces of its arms, and the shape of its head.

2.1.5 The appearance model

The appearance model is planned to map surface properties, such as color, onto the geometry provided by the structural model, as in Figure 2.6. The combination of

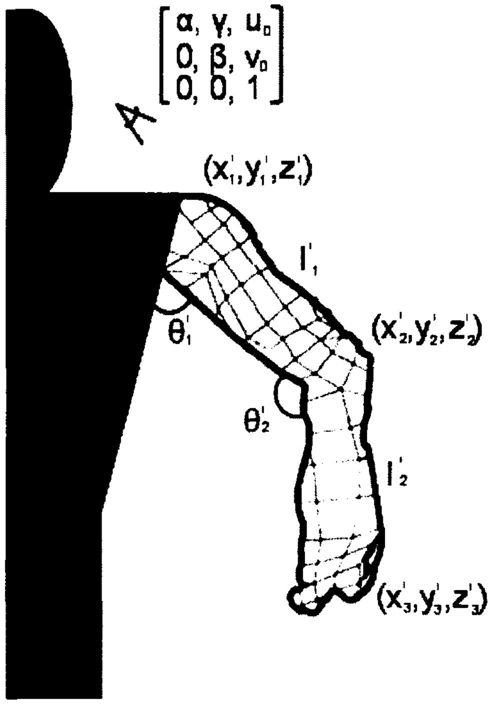


Figure 2.5: The structural model is intended to capture a description of the 3D geometry of the robot, described with respect to its kinematics and vision.

the structural and appearance models is intended to enable the robot to develop an impression of its current appearance in 3D. Modeling appearance in this way is could provide a mechanism to allow the robot to identify that its appearance has changed and have sufficient information in order to detect the mark as reflected in the mirror during the mirror test.

2.1.6 The functional model

The functional model, Figure 2.7, is intended to allow the robot to determine the effect that its actions have on objects in its environment. An example of this would be enabling the robot to infer causal relationships, such as that between enacting a motor command and changes in the visual field. The methods presented by Michel et al. [28], Gold and Scassellati [30], and Stoytchev [57], designed for related tasks, are

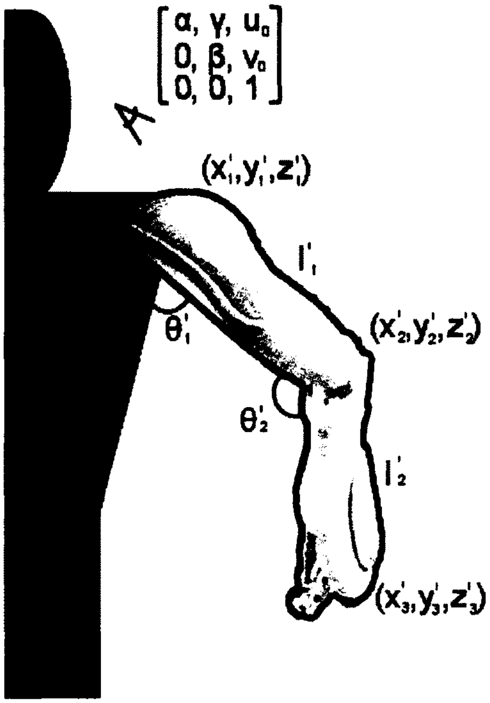


Figure 2.6: The appearance model is intended to add visual detail to the 3D geometry described by the structural model, such as coloration.

examples of similar, existing systems. In the case of the future system, it is planned that a method similar to that presented in [30] will be used in order to perform initial self-other discrimination. When combined with the structural and functional models, it is hoped that this will allow the system to begin constructing a self-model without the need to explicitly instrument the robot with visual markers, as done in the experiments presented in this dissertation.

The current plan for implementing these models is to have the robot use its functional model to segment ego-motion (its own motion) from other motion in the scene, allowing it to determine 3D point cloud structure belonging to the self. Once this structure has been determined, constructing a 3D model from the points belonging to the self could allow the robot to simultaneously describe the structural and appearance models. It is hoped that fitting a current impression of the robot's pose and

Figure 2.7: The functional model is intended to allow the robot to determine the effects of its actions.

structure to observations made by the visual system will allow the robot to track its limbs as they move in space, obviating the need for the markers used in this thesis and constituting a markerless tracking method. Additionally, the construction of such 3D structure may allow the techniques developed in Chapters 7 and 8 to be revisited, allowing the robot to refine its visual calibration from its 3D structure, rather than solely from its motion, and allowing the robot to determine the calibration parameters describing the mirror from a single frame of video. With future advances in computer vision, such rich impressions of 3D scene structure could allow the robot to infer the reflectance properties of non-planar objects in the scene as well as planar mirrors.[3]

[3] It is assumed, in this dissertation, that the mirror that the robot looks into is planar. This is partly due to limitations in the representation of 3D scene geometry (generally assumed to be a collection of points), and partly due to the fact that, because the robot must move in order to determine the perspective of the mirror, it would require an extraordinarily large number of samples in order to infer more complicated geometry and reflectance properties.

2.2 Summary and conclusions

In this chapter, the relevance of the mirror test to psychology and artificial intelligence is briefly discussed. We select this test as a goal to build towards in a line of research toward developing self-aware robots and artificially intelligent systems. Having determined the goal that we are working toward, we outline a plan for constructing a robot that is capable of passing this test. The plan is implemented as a system in six parts: the perceptual model, the end-effector model, the perspective-taking model, the structural model, the appearance model, and the functional model. The first three of these have been implemented and are the subject of this dissertation. Combined, they enable the robot to perform a task that is based on spatial reasoning tasks using mirrors. The emergence of the use of mirrors for spatial reasoning appears in infants at around 8 months, as well as in some animals that are unable to pass the mirror test. This suggests that the use of mirrors as instruments for spatial reasoning may be a precursor to the ability to pass the mirror test. Though much has been accomplished with robots interacting with mirrors, and though this thesis lays a solid groundwork for work in self-aware autonomous systems, to date no robot has passed the mirror test, as posed by Gallup [16].

The three models presented in this chapter that have been left to future work, when combined with the methods presented and developed in this dissertation, are intended to enable the construction of a classifier capable of passing the mirror test. If successful, the robot will be able to construct an impression of its current appearance and an expectation of what that appearance should be as reflected in a mirror. This is intended to provide a model that is sufficient to construct a classifier capable of determining that the robot's appearance has changed, based on differences from its expected appearance, and, moreover, how it has changed. It is worth noting that even if this system should be sufficient to identify a mark applied to the robot, it does not cover matters such as motivation to inspect the mark. The outlined system is only intended to provide the models required to perform this classification task.

Chapter 3

Test platform

The purpose of this chapter is to describe the hardware systems used during the development and evaluation of the techniques presented in this thesis. The system was developed and tested on an upper-torso humanoid robot named Nico (Figure 3.1). Nico was developed as a platform for experiments in social and cognitive development, and is modeled after the human 12-month-old infant at the fiftieth-percentile of growth. The kinematic-visual self-model developed in this thesis utilizes the robot's stereo vision system and models the kinematics of the robot's arm.

In addition to Nico, a Vicon MX motion tracking system is used as an external metrology device. The use of this system allows us to verify the robot's kinematic inference algorithms independently of the performance of its stereo vision system. The configuration used in this thesis includes four cameras used to reconstruct the positions of reflective markers in 3D. The cameras are each mounted with a ring of infrared LEDs. The light from these LEDs is reflected by the markers. The cameras are fitted with filters to remove visible light, allowing the markers to be tracked.

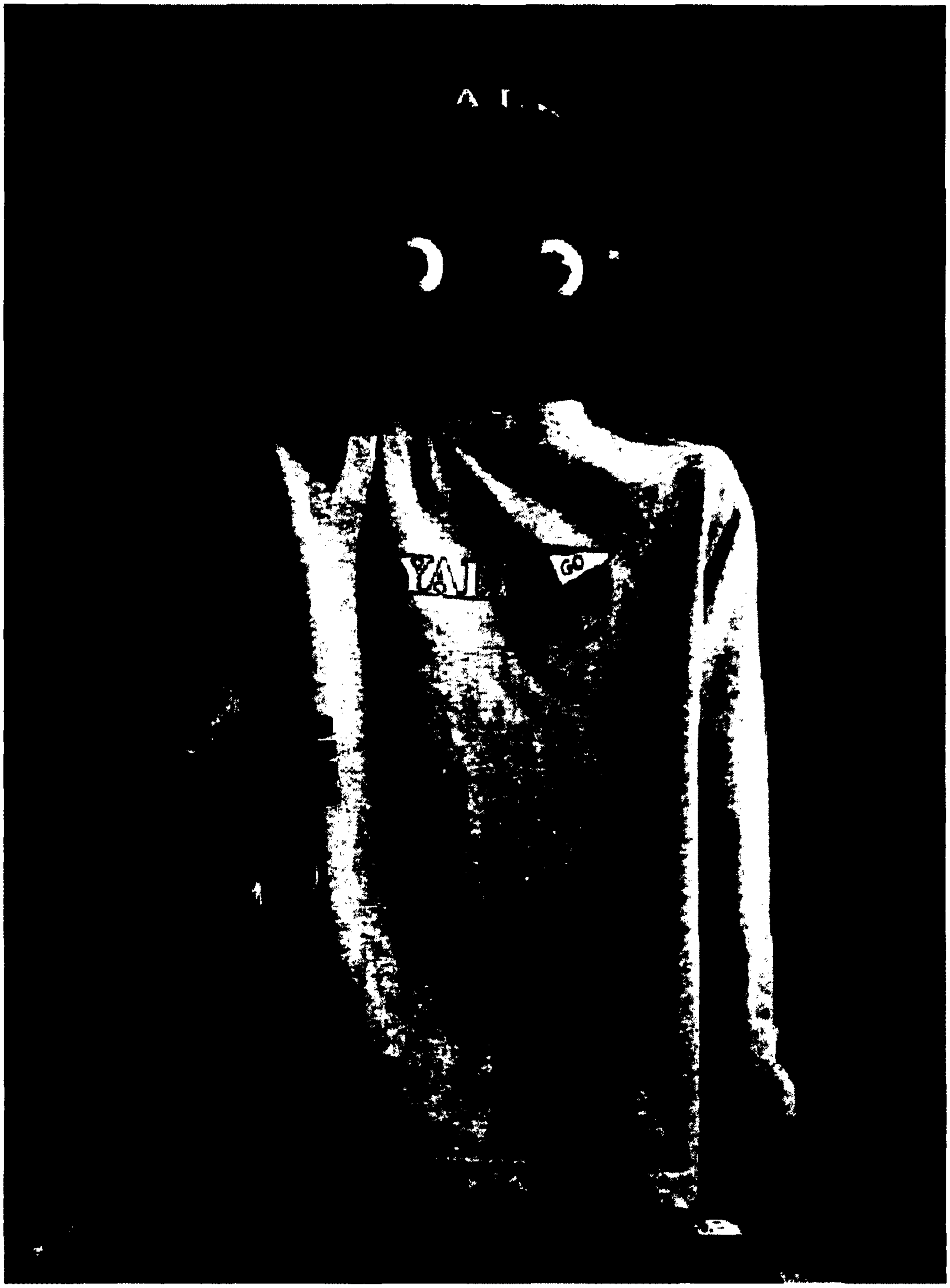


Figure 3.1: Nico is a humanoid robot modeled after the human 12-month-old at the fiftieth percentile of growth.

3.1 Nico, an infant humanoid

Nico's kinematic structure, Figure 3.2, is designed to match that of the human 12-month-old infant. Measurements of the sizes of the robot's body parts are based on data presented in [58], formatted into engineering drawings of the human form in [45]. Nico was designed with twenty three degrees of freedom, six in the head, six in each arm, three in the torso, and two in the hand. Details regarding the actuation and control of Nico's mechanical degrees of freedom can be found in Section 3.1.2. Nico also possesses a stereo vision system consisting of four cameras, intended to approximate the low resolution, wide field-of-view peripheral vision and high resolution narrow, field-of-view foveal vision provided by the human eye. Details of Nico's stereo vision hardware can be found in Section 3.1.1.

For this work, the motion of the robot's arm is restricted in that the two distal joints of the robot's arm do not move. One of the core pieces of software developed in this dissertation infers the kinematics of the robot's right arm. The first reason that the robot's motion is restricted is that the Vicon motion tracker requires polygons of markers to be placed along the surface of the tracked subject. The triangle of markers placed on the robot's arm in order to instrument it for tracking, Figure 3.3, would be broken if the robot rotated either of the two distal joints on its arm (located in the wrist and forearm, respectively). Retaining this locked configuration of the arm during vision-based calibration makes results based on data sampled through the use of the Vicon system comparable to the results utilizing visually-sampled data. Additionally, in the case of the experiments in which a fiducial marker is mounted to the hand, most of the range of motion of these degrees of freedom generally tilts the marker out of the robot's field of view. When the robot's arm motion is randomly generated, this means that a view of the marker sufficient to track it is only visible in

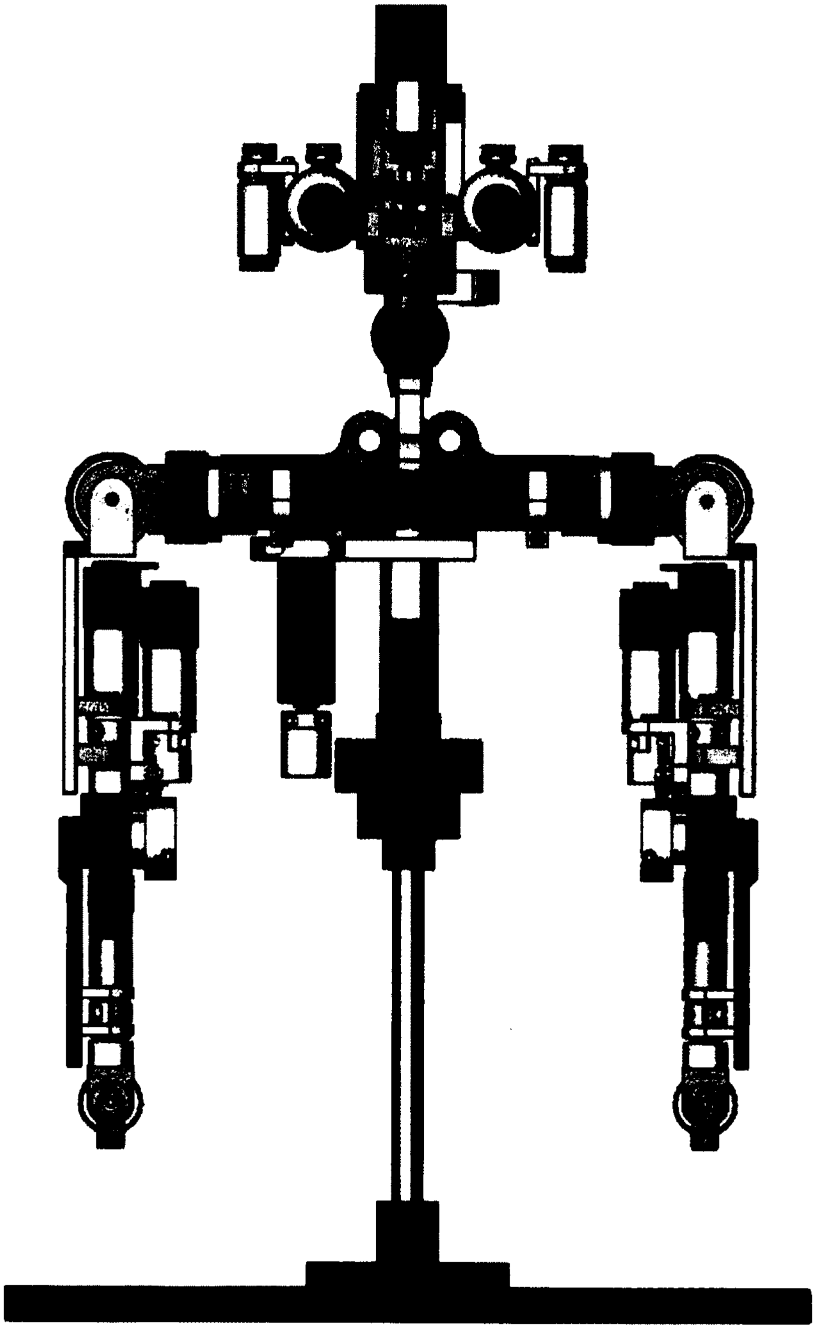


Figure 3.2: SolidWorks rendering of the humanoid robot, Nico.

a minority of poses when the two degrees of freedom at the wrist are moved. In the case of the robot's left arm, it was decided that modeling both arms was redundant, as the modeling case for the right arm readily generalizes to the left.

In experiments in human-robot interaction, Nico "wears" the clothing depicted in Figure 3.1. The clothing helps to define the robot's body shape to human study participants, helps to orient the participant's impression of the orientation of the robot's head, and helps to provide a friendly, human-like appearance to the robot. Unfortunately, the hat partially obstructs the robot's visual field, and the shirt sometimes drapes over the markers used to instrument the robot's arm. As such, for the work in this thesis, the clothing has been removed, as in Figure 3.4.

3.1.1 Stereo vision system

Nico's stereo vision hardware consists of four Elmo MN34H miniature remote camera heads mounted into the robot's head. The cameras are controlled by Elmo CC421E camera control units connected to PXC200a frame grabbers, which are based on the Brooktree Bt878 chipset. Each of the robot's two "eyes" is designed to mount two camera heads. The intention of this design is to simulate the foveal pit in the human eye, which provides the eye with greater visual acuity in the middle of the visual field. This, in turn, raises the visual resolution available for imaging objects that a person is immediately attending to ("looking at"). One camera in each eye has a short focal length ($f/2.8$ aperture, $f = 2.2$mm, 80° horizontal viewing angle), providing a wide field of view. The other camera has a longer focal length ($f/3.5.8$ aperture, $f = 15$mm, 13.5° horizontal viewing angle), providing greater visual acuity in a smaller region.

For the purposes of the experiments detailed in this thesis, it was decided to use only the shorter focal length cameras, in order to reduce the overhead of vision

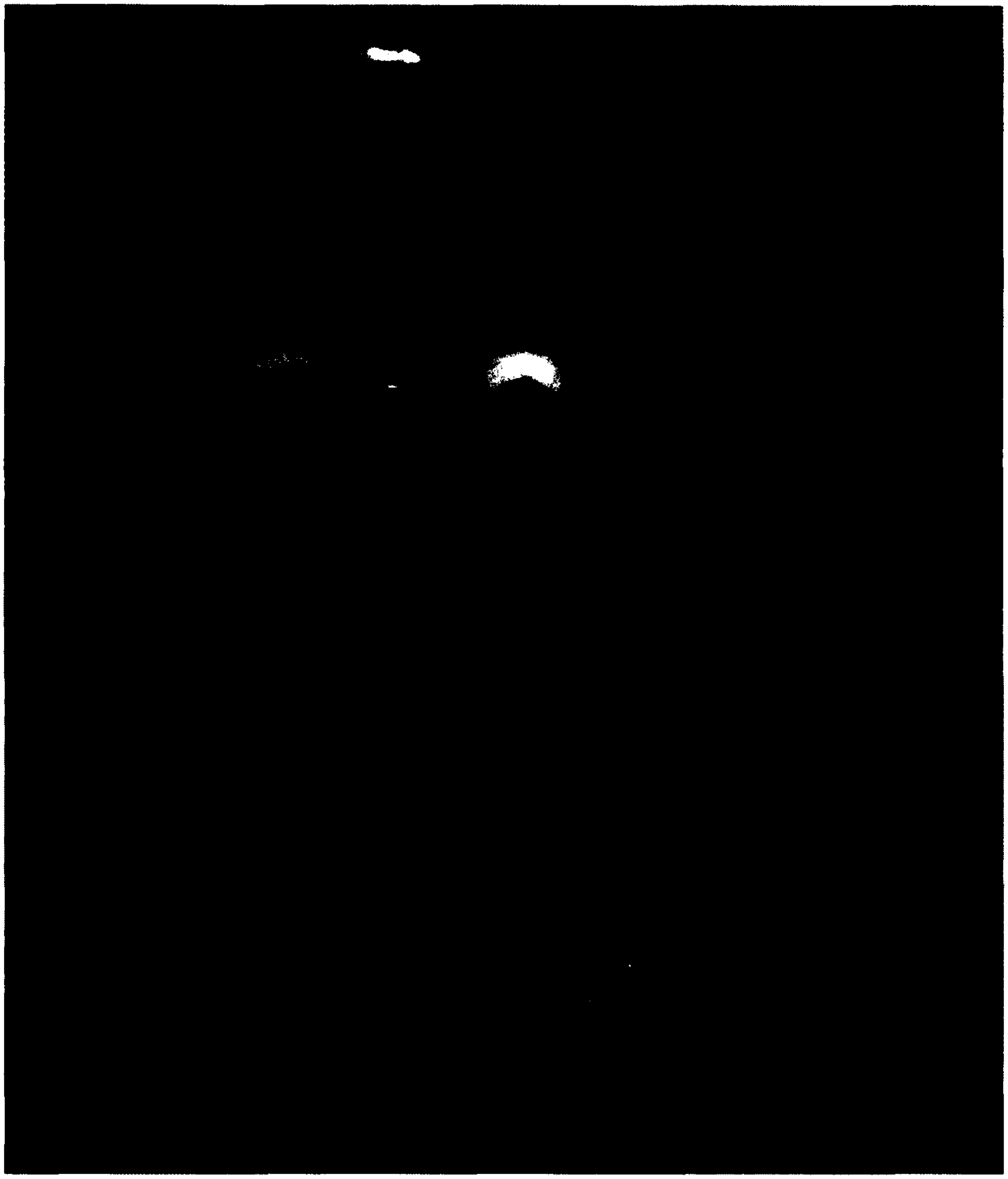


Figure 3.3: The humanoid robot, Nico, instrumented with reflective markers for use with the Vicon MX Motion Tracker. The markers are the balls covered with silver reflective tape mounted to the elbow, forearm, and wrist.

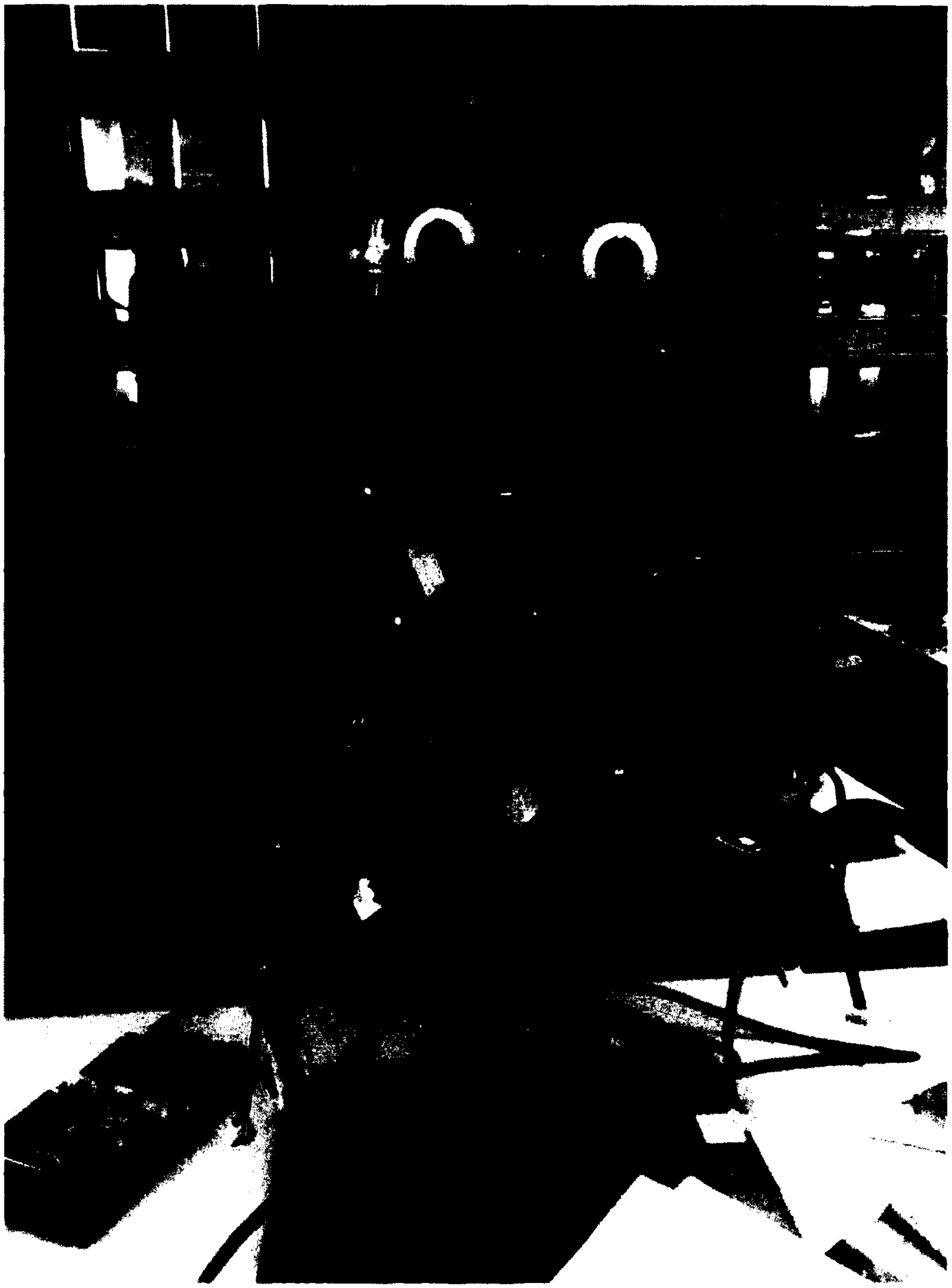


Figure 3.4: Nico, with its clothing removed, exposing its motors and cabling. The arm on the right side of the picture (Nico's left) was not used in this thesis, but was developed as part of a student senior project, in order to allow Nico to lift heavier payloads. In this photo, that arm does not have some of its parts attached.

processing while providing the robot with a field of view that accommodates a larger portion of range of motion of its arm than that captured by the foveal camera. The two cameras used in this thesis are connected to a commodity PC running Ubuntu GNU/Linux [59]. Nico's computer vision software was implemented in a mixture of C++ and Wolfram Mathematica [60].

3.1.2 Actuation and motor control

Nico's body is actuated by a mixture of Faulhaber (MicroMo) and Maxon DC electric motors. Each motor is fitted with a gearhead to reduce the high-speed, low-torque motion of the electric motor to a lower-speed, higher-torque output. Each motor is additionally fitted with a magnetic resonance encoder which measures how far the motor has turned. These motor-gearhead-encoder combinations are then each connected to a JR Kerr Pic Servo 3PH Motion Control Board.[1] The boards are connected to each other serially through RS484 ports forming a network. The JR Kerr SSA-485 Smart Serial Adapter is a USB (or RS232) to RS485 serial converter, which is connected to the network of motion control boards, allowing for the network to be controlled by a commodity PC over USB.

The motor server runs custom software developed for the purpose of simplifying the robot's servo control developed by myself and Graham Radman, called "motorsrv." This software is based on and adapted from two earlier pieces of software developed by Chris Crick, Marek Doniec, Kevin Gold, Elizabeth Kim, Frederick Shic, Ganghua Sun, and others. These, in turn, are based on an update of Brian Rudy's "libnmc," which was developed as a Linux port of the software to control JRKerr motor control boards.

Nico's hand was developed by myself and Marek Doniec. Instead of motor-

[1] http://www.jrkerr.com/

gearhead-encoder combinations, the hand uses only DC electric motors mounted with gearheads. The hand has two degrees of freedom allowing for either the index finger or the remaining three fingers in the hand to be opened and closed separately. Since the hand was designed only to open and close these fingers in order to perform gestures, it was deemed unnecessary to attach an encoder to measure joint angle. Removing the encoders from these combinations allows for a more compact design of the hand, as can be seen in Figure 3.3. The hand is controlled by a custom motor board built around two Texas Instruments LMD182000T H-bridges, which is in turn controlled by an Atmega 16 microcontroller running custom firmware. These are connected to the PC via USB and controlled by a library of custom software.

3.2 The Vicon MX motion capture system

The kinematic inference software used in this thesis was initially developed to estimate the robot's kinematics from the tracked 3D position of the robot's end-effector rather than from pairs of 2D positions tracked by the robot's stereo vision system. This allowed for it to be tested in isolation from the complexities of stereo vision, such that any potential errors in calibration could be attributed to flaws in the algorithm, rather than flaws in the stereo vision system. In order to test kinematic learning under this regime, a Vicon MX Motion Capture System was used to track the 3D position of the robot's end-effector as it underwent a series of motions.

The motion capture system works by attaching reflective markers, Figure 3.3 to the subject to be tracked. These markers are covered in a reflective tape that reflects infrared light emitted by a ring of LEDs mounted to the front of the Vicon T40 cameras used in the tracking system, Figure 3.5. The T40 captures images of the reflectors at a resolution of 4 megapixels and a speed of 515 frames-per-second. These cameras are connected to the MX Giganet core, which synchronizes the cameras and communicates to custom software provided by Vicon called "Vicon Nexus," which runs under Windows. A custom software library and a server were developed to allow the robot's software, which was developed under Linux, to communicate with the Vicon Nexus software over a TCP-IP network. The motion tracking system used in these experiments uses four cameras. Three of these cameras were mounted on tripods in a triangle surrounding the robot each at a distance of approximately a meter. The fourth camera was mounted slightly farther away on a wall-mount mounted near the ceiling of the laboratory, providing an overhead view of the scene.

In this configuration, the robot itself sits on a wheeled desk with chrome legs. In order to prevent the reflectivity of the legs from interfering with the motion capture system, the legs were wrapped in black tape. The Vicon Nexus software requires at least three markers to be mounted to the subject in order to be tracked, so a triangle of markers was placed on Nico's arm using double-sided tape, as in Figure 3.3. The positions of these markers were tracked in 3D and transmitted to Nico's kinematic inference software in real-time. The robot to recorded their positions along with joint angles by moving its arm, stopping, and recording a point. This was done for each arm pose used in the experiment, though only the marker mounted to the hand was used for kinematic inference. For these experiments, the PC typically used for Nico's vision and high-level processes was used to operate the motion tracker, while an additional laptop PC was used to control the robot.

3.3 Summary

In this chapter we have presented the equipment that will be used throughout this dissertation in order to test the algorithms developed therein. We have presented the

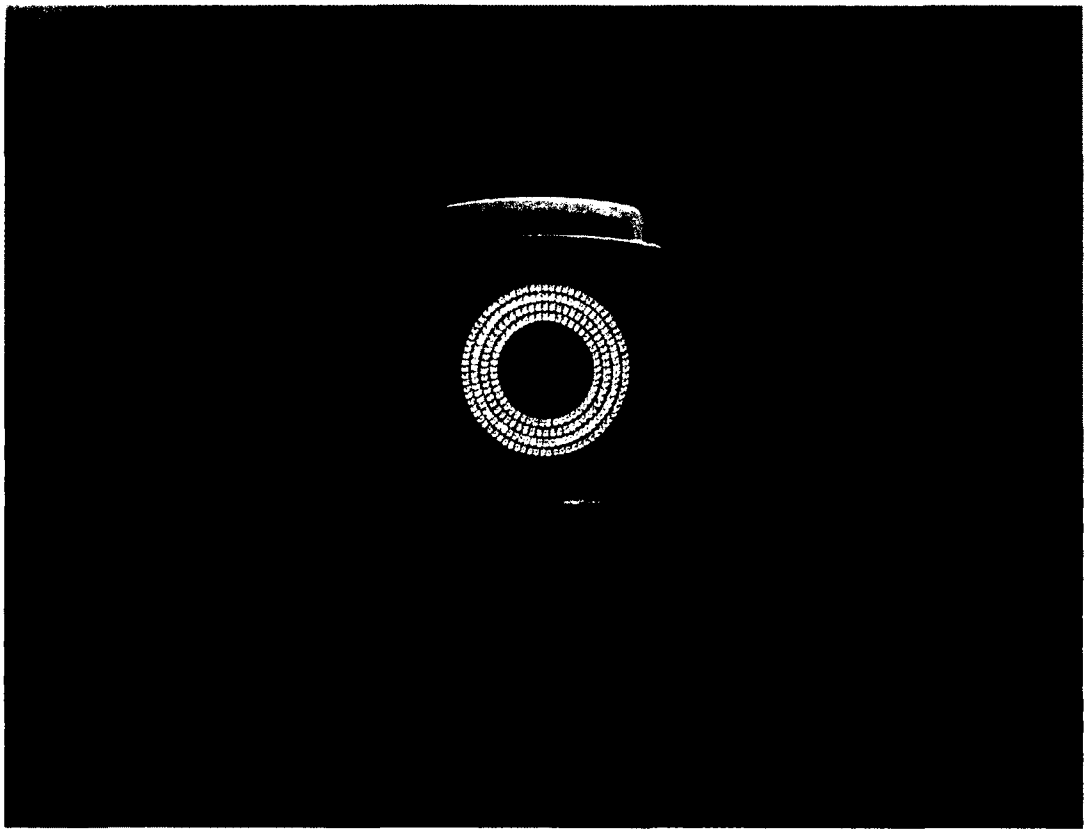


Figure 3.5: A Vicon T40 camera, used with the Vicon MX motion tracking system. The white ring on the periphery of the camera is a set of infrared LEDs used to illuminate the reflective markers with which the tracked subject is instrumented. The black plastic is an infrared filter which allows only the illumination of the LEDs to pass through. The camera itself has a 4-megapixel resolution and is capable of filming at 515 frames per second.

upper torso humanoid robot, Nico, which is modeled after the kinematic structure of the 12-month old infant. We have discussed the robot's actuation and vision system, as well as the computational and software resources available to it. We have also discussed the Vicon MX motion tracking system, which has been used to provide an external source of measurements of the robot's arm position, in order to measure the performance of the system's kinematic inference algorithms separately from its vision performance.

Chapter 4

Background: Homogeneous representations

A little bit of background in homogeneous coordinates and projective geometry is necessary in order to develop the material appearing in the remainder of this thesis. It is common to use homogeneous coordinates, as developed by Möbius [61], in computer vision applications. Under this convention, coordinates are represented in a projective space embedded in a higher-dimensional space. Due to this property and the linear nature of the coordinate system, projective transformations can easily be represented by matrices when using homogeneous coordinates. This makes the representation of the projections induced by cameras relatively straightforward.

Under homogeneous coordinates, rotations and translations can be combined into a single matrix operation representing the rigid transformation of points through space. The output of the Denavit-Hartenberg convention [1], which will be used for kinematic modeling in this thesis, is a 4×4 matrix describing the transformation performed by a joint's motion. When employed, the use of Cartesian representations will be obvious through their representation and usage. At times, concepts will mix the use of Cartesian and homogeneous representations. Guidelines for how these representations are interpreted with respect to each other, or normalized as necessary, will be provided as needed.

4.1 Points

Homogeneous coordinates represent points in projective space, as illustrated in Figure 4.1. Two-dimensional coordinates can be thought of as points in 3D space projected down onto a plane, referred to as the *projective plane.* As such, homogeneous 2D space is projected down from a Euclidean 3D representation. A two-dimensional point is represented as a 3-vector, as in Equation 4.1, illustrated in Figure 4.2. This vector can be thought of as the Cartesian representation of a 3D point, with the corresponding homogeneous 2D point lying on the projective plane along the vector between that point and the origin. Equivalently, it can be thought of simply as the description of the corresponding vector. The ratios $(x : w, y : w)$, refer to the equivalent Cartesian coordinate. The vector b in Equation 4.1, therefore, has only two degrees of freedom and is unchanged by scalar multiplication, with all scalar multiples of b forming an equivalence class. We generalize this concept to higher dimensions in order to arrive at the homogeneous representation of a 3D point, B, as is Equation 4.2.[1]

$$b = \begin{bmatrix} x & y & w \end{bmatrix}^T = (x : w, y : w) \tag{4.1}$$

$$B = \begin{bmatrix} x & y & z & w \end{bmatrix}^T = (x : w, y : w, z : w) \tag{4.2}$$

[1] Though vectors will frequently appear in this discussion, it is common to use a matrix representation in computer vision equations, suggesting the common appearance and usage of the formula or symbol. For the remainder of this document, this convention will frequently be followed.

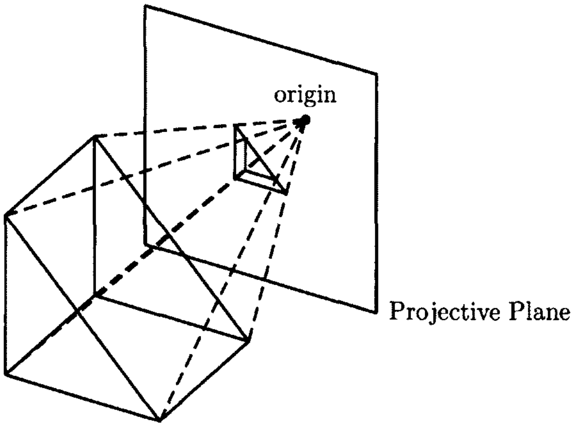


Figure 4.1: The description of each 2D homogeneous coordinate can be thought of in terms of the slope of a line running through the origin to a 3D point. The point through which this line intersects a plane called the projective plane is the 2D point captured by this representation.

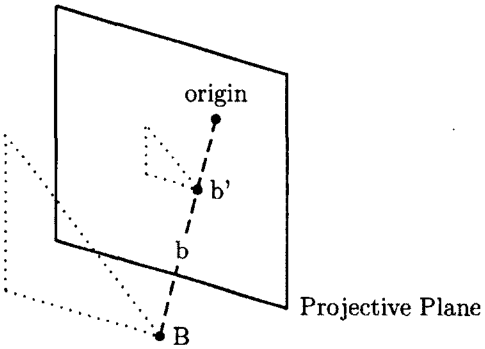


Figure 4.2: The homogeneous vector b represents the 2D point b', which can be thought of as a projection of the 3D point B, or any other 3D point lying along the vector b.

Much of this document will discuss points and transformations performed on points. In Section 6.1 a point representing the position of the robot's hand will be the product of the robot's forward-kinematic model. Collections of reconstructed hand positions will be used to reconstruct planes and circles from points in Section 6.2.1, in order to enable the robot to infer the model of its kinematics. In Section 7.1, we will represent the 2D position of a marker, mounted to the robot's hand, as a point lying at the center of a tracked target. In Section 7.1.2, we will reconstruct its position in 3D, represented as a point, from tracked points in the robot's left and right cameras. In Section 7.2, we will make 2D projections of the 3D point representing the robot's prediction of the position of its hand, based on its forward-kinematic model.

4.2 Lines

Two-dimensional lines can be thought of as lower-dimensional cousins of 3D planes, which will be discussed in the next section. The representation maps onto a familiar equation, Equation 4.3, with the corresponding homogeneous line representation being the vector $l =< a, b, c >$. Lines can be normalized similarly to planes, by scaling such that $< a, b >= 1$, with 2D points being again normalized to their Cartesian equivalents by scaling such that $w = 1$. For normalized points and lines, similar to their 3D cousins, Euclidean distance can be computed as the dot product of the two vectors, with this distance being scaled by the product of the two scalar multipliers in the unnormalized case. Other computations are analogous, such as finding best-fit lines for a set of points via Singular Value Decomposition (SVD).

$$ax + by + c = 0 \tag{4.3}$$

4.3 Planes

The homogeneous representation of a plane is defined implicitly such that the product of the plane with all points lying on it is zero, as in Equation 4.4. The definition of a plane is, like that of a 3D point, expressed as a 4-vector with three degrees of freedom that is unchanged by scalar multiplication. A plane can be uniquely identified by three or more non-collinear points lying on it, allowing us to fit a plane to a set of points by Equation 4.5. Equation 4.5 is solved as a least-squares fit via SVD, using a matrix derived from three or more points. Conversely, a point lies at the intersection of three planes, allowing us to determine the location of a point lying at such an intersection by the same method by substituting planes along the rows of the factorized matrix in Equation 4.5.

$$Q = \begin{bmatrix} \Pi_1 & \Pi_2 & \Pi_3 & \Pi_4 \end{bmatrix} \begin{bmatrix} x & y & z & w \end{bmatrix}^T = 0 \tag{4.4}$$

$$\begin{bmatrix} x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ & \cdots & & \end{bmatrix} \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \Pi_3 \\ \Pi_4 \end{bmatrix} = 0 \tag{4.5}$$

The homogeneous representation of a plane can be thought of as the vector lying normal to the plane using the terms Π_1, Π_2, Π_3 as in Equation 4.6, and a length along that vector which is the distance from the origin to the plane, $-\Pi_4$. Normalization of the homogeneous representation of a plane can be performed by scaling such that the norm $||V|| = 1$. This follows from the Cartesian representation of Equation 4.7, which can be computed by setting w to one, and is often written similarly to Equation 4.8.[2] Homogeneous points can be equivalently normalized by scaling such that $w = 1$. For normalized points and planes, Euclidean distance can be computed as the dot product of the two vectors, with this distance being scaled by the product of the two scalar multipliers in the unnormalized case. Note that this distance is signed, defined with respect to positive and negative sides of the plane.

[2] Note that the right hand side was written as $-\Pi_4$ in order to retain the notation. The right hand side would normally simply be the distance from the origin to the plane.

$$V = \left\langle \Pi_1 \quad \Pi_2 \quad \Pi_3 \right\rangle \tag{4.6}$$

$$\Pi_1 x + \Pi_2 y + \Pi_3 z + \Pi_4 w = 0 \tag{4.7}$$

$$\Pi_1 x + \Pi_2 y + \Pi_3 z = -\Pi_4 \tag{4.8}$$

The 2D projective transformation known as a *homography* can be thought of as the description of a plane moving in 3D, as will be discussed in Section 5.1. In Section 6.2.1, we will fit planes to the trajectory that the robot's hand takes as it moves through space as part of a process of fitting circles to this path in order to compute descriptions of the robot's joints. In Section 9.1.1.2, we will compute the plane in which a mirror lies as a fit between forward-kinematic predictions of the position of the robot's hand and reconstructions of the same hand as reflected in the mirror.

4.4 Rigid transformations

A rigid transformation is any transformation that preserves the distance between two points (by extension preserving angles and parallel lines) for the transformed object. For the purposes of this discussion, we will be concerned with translations and rotations, though reflections can also be considered to be rigid transformations. Homogeneous coordinates afford us the convenience that rigid transformations can be expressed through matrix multiplication, rather than as separate multiplications and additions. They can also be thought of as changes of bases, as in linear algebra, or changes in frames of reference, as in mechanics. The transformation of a point from one position B to another B', as defined in Equation 4.9, is summarized as performed on a vector containing Cartesian coordinates in Equation 4.10, and on homogeneous coordinates in Equation 4.11. Rigid transformations are sometimes summarized in terms of their rotational and translational components, as in Equation 4.12.

$$R = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{1,3} \end{bmatrix}, T = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \tag{4.9}$$

$$B' = RB + T \tag{4.10}$$

$$B' = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} & T_1 \\ R_{2,1} & R_{2,2} & R_{2,3} & T_2 \\ R_{3,1} & R_{3,2} & R_{1,3} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} B \tag{4.11}$$

$$B' = \left[\begin{array}{c|c} R & T \\ 0 & 1 \end{array}\right] B \tag{4.12}$$

In this document, we will use rigid transformations to describe the motion of points through space. In Section 5.2.1, we will develop our model of the projection that a camera imposes on a scene by extending the concept of rigid transformations. In Section 6.1.1, we will construct matrices describing rigid transformations to describe the motion of the robot's joints as it moves through space. In Section 7.1.1.2, we will describe the relationship between a projective transformation called a homography and rigid transformations, in order to describe the method that *Augmented Reality Toolkit* [2] uses to track the motion of a fiducial marker through space.

4.5 Summary

The purpose of this chapter has been to give a brief overview of the homogeneous coordinate system and projective geometry required to develop the remainder of this thesis. We have discussed the representations of points, lines, and planes which will be required to develop the kinematic representations used in Chapter 6. We have developed the basic representation for rigid transformations used under the homogeneous coordinate system, demonstrated the implicit representation of planes, and shown how this representation relates to the representation of points in space. We have also discussed projective geometry, and the projective nature of the homogeneous coordinate system. We will return to these concepts in Chapter 5, where we further develop these concepts for use in computer vision algorithms.

Chapter 5

Background: Computer vision

This chapter will introduce the projective transformations which will be used in the remainder of this thesis, and the parameterization used to characterize these projective transformations with respect to real camera hardware, the *pinhole camera model.* We will begin this discussion with a simple 2D projective transformation known as a *planar homography.* From there, we will develop the pinhole camera model. The chapter will end with a discussion of the projective relationship between two cameras, the inference of which will provide us with the parameterization required to describe rays of light entering the apertures of two physical pinhole cameras, providing us with the basis for stereo triangulation. This projective relationship is known as the *epipolar geometry* of a stereo pair of cameras.

5.1 Homographies

Planar homographies are a simple projective transformation in 2D. Priming this discussion on projections, homographies can be thought of as the 2D image of a plane as it transforms through 3D space, causing it to be viewed from different angles.

Under rigid transformations, the distance between two points is retained. By extension, angles are also preserved. For a projective transformation, these properties do not hold. Consider taking a photograph of a chessboard from two different angles. The 2D positions of the points at the intersections of the corners of this chessboard are projectively transformed between these two images. This transformation is called a *homography.*

For the purposes of this discussion, a homography can be thought of as the general linear projective transformation over 2D points, as in Equation 5.1, where b is the point prior to the homography's transformation and b' is the transformed point. It is expressed as a $3x3$ homogeneous matrix with eight degrees of freedom, as in Equation 5.2.

$$b' = Hb \tag{5.1}$$

$$H = \begin{bmatrix} h_{1,1} & h_{2,1} & h_{3,1} \\ h_{1,2} & h_{2,2} & h_{3,2} \\ h_{1,3} & h_{2,3} & h_{3,3} \end{bmatrix} \tag{5.2}$$

The *Direct Linear Transformation (DLT) algorithm* can be used to compute the homography between two sets of matched points [55]. Typically, one set is a model stored in the computer's memory, and the other is derived from images of a target, such as a chessboard calibration target. A derivation of the direct linear transformation method for computing a homography is provided by Hartley and Zisserman [55]. The derivation proceeds from the fact that the cross product between two identical vectors is zero. As such, minimizing the cross product between points projected by H and their imaged counterparts can be used as a method for creating a linear system to compute homographies. This is done by writing the vector produced by this cross product as in Equation 5.3, then factoring out the terms of H such that they can be found as the right null space of the resultant matrix. The matrix form of this linear system can then be computed from pairs of matched points, each pair of matched points contributing two lines to the matrix and thus two constraints on the linear system. The system can then be solved by Singular Value Decomposition (SVD). Since each pair of matched points contributes two constraints and a homography has eight degrees of freedom, the system can be determined from four pairs of matched points.

$$b' \times Hb = \begin{pmatrix} b'_2 h_3^T b & - & b'_3 h_2^T b \\ b'_3 h_1^T b & - & b'_1 h_3^T b \\ b'_1 h_2^T b & - & b'_2 h_1^T b \end{pmatrix} \tag{5.3}$$

In Section 7.1.1.2, homographies will be used to accurately compute the image of the center of a tracked fiducial marker, by computing the transformation to the center of a 3D plane with respect to the corners of a tracked planar marker. Homographies will also be used in Zhang's Method [62] for camera calibration, in Section 8.1.1.3. In that chapter, we will expand on the concept of homographies, making the relationship between our model of cameras and planes transforming through 3D space more explicit.

5.2 Pinhole camera model

The commonly-used pinhole camera model is used to describe the projective transformation that a camera imposes on a scene as it projects the 3D objects in that scene down to a 2D image. It models pixels imaged by cameras in a computer vision system as rays of light running through the *aperture* of a pinhole camera, then

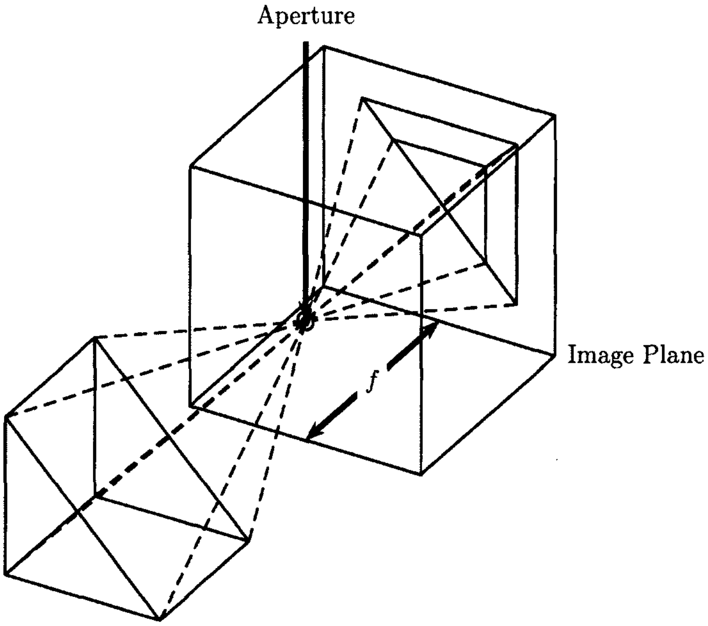


Figure 5.1: Light passes into a pinhole camera through its aperture, a small opening in the front of the camera, which can be constructed simply as a box that blocks out any other light. Because only the rays of light that pass through the aperture touch the image plane, the inverted image of the surfaces off of which this light is reflected appears on this plane. The distance between the aperture and the image plane is what is known as focal length, f. Though in a physical pinhole camera, images are inverted on the image plane and the image plane appears in the back of the camera, it is common in computer vision illustrations to place the image plane in front of the camera in order to simplify the illustration for the reader. This convention will be followed for the remainder of this document.

intersecting with a plane called the *image plane* upon which the sensor, commonly a charge-coupled device (CCD) rests. Figure 5.1 is an illustration of such a camera. The physical instrument described by the pinhole camera model has no lens, making it somewhat different from modern cameras, such as webcams and the robot, Nico's, Elmo camera heads. Modern cameras, however, can be accurately modeled via the pinhole camera model, with the exception that terms for nonlinear distortions induced by the lens must additionally be modeled.

To understand this model, it helps to understand the physical instrument upon which it is based, the pinhole camera (also known as a camera obscura). Consider a point light source emitting white light such that some of that light strikes an object imaged by a pinhole camera. White light contains many different wavelengths of light, with different wavelengths or combinations of wavelengths corresponding to different colors. Some wavelengths of light will pass through the object or be absorbed by it. Others will bounce off of the surface of the object. Because some of the wavelengths of light are absorbed, the remaining reflected wavelengths contain color information regarding the surface of the object. Consider now that there are potentially many light sources and surfaces off of which light can be reflected. As such, a sheet of white paper illuminated by the ambient light in a room does not generally appear to contain an image of the illuminated objects in the room, because light coming from many sources and reflected off of many surfaces covers it entirely. Suppose that we would like to transfer an image onto this sheet of paper by allowing only some of the light in the room to hit its surface. This light has been reflected off of the object or objects to be imaged, contains color information relevant to the object to be imaged, and should strike the sheet of paper at a unique point that is not illuminated by other light sources. Illumination from multiple light sources would cause the color of the designed light, which would produce the desired image, and that from other light sources to be blended. A pinhole camera allows for objects to be imaged by creating a dark space into which light can only enter through a narrow aperture. As such, each pixel in the camera corresponds to only a narrow beam of light running along the line perpendicular to the image plane and running through the aperture, referred to as the *camera center*. The pinhole camera model is a mathematical approximation of the behavior of such a camera.

We will use the pinhole camera model throughout this thesis. We will discuss the calibration of the parameters describing cameras modeled by the pinhole camera model later in this chapter in Section 5.2.1. In Section 7.1.2, we will use the pinhole camera model to reconstruct the position of 3D points from their 2D projections.

We will also discuss projecting 3D points to 2D in this section, as well as in Section 7.2, where we will project the position of the robot's forward-kinematic predictions of the position of its hand into its visual field. In Chapter 8, we will use the robot's body as a calibration target to simultaneously calibrate the kinematics of its arm and the model of its cameras. In Chapter 9, we will model the visual perspective of a mirror using the pinhole camera model.

5.2.1 The pinhole camera's projection

In a pinhole camera, the slope of the ray of light corresponding to a pixel on the image plane as it travels through the aperture of the camera determines the position of an illuminated pixel. As such, each pixel corresponds to one such slope and, equivalently, one such ray of light. This ray of light enters the camera after being emitted from a light source or being reflected off of an object, retaining the color information that it contains as it passes through the aperture. If we consider the 3D homogeneous point $P =< x, y, z, w >$, this slope is fully described by the ratios $(x : z, y : z)$, having 2 degrees of freedom. The pinhole camera model is applied as a 3×4 *projective matrix* called the *camera projective matrix*, which can be used to transform a 3D homogeneous point to a 2D homogeneous point. As we will see, the formulation of this matrix is based on transforming points in space with respect to the position and optical properties of this camera, then finding the slope of the line entering into the camera.

Equivalently, the algebraic way to think of a linear transformation is that it is the computation of the cosine of the points in their current basis to the basis vectors forming the new basis (scaled with the magnitude of these vectors). The basis vectors form the rows of the linear transformation. A geometric interpretation of the pinhole camera model is to treat the camera projective matrix as three planes (stacked on top of each other, forming the matrix) forming the basis of the projected space. The position of the 2D image point is computed based on the signed distance of the 3D scene point with respect to each of these three planes.

5.2.2 Ideal cameras & extrinsic parameters

Briefly consider *ideal cameras*; cameras whose intrinsic parameters (to be discussed in Section 5.2.3) can be expressed as the identity matrix, thus allowing for this discussion to ignore some of the mathematical complexities of the full pinhole camera model for the time being. As previously discussed, only the slope of the ray of light with respect to the aperture and the imaged 3D scene point, B, is necessary in order to determine the 2D location of the illuminated pixel, b'. The camera projective matrix, P, expresses this projection. The camera projective matrix for an ideal camera placed at the origin with an orientation at identity can be expressed as Equation 5.4. The image point corresponding to B as imaged by this ideal camera can be computed according to Equation 5.5.

$$P_{ident} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.4}$$

$$b' = P_{ident}B = \begin{bmatrix} x & y & z \end{bmatrix}^T = (x : z, y : z) \tag{5.5}$$

This simplified version of the pinhole camera model allows us to discuss the position and orientation of the camera in the scene independently of factors intrinsic to the camera itself, such as focal length. The parameters describing a camera's position and orientation are referred to as the *extrinsic parameters.* Briefly, they are:

R The rotation of points about the camera.

C The location of the camera center.

The camera's position is described in terms of its camera center, C. The camera center is a point through which we model all rays of light entering through the aperture of the camera as passing through. We can think of it as lying at the center of the aperture. The transformation expressed by the pinhole camera model transforms points in space with respect to the described imaging device, while simultaneously rank reducing in order to project these points from 3D to 2D. The parameter R refers to the rotation of points about the camera, whereas t describes the translation of points in space with respect to the camera. By Equation 5.6, we see that t can be thought of as the translational component of the transformation to bring the 3D scene points to be imaged into the coordinate system of the camera (with C lying at the origin), from which they will be imaged. We can express the transformation induced by an arbitrary ideal camera by modifying Equation 4.12 (the matrix form of a rigid transformation) to become Equation 5.7.

$$t = -RC \tag{5.6}$$

$$P_{ideal} = \left[\begin{array}{c|c} R & t \end{array} \right] = \left[\begin{array}{c|c} R & -RC \end{array} \right] \tag{5.7}$$

5.2.3 Intrinsic parameters

Whereas the extrinsic parameters describe the position and orientation of the camera in space, the *intrinsic parameters* are those that are intrinsic to the camera itself. They describe the optical properties of the camera, factors which affect the projection that the camera imposes upon the scene without consideration for position and orientation. Briefly, the five intrinsic parameters are:

α, β Focal length, their ratio accounting for non-square pixels.

γ Skew parameter, accounting for non-rectangular pixels.

(u_0, v_0) The principle point, where a line running perpendicular to the image plane and intersecting the camera center would intersect with the image plane.

In describing the orientation of the camera, its z-axis lies perpendicular to the image plane. The point at which a line running through the camera center and lying perpendicular to the image plane would intersect the image plane is referred to as the principle point. The principle point, parameterized as (u_0, v_0) in the pinhole camera model, represents the origin of the image coordinate system. It is also the point in space into which all points in the horizon appear to disappear.

The magnitude of the perspective effect, by which points appear to disappear into the principle point as their distance from the camera increases, is characterized by the focal length of the camera, as is the camera's magnification. The focal length of a pinhole camera f, physically, is the distance between the camera center and the image plane. The pinhole camera model has two parameters for focal length, α and β, with their ratio accommodating the possibility that the height and width of pixels may differ. The parameters α and β can be thought of as two separate parameters describing the height and width of the rectangular terms of each pixel (not accounting for the skew parameter γ) in the camera's receiver times focal length, f. An *orthographic projection* is one in which parallel lines do not appear to intersect in the horizon. The projection of an orthographic camera centered at the origin with identity rotation could be computed simply by taking the (x, y) Cartesian coordinates from their corresponding 3D triples (x, y, z). A pinhole camera with $f = \infty$ would produce orthographic projections.[1] Conversely, the effect of disappearing into the horizon becomes more noticeable for shorter focal lengths. As focal length becomes

[1] Such a camera would also infinitely magnify objects in front of it, thus being only able to image an infinitely small patch unless it had infinitely many pixels. This is just an illustrative metaphor in order to help the reader to understand perspective.

shorter, points at a given range away from the camera move closer to the principle point.

All of these parameters, in practice, are affected by the receiver that samples light entering into the pinhole camera, itself. Because we measure image space in terms of pixels, the size of pixels on the camera's CCD determine the units by which we measure space and the parameters of this model. Slight variations in manufacturing must also be accommodated by this model. The skew parameter γ accounts for the possibility that the receivers pixels may not be rectangular, instead forming parallelograms. As discussed above, the ratio of α and β accounts for the possibility of non-square pixels. In practice, we expect for α and β to be very close, with γ being near zero.

The intrinsic parameters are expressed as a matrix called the *camera intrinsic matrix*, Equation 5.8. The camera intrinsic matrix is applied to a projective matrix containing the extrinsic parameters in order to arrive at the camera projective matrix for a non-ideal camera, such as would be used to approximate a real camera, as in Equation 5.9. An ideal camera is one such that the camera intrinsic matrix is the identity matrix. Ideal coordinates, (x, y), are coordinates corresponding to the image prior to the application of the camera intrinsic matrix, such as would have been imaged by an ideal camera. Image coordinates, (u, v), are those which have been transformed by the camera intrinsic matrix.

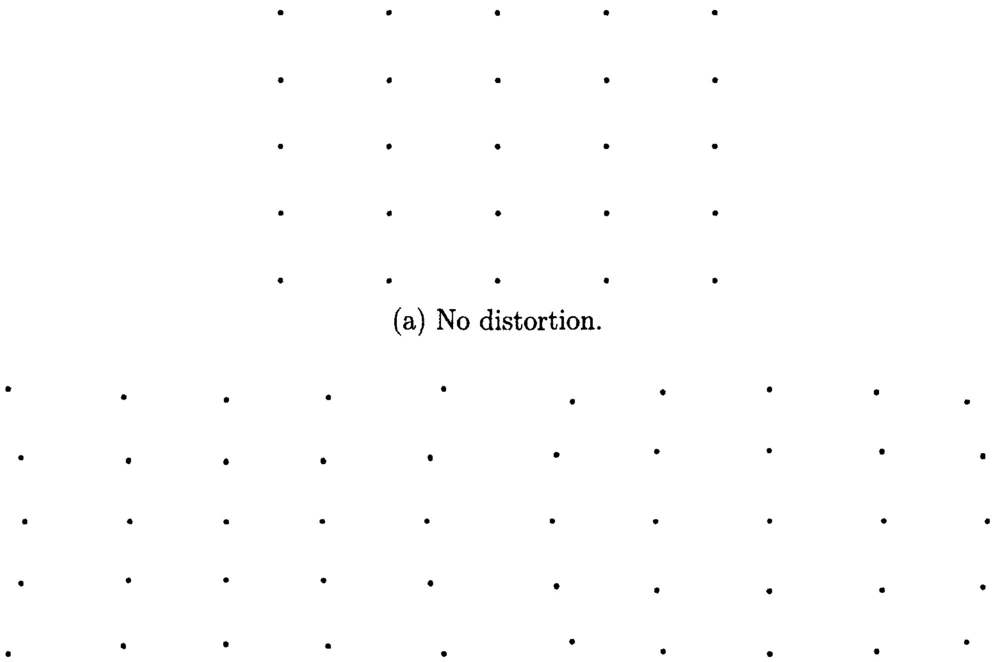
$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.8}$$

$$P = A \left[\, R \, \middle| \, -RC \, \right] \tag{5.9}$$

5.2.4 Lens distortion

In modern cameras, lenses are used instead of pinholes for the purposes of focusing light. The use of lenses has many advantages that are beyond the scope of this document, but, briefly, lenses can be shaped such that very long focal lengths are achieved without needing a physically large instrument, allowing for a high degree of magnification in a compact form factor. This comes at the cost of potentially encountering lens distortion, which occurs because of differences between the angles at which different rays of light are refracted through the lens. Because lenses are manufactured to be radially symmetric, typically the dominant factor of lens distortion occurs as a function of distance away from the center of the lens. This is called *radial distortion*, and is evidenced as images of straight lines appearing to bow inward (pincushion distortion, Figure 5.2b) or outward (barrel distortion, Figure 5.2c) from the center of the lens (which we model as the camera center, corresponding to the principle point). This nonlinear distortion violates the otherwise linear nature of the pinhole camera model and must be corrected.

Nico's vision system is designed to model three terms of radial distortion as a third order polynomial function of distance away from the principle point, with parameters k_1, k_2, k_3 describing this distortion. Distortion applied to ideal image coordinates is computed by Equation 5.10. For the purposes of this discussion, we will use the coordinates (x, y) to refer to undistorted, ideal coordinates; coordinates such that they were imaged by an ideal camera, with A of identity and without radial distortion induced by a lens. To these coordinates, the implicit parameters of the camera can be applied by multiplying by A against their homogeneous representation. When we do so, we refer to their undistorted, non-ideal equivalents as (u, v). When lens distortion is applied to this model, we refer to the equivalent points that are subject



(a) No distortion.

(b) Pincushion distortion.

(c) Barrel distortion.

Figure 5.2: Illustration of a grid of points as imaged through a camera, in order to illustrate radial lens distortion. Subfigure (a) shows an undistorted grid of points. Subfigure (b) illustrates pincushion distortion, in which lines appear to bow inward towards the principle point. Subfigure (c) shows the opposite effect, barrel distortion, in which lines appear to bow outward, away from the principle point.

to distortion as $(\dot{x}, \dot{y})$ and $(\dot{u}, \dot{v})$, respectively.

$$\begin{aligned}\dot{x} &= x + x[k_1 * (x^2 + y^2) + k_2 * (x^2 + y^2)^2 + k_3 * (x^2 + y^2)^3] \\ \dot{y} &= y + y[k_1 * (x^2 + y^2) + k_2 * (x^2 + y^2)^2 + k_3 * (x^2 + y^2)^3] \qquad (5.10)\end{aligned}$$

Variations of this model are common in the computer vision literature [55, 62, 63, 64, 65, 66]. Some researchers have observed experimentally that the first two terms of radial distortion often dominate the model. As such, modeling other distortion terms may be unnecessary and simply lead to numerical instability [64, 62]. Others have noted marked improvements in performance through the modeling of tangential distortion [66]. In the case of Nico's cameras and this software, the software was written for three terms of radial distortion and two terms of tangential distortion, following the model in OpenCV [63]. Empirically, this was determined to be unnecessary for Nico's cameras,[2] and so only the first three terms of radial distortion are used in these experiments.

The application of this model in Nico's computer vision software is somewhat different from that in OpenCV [63] and many other computer vision systems in two ways.

First, it is typical to simply factor in offsets such as (u_0, v_0) into the formulas describing camera distortion. Instead, Nico's computer vision software computes camera distortion with respect to ideal coordinates, then applies the camera intrinsic parameters by multiplying by A. In the case of projecting a 3D point to its distorted 2D coordinates, the procedure is to apply Equation 5.9, then Equation 5.10, then to multiply by A as in Equation 5.8.

[2] Whether or not tangential distortion is modeled is passed as a flag to Nico's custom camera calibration software.

The second difference comes in the form of how points are undistorted by the vision system. It is common to pre-compute an undistortion map, a nonlinear transformation from one 2D image to another, and then apply this undistortion map to the image sampled by the robot's stereo cameras prior to performing other computer vision tasks. In OpenCV [63], this can be performed via the **remap()** function. While this speeds up the undistortion process and allows undistorted images to be treated similarly to other images sampled by the camera, the blending and interpolating of pixels in order to arrive at the undistorted image results in information loss in the undistorted image. Some implementations work around this limitation by scaling up the undistorted image, which can slow down computer vision routines that occur after undistortion while still introducing imprecision into image processing. In the case of Nico's stereo vision system, a different approach is taken. Undistorted points are treated individually via nonlinear optimization over the camera distortion model. Candidate undistorted ideal points are seeded from 2D data based on factors such as the marker tracked by the robot. The undistorted points are then optimized as a least-squares fit to the distortion model using LevMar [67], an implementation of the Levenberg-Marquardt nonlinear optimizer.

Though performing such individual optimizations may be slower than applying a distortion map to undistort the image, performance is acceptable due to to the fact that the system only undistorts a small subset of the pixels in the image, representing tracked points in space. By performing the undistortion process in this way, the robot's vision system has two primary advantages over similar systems. First, the vision system does not lose information during a remapping process. Second, the robot is able to take advantage of potential sub-pixel accuracy in tracked positions, as in the case of fiducial tracking. These two factors contribute to a high degree of accuracy in not only the localization of 2D and 3D points, but also to accuracy in its calibration, as this process is also followed during camera calibration.

5.3 Epipolar geometry

Epipolar geometry describes the projective relationship between two cameras. Consider a pair of cameras used for stereo vision. If one camera were to take a picture of the other camera's camera center, the image of the other camera's camera center would be referred to as the epipole, as in Figure 5.3. For a pair of stereo cameras, the epipoles can be computed as in Equation 5.11, where e_L and e_R are the epipoles, P_L and P_R are the projective matrices, and C_L and C_R are the camera centers for the left and right cameras, respectively.[3]

$$\begin{aligned} e_L &= P_L C_R \\ e_R &= P_R C_L \end{aligned} \tag{5.11}$$

Epipolar geometry is frequently represented in the form of the *fundamental matrix*, F. The fundamental matrix is a 3×3 matrix of rank 2. It is often described in terms of the *epipolar constraint*, as in Equation 5.12. The epipolar constraint can be thought of in terms of *epipolar lines*. Epipolar lines are lines appearing in the opposite camera of a stereo pair, running through that camera's epipole and the corresponding matched image point. The epipolar line l_L, appearing in the left camera's image plane, corresponds to the point p_R appearing in the right camera's image plane. It runs through the corresponding protected point, p_L, in the left camera and through the left camera's epipole, e_L. Returning to Figure 5.3, we can think of l_L

[3] In order to simplify this discussion, we'll refer to the cameras in Nico's stereo vision system as the left and right cameras. We can equivalently think of the left camera as the "first" camera and the right camera as the "second" camera. In some other treatments, it is common to refer to variables describing the second camera using a prime symbol, where the equivalent projective matrices would be P and P', and so forth.

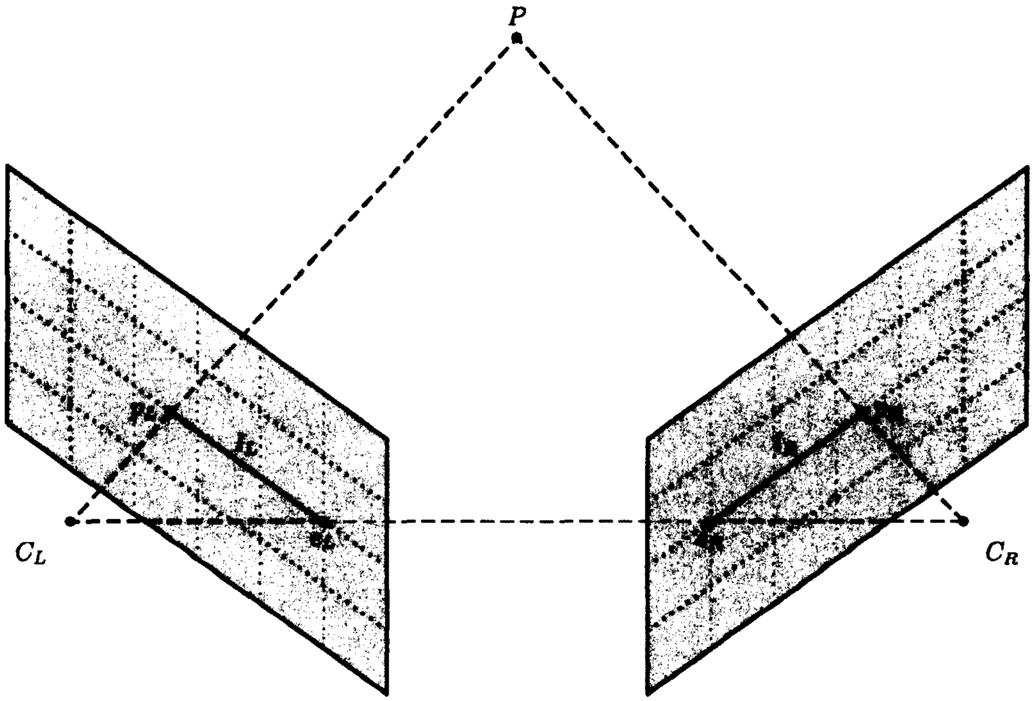


Figure 5.3: The epipolar points e_L and e_R appear on the left and right image planes, respectively. They represent images of the camera centers, C_L and C_R, of opposing cameras in the stereo pair. The epipolar point e_L corresponds to the image point as if the left camera took a picture of the right camera's camera center, C_R. Image points p_L and p_R correspond to the images of P in the left and right cameras, respectively. The epipolar lines running through p_L and e_L, and p_R and e_R, respectively, are images of the rays of light running through the opposing camera center, to the opposing image point, to the scene point, P, for each camera. The epipolar constraint indicates that any image point p_R corresponding to an image point p_L must lie upon the corresponding epipolar line, and vice-versa.

as the image of the ray of light entering the right camera and intersecting the image plane at the corresponding 2D image point, p_R. As such, the epipolar constraint states that for a pair of cameras the image of a point imaged by the one camera is constrained to lie along a line in the other.

$$p_R^T F p_L = 0 \tag{5.12}$$

$$l_R = F p_L = 0 \tag{5.13}$$

The epipolar constraint is frequently used in order to constrain the search for matched points in the left and right images in a stereo pair. Finding this matched pair of points is referred to as the *stereo correspondence problem.* The goal is to locate 2D image points corresponding to the same 3D scene point. Matched image points can, once found, be used to reconstruct the 3D position of their corresponding scene point.

The stereo reconstruction process used in Nico's stereo vision system is described in Section 7.1.2 and involves computing the optimal position of a reconstructed 3D point such that it produces the measured projections. In Nico's vision system, rather than perform a search for matched pairs of image points, the positions of the centroids of tracked markers as they move through space are used. The system's epipolar geometry is used in order to recover extrinsic parameters of the projective matrices describing the left and right cameras, yielding the projective matrices used in this reconstruction process. These matrices encode factors describing the vision system, such as the baseline[4] between the two stereo cameras and their orientation with respect to each other.

A matrix related to F is the *essential matrix*, E, which corresponds to F for

[4] The translation between the two cameras in a stereo pair.

ideal cameras, Equation 5.14.[5] If the left camera is placed at the origin with identity rotation, then the parameters R and t correspond to the same parameters in the right camera's projective matrix, P_R. The relationship between F and E is stated in Equation 5.15. By this relationship it is possible to recover the position and orientation of the right camera with respect to the left camera, as will be discussed in Section 8.1.2. It is also possible to track the motion of a camera in a scene, as in the case of visual odometry.

$$E = [t]_\times R \tag{5.14}$$

$$F = A_R^T E A_L \tag{5.15}$$

5.4 Summary

This chapter has provided a brief overview of several computer vision concepts which will be used throughout the remainder of this document. It starts with a simple example of a 2D projective transformation known as a homography. The nature of the projection imposed by a homography is simple to understand and provides us springboard for discussion of the pinhole camera model, which is used to describe the projection from 3D to 2D performed by real cameras by modeling them as the classical camera obscura. This relationship was discussed in-depth with frequent reference to the physical processes being modeled. The modeling of lens distortion was additionally discussed. We concluded this discussion with a discussion of epipolar geometry, the projective relationship between two cameras in a stereo pair.

Knowing the parameters of the pinhole camera model for a stereo pair of cameras

[5] The notation $[t]_\times$ here refers to the cross product matrix of t. A cross product matrix is a skew-symmetric matrix such that multiplication of the matrix and a vector is equivalent to taking the cross product of the vector used to formulate the matrix and the multiplied vector.

enables the model to be used for the process of *stereo reconstruction*, which will be used in Chapter 7 to reconstruct the 3D position of the robot's end-effector. This will become the basis for the techniques which will allow Nico to infer its kinematics through observations made directly by its stereo vision system. The process of determining these parameters is referred to as *camera calibration*, and will be discussed at greater length in Chapter 8, where we will enable the robot to enhance its own camera calibration by using its own body as a camera calibration target. The specifics of Nico's camera calibration software can be found in Appendix B.

Chapter 6

Kinematic inference

This chapter will focus on the kinematic learning process, ignoring the complexities of stereo vision. In order to test the system in a manner that is free from possible errors induced in the stereo reconstruction process, the kinematic inference techniques introduced in this chapter are tested using a Vicon MX motion capture system. This places the burden of accurately measuring end-effector position onto the motion tracker, rather than on the custom software of the stereo vision system. In the next chapter, combining this capability with stereo vision will allow for the robot to infer its kinematics through observations made by it stereo vision system.

Several papers have been devoted to the subject of learning robot kinematics. Hersch, Sausser, and Billard [36] present a robot which learns the parameters of a model describing the kinematic chain of its arm. Martinez-Cantin, Lopes, and Montesanto [35] present a similar model to Hersch et al. [36], improving on the number of samples required for training by several orders of magnitude, through the use of better optimization techniques and active learning. Sturm, Plagemann, and Burgard [34] present a technique utilizing a Bayesian representation of kinematic chains.

The work of Hersch et al. [36] and Martinez-Cantin et al. [35], as well the work presented in this chapter, are *kinematic calibration* techniques. Hollerbach and Wampler [68] provide a good overview of kinematic calibration techniques, and would classify all three methods as *open-loop methods* due to the use of an external metrology system. The work presented in this chapter differs from the approaches presented by Hersch et al. [36] and Martinez-Cantin et al. [35] in that it utilizes a different representation of the kinematic chain, achieves better spatial resolution by an order of magnitude, and requires fewer training samples. The presented system is initialized using what Hollerbach and Wampler [68] refer to as a *screw-axis measurement method.* This method is a derivation of *circle point analysis*, in which it is observed that the motion of the end-effector of a single revolute joint that is undergoing rotation traces a circle in 3D space. From this circle, the parameters describing the joint can be inferred. By rotating each joint in a kinematic chain in isolation, this method can be extended to entire chains of revolute joints.

6.1 Nico's kinematic model

For the purposes of this discussion, the mechanics describing the motion of a robot's arm can be broken down into two sub-disciplines. These are *kinematics* and *dynamics.*

The kinematics of a mechanism is a description of the motions afforded to it. For instance, a revolute joint is able to rotate, a ball can roll in any direction, and a robot's arm can move to various points in space based on a collection of joint angles. A collection of connected joints, such as those in a robot's arm, forms a *kinematic chain*, its entire range of motion being determined by the range of transformations allowed by each joint and their spatial relationships to each other in the chain. The *forward-kinematic* problem for a chain of revolute joints is that of determining where such a mechanism's *end-effector*, such as the hand of a robot, is in space from a vector of *joint angles*, the angle to which each joint in the kinematic chain is turned. The *inverse-kinematic* problem is that of determining the vector of joint angles required to place the robot's end-effector into a given position in space. A *kinematic model* is the model used to describe the kinematics of a given mechanism. In this work, we model the kinematics of a robot's arm as a series of transformations affecting a single tracked point on the hand.[1]

The model used to describe Nico's motion does not consider dynamics. Dynamics is the description of forces acting upon a mechanism. Through the combination of kinematics and dynamics, we can determine how a mechanism moves, responds to forces acting upon it, and exerts force. The robot, Nico, uses servo control, which simplifies its control by allowing the positioning of its kinematic chains to be described in terms of joint angles. The exertion of force is handled internally by the controller. For this reason, this model disregards dynamics, modeling only the system's kinematics.

6.1.1 Representing kinematics

Nico's kinematic model utilizes the *Denavit-Hartenberg parameters* [1] as its representation of joints and their relationship to each other. The Denavit-Hartenberg parameters are a kinematic modeling convention which represents the rotational axes of revolute joints as lines in space, as in Figure 6.1, where z_{i-1} and z_i are the rotational axes for two neighboring joints. These axes are described with respect to a

[1] This single tracked point is the centroid of a marker attached to the back of the robot's hand, when using fiducial tracking. It is red colored tape wrapped around the tip of the finger in the case of color blob detection. It is a reflective ball attached to the robot's wrist in the case of the Vicon MX motion tracker. In all three cases, the mounting of the marker was chosen as the surface to which it most readily adheres.

line running normal to both, referred to as the *common normal*, x_i. The description of adjacent joints with respect to their common normal allows for each joint to be described in terms of the minimal possible number of parameters: four [1]. These parameters are used to describe the transformation between two coordinate frames. The first is that in which the joint lies and the second is at the endpoint of the joint's motion. For the terminal joint in a kinematic chain, this would be the end-effector. The labelings of the subscripts of these variables have taken two forms in the literature, easily distinguished by whether the x_i axis refers to the x axis of the coordinate system in which the joint lies prior to its motion through θ_i, making the common normal between joint i and $i + 1$ x_{i+1}, or whether this common normal is referred to as x_i. In this presentation of the material, x_i denotes the x axis upon completion of a joint's rotation. Joint i rotates about a line parallel to z_{i-1}, with z_i being the new z axis upon completion of the joint's full transformation.

The parameters, illustrated in Figure 6.1, are:

θ_i The joint angle. Equivalently, the angle between x_{i-1} and x_i about z_{i-1}.

r_i The distance between z_{i-1} and z_i, measured along x_i.

α_i The angle between z_{i-1} and z_i.

D_i The distance between x_i and x_{i+1} along z_i.

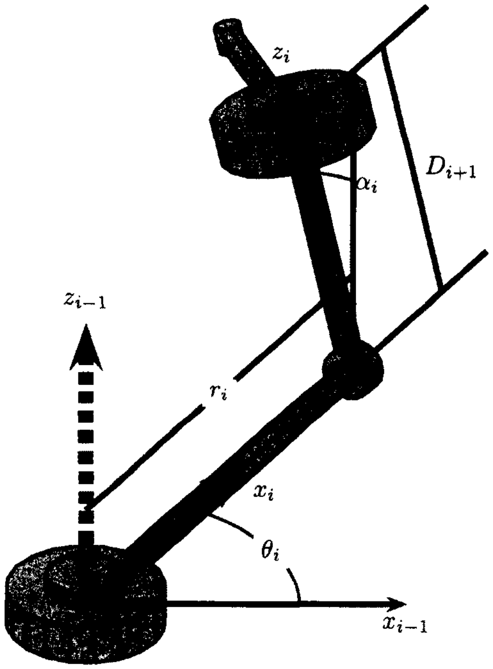


Figure 6.1: The Denavit-Hartenberg convention describes joints with respect to their rotational axes, represented as lines in 3D, z_{i-1} and z_i. Adjacent rotational axes in a kinematic chain are described with respect to the line running normal to both, their common normal, x_i. The variable, r_i denotes the distance between the adjacent z_{i-1} and z_i axes, whereas D_i measures the distance between neighboring x_i and x_{i+1} axes.

The transformation performed by a single joint is represented as a matrix, M_i. The transformation can be decomposed into four operations. Going from the origin of the joint's reference frame to that of the end-effector, these are:

1. Translation from the end of the previous joint to the center of rotation of joint i along the z_i axis, of length D_i, as in Equation 6.1.

$$T_{D_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & D_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

2. Rotation about the z_{i-1} axis. This is the rotation of the joint, of magnitude θ_i, as in Equation 6.2.

$$R_{\theta_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

3. Translation across the common normal between the lines representing the axes z_{i-1} and z_i of magnitude r_i, as in Equation 6.3.

$$T_{r_i} = \begin{bmatrix} 1 & 0 & 0 & r_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.3}$$

4. Rotation, α_i, about the common normal, x_i, from the orientation of z_{i-1} to z_i, as in Equation 6.4

$$R_{\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.4}$$

The matrices from Equations 6.1, 6.2, 6.3, and 6.4 can be composed together as in Equation 6.5 to represent the transformation from the coordinate frame whose origin is based at the end-effector of joint i to the origin of the joint's reference frame, with the resultant transformation appearing as in Equation 6.6.

$$M_i = T_{D_i} R_{\theta_i} T_{r_i} R_{\alpha_i} \tag{6.5}$$

$$M_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & r_i\cos\alpha_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & r_i\sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i & D_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.6}$$

The position of the end-effector is determined by multiplying a chain of M_i matrices together, with one such matrix for each joint in the kinematic chain. The transformation M_0 represents the transformation from the *inertial reference frame*, the base coordinate system in which the kinematic chain is situated, to the frame in which the first joint lies. This transformation can be computed either to be represented as a joint, using the Denavit-Hartenberg parameters, or simply computed as the rigid transformation from the inertial frame to the first joint. In the case of Nico's kinematic inference system, the latter computation is performed, because the numerical optimizer which fits the system's parameters performs better in this case. Computing the position of the end-effector of the kinematic chain, then, can be performed according to Equation 6.7.

$$M_0 \ldots M_n [0,0,0,1]^T \tag{6.7}$$

6.1.2 Encoder offset and gear reduction

The model used to describe Nico's kinematics adds two more terms for each joint, *encoder offset* and *gear reduction.*

Each of Nico's joints is actuated with a motor/encoder/gearhead combination, as discussed in Section 3.1.2. The measurements provided by each encoder are measured in "ticks," or "counts." The number of "counts per turn" an encoder has determines how many ticks are passed for each full revolution of the encoder. The encoder's measurement is performed prior to gear reduction in the gear train, for the reason that the encoder's resolution is then also affected by gear reduction. As such, in the standard setup for a DC electric motor the resolution of the measurement of the joint angle for the motor/encoder/gearhead combination becomes finer with greater gear reduction. Gear reduction occurs in Nico's kinematic chain in addition to that performed by the motor's gearhead because one of the cable-driven joints is additionally gear-reduced, Figure 6.2.

The positional measurement that the robot's encoders provide is relative to a software-set zero point. This zero point can change any time the robot is started, and, therefore, must also be calibrated each time this occurs. Additionally, because the robot's kinematic model is inferred through external measurements, it is likely that the zero of each θ_i will disagree with the corresponding, physical zero of the encoder.

To account for this, the present model adds two parameters to this standard set. Since the zero point of the robot's encoder is unlikely to match the corresponding θ_i, the offset between the two is represented as $\hat{\theta}_i$. Gear reduction, which must also be accounted for is represented as G_i. The joint angle passed to the robot's motor, $\breve{\theta}_i$, then, is computed according to Equation 6.8.

$$\breve{\theta}_i = \hat{\theta}_i + G_i \theta_i \tag{6.8}$$

Figure 6.2: The joint in Nico's elbow which rotates such that it folds the two primary linkages of the arm together is gear reduced through its cable-driven mechanism.

6.2 Kinematic inference

The algorithm developed for inferring kinematic chains composed of revolute joints proceeds in two steps. The first step is to take an initial estimate of the structure of the kinematic chain via a fast set of simple approximations. This step is a version of circle point analysis developed for this thesis, and is described in Section 6.2.1. The second step is to perform a nonlinear refinement of the kinematic chain by minimizing the squared distance between the end-effector position predicted by the forward-kinematic model and that measured by the motion tracking system, as described in Section 6.2.3. This method is inspired by computer vision techniques for camera calibration, which frequently start by taking an initial estimate of a camera's calibration and then refining this estimate.[2] This inference method assumes that the kinematic chain to be inferred is composed of a known, fixed number of revolute joints and that their connections to each other are known. What is inferred is the values of parameters of the Denavit-Hartenberg representation of these joints.

Inference of the kinematic chain proceeds from two datasets. One is a structured dataset of the circular motion of isolated joints in the kinematic chain. The second operates over a dataset of general motion, which can be randomly generated. In practice, the robot's kinematic calibration is first seeded with an estimate derived from circle point analysis. This estimate is then refined as a nonlinear optimization over a dataset consisting of those arm poses used for circle point analysis, concatenated with a set of random arm poses.

[2] For example, Zhang's method [62] takes an initial estimate of a camera's calibration using the homography constraints, then proceeds to refine this estimate by reconstructing the poses of calibration chessboards in a technique related to photogrammetric calibration.

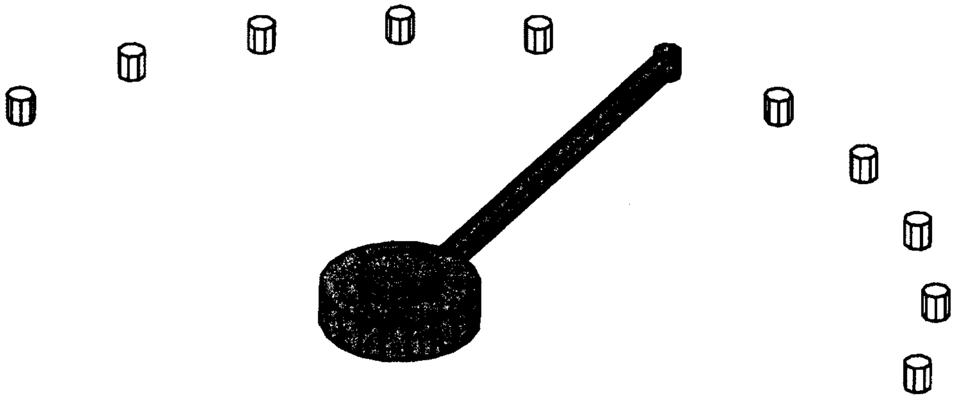


Figure 6.3: The collection of poses that a single revolute joint passes through as it rotates lies along an arc of circle.

6.2.1 Circle point analysis

Circle point analysis (CPA) refers to techniques proceeding from the observation that the path traced by the end-effector as a single joint rotates lies along an arc of circle, as in Figure 6.3.

6.2.1.1 The single joint case

If this joint were the only joint in a kinematic chain, or if one were only concerned with modeling this single joint, knowing that the motion of all other joints in the chain would be held static, then fitting a circle to the path traced by the end-effector as it moves through space would be sufficient to fully parameterize this system, as in Figure 6.4. The rotational axis, the z_{i-1} axis under the Denavit-Hartenberg convention [1], is treated as a line perpendicular to the plane in which this circle lies, running through its center. The parameter r_i can then be measured as the radius of the circle. The α_i can safely be assumed to be zero, because it describes the relationship of the joint to subsequent joints in the kinematic chain. The parameter D_i relates this joint to prior joints in the kinematic chain. As we will see in Section 6.2.1.2, D_i is computed with respect to the prior joint in the kinematic chain. In the case of a single joint, we can assume D_i to be zero, its value having been captured by the transformation from the inertial reference frame to this joint. This places the start of this joint's transformation at the center of the traced circle. The computation of the inertial transformation involves translating to this center of rotation and rotating the coordinate system such that the axis z_{i-1} is the z axis.

Note that each pose of the end-effector has a unique θ_i. The corresponding x_i for each pose can be measured as the vector running from the center of the circle to the end-effector. Thus, each θ_i can be computed as according to Equation 6.9, where x_{i-1} and x_i are expressed as unit vectors, with the x_{i-1} vector resulting from the transformation from the inertial frame.

$$\sin^{-1} x_{i-1} \cdot x_i \tag{6.9}$$

Because three points are sufficient to define a circle, they are sufficient to identify the kinematics of a single joint via this method. Counting parameters reveals that a sample of 3 is also the minimal number of poses required unless we are able to incorporate encoder readings into this measurement. Encoder readings are not used to determine the Denavit-Hartenberg parameters of the joint in this analysis, as the system assumes that gear reduction is initially unknown. Encoder readings are instead used to determine gear reduction and encoder offset once an initial model is inferred from the sample of 3D end-effector positions.

6.2.1.2 Extending to multiple revolute joints

The case of kinematic chains with more than one revolute joint is complicated by the fact that the position of the end-effector is determined by the state of multiple joints interacting with each other, as in Figure 6.5. Though the circle labeled c_i would

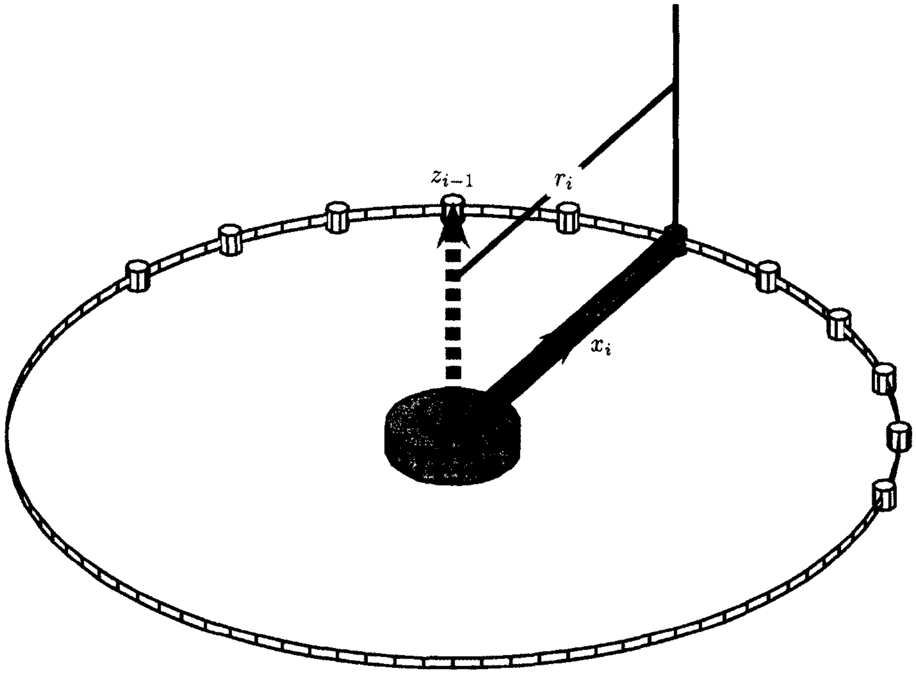


Figure 6.4: The circle traced by the motion of the end-effector of a single joint as it rotates captures the parameters describing that joint. Because α_i and D_i describe the relationship of the joint to other joints in a kinematic chain, their values can be assumed to both be zero in the single-joint case. The radius of the circle traced by the joint is r_i, with the rotational axis of the joint z_{i-1} lying perpendicular to the plane of the joint. The corresponding line in the Denavit-Hartenberg representation running through the center of the traced circle.

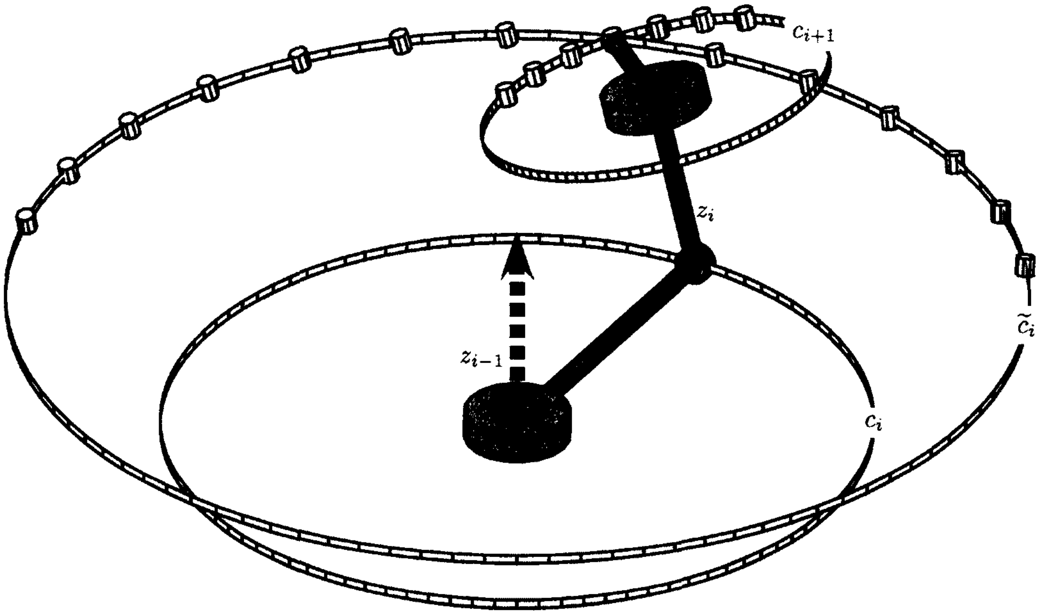


Figure 6.5: Circle c_{i+1} is traced by the motion of the end-effector as joint $i+1$ rotates. As joint i rotates, however, the circle $\tilde{c}_i$ is traced, due to the orientation of joint $i+1$. If the end-effector were attached directly to joint i, lying at the intersection if x_i and z_i, its motion would trace an arc of circle lying on c_i, instead. The z_{i-1} axis lies perpendicular to both c_i and $\tilde{c}_i$, running through the centers of both circles. The z_i axis is similarly related to joint $i+1$, running through the center of c_{i+1}. Section 6.2.1.2 describes how the Denavit-Hartenberg parameters [1] describing these two joints and their relationship to each other can be inferred through measurements relating to these traced circles.

describe the motion of joint i in the case in which there were no other joints in the kinematic chain, an arc of circle lying along $\tilde{c}_i$ is produced when the joint is moved in isolation, due to the configuration of joint $i+1$.

Making use of the intuition used in Section 6.2.1, that the motion of the end-effector traces a circle in space as a single joint rotates in isolation, requires knowledge of the transformation performed by intervening joints between the measured joint and the end-effector. We extend this intuition to two or more joints by fixing the relationship between the measured circles. This is accomplished by measuring the arcs of circle traced by the end-effector as it moves through space with respect to a single home position which lies at the intersection of the circles measured for each joint. This allows for the relationship between these joints to be determined from their respective rotational axes (z_{i-1} for each joint) and the centers of the circles fit to the trajectory that the end-effector follows as each joint rotates.

As three points are required to uniquely identify a circle, a circle is sampled for each joint by moving it into at least two poses away from the predetermined home position. These poses are used to reconstruct the circle uniquely identifying each joint with respect to the rest of the chain. A summary can be found in Algorithm 1. In this implementation, circle fitting is performed using an in-house implementation based on the technique from the NIST Algorithm Testing System [69]. For details, see Appendix A: Circle fitting implementation.

Algorithm 1 Circle Point Analysis

```
Determine an initial, home position for the kinematic chain
for $i = 1$ to $n$ where $n$ is the number of joints in the chain do
    Move kinematic chain to home position
    Move joint $i$ through at least 2 additional positions along its arc of motion
    Fit a circle to the set of 3 or more sampled points for this joint.
end for
return The set of measured circles.
```

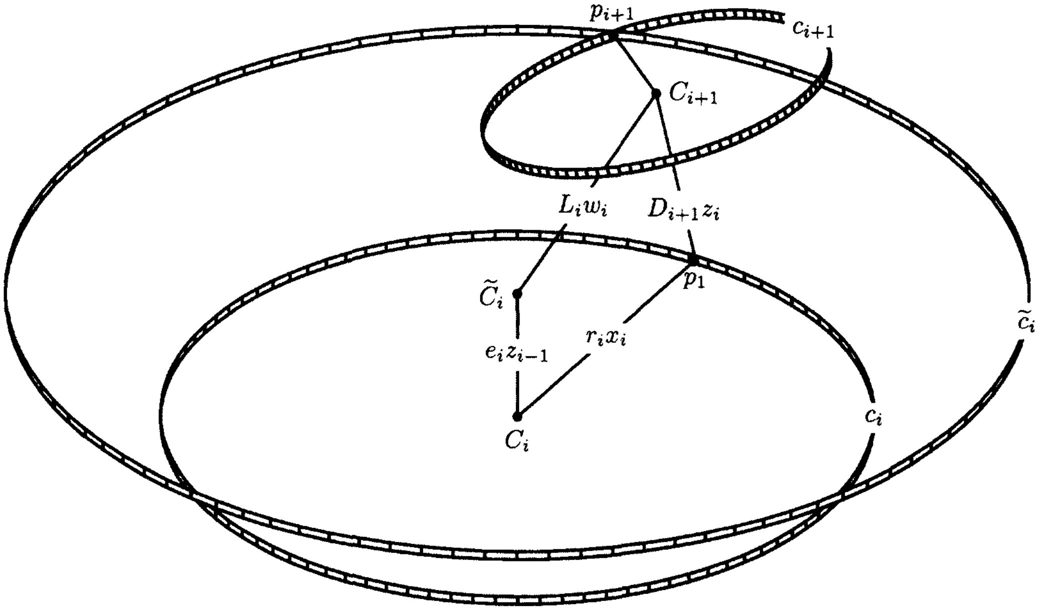


Figure 6.6: This figure illustrates the linear system solved in Section 6.2.1.2 in order to determine the parameters describing a kinematic chain of multiple revolute joints. The common normal, x_i lies perpendicular to both z_{i-1} and z_i (in the case of parallel rotational axes, any common normal lying between the two can be chosen). The centers of the circles c_i, $\tilde{c}_i$, and c_{i+1} have been labeled C_i, $\tilde{C}_i$, and C_{i+1}, respectively. The unit vector L_i runs parallel to $C_{i+1} - \tilde{C}_i$. If L_i, z_{i-1}, x_i, and z_i are expressed as unit vectors, the remaining parameters describing the kinematic chain, r_i, and D_{i+1}, can be found by solving the linear system defined in Equation 6.13 for the lengths of the corresponding line segments in this figure, w_i, e_i, r_i, and D_{i+1}, respectively. Though there are 4 parameters and the system defined in Equation 6.13 forms a rank 3 matrix, we are able to fully constrain this system through the observation that $w_i = ||C_{i+1} - \tilde{C}_i||^2$, allowing us to normalize the solution to the linear system against this constant. It is important to remember that all of these parameters are estimated from measurements of the position of a single point on the robot's end-effector as it moves according to Algorithm 1. This algorithm generalizes to kinematic chains of length greater than 2 because all relevant parameters can be inferred from the collection of traced circles from the axes lying perpendicular to them (corresponding to the z axes of the system), their centers, and their radii, and because their spatial relationship to each other is fixed by passing through the home position (p_{i+1} in this illustration).

To clarify discussion of this material, the following variables are added to the parameters from the Denavit-Hartenberg convention, which describe a system illustrated in Figure 6.6:

p_i The endpoint of the transformation performed by joint i.

$\widetilde{c}_i$ The circle fit to the set of positions that the tracked point on the robot's end-effector passes through as joint i rotates in isolation, during Algorithm 1.

$\widetilde{C}_i$ The center of the circle $\widetilde{c}_i$.

c_i The circle that would be fit to joint i's motion during the measurement performed Algorithm 1 if joint i were the terminal joint in the kinematic chain with the end-effector placed at the intersection of x_i and z_i.

C_i The center of the circle c_i.

L_i The unit vector parallel to $\widetilde{C}_i - C_{i+1}$.

w_i The distance from $\widetilde{C}_i$ to C_{i+1}.

e_i The distance from $\widetilde{C}_i$ to C_i.

The structure of each joint, i, which is part of a kinematic chain, can be deduced from its interaction with its respective subsequent joint in the chain. The terminal joint can be treated as the single joint case, with the remainder of the chain bringing the coordinate system into the frame of that joint, or simplified by using the multi-joint solution with a parallel z_i axis to its z_{i-1} axis, and treating the system as though p_i and C_{i+1} are equal. This structure can be uncovered through the following relationships:

$\widetilde{C}_i$ The point $\widetilde{C}_i$ lies at the center of the circle $\widetilde{c}_i$, the circle traced by the end-effector as joint i rotates.

z_i The z_i axis is known, because it was inferred during the process of inferring the joint at $i+1$. For convenience, z_n is parallel to z_{n-1}, where n is the number of joints in the kinematic chain.

z_{i-1} The z_{i-1} axis lies perpendicular to the plane in which the circle $\widetilde{c}_i$ lies. For convenience, $z_{-1} = (0, 0, 1)$, where z_0 describes the z axis at the end of the inertial transformation, to the base of the first joint. The first joint's axis of rotation lies parallel to z_0.

x_i The x_i axis runs as the normal between z_{i-1} and z_i. In the case of parallel axes, the choice is constrained only to the set of vectors lying normal to both. For convenience, the software would choose x_i such that it runs through $\widetilde{C}_i$, making $\widetilde{C}_i$ equal to C_i.

α_i **and** θ_i The angles α_i and θ_i are found as inverse sine of the dot product over the unit vectors between the relevant axes, $z_{i-1} \cdot z_i$ and $x_{i-1} \cdot x_i$, respectively.

r_i **and** D_{i+1} The relationship between $\widetilde{C}_i$ and C_{i+1} is described by Equation 6.10, allowing the parameters r_i and D_{i+1} to be determined by solving the linear system described in Equation 6.13.

Equation 6.10 describes the relationship of these parameters to each other. It shows that we can trace a path from $\widetilde{C}_i$ to C_{i+1} down z_{i-1}, across the common normal, and then up z_i. The path can then return to $\widetilde{C}_i$ from C_{i+1} across the vector L_i from Equation 6.12, causing the total distance traversed to be zero and allowing the system to be solved as the right null space of the 3×4 matrix described in Equation 6.13. The magnitudes, w_i, e_i, r_i, and D_{i+1} correspond to the distance traveled along the unit vectors L_i, z_{i-1}, x_i, and z_i, respectively. Note that the linear system is under-determined, consisting of a rank three matrix used to determine four constants. As such, this system is only accurate up to a scale factor. However, because w_i can be directly measured according to Equation 6.11, the right null space can be normalized by scaling the result such that w_i is equal to the value determined via direct measurement, allowing for this system to be solved and providing the parameters r_i and D_{i+1}.

$$\widetilde{C}_i + e_i z_{i-1} + r_i x_i + D_{i+1} z_i = C_{i+1} \tag{6.10}$$

$$w_i = ||\widetilde{C}_i - C_{i+1}||^2 \tag{6.11}$$

$$L_i = \frac{\widetilde{C}_i - C_{i+1}}{w_i} \tag{6.12}$$

$$\begin{bmatrix} L_{i_x} & z_{i-1_x} & x_{i_x} & z_{i_x} \\ L_{i_y} & z_{i-1_y} & x_{i_y} & z_{i_y} \\ L_{i_z} & z_{i-1_z} & x_{i_z} & z_{i_z} \end{bmatrix} \begin{bmatrix} w_i \\ e_i \\ r_i \\ D_{i+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.13}$$

To see that this solution is optimal in the number of samples required, consider that a minimum of two additional arm poses must be used in order to find the parameters for each additional joint. Each pose provides three measurements, but also has a unique θ_i, which must be determined in order to accurately describe the pose. As such, a maximum of four constraints is acquired from these two additional poses. The three geometric parameters of the joint, r_i, D_i, and α_i account for three parameters described by these four constraints. With six constraints (each point providing an x, y, and z for its position) and six unknowns for each additional joint (four Denavit-Hartenberg parameters when the joint is in its home position, encoder offset, and gear reduction), this system is minimal in the number of points required to determine the kinematics of the system.[3]

6.2.2 Offset and gear reduction

Now that a model of the robot's kinematics has been inferred, it is necessary to determine the relationship between this model and the encoder readings from the real robot. This allows the model to be used for forward-kinematic estimates of the pose of the physical device.

Estimates of the parameters $\hat{\theta}_i$ and G_i are determined by minimizing the squared difference between the θ_i estimates yielded by circle point analysis (CPA) and the product of Equation 6.8 for joint angles passed to the robot during point sampling. The presented implementation uses the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [70, 71, 72, 73] method, as implemented in FindMinimum in Wolfram Mathematica 9.0.1.0 [60].

6.2.3 Nonlinear refinement

The method developed in Section 6.2.1 is followed by a nonlinear refinement of the inferred kinematic model. The squared distance between the set of predicted end-effector positions yielded by the forward-kinematic model and measured end-effector positions is minimized as according to Equation 6.14, where m is the number of sampled arm poses, n is the number of joints in the kinematic chain, and the term $M_0 \dots M_n[0, 0, 0, 1]^T$ describes the forward-kinematic model determining the position of the robot's end-effector, as in Equation 6.7. Minimization is performed over the set of Denavit-Hartenberg parameters, $\hat{\theta}_i$'s, and G_i's. Optimizations in the presented results use LevMar [67], an implementation of the Levenberg-Marquardt [74, 75]

[3] The inertial frame can also be described as a joint with 3 parameters, demonstrating why the home position is sufficient to determine its parameters. It has no encoder offset or gear reduction, and D_0 can be assumed to be zero.

algorithm in C++.

$$argmin(\sum_{j=1}^{m}||M_0 \dots M_n[0,0,0,1]^T - b_j||^2) \tag{6.14}$$

The decision to first infer a model of the robot's kinematics via circle point analysis, then to refine this estimate via nonlinear optimization comes as a means of finding a fast approximation of the globally-optimal model, allowing for the nonlinear optimizer to find a better optimum by avoiding locally-optimal solutions. For an ideal arm, circle point analysis would yield a flawless model. The real robot, however, is affected by multiple factors that may contribute to error in this estimate. Tracking of the end-effector, for instance, may be affected by the view afforded to the marker from a given angle. This may cause an imperfect estimate of the end-effector's position and cause the measured circle to not perfectly reflect the arm's kinematics. Factors of slack or backlash in the robot's joints may cause them to sag in a non-uniform manner in different poses. Even small errors of this nature could cause large measurement errors in the inferred kinematic chain. Additionally, the method of circle point analysis is limited in the region that it explores, constraining each joint to a single circle of motion away from a chosen home position. Measurement errors which are well-accommodated by a model inferred from samples in this region may describe the sampled dataset well, but not generalize as well as a model inferred from random arm poses.

The process of refining this model via nonlinear optimization allows us to combat these problems. A model inferred via the minimization from Equation 6.14 is not constrained to any particular region of motion as long as the robot is able to accomplish the pose and the marker is in view of the motion tracker. This not only allows the robot to more thoroughly sample the possible space of arm poses, but also to update its kinematic model from data sampled through arbitrary motion.[4] We will see an example of this in Section 8.4.4. Nonlinear optimization also allows for the system to optimize the parameters of multiple joints simultaneously, fitting a model that better captures the system in light of how these parameters interact with each other. It has the shortcoming, however, that for non-convex problems, the optimizer may become "stuck" in local optima. Inferring an initial model via circle point analysis helps to combat this problem by seeding the optimization with a good initial estimate upon which it can improve.

In practice, the robot infers kinematic models via the following procedure. A dataset of arm poses is sampled both according to Algorithm 1 for circle point analysis and from random arm motion. Circle point analysis and the nonlinear refinement from Equation 6.14 are performed on the subset of data that has been sampled according to Algorithm 1. The set of structured arm poses sampled via Algorithm 1 and from random arm motion are then concatenated to each other, and the optimization from Equation 6.14 is then performed over this combined set.

6.3 Evaluation

To test the system, Nico was instrumented with markers for use with the Vicon MX motion tracking system as discussed in Section 3.2. As required by the Vicon Nexus software, three markers were attached to the arm forming a triangle with a static shape to be tracked. Only the marker mounted to the wrist of the robot's hand, however, was used in the evaluation of the algorithms described in this chapter, Figure 6.7. The locations to which these markers were mounted were chosen because they are surfaces to which the adhesive foam tape[5] readily and stably adheres, as-

[4] Naturally, the marker must be in view in every sampled pose.

[5] 3M Double-Coated Urethane Foam Tape.

suring that the marker would not move or fall off during data collection. The back of the robot's hand does not adhere well to the small marker used in this evaluation, because the polished surfaces of the screws in the back of the hand cause the small patch of tape to easily fall off. In the case of experiments appearing later in this dissertation, in which a cardboard-backed fiducial marker is used, the larger square of foam tape adheres more-readily to this surface, having sufficient contact with the 3D printed acrylonitrile butadiene styrene (ABS) body of the robot's hand to adhere sufficiently for evaluation purposes. It is possible for the robot to move into poses such that the tracked markers are obscured from the Vicon MX cameras. During evaluation, such obstructed views were discarded from the sampled data.

Two sets of arm poses were sampled by the robot:

- 104 samples lying along circular paths as described in Section 6.2.1.2.[6]
- 600 random samples. 100 subsampled from this set are used as a test set to evaluate the performance of the inferred model. Up to 500 are used as a training set to be concatenated to the set of 104 points sampled along circular paths for use in nonlinear refinement of the inferred model.

6.4 Results

Because of differences in robotic hardware, it can be difficult to interpret how an algorithm developed and tested on one robot will perform on another. Robots work with different sensors which may have different precision, and their arms and other

[6] The number 104 was arrived at by rotating the four joints in the robot's arm according to Algorithm 1. The endpoints of the robot's arm motion were determined by a combination of physical factors, such as hard-stops or other physical restrictions of the robot's joint motion, and by the limit of the range in which the joint's motion was visible to the motion tracking system. This number also reflects the discarding of some samples due to obstruction of the camera's view during the robot's arm motion (in configurations inside the limits of the circular arm motions performed during execution of Algorithm 1).

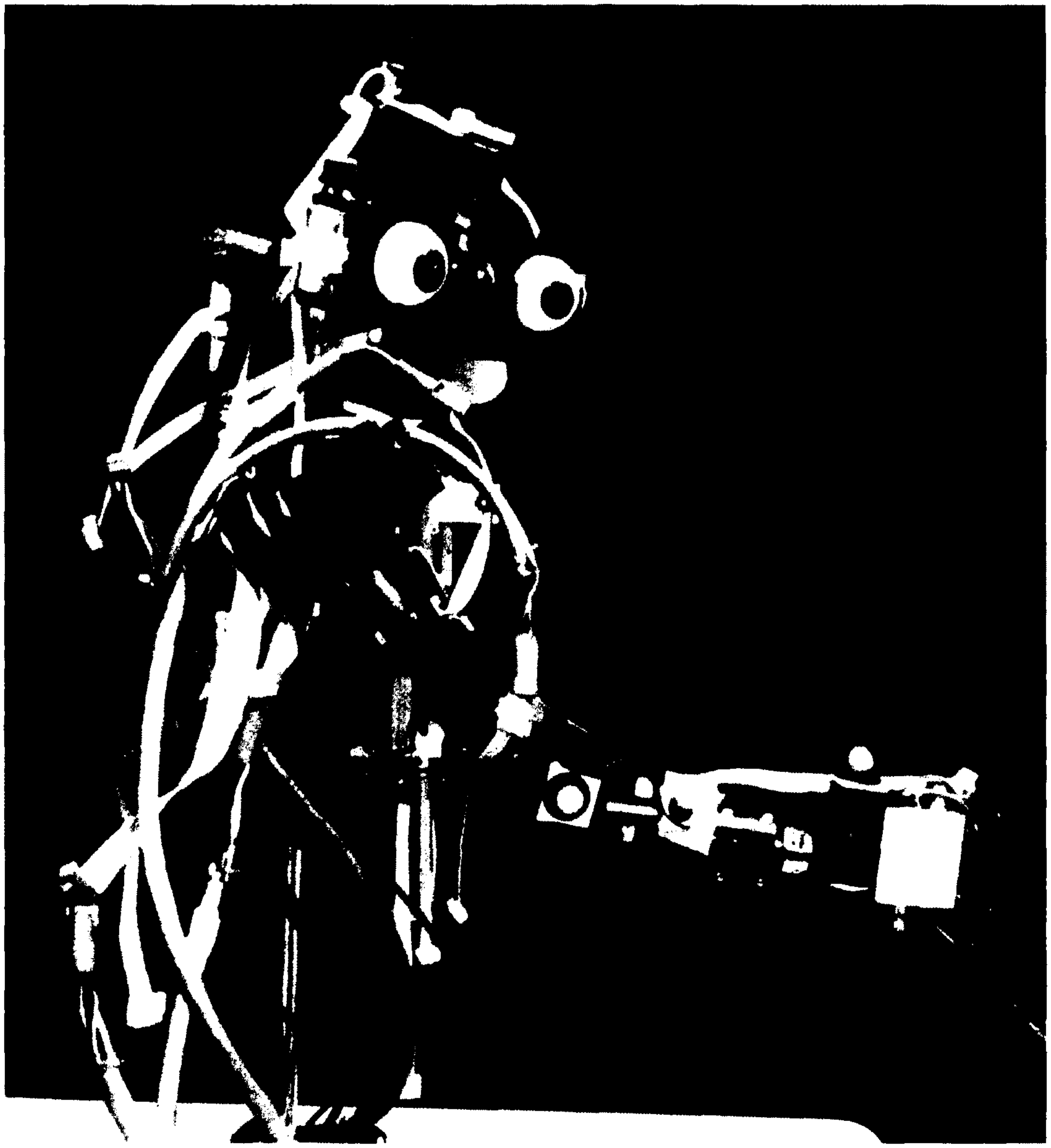


Figure 6.7: The robot, Nico, with reflective markers attached to its arm in order to instrument it for use with the Vicon MX motion capture system. The markers are balls covered with reflective tape on a black base, attached to the wrist, forearm, and elbow of the robot's right arm using squares of urethane foam with an adhesive applied to both sides, forming a deformable double-sided adhesive patch.

hardware may be built on different scales. In humans, the perception of distance is tied to the length of the arm [14]. In Chapter 8, the developed system will mutually tune the calibration of the robot's kinematic model and vision system using data sampled by witnessing the motion of the robot's arm in its visual field. The length of the robot's arm provides a scale that can help to interpret how these results may apply to larger and smaller robotic devices. Therefore, in order to help to interpret the results presented in this and later chapters, they are presented both in millimeters and as a fraction of the length of the robot's arm. In the original presentation of this material [56], measurements of the length of the robot's mechanical linkages, as instrumented for this experiment, were not taken. The length used to scale these results to the length of the robot's arm are taken from the robot's own estimate of its kinematic model, as determined using the methodology described in this chapter (242.482mm). The accuracy of these numbers is discussed later in this section.

First we should establish how well the system can perform on the minimum sample of data required to train the system. This 4 DOF system can be trained via circle point analysis with as few as 9 samples, 3 for the first joint and 2 for each additional joint. We also want to establish the degree to which nonlinear refinement improves the derived model as well as the importance of sample size. The chart in Figure 6.8 shows the performance of models inferred from datasets of 9 and 104 points, respectively, using either (1) CPA, or (2) CPA followed by nonlinear refinement of the model. Performance was tested a set of 100 random arm poses sampled from the set of 600.

We can first see that additional points improve the performance of CPA greatly. The minimum number, 9 samples, yields a model accurate to within 132.60mm (SD=65.53), equivalently 0.55 (SD=0.27) fractional arm length (which for the remainder of this document shall be denoted "arm"). The model trained on 104 points

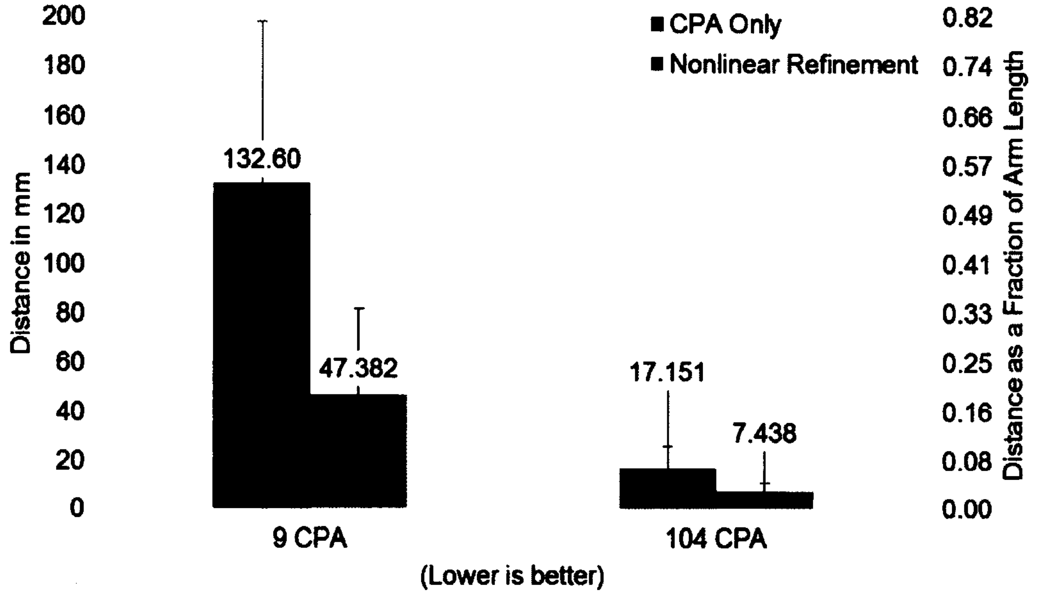


Figure 6.8: Results for a kinematic learning test using the Vicon motion tracker, expressed as distance from predicted end-effector position to measured end-effector position. Evaluation is performed over 100 random samples. "CPA" shows results after circle point analysis with no nonlinear refinement, whereas "Nonlinear" shows performance after nonlinear optimization performed to refine the inferred model. Under each of these two regimes, the system was trained twice, with a set of 9 samples lying along the circular paths required for CPA, and a set of 104 samples lying along the same paths, respectively. The training sets for the systems presented in this chart do not include the set of 100 randomly distributed training samples. The error bars indicate the standard deviation of the distance between the predicted position of the end-effector and the position as tracked by the Vicon MX motion tracking system.

is significantly more accurate, within 17.15mm (SD=9.15) (0.071arm, SD=0.038). This is an improvement of 87.1%. We can attribute this partly to the fact that the circles upon which these measurements lie define the derived model of the arm. The effect of noise on the measurements of the position of the robot's end-effector can cause a significant deviation with regards to both the plane in which the fit circle lies and the radius of the circle. Additionally, if the error occurs in one of the earlier joints in the kinematic chain, proximal to the inertial frame, then the effect of this measurement error is amplified as it interacts with the remaining joints in the chain. Sampling additional points along these circular trajectories provides a much better fit for these circular paths and goes a long way towards mitigating this source of error.

The step of refining the estimated kinematic model through nonlinear optimization can be seen to significantly improve performance in both the 9 and 104 sample cases. In the 9 training sample case, it reduces error to 47.38mm (SD=34.24) (0.20arm, SD=0.014), and in the 104 sample case to 7.44mm (SD=3.51) (0.031arm, SD=0.014). Training the system in this way allows for information sampled along each circular path to inform each joint's model. Information about the path sampled for CPA on one joint may inform the measurement of an aspect of another joint in the model, such as the orientation in which it lies or the length of its mechanical linkages. It also helps to account for mechanical imperfections in the robot's arm, finding the best-fit approximation of the position of the end-effector accounting for factors which are not specifically modeled, such as backlash in the robot's gear-train or the arm sagging slightly due to the force of gravity.

The addition of samples of random arm poses also significantly improves the overall performance of the system. The remaining 500 random arm poses were sampled into smaller subsets of 50, 100, and 500. These poses were concatenated to both the

9 and 104 sample training sets and the process of nonlinear refinement of the estimated model was repeated on these larger training sets. The purpose of this test is to evaluate the importance of additional random training samples to the algorithm and to establish an estimate of how many samples are required to fully train the system. It also provides an empirical measurement of how much improvement in the estimated model could be expected due to the addition of random arm poses to the training set.

The results of this test can be seen in Figure 6.9. The chart shows the sensitivity of the system to the initial estimate provided by circle point analysis. Having a better initial estimate of the system significantly improves the performance of the system, which appears have settled into a local optimum between 100 and 500 additional random training samples in the 9 CPA sample case, with performance not improving significantly between these two cases (100: M=15.32mm, SD=9.54mm M=0.063arm SD=0.039arm; 500: M=14.86mm SD=8.87mm, M=0.061arm SD=0.037arm). The 104 CPA sample training condition significantly outperforms these systems, even in the absence of additional random training data (M=7.44mm, SD=3.51 M=0.031arm, SD=0.014). Performance appears to asymptote after about 100 samples (M=3.19mm SD=1.71mm, M=0.013arm SD=0.0071arm), improving only slightly after 500 random samples (M=2.97mm SD=1.78, M=0.012arm, SD=0.0074arm). The system is also able to train these models quite quickly. It took 1.55 seconds to perform circle point analysis on the 104 point dataset. Nonlinear refinement took 0.92 seconds on an Intel Core i7-3630QM CPU clocked at 2.40 GHz when no additional random points were added, 1.62 seconds for 50 random points, 2.41 seconds for 100 random points, and 14.21 seconds when 500 randomly distributed arm poses were added to the training set.

In order to test the consistency and reliability of this method, the nonlinear

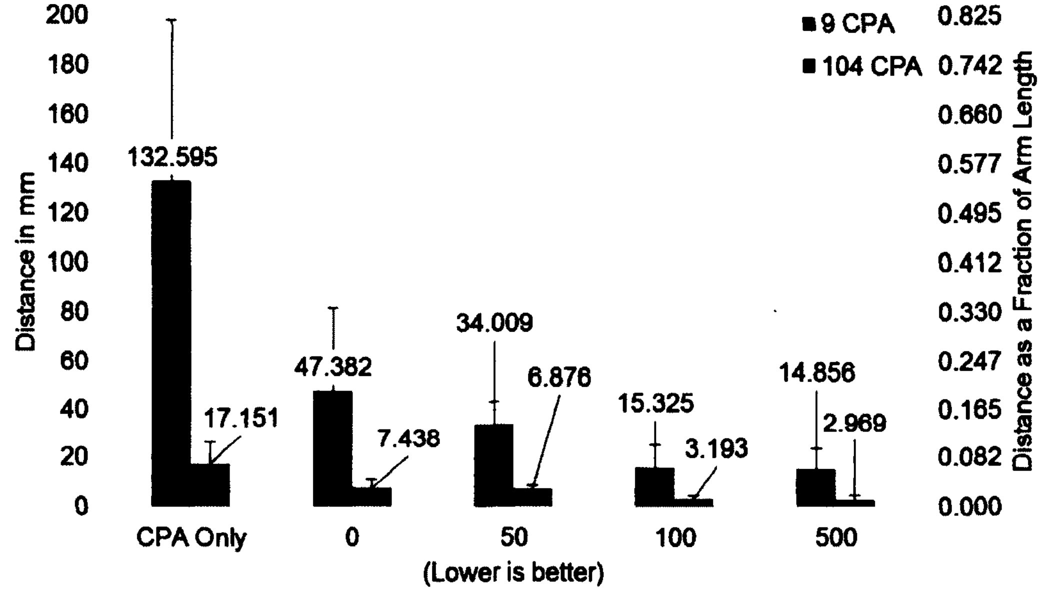


Figure 6.9: Results for a kinematic learning test using the Vicon motion tracker, expressed as distance from predicted end-effector position to measured end-effector position. Evaluation is performed over 100 random samples. "CPA" shows results after circle point analysis with no nonlinear refinement, whereas "Nonlinear" shows performance after nonlinear optimization performed to refine the inferred model. Under each of these two regimes, the system was trained twice, with a set of 9 samples lying along the circular paths required for CPA, and a set of 104 samples lying along the same paths, respectively. To these sets of 9 and 104 samples additional sets of $0, 50, 100$, and 500 randomly sampled arm poses were added during the nonlinear refinement phase, respectively.

refinement step was performed 100 times, each time initialized using an inferred kinematic chain from the 104 point dataset. Each of these 100 models used a random subset of the 500 random arm poses sampled for testing. The mean error over the training set for the fully-trained model is 2.47mm or 0.01arm. The standard deviation in this error between these models is 0.091mm or 0.00038arm. Over the test set, the mean error is 3.21mm (SD=0.039) or 0.013arm (SD=0.00016). In this test, the lengths of the robot's mechanical linkages were also estimated, yielding the estimates of linkage length used in this chapter to scale results with respect to the length of the robot's arm. The first link, running from the robot's shoulder to its elbow is 126.82mm (SD=0.45) long. The second, running from the elbow to the center of the marker placed on the robot's wrist is 115.66mm (SD=0.84). The mean time this nonlinear refinement took was 2.52 seconds (SD=0.14). These results demonstrate the repeatability of this process in allowing for the robot to infer the parameters describing its kinematic chain.

6.5 Summary and conclusions

This chapter has discussed a process by which a robot can infer its revolute joint kinematics. The system is trained on a set of the robot's observed arm poses through a combination of encoder data and measured 3D end-effector position in two sets of poses. The first set of poses is a structured dataset, consisting of each joint in the observed kinematic chain moving in isolation. Because the joints in the chain are revolute, when moved in isolation the end-effector traces a path lying on an arc of circle as it moves through space. This provides inspiration for an algorithm which measures the set of joints and infers the parameters describing the kinematic chain by measuring these circles and their relationships to each other.

The second dataset is a set of random arm poses. This dataset is used to refine the initial estimate of the kinematic chain by performing nonlinear optimization over the entire model. This concept of finding a fast initial estimate to avoid local optima then refining this estimate through optimization techniques mirrors a pattern established by the computer vision community in the domain of camera calibration techniques [55, 62].

To evaluate the system, Nico was instrumented with reflective markers for use with a Vicon MX motion capture system, allowing the pose of the robot's arm to be measured as a combination of encoder ticks (measured at each motor) and 3D positions of the end-effector in space (measured through motion capture). The system was demonstrated to learn a model of the robot's arm kinematics that agrees with measurements made by the motion capture system to within 2.97mm (SD=1.78)mm, M=0.012arm (SD=0.0074)arm after 500 samples, with performance appearing to asymptote at around 100 additional random samples in both the 9 (M=15.323mm SD=9.54mm) and 104 (M=3.19mm SD=1.71mm) CPA sample cases. These results appear to approach the precision to which the system is able to perform, because the markers that were used with the Vicon MX motion tracker have a diameter of 14mm. Though difficult to directly compare to other work in the area, due to differences in hardware [36, 35], these results are competitive with related state of the art techniques when using even a minimal set of data. When fully-trained, these techniques improve in performance over those techniques by an order of magnitude.

Chapter 7

Integrating kinematics and computer vision

Chapter 6 presents an algorithm that allows the robot to infer the description of its arm's kinematics from samples of its motion. In that chapter, a commercial motion tracking system was used to track the robot's end-effector. This chapter seeks to enable the robot to infer its kinematics without such external instrumentation, through the use of its stereo vision system. Moreover, the goal of this chapter is to integrate the robot's models of kinematics and vision. We represent these under a single, unified representation. This representation is a kinesthetic-visual self-model, inferred by the robot through its own sensing while using its own body.

As a starting point for inferring the robot's kinematics through vision, this chapter will explore the possibility of using the robot's stereo vision system as the source of data on the robot's arm. As discussed in Chapter 1, it is typical for a robot's visual calibration and kinematic calibration to come from two different data sources. In the case of kinematic calibration, this source is often an expert engineer who provides the robot with a model based on the engineering drawings created during its design process. In the case of visual calibration, the source is often a set of calibration techniques in which the robot's vision system images a target of known structure and infers the projection which produced the corresponding images. Such techniques will be further discussed in Section 8.1.

Learning a robot's kinematics through its visual system has previously been explored by other researchers. Several of the kinematic learning algorithms discussed in Chapter 6 were developed in order to integrate kinematics and vision. For instance the systems developed by [36] and [35] both infer the kinematic chain producing motion at the end-effector of a robot's arm through the use of the robot's stereo vision system. These projects come from work in robotic body schemas in which researchers attempt to emulate the human body schema on robotic platforms. The merging of the kinematic and visual sense constitutes a form of a visual-kinesthetic body schema. Hoffmann, Marques, Arieta, Sumioka, Lungarella, and Pfiefer [33] provide a good overview of current work in this area. The visual sense is the most well-developed rich sensing modality in modern robotics and forms the dominant sense in many modern robotic systems, making it a natural choice of sensory modality to emphasize in modern robotic body schema research. Similarly, the construction of kinematic models upon which such systems are built is well-understood. Similar to the work presented in this chapter, the goal of learning a robot's kinematics through its visual sense is to meaningfully integrate the two systems. Both [36] and [35] report their results in terms of agreements between the predictions of the forward-kinematic models of their robots and the 3D measurements made by their stereo vision systems.

In the sense that this work concentrates on the intermodal problem of combining kinematics and vision, it is similar to the work of Yoshikawa, Tsuji, Hosadam and Asada [76], Gold and Scassellati [30], and Stoytchev [57]. These systems attempt to merge kinematics and vision in such a way that actions performed on the robot's kinematic chain are predictive of 2D changes in images sampled by its stereo vision system. This differs from [36] and [35], in that those systems attempt to merge the 3D capabilities of their vision systems with the 3D tracked positions of the robot's end-effector. Yoshikawa et al. [76] and Gold and Scassellati [30] both focus on the problem of segmenting portions of the robots body in images from the robot's vision system. In the case of [76], inferences of which pixels do and do not constitute the robot's body parts are made through correlations between motor state, in the form of joint angles, and what is imaged when the robot is in a particular motor state. Things that a robot consistently sees when its body is in a particular pose are considered to be part of the self. In the case of [30], a Bayesian classifier is constructed which allows the robot to segment itself from its environment in images based on correlation of the enactment of a motor action and changes in the visual field. The temporal correlation between a motor action and motion reflected in the visual field is what is used to perform the segmentation task. Stoytchev [57] presents a model based on a related idea. In experiments, monkeys have been able to perform manipulation tasks by witnessing their arms in video monitors rather than directly in their visual field [25]. Stoytchev's experiment focuses on making the inferences required to transform the robot's body schema such that it is able to perform such spatial reasoning through images witnessed through a video monitor.

The work presented in this chapter differs from these models in that it utilizes a parameterized model of the robot's kinematics, concentrating on mutually calibrating this model and that of the robot's vision system. The result is that the robot's kinesthetic-visual self-model is predictive of the position of its end-effector in its visual field in both 2D and in 3D. Additionally, while much of the other work on such intermodal perception problems focuses on the biological plausibility of particular approaches, the presented method is intended to build on classical engineering techniques such that it may be easily integrated into existing systems.

7.1 Tracking the robot's hand using computer vision

Under the typical process of separately developing the robot's kinematic model by hand, but calibrating the robot's stereo vision calibration through the use of an external target, the position of the end-effector under the two disparate models is likely to disagree. The bases of these two systems may disagree in scale, orientation, or position, and even slight deviations between these two systems can lead to significant differences in where they estimate the robot's end-effector to be.

The goal of the system developed in this chapter is to allow the robot to infer an estimate through forward-kinematics that tightly agrees with the 3D reconstructed position of its end-effector in its visual field. To do so, it will infer an estimate of the robot's arm's kinematics through arm poses in which the end-effector is tracked by the robot's vision system. Because the reconstructed arm poses are sampled through the stereo vision system, the reconstructed position of the end-effector is represented in the same basis as the vision system. As such, one can expect a high degree of agreement between the estimates provided by these two systems. Moreover, one can regard the system as unified, as the two models will be mathematically compatible with each other.

To allow the robot to use its stereo vision system to reconstruct an estimate of its arm's kinematic chain, software was developed that allows the robot to track the position of the end-effector in the left and right images sampled by its eye cameras and reconstruct the end-effector's position in 3D. This allows the vision system to replace the functionality of the Vicon MX motion tracker used in Chapter 6. The model yielded by the system developed in this chapter is capable of performing the following tasks:

- Tracking a marker attached to the robot's end-effector in 2D in the left and right images sampled by the robot's stereo vision system.
- Reconstructing the 3D position of this marker from the tracked centroids yielded by the left and right images.
- Estimating a model of the kinematics of the robot's arm, using the techniques developed in Chapter 6.
- Using this model as a forward-kinematic model, estimate the end-effector's position in 3D. This 3D estimate of the end-effector's position will be close to where the stereo vision system reconstructs its position to be.
- Combining the forward-kinematic estimate of the robot's end-effector with the projective capabilities of the stereo vision system to estimate where the robot's end-effector will appear in left and right stereo images in 2D.

For the task of tracking the end-effector in the robot's visual field, two methods have been implemented:

Color Blob Detection: A method which tracks an object of a specific color.

Fiducial Tracking: A method which tracks a marker with a known image printed onto it.

The system reports the 2D tracked position of the marker as lying at its center in each image and reconstructs its position in 3D from the tracked positions in the left and right cameras. Custom software was developed in order to support the stereo vision functionality necessary for this process.

7.1.1 Object tracking

The robot, Nico, has two eye cameras mounted into its head that are capable of capturing 640 × 480 color video at 30 frames per second.[1]. This vision system has been developed to track the 2D centroid of a marked object as it moves through space as imaged by these cameras. This centroid is tracked in both views, allowing its 3D position to be reconstructed from the pair of tracked centroids. Two methods for performing the task of object tracking have been implemented. These are color blob detection and fiducial tracking. It should be noted here that Nico's stereo vision system performs object tracking on the original images as sampled by the robot's cameras. The images have not been undistorted prior to this image processing. The computed centroids are individually undistorted and converted to distortion-free ideal image coordinates prior to stereo reconstruction.

7.1.1.1 Color blob detection

The primary advantage of color blob detection is that it is simple to instrument an object to be tracked. The system can quickly be tuned to an object that has been painted or wrapped in colored tape, allowing a number of objects to be easily tracked. Additionally, the colored object can be tracked from all sides that the color is visible from. For bright colors that are easily illuminated and differentiated from other colors, only a small patch is necessary in order for the object to be tracked as it moves through the robot's immediate workspace.

[1] For details on the robot's vision hardware, see Section 3.1.1

Color blob detection has the drawback that specular highlights[2] will generally not be recognized for their underlying color by the segmentation algorithms employed. Additionally, the real detected position corresponding to the centroid of a segmented color blob moves along the surface, depending on what section of the object is imaged by the vision system. That is to say, any point on the colored 3D surface is a valid candidate to be chosen as the tracked point in any given image of the tracked object, limiting the precision of the tracking system to the volume of the tracked colored surface. In the case of the experiment in Section 8.4.4, in which the robot tracks the endpoint of a marked screwdriver, the colored patch of tape only measures about 1cm × 1cm, but is tracked successfully. Using the smallest possible patch of colored tape limits the trackable volume, thus improving precision in this context.

This implementation of color blob detection was constructed primarily using OpenCV [63]. Blob detection was performed through a series of color channel subtractions and thresholds. It was decided to label the tracked end-effector using red electrical tape, making red the color to be tracked. RGB images sampled by each camera are input to the blob detection algorithm. A copy of the image is made, converting the format into HSV (hue, saturation, value). The first image is then split into three intensity images from the red, green, and blue channels, respectively. The second is split into hue and saturation images, with value being discarded. The algorithm first finds pixels that are more red than green and more red than blue by producing two images by subtracting the corresponding color channel from the red channel image. The hue and saturation images are thresholded based on empirically determined values.[3]

The logical AND of these four images is computed to include only those pixels

[2] The white or bright areas on a surface where one can see the reflection of a light source.

[3] An interface was developed that allows an operator to select an object to be tracked by clicking on the object. This determines values for hue and saturation, which then can be manually tuned.

that occur in all four images. The resulting image is then blurred in order to help remove noise. This is followed by a relaxation labeling process, in which a pixel is determined to be "on" if three or more of its six neighbors are "on" and off if not. The intention of this process is to clean up the labeling in the processed image, filling in regions that may not be properly labeled due to imaging effects such as specular highlights, or joining two image blobs that may be disjoint due to noise.

The largest contiguous region of pixels determined to correspond to the chosen color and its centroid are then determined through the CVBlob library [77]. CVBlob determines the centroid of the tracked object to lie at the center of a square bounding box surrounding the tracked object, rather than as the mean of the tracked pixels.

7.1.1.2 Fiducial tracking

The advantage of fiducial tracking over color blob detection is that it is a higher-precision technique. Fiducial tracking works by tracking a marker, in this case a plane with a known image printed onto it. The Augmented Reality Toolkit (ARToolKit) library [2] has become a popular implementation of fiducial tracking. It tracks images of black and white boxes with identifying icons inside, as in Figure 7.1, allowing for multiple markers to be differentiated from each other. Instrumenting the robot to track such a marker is simply a matter of printing the marker, placing it onto a rigid backing, and adhering the marker to the robot, as in Figure 7.2.

The implementation of fiducial tracking used in this system extends the fiducial tracking capabilities of Augmented Reality Toolkit. The purpose of this extension is to enhance the quality of the library as a tracker of the 2D projection of the center of the 3D fiducial marker. ARToolKit returns, as the centroid of the marker, a position that is determined by the algorithm used to fit the position of the marker on the screen. This position, however, does not always correspond to the image of the

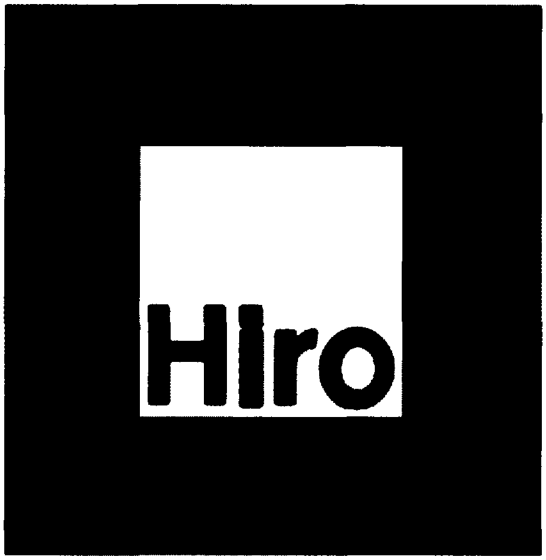


Figure 7.1: Fiducial image included with ARToolKit [2] used to instrument Nico's hand.

center of the physical marker. The detection of the corners of the black box pattern on the border of the marker, however, is always precisely where the corners of the box appear. As such, software was developed to compute a homography representing the transformation from the model marker to the imaged marker using these corners as input.

For this process, homographies are computed as described in Section 5.1, with the exception that both the image points (b_i') and the model points (b_i) are normalized prior to computation of the homography. The normalization is computed such that the center of each collection of points is at $(0, 0)$, and the mean distance from the origin to each point is $\sqrt{2}$, as in the case of the Normalized Eight Point Algorithm [78]. This normalization can be computed as the transformation in Equation 7.2. In this formula, s is the scaling factor to scale the points such that the mean distance (d in the case of the original points) to the center of the collection of points is $\sqrt{2}$, and w and h are the width and height of the bounding box containing the collection, respectively. The points are normalized by transforming by their corresponding

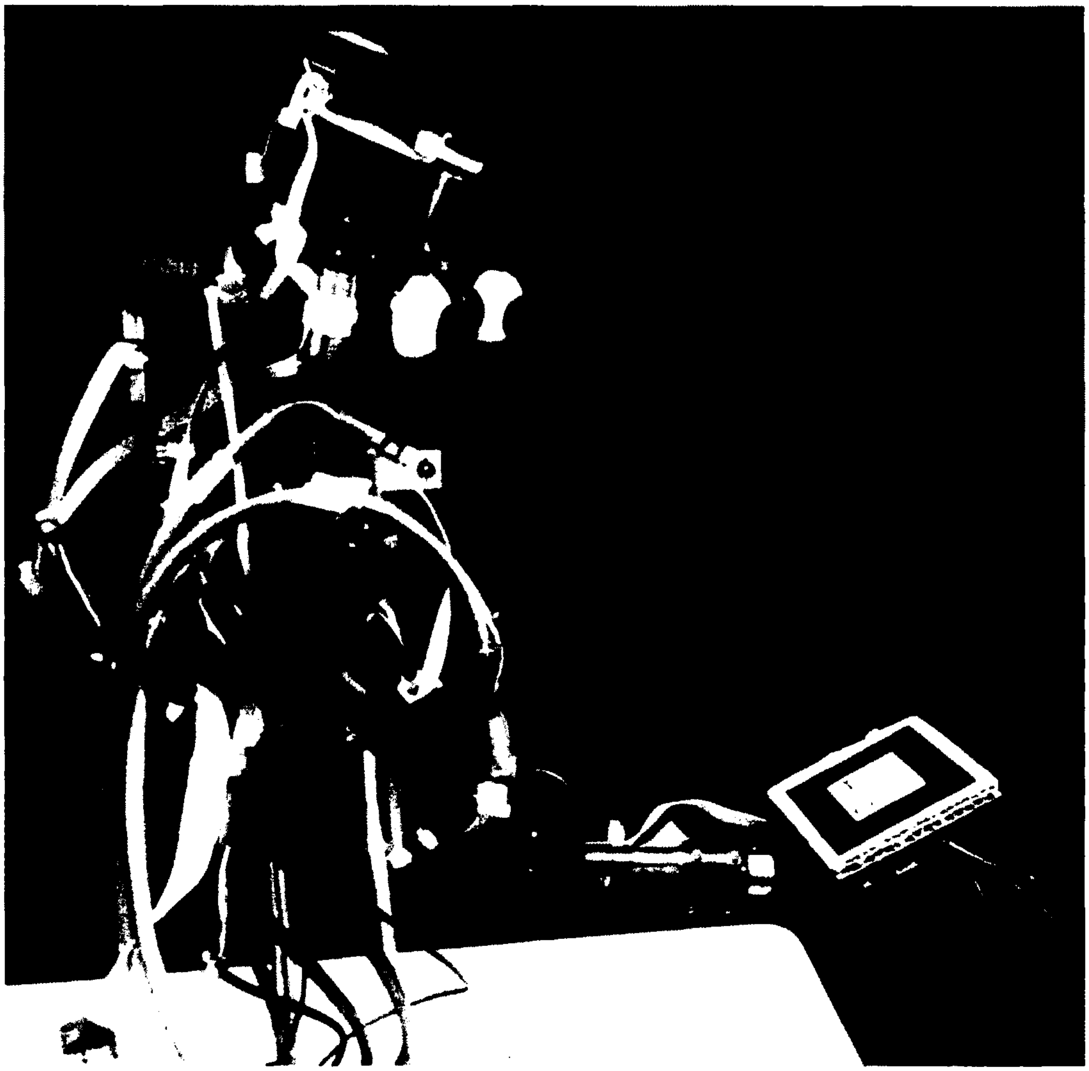


Figure 7.2: In this image, Nico has been instrumented with a fiducial marker adhered to the back of its hand. The marker, a black box with the name "Hiro" at the bottom, is tracked using ARToolKit [2], with 2D position being more-precisely tracked using custom software that operates on top of this toolkit.

normalization matrix, N.

$$s = \frac{\sqrt{2}}{d} \tag{7.1}$$

$$N = \begin{bmatrix} s & 0 & -sw \\ 0 & s & -sh \\ 0 & 0 & 1 and \end{bmatrix} \tag{7.2}$$

After computing the homography from the collection of normalized points, the homography is denormalized as in Equation 7.3, where H is the computed homography and $\tilde{H}$ is the homography subject to normalization. This normalized method for computing homographies is discussed in [55].

$$H = N'^{-1}\tilde{H}N \tag{7.3}$$

From this homography, the image of the center of the fiducial can be computed by projecting the point $b = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$. This assumes that the center of the model is the origin. This can simply be done by computing the homography with respect to the model points $\begin{bmatrix} -1 & -1 & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & -1 & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$, and $\begin{bmatrix} -1 & 1 & 1 \end{bmatrix}$, corresponding clockwise to the images of the corners of the border of the fiducial marker. Because this is a form of interpolation between the images of the corners of the fiducial, the center of the fiducial marker can be localized to sub-pixel accuracy.

7.1.2 Stereo reconstruction

Classically, the problem of recovering 3D positions and geometry from scene images, in the case of stereo vision,[4] can be broken down into two subproblems. The first of

[4] Other computer vision methods in which this is a relevant problem include monocular approaches such as fiducial tracking in one camera or shape from shading and approaches such as structure from motion.

these subproblems is the stereo correspondence problem, determining which points in the first image correspond most closely to points in the second image. The second is the problem of reconstructing 3D scene geometry from these matched points.

In the case of this system, the centroids of the left and right images of the tracked marker are the only points we are interested in using for stereo reconstruction. As such, the need to solve the stereo correspondence problem is obviated. These exact points are used.

Stereo reconstruction is performed via the standard technique of computing the point such that the distance in the left camera and the right camera between projections of that point and the sampled images[5] of that point is minimized as a least-squares fit via Singular Value Decomposition (SVD). Projections of 3D scene points are then redistorted according to the camera's distortion model, as discussed in Section 5.2.4. In this system, all vision functions are computed such that lens distortion is directly accounted for through the stereo vision system. As previously discussed, each image point corresponds to a single ray of light running through the camera center, to the 2D image point on the image plane, to the corresponding 3D scene point. For a pair of stereo cameras, the rays of light corresponding to a 3D scene point for two matched image points will intersect at the 3D scene point, as in Figure 5.3. The task of stereo reconstruction can also be thought of as finding an approximation of the point at which these rays of light intersect. In this system, it is solved as a system of equations such that the image points p_L and p_R correspond to projections of P. Factoring out the elements of P such that they can be found as the right null space of the matrix representation of this linear system allows P to be found via SVD. For a full discussion of this technique, see [55].

As stated before, one approach to dealing with image lens distortion is to undis-

[5] As undistorted ideal image coordinates.

tort input images prior to performing stereo reconstruction, which can lead to error in the stereo reconstructions of matched points. In this system, individual points are undistorted from the 2D tracked marker images. Stereo reconstruction is then performed on the distortion-free ideal image coordinates corresponding to these images. In the case of projecting 3D points to their 2D representations, the 2D coordinates have the lens distortion applied prior to application of the camera intrinsic matrix. As such, this stereo vision system incorporates the explicit computation of lens distortion into every projection and reconstruction operation, allowing for a corresponding improvement in the accuracy of the processes performed by the stereo vision system.

7.2 Integrating vision and kinematics

As can be seen throughout this thesis, homogeneous representations of geometry have been used to represent both the robot's kinematic chain and the processes performed by its vision system. Incompatibility between classical representations of these aspects of a robot's design is not the result of a representational shortcoming, but due to the fact that these systems have not been mutually calibrated. In other words, unifying these two models is a matter of assuring that the origins, orientations, and scales of the bases in which they are represented match.

Integrating models of kinematics and vision has been previously explored in various contexts. The hand-eye calibration problem can be stated as determining the transformation between a camera mounted on a kinematic chain and the existing description of that chain [79]. Hand-eye calibration has been studied extensively (Zhao and Liu provide a good overview [80]), and is useful for visual servoing tasks, such as guiding robotic arms performing welding tasks [81].

Relatedly, Pradeep, Konolige, and Berger [40] performed a bundle-adjustment approach to the global optimization of both sensor and kinematic calibrations. The most obvious difference between their approach and that presented in this work is that their system refines its calibration based on images of a chessboard calibration target held in the robot's gripper, whereas this system tracks the motion of the robot's hand, using the arm's kinematics as the spatial invariant against which calibration is performed.

More recent work from the body schemas literature has attempted to create a mutual calibration between the robot's kinematics and vision system by inferring a calibration of the robot's kinematics with images sampled through the stereo vision system [36, 35]. In these recent papers, however, the stereo vision system is treated simply as a device for inferring the 3D position of the robot's end-effector. As such, this integration has been limited to compatibility between the 3D coordinates yielded by the robot's forward-kinematic model predicting the position of its end-effector in its visual field and the 3D reconstructions of positions of the tracked end-effector.

This work seeks to achieve a more complete, unified model of kinematics and vision by exploring their combined representation. Given the accurate calibration of such a unified model, the position of the end-effector in the visual field can be determined by combining the forward-kinematic model found in Equation 6.7, with the projective model for each camera. This is done by substituting the projected point in 3D, P, with the forward-kinematic model, as in Equation 7.4. The result of this process is a 2D prediction of the position of the end-effector in the robot's visual field, demonstrating the extent to which the kinematic and visual aspects of this model have been combined.

$$p_{end-effector} = K[R| - RC]M_0 \ldots M_n[0, 0, 0, 1]^T \tag{7.4}$$

This differs from previous work in body schemas, in that the robot is able to simultaneously predict the position of its end-effector in both 2D and 3D in a manner that is fully-integrated with the robot's stereo vision capabilities. The implementation extends standard representations and techniques, allowing it to be implemented into a wide variety of systems and a variety of standard hardware platforms.

7.3 Evaluation

Performance of this system was evaluated in two tests using color blob detection, Figure 7.3, and fiducial tracking, Figure 7.2. For each of these tests, a dataset of arm poses and tracked centroids was sampled. Both samples included a set of arm poses for circle point analysis, and 200 random arm poses, subsampled into 100 poses for training and 100 poses for testing. When sampling arm poses for circle point analysis, the system is limited both by the mechanical limits to which each joint in the robot's arm can be turned and the field-of-view (FOV) of the robot's cameras. As such, 60 arm poses were used for circle point analysis in both datasets.

Performance evaluation is similar to that in Chapter 6, wherein we are concerned with how closely predictions made by the robot's forward-kinematic model match measurements made by the mechanism tracking the robot's end-effector. In this case, however, the tracker is the robot's stereo vision system using either color blob detection or fiducial tracking.

The robot's stereo vision system, while capable of reconstructing the position of points in 3D, does not naturally do so under well-known units. The position of the robot's end-effector is not a priori reported in terms of inches or millimeters. In order to make the numbers reported by the robot's vision system more easily understood, the system was calibrated to millimeters using a chessboard calibration

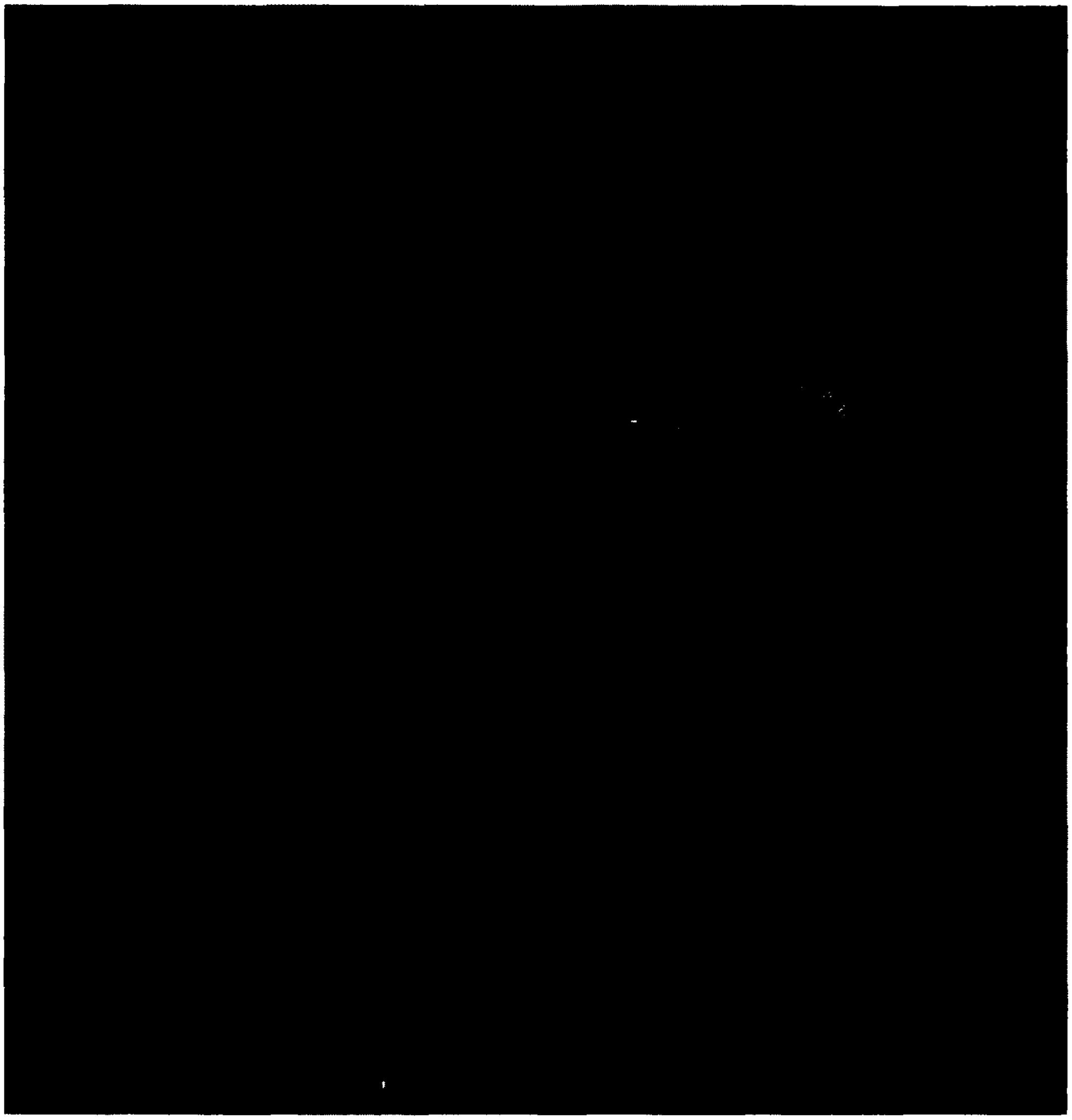


Figure 7.3: The humanoid robot, Nico, instrumented to track the motion of the tip of its finger using color blob detection. The region to be tracked is wrapped in red electrical tape. Its bright primary color is easily distinguished from other patches of color in the robot's visual field.

target, Figure 7.4. During the initial stereo calibration process (discussed in Appendix B), this target is imaged several times in the robot's visual field. The squares on the chessboard have sides that are 28mm long. In order to calibrate the system to millimeters, reconstructions of 3D positions of the corners of the chessboard squares are computed from this dataset. The mean length of the reconstructed square edges is computed from these reconstructions. This 28mm is then divided by this mean, providing a scalar conversion factor from the robot's visual basis to millimeters. In discussions of the robot's performance in which metric units are used, a conversion factor computed in this manner is employed. In this chapter, this is the 3D distance between the robot's predicted end-effector position and the position measured by the robot's visual system. Calibrating the robot's vision system to a known system of measurement, as is done here, is not required by the methods described in this dissertation. The purpose in calibrating the system to millimeters is to enable the reader to interpret the system's measurements using a familiar unit of measure.

Because the robot is able to simultaneously predict the position of its end-effector in 3D and 2D, we also report results in terms of the distance between the predicted 2D position of the robot's end-effector in its visual field and the 2D centroid reported by the tracker. This metric is interesting because it demonstrates how tightly-unified the robot's kinematic and visual models are.

7.4 Results

Results are reported as distance between the tracked end-effector position and that predicted by the robot's forward-kinematic model. In Figures 7.5 and 7.6 we can see that predictions made by the robot's integrated self-model for end-effector position in 3D closely match the measurements made by the vision system when using both color

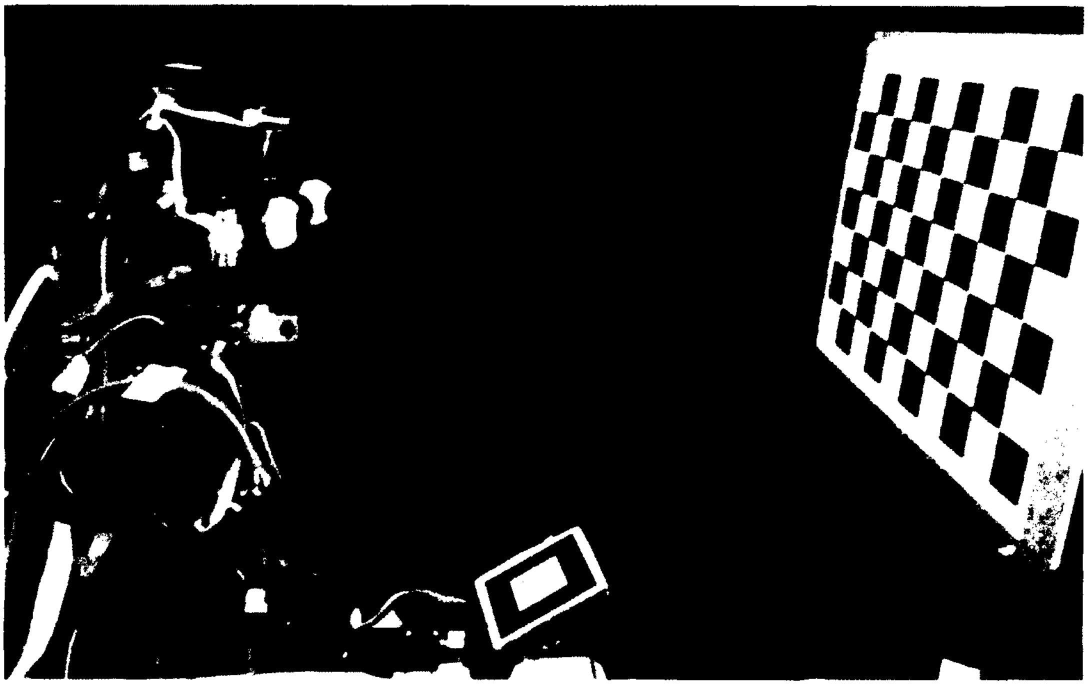


Figure 7.4: Seeding the robot's stereo camera calibration begins with imaging a chessboard calibration target through the vision system.

blob detection (M=3.77mm SD=2.30mm, M=0.012arm SD=0.0073arm) and fiducial tracking (M=2.74mm SD=1.39mm, M=0.010arm SD=0.0051arm). The performance of the system when instrumented for fiducial tracking and when instrumented for color blob detection is more similar when expressed in terms of arm length, despite the greater error when using color blob detection. This is because of differences in the position of the tracked centroid (at the tip of the finger in the case of color blob detection, on the back of the hand in the case of fiducial tracking). For color blob detection trails, the length of the arm is 317mm. For fiducial tracking, the length of the arm is 272.[6]

Looking to Figure 7.7, we also see that predictions of the end-effector position, when predicted in 2D as projected into the robot's visual field (mean measured in

[6] The length of the arm in both of these cases was determined by measuring the arm by hand with calipers. The lengths differ because of differing locations of the respective tracked markers.

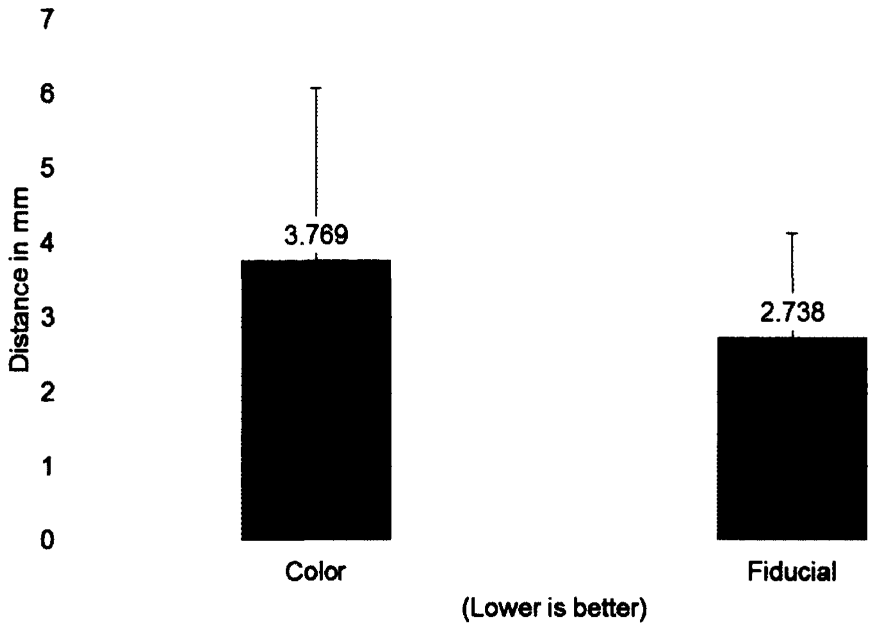


Figure 7.5: Three-dimensional distance between end-effector position predicted by the robot's self-model and that measured by the robot's stereo vision system, reported in millimeters.

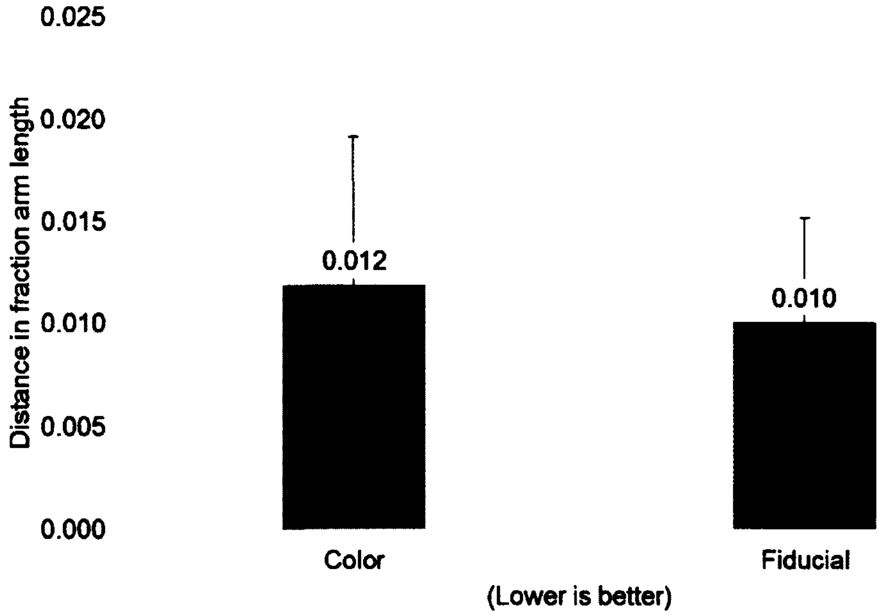


Figure 7.6: Three-dimensional distance between end-effector position predicted by the robot's self-model and that measured by the robot's stereo vision system, reported as a fraction of the length of the robot's arm.

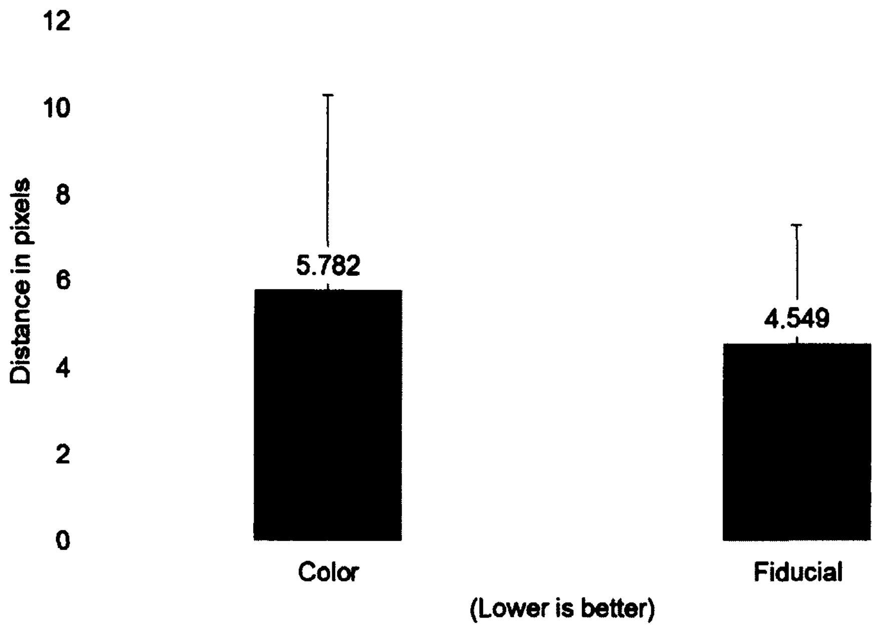


Figure 7.7: Two-dimensional distance between end-effector position predicted by the robot's self-model and that measured by the robot's stereo vision system, reported in pixels.

both left and right stereo cameras) closely match measurements made by the stereo vision system. In the case of color blob detection, agreement is to within M=5.78 pixels (SD=4.50 pixels), with fiducial tracking M=4.55 pixels (SD=2.74 pixels).

In the case of both fiducial tracking and color blob detection, we see high standard deviations in the prediction error of the model. This is partly due to the effect of visual perspective. As the robot's end-effector comes closer its eye cameras, a millimeter becomes significantly larger. Being within a volume of several millimeters can be less than a pixel when the robot's arm is fully extended, whereas when the hand is close to the face the same volume contains a larger region of visual real-estate.

Similarly, the error in 3D is slightly higher in the case of color blob detection, though both are close (color blob detection M=3.77mm SD=2.30mm, M=0.012arm SD=0.0073arm; fiducial tracking M=2.74mm SD=1.39mm, M=0.010arm

SD=0.0051arm). In the case of fiducial tracking, because the marker can be tracked to sub-pixel accuracy, it can be assumed that the error is primarily due to shortcomings in the calibration of the self-model and the limits of the mechanical and sensory precision of the robot. In the case of color blob detection, the system has one more important source of error in that the tracked centroid could lie within a wide region along the surface of the marked region of the finger, as in Figure 7.3.[7]

7.5 Summary and conclusions

In this chapter, the robot's stereo vision system was integrated into its learning process. The robot has two methods which allow it to track the 2D position of its end-effector as imaged in the left and right cameras of its stereo vision system. These methods are color blob detection and fiducial tracking. The tracked centroids of the marked portion of the robot's hand are used to reconstruct its position in 3D using standard stereo vision techniques. Special attention is paid to the high precision tracking of the end-effector in 2D and reconstruction of this single point in 3D at the cost of potentially slower reconstruction.

The constructed tracker is then used with the kinematic inference techniques developed in Chapter 6 to infer a model of the robot's arm kinematics using data sampled by its stereo vision system. Because the data from which this model is derived are sampled by the stereo vision system, the basis in which the kinematic model is represented is the same as that of the stereo vision system. This unifies the two models, allowing us to treat them as a single unified self-model, or, relatedly, a kinesthetic-visual body schema. This unification allows the robot to make forward-kinematic predictions not only in 3D, of where the hand will be in space after a

[7] The red colored region consists of the distal phalange of the robot's index finger, measuring roughly 14mm×8mm×9mm, wrapped in red colored tape.

given motion, but in 2D, of where it will be projected into its visual field. Tests demonstrate the tight coupling between the robot's kinematic and visual capabilities and the accuracy of the inferred model.

Chapter 8

Simultaneous refinement of kinematics and vision

Going back to the motivations for this thesis, one of the interesting things about how children learn about their bodies and senses is that they learn about them by using them in conjunction with each other. They learn about their bodies through what they perceive of them through their senses, and about their senses by using them in conjunction with their bodies to interact with the world. The product of learning about these two components of the self together is that they are calibrated to each other. In a robotic system, this is the same as saying that the origin, orientation, and scale of the bases in which the vision and kinematic representations are expressed are the same, or that the overall system is at least able to compute the transformation allowing the two distinct systems to convert between their respective representations. This allows data to be shared between them. In humans, evidence indicates that our perception of space is tied to our knowledge of the body. For instance, Volcic et al. [14] demonstrated that altering study participants' perception of the length of their arms altered their perception of distance. The theory of the body schema relies on the idea of such a mutual calibration of bodily and sensory representations, allowing data regarding the posture of the body to be combined with sensory information [4]. If we accept this premise, then what Volcic et al. [14] have demonstrated is that altering one parameter of this calibration, bodily proportions, has effects on another component of the system, the perception of space.

In Chapter 7, a system was developed which allows the robot to learn about its body through its senses. It was demonstrated that this calibrated the inferred kinematic model to the model of the visual system, allowing data to be meaningfully combined across these two modalities and producing capabilities related to the body schema in the form of the prediction of end-effector position in both 2D and 3D in the robot's visual field.

In this chapter we will use the model developed in Chapter 7 to allow the robot to use its own body as a visual calibration target. In doing so, the robot will be able to simultaneously enhance its visual and kinematic calibrations. As its knowledge of the body is refined, the body becomes a better calibration target for the vision system. Using this better target improves visual calibration and in turn provides the robot with more accurate measurements of the body, allowing it to refine its knowledge of its body. In the algorithm presented in this chapter, this process is performed as a single global optimization over the combined parameterization of the robot's kinematic and visual systems. In this sense, the present approach is similar to bundle adjustment [82] with the exception that the calibration target is not the 3D geometry of a reconstructed object but the kinematic structure of the robot itself. Note that the system developed in this chapter builds on the calibration developed in Chapter 7, which requires that the robot's stereo vision system already be calibrated. As such, the software developed in this chapter enables the robot to enhance its existing camera calibration, but requires an initial seed calibration which is then improved upon.

In humans, one of the consequences of learning about the body and the senses through their coordination with each other is that this knowledge adapts as our bodies and senses change. The series of experiences discussed in the introduction to this dissertation in which glasses mounted with prisms and mirrors transform the visual fields of study participants [9, 10, 11] are one example of this. In this series of experiments, observations are made of how the mind adapts to shifts in the visual field over time. Going back to the study performed by Volcic et al. [14], the experimenters found adaptations in depth perception that are consistent with alterations in the perceived length of the arm, suggesting a tight integration between sensing and body structure as represented in the brain. In another previously discussed experiment [7], when study participants touched objects with an L-shaped tool they experienced the tactile sensation at the tip of the tool. This short-term sensory adaptation, integrating knowledge of the position of the tool with respect to the body, can be interpreted as an integration of the tool into the body schema. In this chapter, we will demonstrate the capability of the robot's integrated sensorimotor inference algorithm to adapt to changes in the body schema by incorporating a tool into the description of its kinematic chain.

8.1 Classical approaches to camera calibration

For the purposes of this discussion, we can think of camera calibration as the process of determining the parameters of the pinhole camera model which describe a pair of cameras in a stereo vision system. This process involves determining the position and orientation of the two cameras, their extrinsic properties and intrinsic properties. This section provides a very brief introduction to camera calibration techniques.

8.1.1 Estimating the intrinsic parameters

As discussed in Section 5.2.3, the intrinsic parameters are those which describe the camera itself, rather than its pose in a scene. To the novice exploring camera calibration methods, what is determined by camera calibration can be a confusing matter. This section will be divided into two parts. The first subsection describes methods which are often used to find the intrinsic parameters of a camera.[1] The second subsection describes methods for estimating a camera's epipolar geometry, from which the poses of the left and right cameras in a stereo rig with respect to each other can be identified.[2] For specifics on the software used to provide the initial calibration of Nico's stereo vision system, see Appendix B.

8.1.1.1 Photogrammetric calibration

Photogrammetric calibration involves imaging a target with known geometry (B_{model}) with a 3D model of this target loaded into the calibration software. Camera calibration is computed from the image of the target (b_{image}) by inferring the matrix that produces the projection necessary to produce the 2D image of landmarks along the target's surface ($P_{calibrated}$). This can be performed as an optimization of the form found in Equation 8.1. The most well-known algorithm to solve this calibration is the Direct Linear Transformation (DLT) algorithm [55].

$$argmin(||P_{calibrated}B_{model} - b_{image}||^2) \tag{8.1}$$

[1] Though photogrammetric calibration [65], Zhang's method [62], and bundle adjustment [82] can each be used to determine an estimate of a camera's extrinsic parameters.

[2] We only need to know their pose with respect to each other in order to perform stereo vision, since we can center our coordinate system arbitrarily. It is common practice to simply place the coordinate system such that the upper-left sub-matrix of the ideal P matrix describing the left camera without its intrinsic parameters applied ($[\ R \mid -RC\ ]$ is identity.)

8.1.1.2 The homography constraints

In another technique, constraints are derived from homographies computed over planar surfaces imaged by the camera. In many "non-naturalistic" scenes, such as photos of buildings in cities or the interiors of classrooms, such planes can be found as the faces of buildings, windows, walls, or the surfaces of tables, providing a means to use related techniques without the need for an explicit calibration target. The *homography constraint* can be used to calibrate a camera from a collection of homographies computed from imaged planes [55, 62].

When we look at an image of a scene, we notice that objects farther away from the camera appear to disappear into the distance. Equivalently, parallel lines appear to intersect in the horizon and evenly spaced points along a 3D line appear to come closer together as they move farther away from the camera.[3] The point at which two neighboring, equally-spaced points along such a line appear to be the same point is referred to as the line's *vanishing point.* Because we model cameras using projective geometry, we are able to conveniently represent vanishing points and the intersections of parallel lines as lying on a plane that is infinitely far away from the camera. This plane is referred to as the *plane at infinity.*

The *absolute conic* is a conic lying in the plane at infinity. The *image of the absolute comic* (ω), is a constant that is tied to the calibration of a camera. It can be computed using only the camera intrinsic matrix, as in Equation 8.2.

$$\omega = A^{-T} A^{-1} \tag{8.2}$$

The vanishing point of a 3D line in a scene can be computed as its intersection with a parallel line running through the camera center [55]. Relatedly, the cosine of

[3] Think of evenly spaced lampposts along a straight road that disappears into the horizon.

the angle between two scene lines can be computed with respect to their vanishing points by Equation 8.3, where v_1 and v_2 are the vanishing points corresponding to the two scene lines [55].

$$\cos\theta = \frac{\sqrt{v_1^T \omega v_2}}{\sqrt{v_1^T \omega v_1}\sqrt{v_2^T \omega v_2}} \tag{8.3}$$

Taking the X and Y axes from the homography provides us with two constraints based on this, Equations 8.4 and 8.5. Rearranging these terms forms a pair of linear constraints of the familiar form $Ax = 0$, allowing us to find the image of the absolute conic (ω). From this, we are able to compute an estimate of a camera's intrinsic parameters from a set of three or more homographies.

$$h_1^T \omega h_2 = 0 \tag{8.4}$$

$$h_1^T \omega h_1 = h_2^T \omega h_2 \tag{8.5}$$

8.1.1.3 Zhang's method

Zhang's method [62] is probably the most prevalent method currently used for calibrating cameras for use in stereo vision. It builds on methods described earlier in this section. In this method, a chessboard pattern is printed and adhered to a flat surface. This chessboard calibration target is then imaged in multiple poses.[4] Zhang's method attempts to combine the homography constraints with photogrammetric calibration in order to arrive at a method that combines the simplicity imaging planar targets with the precision of photogrammetric calibration.

[4] Not necessarily orthogonal poses.

Zhang s method works as follows:

1. Image chessboard calibration target in several poses (no fewer than three).
2. Compute a homography for each pose.
3. Compute a candidate calibration utilizing the homography constraints.
4. Determine the pose of the chessboard in each image, by Equation 8.6.
5. Perform optimization over chessboard poses and camera intrinsic parameters.

Zhang's method also provides a means for estimating lens distortion, which will not be discussed here. Details of this method can be found in [62].[5]

The homography induced by imaging a planar target with a camera can be described by Equation 8.6, where H is the homography, A is the camera intrinsic matrix, r_1 and r_2 are the first two columns of the rotation matrix describing the pose of the target with respect to the camera, and t is the translation of the target with respect to the camera.[6]

$$H = A \begin{bmatrix} r_1 & r_2 & t_3 \end{bmatrix} \tag{8.6}$$

After estimating A via the homography constraints, Zhang's method can estimate the pose of the chessboard calibration target in each image by Equation 8.7. Knowing that r_1 and r_2 are orthogonal unit vectors, after normalizing the resultant matrix such that $||r_1|| = 1$ and $||r_2|| = 1$, r_3 can be found via the cross product, providing the full pose of the chessboard calibration target. Optimization is then performed over the entire model of chessboard poses and camera calibration parameters by minimizing

[5] In the case of Nico's camera calibration software, the linear approximation of lens distortion is not implemented. Instead, the optimizer starts with a model assuming no lens distortion and estimates lens distortion strictly via optimization.

[6] For a derivation of this, see [62].

the distance between projected images of the corners of an internal model of the chessboard calibration target and their imaged counterparts.

$$\begin{bmatrix} r_1 & r_2 & t_3 \end{bmatrix} = A^{-1}H \tag{8.7}$$

8.1.1.4 Bundle adjustment

Bundle adjustment [82] is generally considered to be a necessary approach to arriving at accurate descriptions of scene geometry reconstructed from a series of images. It can be thought of as adjusting the model of the bundle of rays of light passing through the aperture of the camera in order to accurately model them.

In bundle adjustment, 3D reconstructions of scene points are computed from their 2D images. These scene points then become a sort of calibration target for the camera. One method for doing this is to reproject the reconstructed 3D scene points back to 2D. The distance between the original 2D imaged point and the reprojected 2D model point (reprojection error) is then minimized over the full calibration of each camera.[7] In doing so, the reconstructed model of the 3D geometry of the imaged object becomes better during optimization (due to improved camera calibration parameters), and the camera calibration parameters become better over time (due to improvements in reconstructions of the imaged object).

[7] This can be done between the left and right cameras in order to perform a stereo calibration. When done between frames of video with a moving camera or moving object, this technique is known as structure from motion. Bundle adjustment can be effectively used in both of these scenarios. In the case of a moving camera, recovering the full pose of the camera, generally in the form of the extrinsic parameters of the camera's projection, is a form a visual odometry. Many variations of this formulation are possible.

8.1.2 Estimating epipolar geometry

The goal in estimating the epipolar geometry of Nico's stereo vision system is to recover the position of the left and right cameras with respect to each other. As discussed in Section 5.3, for cameras with known intrinsic parameters the extrinsic parameters can be recovered by means of factorization from the essential matrix, as in Equation 5.15. A number of methods are available to perform this process [55, 83, 84], but in this subsection only two will be discussed.

8.1.2.1 Factorizing from other calibrations

Calibration methods such as photogrammetric calibration, Zhang's method [62], and bundle adjustment [82] often produce estimates of the pose of the camera with respect to a calibration target as part of their output.

In the case of photogrammetric calibration, the transformation describing the two cameras can be found by multiplying the inverse of the 4×4 transformation positioning points with respect to the left camera by the 4×4 matrix transforming points with respect to the right camera.

Zhang's method produces a number of transformations with respect to poses of the calibration target as part of its output. From each of these transformations, the transformation between the left and right camera can be computed similarly to the case of photogrammetric calibration, and a mean can be computed from this list of transformations.

In the case of bundle adjustment, if bundle adjustment is performed between a stereo pair of cameras, the resultant projective matrices will reflect their position and orientation with respect to each other.

8.1.2.2 The eight-point algorithm

The eight-point algorithm is a formulation of a linear system that can be used to estimate the fundamental matrix [78]. By Equation 5.12, it can be seen that a linear system can be constructed by choosing matched points in the left and right images of a stereo pair that allows for the fundamental matrix to be found as a least-squares fit via singular value decomposition (SVD). The fundamental matrix is a 3×3 homogeneous matrix, meaning that the transformation applied is invariant to scalar multiplication. As such, F has eight degrees of freedom. Each matched pair of points only provides one constraint, the distance determined by Equation 5.12. Therefore eight or more such pairs of matched points are required to find a least-squares estimate of F via this formulation, giving the algorithm its name.

In order to improve performance, it is common to normalize sampled image points in the left and right images prior to application of the eight point algorithm. This normalization puts the centroid of the collected image points at the origin, and sets their mean distance from the origin to $\sqrt{2}$, making the "average" point $< 1, 1, 1 >$. [78] The resultant fundamental matrix is then denormalized by multiplying by the inverses of the normalizing transformations. In this case, the algorithm is referred to as the *normalized eight-point algorithm.*

Once the matrix F has been estimated, it is possible to recover E by multiplying the inverses of the A matrices for the left and right cameras, inferred by other camera calibration techniques. From E, the extrinsic parameters can be obtained via factorization. The formulation of this factorization is beyond the scope of this discussion. It should be noted here, however, that the factorization of R and t is only accurate up to a four-fold ambiguity [55]. Each of these factorizations admit the same images of one of four hypothetical scene points, imaged by one of four sets of cameras. In only one of these factorizations, however, is the point in front of both cameras.[8] Since a physical camera can only image points that are in front of it, the factorization wherein the point is in front of both cameras is selected as the correct one.

8.2 Using the body as a visual calibration target

Taking cues from techniques such as photogrammetric calibration and bundle adjustment and inspiration from the fact that infants learn about their senses through the interaction of those senses with their bodies, we can now develop a technique that allows a robot to refine its visual calibration by witnessing the motion of its body in its visual field.

If a model of the robot's kinematics is known a priori, then imaging the position of its hand in multiple poses produces a collection of known 3D points in space. Such a collection can then be used as the collection of model points for photogrammetric calibration. The motion of the arm produces the set of 3D points in space, rather than an object of known 3D structure.

By using the techniques developed in the previous chapters to infer a model of the robot's kinematics we are able to create such a model. Once inferred, this model can be used in camera calibration processes. In Chapter 7 we presented formula for predicting the 2D position of the robot's end-effector in its visual field, Equation 7.4. In Chapter 6 we optimized the 3D distance between the position of the end-effector measured by the robot's stereo vision system and the position predicted by the robot's self-model. Though this measurement is performed in 3D, by Equation 7.4, we can extend this system to optimize the distance between the

[8] Since, mathematically, it is possible to image a scene point that is behind the pinhole camera.

2D projected predictions of end-effector position in the robot's visual field, as in Equation 8.8, where $p_{tracked_j}$ is the tracked end-effector position in 2D in the visual field over the left and right camera views, respectively. If we optimize only the kinematic parameters, then this algorithm performs a nonlinear refinement similar to that presented in Section 6.2.3 performed in 2D rather than 3D. If we optimize over only the robot's visual parameters, then this algorithm constitutes a form of photogrammetric calibration, by way of the robot's kinematic model. Performing this optimization globally, over both the kinematic and visual portions of the robot's self-model, constitutes a camera calibration algorithm similar to bundle adjustment. Rather than simultaneously refining a reconstruction of a 3D object and the camera calibration parameters used in that reconstruction, we refine the inferred kinematics of a robot alongside these camera calibration parameters. This mirrors the notion stated in the introduction of this dissertation, taking inspiration from the infant developmental process. The robot learns about its body through its senses, and about its senses by using them in conjunction with its body.

$$argmin(\sum_{j=1}^{m} K[R| - RC]M_0 \ldots M_n[0,0,0,1]^T - p_{tracked_j}) \tag{8.8}$$

8.3 Evaluation

Having already collected the relevant datasets during the evaluations performed in Chapter 7, the evaluations of the algorithm presented in this chapter towards the application of inferring a kinematic-visual self model in this chapter proceed by applying the algorithm developed in this chapter to a model seeded as the product of the tests in Chapter 7.

In this chapter we present an analysis of both improvements in the learned kinematic model and improvements in the refined visual calibration as a product of using the techniques presented in this chapter, which allow the simultaneous refinement of the visual and kinematic aspects of the self-model.

Additionally, we evaluate the capability of the robot to adapt its self-model by demonstrating its ability to incorporate a tool mounted in its hand into the self-model. In this test the robot has a screwdriver mounted in its end-effector, Figure 8.1. Because the fiducial marker would not stick to the chrome finish of the screwdriver, only color blob detection was used in this test. The model was seeded with the self-model trained in Chapter 7, trained on a dataset of 52 random arm poses with the tool mounted in its hand, and tested on a set of 100 additional random arm poses.

8.4 Results

The technique of using the body as a visual calibration target does not work when the end-effector is tracked via color blob detection. This is because the tracked centroid in the left and right images may lie along a large patch of the colored surface, whereas the chessboard corners are localized to sub-pixel accuracy. As such, the matching between pixels in the left and right images is less precise when observing the tracked color blob, causing the calibration of the vision system to be harmed when using this method. For that reason, the discussion of results in this section will focus on the case of using the stereo vision system with fiducial tracking in all cases except for tool use.

8.4.1 Impact on stereo vision calibration

Estimates of the accuracy of stereo reconstruction were computed before and after full model learning. In order to do so, we estimate the accuracy of stereo reconstruc-

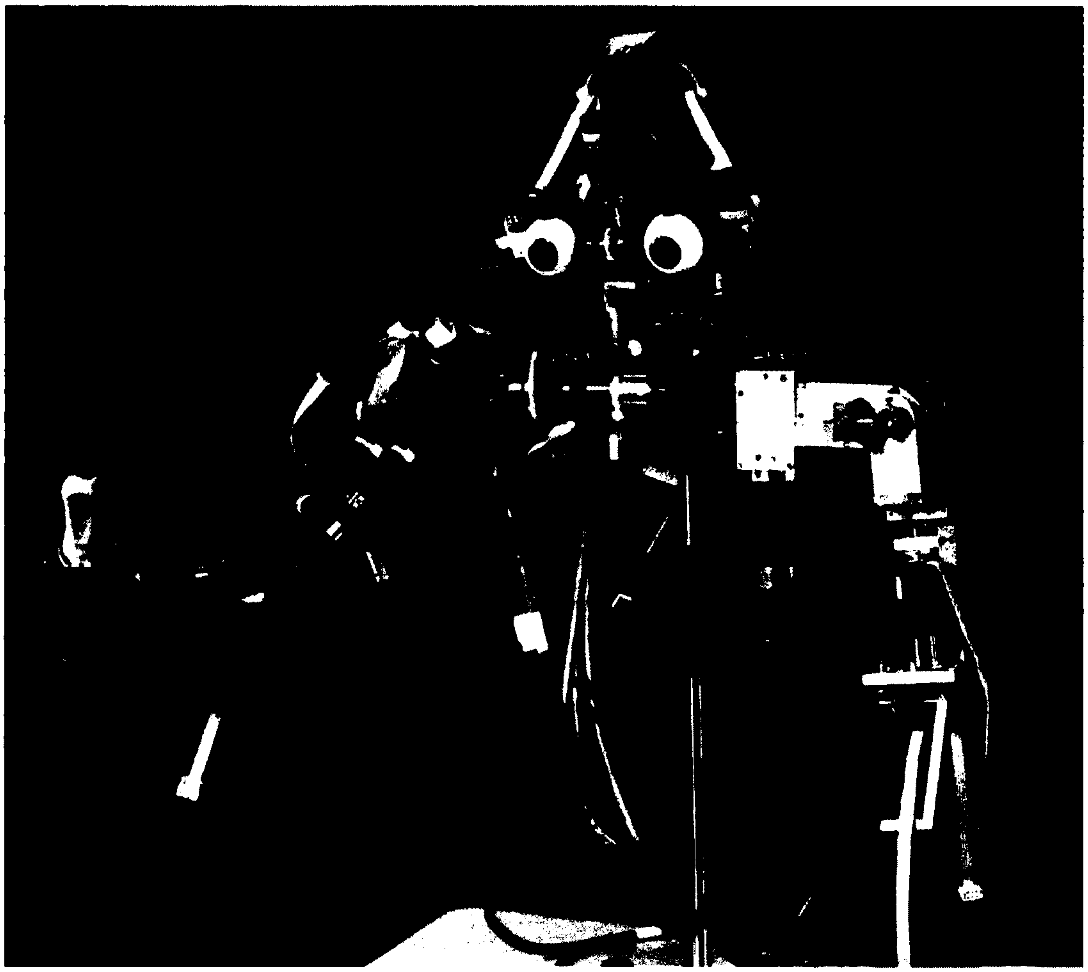


Figure 8.1: The humanoid robot, Nico, with a screwdriver taped to its gripper. The tip of the screwdriver has been marked with red tape in order to enable the vision system to track its position. Photo credit: Sadie Wechsler.

tions by reconstructing the 3D positions of the chessboard corners on a chessboard calibration target in multiple poses, using the dataset acquired to initially calibrate the robot's cameras. Knowing that the sides of each square on the chessboard are 28mm long allows us to compute the conversion factor from the computer vision system's internal units to metric units. We use the variance in the length of the reconstructed chessboard square sides as an indicator of the positional accuracy of stereo reconstructions. In the case of color blob detection, vision accuracy decreases, from a standard deviation in the length of reconstructed chessboard squares of 1.30mm to 2.79mm. In the case of fiducial tracking the simultaneous refinement of visual and kinematic parameters improves the quality of stereo reconstructions. Estimates of the accuracy of stereo reconstructions improve from within 1.59mm to within 1.31mm, over the original set of stereo calibration data, including chessboard calibration targets imaged in a variety of poses at several different ranges. While the improvement is slight, we have previously demonstrated greater improvements [56]. Optimizing the system's 2D performance on predicting the position of its end-effector in the visual field allows for the robot to refine its camera calibration by using the robot's kinematics as its calibration target.

8.4.2 Performance in predicting end-effector position

Because fiducial tracking works better with this technique than color blob detection, we will focus on fiducial tracking results for the time being. In the case of fiducial markers, wide variances in localization of the marker in the visual field are not an issue, because the tracked centroid is very precisely placed at the center of the marker, through the techniques discussed in Section 7.1.1.2. In this section, three results are presented for comparison. The first is the performance of the system after kinematic learning, as presented in Chapters 6 and 7. The second is the results for the simultaneous refinement of the robot's kinematic and visual parameters, as presented in this chapter, with the camera intrinsic parameters pinned.[9] In this case, only the positioning of the cameras with respect to each other, and the kinematic parameters of the robot may be updated by the optimizer. As such, this result shows the improvement in performance in the absence of updated camera intrinsic calibrations for the left and right cameras for the robot's stereo pair. The final result presents the improvement in performance when the robot is able to optimize the full model, simultaneously refining kinematic and visual parameters with respect to each other.[10]

Figure 8.2 compare the performance of kinematic estimation and simultaneous refinement of kinematic and visual parameters in 2D, as measured in pixels. We can see that the combined learning consistently improves 2D performance. Of course, because the 2D distance between predicted end-effector position and measured end-effector position is the quantity that is optimized by Equation 8.8, improvement on this measure is expected, as long as the algorithm works properly and does not over-fit the model. Performance improves from M=4.55 pixels (SD=2.74 pixels) in the case of learning only kinematics to M=2.61 pixels (SD=1.71 pixels) in the case of optimization over the full model. When we pin the calibration parameters of the cameras, we see a more modest improvement in performance (M=3.38 pixels SD=1.99 pixels).

Figure 8.3 provides statistics regarding the 3D performance of the system. Interestingly, we see that when the camera intrinsic parameters are pinned, it harms 3D performance. In this case, over-fitting 2D performance on an imperfectly calibrated vision system comes at the expense of 3D performance (kinematic estimation:

[9] To pin a parameter is to disallow the optimizer from changing it.

[10] Recall that lens distortion is not considered during these optimizations.

M=2.74mm SD=1.39mm, M=0.011arm SD=0.0054arm; vision parameters pinned: M=4.23mm SD=2.87mm, M=0.016arm SD=0.0011arm). By training the kinematic and visual models to each other, however, the two converge. The continually improving estimates of the robot's kinematic structure improve its ability to serve as a stereo calibration target,[11] while the refinements in stereo reconstruction inform the accuracy of the kinematic model. Note that the extrinsic parameters can be tuned by the optimizer in both cases. Therefore, the improvement is in part due to the algorithm's ability to improve the intrinsic calibration of each camera. The manner in which improvements in the accuracy in one model inform the calibration of the other is reminiscent of Rochat's theory of the development of the Ecological Self [3]. The final model under simultaneous refinement is accurate to within 1.99mm (SD=1.24mm), 0.0078arm (SD=0.0048arm). This demonstrates the ability of the robot to successfully use its own body as a calibration target for its vision system, while simultaneously learning a more accurate model of its kinematics.

8.4.3 Estimates of linkage lengths

The arm of the robot used in this experiment comprises two main segments with paired joints at the intersection of those segments. To verify the estimated model of the robot's kinematics, external measurements of the two main segments were obtained for comparison against the robot's internal estimates. The first segment goes from the robot's shoulder to its elbow and is 130mm long. The second goes from the elbow to the end-effector and is 127mm long. As shown in Figures 8.4 and 8.5, estimates of linkage lengths are accurate to within 1cm (7% of the length of the linkage) for both linkages, respectively (First: 139.47mm, 1.073arm; Second:

[11] Compare to classical photogrammetric techniques, in which the projection of a target of known shape is computed from images of it.

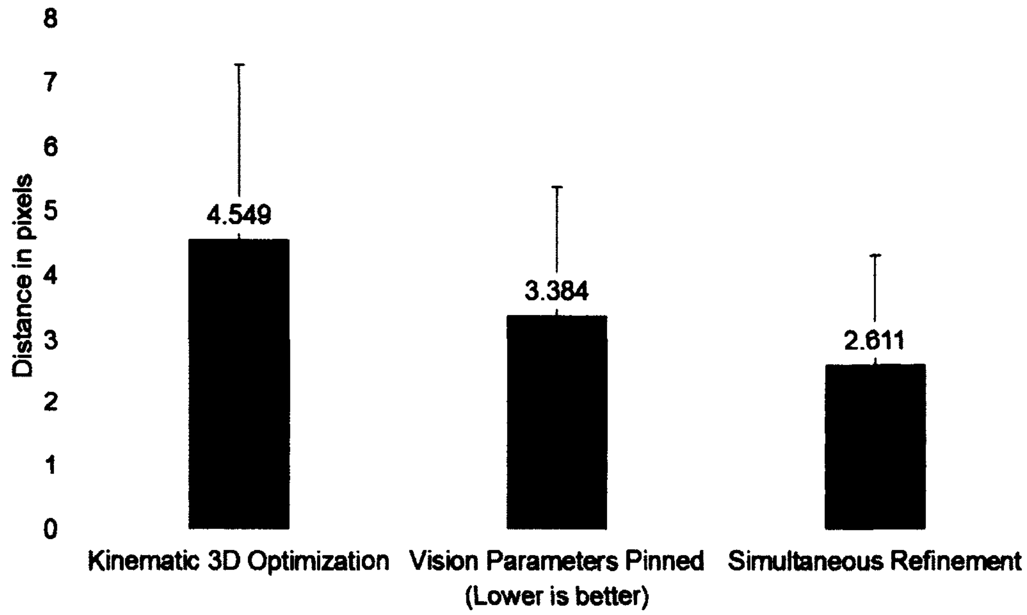


Figure 8.2: Comparison of performance in 2D between kinematic and full-model learning. The test is performed over 100 random samples. Results labeled "Kinematic Learning" use CPA and nonlinear refinement. Results labeled "Full Model Learning", and "Intrinsics Pinned" use the technique outlined in Section 8.2, to improve on the "Kinematic Learning" results. The "Intrinsics Pinned" case does not attempt to refine the camera intrinsic parameters.

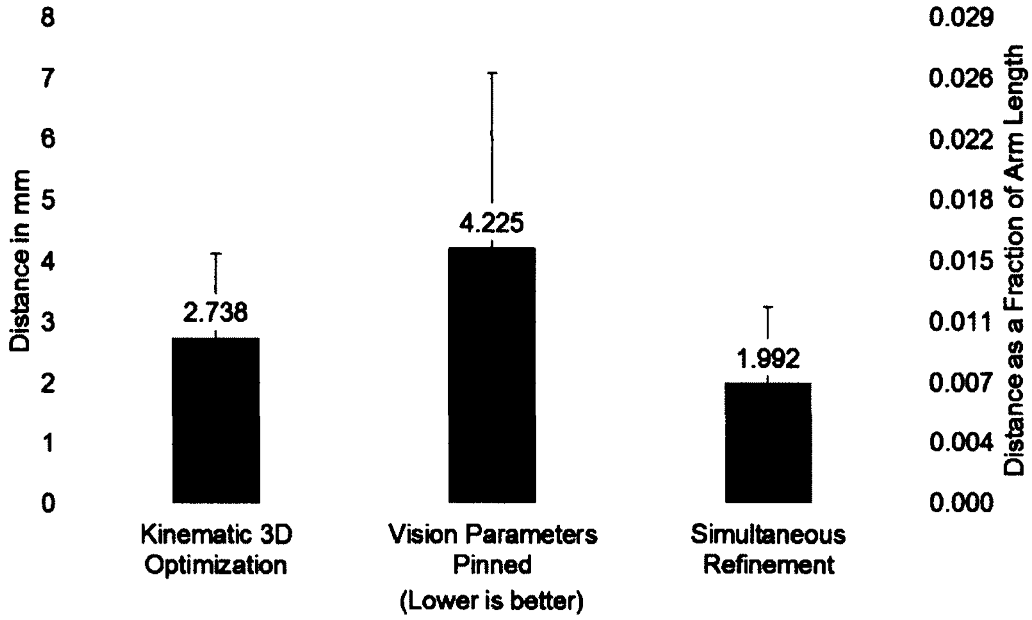


Figure 8.3: Comparison of performance in 3D between kinematic learning and full-model learning.

132.57mm, 1.044arm), when trained using full-model learning.

8.4.4 Tool use

As previously discussed, the system is able to adapt to tool use by retraining an already-initialized model using the techniques developed in this chapter. For this test, a screwdriver was placed into the robot's end-effector, its tip marked with colored electrical tape, as in Figure 8.1. This test updates the model presented in Section 7.3, in which the robot learns a model of its arm kinematics while tracking the tip of its index finger, marked with red electrical tape. Because color blob tracking data cannot be used for camera calibration, both camera intrinsic and extrinsic parameters are pinned. The system, as trained on the robot's hand, tracked its end-effector to within 5.72mm (SD=5.00)mm, 0.020arm (SD=0.018)arm, and 3.82 pixels (SD=2.33) pixels. Upon retraining with the screwdriver, the system adapted, track-

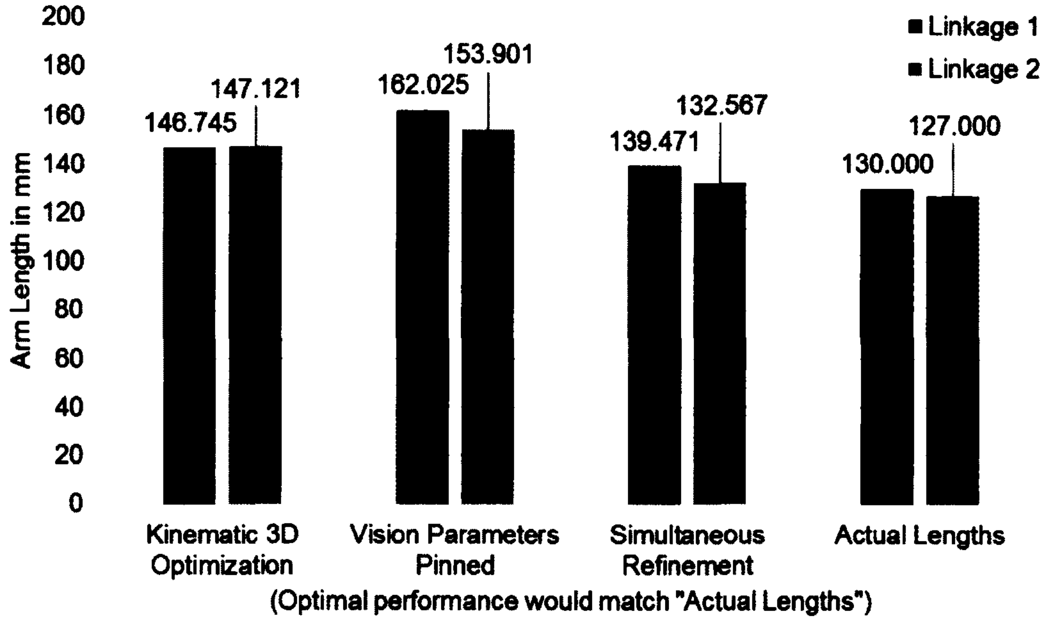


Figure 8.4: Estimate of linkage lengths expressed in millimeters, compared to linkage lengths as measured using calipers.

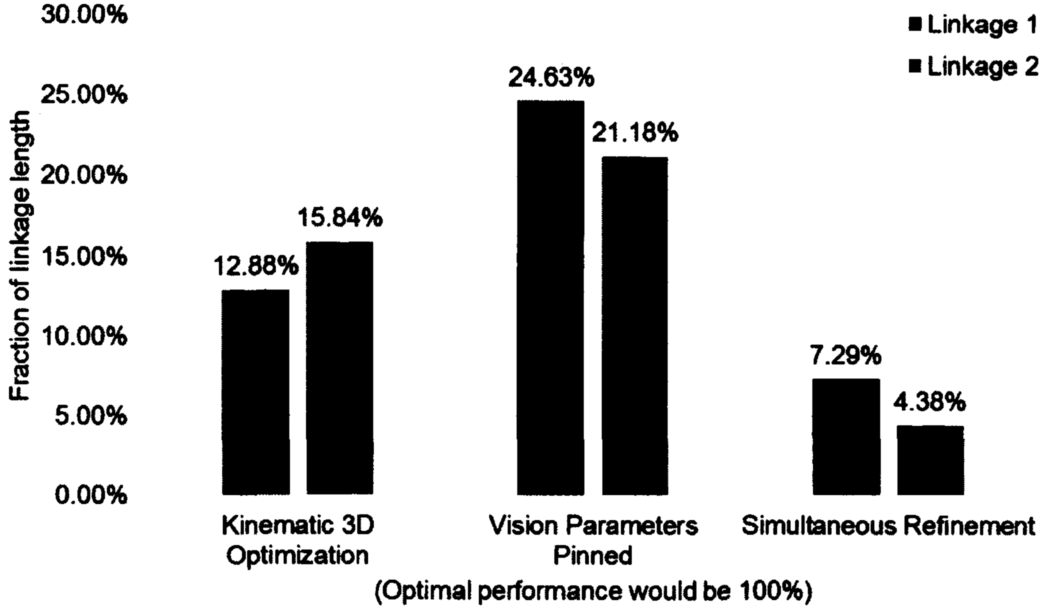


Figure 8.5: Estimate of linkage lengths expressed as a fraction of linkage lengths measured using calipers.

ing the end-effector to within 7.42mm (SD=5.99)mm, 0.026arm (SD=0.021)arm[12], 5.09 pixels (SD=3.09) pixels.[13] Performing nonlinear refinement in 3D prior to optimizing the system over the 2D positions in the visual field provides a slight boost in performance, (M=6.83mm SD=5.47mm, M=0.024arm SD=0.019arm, M=5.06 pixels SD=3.19 pixels).

8.5 Summary and conclusions

In this chapter, the kinematic inference process developed over the previous chapters is enhanced in order to enable the system to simultaneously refine the kinematic and visual parameters of its self-model. The key factor that allows us to do this is the fact that the system uses a united kinesthetic-visual self-model, rather than separate kinematic and vision models. The ability of the system to predict its end-effector position in 2D, as projected into the robot's visual field, rather than only in 3D, means that we can treat the robot's kinematic structure as a calibration target. This is reminiscent of existing camera calibration approaches discussed in this chapter, in which a target of known shape is imaged (photogrammetric calibration), in which a reconstruction of scene structure is reprojected into the visual field in order to allow reconstructed targets to serve as calibration targets (bundle adjustment), and in which initial estimates provide the priming necessary to use optimization in order to produce a more-refined calibration (Zhang's method [62]).

Evaluations of the method of using the robot's kinematic structure as a camera

[12] The use of "arm" here is ambiguous. Here, we mean as a fraction of the length of the arm without the screwdriver mounted, so results are comparable to the original results for performance against the end-effector.

[13] In this case, the tip of the screwdriver is treated as a new position for the end-effector in the existing kinematic chain. This is to say, it is as if the tip of the finger is replaced by the tip of the screwdriver, not as if a new element is introduced after the terminal joint in the existing kinematic chain.

calibration target show that this technique both improves the robot's estimate of its kinematic structure and improves its camera calibration. This shows the power of the technique to be used for online calibration on real robots. Because the technique operates on data that can be sampled by the robot as it operates, the possibility exists of using it to adjust and adapt a self-model in an online fashion, during operation. We demonstrate this by having the robot adapt its self-model in order to incorporate a tool into its kinematic chain.

Chapter 9

Inferring the visual perspective describing reflections in a mirror

When we look into a mirror, the image that we see is a reflection of what actually exists in space. Objects in this reflection appear as if they exist on the other side of the mirror, opposite their real-world counterparts. If one were to naïvely reach towards these reflections, their hand would hit the glass of the mirror, rather than the object that they are reaching for. By understanding this reflection, however, one is able to use the mirror as an instrument to make accurate inferences about the positions of objects in space based on their reflected appearances. When we check the rear-view mirrors on our cars for approaching vehicles or use a bathroom mirror to orient a hairbrush, we make such instrumental use of these mirrors.

As discussed earlier in this dissertation, the use of mirrors for spatial reasoning is a precursor to what is tested in the mirror test, as originally proposed by Gallup [16]. The mirror test has become the classical test of self-awareness in humans and animals. In this test, after an animal is given time to acclimate to the presence of a mirror, it is anesthetized and marked on the face with odorless, non-tactile dye.

The animal's reaction to their reflection is used as a gauge of their self-awareness, based on whether they inspect the mark on their own body, or react as if it does not appear on themselves, as in cases where they react as if it is a mark on another animal. Children[1] develop the necessary skills to pass this test by around 18 months [15].

Tests have been devised to determine whether animals that are unable to pass the classical mirror test are able to use mirrors as instruments to solve spatial reasoning tasks. These tests have shown that there is a larger category of animals that are capable of such instrumental use. Infants who are too young to pass the mirror test can retrieve an object that is presented behind them in a mirror at around 8 months, demonstrating a self-centered awareness of space and reflectance [15]. Marmosets (which fail the mirror test) are able to use a mirror to obtain food pellets that are visible only in a mirror reflection [24]. Using both mirrors and monitors displaying live video feeds of their arms, chimpanzees can overcome inversions and rotations of these images, manipulations which break the spatial relationship that can be established by looking into a mirror. They are able to use images for spatial reasoning, thus demonstrating even more general spatial reasoning capabilities than mirror use [25].

In this chapter, the self-model developed in the preceding chapters is used in order to infer the visual perspective of a mirror in the robot's environment. Knowing its kinematics, and having its kinematic model tightly calibrated to its stereo vision system, the robot is able to again use its body as a visual calibration target. This time, knowledge of how the body moves in space allows the robot to calibrate the visual perspective describing reflections in a mirror by constructing a virtual camera that exists as if on the other side of the mirror. The projections of 3D scene geometry

[1] Who are discreetly marked with rouge makeup, rather than anesthetized and marked.

made by this virtual camera accurately represent the image reflected in the mirror. As such, the robot can construct a pair of virtual cameras describing the relationship of its stereo vision system to reflections in the mirror. The visual perspectives of these cameras are consistent with observations of the position of the robot's end-effector, as reflected in the mirror when the robot moves into different poses. In this way, self-knowledge regarding its kinematics and vision system enables the robot to use a mirror for spatial reasoning.

9.1 The mirror-perspective model

Consider the scenario in Figure 9.1. The robot is only able to observe the reflection of its end-effector as reflected in the mirror, as the hand is not directly positioned in the visual field. Naïve reconstructions of reflections of the end-effector's position in space will place it behind the plane of the mirror, rather than in front of the mirror where it actually is. In order to overcome this, the system developed in this chapter will accurately reconstruct positions of objects reflected in the mirror by reconstructing them from the perspective of a virtual stereo vision system, based on the reflections of the perspectives of the physical cameras in the robot's stereo vision system, as in Figure 9.2.

The mirror-perspective model allows the robot to estimate this visual perspective. It leverages the robot's perceptual and end-effector models, allowing the robot to compute a virtual calibration for the perspective of each of its stereo cameras, for objects that they witness reflected in the mirror. We will call each of these cameras a *mirror-perspective camera.* The basic method for calibrating these cameras is for the robot to move into several poses, yielding a known set of 3D points in space, and their corresponding 2D images. In this way, the technique in this section is a form of

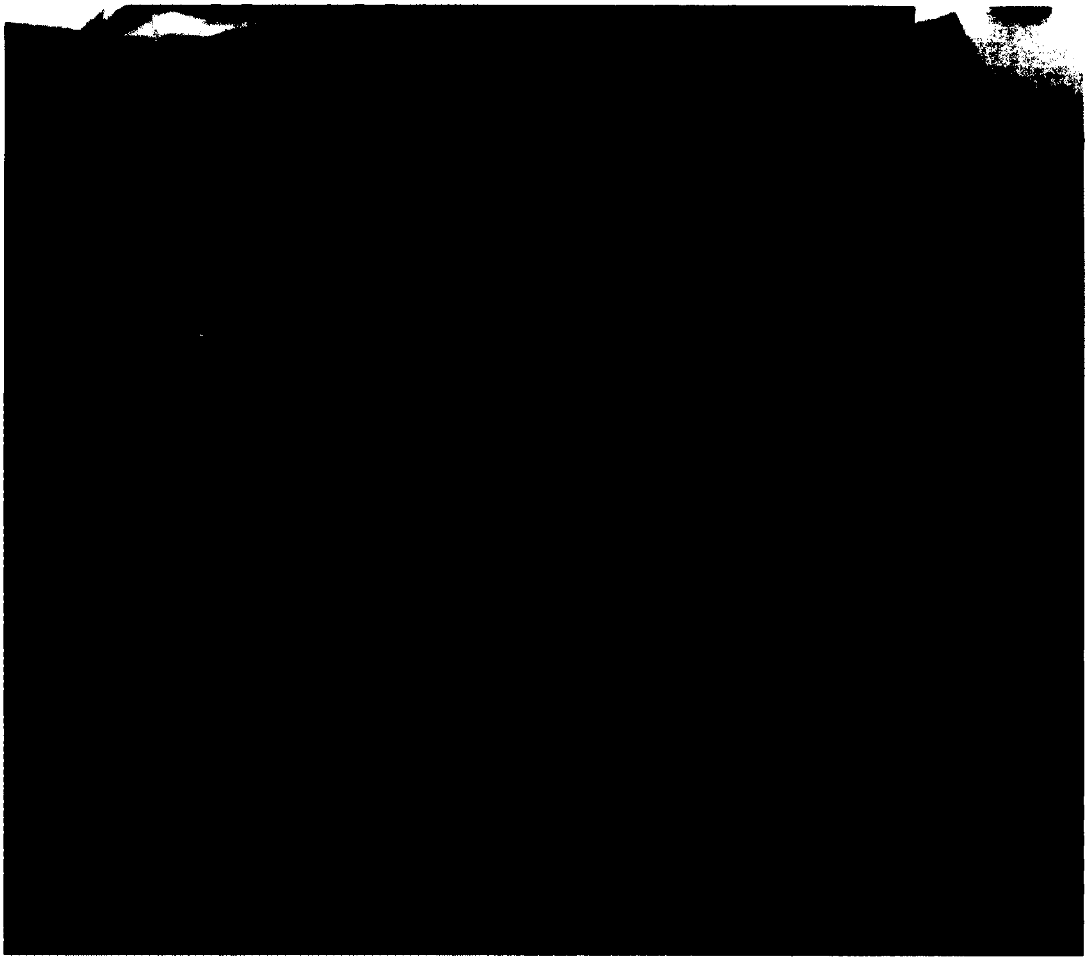


Figure 9.1: The humanoid robot, Nico, as configured for evaluation of the system.

photogrammetric calibration [55], with the robot acting as its own calibration target. The ability to perform this kind of calibration is an extension of the self-calibration technique developed in Chapter 8.

The self-model which we have developed throughout the course of this thesis will be the starting point from which the robot will be able to estimate the mirror-perspective model. We will label the position of the end-effector, computed by the end-effector model developed in previous chapters, J, as in Equation 9.1. The matrix K describes the intrinsics of a calibrated camera, as in Equation 9.2. The extrinsic parameters describing the positioning of the left and right stereo cameras in space

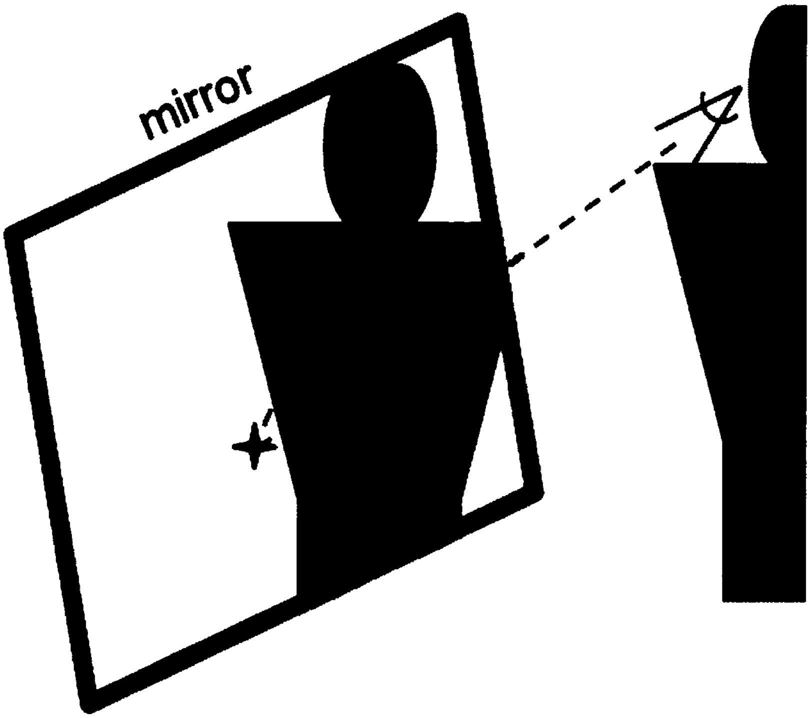


Figure 9.2: The mirror-perspective model works by computing the visual perspective representative of reflections in the mirror. The position and orientation of this visual perspective are computed as the reflections of the position and orientation of the physical camera.

will be labeled R and C, as in Equation 9.3. They can be combined with the intrinsics as in Equation 9.4. The 2D projection of the position of the end-effector in the visual field can be predicted by Equation 9.5.

$$J_E = M_0 \dots M_n[0,0,0,1]^T \tag{9.1}$$

$$K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9.2}$$

$$O = [R| - RC] \tag{9.3}$$

$$P = K[R| - RC] \tag{9.4}$$

$$j_E = K[R| - RC]M_0 \dots M_n[0,0,0,1]^T \tag{9.5}$$

Because this model deals with mirrors, it will frequently be the case that variables are related based on their disposition with respect to the mirror. Variables referring to quantities based on reflections, rather than the original, physical properties of the robot, are marked with a caret. For example, whereas J_i is the position of the robot's end effector, $\hat{J}_i$ is the reconstruction of its reflection in the mirror. Because the robot samples many poses of its arm, the subscript i is used to refer to a set of variables describing a single pose.

Because an object's image in a mirror is a reflection of its real-world counterpart, its position in space can be correctly interpreted from the perspective of a camera whose position and orientation have been reflected with respect to the plane of the mirror. The goal of this model is to determine the parameters describing the mirror-perspective camera, $\hat{P}$, of a real-world camera, P, observing objects reflected in a mirror. The mirror-perspective can be determined by reflecting the camera's position and orientation against the plane of the mirror. The intrinsic parameters for the camera and its reflection are the same, requiring only the position $\hat{C}$ and orientation $\hat{R}$ of the mirror-perspective camera to be estimated. In the case of this system, an initial estimate of the calibration of the mirror-perspective camera is computed in this fashion. It is then refined through an extension of the technique developed in Chapter 8.

9.1.1 Estimating the mirror-perspective camera

The calibration process for mirror-perspective cameras mirrors the calibration processes used throughout this dissertation. First, a rough estimate of the correct calibration for the mirror-perspective camera is obtained. This estimate is then refined. In the case of the mirror-perspective camera, the initial estimate is obtained from an estimate of the plane in which the mirror lies. This estimate is then refined by substituting in the estimated camera and performing the visual refinement first developed in Chapter 8. Because we are able to assume that the robot's arm is well-calibrated, instead of updating the arm calibration we pin the parameters describing the robot's kinematics and perform optimization only over the virtual projection describing the mirror camera.

The procedure for inferring the perspective of a mirror is as follows:

1. Sample reflected end-effector images - Record three versions of end-effector position:

   J_i Predicted by the end-effector model and appearing in front of the mirror.

   $\hat{J}_i$ Reconstructed by the perceptual model from the point of view of the robot's cameras and appearing behind the mirror.

   $\hat{j}_{l_i}$ & $\hat{j}_{r_i}$ Two dimensional positions of the end-effector in both cameras, appearing as reflections in the mirror.

2. Compute an initial estimate of each mirror-perspective camera, based on the plane in which the mirror lies.

3. Nonlinear refinement of the pair of mirror-perspective cameras.

9.1.1.1 Sample reflected end-effector images

First the robot moves its end-effector into a set of random poses that can be witnessed in the mirror. It records J_i, $\hat{J}_i$, and $(\hat{j}_{li}, \hat{j}_{ri})$ for each pose. J_i is the position of the end-effector computed by the robots end-effector model. The coordinates $(\hat{j}_{li}, \hat{j}_{ri})$ are the images of the end-effector's reflection in the mirror. $\hat{J}_i$ is reconstructed from $(\hat{j}_{li}, \hat{j}_{ri})$ by the robot's perceptual model.

9.1.1.2 Compute initial estimate

To simplify the process of computing the estimate of mirror-perspective camera position and orientation, we assume that the camera is situated at the origin with $R = \mathbb{I}$. Because the robot's cameras are calibrated, this can be accomplished by transforming the sampled J_i's and $\hat{J}_i$'s into the camera's coordinate frame prior to computing the mirror plane, and transforming $\hat{R}$ and $\hat{C}$ back after they are computed.

Mirror plane estimation: Because J_i and $\hat{J}_i$ should lie symmetrically about the plane, for each arm pose, the plane in which the mirror lies can be approximated as follows.

First the vector perpendicular to this plane is computed. This is computed as the mean vector from the J_i's to the $\hat{J}_i$'s, using their Cartesian representations, as shown in Equation 9.6.

$$< \Pi_1, \Pi_2, \Pi_3 >= \frac{\sum_{i=1}^{n} \hat{J}_i - J_i}{n} \tag{9.6}$$

The plane corresponding to the correct orientation, centered at the origin is then computed by Equation 9.6. The distance of the J_i's and $\hat{J}_i$'s from this plane can then be used to compute Π_4, as in Equation 9.8, placing the mirror plane equidistant to the two sets of points.

$$Q_{origin} = < \Pi_1, \Pi_2, \Pi_3, 0 > \tag{9.7}$$

$$\Pi_4 = -\frac{\sum_{i=1}^{n} Q_{origin} \cdot \hat{J}_i + \sum_{i=1}^{n} Q_{origin} \cdot J_i}{2n} \tag{9.8}$$

$$Q = < \Pi_1, \Pi_2, \Pi_3, \Pi_4 > \tag{9.9}$$

Estimating mirror-perspective camera position and orientation: Figure 9.3 provides a diagram of the position and orientation of the real camera and the mirror-perspective camera with respect to the mirror. The reversal of mirror images is accounted for by having the mirror-perspective camera oriented such that it is looking away from the mirror, as if points are being imaged from behind it.

Computing the mirror-perspective camera's position: Because Q is expressed in the camera's coordinate frame, $< \Pi_1, \Pi_2, \Pi_3 >$ is the vector perpendicular to the mirror from the camera's position. Normalizing Q such that $< \Pi_1, \Pi_2, \Pi_3 >$ is a unit vector allows the position of the mirror-perspective camera to be computed by Equation 9.10.

$$\hat{C} = -2\Pi_4 < \Pi_1, \Pi_2, \Pi_3 > \tag{9.10}$$

Computing the mirror-perspective camera's orientation: Camera projection matrices can be interpreted as sets of three planes from which a signed distance of a 3D point is computed in order to determine its projection. Relatedly, the rows of camera extrinsic matrix (O), Equation 9.3, describe planes that position and orient the camera's coordinate system. The first two rows of O describe planes lying between the X

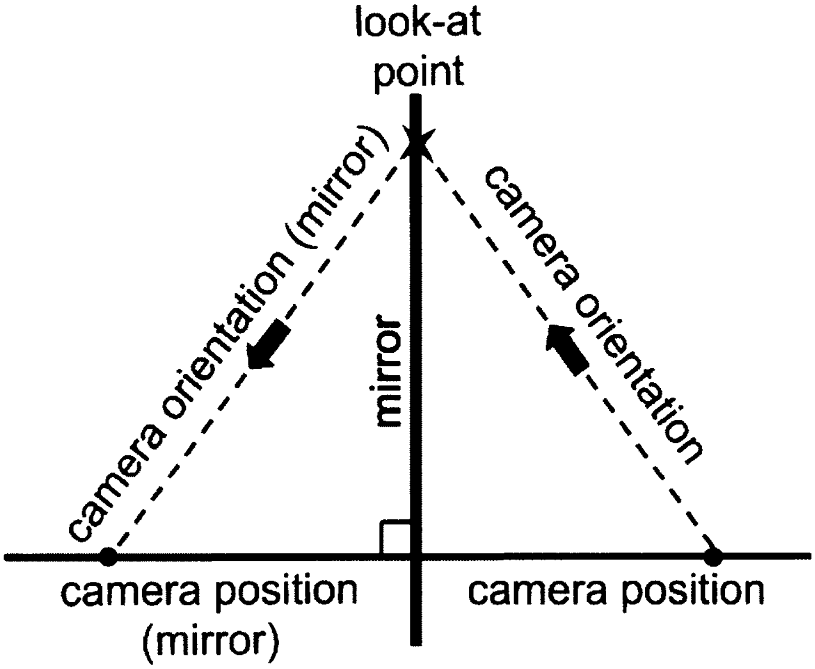


Figure 9.3: Diagram of the position and orientation of the real camera and mirror-perspective camera with respect to the mirror.

and Z axes, and the Y and Z axes of the camera's coordinate system, respectively. Knowing that three planes meet at a single point, the intersection of the camera's z-axis with the mirror plane, L, can be computed according to Equation 9.11. The z-axis of the mirror-perspective camera, then, can be computed according to Equation 9.12. Its rotation, $\hat{R}$, is the transpose of the rotation from the canonical z-axis ($< 0, 0, 1 >$) to the mirror-perspective camera's z-axis, computed as a rotation about the axis perpendicular to both.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \Pi_1 & \Pi_2 & \Pi_3 & \Pi_4 \end{bmatrix} L = 0 \tag{9.11}$$

$$\hat{Z} = \hat{C} - L \tag{9.12}$$

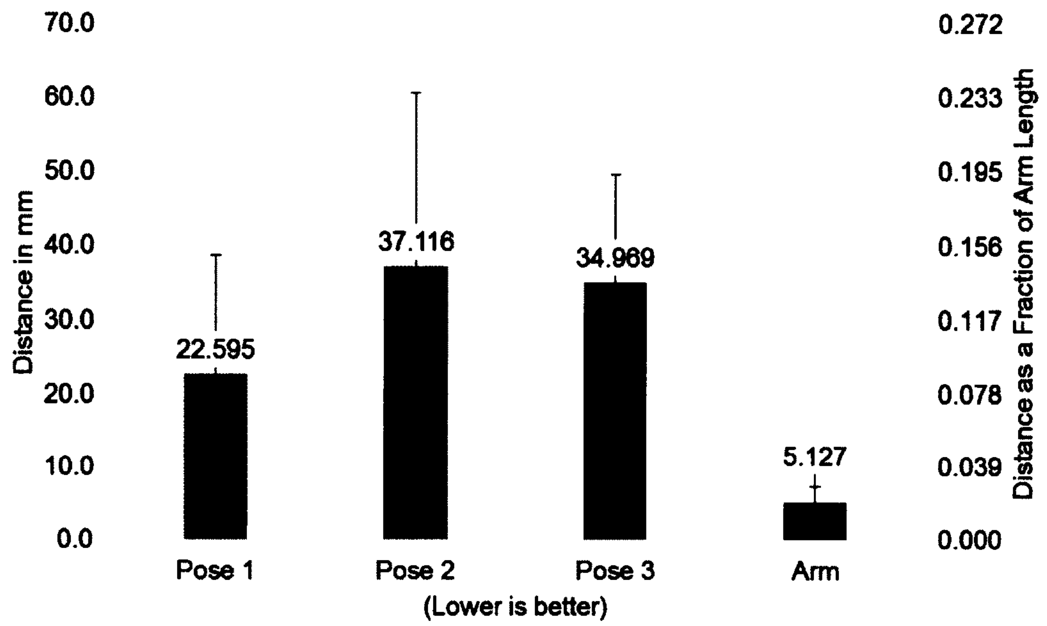


Figure 9.4: Distance between end-effector position predicted by the end-effector model and as tracked in the visual field, as viewed in a mirror placed in three different poses with respect to the robot. Results are reported in millimeters and as a fraction of the length of the robot's arm. "Arm" denotes the robot's ability to perform this task on the arm when witnessed directly in the visual field.

9.1.1.3 Nonlinear refinement

The estimate yielded by the previous step can be refined by minimizing the distance between estimated projections of the robot's end-effector position and their imaged equivalents for m samples, according to Equation 9.13. Optimizations in the presented results use LevMar [67], an implementation of the Levenberg-Marquardt algorithm in C++.

$$f(R,C) = \sum_{i=1}^{m} \| K[\hat{R}| - \hat{R}\hat{C}]J_i - \hat{j}_i \|^2 \tag{9.13}$$

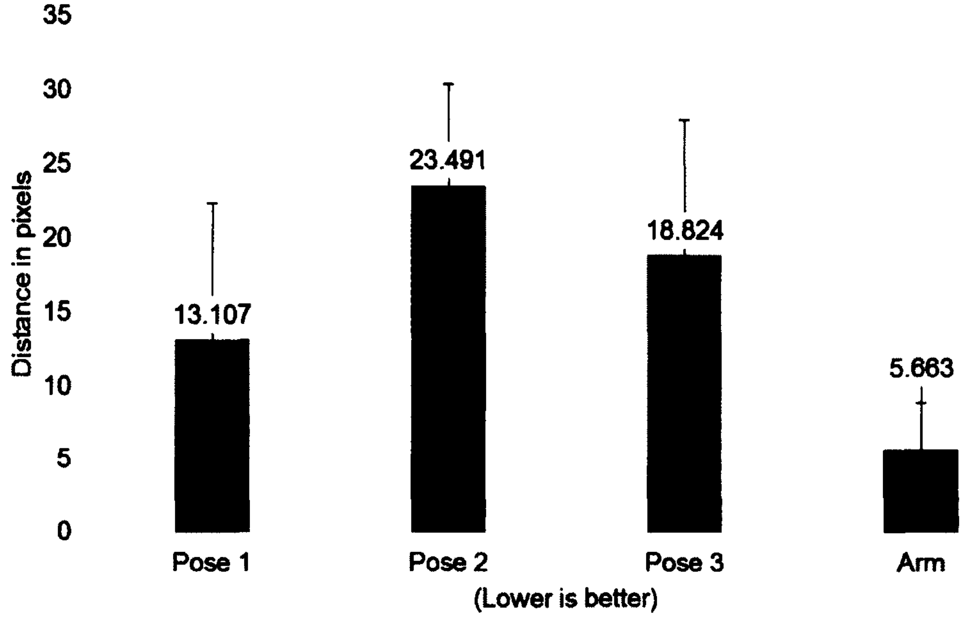


Figure 9.5: Distance between end-effector position predicted by the end-effector model and as tracked in the visual field, as viewed in a mirror placed in three different poses with respect to the robot. Results are reported in pixels. "Arm" denotes the robot's ability to perform this task on the arm when witnessed directly in the visual field.

9.2 Evaluation

The system was implemented and evaluated on the upper-torso humanoid robot, Nico, seen in Figure 9.1. The end-effector and perceptual models were calibrated in the following way. The stereo vision system was first calibrated as described in Appendix B. Known-good intrinsic parameters were then substituted for the estimates yielded by the calibration process, which then re-performed the bundle-adjustment procedure, pinning the intrinsic parameters, in order to derive an accurate estimate of the extrinsic parameters. Known-good radial distortion parameters were used throughout this process. Kinematic parameters that had been estimated in previous experiments were provided to the system to initialize the end-effector model. The arm was then moved into 100 new, unique poses in order to re-calibrate the end-effector and perceptual models to changes in the pose of the eye cameras, zeroing of the robot's encoders, and a new fiducial marker. This larger fiducial marker had to be used because the robot's cameras were not of sufficient resolution[2] to find the one used in previous chapters when reflected in the mirror, due to the larger apparent distance from its cameras.

Testing was performed over datasets containing 150 poses of the robot's arm, 50 of which were used for training, 100 for testing. Three such datasets were sampled with the robot observing its arm in a mirror, which was oriented in a different pose for each dataset, dominating the robot's field of view. Though the system was tested by batch-processing these datasets, for efficiency, no apparent technical barriers exist to developing a version of this system which operates in real-time.

To measure system performance, the mean distance between predictions of end-effector position and measured end-effector position in 3D and 2D are reported. This provides an estimate of how well the mirror-perspective model has been measured with respect to the robot's existing end-effector and perceptual models, though it has the shortcoming that the end-effector will appear more distant in the mirror and, thus, measurements are inherently less accurate. It relates well, however, to the goal of passing the mirror test. The main difference is that the robot makes predictions regarding the position of its end-effector in the mirror based on self-knowledge, rather than predictions regarding its appearance in the mirror. This test is also a form of instrumental mirror use, in that the robot compares predictions of its end-effector position based on its forward-kinematic model, and measured positions based on observations made in the mirror, in its egocentric frame.

[2] Camera resolution is 640 × 480.

9.2.1 Data collection

A mirror was mounted to a moveable, tiltable whiteboard, and placed into the field of view of the robot. Three positions were chosen for the mirror. For each pose a set of 50 training and 100 test samples of with the arm in various poses, imaged as reflected in the mirror, was collected.

9.2.2 Results

As can be seen in Figures 9.4 and 9.5, the system performs well even after training on only 10 arm poses. While the robot is able to predict the position of the end-effector viewed directly in its visual field much better than it is able to in the mirror, its performance is similar to that of other competing systems that do not attempt to perform this task while using a mirror. Recent such systems include ones presented by Hersch, Sauser, and Billard [36] and Martinez-Cantin, Lopes, and Montesano [35], who both report performance to be within 5cm, and attempt neither the task of predicting end-effector position in pixels, nor the task of predicting end-effector position in a mirror. While this is an indicator that the system performs this novel task well for a first attempt, it is important to bear in mind that all three of these system use many different components, including different robotic hardware and different software in almost every stage of the computational process. It is difficult, therefore, to attribute differences in performance to any single component.

Part of the system's degrade in performance when performing this task is due to mechanical imperfections in the robot's hardware. The robot's arm has a "wobble" in it which in the worst case causes it to deviate from its planned position by 15mm. A spring was added to the arm that restricts this deviation quite well when the robot's arm is in the upper portion of its visual field. However, in order to be seen in the mirror, the robot's arm needs to be moved down towards the robot's waist. With more slack in the elastic element, the wobble is more pronounced in these poses. Time did not allow for repairs to be made to the robot's arm, as not all of the components for an upgraded arm arrived in time for publication.

Another factor contributing to this error when the robot witnesses its arm in the mirror can be attributed to the apparent distance of the end-effector when viewed in the mirror. The apparent distance of the object, when viewed in the mirror, combines the distance of the robot from the mirror and the distance of the object from the mirror. As a result, the view of the object is much farther away. The range from the left camera of the stereo vision system to the mirror in each of the three tests was 106.14mm, 144.29mm, and 107.45mm, respectively. In tests in which the arm is witnessed directly by the vision system, the maximum distance between the vision system and the arm is the length of the robot's arm plus the distance between the robot's shoulder and head. In the case of the mirror, it is more difficult to directly evaluate this, because the maximum distance between the robot's arm and the vision system is determined by the angle of the mirror. Also, the robot mostly moved its arm such that it was behind the torso, in order to avoid bumping into the mirror. This adjustment was made by hand, as, in early trials of the system, the random pose generator frequently hit the mirror, knocking off the fiducial marker.

Reconstructions of the tracked point under this regime are subject to a higher degree of error due to this, leading to a greater degree of disagreement, as the same area of visual angle contains a greater physical area. This is consistent with the fact that performance in pixels is more similar between the arm in the visual field and the arm in the mirror, than performance in millimeters. Because the mirror-perspective cameras are optimized independently from each other, it is possible for the system to estimate positions and orientations for these cameras which changes their position and orientation with respect to each other. This also contributes to error. By optimizing the position of the mirror, and computing the mirror-perspective cameras from this position, we should be able to improve performance. This is saved for future work.

9.3 Summary and conclusions

In this chapter, we have developed a model that allows the robot to determine a set of camera calibration parameters that is consistent with the visual perspective that it witnesses when observing reflections in a mirror. To do so, it uses self-knowledge about its body and senses in the form of kinematic and visual calibration information. The procedure described in this chapter involves starting with a calibrated kinesthetic-visual self model as developed in the previous chapters of this thesis. This serves as the starting point from which we are able to construct the system which is able to infer the visual perspective of the mirror. In order to do so, the robot observes the position of its end-effector with its arm in multiple poses. An algorithm is developed which allows the robot to infer the plane of the mirror based on the symmetry between the position of the end-effector predicted by the robot's self model, and the reconstruction of that end-effector position, as reflected in the mirror. From this plane, an estimate of a virtual camera is computed. This virtual camera represents the visual perspective that the robot's physical camera sees as it looks into the mirror, and is derived from the reflections of the extrinsic properties of the real camera, with respect to the mirror. This estimate is refined based on a simple extension of the self-calibration techniques developed earlier in this thesis.

To our knowledge, this is the first robotic system to attempt to use a mirror in this way, representing a significant step towards a cohesive architecture that allows robots to learn about their bodies and appearances through self-observation, and an important capability required in order to pass the mirror test.

Chapter 10

Toward self-aware robotic systems

The goal of this thesis has been to take a practical engineering approach to understanding problems related to primitive forms of self-awareness that can be designed into robotic systems. At the start of this dissertation we chose as our goal to construct a robot which learns about its body and senses through experience. We then put this goal into context by showing how it is a subgoal towards the construction of a system which passes a well-known test of self-awareness, the mirror task. This chapter will summarize the accomplishments of this thesis, re-evaluate their context with respect to the mirror-task, and discuss possible applications and extensions of this work.

10.1 Summary

In this thesis, a system was developed which emulates one of the earliest forms of self-awareness to develop during infancy, knowledge of the body and senses. We introduced the term self-model, to describe the unified model of the body and the senses that the robot develops, and self-modeling, the process by which a machine learns about its body and senses through data sampled during operation. The robot constructed such a self-model describing its kinematics and stereo vision system and their relationship to each other.

This self-model differs from many models of kinematics and vision in that the kinematic and visual components of the models are mutually calibrated to each other, allowing accurate computations to be performed which combine the two. In this sense the mutually calibrated models constitute a body schema by which the robot's visual sense can be interpreted with respect to the current kinesthetic pose of the frame of the body.

We demonstrated the ability of the system to adapt in an online fashion in order to incorporate a tool into the robot's kinematic chain. We also demonstrated the capability of the system to use the robot's self-knowledge in order to model an object in its environment, a mirror. The self-model enables the robot to use its body as a calibration target for its stereo vision system, simultaneously refining its stereo vision calibration and its kinematic calibration. In the case of the mirror, the robot again uses its body as a visual calibration target, this time to infer the visual perspective of the mirror and constructing a virtual camera that represents this perspective.

The system presented in this thesis is motivated by the developmental process experienced by infants, but is not based on biology. Instead, models typical of modern robotics design have been updated, incorporated, and enhanced in order to construct this system. This system could be incorporated into modern robots, providing them with useful capabilities. Constructing such systems may provide not only practical solutions to present engineering challenges, but also help us to tackle the problem of clarifying discussion of what we mean when we refer to self-awareness.

10.2 An Architecture for Mirror Self-Recognition

At the start of this thesis a six phase plan for the construction of an architecture to allow a humanoid robot to pass the mirror-test [16] was introduced. The mirror test was picked as a milestone to work towards in the development of self-awareness in robots and artificially intelligent systems because it provides a recognizable goal and because it encompasses components that are useful and meaningful to modern robotic systems. To pass the mirror test, the robot must learn about its body and senses and their interactions. This knowledge must capture the appearance of the robot, allowing it to identify a change to its appearance. It also must be thorough enough to allow the robot to inspect a mark placed on its body. Finally, it must allow the robot to identify that it is still looking at its body as it moves into multiple poses.

The components of the planned self-model for mirror self-recognition are:

The perceptual model The perceptual model captures data regarding the robot's sensors. Importantly, it places the robot's sensors in context on its physical frame, allowing sensor data to be interpreted with respect to the robot's current kinematic pose. In this sense, it is related to the idea of the body schema [4, 33]. The perceptual model also ideally would be learned through data sampled during operation.

The end-effector model Enabling the robot to track the motion of its body through space, the end-effector model is a model of the robot's kinematics. Importantly, the robot is able to infer the end-effector model through data sampled during operation.

The perspective-taking model Enabling the robot to understand and interpret other visual perspectives, the perspective-taking model could have many uses. Taking the visual perspective of other agents in an interaction could allow for social forms of self-awareness to be explored. In this dissertation, the robot infers the visual perspective of a mirror by witnessing its own motion therein.

The structural model The structural model is intended to allow the robot to infer the 3D structure of its body. This physical model could have many potential uses, such as determining the pose of a contact patch between the robot's end-effector and objects that it is attempting to manipulate.

The appearance model The appearance model is intended to add information to the structural model regarding the appearance of the robot. To the 3D structure, visual information such as the coloration along the surface of the body will be added. This addition of the appearance model will hopefully allow the robot to visually identify itself without the use of external markers and provide the means for developing an expectation of the robot's current appearance.

The functional model The functional model should describe how the robot is able to interact with its environment. We currently plan for it to be a causal model between the robot's actions and their outcomes.

10.2.1 The end-effector and perceptual models

The rich model of the self outlined in this thesis is, at its core, an enhancement of modern robotic modeling techniques. In modern practice, it is common for the robot's kinematic and visual representations to come from two different sources, but such models are often both present in complicated robots which use computer vision as part of the process of performing manipulations. Mutually calibrating the two, as in the first phases of the construction of the outlined architecture, is an enabling technology. In Chapter 8, we saw how treating the two models in a unified fashion could enable a robot to self-calibrate. Importantly, the unified model allows the robot to make highly accurate measurements with its visual system that agree with its kinematic predictions.

10.2.2 The perspective-taking model

The perspective-taking model is really an enhancement of the perceptual model. It enables the robot to interpret its sensor data from a non-native perspective. In a social context, this could mean taking the perspective of another agent in an interaction. In this thesis, a version of this model is presented that enables the robot to infer the visual perspective representing the transformation of 3D scene points into the image witnessed by the visual system as reflections in the mirror. The robot uses knowledge of itself, in the form of its end-effector and perceptual models, in order to infer this perspective transformation.

10.2.3 The mirror as an instrument for spatial reasoning

When we outlined the mirror test [16] as a milestone in the development of self-aware systems, we put it into context with the development of other self-awareness related capabilities. One that appears earlier in infant development is the ability to use a mirror as an instrument for spatial reasoning. In Chapter 9, we demonstrated the ability of the robot to do just this. By inferring the visual perspective of the mirror, the robot is able to accurately determine the position of objects in the environment with respect to its body. We demonstrated this capability by predicting the position of the robot's end-effector as an image reflected in the mirror. The robot also reconstructed its end-effector position from its reflections in the mirror, agreeing within centimeters of the position predicted by the end-effector model.

10.3 Future work and potential applications

This thesis lays out a framework for the construction of robot self-models. Significant prior work exists in a number of areas of artificial intelligence and robotics dealing with concepts relating to the self. The self is a natural part of collaboration and communication. Infants learn about the self when learning to perform manipulations. Fault detection, diagnosis, and recovery is a form of reasoning about the self as a physical entity. By developing rich self-models, we may be able to develop a unified framework for understanding these topics and utilize this framework to develop powerful solutions to practical engineering problems.

10.3.1 The structural and appearance models

Left to future work in the architecture laid out in Chapter 2 are the structural and appearance models. These models are intended to provide the robot with information regarding its 3D structure and appearance information regarding the surface of that structure. The idea behind these models is to enable the robot to compute a prediction of its appearance. Changes in the robot's appearance would cause this prediction to be violated, leading to investigation of the mark.

The plan is to create a markerless tracker based on vision-based simultaneous localization and mapping (SLAM) [85] techniques. There is significant current interest in the use of natural features for feature tracking (Lee and Höllerer provide a good overview [86], but devices such as the Microsoft Kinect$^{\mathrm{TM}}$, the Leap Motion$^{\mathrm{TM}}$tracker, and the Organic Motion$^{\mathrm{TM}}$motion tracker all demonstrate the consumer interest in this area), features that occur along the robot's frame without the necessity for explicitly marking the frame. For instance, the system presented in this dissertation requires a marker to be placed on the robot's end-effector in order to track its motion through space. Such techniques would enable robots to identify and track objects without explicit instrumentation.

Not appearing in this dissertation is current work on the construction of such a markerless tracking system. This work builds on previous work in the Social Robotics Lab on self-other discrimination [28, 30], allowing the robot to identify portions of its visual field that change based on the robot's ego-motion. Instead of discerning the motion of pixels in the 2D images in the robot's cameras, the robot witnesses changes in grid occupancy of 3D point cloud data. In this incomplete system, the robot constructs a 3D model objects by observing them as they move through the robot's visual field by registering previously viewed 3D geometry to current reconstructions as objects move through space.

10.3.2 The functional model

Michel, Gold and Scassellati [28] and Gold and Scassellati [30] constructed systems that allow for a robot to identify the pixels constructing itself in its visual field based on its ego-motion. While these models are able to identify said 2D pixels in the visual field, they do not directly extend to the 3D context of reconstructed points moving through space. They perform this task by modeling the causal relationship between the robot's motor actions and changes it witnesses in its cameras.

The first goal of the functional model is to extend that work to the domain of 3D perception on the part of the robot. One approach to this would be to use the existing systems [28, 30] and then integrate self-other data into reconstructed point clouds. Another approach would be to perform a task based on these methods, but to perform it over reconstructed point cloud data. If successful, this would enable the system to perform a self-recognition task in order to identify point cloud data which belong to parts of its body as they move. These could then be segmented into body segments, enabling the robot to identify its limbs from their motion, then integrate this into a self-model of the body.

The functional model is intended eventually to capture a more general capability than this, however. Experiments have provided a wealth of evidence that neonates and infants are capable of reacting to stimuli as though they are able to reason about causal relationships [87, 88, 89, 90], and it has been theorized that this is a fundamental skill from which other skills, such as motor skills are learned [3]. The eventual goal of the functional model is to capture the causal relationships between all of the robot's actions and its environment as well as how objects interact with each other, providing a groundwork for learning motor skills and naïve physics.

10.3.3 Self-calibrating fault-detecting robots and machinery

As more complex robots are deployed to a greater variety of tasks it would be preferable for them to be able to self-calibrate and self-diagnose faults. Consider the possibility of a precision manufacturing robot with self-modeling capabilities. In a modern factory, such a robot may need regular maintenance and calibration from expert personnel. Through extensions of self-modeling techniques, however, it may be possible to compensate for changes to the physical machine itself in software. Such a system may also be able to identify faults that must be corrected by maintenance personnel. In a factory setting, this could improve the quality of the factory's output (as the machine is always well-calibrated) or enable more precise manufacturing techniques that are currently not economically feasible.

In many domains, self-modeling may make more complicated robots feasible. Because we cannot rely on all users to be experts in robotics, it is unreasonable to expect them to be able to correct faults in complex robotic devices. The owner of an autonomous vehicle may not be able to identify that a sensor is faulty due to environmental damage, such as damage from being hit by a hailstone. Such damage may also be sustained during operation, requiring immediate attention in order for the system to operate safely. In a more familiar scenario, a car that autonomously identifies that it has popped a tire may be able to assist the driver in bringing the vehicle to a halt in a safe location before requesting assistance from a road service. Additionally, as commercial robots become capable of complex manipulations and behaviors that are currently a challenge in laboratory environments, we will not be able to rely on home users to perform the tweaks and calibrations that field experts perform. As we put more complicated robots into outer space, they will become more susceptible to damage during their missions. In this scenario, the experts may be able to instruct the robot from the ground, but will not be capable of physically accessing a robot in order to perform a repair. These robots will need to perform these tasks for themselves.

10.3.4 Tactile Sensing

Recently, the development of inexpensive, flexible tactile sensors has increased interest in tactile sensing in the robotics community. The work in this dissertation has concerned itself with combining the robot's kinematic and visual models, enabling them to be used in concert with each other. When a person uses their hands to feel for the positions of objects in space, they perform a similar reasoning task. Another interesting direction in which to take this work would be to integrate tactile sensors into the self-model. Work related to this idea has been performed by Fuke, Ogino, and Asada [91].

10.3.5 Understanding Others by Reflecting on the Self

A domain of self-reasoning that has not been extensively discussed in this thesis is reasoning about the self as a social actor. Theory of mind tasks such as reasoning about ones own mental states and making attributions of mental state to other agents in an interaction form the groundwork for much of social reasoning in humans, and could provide a framework for such social reasoning in robots [39]. Reasoning about other agents by reflecting both on the self and what is known about the agent through an extension of the self-model could provide a means to this end.

One such social reasoning task is the Sally-Anne task [92]. In this task, a narrative similar to the following is used. Sally likes an object, which she places into a box. Sally then leaves the room, and her friend Anne hides the object from her by moving it from one box to another box. When Sally returns, in which box will she look for the object? Correctly answering this question requires theory of mind concepts regarding the knowledge of the disposition of the object.

Previous work has attempted to perform similar tasks by incorporating a model of the attentional states of multiple agents in an interaction [38]. In such an approach, one attempts to model the knowledge obtained through visual attention, then make inferences about the mental states of other agents in an interaction. One potential use of combining eye-tracking or other visual attention related data with the perspective-taking model is to model the visual perspective of other agents in the interaction in order to perform such social reasoning. From here, we could explore several possible branches of research such as the emergence of joint attention, theory of mind, and the mind reading model of communication.

Another possible, related avenue of research is to use rich models of a other agent's mental and physical states in tasks such as learning by demonstration. The typical mode for learning by demonstration is to model an agent's physical state and to map it onto the body of the robot [93]. This provides a set of primitives enabling the robot to learn to perform a task. Incorporating factors such as visual attention and other components of a theory of mind based knowledge base could allow us to explore new avenues of research with respect to learning by demonstration, such as modeling the goals, beliefs, and intents of other agents in an interaction, in order to take a goal-directed approach to social learning tasks.

There is also a wealth of interesting work that could be done in the domain of collaborative planning and reasoning, using self-other reasoning as a basis for planning. Whereas in standard planning domains the planner is able to account for the actions of all of the agents acting on a plan, in human-robot domains planners acting on the behalf of the robot are unable to directly control a human's actions. Moreover, multi-agent reasoning can involve adversarial reasoning about the actions of an opponent, as in a game or competition like robot soccer. In such domains, reasoning about the capabilities and intentions of other agents may help in order to devise an optimal plan. In cooperative domains, such as collaborative manufacturing, such planning techniques could be used to explore factors such as communication and coordination of efforts towards the completion of a task.

10.4 Closing Thoughts

This research started with the question, "What if robots learned about themselves, rather than about tasks?" The system developed in this thesis is intended to lay the groundwork for answering the many forms that that question can take. By posing the question of self-awareness in its simplest form, we have attempted to provide a framework that provides meaningful functionality and a firm basis from which to move forward in the study of this important question that is agnostic to any as of yet unproven or philosophical ideas with regards to self-awareness.

In this thesis, a robot has constructed a simple self-model through data sampled during operation. That self-model is demonstrated to provide highly accurate data regarding the robot's kinematic structure and sensing. It has been demonstrated to be able to be adapted, as in the case of tool use, and to provide unique self-reflexive capabilities, in the form of reasoning about a mirror in the robot's environment. The constructed system is an attempt to turn classical robot learning and reasoning tasks on their head, learning about the robot in its environment rather than the environment that the robot is in. By starting with problems related to modeling primitive forms of self-awareness that develop during infancy, it is my hope that the work in this thesis will contribute to the study of more complicated forms of self-awareness and ultimately towards the understanding of self-awareness in artificially-intelligent robotic systems.

Appendices

Appendix A

Circle fitting implementation

The implemented method for fitting circles roughly follows the methods described in [69]. The rotation from the estimated best-fit plane to the XY plane can be normalized by decomposing its matrix via Singular Value Decomposition (SVD), Equation A.1, then recomposing such that the singular values are all one, Equation A.2. The variable $\hat{R}$ is an approximation of this rotation produced by the formula from [69], and $\tilde{R}$ is the normalized approximation of the optimal rotation matrix. The variables U, Σ, and V are products of the SVD. The matrix σ contains the singular values along its diagonal. Removing this matrix in Equation A.2 normalizes the rotation matrix such that applying its transformation does not rescale transformed points along any axis in the new basis.

$$\hat{R} = U\Sigma V^T \tag{A.1}$$

$$\tilde{R} = UV^T \tag{A.2}$$

$$B = \begin{bmatrix} x & y & z & w \end{bmatrix}^T = (x : w, y : w, z : w) \tag{A.3}$$

Algorithm 2 Circle fitting

```
Determine the best fit plane to the sampled points, by Equation A.3.
Rotate points into XY plane, as according to [69], but normalize the rotation matrix such that the rotation does not scale points in any direction.
Compute a naïve estimate of the center and radius of the 2D circle lying in this plane. This is computed as the mean $(x, y)$ coordinates of the collection of points and mean offset of those points from that centroid.
Minimize 2D circle fit objective function as in [69].
Invert estimated parameters into original coordinate frame.
Minimize 3D circle fit objective function as in [69].
return The estimated circle.
```

Optimization was performed in Wolfram Mathematica 9.0.1.0 [60] using the Principal Axis Method [94]. It was found that performance improved if the system was first optimized with the radius pinned, then the entire system was optimized.

Appendix B

Implementation of Nico's camera calibration system

Nico's camera calibration system has implementations of Zhang's method [62], the normalized eight-point algorithm [78], and bundle adjustment [82]. Some systems obtain estimates of the camera extrinsic parameters by computing them from chessboard position estimates from Zhang's method. Instead, this implementation estimates the extrinsic parameters using the Normalized Eight Point Algorithm. This is followed up by a modification of the minimization performed in Zhang's method in order to simultaneously refine the calibration of the left and right cameras. This assures that their calibrations agree with each other. In the final step, the system performs bundle adjustment in order to assure the best possible calibration. The system also allows certain parameters to be pinned to known-good values. The system makes multiple passes through the final bundle adjustment optimization, pinning parameters such as γ, u_0, and v_0 to their nominal values $0, 320$ and 240, respectively. This assures a very well-calibrated system. The entire method is detailed in Algorithm 3.

Algorithm 3 Calibration method for computing the robot's initial camera calibration.

```
Image chessboard calibration targets as in the case of Zhang's method.
Compute estimates of camera intrinsic parameters for the left and right camera and chessboard calibration target positions with respect to the left camera by Zhang's method.
Complete optimization for both left and right cameras via Zhang's method.
Estimate epipolar geometry using the Normalized Eight Point Algorithm, using the chessboard calibration images from the previous steps.
Compute the essential matrix, using camera intrinsic estimates found through Zhang's method.
Compute estimates of $R$ and $t$, the rotation and translation of the right camera with respect to the position of the left camera from the essential matrix.
Estimates of $R$ and $t$ are subject to the four-fold ambiguity discussed in [55]. Fix estimates of $R$ and $t$, if necessary.
repeat
    Minimize squared distance between projected 2D images of model chessboard corners and imaged chessboard corners from the set of calibration images. Optimize over $R$, $t$, the position and orientation of chessboards, and lens distortion parameters $k_1, k_2, k_3$. Perform this optimization simultaneously for both cameras. This step is an enhancement of the minimization performed in Zhang's method.
    Compute reconstructions of 3D positions of chessboard corners, then reproject them to 2D. Minimize squared distance between imaged 2D chessboard corner positions and their reprojections. This is a version of bundle adjustment, optimizing reprojection error [55].
until Summed squared reprojection error converges to a small value.
```

Implementation for the normalized eight-point algorithm is written in Wolfram Mathematica [60]. Two implementations of Zhang's algorithm are written, one in Mathematica which is used to obtain a highly-precise estimate of the camera intrinsic matrices, one in C++ which runs much faster. The Mathematica implementation of Zhang's method uses **FindMinimum[]**, for optimization. The optimizations written in C++ all use LevMar [67], an implementation of the Levenberg-Marquardt non-linear optimizer [74, 75]. Features such as pinned variables are handled via a custom C++ library which manages the parameters and values passed to LevMar.

Bibliography

[1] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Transactions of the ASME Journal of Applied Mechanics*, vol. 23, pp. 215–221, 1955.

[2] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality.* Washington, DC, USA: IEEE Computer Society, 1999, pp. 85–94.

[3] P. Rochat, *The Infant's World.* Cambridge, Massachusetts and London, England: Harvard University Press, 2001.

[4] H. Head, *Studies in Neurology.* London, England: Oxford University Press, 1920, vol. 2.

[5] P. Rochat and S. J. Hespos, "Differential rooting response by neonates: Evidence for an early sense of self," *Early Development and Parenting*, vol. 6, no. 3–4, pp. 105–112, 1997.

[6] P. Rochat, E. M. Blass, and L. B. Hoffmeyer, "Oropharyngeal control of hand-mouth coordination in newborn infants," *Developmental Psychology*, vol. 24, no. 4, pp. 459–463, 1988.

[7] S. Yamamoto, S. Moizumi, and S. Kitazawa, "Referral of tactile sensation to the tips of L-shaped sticks," *Journal of Neurophysiology*, vol. 93, pp. 2865–2863, 2005.

[8] M. Botvinick and J. Cohen, "Rubber hands 'feel' touch that eyes see," *Nature*, vol. 391, no. 6669, p. 756, 1998.

[9] G. M. Stratton, "Some preliminary experiments on vision without inversion of the retinal image," *Psychological Review*, vol. 3, no. 6, pp. 611–617, 1896.

[10] C. S. Harris, "Adaptation to displaced vision: Visual, motor, or proprioceptive change?" *Science*, vol. 140, no. 3568, pp. 812–813, 1963.

[11] S. M. Morton and A. J. Bastian, "Prism adaptation during walking generalizes to reaching and requires the cerebellum," *Journal of Neurophysiology*, vol. 92, no. 4, pp. 2497–2509, October 2004.

[12] I. Kohler, *The Formation and Transformation of the Perceptual World.* New York, USA: International Universities Press, 1964.

[13] D. Linden, U. Kallenbach, A. Heinecke, W. Singer, and R. Goebel, "The myth of upright vision. a psychophysical and functional imaging study of adaptation to inverting spectacles," *Perception*, vol. 28, 1999.

[14] R. Volcic, C. Fantoni, C. Caudek, J. A. Assad, and F. Domini, "Visuomotor adaptation changes stereoscopic depth perception and tactile discrimination," *The Journal of Neuroscience*, vol. 33, no. 43, pp. 17081–17088, October 2013.

[15] B. I. Bertenthal and K. W. Fischer, "Development of self-recognition in the infant," *Developmental Psychology*, vol. 14, no. 4, pp. 44–50, 1978.

[16] G. G. Gallup, "Chimpanzees: Self-recognition," *Science*, vol. 167, no. 3914, pp. 86–87, 1970.

[17] D. Povinelli, "Monkeys, apes, mirrors and minds: The evolution of self-awareness in primates," *Human Evolution*, vol. 2, no. 6, pp. 493–509, 1987.

[18] B. Amsterdam, "Mirror self-image reactions before age two," *Developmental Psychobiology*, vol. 5, pp. 297–305, 1972.

[19] G. Boulanger-Balleyguier, "Premières réactions devant le mimoir," *Enfance*, vol. 1, pp. 51–67.

[20] J. Brooks-Gunn and M. Lewis, "Mirror-image simulation and self-recognition in infancy," Denver, Colorado, USA, April 1975.

[21] C. R. Darwin, "A biographical sketch of an infant," *Mind*, vol. 2, pp. 285–294.

[22] J. C. Dixon, "Development of self-recognition," *Journal of Genetic Psychology*, vol. 91, pp. 251–256.

[23] R. Zazzo, "Des jumeaux devant le mimoir: Questions de méthode," *Journal de Psychologie*, vol. 4, pp. 389–413.

[24] A. Heschl and J. Burkart, "A new mark test for self-recognition in non-human primates," *Primates*, vol. 47, no. 3, pp. 187–198, 2006.

[25] E. J. Menzel, E. Savage-Rumbaugh, and J. Lawson, "Chimpanzees (pan troglodytes) spatial problem solving with the use of mirrors and televised equivalents of mirrors," *Journal of Comparative Psychology*, vol. 99, no. 2, pp. 211–217, 1985.

[26] M. T. Cox and A. Raja, Eds., *Metareasoning: Thinking about Thinking.* Cambridge, Massachusetts, USA: The MIT Press, 2011.

[27] S. Baron-Cohen, *Mindblindness.* Cambridge, Massachussetts, USA: The MIT Press, 1995.

[28] P. Michel, K. Gold, and B. Scassellati, "Robotic self-recognition," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Japan, September 2004.

[29] J. Takeno, K. Inaba, and T. Suzuki, "Experiments and examination of mirror image cognition using a small robot," in *Proceedings of the 6th International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, Espoo, Finland, June 2005.

[30] K. Gold and B. Scassellati, "A bayesian robot that distinguishes "self" from "other"," in *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, Nashville, Tennessee, USA, August 2007.

[31] E. Ackerman, "Qbo robot passes mirror test, is therefore self-aware," Retrieved from http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/qbo-passes-mirror-test-is-therefore-selfawar, 2011.

[32] Y. Yoshikawa, K. Hosoda, and M. Asada, "Unique association between self-occlusion and double-touching towards binding vision and touch," *Neurocomputing*, vol. 70, pp. 2234–2244, August 2007.

[33] M. Hoffmann, H. Marques, A. H. Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, "Body schema in robotics: a review," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 4, pp. 304–324, December 2010.

[34] J. Sturm, C. Plagemann, and W. Burgard, "Body schema learning for robotic manipulators from visual self-perception," *Journal of Physiology-Paris*, vol. 103, no. 3–5, pp. 220–231, September 2009.

[35] R. Martinez-Cantin, M. Lopes, and L. Montesano, "Body schema acquisition through active learning," Alaska, USA, 2010.

[36] M. Hersch, E. Sauser, and A. Billard, "Online learning of the body schema," *International Journal of Humanoid Robotics*, vol. 5, no. 2, pp. 161–181, 2008.

[37] B. Scassellati, "Theory of mind for a humanoid robot," *Autonomous Robots*, vol. 12, pp. 13–24, 2002.

[38] M. Berlin, J. Gray, A. L. Thomaz, and C. Breazeal, "Perspective taking: An organizing principle for learning in human-robot interaction," in *Proceedings of the 21st AAAI Conference on Artificial Intelligence (AAAI)*, Boston, Massachusetts, USA, July 2006.

[39] C. Breazeal, J. Gray, and M. Berlin, "An embodied cognition approach to mindreading skills for socially intelligent robots," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 656–680, 2009.

[40] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach," in *Proceedings of the 12th International Symposium on Experimental Robotics (ISER)*, New Delhi, India, December 2010.

[41] A. Darwiche, "Model-based diagnosis under real-world constraints," *AI Magazine*, vol. 21, pp. 57–73, 2000.

[42] J. de Kleer and B. C. Williams, "Diagnosing multiple faults," *Artificial Intelligence*, vol. 32, no. 1, pp. 97–130, 1987.

[43] J. de Kleer and B. C. Williams, "Diagnosis with behavioral modes," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI)*, Detroit Michigan, USA, August 1989.

[44] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta, "Methods and technologies for the implementation of large-scale robot tactile sensors," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 389–400, June 2011.

[45] A. Tilley and H. Associates, *The Measure of Man and Woman: Human Factors in Design.* Wiley, 2002.

[46] G. J. Gallup, J. Anderson, and D. Shillito, "Dynamic ambient paradigms," in *Paradigm Gems 2*, A. Doe, Ed. Addison Wesley, 2005, pp. 223–233.

[47] J. Anderson and G. J. Gallup, "Which primates recognize themselves in mirrors?" *PloS Biology*, vol. 9, no. 3, 2011.

[48] J. Plotnik, F. de Waal, and D. Riess, "Self-recognition in an asian elephant," *Proceedings of the National Academy of Science*, vol. 103, no. 45, pp. 17053–17057, 2006.

[49] D. Reiss and L. Marino, "Mirror self-recognition in the bottlenose dolphin: A case of cognitive convergence," *Proceedings of the National Academy of Science*, vol. 98, no. 10, pp. 5937–5942, 2001.

[50] F. Delfour and K. Marten, "Mirror image processing in three marine mammal species: killer whales (orcinus orca), false killer whales (pseudorca crassidens)

and california sea lions (zalophus californianus)," *Behavioral Processes*, vol. 53, no. 3, pp. 181–190, April 2001.

[51] O. G. Helmut Prior, Ariane Schwarz, "Mirror-induced behavior in the magpie (pica pica): evidence of self-recognition," *PLoS biology*, vol. 6, no. 8, August 2008.

[52] G. G. Gallup, "Self-awareness and the emergence of mind in primates," *American Journal of Primatology*, vol. 2, pp. 237–248, 1982.

[53] R. Epstein, R. P. Lanza, and B. Skinner, ""self-awareness" in the pigeon," *Science*, vol. 212, no. 4495, pp. 695–696, May 1981.

[54] R. W. Mitchell, "Kinesthetic-visual matching and the self-concept as explanations of mirror-self-recognition," *Journal for the Theory of Social Behavior*, vol. 27, no. 1.

[55] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[56] J. Hart and B. Scassellati, "A robotic model of the ecological self," in *Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, Bled, Slovenia, October 2011.

[57] A. Stoytchev, "Toward video-guided robot behaviors," in *Proceedings of 7th International Conference on Epigenetic Robotics*, 2007, pp. 165–172.

[58] D. Papalia and S. Wendkos-Olds, *Human Development*. New York, USA: McGraw-Hill, 1989.

[59] C. G. Limited, "Ubuntu," [web page] `http://www.ubuntu.com/`.

[60] Wolfram Research, Inc., *Mathematica*, version 9.0.1.0 ed. Champaign, Illinois: Wolfram Research, Inc., 2013.

[61] A. Möbius, *Der barycentrische Calcul*, 1827.

[62] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 2000.

[63] G. Bradski, "The opencv library," *Dr. Dobb's Journal of Software Tools*, 2000.

[64] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, pp. 323–344, 1987.

[65] D. C. Brown, "Close-range camera calibration," pp. 855–866.

[66] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980.

[67] M. Lourakis, "Levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++," [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, July 2004.

[68] J. Hollerbach and C. Wampler, "The calibration index and taxonomy for robot kinematic calibration methods," *International Journal of Robotics Research*, vol. 14, no. 573–591, 1996.

[69] C. M. Shakarji, "Least-squares fitting algorithms of the nist algorithm testing system," *Journal of Research of the National Institute of Standards and Technology*, vol. 103, no. 6, pp. 633–641, 1998.

[70] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms," *Journal of the Institute of Mathematics and Its Applications*, vol. 6, pp. 222–231, 1970.

[71] R. Fletcher, "A new approach to variable metric algorithms," *The Computer Journal*, vol. 13, no. 3, pp. 317–322, 1970.

[72] D. Goldfarb, "A family of variable-metric methods derived by variational means," *Mathematics of Computation*, vol. 24, no. 109, pp. 23–26, January 1970.

[73] D. F. Shanno and P. C. Kettler, "Optimal conditioning of quasi-newton methods," *Mathematics of Computation*, vol. 24, no. 111, pp. 657–664, July 1970.

[74] K. Levenberg, "A method for the solution of certain non-linear problems in least-squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[75] D. W. Marquardt, "An algorithm for the least-squares estimation of nonlinear parameters," *SIAM Journal of Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[76] Y. Yoshikawa, Y. Tsuji, K. Hosada, and M. Asada, "Is it my body? - body extraction from uninterpreted sensory data based on the invariance of multiple sensory attributes," in *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 2325–2330.

[77] C. C. Liñán, "cvBlob," http://cvblob.googlecode.com. [Online]. Available: http://cvblob.googlecode.com

[78] R. I. Hartley, “In defense of the eight-point algorithm,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, 1997.

[79] R. Tsai and R. Lenz, “Real time versatile robotics hand/eye calibration using 3d machine vision,” in *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, vol. 1, April 1988, pp. 554–561.

[80] Z. Zhao and Y. Liu, “Integrating camera calibration and hand-eye calibration into robot vision,” in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA)*, June 2008, pp. 5721–5727.

[81] M. Dinham and G. Fang, “Low cost simultaneous calibration of a stereo vision system and a welding robot,” in *Proceeedings of the 2010 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, December 2010, pp. 1452–1456.

[82] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, *Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms, held during ICCV '99, Corfu, Greece, September 21-22, 1999, Proceedings*, ser. Lecture Notes in Computer Science. Springer, 2000, vol. 1883, ch. Bundle Adjustment – A Modern Synthesis.

[83] Q. tuan Luong, R. Deriche, O. Faugeras, and T. Papadopoulo, “On determining the fundamental matrix: Analysis of different methods and experimental results,” INRIA, Sophia-Antipolis, France, Tech. Rep. Technical Report RR-2927, April 1993.

[84] Z. Zhang, “Determining the epipolar geometry and its uncertainty: A review,” *International Journal of Computer Vision*, vol. 27, no. 2, pp. 161–192, 1998.

[85] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent robotics and autonomous agents series. Mit Press, 2005.

[86] T. Lee and T. Höllerer, "Multithreaded hybrid feature tracking for markerless augmented reality." *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 3, pp. 355–368, 2009.

[87] H. Papoušek, "Experimental studies of appetitional behavior in human newborns and infants," *Advances in Infancy Research*, vol. 7, pp. xix–liii, 1992.

[88] P. Rochat and S. J. Senders, "Active touch in infancy: Action systems in development," in *Newborn attention: Biological constraints and influence of experience*, 2nd ed., M. Weiss and P. Zelano, Eds. Norwood, NJ, USA: Ablex, 1991, pp. 412–442.

[89] C. Rovee-Collier, "Learning and memory in infancy," in *Handbook of Infant Development*, 2nd ed., J. Osofsky, Ed. New York, NY, USA: John Wiley and Sons, 2005, pp. 98–148.

[90] G. Walton, N. Bower, and T. Bower, "Recognition of familiar faces by newborns," *Infant Behavior and Development*,.vol. 15, no. 2, pp. 265–269, 1992.

[91] S. Fuke, M. Ogino, and M. Asada, "Body image constructed from motor and tactile images with visual information," *International Journal of Humanoid Robotics*, vol. 4, no. 2, pp. 347–364, 2007.

[92] S. Baron-Cohen, A. M. Leslie, and U. Frith, "Does the autistic child have a "theory of mind"?" *Cognition*, vol. 21, pp. 37–46, 1985.

[93] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 51, pp. 469–483, 2009.

[94] R. Brent, *Algorithms for Minimization without Derivatives.* Dover, 2002.