Object Recognition Using Tactile Array Sensors

by

Zachary Pezzementi

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

May, 2011

Abstract

In this dissertation, we explore the use of tactile force sensors to understand properties of the surfaces of objects they are pressed against. We develop a model of such sensors as *imaging devices*, which facilitates the use of techniques from computer vision and image processing with the "tactile images" they provide. The goal is object recognition, and three approaches are presented for distinguishing amongst objects from a previously-encountered set: The first approach is entirely geometric in nature and borrows ideas from mobile robotics. Object surfaces are modeled as occupancy grid maps, and recognition is posed as a problem of localizing the sensor within one of these maps. The second approach applies techniques from computer vision to characterize the *tactile appearance* of objects. Finally, the third approach combines information from the previous two to describe the *spatially-varying appearance* of objects' surfaces. These methods are evaluated in experiments using a physical tactile force sensing system and in a variety of simulations based on our imaging model, and they all exhibit strong performance in their respective domains.

ABSTRACT

Advisor: Professor Gregory D. Hager

Readers: Professor Gregory D. Hager (JHU - Computer Science)

Professor Allison M. Okamura (JHU - Mechanical Engineering)

Assistant Professor Michael Kazhdan (JHU - Computer Science)

Professor Peter Allen (Columbia - Computer Science)

Acknowledgements

I would like to thank my advisor, Prof. Gregory Hager for all his guidance and tutelage over the years. He has fundamentally changed how I think about problems, and I always felt better about things after meeting with him. My thanks also go to Prof. Allison Okamura, with whom I collaborated on several projects, and who first got me interested in haptics; when I first arrived at Johns Hopkins, I never expected it would play such a major role in the research I did here. I thank her also for serving on my Graduate Board Oral (GBO) exam committee and being a reader on my dissertation. I thank Prof. Misha Kazhdan for serving on my GBO committee, for being a reader on my dissertation, and for giving me new insight into the uses of the FFT. I thank Prof. Peter Allen for his useful and directive advice when I was first beginning my thesis work and for being a reader on my dissertation. I also thank Prof. Russ Taylor and Prof. Noah Cowan for serving on my GBO committee.

I extend thanks to everyone else who contributed directly or indirectly to this research. I owe a lot to all the undergraduate students who spent summers and time during the semester on this work: Lucas Estrade, Erica Jantho, Alex Ropson, Caitlin Reyda, Sunny Chen, and Ruby Tamberino. It would have taken a lot longer without their help.

ACKNOWLEDGEMENTS

I thank Jim Gwilliam for including me in his research on human tactile sensing in what proved to be an informative comparison with current robotic capabilities. I also thank Jim and Dr. Steven Hsiao for access to experimental equipment essential for characterizing our sensors. Thanks go also to my collaborator on the exploration process in this work, Erion Plaku, my co-authors from my haptics project group, Dan Ursu and Sarthak Misra, my co-authors on visual tracking and registration work, particularly Sandrine Voros and Ioana Fleming.

I am also thankful for all the other present and past members of the Computational Interaction and Robotics Lab (CIRL): Le Lu, Maneesh Dewan, Henry Lin, Ben Mitchell, Balazs Vagvolgyi, Raphael Sznitman, Hassan Rivaz, Sharmi Seshamani, Carol Reiley, Dan Mirota, Dan Abretske, Pezhman Foroughi, Kel Guerin, Haluk Tokgozoglu, Narges Ahmidi, Nicolas Padoy, Hani Girgis, Eric Meisner, and Rogerio Richa. I am appreciative of all their advice when I got stuck and of our many stimulating conversations over lunch or a drink and around the lab. Additionally, I'd like to thank the administrative staff in CS and the LCSR, particularly Luwanna Spells, Alison Morrow, Jamie Meehan, Cathy Thornton, Debbie DeFord, and Elisa Ahmanson, and all the other friends I made in the LCSR, who are too many to list; they've all made my time here much more enjoyable.

I was quite fortunate to attend a small liberal arts college with an engineering program as an undergraduate. While there, I was able to indulge my varied interests by taking classes in diverse topics and interacting with an even more diverse group of smart and talented people while still pursuing degrees in engineering and computer science. I am especially grateful to my undergraduate advisor, Prof. Bruce Maxwell. It was the classes I took with him (and the research that later led to) that sparked my initial interest in vision and robotics and led me to pursue a Ph.D. in the topic.

Finally, I thank my family for their continued and unwavering support. In particular, my parents, Maureen and Leo Pezzementi, made many sacrifices to ensure that I would have a superior education, and they led me to an interest in science through their own example. And of course I am grateful to my wife, Caroline Carlson; her love, encouragement, and company throughout these years will make me always look back fondly upon my time here.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

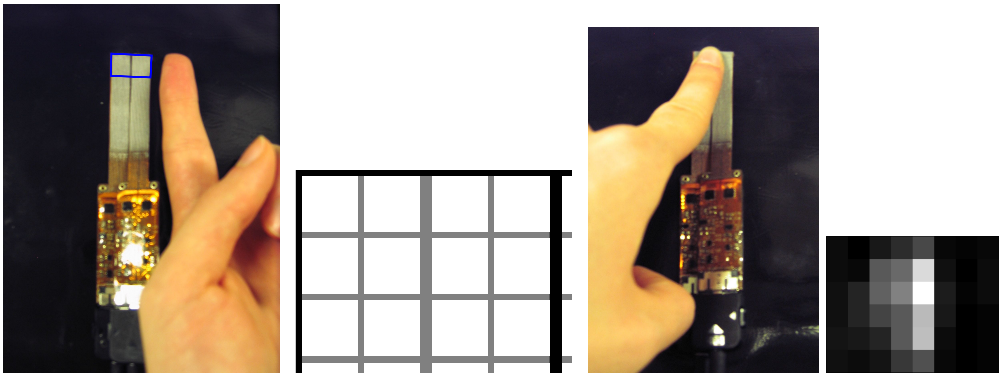
Myriad robotic interaction or automation tasks require the manipulation of objects in the environment. Unless that environment is extremely controlled, the robot or its controller must be able to sense characteristics of the objects involved, such as shape, position, and orientation, to enable grasping or other manipulation. A great deal of research has been conducted on how to extract such information from machine vision systems; therefore, that sensing modality is the default choice for many automation tasks. However, vision systems require both controlled lighting and line of sight to the objects. Maintaining line of sight is particularly problematic in manipulation tasks using a robotic arm, because the end effector usually completely obscures the portion of the object with which it is interacting, which is inconveniently the location one is most interested in sensing at that time.

A natural solution to the problem is to place sensors on the robot's end effector (e.g., fingertips or their equivalent). Since the end effector will be in contact with the object, force sensing is the logical modality to use. Tactile array sensors consisting of a grid of force sensors allow characterization of fine details of objects' geometry and surface properties. Perhaps more importantly, tactile sensing also allows the recovery of physical and dynamical properties that it would be impossible to measure with vision alone, such as surface hardness or roughness.

In order to interact with objects in the environment, one would like to be able to identify those objects, or at least to characterize their shape and surface properties to enable the formulation of stable grasps for manipulation. It is also important to be able to estimate the pose of a manipulated object through touch sensing, to verify, for example, whether it has shifted in the robot's grasp, such that the robot needs to re-grasp it to avoid dropping it. Alternatively, consider the act of fishing in your pocket to find a key among the other items that might also be there, pulling out that key and using it to unlock a door. Humans are able to accomplish this task using only touch sensing and the visual information provided by a glance at the door to find the location of the lock, yet several steps of this process remain major challenges for a robot. In this work, we focus on the application task of object recognition: during a training period, one is presented with a set of objects to explore and learn about; then, one is presented with some unknown object, and asked which of the previously seen objects it is. A useful sub-task within this problem is to estimate the pose of the unknown object, provided there is enough information to do so. If one is still uncertain of the choice from the currently available information, one would like to be able to provide hypotheses for the probability of each object being a match, to guide further tactile exploration toward providing the most relevant information to resolve the ambiguity. Many researchers have approached this problem using data from high-density laser range scanners, making use almost entirely of geometry information for decision-making. This research will instead make use of tactile sensing, allowing future work to also explore simultaneous manipulation of the object.

1.1 Overview of Tactile Array Sensors

At this point, it is useful to define exactly what is meant by a tactile force sensor, to give a few examples of their usage and provide some intuition for their output. Tactile sensing in humans is generally defined as any sensation associated with the skin, as opposed to force sensing, which originates in muscles and joints. This comprises a variety of physical phenomena, including temperature, vibration, and surface texture. Artificial sensors have been developed for robotic sensing of all of these phenomena, reviews of which are provided in Tegin and Wikander (2005); Dargahi and Najarian (2005). Our focus is on the last of these phenomena, the local geometric properties of the surfaces of objects and features and statistics derived from these properties. In this work, a tactile force sensor should be taken to mean a sensor that detects distributions of force over a surface. These are usually built as a collection of many individual pressure sensing elements arranged into a grid to form a single composite sensor. Such sensors have been available for decades, but their response characteristics (e.g. noise, linearity), size, and resolution have steadily improved



(a) Sensors with Finger (b) Sensor Layout (c) Pressing on Sensors

Figure 1.1: (a) PPS DigiTact sensors alongside a human finger. The sensing area is highlighted in blue, and it can be seen to be comparable to that of a fingertip. (b) The layout of individual sensing elements within the sensing area. (c) The response of the sensors when a finger is pressed against them with moderate force.

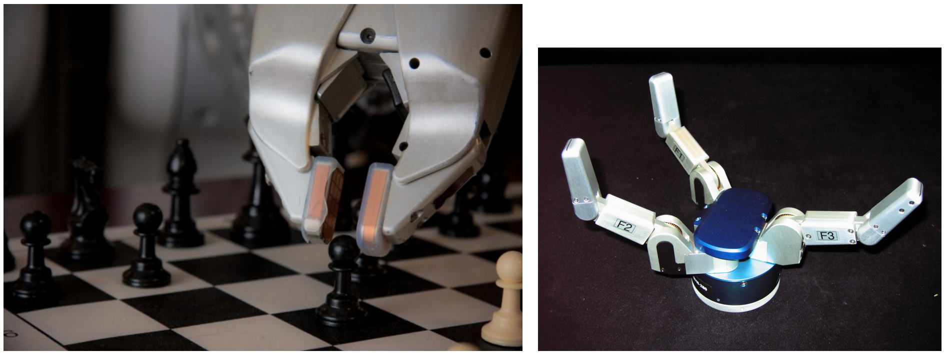
with fabrication processes. Because the outputs of this type of sensor consists of a number of force measurements arranged in a grid[1], it is natural to think of them as "tactile images", and this perspective forms the basis of most of the ideas presented in this work.

Most tactile sensors used in robotics are based on either capacitive or resistive sensing.

- In **capacitive** tactile sensors, each sensing element consists of two electrodes separated by a gap of air, as shown in Figure 1.3. When a force is applied, the two elements are pressed closer together, shrinking the air gap and increasing the capacitance of the circuit, which can be measured electrically.

- In **resistive** tactile sensors, sensor elements are made of a piezo-resistive material whose resistance changes under an applied force.

[1] It is not actually necessary for sensing elements to be arranged in a rectangular grid for most of the techniques developed in this work to be applied, but that is the configuration in which the sensor readings' image-like qualities are most apparent.



(a) Gripper of Willow Garage's PR2 (b) Barrett Hand

Figure 1.2: (a) The gripper of Willow Garage's PR2 (Willow Garage, 2011) in a fine manipulation task with its capacitive tactile sensor arrays visible on the fingertips. (b) The Barrett Hand, which is available with similar tactile sensors integrated in the fingertips.

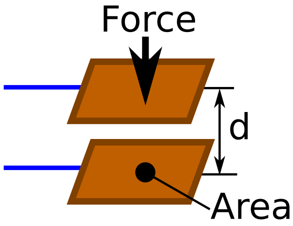


Figure 1.3: Capacitive sensing method: An applied force decreases the distance, d, between the two plates, resulting in a measurable change in the capacitance, $C \propto \frac{Area}{d}$

Typically, resistive sensors can be made smaller, with higher spatial resolution than capacitive sensors, but capacitive sensors have better force sensitivity and a more repeatable response. The sensors used throughout this work are capactive, but the algorithms we introduce are equally applicable to resistive sensors.

At the time of writing, the major commercial sources of tactile sensors for robots are Pressure Profile Systems, Inc. (PPS, Los Angeles, CA, USA) for capacitive sensors and Tekscan (Boston, MA, USA) for resistive sensors. The sensors used in this work are a custom DigiTacts system made by PPS. Figure 1.1(a) shows the sensors alongside a human finger to give an idea of scale. The actual sensing area is highlighted in blue, and Figure 1.1(b) shows how individual sensing elements are laid out within that area. Each sensing element is square and 2mm to a side (including some non-sensing material between elements) and acts as an individual pressure sensor. The response of pressing a finger against the sensor with moderate force is shown in Figure 1.1(c) as a gray-scale image. The intensity of each element is proportional to the input force detected. All of the experiments and simulations in this work are based on this sensor system.

These sensors are representative of those used in high-end manipulators. Similar capacitive tactile sensor systems are being integrated into current- and next-generation models of the grippers of well-known robot. Another system made by PPS is currently in use for tactile sensing on Willow Garage's PR2 robot (Figure 1.2(a)), and PPS capacitive sensors are also being integrated into the fingertips of the next model of the Barrett Hand (Figure 1.2(b)).

For comparison, Tekscan's highest-resolution resistive sensor currently available is model 5027, which has a 44×44 array of sensing elements in a square of side length 1.1 in. and a manufacturer-reported sensing range of 50 to 500 psi. Even higher resolutions are possible using recently developed optical tactile sensors, such as those of Johnson and Adelson (2009); here, the resolution is determined by the camera used to view the underside of a gel that acts as the sensing surface. These sensors are still much larger than their capacitive or resistive counterparts, though, and are not yet commercially available. Nonetheless, all of the algorithms presented in this work are directly applicable to the outputs of these alternative types of sensors.

Most of the techniques presented can also be applied to tactile sensors that consist of only a single pressure sensor combined with accurate proprioception information. This extends also to skin-like tactile sensors, such as those of the ROBOSKIN group (Dahiya et al., 2009), provided one is able to accurately determine the positions of individual sensing elements. For the methods in later chapters, it may be necessary to mosaic together the outputs of such sensors to form images in order to apply "appearance"-based analysis or to adapt the image representation to support an off-grid sampling arrangement, but the principles of each method are still applicable.

1.2 Problem Statement

Let $O = \{\mathbf{O}_1, \mathbf{O}_2, ..., \mathbf{O}_{n_O}\}$ be the identities of a set of objects that one would like to recognize. The system performing the recognition consists of a robot manipulator with one or more tactile force sensors attached to end effectors. Each object is explored by the robot in a training period to build a model to be used for recognition. The information acquired during this process is denoted $\mathbf{M}$, which can be thought of as a collection of object maps. Then, evaluation consists of a series of trials in which an unknown object from the set, $\mathbf{O}_j$, is presented to the robot for identification. At any given time in the process of exploring the unknown object, the system should be able to provide a probability distribution of the object identity over O, given all the data acquired up to the present. Formally, let z_t denote a sensor measurement taken at time t and $z_{1:t}$ be the set of all readings taken up to time t for a given object. For our purposes, a sensor reading consists of pressing the sensor against the surface of the object in one location and recording its response in static contact. Similarly, let $u_{1:t}$ be the set of controls sent to the robot to produce those measurements, allowing estimation of the position and orientation of the sensor. Then our goal is to estimate $\Pr(\mathbf{O}_i \mid z_{1:t}, u_{1:t}, \mathbf{M})$. The best estimate of the object identity can then be obtained as $\arg\max_i \Pr(\mathbf{O}_i \mid z_{1:t}, u_{1:t}, \mathbf{M})$, but the full probability distribution may also be used for higher-level reasoning.

During the training process, the robot is allowed to exhaustively explore the object to build its internal model. During recognition, however, the information necessary to estimate object identity should be minimized. We would like to be able to identify the unknown object using as few sensor readings as possible.

1.2.1 Thesis Statement

Tactile force sensors provide rich information about object surfaces. In addition to contact positions and orientations, tactile sensor info can be used to recognize and localize objects without the use of other sensing modalities.

1.2.2 Problem Setup Details and Assumptions

We make a few practical assumptions, which do not limit the applicability of the algorithm. First, we assume that the objects are all smaller than the workspace of the robot system and significantly larger than the sensor resolution, to allow full and meaningful exploration. Second, we assume that the distinguishing features on objects' surfaces are also detectable at the sensor resolution.[2] In this work, we also assume that the objects are rigid and that the robotic system need not account for the effects of its exploration on their deformation. We make no further assumptions on the topology or convexity of the objects though.

Although the pose of the object to be recognized is unknown, we assume that it remains constant throughout exploration; i.e., the haptic system is able to explore the object without disturbing it. This requirement is relaxed in Chapter 4 though.

This work focuses only on the modeling and recognition tasks, and not the exploration process. We will therefore not consider details of the kinematics of the robot system, and we restrict our examination of control to that which is tightly coupled to the sensing process. The robot kinematics are also assumed to be known precisely enough to not introduce appreciable error.

[2] These assumption account for the fact that the set of objects one would try to recognize using a micromanipulator is different from the set that would be reasonable to differentiate using a human-sized robot hand. It is important to realize, however, that the scales at which sensing occurs (i.e., the relative scales of the object and sensor and the resolution of the sensor) fundamentally determine what object structural features and surface textures can be detected and characterized. For instance, a sea urchin observed by a large, low-resolution sensor might be indistinguishable from a smooth ball.

1.3 Prior Work on Haptic Object Recognition

Haptic object recognition using tactile sensors first got wide attention in the mid-1980s. Bajcsy (1984), outlined the types of measurements that can be made by a single-fingered robot exploring an object, and Bajcsy and Hager (1984) outlined a set of basic primitives to characterize all tactile features. Soon thereafter, Lederman and Klatzky (1987) performed a set of psychophysical experiments to analyze how humans go about haptic exploration, and they presented a set of "exploratory procedures" (EPs) that became the basis for many robotic exploration algorithms. These EPs included procedures for estimating a wide variety of object properties, including geometric properties like weight, volume and shape as well as tactile percepts like temperature, texture, and hardness. Tactile sensors available at the time were deemed inadequate for much more than measuring contact *vs.* no contact (Bajcsy, 1984), and they were generally used only to localize a fingertip's contact point(s) (Bay, 1989; Allen and Michelman, 1990).

As a result, early work in haptic object recognition focused almost entirely on object geometry. Grimson and Lozano-Perez (1984) viewed the problem as purely combinatoric, enumerating all possible objects and (discretized) poses from their precomputed library that could fit a given set of contact points. As shape modeling progressed, other methods made use of generalized cones (Fearing, 1990b), superquadrics (Allen and Roberts, 1989), polyhedral models (Casselli et al., 1995),and self-organizing maps (Faldella et al., 1997), also constrained by a set of contact points. All of these methods were restricted to dealing with convex objects, though, and all but Allen and Roberts (1989) had little tolerance for change in pose.

With increased availability of laser scanners and the ubiquity of cameras, most research in 3D object recognition shifted to the vision domain. As we will show, some of the algorithms that resulted from this research can be applied to the haptic domain (and more relevant related work will be presented with the applications), but research into object recognition using only haptic information dwindled until recently. Meanwhile, most research using tactile sensors has focused on aiding manipulation and using the sensors to evaluate and maintain stable grasps of an object (Fearing, 1990a; Son and Nowe, 1996; Allen et al., 1999). Some investigators have begun to explore some of the active sensing of dynamic properties described in Bajcsy (1984), to improve the quality of object models that can be rendered to a user (MacLean, 1996; Pai et al., 2000; Patoglu and Gillespie, 2003; Kuchenbecker et al., 2011), but these techniques have not yet, to our knowledge, been combined with geometry for object recognition.

Gorges et al. (2010) investigates the use of an exploration strategy based on focus of attention for recognizing objects using tactile array sensors. Iterative closest point (Besl and McKay, 1992) is used to perform recognition from clouds of contact points acquired during exploration. Their simulations show the strategy improves recognition performance (over random exploration) on one object from their set of seven.

1.3.1 Comparison to Vision and Range Sensing

In the haptic domain, a single sensor reading contains much less information than a stereo pair or a range image, since the tactile sensors have many fewer elements than the number of pixels in a high-resolution image and the tactile sensors are only able to sense surfaces with which they are in contact. The latter difference results in, effectively, a much smaller field of view. Geometric information, however, is directly available as the point of contact in the robot frame, given by the resolved position of the fingertip, from the robot's forward kinematics. The readings from the tactile sensors then provide information about local surface texture. How best to extract information about geometric and textural surface properties from the raw sensor values is a relatively new field of research.

A variety of 3D shape-matching algorithms grew out of research into 3D object recognition using vision and range data, an overview of which is available in Tangelder and Veltkamp (2004). Most of the algorithms from the vision/range-sensing domain are not directly applicable to our situation, since they rely on matching models of the full object or knowing the object pose, whereas we would like to be able to do estimation using only partial shape information, before even a rough gross shape model is available. This requirement narrows the classes of applicable shape matching algorithms in Tangelder and Veltkamp (2004) to those based on surface connectivity subgraph matching (arising largely from the CAD community) and those based on local descriptors. Because of the instability of topology estimates when dealing with real sensor data, local descriptor-based methods (e.g. Chua and Jarvis, 1997; Johnson and Hebert, 1997; Gal and Cohen-or, 2006) seem the most promising for application to haptics, and these methods inspired the appearance-based approach of Chapter 4 and its integration with geometry information in Chapter 5.

1.3.2 Haptic Localization and Mapping

One of the driving insights in the techniques presented in this dissertation is the relationship between object recognition and the localization and mapping problems in mobile robotics. Okamura (2000) described techniques for simultaneously manipulating and exploring the surface of an object, and Schaeffer and Okamura (2003) enunciated the goal of haptic SLAM (Simultaneous Localization and Mapping). Although the techniques in that work are limited to haptically localizing a known object in an unknown pose, they form an important conceptual basis for the work in Chapter 3. Further work on haptic localization of known objects is discussed in that chapter. We build upon this work to model unknown objects with the aim of enabling the full simultaneous localization and mapping process.

1.3.3 Tactile Feature Extraction

Previous work on the extraction of features from tactile sensor readings has been targeted at domain-specific problems, whereas our goal is to develop a recognition system that is applicable to a broad range of shapes. Allen and Michelman (1990) describe work on feature extraction using tactile sensors with an object recognition framework that uses them for localizing contact points. Stansfield (1987) introduces a similar feature extraction strategy, also for features differentiating point, edge, and planar contacts, but no more extensive features are included (such as for describing surface texture or handling multiple contact points). Okamura and Cutkosky (1999) uses cylindrical tactile sensors to extract curvature-based features of an object surface derived from the path of the sensor moving across the surface, while the sensor itself is used to maintain contact. Briot et al. (1979) presents a probabilistic approach to tactile mapping and identification of 3D objects, but objects were restricted to being piecewise planar, where the entirety of one of these planes must be visible in each sensor reading. Similarly, Hillis (1982) presents a system for recognition of 6 nuts and bolts smaller than a tactile sensor that is capable of distinguishing the slot in the top of a screw, but he extracts a set of only 3 binary features to differentiate this small set (round vs long, bumpy vs not, and stability when rolling). However, neither of Briot et al.'s or Hillis's work appears to have been followed up upon. More recently, Petriu et al. (2004) presented a neural-network-based method for using tactile force sensors to identify and localize objects whose faces have all been embossed with a set of symbols from a predefined library.

The potential descriptiveness of even very low resolution tactile information has been demonstrated in human subject tests, however, suggesting automated systems could also extract useful information from sparse tactile data. In psychophysical experiments, Yanagida et al. (2004) showed humans sitting in a chair equipped with a 3×3 grid of "tactile" displays were able to achieve 87% accuracy in a character recognition task.

Concurrent with this research, one other group (Schneider et al., 2009) recently began investigating the concept of surface “appearance” using tactile sensors. This approach takes its inspiration from appearance-based object recognition in the vision domain, in which an object is represented by a collection of small image patches that make it up, without regard to the relative geometry of those patches; the object is therefore modeled entirely on local characteristics, while its global structure is left free to vary. Conceptually, this amounts in the haptic domain to describing an object by its local surface texture. The details of the approach are described further in Chapter 4, and we explain how our work contrasts with and expands upon that of Schneider et al. (2009).

1.4 Contributions

The major contributions of this dissertation are outlined below. These contributions are elaborated upon in their respective chapters.

1.4.1 Imaging Model for Tactile Force Sensors

We develop a model of tactile force sensors as imaging sensors, in particular, analyzing the effect of the commonly-used rubber covering placed over top of tactile force sensors on the tactile images they produce. While simple in concept, such a model allows numerous techniques from the computer vision and image processing literature to be applied to the haptic domain, including the remainder of the contributions below.

1.4.2 Tactile Force Sensor Simulator

The above imaging model for tactile force sensors has been incorporated into a simulator for tactile force sensors and released to the public (available from `https://cirl.lcsr.jhu.edu/Research/MAPS/TFSS`). Our model is implemented to make use of graphic rendering techniques to efficiently simulate the response of virtual tactile sensor being pressed against arbitrary 3D objects. It can be easily incorporated into larger simulation environments and thus facilitates the development of novel haptic interaction techniques. It also allows evaluation of various tactile force sensor characteristics (e.g., size, resolution, covering material and thickness) without having to go through the expensive process of building or buying a prototype of each.

1.4.3 Tactile Object Surface Mapping

"Occupancy grid" mapping is a popular technique in mobile robotics that divides the world into a set of cells arranged in a grid, each of which is either occupied or empty; this is commonly the starting point for path planning and navigation algorithms. We adapt this technique to the haptic domain and present a method for generating such maps of the surfaces of objects from tactile array sensor readings. This method provides a way to generate area or volumetric models of objects from touch data that can then be used to estimate properties of the object from touch or even in other modalities (e.g., for visual recognition).

1.4.4 Tactile Object Recognition and Localization from Occupancy Grid Maps

We develop an algorithm for using a set of occupancy grid maps of objects to recognize the objects those maps represent and to estimate their pose during haptic exploration. To our knowledge, this is the first application of sequential state estimation techniques to object recognition.

1.4.5 Characterization of Tactile Appearance

We explore the concept of tactile appearance and analyze several ways of describing objects' surface texture. These are incorporated into an appearance-based recognition framework inspired by "bag-of-features" techniques from computer vision. A set of novel descriptors are introduced for characterizing surface texture. The requirements on the exploration process for extracting consistent, useful information are also analyzed, and a novel set of controllers that meets these requirements is defined and demonstrated.

1.4.6 Unified Geometry-and-Appearance Framework for Tactile Recognition and Localization

We build upon the occupancy grid-based (geometric) recognition approach and the appearance characterization methods mentioned above to create an algorithm that makes use of both geometry and appearance information. An object is described by how its tactile appearance varies spatially across its surface, and pairs of observed surface patches of an object are used to probabilistically constrain its identity and pose. This is the first recognition algorithm to our knowledge to combine local tactile appearance and geometry information, and the method can also be applied to (or in combination with) other sensing modalities, such as vision.

1.5 Format of the Dissertation

The chapters of this dissertation can be read out of order to some degree, but there are some important dependencies in the material presented:

- All of the algorithm evaluations are based on the sensors characterized in Chapter 2, and the simulations of Chapters 4 and 5 make use of the sensor model also introduced in Chapter 2.

- Additionally, the spatially-varying appearance approach of Chapter 5 builds on ideas from the geometry-only and the appearance-only methods (of Chapters 3 and 4 respectively), so it would be useful to review those chapters first.

- As one might expect, the conclusions in Chapter 6 reflect upon the entire dissertation.

That said, an effort has been made to refer back to specific relevant previous material to support out-of-order reading whenever possible.

Notation is generally defined in each chapter when it is first used. A glossary of notation used throughout the dissertation (i.e., in more than one chapter) can be found at the end of the dissertation.

1.6 Relevant Publications

This dissertation is based in part on the following publications:

- Z. Pezzementi, E. Jantho, L. Estrade, and G. D. Hager. Characterization and simulation of tactile sensors. In *Haptics Symposium*, pages 199–205, Waltham, MA, USA, 2010

- Z. Pezzementi, C. Reyda, and G. D. Hager. Object mapping, recognition, and localization from tactile geometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011b

- Z. Pezzementi, E. Plaku, C. Reyda, and G. D. Hager. Tactile object recognition from appearance information. *IEEE Transactions on Robotics*, 2011a

Chapter 2

Simulating Tactile Force Sensing

When evaluating the feasibility of a wide variety of scenarios, a simulation system is particularly useful. Since we wished to analyze the effects of varying several parameters of our sensor system, we formulated a sensor system with the following requirements:

1. Its output should reproduce the response of real sensors with sufficient fidelity to validate algorithms applied to it.

2. It should also be efficient to support real-time simulation of robotic interactions with simulated 3D objects.

This first condition requires an understanding of the characteristic response function of sensors of this type. We first describe our test set of tactile force sensors and our investigation and modeling of this function as a set of imaging transformations in Section 2.1. The details of the tests conducted to derive this model follow in Sections 2.2 and 2.3, forming most of this chapter. We then describe some design guidelines that emerge from this view of tactile sensors, with regards to selecting an appropriate covering for one's sensors, in Section 2.4. Next, we address the second requirement above: Section 2.5 describes how the models we developed were incorporated into an efficient tactile sensor simulator and gives some comparison between the output of our test sensor system with the simulation thereof. Finally, we conclude in Section 2.6 with some discussion of the consequences of our results.[1]

2.1 Tactile Force Sensors as Imagers

Tactile force sensors are typically covered with a soft, elastic material, such as rubber. The main purpose of this covering is to protect the sensors from damage, particularly from shear forces. It also has some major effects on the response of the sensor, though:

- A covering material (that is softer than the sensors themselves) increases the range of displacements that can be detected by effectively acting as a position-to-force transducer.

- A covering will tend to spread the response of the sensor to a stimulus across the sensor. E.g., a point load applied to the top of the covering will result in force over an area at the bottom of the covering, which is in contact with the sensor.

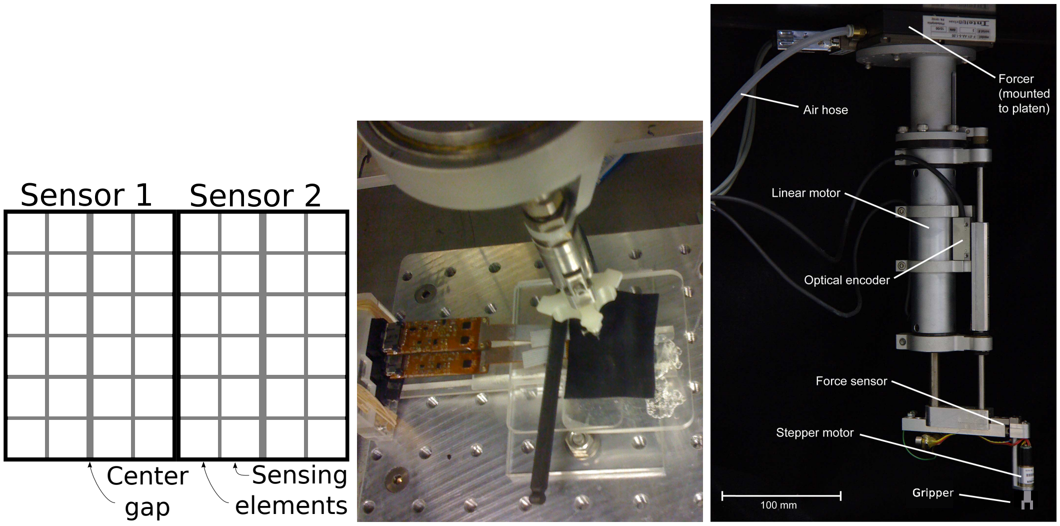
Because of the latter effect, the covering acts as a low-pass filter, offering an interesting trade-off between spatial and force resolution through the selection of its thickness and

[1] This chapter covers work presented in Pezzementi et al. (2010). This work was carried out with the assistance of Erica Jantho and Lucas Estrade during their Research Experience for Undergraduates residences.

stiffness.

As in (Fearing and Binford, 1991), we use a point spread function (PSF) model of the effect of the covering. Fearing and Binford analyzed the sensitivity of a hemispherical tactile sensor for discriminating simple shaped indenters. Based on careful physical measurements, they modeled the effect of a compliant sensor covering using an impulse response function taking the form of a truncated quadratic. They then suggest the use of inverse filtering techniques to infer the actual surface contact structure. Others propose the use of SVD (Ellis and Qin, 1994), neural nets (Canepa et al., 1992b), or radial basis functions (Canepa et al., 1992a) to the same end. However, any such approach is inherently ill-posed, as the associated deconvolution problem does not have a unique solution. In this chapter, we instead take the view that a tactile force sensor can be simply characterized as a device for producing tactile images, with no attempt to infer the physical causes of images. Our focus is therefore on modeling the forward tactile image formation function, rather than inferring the inverse function. Since the deformation in our system is relatively simple (the vast majority of the motion is restricted to one degree of freedom of a uniform, isotropic material) we show that PSF models suffice to capture the effects of deformation.

The use of an imaging model that maps physical effects to image transformations allows us to re-cast the task of simulating tactile sensors as one of graphics rendering. This is practically demonstrated by our simulator (described in Section 2.5), which we show produces images that are qualitatively similar to those obtained from the physical sensor. This in turn opens the door to a wealth of techniques from computer vision and image processing,



(a) (b) (c)

Figure 2.1: (a) Diagram of our sensing system consists of a 1-by-2 arrangement of sensors, each of which comprises a 6-by-4 array of 1.8mm square sensing elements. 0.2mm of (non-sensing) spacing is necessary between the individual sensors, except for the middle column of each sensor, where this gap is doubled. (b) The actual sensors mounted in the workspace of the robot with indenter. (c) The robot arm used to stimulate the sensors.

including “tactile mosaicking” and image-based representations of surface “appearance”, which are demonstrated in Chapters 3 and 4 respectively. The tests used to characterize the force response of the sensor as a function of displacement of the covering are presented in Section 2.2, and the characterization of the point spread function associated with the covering follows in Section 2.3. The test system used for the tests in both these sections is presented next, in Section 2.1.1.

2.1.1 Test System Details

The tactile force sensors we modeled in this work were a custom DigiTacts system from Pressure Profile Systems (PPS, 2008), consisting of two small sensors, each of which was 12mm-by-8.5mm and contained a 6×4 grid of square sensing elements with 1.8mm on a side, for a total of 48 sensor elements making up a 6×8 "tactile image", as shown in Figure 2.1(a). There was a 0.1mm gap at the edges of and between adjacent sensing elements, except down the center of each sensor, where the gap was 0.2mm, for a roughly-uniform spatial resolution of 2mm. The total sensor system footprint is 12mm×16.7mm. The sensing modality was capacitive, with a sample rate of 30 Hz, a manufacturer-reported sensitivity of 0.1 psi, and a sensing range of 0-20 psi, though our interaction forces were restricted to the bottom of this range. The sensors were covered with a deformable covering to avoid damaging the sensors and to investigate the effect of this covering on the sensors' imaging properties. This covering was composed of layers of polyurethane with a durometer rating of 40 OO (McMaster-Carr Part 8824T112), with thickness varying from 0.04" to 0.2".

The robot arm used to stimulate the sensors, shown in Figure 2.1(c), was a custom-made platen-forcer system with a resolution of 3μm in the horizontal directions and 1μm in the vertical direction. The sensors were fixed to a flat surface within the robot arm's workspace, and the arm was outfitted with a cylindrical indenter 1mm in diameter, visible in Figure 2.1(b), which was pressed into the sensor covering to stimulate the sensors in the tests of Sections 2.2 and 2.3.

2.2 Force Response Characterization

We first performed a set of indentations to determine the linearity and uniformity of response across the sensors' surfaces, using the setup described in Section 2.1.1. For this test, the thinnest covering (0.04") was used, to minimize the point spread effect and more easily concentrate forces on a single element at a time. The robotic arm was programmed to press into the center of each sensing element of the tactile sensor, depressing the surface of the covering incrementally by displacements of 0.2, 0.3, 0.4, 0.5, and 0.6 mm. For each indentation, the sensor readings over the 250 ms period of contact for each sensor element were averaged to get a single reading that was used for further processing.

The arm's position was calibrated by pressing down on a sensing element to obtain a response that only excited that single element (and no adjacent elements). This process was repeated for two distant sensing elements close to opposing corners of the sensor. The locations of those elements in the robot frame were measured to establish the sensor's position and orientation in the robot frame. For each of these elements, the position of the arm was adjusted to obtain a response which only excited that element (and no adjacent elements). The height of the surface of the covering in the vertical direction was measured at a single point and assumed to be uniform across the sensor. The robot then indented the rubber covering over each sensor element at the estimated center of the element, starting at the surface.

The overall response of the sensors was determined to be nearly linear in the range tested, though responses of individual sensing elements displayed some variation, as shown

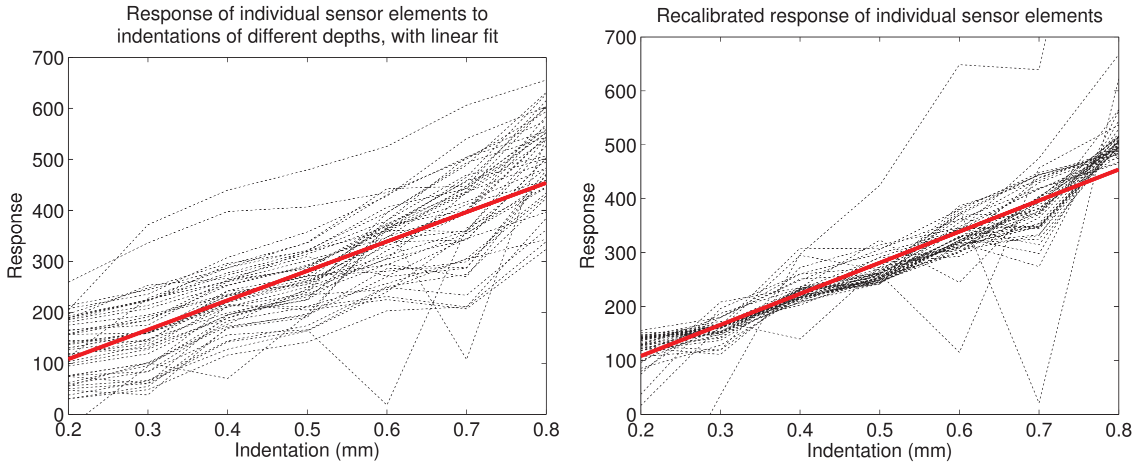


Figure 2.2: (a) Measured responses to displacements of different depths. (b) Re-calibration of the same values to a consistent linear fit. In both cases, dotted black lines give the response of individual sensor elements to a point load at varying indentation levels. The solid red line shows a linear fit to these values.

in Figure 2.2. Each dotted line in this figure shows the response *vs.* indentation depth of an individual sensor element, while the thick red line shows a linear fit to these values, which has slope a and intercept b[2] . Two elements (each located in the same corresponding position on each sensor) were omitted from this fit due to observed systematic errors. They responded to a stimulus of one adjacent element, but not to the element itself. Figure 2.2 shows the result of performing a linear fit to the response of each individual sensing element, $res_i(d) \approx a_i d + b$, and re-linearizing the output to a consistent linear output, $adj(d)$, according to

$$adj_i(d) = b + a\frac{res_i(d) - b_i}{a_i} \tag{2.1}$$

[2] These data were collected after repairing some damage to the sensors that was discovered to have been present when the corresponding data in Pezzementi et al. (2010) were collected. With the repaired sensors, the linear trend is more apparent.

2.3 Point Spread Characterization

Next we wished to characterize the effect of the sensor covering on the tactile imaging process. As discussed in Section 2.1, the covering acts as a low-pass filter on the spatial response of the sensor, so we conducted tests to determine the frequency characteristics of this filtering and how they vary with the thickness of the covering.

Shimojo (1994) investigated the low-pass filtering effects of tactile sensor coverings and fit models to them that strongly resemble Gaussian point spread functions. We therefore investigated the accuracy of approximating the response as a Gaussian, first with some basic finite element tests, then with data collected from the physical system.

2.3.1 Finite Element PSF Tests

We tested the response of a linear finite element model (FEM) of our sensor-and-covering system to a point load applied at the center of the top of the covering. The forces that would be measured by the tactile sensor were taken to be the forces at the bottom layer of the FEM. The FEM was composed of layers of 100×100 nodes each, and the number of layers in the FEM was varied to analyze the effect of the thickness of the covering on the sensor response. In our setup of the sensor system, the covering material always extended well beyond the edges of the sensors; as a result, sensing always occurred far from the edges of the covering material, and the response was assumed not to vary across the sensor (except through the different responsiveness of the individual sensing elements, characterized in Section 2.2).

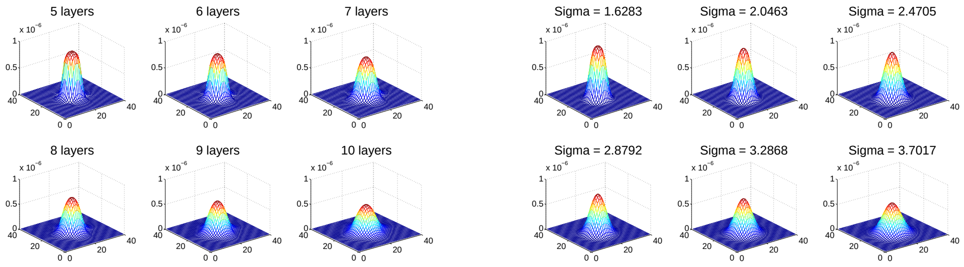
Figure 2.3(a) shows the sensor response predicted by the FEM with 6 different thicknesses. Let a Gaussian with mean μ and standard deviation σ be denoted as

$$\mathbf{G}_{\mu,\sigma}(r) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(r-\mu)^2}{2\sigma^2}\right) \tag{2.2}$$

Figure 2.3(b) shows the result of instead convolving the input point load with a Gaussian point spread function, $\mathbf{G}_{0,\sigma}$ with spread $\sigma = 0.5t + 0.9$, where t is the thickness (number of layers) of the covering, obtained by fitting to the FEM data. The two are nearly identical, particularly after the discretization effects inherent in array-type sensors. We therefore used a Gaussian point spread function with varying σ to fit to the physical sensor responses. It should be noted that the linear elastic deformation model used by the FEM can only be considered accurate for small deformations, but it nonetheless motivated the collection of data to verify this trend. Moreover, although the stress/strain relationship is in fact nonlinear, whereas Gaussian filtering is linear, our experiments with the physical sensors indicate that the linear trend nonetheless holds for different forces in our interaction range applied to a covering of a particular thickness.

2.3.2 Physical Sensor PSF Tests

Because of the relatively low spatial resolution of our tactile sensors, it was not possible to estimate the point spread from a single tactile image, as only a highly-discretized version of the response is visible in each image. Instead, we collected many sensor readings



(a) FEM Response

(b) Gaussian PSF

Figure 2.3: (a) Finite element model prediction of impulse response of coverings of different thicknesses. The node forces at the center of the bottom layer of a 100x100 linear elastic FEM are shown when a point force is applied to the top center node. The number of vertical layers of elements in the model is varied from 5 to 10, giving the differently-shaped responses shown. (b) Convolution of the same input with a Gaussian, $\mathbf{G}_{0,\sigma}$, with $\sigma = 0.5t + 0.9$

while translating the input force across the covering surface in small increments, and we performed our fits on the aggregated data.

2.3.2.1 Test Design

Because of this translation of the input force, the tests conducted were similar to those of Section 2.2, but on a smaller scale. We again used the test setup from Section 2.1.1. In this test, the force outputs were corrected by the per-pixel linear response model obtained in Section 2.2 to obtain a more uniform response. Indentations were made at points in a 15×15 grid with a spacing of 0.1 mm centered on one sensing element, chosen because it exhibited low noise characteristics and seemed representative of the overall sensor response. The indentations therefore traversed that element's entire sensing area as well as

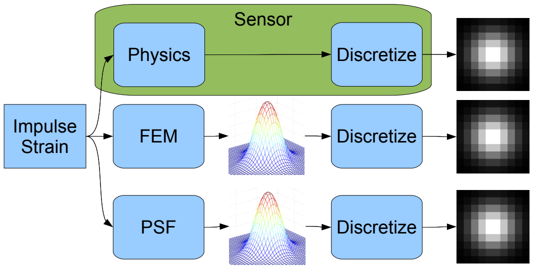


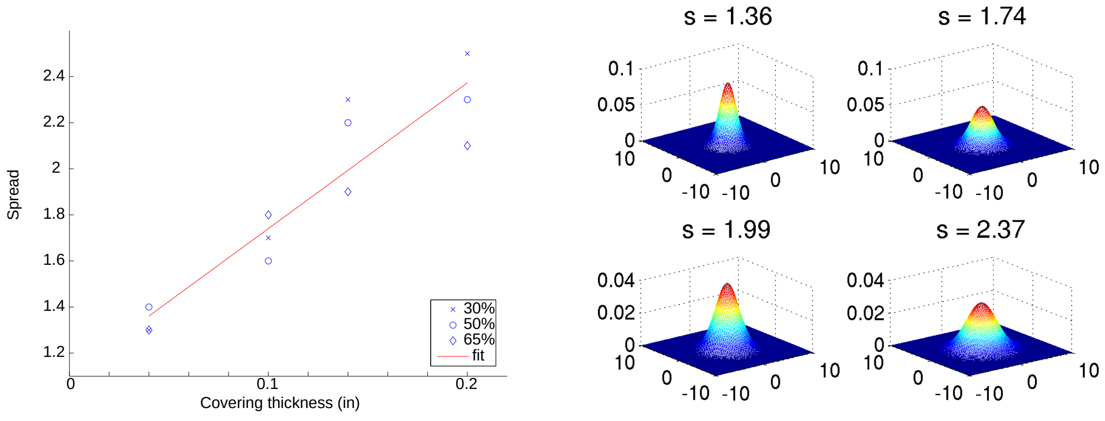
Figure 2.4: Different imaging models for tactile sensing. Nearly identical outputs can be obtained from a finite element model (FEM) or point spread function (PSF) model as from the actual sensor.

the gaps between sensing elements and partially overlapped with the adjacent elements. Tests were conducted with coverings of thicknesses 0.04 in., 0.1 in., 0.14 in., and 0.2 in., built from 0.4 in. and 0.1 in. layers of the polyurethane material used in the force response tests of Section 2.2. At each thickness, indentations were made to 30%, 50%, and 65% of the covering thickness to determine whether the response characteristics of Section 2.2 still held as the covering thickness was increased.

2.3.2.2 Results

To estimate the sensor-covering system's impulse response, we fit a rotationally-symmetric zero-mean Gaussian function to the data, varying its "spread" parameter, σ, giving $\mathbf{G}_{0,\sigma}(r)$ where $r = \sqrt{x^2 + y^2}$.

At a higher resolution, a set of simulated input profiles, $\mathbf{D}_i$, $i = 1, 2, \ldots, 225$ were defined to match the shape and location of the robot indentation device used to collect sensor data. A candidate point spread function, $\mathbf{G}_{0,\sigma}$, corresponding to a particular choice of σ, was



(a) (b)

Figure 2.5: (a) Point spread function spread parameter vs covering thickness, with linear fit. (b) Exemplar point spread functions with the parameters corresponding to the fit at each covering thickness.

applied to this profile. The result was then normalized and down-sampled by a function, $\mathscr{S}$, to match the resolution of the actual sensor output, $\mathbf{I}_i$, accounting for gaps between sensing elements. The resulting simulated sensor reading is thus $\mathbf{E}_{i,\sigma} = \mathscr{S}(\mathbf{D}_i * \mathbf{G}_{0,\sigma})$, where $*$ denotes convolution in the spatial domain. The correlation between measured and simulated images, $q_{i,\sigma}$, was used as the quality of fit metric for each image, and an overall quality of fit was computed as

$$q_{i,\sigma} =< \mathbf{E}_i, \mathbf{I}_i > \tag{2.3}$$

$$Q(\sigma) = \sum_i q_{i,\sigma}. \tag{2.4}$$

The best-fit value, σ^+ was found by maximizing $Q(\sigma)$ over all values of σ.

Figure 2.5(a) shows σ^+ values for each thickness and indentation depth. A linear fit was applied to these points to derive an estimate of σ^+ as a function of covering thickness,

t, both expressed in sensor element widths. The result is:

$$\sigma^{+} \approx 1.11 + 0.497t \tag{2.5}$$

Exemplar PSFs with σ values from this equation for each covering thickness are shown in Figure 2.5(b). The large value of the intercept of Equation 2.5 is likely due to the fact that the sensor has an inherent point spread function itself, even without a covering applied.

2.4 Choosing a Covering

It is clear from Section 2.3 that the covering on top of the sensor strongly affects the type of readings it will produce. As discussed in Section 2.5, we model the sensor and covering as a unified system for producing simulated outputs. A natural question would be how to choose an appropriate covering, or what factors are important to consider when designing a sensor configuration for a particular application. We begin this section with some theoretical analysis of spatial acuity, the sensor property that seemed most relevant to our chosen application, the object recognition problem, in Section 2.4.1, then address some of these other concerns in Section 2.4.3.

2.4.1 Optimizing for Spatial Acuity

A simple model of the sensor's discretization processes was developed to analyze the optimal covering thickness to use to maximize spatial acuity. The response of element i of a discretizing sensor was considered to be defined by a function $\mathbf{S}_{\mathbf{w}_i,\mathbf{f}}(\mathbf{X})$ of spatial resolution (sensor element width) **w** and force resolution **f** defined by

$$\mathbf{S}_{\mathbf{w}_i} = \int_i \mathbf{X} \tag{2.6}$$

$$\mathbf{S}_{\mathbf{w}_i,\mathbf{f}} = \lfloor \mathbf{S}_{\mathbf{w}_i}/\mathbf{f} + 0.5 \rfloor \mathbf{f} \tag{2.7}$$

This function takes the continuous input function **X**, representing force over the surface of the sensor, and outputs a discretized version akin to the digital output of our sensors. Spatial discretization is accomplished by integrating the force profile **X**, over the area of each sensing element, representing the computation of the total resolved force acting in the measured direction. Force discretization consists of rounding the result to the nearest multiple of **f**. For simplicity, the input and output of **S** were defined to be one-dimensional, though the analysis generalizes naturally to the case of a two-dimensional grid of sensing elements. The output range was chosen to comprise a number of sensor elements sufficient to describe the response to an impulse applied at any point on one sensing element. We used the element being stimulated plus two elements on either side, for a total of five sensing elements, based on the assumption that for reasonable force resolutions, changes in the response to more distant elements would be negligible.

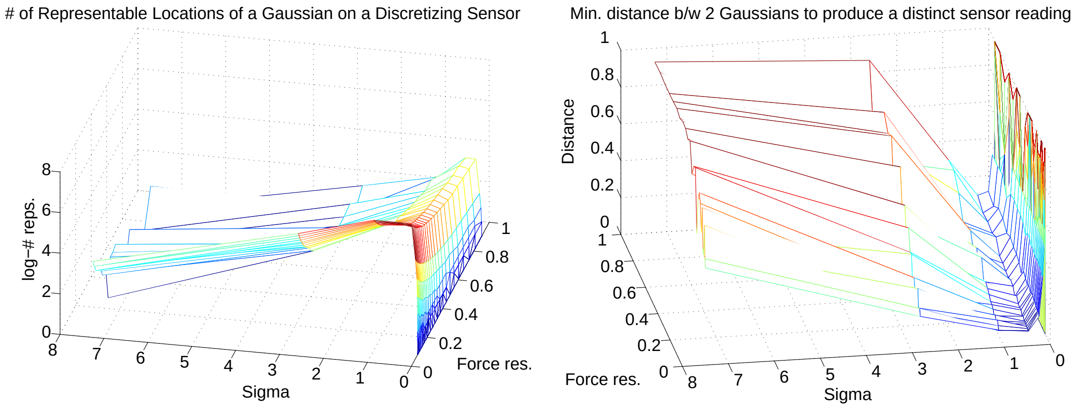
The impulse response was modeled, as described in Section 2.3, using a Gaussian function (centered at the point of application of the impulse, μ, as defined in Equation 2.2) with standard deviation, σ, increasing with the thickness of the covering.

We considered the performance on two tasks as criteria for measuring spatial acuity: spatial localization of a point force, and two point discrimination. In both cases, we searched for the optimum σ value with respect to sensor elements of width 1 unit, over different levels of force discretization.

2.4.1.1 Spatial Localization

For the spatial localization task, the goal is to identify the location at which a point force was applied as precisely as possible. That is to say, for every possible value of μ, we would like to find the σ that minimizes $\mu_e - \mu$, the average difference between the true point of application, μ, and the estimation thereof, μ_e. Rather than choosing a particular estimation procedure (such as function fitting with maximum extraction), we instead consider an upper bound on feasible estimation accuracy. Let $\mathbf{r}_{\sigma,\mathbf{f}}$ be the number of distinct representations produced by a point force applied at every possible location within a single sensor element, that is the number of different sensor readings which would result from moving $\mathbf{G}_{\mu,\sigma}$ from one end of an element to the other. This is an upper bound on performance in the following sense: If two inputs produce identical outputs, they can not possibly be differentiated, so the number of distinct outputs limits possible performance on the task, regardless of the estimation method used.

Intuitively, one would expect extreme values of σ to produce poor results: As σ approaches 0, G approaches an impulse, which excites only the sensor element it falls on and does not allow any inference about where within the element the force was applied based on the response of adjacent elements. At the other extreme, G approaches a uniform response across all sensor elements, which is even less informative. One would expect, therefore,



(a)

(b)

Figure 2.6: Spatial acuity as a function of σ, the standard deviation of the Gaussian point spread function: (a) The number of distinct representations produced by varying the location of the Gaussian over a single sensor element shows an optimum σ value around 0.3 for all force resolutions. (b) The minimum distance between two Gaussians which will produce a distinct sensor response from a single Gaussian shows an optimum σ close to 0.35, also independent of force resolution.

that an optimal value lies somewhere in between.

A Gaussian input, $\mathbf{G}_{\mu,\sigma}$ was defined with μ initialized to 0, which corresponds to being centered on the left edge of the central sensing element. μ was then increased, moving the input across the central sensing element over a total of 50 steps, for a resolution of $0.02\mathbf{w}$ units. Space and force discretization were applied, as in Equations 2.6-2.7, and the number of distinct sensor outputs was recorded for 25 values each of $\mathbf{f}$ and σ, drawn from a log scale. Figure 2.6(a) shows the (log of the) number of distinct representations produced by the procedure described above, while varying both the spread of the input and the level of discretization of the output force values. For all levels of force discretization, there is a prominent ridge in the graph around $\sigma \approx 0.3$.

2.4.1.2 Two-Point Discrimination

We used a similar approach for the two point discrimination task. In this case, two Gaussian inputs were defined and initialized to the same location, so that the result (by superposition) is a single Gaussian of twice the amplitude. The two inputs were then gradually moved apart, maintaining the same center point, but increasing a separation distance in increments again of 0.02**w** units. Again, space and force discretization were applied according to Equations 2.6-2.7, and the separation distance between the two Gaussians was incremented until the resulting discretized sensor output changed to a distinct value. This was taken as a lower bound on the distance between two such stimuli for them to be distinguished from a single stimulus of twice the magnitude, i.e. to discriminate two points from one. This procedure was repeated at 10 different locations within the sensor element (offsets from the edge in increments of 0.1) of the center of the two Gaussians, and the average minimum distance was recorded. Figure 2.6(b) shows the minimum distance between the Gaussians at which that change to a distinct sensor reading occurred, again as a function of the spread of the two Gaussians and the force discretization. Once more, an optimal value occurs near $\sigma = 0.3$ for all force discretization levels.

2.4.2 Avoiding Aliasing

Another important optimization criterion, is avoiding aliasing in the output. Distinct sensor readings are only useful insofar as the shape of the sensor response remains uniform as the input is translated across the sensor surface. The maximum frequency of spatial variation we should be able to detect without aliasing is the Nyquist rate of 2 sensor element widths.

A design criterion for avoiding aliasing would therefore be to choose a σ which attenuates frequencies above the Nyquist rate. Consider now the Fourier transform of a Gaussian PSF, $\mathscr{F}_x[\mathbf{G}_{0,\sigma}(x)] = \hat{\mathbf{G}}_{0,\sigma}(\omega) = e^{-2(\pi\sigma\omega)^2}$. If we introduce a new variable,

$$\sigma' = 1/(2\pi\sigma), \tag{2.8}$$

then this becomes $\hat{\mathbf{G}}_{0,\sigma}(\omega) = e^{-\omega^2/(2\sigma'^2)}$. The spatial Nyquist rate, in sensor element units, is $1/2$, and we desire $3\sigma' = 1/2$ so that almost all of the frequency content lies above this rate. This corresponds to a choice of $\sigma' = 1/6$. Substituting into Equation 2.8 and solving for σ yields $\sigma \approx 1.0$. Since this is a lower bound on sigma, it supersedes the value of 0.3 found above. By inverting Equation 2.5, we see that this value is below the range of what is attainable with our strips of polyurethane, but close to the $\sigma = 1.36$ given by our thinnest available covering, which was used for the tests in later chapters.

2.4.3 Other Considerations

Another particularly important consideration when choosing the thickness of the covering is that it determines the range of displacements the sensor is capable of sensing, let alone discriminating. i.e., a sensor equipped with a 0.04” covering that is pressed against a surface with ridges and valleys 0.1” in depth will not sense those valleys at all, let alone any texture they may contain. Thicker coverings (of a softer material) also allow the same range of forces to map to a larger range of displacements; put another way, they allow a greater range of displacements to be discriminated with a given force resolution. With the information from Section 2.3, one can see that there is a trade-off between one's ability to sense and discriminate displacements applied to the covering and the ability to localize the source of individual individual inputs. The optimal covering will therefore depend on one's relative prioritization of these goals for the task at hand. As is shown later in Section 4.4.2.5, in the case of object recognition, the benefits of a thicker covering seem to far outweigh its drawbacks.

2.5 Simulation

Based on the tests of Sections 2.2 and 2.3, we derived an imaging model of tactile sensing appropriate for machine simulation of the sensing process, illustrated in Figure 2.7. Our formulation of the simulator is described in Section 2.5.1, followed by a comparison of its outputs to those of the real sensor in Section 2.5.2. Our implementation of this model, the Manipulating and Perceiving Simultaneously project's Tactile Force Sensor Simulator (MAPS-TFSS), has been released under the Gnu Public Licence Version 3 (GPL-v3) and is available for download from our website.

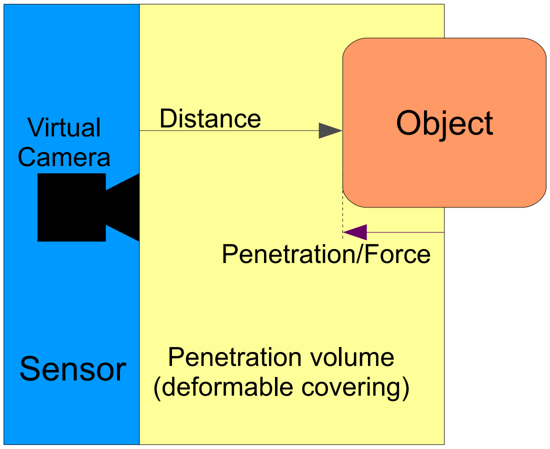


Figure 2.7: Simulation sensing model: a virtual camera detects distances to objects penetrating the sensor covering, then shaders convert these measurements to force readings.

2.5.1 Simulator Formulation

The PSF-based modeling of tactile sensors lends itself to straightforward computational simulation. To simulate the response of the sensor, we begin with a simple deformation model. The sensor and the material covering it are viewed as a unit, and the sensor is considered to be a position-force transducer, where penetrations of the covering material are considered the input displacements.

As shown in Figure 2.7, there are two planes of interest for determining the sensor response:

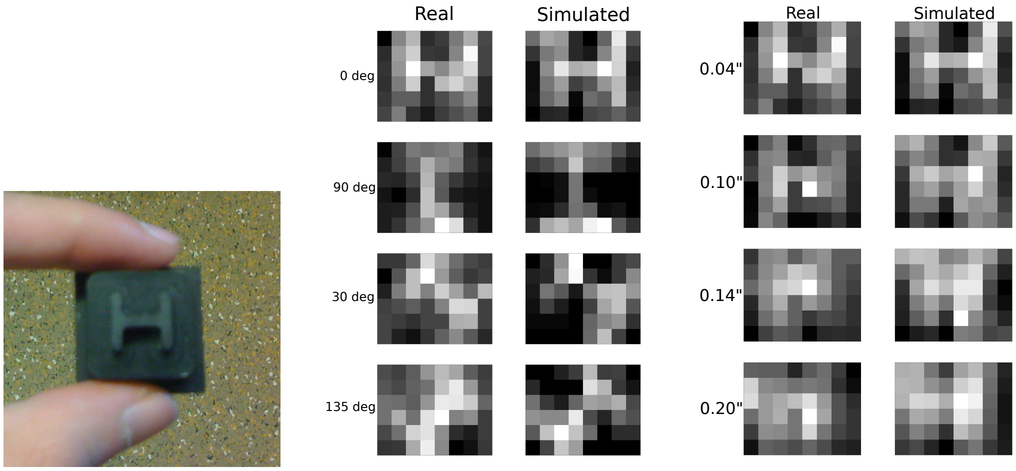
- the surface of the sensing elements
- the surface of the deformable material covering them.

An object is considered to be in contact with the sensor as soon as it collides with the deformable material. The object is allowed to penetrate this covering (think of it as displacing the surface), and the penetration distance above each sensing element is then calculated. This process can be implemented efficiently using standard graphical rendering techniques. The sensor is modeled as a camera under orthographic projection, and with the viewing volume defined by the deformable surface material. Under this model, the z-buffer of the rendering pipeline conveniently provides the depth of penetration for each sensor element.

The force response function is then modeled in two steps:

- First, a Gaussian blur is applied to the computed penetration distances (rendered at higher resolution than the final sensor response), to simulate the point spread function associated with the covering material.
- A nonlinear function obtained in calibration then maps the resulting displacements to forces measured by the sensor.

In our simulation, we have implemented these operations using OpenGL and GLSL shaders whose parameters were determined by the experiments described in Sections 2.3 and 2.2, respectively. As a result, we are able to achieve extremely rapid hardware rendering of tactile sensor images.



(a) H Indenter

(b) Oriented H samples

(c) Thickness comparison

Figure 2.8: Comparison of real and simulated images of an H-shaped indenter. (a) The H-shaped indenter. (b) The H at several angles, all using the thinnest covering. Angles are relative to vertical. (c) The H in the same vertical orientation with each of the 4 thicknesses shows the effect of the increasing point spread. All images are scaled to the range zero to one for readability.

2.5.2 Comparison of Simulator and Real Sensor Response

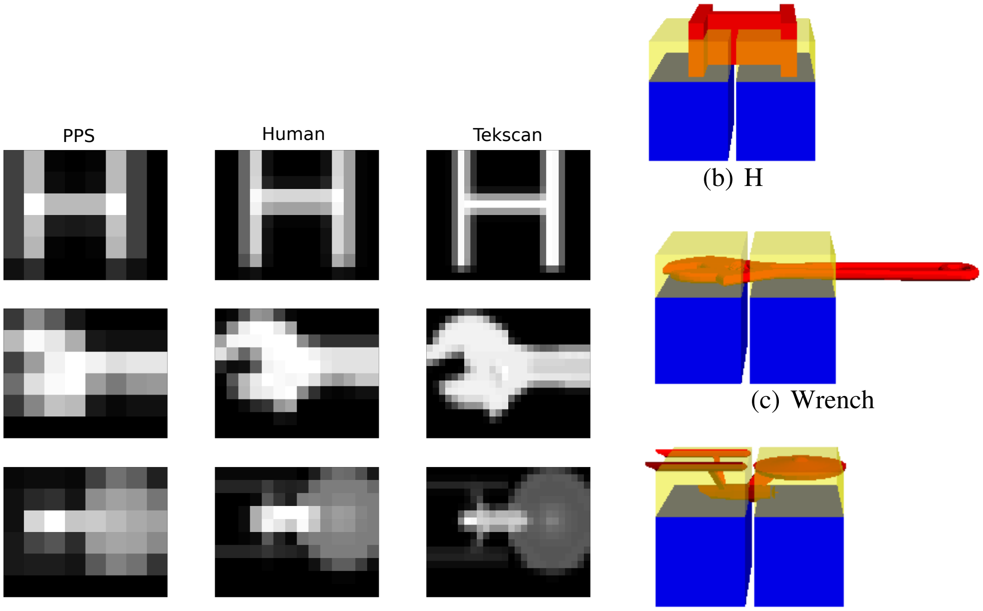
A comparison between the modeled response and the response of the actual sensor is depicted in Figure 2.8. An H-shaped indenter (Figure 2.8(a)) was applied to the sensors and a 3D model of the indenter was applied to simulated sensors in the same configuration (Figure 2.9(b)). The outputs of each are shown side-by-side, normalized (each independently) for readability. Figure 2.8(b) shows images that resulted when the indenter is applied near the center of the sensor in several different orientations, using the thinnest covering (with the smallest point-spread), and Figure 2.8(c) shows images of the indenter in the same vertical orientation, but using different covering thicknesses. Note the visible defocusing of

Orientation	0°	90°	30°	135°
	0.929	0.878	0.827	0.872
Thickness	0.04 in.	0.1 in.	0.14 in.	0.2 in.
	0.929	0.939	0.934	0.958

Table 2.1: Correlations between real and simulated tactile images

the image as thickness increases, both in the real and simulated images. Table 2.1 gives the correlations between the normalized real and simulated tactile images for each of the pairs shows in Figure 2.8, computed as in Equation 2.3.

Figure 2.9 illustrates the extensibility of the simulator and the effect of increasing the resolution of a tactile sensor. Here we compare, in simulation, the resolution of the sensors characterized in this work (PPS) with that of the human fingertip and that of another tactile sensor system, model 5027 available from Tekscan, Inc. TekScan (2011). The PPS sensors are shown as before, as an 8×6 arrangement of sensing elements, with a density of 25 sensing elements per cm^2. The other two systems are simulated as sensors with each of their characteristic resolutions adapted to a similar aspect ratio. The sensors in human skin that most closely resemble ours are Merkel receptors. They are responsible for form and texture perception and have a spatial resolution of about 1 mm (Iggo and Muir, 1969; Phillips and Johnson, 1981). The innervation of Merkel receptors in the skin of the human fingertip is estimated at 70 sensors per cm^2 (Vallbo and Johansson, 1984), so human sensation is



(a) Simulator output for each resolution

(b) H

(c) Wrench

(d) Ship

Figure 2.9: (a) Comparison of simulation of different-resolution sensors depressed by complex objects. Row one shows the H indenter, row two a wrench, and row three a miniature starship. Column one corresponds to the resolution of the PPS sensors, column two to that of the human finger, and column three to that of the Tekscan sensors. (b-d) External views of the H, wrench and starship inputs. In each, the blue blocks represent the sensors. The translucent yellow volume is their covering, which is being penetrated by the red model of each object.

simulated with a resolution of 20×14. Finally, the Tekscan system features 248 sensing elements per cm^2, and is represented by a resolution of 36×26. All three have a point spread with a σ of one sensor element width, and none have the per-element sensitivity adjustments applied in the previous simulations of the PPS sensors.

2.6 Discussion

In this chapter, we have presented an imaging model for tactile force sensors and a simulation method that realizes this method to enable real-time rendering of tactile sensor responses. This model was used to provide some guidelines in the choice of an appropriate covering, and the simulator was demonstrated to produce responses very similar to those of the real sensor system on practical inputs. A few points merit further discussion, regarding the applicability of the model and the remaining inconsistencies with the real system.

This model of the sensor and associated simulator was developed for the purpose of analyzing the performance of the recognition algorithms presented in the remainder of this dissertation in simulations of the interaction between the sensor and arbitrary 3D objects. To that end, it focuses on static contact situations and emphasizes efficiency over accuracy. As a result, effects such as shear forces, hysteresis, and viscoelastic properties of the covering material remain un-modeled. Applications where these effects play a large part would be better suited to a different model, but it provides a powerful tool for analyzing the types of interactions investigated in this work.

2.6.1 Range of Linearity

A system consisting of tactile sensors embedded beneath a compliant covering was demonstrated to be well-approximated by a point-spread function imaging model. The point spread approximation seems to work well in the linear range of the sensor-covering system; for very large deformations, however, the response begins to grow nonlinearly as the covering undergoes elastic hardening. Such large deformations are not encountered in typical interactions with our system. Further, the stiffness of the material used for the covering can be selected to maximize the linear range of response for the interaction forces expected. For applications that require such large deformations, however, a similar characterization would be necessary that included the expected range of displacements to determine if the point spread approximation needed to be modified.

2.6.2 Removable Covering

Having a removable covering was essential to our ability to analyze the effects of different coverings, but it also presents significant challenges. When the covering is not permanently affixed to the sensors, the response characteristics of the sensor-covering system have a much greater potential to change over time. Viscoelastic effects associated with the covering material are always present, even on permanently-affixed coverings. With a non-permanent covering, air gaps between the sensors and covering may shift much more easily, making and breaking contact between the two materials as a result of applied normal forces or shear forces. These shifts of the covering with respect to the sensors can cause significant unexpected changes in the sensor response that are very difficult to model, and we suspect that most of the remaining discrepancies between the true and simulated sensor responses are due to these effects. A sensor with a permanently-affixed covering is likely to be better-behaved.

2.6.3 Applications

As demonstrated in Section 2.5, our simulation framework can be used to simulate the response of any PSF-modeled tactile sensor. This offers opportunities for a variety of applications:

Replication Once a particular sensor has been characterized, any number of identical sensors may be replicated in simulation by simply copying the characterization parameters. The characteristics of a suite of sensors can then be analyzed for prototyping purposes without needing to have the full physical system.

Avoiding Hardware Requirements Once someone has characterized a given sensor and made the relevant parameters available, other researchers can freely simulate that sensor without needing to invest in the physical system. This capability can allow researchers to investigate algorithms that make use of cost-prohibitive sensor systems and determine which are worth validating on a real system.

Parameterization Algorithms that make use of a particular type of tactile sensor can be tested with different parameterizations of that sensor, e.g. to determine requirements on resolution or covering material. We make particular use of this in Chapter 4.

Predicting Future Capabilities The manufacturing and/or economic limitations of the current generation of tactile sensors need not apply in simulation. Therefore, arbitrarily accurate tactile sensors could be created in simulation, allowing the investigation of the effects of spatial and force resolution on tactile image processing, tactile object recognition, and tactile-information-based manipulation algorithms for future generations of hardware.

This model was used extensively in the chapters that follow, and we take advantage of several of these opportunities in the remainder of this dissertation.

Chapter 3

Geometry-Based Tactile Recognition

3.1 Introduction

As described in Chapter 1, the recognition task consists of two parts:

Training: We present the robot with a set of objects we would like it to be able to recognize, and it is allowed to explore those objects extensively to build up its model for recognition.

Testing: Then an unknown object is placed in the robot workspace in some unknown pose. The robot is allowed to take a series of tactile sensor readings of the object, and at any time it must be able to provide an estimate of the identity of the object.

An intuitive approach to take in performing tactile recognition is to build up a geometric model of the surfaces of the objects to be recognized and then to compare the observed set

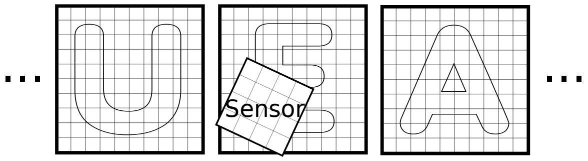


Figure 3.1: Illustration of recognition as a problem of localization within a map. The position of the sensor within the map tells you the pose of the object, and which of the maps the sensor is localized within gives the object identity.

of tactile sensor reading to the geometric models in order to determine which of the models could have produced the readings. [1]

3.1.1 Recognition as Mapping and Localization

Our approach to geometry-based recognition is rooted in the mobile robotics literature. Consider each of the objects to be recognized as being analogous to an environment for mobile robots to explore, where in this case the mobile robots are tactile sensors (on the end-effectors of some larger robot). Then the training phase can be thought of as an opportunity to build an explicit map of each object. Correspondingly, once you have a map of each object, the recognition task during testing becomes one of finding the location of the sensor in one of these maps. Then the identity of the unknown object is given by which of the maps that location lies within, as illustrated in Figure 3.1.

[1] This chapter covers work presented in Pezzementi et al. (2011b). This work was carried out with the assistance of Caitlin Reyda during her Research Experience for Undergraduates residence.

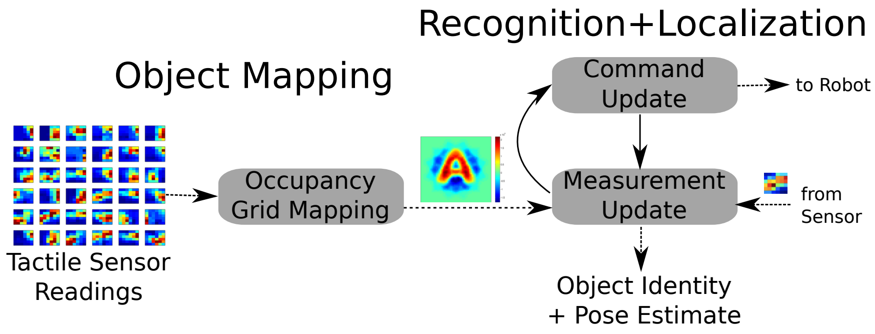


Figure 3.2: Data-flow for geometry-based recognition process

3.1.1.1 Mapping at a high level

We use an occupancy grid map (Elfes, 1987; Moravec, 1988), popular in mobile robotics, as our geometric representation of the objects to be recognized. With this technique, the world (object and its immediate surroundings) is discretized into a set of cells arranged in a grid, and we estimate the probability that each cell is either full (occupied) or empty. Specifics of how we build these maps are discussed in Section 3.2. Such maps allow probabilistic reasoning about what sensor readings should be expected in what poses. If desired, they can also be converted to alternative surface representations for use in other applications.

3.1.1.2 Localization at a high level

We follow the convention of other authors in the field (e.g. Thrun et al., 2005) of breaking the exploration process into a series of discrete time steps, where during each time step t a **control** u_t input is sent to a robot (for example, to press the sensor down in a particular location), and a **measurement** z_t is observed as a result of that action. In our case, the measurement consists of a tactile sensor reading along with a position and orientation of the sensor in the robot coordinate system, and the time step is defined by the period required for the robot to control the tactile sensor into stable contact with the object surface. In this format, Bayes filters are readily applicable. Let us refer to the **state** we are trying to estimate, which contains the object identity and its position and orientation, as x_t, to the set of all states, control inputs, and measurements up to time t as $x_{1:t}$, $u_{1:t}$ and $z_{1:t}$ respectively, and to the information in our maps as $\mathbf{M}$. Then Bayes filters allow estimation of $\Pr(x_t \mid u_{1:t}, z_{1:t}, \mathbf{M})$ at each time step, requiring only that we provide ways of estimating two quantities:

Command Update – $\Pr(x_t \mid x_{t-1}, u_t)$, the probability of how state changes with a given control input, in our case defined by the kinematics of the robot.

Measurement Update – $\Pr(z_t \mid x_t, \mathbf{M})$, the probability of observing the current sensor reading in a hypothesized state, x_t. This will be evaluated for many possible state hypotheses.

The entire work-flow is illustrated in Figure 3.2. By iterating through command and measurement updates at each time step, we continually refine a probability distribution that starts with an uninformed prior and (hopefully) converges upon the true solution. More details, including our methods for estimating the two quantities above, follow in Section 3.3.

This approach has been widely used for mobile robot localization (Thrun et al., 2005; Dellaert et al., 1999; Fox et al., 1999; Simmons and Koenig, 1995). Previous authors have used similar techniques to address object localization using force sensing, but these all assumed the object identity and geometry were known (Gadeyne and Bruyninckx, 2001; Chhatpar and Branicky, 2005; Petrovskaya et al., 2007; Platt et al., 2010). In Platt et al. (2010), the object being manipulated is a flexible material with an embedded hard object (such as a grommet).

The experiments conducted in this chapter all deal with 2D objects (that is, mapping and recognition of a single face of a 3D object). Although all of the techniques presented generalize to 3D objects as well, there are computational challenges in the representation of probabilities in the space of possible poses in the localization portion of the algorithm as the dimension of this space rises to six. Dealing with these computational issues to enable the technique to be fully applicable to 3D objects forms some of the primary motivation for the work in Chapters 4 and 5.

3.2 Tactile Map Building

In order to perform geometric object recognition, we first need a geometric model of each of the objects to be recognized. Such a model would be constructed during a training session by freely exploring the object. In this chapter, the information gathered during this session consists of tactile images and the locations, relative to some fixed coordinate system on the object, at which the images were gathered. Following the notation established in Section 3.1.1.2, we view this as a time-series process that provides, at each time step, a tactile sensor measurement, z_t, and an estimate of the location (state), x_t, at which that measurement was collected.

3.2.1 Map Representation

Given a series of measurements and locations, it is possible to build a *tactile mosaic* that represents, in the force domain, the surface of the object being explored. We use an occupancy grid mapping technique (Elfes, 1987; Moravec, 1988) to convert such a collection of tactile force sensor readings into an object-specific *map* that represents object geometry. More specifically, a map, $\mathbf{M}_j$, of object j is generated by dividing the workspace into a set of cells, $\{\mathbf{m}_i\}$, each of which will contain an estimate of the probability that the corresponding location is filled (i.e. generates a force reading) or vacant. We therefore wish to estimate, for each location, the probability $\Pr(\mathbf{m}_i \mid z_{1:t}, x_{1:t})$.

The map estimates are stored in log-odds form,

$$l_{t,i} = \log \frac{\Pr(\mathbf{m}_i \mid z_{1:t}, x_{1:t})}{1 - \Pr(\mathbf{m}_i \mid z_{1:t}, x_{1:t})} \tag{3.1}$$

for numerical stability, since this maps the normal range of probabilities from $[0,1]$ to $[-\infty,\infty]$ as shown in Figure 3.3. We follow the standard method (Thrun et al., 2005) of defining a measurement model to directly estimate $l_{t,i}$ from $l_{t-1,i}$:

$$l_{t,i} = l_{t-1,i} + \log \frac{\Pr(\mathbf{m}_i \mid z_t, x_t)}{1 - \Pr(\mathbf{m}_i \mid z_t, x_t)} - \log \frac{\Pr(\mathbf{m}_i)}{1 - \Pr(\mathbf{m}_i)} \tag{3.2}$$

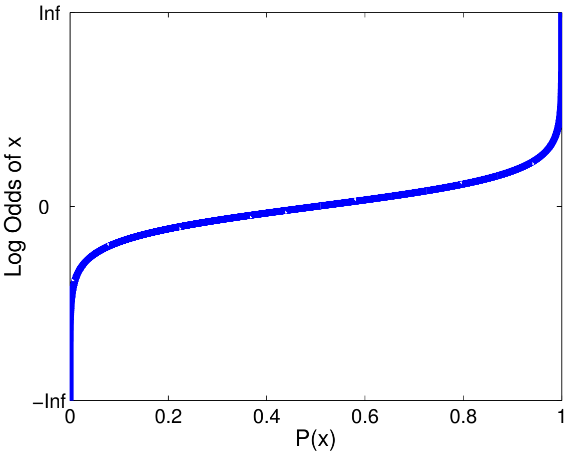


Figure 3.3: Log-odds form maps occupancy probabilities to the range $[-\infty,\infty]$

$\Pr(\mathbf{m}_i)$ is taken as the prior for occupancy of a grid cell independent of its location, which can be estimated from the average object surface area and the area being modeled. The second term on the right hand side of (3.2) is referred to as the "inverse sensor model", since it inverts the normal forward model of measurement formation to infer map properties, and it must be estimated for a particular sensor system or learned from data. A model for our sensor system is derived below.

3.2.2 Inverse Sensor Model

In order to convert the image from the force to the contact domain, we chose to binarize inputs rather than use the continuous readings. This conversion adds robustness to effects such as sensor noise, non-uniformities in the sensor response, and inconsistencies in the interaction with the sensor covering. Let the individual sensor elements of reading z_t be

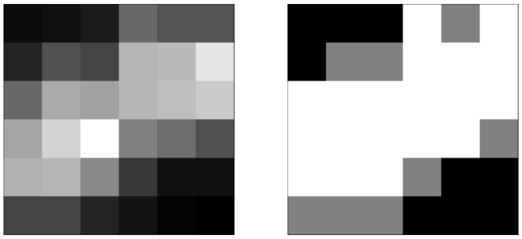


Figure 3.4: Illustration of the contact detection process of Equation 3.3. On the left is a sample input tactile image, with gray-scale intensity corresponding to force magnitude. The contact detection result is shown on the right, with detected contacts shown in white, "no contact" detections in black, and "unknown" elements in gray.

denoted $e_{t,j}$ for $j = \{1...n_e\}$. To determine whether element $e_{t,j}$ has detected contact or no contact with the object, the following classification function was found to work well in practice:

$$c(e_{t,j}) = \begin{cases} \textsc{InContact} & \text{if } e_{t,j} > T \text{ AND } e_{t,j} > 0.3e_{max} \\ \textsc{NoContact} & \text{if } e_{t,j} < T \text{ AND } e_{t,j} < 0.3e_{max} \\ \textsc{Unknown} & \text{otherwise} \end{cases} \qquad (3.3)$$

with $e_{max} = \max_j e_{t,j}$ and T equal to about 2 kPa for our sensors. The dependence on e_{max} makes the classification robust to changes in the applied force, due to the possibility of forces on elements adjacent to the point of contact because of the point spread effect, while the T term is intended to avoid false detections due to sensor noise. The result on a typical sensor reading is illustrated in Figure 3.4. Informal evaluation showed this classifier produced the correct classification about 90% of the time and a random value the remaining 10% of the time. This performance was found to be sufficient for our purposes.

The response of these sensors was shown in Section 2.3 to be well-approximated by a displacement map convolved with a symmetric Gaussian point spread function, $\mathbf{G}_{0,\sigma}$.

The inverse of this measurement model, $\mathbf{G}'_\sigma$, i.e. the distribution of object locations which might produce a contact reading (or the empty locations which may produce "no contact" readings) was approximated by a Gaussian with half the standard deviation of the point spread function. This value was used as an estimate of the locations where the force response would be more than 30% of the maximum value.

Let $\overline{\mathbf{m}_i}$ denote the location of the center of mass of the map cell $\mathbf{m}_i$ and $\overline{e_{j,t}}$ denote the location of the center of sensor element e_j when the sensor is at the state x_t. We now view each a map cell, $\mathbf{m}_i$, as a binary random variable, and define the probability of occupancy as follows:

$$\mathbf{G}'_\sigma(\mathbf{m}_i; e_{t,j}, x_t) = \frac{1}{\sqrt{\pi\sigma^2/4}} \exp\left(-\frac{\|\overline{\mathbf{m}_i} - \overline{e_{t,j}}\|^2}{8\sigma^2}\right) \tag{3.4}$$

$$\begin{aligned}&\Pr(\mathbf{m}_i | e_{t,j}, x_t) = \Pr(\mathbf{m}_i) + \\ &\begin{cases} (1 - \Pr(\mathbf{m}_i))\,\mathbf{G}'_\sigma(\mathbf{m}_i; e_{t,j}, x_t) & c(e_{t,j}) = \textsc{InContact} \\ -\Pr(\mathbf{m}_i)\mathbf{G}'_\sigma(\mathbf{m}_i; e_{t,j}, x_t) & c(e_{t,j}) = \textsc{NoContact} \\ 0 & c(e_{t,j}) = \textsc{Unknown} \end{cases}\end{aligned} \tag{3.5}$$

The effect of this rule is illustrated in Figure 3.5. Measurements of InContact in the vicinity of a map cell increase its probability above the occupancy prior, while those of NoContact decrease it below the prior and those of Unknown have no effect.

Distinct measurements are assumed to be independent[2], making map updates order-

[2] This is a standard assumption in occupancy grid mapping algorithms since Elfes (1987). Although methods that do not depend on this assumption have been developed (for example, Thrun, 2003), they are considerably more computationally demanding, and the structure of our measurement collection process should not result in a strong enough violation of the assumption to expect significant differences in the resulting maps.

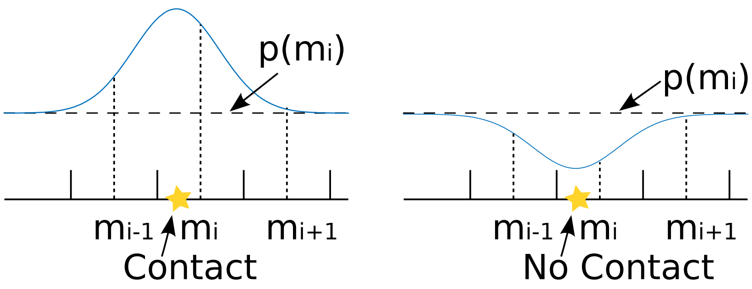


Figure 3.5: Illustration of the occupancy inference rule of Equation 3.5.

independent. Once the probabilities in the measurement model are converted to log odds form, map updates are simply additive:

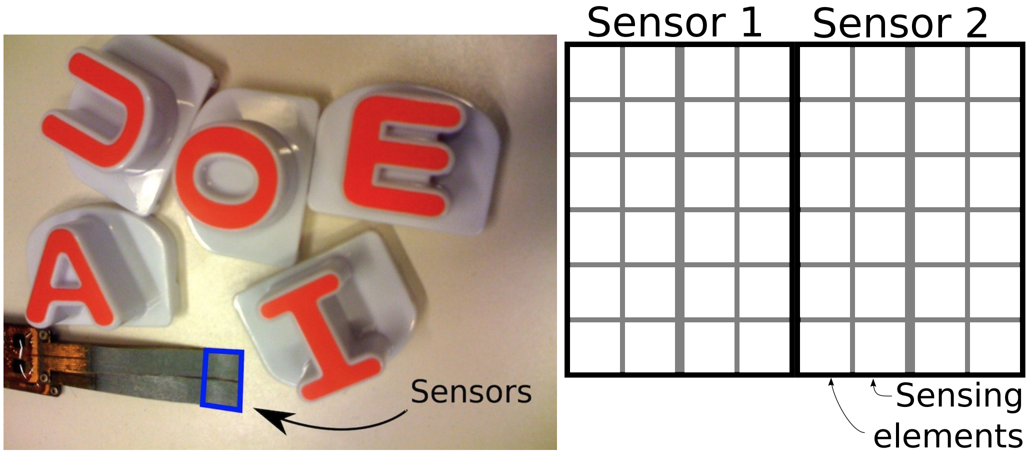
$$l_{t,i} = l_{t-1,i} + \sum_{j=1}^{n_e} \log \frac{\Pr(\mathbf{m}_i \mid e_{t,j}, x_t)}{1 - \Pr(\mathbf{m}_i \mid e_{t,j}, x_t)} - \log n_e \frac{\Pr(\mathbf{m}_i)}{1 - \Pr(\mathbf{m}_i)} \tag{3.6}$$

Since the map is built off-line and the updates are order-independent, a grid of arbitrary resolution can be efficiently built in parallel on a cell-by-cell basis.

3.2.3 Sensor and Object Setup

We illustrate this method in 2D on capital vowels from a child's set of raised letters (from a "Fridge phonics" magnetic alphabet set, LeapFrog Enterprises, Inc., Emeryville, CA, USA), shown in Figure 3.6 along with our sensors. See Section 2.1.1 for detailed information on the sensor system.

The letters were approximately 2.5 cm per side, so less than a quarter of the letter was visible in any single reading. In order to cover the entire object, readings were collected at 16 planar positions arranged in a 4-x-4 grid with a spacing of 6.8 mm and rotated in the



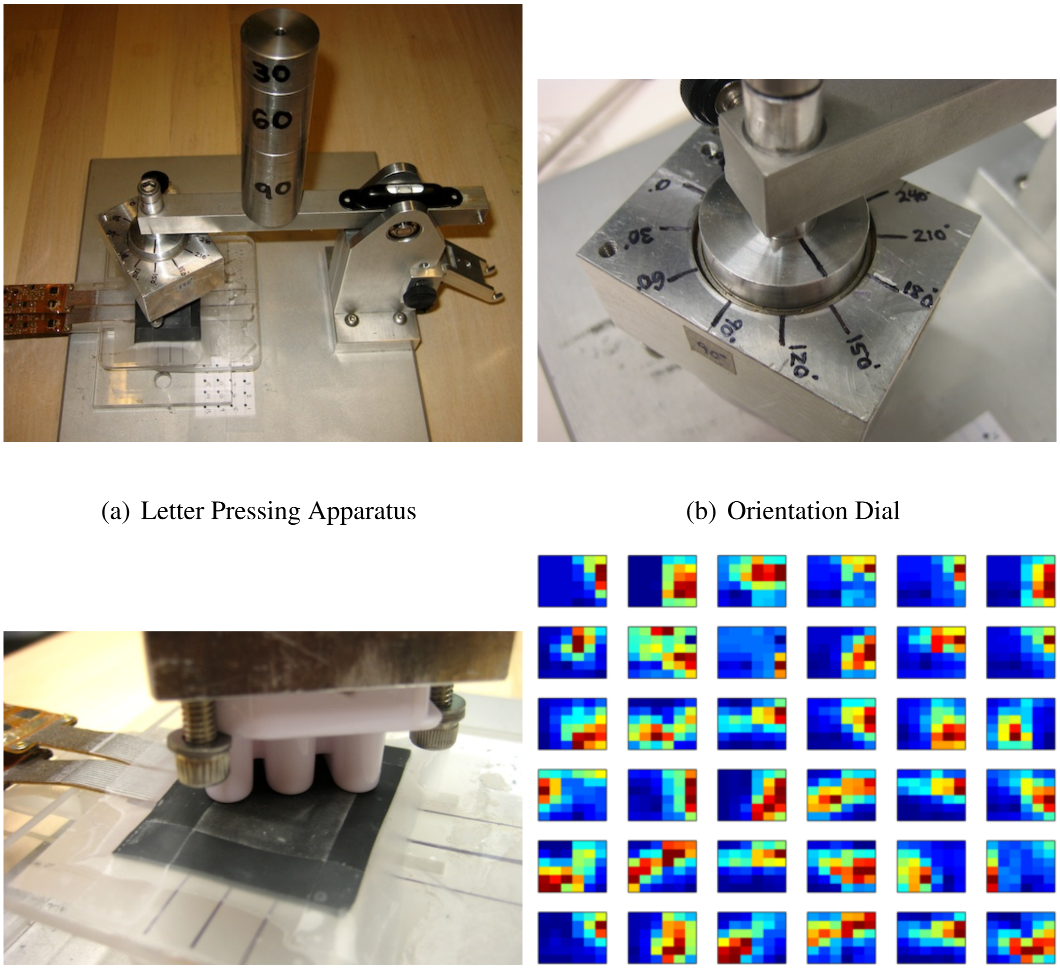
(a) Letters and Sensors

(b) Sensor Layout

Figure 3.6: (a) shows the set of raised letters used in the geometry experiments alongside the PPS DigiTacts sensors, with the sensing area highlighted in blue. (b) shows the layout of sensor elements within that highlighted area.

plane at 12 evenly-spaced angles at each location for a total of 192 readings. A mechanical system was constructed to position the letters coplanar with the sensors and press them down with a consistent force (Figure 3.7). Sample tactile readings collected from the “A” model with this system are shown in Fig. 3.7(d). Due to the structured nature of the data collection, we assumed no error in the state estimates during training, $x_{1:t}$, associated with the measurements $z_{1:t}$.

Visualizations of the resulting occupancy grids of these letters are shown in Fig. 3.8.



(a) Letter Pressing Apparatus

(b) Orientation Dial

(c) Letter-Sensor Contact

(d) Example Tactile Images

Figure 3.7: (a-c) show the apparatus used for pressing letters against the sensors during data collection. (d) shows some sample images acquired with this apparatus from the “A” object. Color varies with force intensity, with “hot” colors corresponding to large forces.

3.3 Recognition and Localization

We view the recognition task as a sequential state estimation problem, where, at each time step, t, we are given a measurement, z_t, and we wish to estimate the robot state, x_t, at which that measurement was taken. That state consists of the identity of the object, C,

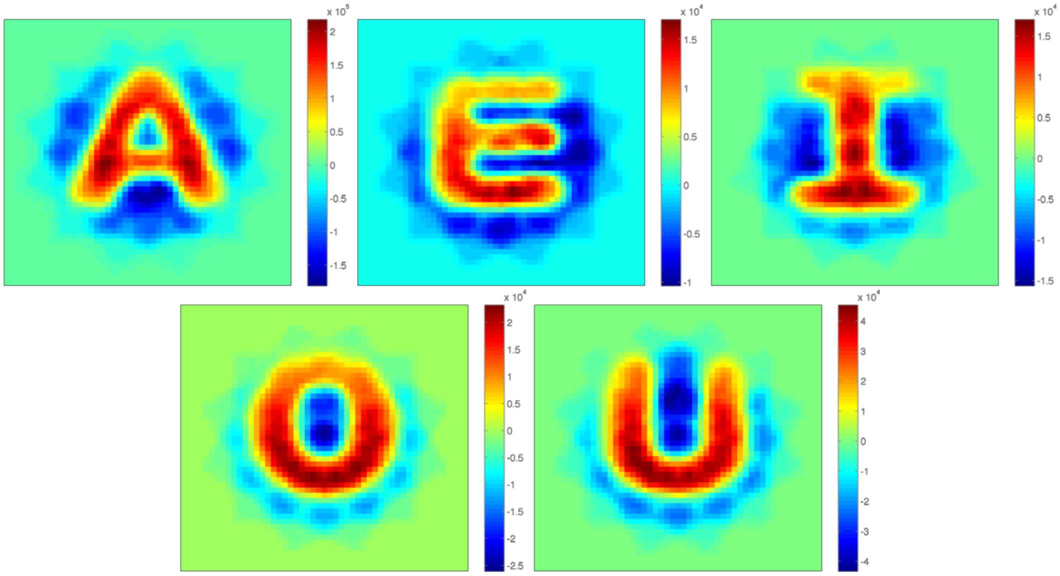


Figure 3.8: Reconstruction of letters from tactile images. Pixels are colored by probability of occupancy in log-odds form. This probability can be seen to be high (denoted by "hot" colors) on the letters themselves, to be low ("cool" colors) in the surrounding area where measurements were taken, and to decay to the (greenish) prior probability in the surrounding un-sensed area where no sensor readings were taken.

along with the pose of the sensor in the object coordinate frame, $\mathbf{T}_S^O$. We assume that we also have, for each measurement, an estimate of the pose of the sensor in the robot frame, $\mathbf{T}_S^R$, from the robot forward kinematics. These frames are illustrated in Fig. 3.9. Assuming both the robot base coordinate system and the object remain fixed in the world, the problem reduces to estimating the (constant) transformation between robot and object frames, $\mathbf{T}_R^O$ since we can recover $\mathbf{T}_S^O$ as $\mathbf{T}_R^O\mathbf{T}_S^R$. For our manual sampling method, $\mathbf{T}_R^O$ was defined to be the identity transformation at map building time—i.e. the object frame is simply the robot frame. In the two-dimensional case, $\mathbf{T}_S^O$ can be decomposed into two translation components and an angle, giving $x = [C, \mathbf{t}_x, \mathbf{t}_y, \theta]$.

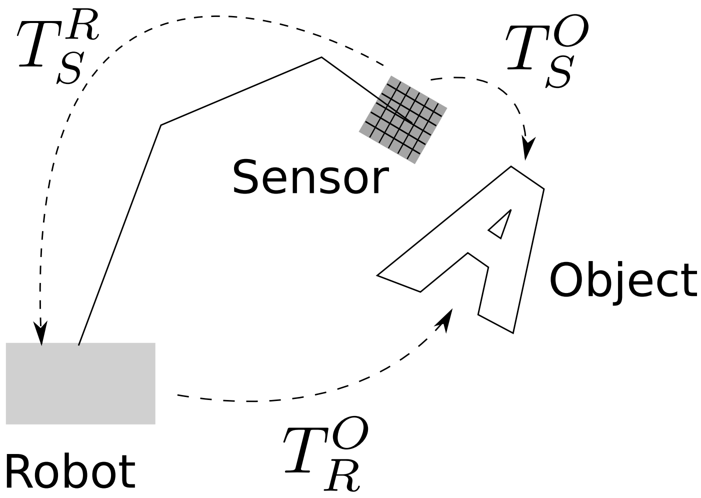


Figure 3.9: Reference frames used in tactile exploration of the unknown object

3.3.1 Non-Parametric Bayes Filters

There are a variety of specializations of the Bayes filter available that make different approximations in the estimation of x_t. The best-known of these is the Kalman filter (Swerling, 1958; Kalman, 1960) and its variants. It is unsuitable for our task, however, because our probability distributions are highly non-Gaussian (generally starting as uniform) and usually multi-modal. Non-parametric estimation methods were chosen because of their strength at representing multi-modal distributions and the relative ease of dealing with combined discrete and continuous state with these methods. The task can then be viewed as an application of either the classic histogram filter (Simmons and Koenig, 1995) or the more widely known particle filter (Dellaert et al., 1999; Fox et al., 1999) localization techniques from mobile robotics. Each has its relative theoretical advantages, so we present both methods for comparison.

Following the notation of (Thrun et al., 2005) again, both filters provide ways of estimating a probability distribution, $\Pr(x_t \mid u_{1:t}, z_{1:t}, \mathbf{M})$ over state at the current time, t, given a series of commands, $u_{1:t}$, sent to the robot and measurements, $z_{1:t}$, received from the sensors and the maps learned in Sec. 3.2.

In the histogram filter, the distribution is estimated using a multi-dimensional histogram. The state space is decomposed into n_H regions, and each histogram bin corresponds to one of these regions, $\mathbf{x}_k$, with $k \in \{1..n_H\}$. The value in that bin, $p_{k,t}$, represents the probability of the state lying within the corresponding region. The particle filter is a Monte-Carlo method, where the distribution is modeled by a set of n_M samples called particles, written as $\mathscr{X}_t = \{x_t^{[1]}..x_t^{[n_M]}\}$.

In both cases, at each time step, we must perform a command update and a measurement update, as illustrated in Fig. 3.2. The command update requires estimation of $\Pr(x_t \mid x_{t-1}, u_t)$ for a control input u_t. The control input only affects $\mathbf{T}_S^R$, which is known exactly. Since neither the object identity nor $\mathbf{T}_R^O$ is changing from one time step to the next, the command update only requires updating the robot kinematics estimates. It is in the measurement update, therefore, that the distribution is reshaped.

3.3.2 Estimating Measurement Likelihood

Both particle filtering and histogram filtering implement approximations to a Bayesian filter (Thrun et al., 2005), and thus both require a measurement likelihood model, $\Pr(z_t \mid x_t, \mathbf{M})$. This can be computed from the measurement model mentioned in Section 3.2. For the sensor system used in our experiments, each reading consists of 36 individual element responses (the center 6×6 region of the sensor), each of which may detect INCONTACT, NOCONTACT, or UNKNOWN, according to the classifier of (3.3). The expected measurement of each sensor element $e_{t,j}$ is computed by interpolation of the occupancy grid built in Section 3.2. The location of the element in the sensor frame, $pos(e_{t,j})^S$, is transformed to the object frame using the hypothesized object pose to get $pos(e_{t,j})^O = \mathbf{T}_S^O pos(e_{t,j})^S$. Then the occupancy grid is queried for a log-odds value at the location $pos(e_{t,j})^O$ through bilinear interpolation of the grid point estimates to give $\Pr(occ_{t,j} \mid x_t, \mathbf{M})$, where $occ_{t,j}$ represents the occupancy of the grid cell at the location of sensing element j at time t. The element measurements are assumed to be independent, so the overall measurement probability is then taken as the product of the individual element probabilities: $\Pr(z_t \mid x_t, \mathbf{M}) = \prod_j \Pr(e_{t,j} \mid x_t, \mathbf{M})$. Then the individual element likelihoods are estimated from the classification likelihoods as $\Pr(e_{t,j}) \mid x_t, \mathbf{M}) = \Pr(c(e_{t,j}) \mid x_t, \mathbf{M})$ with

$$\Pr(c(e_{t,j}) \mid x_t, \mathbf{M}) = \Pr(c(e_{t,j}), occ_{t,j} \mid x_t, \mathbf{M}) + \Pr(c(e_{t,j}), \neg occ_{t,j} \mid x_t, \mathbf{M}) \tag{3.7}$$

$$= \Pr(c(e_{t,j}) \mid x_t, \mathbf{M}, occ_{t,j}) \Pr(occ_{t,j} \mid x_t, \mathbf{M}) + \Pr(c(e_{t,j}) \mid x_t, \mathbf{M}, \neg occ_{t,j}) \Pr(\neg occ_{t,j} \mid x_t, \mathbf{M}) \tag{3.8}$$

3.3.3 Particle Filter Estimation

For the particle filter, the command update consists simply of copying over the particles from the previous time step, with the components of $\mathbf{T}_R^O$ corrupted by a small amount of

Gaussian-distributed noise. One could view this as modeling uncertainty in $\mathbf{T}_S^R$ or small changes in the pose that might occur from one time step to the next, but, in either case, this injection of randomness is necessary to ensure the particles cover the space, even if the object is completely stationary (Dellaert et al., 1999). Let the components of particle m be $x_t^{[m]} = [C_t^{[m]}, \mathbf{t}_{x,t}^{[m]}, \mathbf{t}_{y,t}^{[m]}, \theta_t^{[m]}]$. To form the updated particle, $\bar{x}_t^{[m]}$, the identity component is copied over without modification. In our experiments, the other components were corrupted with noise sampled from $N(0,\sigma)$, with $\sigma = 1$ mm for $\mathbf{t}_{x,t}^{[m]}$ and $\mathbf{t}_{y,t}^{[m]}$ and $\sigma = 0.1$ radian for $\theta_t^{[m]}$.

During the measurement update, each particle is assigned an importance weight, $w_t^{[m]} = \Pr(z_t \mid \bar{x}_t^{[m]})$, which is evaluated as described above in Section 3.3.2. Then, the particles for the next time step are generated via importance sampling by drawing n_M new particles from $\{\bar{x}_t^{[m]}\}$ with replacement, where particle m is drawn with probability proportional to $w_t^{[m]}$.

At each time step, the current object identity estimate, $\hat{C}$, is taken as the model hypothesized by the most particles, weighted by their importance. Let $\mathrm{Ind}(A,B)$ be an indicator function defined

$$\mathrm{Ind}(A,B) = \begin{cases} 1 & \text{if } A = B \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

Then the current identity estimate is given by

$$\hat{C} = \arg\max_C \sum_{m=1}^{n_M} w_t^{[m]} \mathrm{Ind}(C_t^{[m]}, C) \tag{3.10}$$

The current state estimate, $\hat{x}$, is obtained from the estimated mode of the posterior distribution as the pose corresponding to the maximum value of a kernel density estimate over the particles, weighted by their importance. Using a conical kernel to estimate the density, we take a Parzen window (Parzen, 1962) estimate of the mode as

$$cone(d) = \max\left(0, 1 - \frac{d}{\rho}\right) \tag{3.11}$$

$$Parzen(x) = \sum_{m=1}^{n_M} w_t^{[m]} cone(x - \bar{x}_t^{[m]}) \tag{3.12}$$

$$\hat{x} = \arg\max_x Parzen(x) \tag{3.13}$$

$Parzen(x)$ was evaluated on a grid of resolution 0.5mm and 0.04π radians, with ρ set to 1 in both mm and radians.

3.3.4 Histogram Filter Estimation

In our experiments, discretization of the state space is accomplished simply by division into n_H equal-volume rectangular regions. The probability for each histogram bin is estimated based on the centroid $\check{x}_k$ of the corresponding region $\mathbf{x}_k$.

As stated previously, the command update only requires updating the robot kinematics estimates, so no change to the histogram is needed. For the measurement model, each bin is updated as

$$p_{k,t} = p_{k,t-1} \Pr(z_t \mid \check{x}_k, \mathbf{M}) \tag{3.14}$$

Letting C_k be the identity corresponding to region $\mathbf{x}_k$, the current identity estimate is given by

$$\hat{C} = \arg\max_C \sum_k p_{k,t} \mathrm{Ind}(C_k, C) \tag{3.15}$$

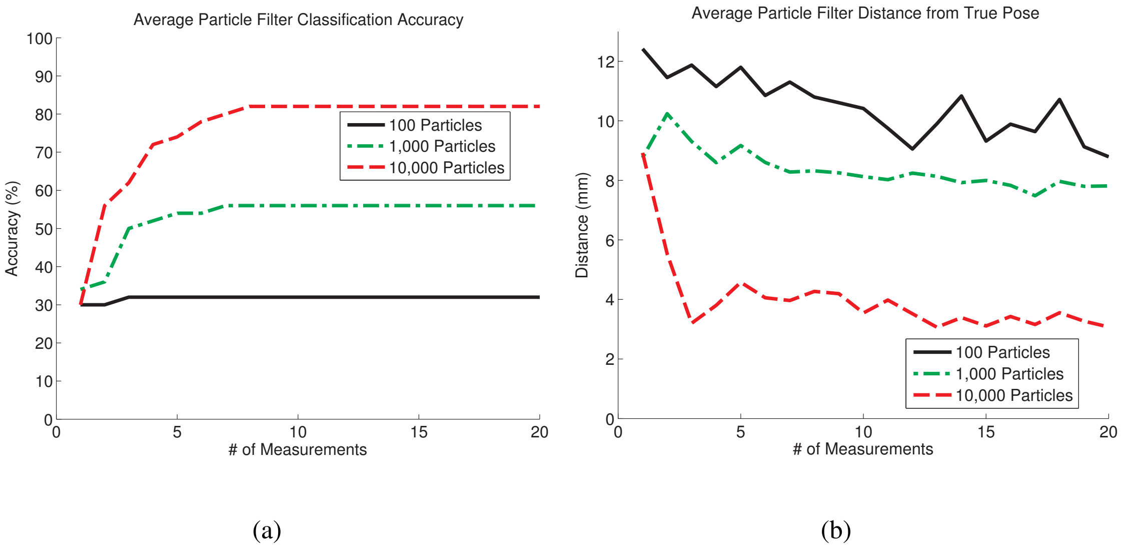


Figure 3.10: Particle filter performance at different resolutions.

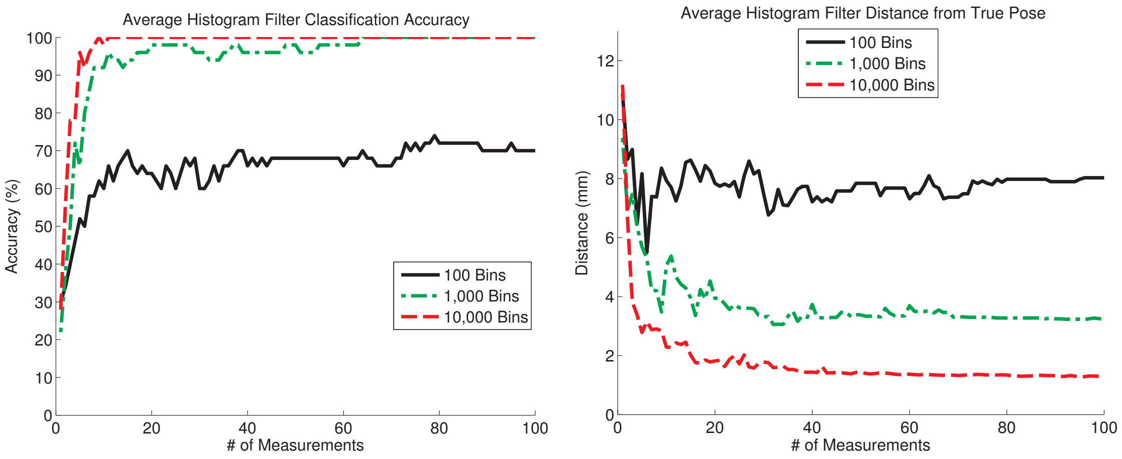
and the current state estimate is once again taken as the mode of the distribution, i.e.

$$\ell = \arg\max_{k} p_{k,t} \tag{3.16}$$

$$\hat{x} = \check{x}_{\ell} \tag{3.17}$$

3.4 Experiments

The geometric recognition method was tested by repeatedly performing recognition on all of the models in our 5 letter test set under different random 2D transformations. Transformations consisted of a translation selected uniformly at random from the range $[-10, 10]$ mm in both the x- and y-directions and the entire range of rotations in the plane. For each model, performance metrics were averaged across 10 trials. The method was



(a)

(b)

Figure 3.11: Histogram filter performance at different resolutions.

tested using both particle and histogram filtering, for comparison.

The sensor readings used for testing consisted of an alternate set of readings taken in the same positions and orientations as in the training set used to build the object maps. In each trial, a sequence of readings was selected from the test set at random, without replacement. A motion command was then generated for each reading to simulate the situation of a robot exploring the object in the unknown pose. The location information associated with each reading was transformed according to the random pose generated above, then the command and measurement were presented to the recognition algorithm.

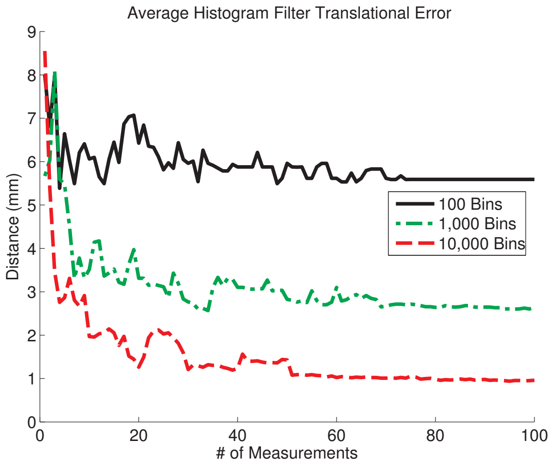


Figure 3.12: Histogram filter translational error at different resolutions.

3.4.1 Classification Accuracy

Accuracy in identifying the unknown object was recorded as a function of the number of readings seen. Performance in this metric is shown for the particle filter with 100, 1,000, and 10,000 particles in Fig. 3.10(a). The performance of the histogram filter under the same metric is shown with those same numbers of bins (broken down as shown in Table 3.1) in Fig. 3.11(a). The particle filter's performance leveled out for all numbers of particles after around 10 measurements as the particles prematurely converged to a single (often incorrect) hypothesis, and this approach only achieved an accuracy of 74% with 10,000 particles. The histogram filter, however, was able to achieve 100% accuracy with only 1,000 histogram bins after about 50 measurements and with 10,000 bins with only 9 readings. The computation required for a single histogram bin is roughly equivalent to that for a single particle, so this was a striking difference in performance.

Total Bins	x-translation	y-translation	Angle
100	5	5	4
1,000	10	10	10
10,000	25	25	16

Table 3.1: Histogram bin resolution in each state space dimension

Attempts to avoid premature convergence of the particle filter through the injection of random particles, as in Augmented MCL (Gutmann et al., 1998) did not improve performance. We also tried injecting particles sampled from a histogram filter running at a coarser resolution, but we found that the performance was not significantly better than that of the histogram filter alone.

3.4.2 Localization Accuracy

The average distance between the current state estimate and the true state was also computed for each iteration when the state estimate's object identity hypothesis was correct. This distance was computed using the metric from (Chirikjian and Zhou, 1998) (Eqn. 4), which uses an object's mass and moments of intertia to compute a scalar distance between two transformations, taking into account both translation and rotation. In our case, mass and moments of inertia were estimated from the object maps assuming uniform object density. The units used to report error were normalized with respect to the energy required to translate the object a distance of 1 mm, though a component of the reported values is also due to rotational error. This metric was modified to account for the rotational symmetry of two of the letters: “O” was considered to be fully rotationally symmetric and to have no rotation error, while “I” was considered to have two-fold rotational symmetry. The angular error for each letter, denoted $dAng_{Let}(\hat{\alpha})$ for letter Let, is given as

$$dAng(\hat{\alpha}) = |mod(\hat{\alpha} - \alpha, 2\pi)| \tag{3.18}$$

$$dAng_A(\hat{\alpha}) = dAng_E(\hat{\alpha}) = dAng_U(\hat{\alpha}) = dAng(\hat{\alpha}) \tag{3.19}$$

$$dAng_I(\hat{\alpha}) = |mod(\hat{\alpha} - \alpha, \pi)| \tag{3.20}$$

$$dAng_O(\hat{\alpha}) = 0 \tag{3.21}$$

Transformation error is shown for the particle filter in Figure 3.10(b) and for the histogram filter in Figure 3.11(b). To give an idea of the portion of error due to translation *vs.* rotation, the translational component of the error for the histogram filter is shown in Figure 3.12. Note that the performance of the histogram filter in this metric is inherently limited by the histogram resolution, therefore the particle filter should have a strong advantage. However, the premature convergence of the particle filter impacts localization accuracy as well, and the histogram filter nonetheless performed better again for all numbers of particles/bins.

3.5 Discussion

In this chapter, a method for generating rich surface models from array-type tactile force sensors was presented and its use was demonstrated on a set of real objects. A recognition algorithm was also described, which uses these surface models to estimate an unknown object's identity and pose, using only a small number of measurements. We defined two solutions to our formulation of the task as a sequential state estimation problem and compared their object identification and localization performance. The histogram filtering algorithm was able to achieve 100% accuracy on the test set with as few as 46 sensor readings using 1,000 bins and with only 9 readings using 10,000 bins, while objects were localized to within 1.3 mm of their true positions on average. Histogram filtering was shown to out-perform particle filtering in all cases.

It is important to remember that in this work we have focused entirely on the interpretation of whatever static sensor readings were available, but an active and informed exploration process could also greatly improve recognition.

The major limitation of a purely geometrical approach is in it scalability to full three-dimensional objects. In that case, the pose state space becomes six-dimensional, requiring one to square the number of bins in the histogram filter to achieve an equivalent resolution. A similar (though perhaps slightly less dramatic) increase in the number of particles used would be necessary for the particle filter as well. For many applications, such an increase would be computationally prohibitive, so it would be necessary to introduce methods to reduce the search space.

3.5.1 Interest Points

One possibility for reducing the state space that needs to be searched is through the use of interest points. Distinctive points on the surface of the object could be sought out and their positions recorded in the object model. Then when such a point is encountered on an unknown object, the set of possible poses that need to be searched will be greatly reduced.

Unlike in visual sensing, when interest points, e.g. SIFT (Lowe, 1999) features, may be simply searched for within available images, they must be actively sought out in touch sensing. It is certainly possible to develop exploratory procedures to seek out interest points, and preliminary tests indicate that relatively simple controllers could be used to move a tactile sensor across the surface of an object to converge over SIFT-like features. An interest point-driven approach implies a choice of exploration algorithm, however, since interest points could not be expected to be found often by chance. We have tried up to this point to remain agnostic to the exploration algorithm used, however, simply making the best use of whatever information was provided, and so we prefer a less restrictive approach.

3.5.2 Appearance

Another possible source of additional information is through characterization of the tactile appearance of objects, not only at distinctive points, but across their entire surface. Such a characterization provides information effectively “for free” in the sense that it does not require any specialized exploration algorithm (though it could of course still benefit from informed exploration). Chapter 4 investigates the characterization of objects' tactile appearance and presents a recognition algorithm using only that information and no geometry information. Then Chapter 5 incorporates the lessons learned from both geometry-only and appearance-only approaches into a unified approach that characterizes the spatially-varying appearance characteristics of objects.

Chapter 4

Tactile Appearance

4.1 Introduction

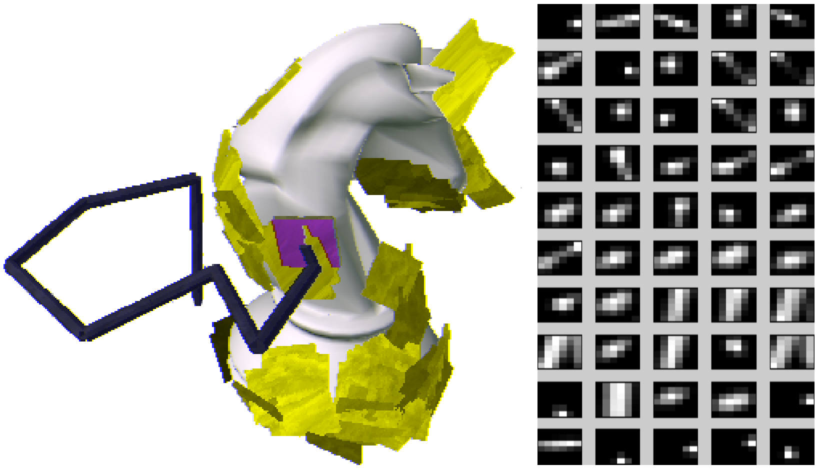
As mentioned previously, by thinking of sensor readings as images and using the sensing model of Chapter 2, we bring to bear a large body of work from computer vision. The methods from the vision domain investigated in this chapter deal with the concept of the local appearance of an object.

Though tactile images share many characteristics with visual images, they also have some important differences. Considering these differences provides some useful intuition as to what changes must be made to how images are handled in this new domain. The interpretation of the information in force images is somewhat simpler than in the visual case, since there are no perspective effects and there is only one channel of intensity information. At the same time, the collection of images is considerably more difficult, since each small patch must be obtained by actively interacting with the environment, whereas hundreds of features can be extracted from a single image obtained passively in the visual case.

Because of this coupling between sensing and manipulation, some care must be taken with regard to exactly how sensor readings are collected. For instance, simply making contact with the surface of an object is not sufficient to guarantee that a measurement is informative. Pressing down too hard can cause all sensor elements to saturate, resulting in a loss of information, since one is no longer able to discriminate fine surface features within the sensed region; of perhaps more concern is potential damage to the sensors, robot, or object. Similarly, pressing down while the sensor is in a non-ideal orientation causes elements on one side of the sensor to saturate, once again causing a loss of information.

We would like to develop a characterization of the surface textural properties of an object being explored. In order to do this effectively, we need to guarantee that if the same portion of the surface of an object is sensed at two different times, *regardless of the pose of the object or the sensor's approach trajectory*, that it will be perceived in the same way. We therefore develop an exploration strategy to accomplish this (discussed in Section 4.2) and expand our simulation of tactile sensors from Chapter 2 to the full robotic exploration task. The resulting simulator, illustrated in Figure 4.1 forms the basis for most of the experiments in this chapter.

Due to the aforementioned differences in the image formation process, it is not obvious how much of our knowledge of visual images will be transferable to tactile images. Accordingly, we take the approach of adapting and testing a variety of promising methods from



(a) Tactile exploration in simulation

(b) Tactile images

Figure 4.1: Depiction of a chess piece being explored by our simulated robotic arm (shown in dark blue) and tactile sensor system (shown in purple). Note that the tactile exploration method does not know the position, orientation, or the geometry of the object. Yellow patches show the sensor placements at which local controllers converged and a local appearance feature was extracted and recorded. The corresponding tactile images are shown to the right.

the vision literature, as well as developing novel ways of representing tactile information.

In this chapter, we are interested specifically in the interpretation of tactile images (separate from the geometric information also contained in a tactile sensor reading), which describe local surface "appearance". All of the characteristics that our sensors detect are ultimately geometric in nature, but we consider appearance in this context to consist of those characteristics that describe local variations in the geometry of the object's surface, separate from the gross shape of the object. We have therefore isolated the appearance portion of the object recognition task from its geometric counterpart to better observe the effects of changes in the appearance representation. Inspired by the success of bag-of-features techniques in the vision domain, we present an appearance-based recognition algorithm adapted to the domain of tactile data. Appearance-only algorithms are particularly useful for systems that cannot accurately measure the positions at which contacts are made, if the object is perturbed during exploration, or if the object's geometry is somewhat deformable (e.g., it has articulated joints, but surface characteristics remain consistent). A good understanding of information provided by appearance alone will also better inform the design of algorithms that make use of geometry information as well, as follows in Chapter 5.[1]

4.1.1 Related Work

First we briefly review the origins of appearance-based recognition approaches in the computer vision literature. Then we discuss the most related work that has applied this technique to the tactile domain.

4.1.1.1 Origins in Vision

Some of the most successful object recognition systems in the vision literature are based on local features, descriptions of statistics in a local neighborhood or "patch" of an image, often without any associated geometry information about the locations in the image from which these patches were sampled (Csurka et al., 2004; Nowak et al., 2006; Nister and Stewenius, 2006). An overview of this work is provided in Jurie and Triggs (2005). These

[1] This chapter covers work presented in Pezzementi et al. (2011a). Parts of this chapter were carried out in collaboration with Erion Plaku and with the assistance of Caitlin Reyda during her Research Experience for Undergraduates residence.

"bag-of-features" methods typically sample small patches of an image and use one of several descriptors to extract feature vectors from these patches. They then represent objects as producing distributions over these feature vectors. Interestingly, a recent psychophysical study indicates humans may also use local feature-based processing for tactile recognition (McGregor et al., 2010). The performance of local descriptors has been comprehensively studied on visual data (Mikolajczyk and Schmid, 2005; Zhang et al., 2007), and other groups have applied local descriptor techniques to shape retrieval on other types of 3D models (Fehr et al., 2009; Bronstein and Kokkinos, 2010). However, only one other group that we are aware of (Schneider et al., 2009, contrasted below) has applied these techniques to tactile data. In this chapter, we develop novel methods for adapting the feature-based approach to haptics and demonstrate its effectiveness in the new domain.

Some other groups have also incorporated some form of geometry information with their local descriptor representation (e.g. Fergus et al., 2003; Lazebnik et al., 2006; Cao et al., 2010). In this chapter, we focus on appearance information alone, and the incorporation with geometry is covered in Chapter 5.

4.1.1.2 Tactile Sensing

The recent work by Schneider et al. (2009) is most closely related to ours, since it also applies bag-of-features to data from tactile force sensors. The work presented in this chapter goes farther than Schneider et al. in several important ways:

- In their experiments, the pose of the objects is always known, considerably simplifying both the recognition problem and the process by which sensor readings are collected. The latter is treated as simply the selection of the height at which to grip the object. In this work, however, we leave the object pose as unknown (bounded only to be within the robot workspace) and we present exploration algorithms to collect consistent sensor readings in the face of this additional challenge.

- Additionally, Schneider et al. simply use the raw tactile sensor images as features, whereas we investigate several possible descriptors for extracting informative features.
- In this work, we also investigate the effects of several parameters of the sensors and the exploration process on recognition performance, helping to give a broader understanding of the situations in which this type of approach is most effective.

Some of the work discussed in Section 1.3 has addressed the issue of how to conduct haptic exploration of an unknown object, but generally with the goal of constraining the object's geometry, rather than that of collecting informative and consistent tactile force readings. Schneider et al. discuss the selection of maximally-informative grasps using entropy minimization, but they do not address the gripping process or its effect on the resulting tactile images. Gorges et al. (2010) presents an "attention-based" algorithm for exploring an object with tactile sensors to optimize recognition performance. They demonstrate an increase in recognition performance using this approach rather than random exploration on one object from a set of seven. Their evaluations are conducted in a simulation environment similar to ours, though they do not model the kinematics of the robot arm that the hand is attached to. Their recognition process is based on iterative closest point (Besl and McKay, 1992), rather than appearance information. Kraft et al. (2009, in the appendix) describe a pair of PI controllers for collecting tactile force sensor readings with consistent applied force and orientation, with the goal of estimating the surface normal. We derive a new but similar set of controllers that also align the tactile sensor with the object surface normal and apply a target force with the goal of extracting consistent sensor readings of a given patch of object surface.

4.2 Exploration

When exploring an unknown object, the objective is to collect sensor measurements from various locations on the object surface that would enable the recognition method to identify the unknown object. The fact that there is no a priori information about the position, orientation, and the geometry of the unknown object makes the exploration more challenging. Only the workspace boundaries of the robot are known, and it is assumed the object is somewhere within these bounds. The exploration is carried out in a simulator, which models the robot (as an articulated arm) and the behavior of the haptic sensor, which is attached to the end-effector. Details of the simulator can be found in Section 4.4.1.

Exploration strategies employed in this chapter vary from local strategies that attempt to cover one area and then move on to explore the next neighboring area, to global strategies

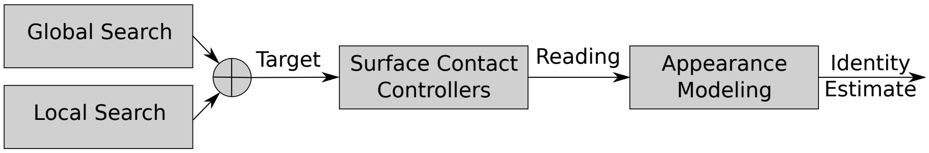


Figure 4.2: Illustration of exploration process for collecting each sensor reading.

that attempt to take sensor measurements from all over the surface of the unknown object. Exploration makes use of a surface contract controller, which enables the robotic system to take consistent sensor measurements regardless of the sensor's angle of approach to the surface of the unknown object. The rest of this section describes in more detail the exploration strategies (Section 4.2.1) and the surface contact control (Section 4.2.2).

4.2.1 Strategies to Explore the Unknown Object

Drawing from sampling-based motion planning (Choset et al., 2005; LaValle, 2006), the underlying idea in exploration is to sample various poses inside the robot workspace and compute collision-free motions that move the robot arm so that the sensor achieves the desired pose. The planner maintains a tree data structure, which is rooted at the initial configuration of the robot arm. The tree vertices consist of collision-free configurations, while edges indicate collision-free motions between the configurations that they connect.

The planner employs two strategies to grow the tree, one geared towards global exploration and another towards local exploration. At each iteration, the planner makes a probabilistic selection of which strategy to use; the local strategy is selected with probability L and the global strategy is selected with probability $1-L$. A study of the impact of L on the overall performance is presented in Section 4.4.2.3.

To guide the exploration to obtain a global view, the planner samples a target position p uniformly at random inside the workspace boundaries. Then the planner selects the configuration q from the tree whose associated sensor location is closest to p. This strategy, drawing from the rapidly-exploring random tree (LaValle and Kuffner, 2001) algorithm, has the effect of pulling the exploration toward new and different locations to ensure global coverage.

To guide the exploration based on local coverage, the planner imposes an implicit uniform grid over the workspace. Each time the sensor makes contact with the unknown object and a measurement is taken, the location ℓ of the sensor is added to the corresponding grid cell. In this way, each grid cell maintains a list of locations from which sensor measurements have been taken. From the list of non-empty grid cells, a cell c is then selected with probability inversely proportional to the number of measurements taken from locations inside that cell. Thus, the planner gives preference to cells that have few measurements, since further exploration of these cells may increase the local coverage. The planner then selects a location ℓ uniformly at random from all the locations associated with c and samples a target position p uniformly at random inside a small sphere centered at ℓ. The configuration from which to expand the tree is then selected as the configuration in the tree that is closest to p. In this way, the planner attempts to increase the local coverage of the selected cell and move the exploration toward neighboring areas.

After a configuration q in the tree and a target position p are selected, the objective of the planner is to expand the tree from q toward p. Recall that the planner only knows the workspace boundaries and has no a priori information about the position, orientation, and the geometry of the unknown object. For this reason, the planner takes small steps toward p. In particular, at each iteration, the planner computes the direction from the location of the sensor to p and attempts to move in that direction to a nearby point p'. The planner employs numerical inverse kinematics to compute the configuration q' that places the sensor at location p'. The planner then relies on a controller to slowly move the robot arm from configuration q to q'. If at any time during this movement the object is sensed, the planner switches to the surface contact control scheme, which is described in the next section, to obtain a measurement.

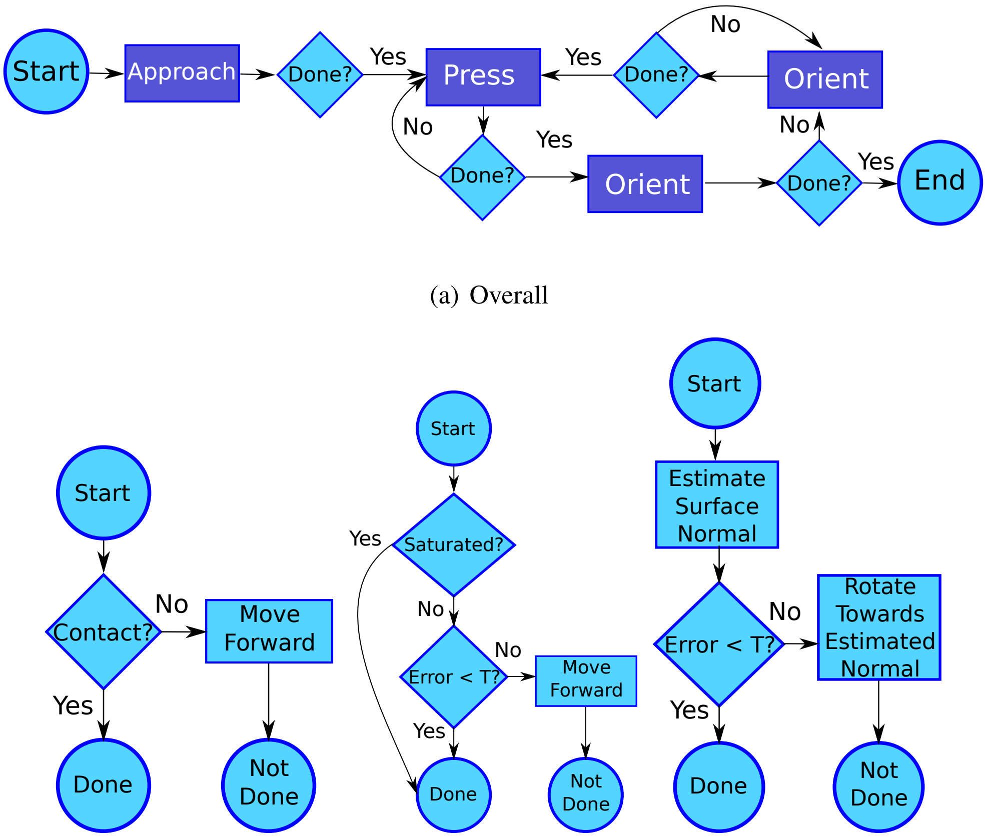
As evidenced by the experiments, this combination of local and global strategies allows for an effective exploration of the surface of the unknown object. The exploration process is illustrated on a 3D model of a chess piece in Figure 4.1, which shows where 100 tactile images were extracted using the planner, alongside depictions of the first 50 of these images.

We also note that, since the exploration strategy does not rely on knowing the position, orientation, and geometry of the object, the exploration strategy is suitable even when the unknown object is perturbed between sensor readings, e.g., as a result of robot manipulations. In fact, such motions have no effect on the global strategy, since the global strategy is guided by uniform sampling inside the workspace. The effect on the local strategy is also minimal. If the planner takes a sensor measurement at location ℓ, it will attempt to take another sensor measurement at a target position p sampled uniformly at random inside a sphere centered at ℓ. As such, even if the unknown object is perturbed when taking a sensor measurement at ℓ, it is likely that it moved locally so that sampling in a local neighborhood of ℓ is generally suitable to accommodate such motions. As the experiments indicate in Section 4.4.2.4, the overall approach remains effective even when the unknown object is perturbed between sensor readings.

4.2.2 Surface Contact Control

The objective of the surface contact control scheme is to extract a consistent descriptor each time a sensor measurement is taken at a given object location, regardless of the sensor's angle of approach to the surface, to provide the object recognition scheme with reliable estimates of the local surface properties. Because of the small field of view of typical tactile sensors, normalization of the image with respect to the contact pose cannot be expected to be achievable solely through post-processing of the resulting images. Therefore, to achieve measurement consistency, some level of closed-loop control is necessary. The entire control scheme used in this portion of the exploration process is illustrated in Figure 4.3(a). Three surface contact controllers are used to establish consistent sensor poses. All controllers use the output of the tactile sensor to compute commands for the robot arm.

The overall strategy begins with the Approach controller (Figure 4.3(b)), which moves the sensor in a given direction until it comes into contact with the object. Achieving contact



(a) Overall

(b) Approach

(c) Press

(d) Orient

Figure 4.3: Surface contact controller flow charts. (a) shows the flow of control between surface contact controllers, and (b), (c), and (d) depict the individual controllers.

then engages the Press controller (Figure 4.3(c)), which continues to move the sensor along the same axis until achieving a target average pressure reading. Then the Orient controller is engaged to bring the sensor as close to coplanar with the object surface as possible. Finally, control is passed sequentially back and forth between Press and Orient until both controllers consecutively issue no command.

While it would be possible to implement surface contact control using standard closed-loop force feedback controllers, with the variety of goals and the complexity of making and breaking contact, we found a step-wise formulation to be useful. The Approach and Press controllers' implementations are fairly straightforward, while that of Orient is more involved. Approach implements essentially a guarded move, terminating as soon as any sensor element response goes significantly above zero. Press is implemented as a PD controller with a second termination criterion if any single sensor element becomes close to fully saturated. The Orient controller operates by fitting a plane to the pressure readings of the individual sensor elements (implicitly fitting a plane to the surface being sensed) and commanding the robot to re-orient the sensor normal to the plane fit normal, as shown in Algorithm 4.2.2. In our experiments, $step$ in line 11 was set to 0.3. This process is repeated until either the normals converge to within a thresholded angle of each other or a maximum number of iterations is reached (to deal with cases of oscillation or too slow convergence).

It can be seen later in the experiments (Figures 4.6 and 4.7) that these controllers converge upon features such as edges and corners as well as flat surfaces. Their convergence characteristics are analyzed quantitatively in the presence of noise in Section 4.4.3.

4.3 Interpreting Tactile Data

Each tactile image obtained during testing or training is converted into a feature vector for further processing. Drawing from the computer vision literature, we make use of image descriptors that have worked well for a wide variety of recognition problems, as described in Section 4.3.2. The objective of the image descriptors is to extract the most relevant

Algorithm 1 Orient Controller

```
$pts \leftarrow \emptyset$
for all sensor elements $i$ do
    if $val(i) > contactThresh$ then
        $p \leftarrow point3D(getX(i), getY(i), estimateDepth(val(i))$
        add $p$ to $pts$
    end if
end for
$normal \leftarrow fitPlane(pts)$
$sensorN \leftarrow toWorldCoords(point3D(0,0,1))$
$surfaceN \leftarrow toWorldCoords(normal)$
$target \leftarrow step \cdot surfaceN + (1 - step) \cdot sensorN$
$cmd \leftarrow rotationFromTo(sensorN, target)$
return $cmd$
```

information for characterizing the local surface properties. Moreover, since surface contact controllers (Section 4.2.2) control for orientation except about the axis normal to the sensor surface (and this angle is not recorded), the descriptors need to be invariant to rotations about this axis. The extracted features are then used in the recognition process, as described next.

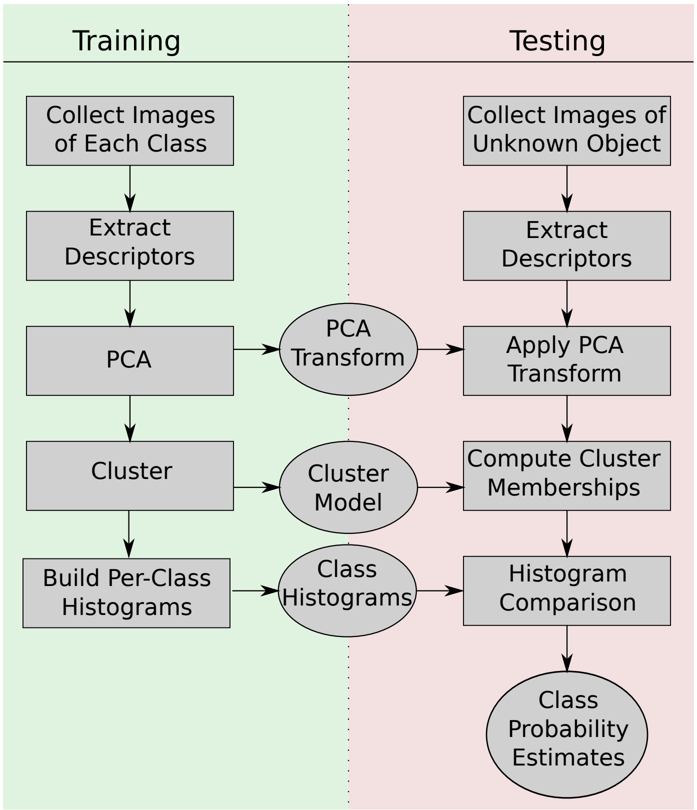


Figure 4.4: Process for learning bag-of-features models for each object class and applying them to classify unknown objects.

4.3.1 Bag-of-Features Modeling

A bag-of-features approach (Jurie and Triggs, 2005) is developed to model the appearance of objects. The major steps of the process for learning this model (Training) and for applying it to recognition (Testing) are illustrated in Figure 4.4.

4.3.1.1 Obtaining Appearance Distributions

The processes for training and testing share many common steps, as shown in Figure 4.4, with the testing pipeline using models acquired during training (shown in ellipses).

We'll go over the training process and the generation of these models first, then how they're used in testing.

4.3.1.1.1 TRAINING

Let $\mathbf{O}_1, \ldots, \mathbf{O}_{n_O}$ denote the object classes used for training. For each object class $\mathbf{O}_j$, a set $\{\mathbf{I}_{j,1}, \ldots, \mathbf{I}_{j,n_I}\}$ of n_I images is collected via the exploration procedure described in Section 4.2. Then one of several algorithms for extracting appearance information is applied. These algorithms, described in Section 4.3.2, are referred to as descriptors and denoted $\mathbf{d}$. For each descriptor $\mathbf{d}$, a set of appearance feature vectors $\mathbf{V}_j = \{\mathbf{v}_{j,1}, ..., \mathbf{v}_{j,n_{Vj}}\}$ is extracted for all of the images from object class j, with the features from each image denoted $\mathbf{d}(\mathbf{I})$. The collection of features from all objects is then reduced in dimensionality by using PCA and discarding the least significant components that account for up to 10% of the variance.[2] The reduced feature vectors are then grouped into clusters $\{\mathbf{c}_1, \ldots, \mathbf{c}_{n_C}\}$ by a learned clustering function, $\mathbf{C}(\mathbf{v})$, which takes a feature, $\mathbf{v}$, and outputs its cluster membership. The choice of an appropriate clustering method is discussed further in Section 4.4.2.1.

4.3.1.1.2 TESTING

During testing, the data consist of images obtained by the exploration procedure in Section 4.2.1. Descriptors are extracted from each image, as in training, then their dimensionality is reduced with the PCA transform from training. Finally, the cluster membership

[2] The dimensionality reduction was chosen to avoid loss of information while improving efficiency, and it had very little impact on performance. LDA might be used for a more aggressive reduction.

function obtained in training is applied to the descriptors to form an empirical distribution on cluster membership. This process gives a histogram representing the probability of drawing observed features $\mathbf{d}(\mathbf{Z})$, taken from the observed measurements $\mathbf{Z} = \{z_1, ..., z_{n_U}\}$, from each cluster given data from the unknown object, which is denoted as $\Pr(\mathbf{c}_i \mid \mathbf{Z})$.

4.3.1.2 Matching Appearance Distributions

Then, given an estimate of $\Pr(\mathbf{c}_i \mid \mathbf{O}_j)$ for each object class $\mathbf{O}_j$ from training, the best-matching object identity, C, is taken as that which minimizes the K-L divergence (Kullback and Leibler, 1951) between the distributions $\Pr(\{\mathbf{c}_i\} \mid \mathbf{O}_j)$ and $\Pr(\{\mathbf{c}_i\} \mid \mathbf{Z})$,

$$C = \min_j D_{KL}\left(\Pr(\{\mathbf{c}_i\} \mid \mathbf{Z}) \mid\mid \Pr(\{\mathbf{c}_i\} \mid \mathbf{O}_j)\right) \tag{4.1}$$

$$D_{KL}\left(\Pr(\{\mathbf{c}_i\} \mid \mathbf{Z}) \mid\mid \Pr(\{\mathbf{c}_i\} \mid \mathbf{O}_j)\right) = \sum_i \Pr(\mathbf{c}_i \mid \mathbf{Z}) \log \frac{\Pr(\mathbf{c}_i \mid \mathbf{Z})}{\Pr(\mathbf{c}_i \mid \mathbf{O}_j)} \tag{4.2}$$

$$= \sum_i \Pr(\mathbf{c}_i \mid \mathbf{Z}) \log \Pr(\mathbf{c}_i \mid \mathbf{Z}) - \sum_i \Pr(\mathbf{c}_i \mid \mathbf{Z}) \log \Pr(\mathbf{c}_i \mid \mathbf{O}_j) \tag{4.3}$$

Since $\Pr(\mathbf{c}_i \mid \mathbf{Z})$ is fixed in the optimization, the first term can be dropped, leaving

$$C = \text{argmin}_j - \sum_i \Pr(\mathbf{c}_i \mid \mathbf{Z}) \log \Pr(\mathbf{c}_i \mid \mathbf{O}_j) \tag{4.4}$$

This minimization can also be interpreted as a maximization of the likelihood of the data over object identity, as shown in the Appendix.

Other methods of comparing histogram distributions, such as histogram intersection and χ^2, were also considered, but experiments indicated that the above formulation gave significantly better results. This formulation is also more easily adaptable to integration with a guided search framework for evaluating the potential information content of future measurements and choosing an appropriate exploration strategy, given the maximum likelihood interpretation derived in the Appendix.

4.3.2 Descriptors

Several different descriptors, as described below, were considered for representing the essential information from the sensor readings in an intensity- and rotation-invariant way. We first present the descriptors that are adapted directly from their counterparts in the computer vision literature, SIFT and MR-8. The remaining descriptors are novel. We also investigated additional vision-inspired descriptors based on steerable filters (Freeman and Adelson, 1991) and the Schmid texture descriptor (Schmid, 2001), but they have been omitted due to poor performance.

4.3.2.1 Vectorize

This descriptor simply takes a tactile image and concatenates its columns to form a vector. The result should not be rotation-invariant unless the images happen to be rotationally symmetric. This is our negative control, and can be considered a “do-nothing” descriptor, inspired in part by (Varma and Zisserman, 2003), to show a baseline performance level provided by the rest of the method in the absence of a specially-tailored descriptor. This is also the descriptor used by Schneider et al. (2009) in their work in which the object orientation is fixed.

4.3.2.2 SIFT

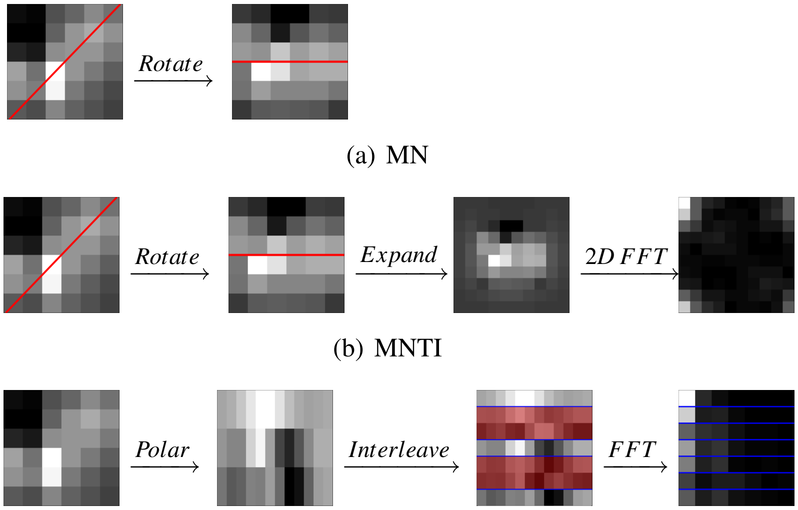
SIFT features have been shown to perform extremely well in visual texture discrimination (Mikolajczyk and Schmid, 2005; Zhang et al., 2007). In adapting these features to the tactile domain, we follow many others in the vision community (e.g. Lampert et al., 2009; Fergus et al., 2005; Bosch et al., 2006) by applying only the descriptor portion of the SIFT algorithm to characterize image patches. This practice seems particularly appropriate since the tactile images already represent patches of the object surface. The standard SIFT descriptor of Lowe (1999) is applied to the entire tactile image, as implemented in the VLFeat library (Vedaldi and Fulkerson, 2008), at a scale corresponding to the size of the image and orientation derived in the standard SIFT way. To avoid histogram sparsity issues, the computation was switched from a 4x4 to a 2x2 grid of sampling areas at low resolutions, giving a 32-element vector rather than the usual 128. However, no significant differences in performance were observed in this context, as compared to using the full 128-element descriptor, even for the smallest images.

4.3.2.3 MR-8

Varma and Zisserman compared various filter sets for texture classification, and we chose their best-performing filter set, MR-8 (Varma and Zisserman, 2005), as one of our descriptors, as implemented by Geusebroek et al. (2003). The Maximal Response set consists of first- and second-order derivatives of oriented Gaussians at different scales and angles as well as two rotationally symmetric filters: a Gaussian and a difference of Gaussians. Rotational invariance is achieved by only taking, from the set of all angles for each oriented filter, the largest-magnitude response, on a pixel-by-pixel basis. The oriented filters consist of 3 scales and 2 orders of derivatives, evaluated at 6 angles each. So taking the maxima of these gives 6 responses, plus the two symmetric filters' responses, for a total of 8. The responses of subsections of the tactile image to all 8 of the filters selected by the process above are concatenated to form feature vectors. Then the set of feature vectors from all sections of the image (4 overlapping sections in the 6x6 case) are returned as the image's descriptor. This descriptor is unique among those presented in this work in that it returns multiple feature vectors for each input tactile image. However, it is also by far the most computationally expensive, particularly for large images.

4.3.2.4 Moment-Normalized

First, the image is masked so that only pixels within the largest inscribed circle about the image center are retained, and other values are set to zero. Then, following (Hu, 1962), the descriptor computes spatial moments for the image with respect to the image center (not its center of inertia), normalizes them for scale, and extracts the image's principal axes. The angle of the major axis is taken as a measure of orientation, and the 180 degree ambiguity is resolved with the use of the sign of a 3rd-order moment (again from (Hu, 1962), though this may still fail for certain types of symmetry). The image is then rotated spatially with bilinear interpolation so that the computed major axis direction is aligned with the positive-X-axis, as illustrated in Figure 4.5(a). Finally, the resulting image is converted to a vector



(a) MN

(b) MNTI

(c) PF

Figure 4.5: Illustration of the major steps for extracting features using novel descriptors

as in Vectorize. It should be invariant to intensity changes and rotation, though local control should have already eliminated most intensity variations.

4.3.2.5 Fourier-Based Descriptors

The remaining two descriptors make use of the well-known fact that the magnitudes of the Fourier coefficients of a periodic signal are invariant to phase shifts of that signal, as shown below. Let the Fourier transform of a function $f(x)$ be given by

$$\hat{f}(\omega) = \mathscr{F}\left(f(x)\right) \tag{4.5}$$

$$= \int_{-\infty}^{\infty} f(x)e^{2\pi i x\omega} \tag{4.6}$$

Then the transformation of a translated version of the signal, $f(x-\alpha)$ is

$$\mathscr{F}(f(x-\alpha)) = \int_{-\infty}^{\infty} f(x-\alpha)e^{2\pi i(x-\alpha)\omega}dx \tag{4.7}$$

$$= \int_{-\infty}^{\infty} f(x-\alpha)e^{2\pi ix\omega}e^{2\pi i(-\alpha)\omega}dx \tag{4.8}$$

$$= e^{2\pi i(-\alpha)\omega}\int_{-\infty}^{\infty} f(x-\alpha)e^{2\pi ix\omega}dx \tag{4.9}$$

$$= \underbrace{e^{2\pi i(-\alpha)\omega}}_{\text{magnitude 1}}\hat{f}(x) \tag{4.10}$$

The exponential term always has magnitude 1, so phase shifted versions of a signal all have the same Fourier magnitudes. This can be thought of as translation of a linear signal on a repeating domain or rotation of a signal defined on a circle.

Here the signal is composed of tactile image elements. The Polar-Fourier descriptor uses this property to obtain invariance to rotation, while the Moment-Normalized Translation-Invariant (MNTI) descriptor uses it for translational invariance. It is important to realize, however, that translation/rotation results in a structured change to phases of Fourier coefficients, and so discarding the phase components results in a representation that is invariant to more transformations than just translations/rotations. This step also results in a loss of information, with the effect that the original image can no longer be uniquely reconstructed from the resulting descriptors.

4.3.2.5.1 POLAR-FOURIER

Formulation

This descriptor begins by masking out the corners of the image, as in the moment-normalized descriptor. Then the image $\mathbf{I}$ is re-sampled using polar coordinates to produce a new rectangular image $\mathbf{I}_P$ whose axes are radius and angle. Let (x_0, y_0) be the center of the original image, D be the diameter of the image's largest inscribed circle, and i and j vary as $\{1, 2, \ldots, D\}$. Then $r = \frac{i}{2D}$, $\theta = \frac{2\pi j}{D}$, and

$$\mathbf{I}_P(i,j) = \mathbf{I}\left(x_0 + r\cos(\theta), y_0 + r\sin(\theta)\right), \tag{4.11}$$

where $\mathbf{I}(x,y)$ indexes the original image's pixels. In this way, each row of this image corresponds to a single circle (each of a different radius), and moving across columns traces out a circle. Between each consecutive pair of rows of this image, two new rows are added, corresponding to the sum and the difference of the surrounding rows, to form $\mathbf{I}_Q$:

$$\mathbf{I}_Q(3i, j) = \mathbf{I}_P(i,j) \tag{4.12}$$

$$\mathbf{I}_Q(3i+1, j) = \frac{\mathbf{I}_P(i,j) + \mathbf{I}_P(i,j)}{2} \tag{4.13}$$

$$\mathbf{I}_Q(3i+2, j) = \frac{\mathbf{I}_P(i,j) - \mathbf{I}_P(i,j)}{2} \tag{4.14}$$

The Fourier transform of each row of this new image is taken, and the magnitudes of the resulting coefficients are recorded. This process is illustrated on a sample input in Figure 4.5(c). From these coefficients, a vector is formed by choosing the N lowest-frequency coefficients from each row, where N is proportional to the radius at which that row's points were sampled, rounded to the nearest whole number.

Rationale

In the Polar-Fourier domain, a rotation of the original image about its center results in only a change in phase of the Fourier coefficients, so the coefficient magnitudes should be invariant to rotation. Rotations only cause a particular family of phase changes, though, so discarding phase information completely also leaves the descriptor invariant to many other transformations, such as independent rotations of the various "rings" of the original image represented by the rows of the polar representation.

The extra rows added in Equations 4.13 and 4.14 to form $\mathbf{I}_Q$ serve to provide information on how adjacent rings of the original image were related, to mitigate the effects of losing such relationships when discarding the phases of the Fourier components in this polar space. Note that in the conversion to polar coordinates, sampling is denser toward the center of the image than at its outskirts. The use of only the low-frequency coefficients in the final step is intended to provide frequency content corresponding as closely as possible to a uniform sampling of the original image. In order to have the total magnitude of the descriptor also correspond to a uniform sampling of the original image (when computing correlation), it is necessary to normalize each row of $\mathbf{I}_Q$ by dividing by (the square root of) the radius of the circle from which it was sampled. We found that performing this normalization resulted in a decrease to performance or no significant change, however, so the fact that it is omitted amounts to placing increased weight on information toward the center of the tactile image. A version of the descriptor that does perform this normalization is referred to as radially-weighted Polar-Fourier (rwPF). Similarly, a version of the descriptor that does not perform the step of interleaving information about adjacent rows, effectively using $\mathbf{I}_P$ in place of $\mathbf{I}_Q$, is referred to as non-interleaved Polar-Fourier (niPF).

This descriptor can also be thought of as a generalization of "spin images" (Johnson, 1997). In this case, the set of zero-order Fourier component magnitudes from each row would correspond to the direct application of the spin image technique to tactile images, while the other Fourier magnitudes provide strictly more information. It is also similar to the generalization of "shape contexts" (Belongie et al., 2002) to 3D in Frome et al. (2004), which use either spherical harmonic magnitudes or polar coordinates with redundancy to achieve rotational invariance.

4.3.2.5.2 MNTI

Finally, we also include a modification of the Moment-Normalized descriptor to be invariant to translations, which is referred to as moment-normalized translation-invariant (MNTI).

Formulation

This descriptor follows the same procedure as Moment-Normalized up to the final vectorization step. Instead of vectorizing the result, the image is enlarged to 150% its original size and padded with new pixels set to values linearly interpolated between the original image's boundary pixel values and zero, as shown on an example input in Figure 4.5(b). Then MNTI is obtained by taking a 2D spatial Fourier transform, and once again recording only the magnitudes of each Fourier component. The final descriptor is the vectorization of the low-frequency components corresponding to the size of the original image.

Rationale

The primary goal of this modification of the Moment-Normalized descriptor is for a small translation of the sensor with respect to the object surface to result in a small change to the descriptor; if descriptor $\mathbf{m}_1$ is collected at position $\mathbf{d}$ and descriptor $\mathbf{m}_2$ is collected at nearby position $\mathbf{d}+\Delta$, we would like the correlation $\mathbf{m}_1 \star \mathbf{m}_2$ to still be large. With the original descriptor, this may not be the case, since a δ on the order of a pixel size can cause the descriptors to become decorrelated.

As in the previous descriptor, discarding the Fourier phase has the effect of adding invariance to a set of transformations. Since we are taking the 2D transform in the original image space, this includes the set of 2D translations (once more along with many others). Application of the Fourier transform, however, assumes the image is a repeating signal that wraps around at the image boundary. Since our image is non-repeating, performing this transformation without modification would result in large discontinuities at the image boundaries, leading to “ringing” effects, particularly at high frequencies. The padding step is intended to mitigate these boundary effects; it leaves the boundary values uniform, with no implicit discontinuities. Only the low frequency components are used in the final step to remove the effects of the artificial enlargement of the image and to focus attention on low-frequency characteristics of the signal, leaving it less sensitive to translations.

4.4 Evaluations and Results

The evaluations highlight the effectiveness of the proposed framework in recognizing unknown objects from sensor measurements gathered during exploration. The evaluations indicate a high degree of recognition accuracy for various simulated shapes. In addition to strong performance in simulation, the proposed framework is shown to be effective in recognizing objects based on real sensor measurements.

4.4.1 Evaluation Setup

4.4.1.1 Simulated Robotic System

The simulated robotic system consists of an articulated arm equipped with a haptic sensor at the end-effector. In its initial configuration, the first link of the robotic arm is perpendicular to the xy-plane and all the other links are perpendicular to the yz-plane. Any two consecutive links of the robotic arm are connected by a joint that allows rotations about the y- and z-axes. The tactile sensor (simulated by the method described in Chapter 2) is connected to the last link via a universal joint. This particular robotic system was chosen to provide a concrete setup for developing and testing the exploration strategies. Note, however, that the exploration strategies in this paper are general and can be used with any robotic system for which forward kinematics are available.

4.4.1.2 Data Collection During Training

Note that the planner (Section 4.2) is employed only during the testing stage of the framework when the objective is to identify an unknown object from various sensor measurements taken during exploration. During the training stage, the position, orientation, and geometry of the object are known, so much simpler strategies can be used to collect measurements. Measurements during training are collected by placing the sensor at various locations close to the surface of the object and then allowing the local controllers to converge. Specifically, for each reading collected in training, an element of the triangular mesh representing the object surface is selected with probability proportional to its area. A point p is then sampled uniformly at random inside the selected triangle and a point q is obtained by moving a small distance in the direction of the triangle normal. The sensor is then placed at location q facing toward the triangle. If p lies in a concave region, then it is possible for the sensor to intersect the object surface when placed at q; if this occurs, the process restarts with the sampling of a new point, p. Otherwise, a small perturbation is applied to the sensor orientation, and then the local controllers are used to approach the surface and take a measurement. This process is repeated until a specified number of sensor measurements are obtained. In this way, the exploration strategy during training is computationally fast and allows us to obtain good coverage of the object.

4.4.2 Simulations with 3D Objects

The effectiveness of the framework was first tested on various simulated shapes. The effect of different clustering methods as well as the choice of descriptor under several resolution, noise, and covering configurations were also evaluated. These simulation evaluations allowed us to select good parameters for the framework before applying it to real objects and sensors.

For the simulation evaluations, a set of 10 shapes from the Princeton Shape benchmark

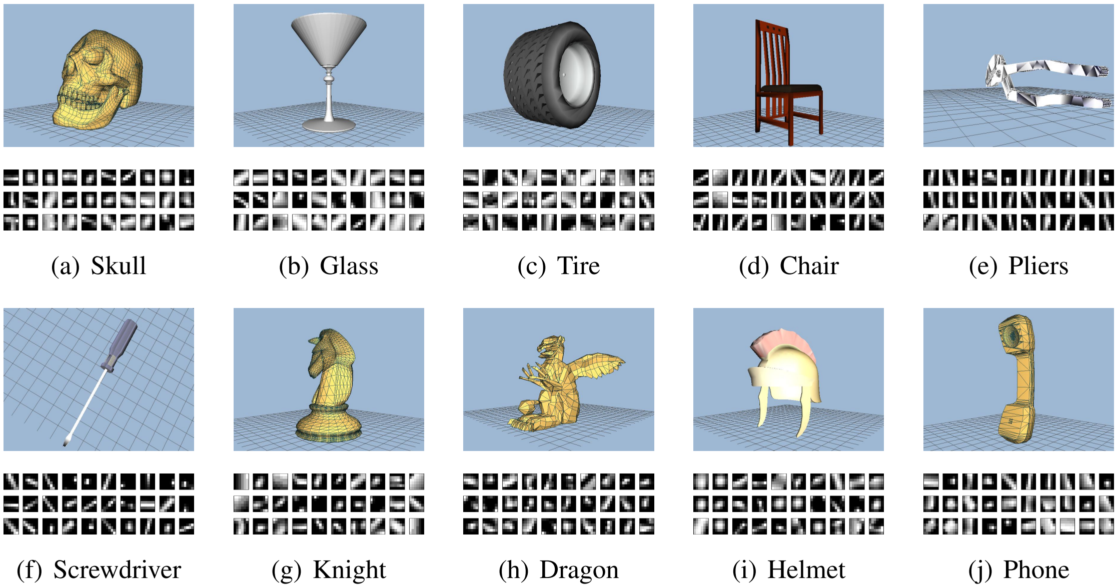

(a) Skull
(b) Glass
(c) Tire
(d) Chair
(e) Pliers
(f) Screwdriver
(g) Knight
(h) Dragon
(i) Helmet
(j) Phone

Figure 4.6: The set of models from the Princeton Shape Benchmark (Shilane et al., 2004) used for testing. Some are shown as wire-frames or colored for clarity, but only geometry information was used in evaluations. Below each is a sampling of 30 6x6 tactile images measured from that object during training.

(Shilane et al., 2004) was used, as shown in Figure 4.6. The sample shapes were selected to span several different object categories and to present a variety of interesting surface geometries, in order to cover a large portion of the range of local appearance characteristics that descriptors would need to represent.

For training, the sampler (Section 4.4.1.2) was used to collect 1000 tactile images of each object for learning models of the objects, plus another 100 samples of each object to form a validation set that was used to evaluate performance during the training process. Then for testing, the planner (Section 4.2) was used to collect a further 100 samples of each object which were compared with the learned models. This testing stage was then repeated 3 times, and the results were averaged to smooth out inconsistencies due to small amounts

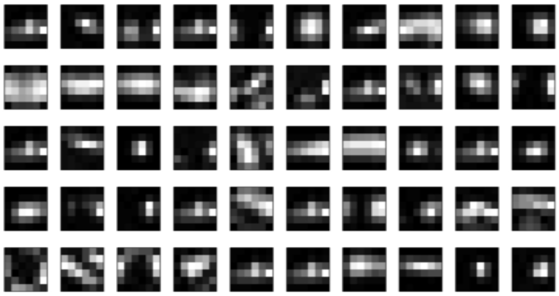


Figure 4.7: A sampling of cluster centers from training on the Princeton set with the Moment-Normalized descriptor. Each image represents the mean of a Gaussian mixture component, back-projected into the original 6x6 tactile image space.

of data.

4.4.2.1 Clustering

A variety of clustering methods was evaluated for forming the bag-of-features models. The standard k-means approach was used as a starting point, with the initialization method described in Arthur and Vassilvitskii (2007). We considered it essential that the clustering algorithm provide an efficient membership function that can be applied to new data after training, which removed many algorithms from consideration. We began by applying k-means repeatedly with various values of *k*, to mitigate sensitivity to initialization conditions. Performance during this process was measured using a validation set, consisting of data reserved from the training set. The "best" model was maintained as that which displayed the highest classification accuracy on the validation set. Since classification accuracy was a relatively coarse measurement, ties were broken by considering classification reliability, the total probability weight allocated to correct classes.

In order to visualize the clustering results, the cluster centers were back-projected through PCA and then reshaped into the original image space. This is only possible with the Vectorize and Moment-Normalized descriptors, as the others involve a loss of spatial information that prevents reconstruction of a unique representative image. Inspecting the cluster centers resulting from k-means revealed several clusters that either seemed redundant or appeared not to correspond to real data points. These effects motivated investigation of soft clustering techniques to mitigate the discretization inherent in k-means, so we then turned to Gaussian mixture models (GMMs). We began with the same image descriptors as in k-means, reduced in dimensionality with PCA. A single mixture model with k components was fit to the data. Let a mixture model, $\mathbf{GMM}$, consisting of $n_{\mathbf{GMM}}$ components, $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_{n_{\mathbf{GMM}}}\}$, be defined as

$$\Pr(x \mid \mathbf{GMM}) = \sum_{i=1}^{n_{\mathbf{GMM}}} \Pr(x \mid \mathbf{g}_i) \Pr(\mathbf{g}_i) \tag{4.15}$$

$$\Pr(x \mid g_i) = |\Sigma_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i (x-\mu_i)}, \tag{4.16}$$

In order to compute the cluster membership defined in section 4.3.1, this set of probabilities was interpreted as a soft binning function into a histogram where each bin corresponds to a mixture component. The likelihood of each cluster, $\mathbf{c}_i$, associated with the object's entire feature set, $\mathbf{V} = \{\mathbf{v}_\ell\}$, is computed as

$$\Pr(\mathbf{c}_i \mid \mathbf{V}) = \eta \sum_{\ell} \Pr(\mathbf{v}_\ell \mid \mathbf{g}_i) \Pr(\mathbf{g}_i) \tag{4.17}$$

where η is a normalization constant. Sets of data points were then "binned" and summed to form histograms of cluster/component representation, which were compared using the same method described in Section 4.3.1. A sampling of 48 cluster centers using the Moment-Normalized descriptor is shown in Figure 4.7. As in the k-means case, the mixture component means are back-projected through the PCA transformation and reshaped into the original image space. The differing covariances associated with each mixture component add additional information to this clustering result, but they are not visualized here. The phenomenon of repeated clusters mentioned above can still be observed to some extent, but soft membership allows weighted association with all clusters simultaneously, imparting much more information than discrete association with a single (potentially outlier) cluster. Accordingly, performance using GMMs was substantially higher, but so was computation time.

4.4.2.2 Descriptor Comparison

Figure 4.8 compares the performance of the various descriptors on the set of Princeton models, as a function of the number of tactile readings of the object surface that were sampled. In this format, which is used for all our graphs, each data point tells the empirical probability of correctly identifying any unknown object given that number of samples of its surface, using the indicated descriptor. More samples tend to give a better estimate of the true appearance distribution, leading to higher recognition accuracy, but adding non-representative samples can also lower accuracy.

The descriptors taken directly from the vision literature (SIFT and MR) gave poor results, generally no better than simply using the original image (Vectorize). Polar-Fourier

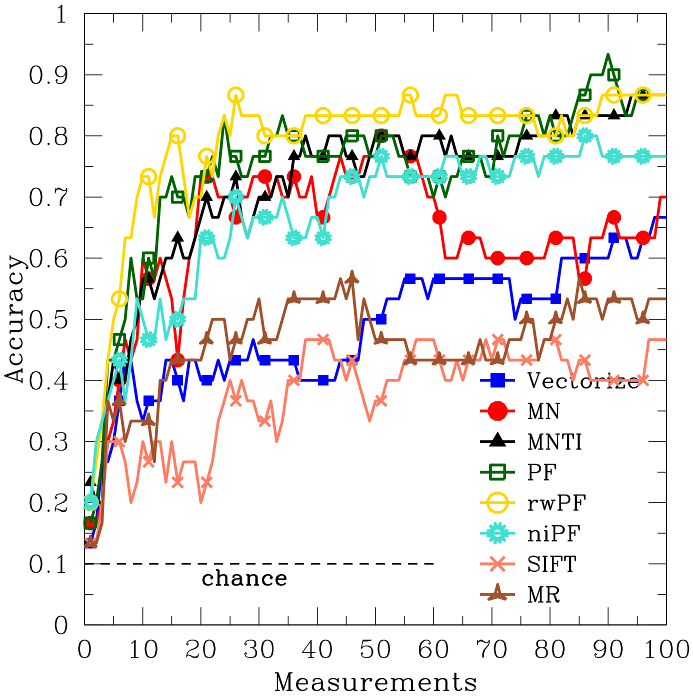
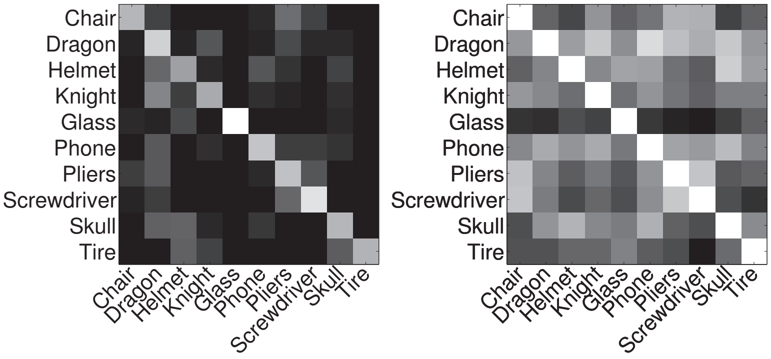


Figure 4.8: Comparison of various descriptors on Princeton set at 6×6 resolution and 10% covering thickness.

(and its variants) and MNTI perform best in terms of classification accuracy, followed by the Moment-Normalized descriptor. Two variations on the standard Polar-Fourier descriptor are shown, rwPF and niPF. rwPF uses the radial scaling normalization described in Section 4.3.2.5, and niPF skips the step of interleaving information relating adjacent rows. niPF performs worse than PF, and rwPF have approximately the same performance. PF and rwPF were verified to have approximately the same performance in the remaining simulation configurations, so we use the standard PF descriptor to minimize computation. For this reason, only the performance of these top 3 descriptors (PF, MNTI, and MN) and Vectorize are shown for subsequent tests, though this performance trend was verified to continue

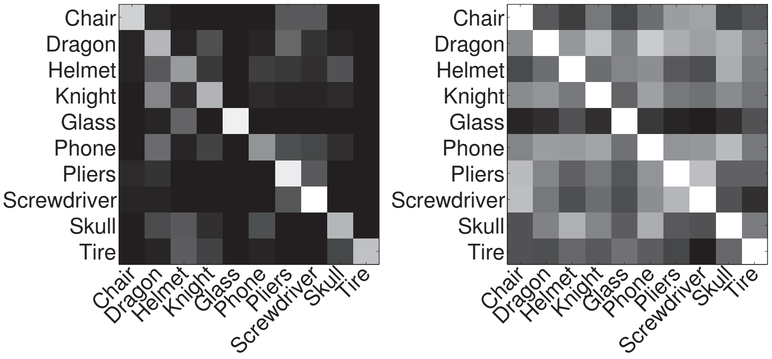


(a) Confusion (b) Affinity

Figure 4.9: Confusion matrices for MNTI at 6×6 resolution and 10% covering thickness. Confusion shows the probability weight associated with each object identity after 100 sensor readings when the true state is given by the row. Note that this relates more to classification confidence than to the graph of classification errors, which is coarser and sparser but shows the same general trends. Affinity shows the correlation between the histogram representations of each object used for recognition, independent of the test data.

under other sensor configurations.

Polar-Fourier and MNTI performed consistently and about equally well in nearly all our tests, despite having rather different formulations. Both, however, make use of the magnitudes of Fourier coefficients to obtain invariance with respect to a class of image transformations (and therefore the corresponding physical transformations). Despite the fact that there is a significant loss of information in this process, the invariance gained seems to consistently increase performance. Confusion matrices are shown for these MNTI and PF in Figures 4.9 and 4.10 respectively. With both descriptors, the Glass model is the most distinctive, while Dragon appears to be most confusing. This is most likely due to the Dragon's large variety of features, since it appears as a possible match for most other models, but those models do not as often appear as matches for Dragon. The Pliers and



(a) Confusion (b) Affinity

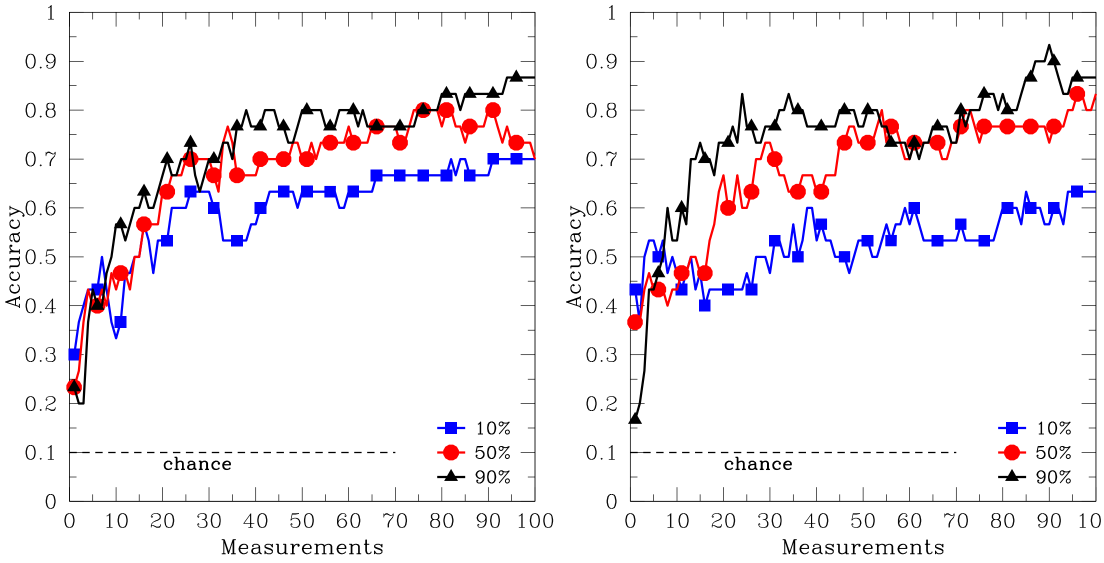
Figure 4.10: Confusion matrices for PF at 6×6 resolution and 10% covering thickness. The format is the same as Figure 4.9.

Screwdriver also appear particularly similar to each other, as both contain many long thin features.

4.4.2.3 Exploration Strategy

We also examined the trade-off between global and local exploration by varying a parameter *L*, to select between the two exploration strategies described in Section 4.2.1. *L* defines the probability at each iteration of exploration of choosing the local exploration strategy, with the remaining probability assigned to the global strategy. Figure 4.11 shows performance under three different values of *L*.

In the ideal case, a random global exploration such as that provided by our sampler seems optimal for appearance-only recognition, as it provides the least biased estimate of the true distribution of object surface appearances. Several practical considerations make



(a) MNTI

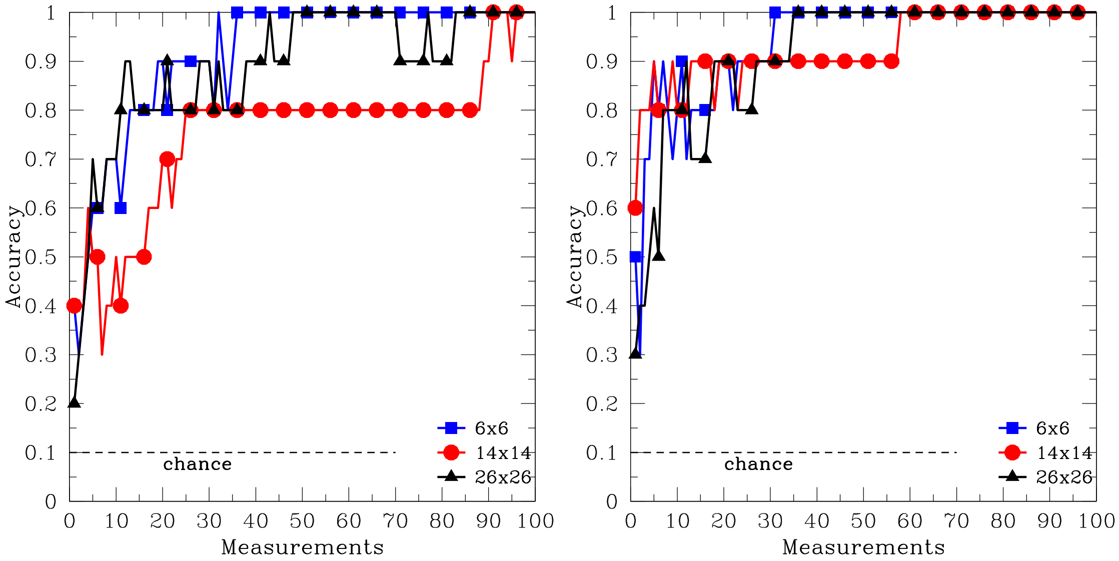
(b) Polar-Fourier

Figure 4.11: Comparison of different exploration strategies by varying L on Princeton set at 6×6 resolution and 10% covering thickness.

this approach infeasible in general though.

First, measuring only the number of sensor readings taken ignores some of the real costs of collecting those measurements. For real robots, randomly sampling the surface of an object is significantly more expensive than focusing on a local area, in terms of time and energy required to move the manipulator between the positions at which each measurement is taken. Additionally, constraints imposed by robot kinematics and collision avoidance restrict the positions and orientations the sensor can reach.

As a result of the above restrictions, the measurements available to the recognition algorithm represent a biased estimate of the true distribution of surface appearance. In our case, local exploration was more fruitful than global, (as can be seen by the stronger



(a) Moment-Normalized

(b) Polar-Fourier

Figure 4.12: Performance of MN and PF on validation data for the tests of Section 4.4.2.6. The validation data were collected using the sampler of Section 4.4.1.2, whereas test data were collected using the full planning algorithm of Section 4.2. Performance on the validation data, approximating a uniform sampling of the object surface, is much stronger than that achieved with exploration.

performance with high values of *L* in Figure 4.11), suggesting it produced a less biased estimate. This is probably because the global strategy tends to have a greater bias toward sampling the side of the object facing the initial position of the robot.

One thing we wish to stress, however, is that when good coverage of the object is available, giving an accurate estimate of the true appearance distribution, our method exhibits much stronger performance. During training, for example, nearly every model perfectly classifies the validation set, which is collected by the same method as the training data (but has no overlap with it), using very few samples. For a representative example, Figure 4.12 shows the performance of the Moment-Normalized descriptor on validation data at the 6×6 and 26×26 resolutions, which is significantly stronger than on the data from exploration, as shown in Figure 4.15(a). Undertaking a real blind exploration process makes the problem substantially more difficult, and the performance effects of varying L show how important the exploration can be to the overall recognition process.

4.4.2.4 Object Perturbation

Fig 4.13 shows recognition performance where the object pose is perturbed by a small amount (up to 10 degrees in orientation and 10% of the object width in translation, simultaneously) each time a sensor reading is taken, compared to the standard case where the object is fixed. As the results indicate, the exploration and recognition process remains effective even if the object pose is perturbed after each sensor reading.

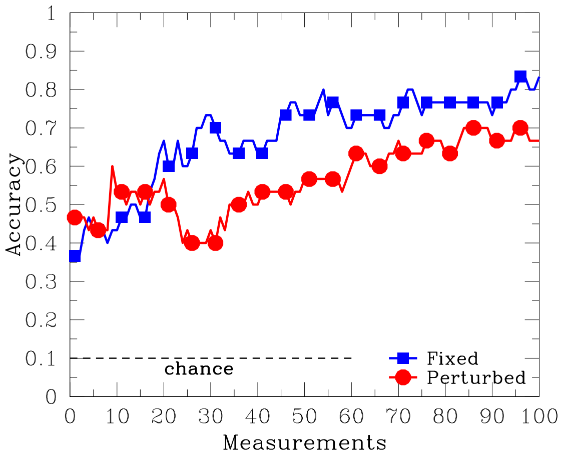
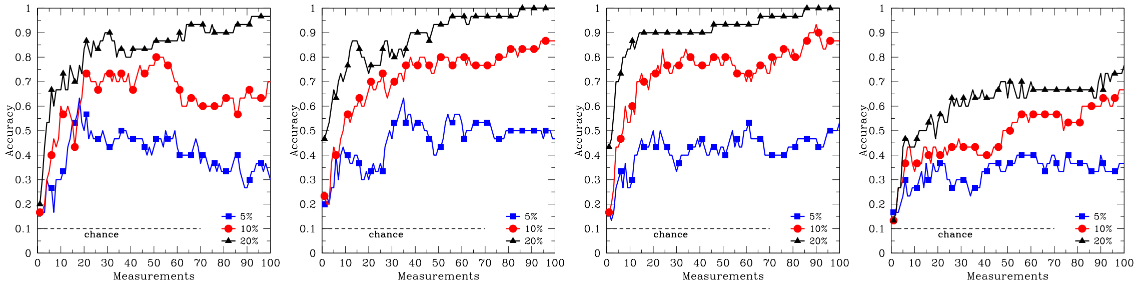
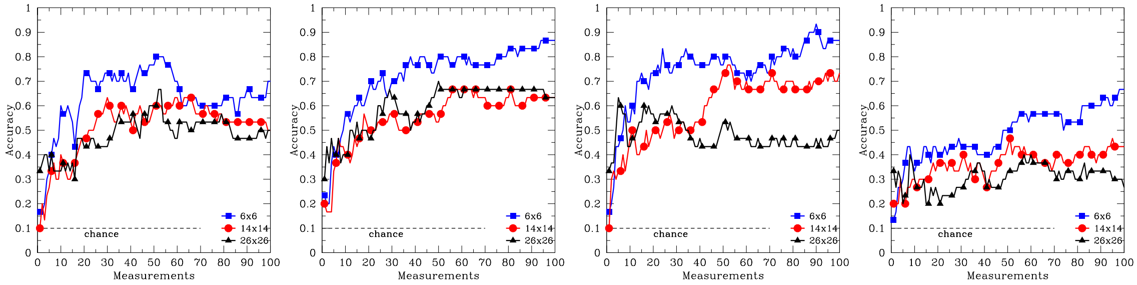


Figure 4.13: Comparison of performance when object is fixed to when object pose is perturbed each time a sensor reading is taken.



(a) Moment-Normalized (b) MNTI (c) Polar-Fourier (d) Vectorize

Figure 4.14: Performance of 3 top descriptors and Vectorize on Princeton set while varying covering thickness.

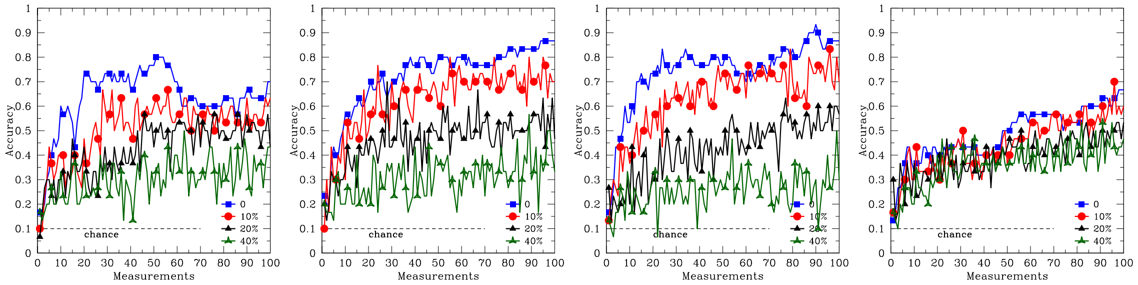


(a) Moment-Normalized (b) MNTI (c) Polar-Fourier (d) Vectorize

Figure 4.15: Performance of 3 top descriptors and Vectorize on Princeton set while varying sensor resolution.

4.4.2.5 Varying Covering Thickness

Next, the effects of varying the thickness of the sensor's covering were investigated. In simulation, changing the covering thickness has two effects: thicker coverings increase the "viewing volume" of the sensor, allowing the detection of larger ranges of depths; they also increase the variance of the Gaussian point spread function associated with the covering, resulting ultimately in blurrier images. We would expect the former effect to help



(a) Moment-Normalized (b) MNTI (c) Polar-Fourier (d) Vectorize

Figure 4.16: Performance of 3 top descriptors and Vectorize on Princeton set with different levels of additive noise.

performance, whereas the latter should be detrimental.

The simulation results are shown in Figure 4.14. It seems that the benefits of a larger viewing volume far outweigh any drawbacks from the point spread, as recognition rates are consistently higher with thicker coverings using any descriptor.

4.4.2.6 Varying Sensor Resolution

The results of varying the resolution of the sensor are shown in Figure 4.15. Three resolutions were chosen to correspond respectively to the PPS sensors (6×6), the rough sensing resolution of the human finger over an equivalent area, based on the density of Merkel receptors in the fingertip (14×14), and the sensing density of a high-resolution resistive sensor available from Tekscan (TekScan, 2011) over that area (26×26).

Surprisingly, these results show that increasing the sensor resolution does not generally increase performance in this framework with any of the descriptors tested. In fact, high resolutions often hurt performance. We believe this is due to the highly non-linear process of the discretization of the tactile image signal, particularly under the effects of small translations.

In fact, consider the situation of comparing two tactile images, *A* and *B*, of nearly the same area of an object's surface, but there is a small displacement in the sensor position where *A* and *B* were taken. At low resolutions, small translations of the sensor with respect to the object surface result in little change to what portion of the surface lies within the area of a single sensing element. At high resolutions, however, a small translation can cause each individual pixel to be sensing a completely new patch of surface. When comparing *A* and *B*, therefore, one would expect low-resolution versions to be more strongly correlated on a pixel-by-pixel basis than high-resolution versions of the same images.

These translation effects can be mitigated in the handling of the images, but at the obvious cost of increased complexity. One place to address the issue is in the choice of the descriptor to use. The MNTI descriptor was derived from MN to be robust to translation effects. Indeed, this descriptor shows less of a decrease in performance than MN or PF as resolution increases, but the effect remains, and it still dominates any gains from the increased information content of these higher-resolution images.

4.4.2.7 Robustness to Noise

Figure 4.16 shows the performance of the top 3 descriptors under the influence of noise. During training and testing, each tactile image was corrupted with uniformly-distributed zero-mean additive noise, with magnitude equal to 10%, 20%, or 40% of the sensing range.

e.g., for values normalized to the sensing range, an input value of 0.5 may range from 0.3 to 0.7 after applying 40% additive noise. Additionally, the performance under noise-free conditions is included as “0”.

All descriptors clearly suffer from the effects of noise. The effects on performance are also quite sporadic, as can be seen from the choppiness of these graphs, as compared to the preceding ones. The general trends in performance also remain the same as the level of noise increases.

4.4.3 Convergence of Surface Contact Controllers

Convergence characteristics of the surface contact controllers were tested in simulation on the Dragon model, with results shown in Figure 4.17. Both the percentage of approaches in which the controllers successfully converged and the average number of iterations required for successful convergences were recorded under various levels of noise. Noise was added in the same manner as in Sec. 4.4.2.7, ranged as a percentage of the total observed force range. All tests were conducted on the Dragon model (see Figure 4.6(h)), due to its variety of interesting surface features. At each noise level, sensor readings were collected until 100 successful convergences. The controllers continued to consistently converge with noise levels as high as 75% and did not begin to have large failure rates until the noise level exceeded the force signal. The number of iterations required for convergence increased steadily with noise levels above 50%. Perhaps surprisingly, small levels of noise improved both convergence rate and time required over the noise-free case.

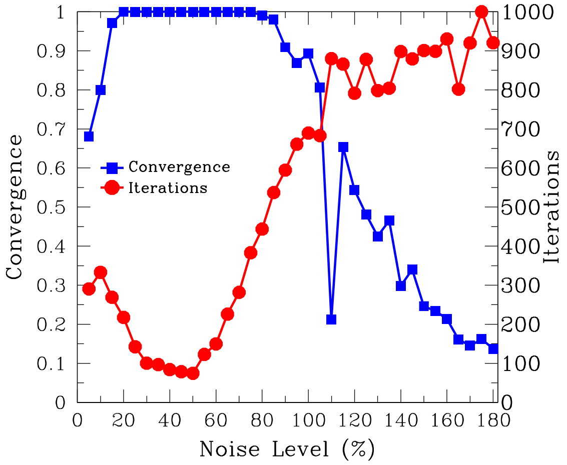


Figure 4.17: Convergence characteristics of local controllers on Dragon model.

4.4.4 Evaluations with Raised Letters

The next set of evaluations attempts to differentiate the same set of raised letters as in the test of the geometry-only method of Chapter 3, shown again for convenience in Figure 4.18, using our DigiTacts sensor system (PPS, 2008). The letters were approximately 2.5 cm per side, while the portion of the sensor being used was approximately 1.2 cm square. Evaluations were conducted with both physical and simulated versions of this system.

4.4.4.1 Simulation

We began, again, with simulations to confirm that the trends observed in the Princeton set still applied to a set of objects with different geometric properties. Simulated letters were generated using a font that was chosen to closely resemble that of the physical letters.[3]

[3] The font in Magenta and Triantafyllakos (2008) was used for all letters except capital "I", for which the font in 1001 Free Fonts (2010) was used, because it had cross-bars as in the physical letters.

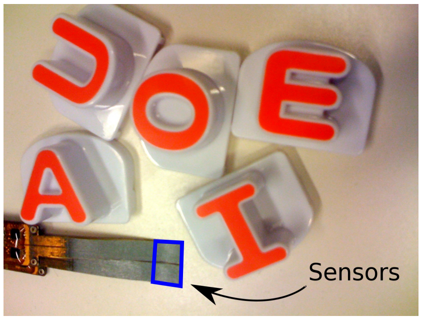


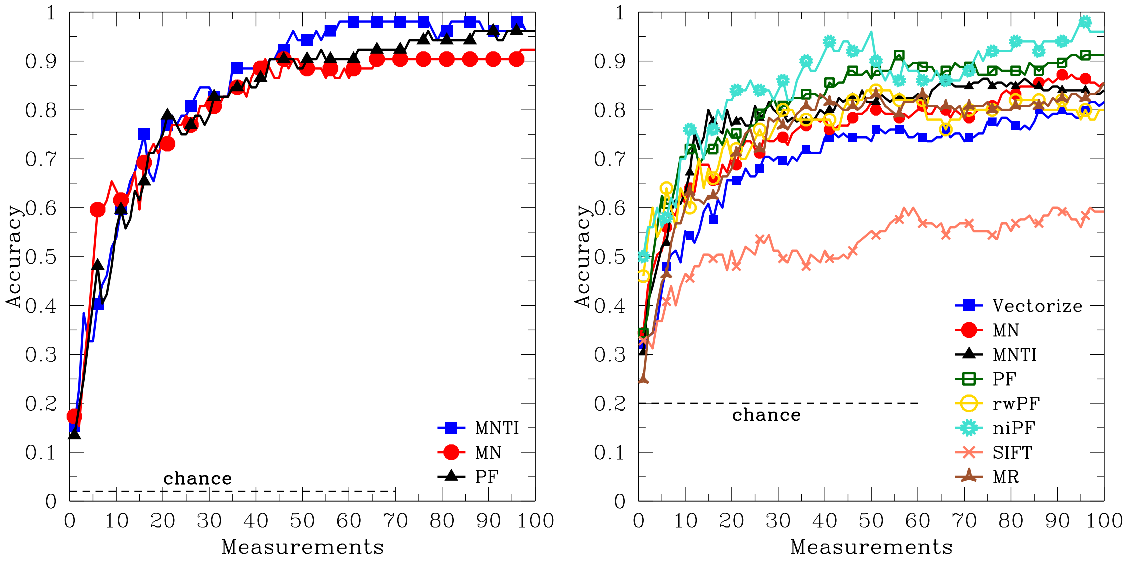
Figure 4.18: Image of the capital vowels from the set of raised letters used in the evaluations of section 4.4.4, alongside the PPS DigiTacts sensors, with the sensing area highlighted in blue.

As before, we used a training set of 1000 images plus an evaluation set of 100 images, then tested on a separate set of 100 images. This time, however, the robot was restricted to approach only from above the letter models. Since we were not focusing on the exploration process in this case, the sampler in Section 4.4.1.2 was used to collect all readings.

Using the same methodology as mentioned previously, we learned models for all 52 upper-case and lower-case letters. Figure 4.19(a) shows the results of this training and testing, using the three top descriptors from before. All three achieve over 90% accuracy, with PF and MNTI, again, outperforming MN with over 95% accuracy each. Performances appear to have converged to their asymptotic values as a function of number of samples at around 60 samples.

4.4.4.2 Physical Sensors

The effectiveness of the framework was also tested on physical sensor readings with our DigiTacts sensor system. For the experiments with the real sensors, only a subset of



(a) Simulation

(b) Physical Sensors

Figure 4.19: Performance of 3 top descriptors as a function of number of samples, on (a) simulated raised letter recognition, and (b) with the physical DigiTacts system.

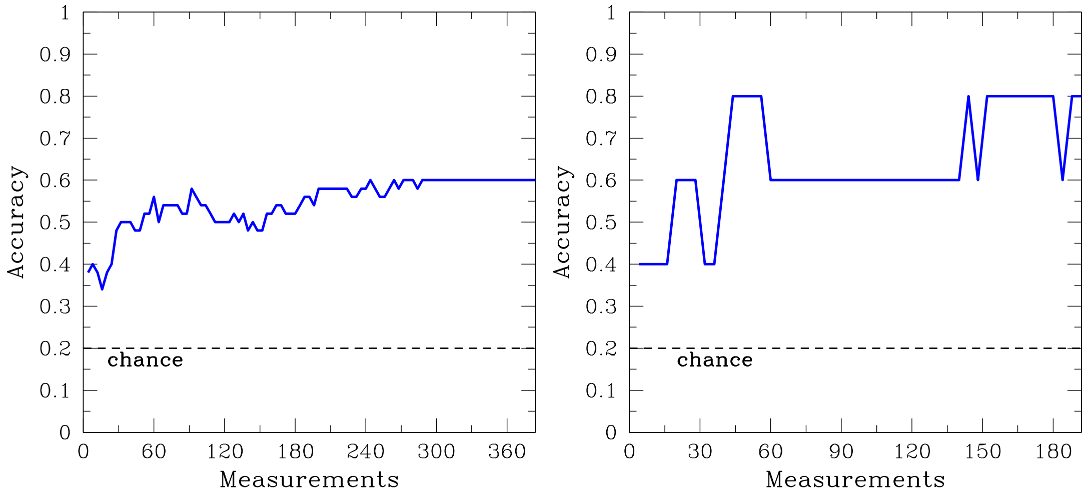
the alphabet was used, due to the time required to emulate the data collection of a robotic system. In particular, the subset consisted of the uppercase vowels, A, E, I, O, and U. A mechanical system was designed to keep the letters level with the sensors, while applying a uniform load at 16 regular positions with 12 angles of rotation. This entire set of configurations was repeated two times to collect a total of 384 readings for each letter.

These 384 readings were then pruned of those configurations for which that particular letter did not make contact with the sensor and post-processed to normalize for the differences in responsiveness of the individual sensor elements identified in our calibration process, as described in Chapter 2, Sections 2.2 and 2.3. Then the remainder were randomly divided into training, validation, and testing sets of size 200, 50, and 100 readings per letter respectively, and the same training and testing process as above was used. In order to avoid the results being too skewed by the small sample sizes, performances were averaged over 7 trials of this full division, training, and testing process. The results are shown in Figure 4.19(b). Note that the training sets are still much smaller than those that were available in the simulation evaluations.

In this test, the performance trends are similar to those in the simulation tests, but the niPF, MR and Vectorize descriptors do better than before, performing nearly as well as the three novel descriptors. We believe this is due to inconsistencies in the response of individual sensor elements that were not characterized sufficiently well in our calibration process. All of the other descriptors make the assumption that each sensor element responds identically (after post-processing) in the course of their respective ways to add rotation-invariance. Using Vectorize, however, the response of each element appears in the same location in the resulting descriptor, allowing the system to learn these inconsistencies. Since MR produces multiple feature vectors based on different portions of the image, it also allows the system to pick up on these trends. niPF appears to benefit from the additional invariance introduced by not relating adjacent rows, emphasizing the usefulness of invariance for achieving robustness.

4.4.4.3 From Simulation to Reality

Preliminary tests suggest that it is feasible to learn models of objects by exploring simulated versions of them, then apply those models to recognizing the physical objects



(a) Average Performance

(b) Single Trial

Figure 4.20: Performance when recognizing the physical letters using a model trained on simulated exploration data. (a) shows the performance averaged over several trials testing on different orderings of the physical data. (b) shows the results of a single trial, using the ordering in which the sensor readings were collected. This trial demonstrates stronger performance on a more accurately modeled subset of the data, where only O and U are sometimes confused.

using real sensors. This capability could allow the recognition of previously unencountered objects, provided that a 3D model of the object is available, as well as avoiding the time-consuming process of fully exploring the object with a real robot.

4.4.4.3.1 Results

Figure 4.20 shows the results of recognizing the letters using test data taken from the data set of Section 4.4.4.2 with a model trained in simulation. Performance is shown for sensor readings presented to the recognition algorithm in the order they were collected, as well as averaged over several trials where the order of presentation was randomized.

Recognition rates peak at 80% recognition, but there are large fluctuations because there are only 5 objects. While there is potential for improvement, these results demonstrate recognition performance substantially above chance on objects that had never been physically sensed.

4.4.4.3.2 BRIDGING THE GAP

Some additional steps were necessary to bridge the gap between the simulated and real worlds. During the training process, the simulated tactile images were corrupted with noise to account for the greater variance of the response of the real sensors. Uniform, independent, identically-distributed additive noise was applied on a per-element basis, with magnitude on the order of 30% of the observed force range. Some post-processing was also applied to the physical sensor images. The response of each element was replaced by its square root to account for two effects: First, the displacements being applied to the sensors may have been slightly above the range in which the force response can be estimated as linear; they could be better explained by a quadratic relationship, so the square root adjusts for this difference. Second, in our mechanical system, the physical sensors were not always as flush with the object surface as the converged position of the sensor in simulation, so this adjustment mitigated the biases introduced by this surface misalignment. Finally, a small Gaussian blur was applied to each physical tactile image to minimize the effects of inconsistencies and non-uniformities in the real sensor response.

4.5 Discussion

In this chapter, we presented a method for characterizing 3D objects using local tactile-appearance features, along with techniques for exploring unknown objects to collect data on such features using tactile force sensors. This work established a strong link between exploration (action) and information in the domain of haptic perception. Simulations showed the method's strong performance on simulated data and the effects of varying several algorithm parameters. The algorithm was found to perform best using sensors with low spatial resolution and a thick, soft covering material. Two novel image descriptors, Polar-Fourier and MNTI, were developed, and both were shown to perform well in a range of situations. An exploration algorithm favoring local over global search was found to produce more consistent and higher-quality results. The simulations also indicated that the exploration and recognition remain effective even when the unknown object is perturbed after each sensor reading. Finally, we demonstrated the method on real-world data using a set of raised letters, along with recognition tests on simulated versions of these letters for comparison. Preliminary results for applying models learned in simulation toward recognition of the real-world objects show promise for the generation of tactile appearance models applicable in the physical world for any object of which one has a 3D model. In addition to allowing great savings in robot time, this capability provides support for cross-modality learning for recognition. For instance, a 3D model of an object could be acquired from vision, yet it could still be used for recognition in the tactile domain.

Opportunities for future work include extending the notion of appearance to deal with multiple sensors and contact locations, or sensors of larger extent with potentially irregular geometry, such as those embedded in the fingers and palms of robotic hands. Our simulator could be integrated into a planning system that could optimize exploration for a real robot to balance benefits and costs, such as expected information gain for a given exploratory procedure and the required time or energy. Independent of improvements to the fidelity of sensor simulation, further work could be done to increase performance on real sensors by investigating descriptors that are more robust to inconsistencies in the sensor response, such as those described in Section 2.6.2.

Within the context of this dissertation, the work has established a set of tools for characterizing object appearance. These could then be applied to a framework that also makes use of geometry information to characterize the spatially-varying surface texture of objects to build even richer haptic models of objects, as presented in Chapter 5.

4.A Appendix: Maximum-Likelihood Interpretation of Identity Estimate

As mentioned in 4.3.1, the best-matching object identity, C, can equivalently be taken as that which maximizes the likelihood of the observed data:

$$C = \arg\max_{j} \Pr(\mathbf{Z} \mid \mathbf{O}_j) \tag{4.18}$$

For each class, this likelihood can be computed as the probability of observing each feature independently, i.e.,

$$\Pr(\mathbf{Z} \mid \mathbf{O}_j) = \prod_{\ell=1}^{n_U} \Pr(\mathbf{C}(\mathbf{d}(z_\ell)) \mid \mathbf{O}_j) \tag{4.19}$$

Setting k_i to the number of observed features associated with each cluster, we can factor the above into the the components corresponding to each cluster by expanding and regrouping:

$$\Pr(\mathbf{Z} \mid \mathbf{O}_j) = \prod_{i=1}^{n_C} \Pr(\mathbf{c}_i \mid \mathbf{O}_j)^{k_i} \tag{4.20}$$

In practice, we are given a histogram representing $\Pr(\mathbf{c}_i \mid \mathbf{Z})$. However, this is simply a multinomial from which we can compute the expected number of features observed from cluster $\mathbf{c}_i$ as $k_i = n_U \Pr(\mathbf{c}_i \mid \mathbf{Z})$. Substituting this into (4.20) gives

$$\Pr(\mathbf{Z} \mid \mathbf{O}_j) = \prod_{i} \Pr(\mathbf{c}_i \mid \mathbf{O}_j)^{n_U \Pr(\mathbf{c}_i \mid \mathbf{Z})} \tag{4.21}$$

Taking the log of both sides yields

$$\log \Pr(\mathbf{Z} \mid \mathbf{O}_j) = n_U \sum_{i} \Pr(\mathbf{c}_i \mid \mathbf{Z}) \log \Pr(\mathbf{c}_i \mid \mathbf{O}_j) \tag{4.22}$$

Dropping the n_U term, which is fixed over the optimization, would therefore give a notion of the “average log likelihood” of a data point, independent of the amount of data observed. Maximizing this quantity is equivalent to minimizing (4.4).

Chapter 5

Combining Geometry and Appearance

After the experiments and analysis of Chapters 3 and 4, we have two complementary approaches to tactile recognition, one based on a geometric model of objects and one making use entirely of appearance information. The goal of the work in this chapter is to use what we have learned from each method to create a new method that makes use of combined geometry and appearance information.

Recall from Section 3.5 that the main limitation of the geometric method was in its ability to scale to 3D space, due to the increased computational demands of maintaining a probability distribution over a six-dimensional pose space. With the approach of Chapter 3, the probability distribution is represented by a set of samples (using either a particle filter or histogram filter), and each measurement update requires evaluating the likelihood of the measurement in the pose corresponding to each of these samples. Depending on the precision requirements and the smoothness of the underlying distribution, the number of samples necessary to effectively cover the space can increase drastically as its dimensionality grows. Chapter 4 has now given an additional interpretation of each tactile image though, providing some additional information that can be brought to bear on the problem. In this chapter, we build a map of object surfaces that describes how appearance changes over its surface, which we refer to as a characterization of the object's **spatially varying appearance (SVA)**. With this additional information, we can update our probability distribution much more efficiently by modeling a slightly different probability.

The key change is to the measurement update, and it can be thought of as follows:

- With the geometric method, one must evaluate the probability of a given measurement by considering whether it could be produced given a set of hypothesized object identities and poses. These hypotheses need to cover all the combinations of object identity and pose that are plausible based on the information received up to that point.
- With the new approach, we instead consider the set of all possible locations on the surface of particular objects that could have given rise to the observed sensor reading. Then these locations are combined with the equivalents from all the previously observed measurements to further constrain the object's identity and pose.

We first go over the algorithmic underpinnings of the approach in Section 5.1.2, then the application to Bayes filters is explained in Section 5.2. The formulation and implementation of the map-like data structure we use to support recognition are presented in Section 5.3, and the use of that structure to update the state probability distribution is covered in Section 5.4.

5.1 Comparison to Prior Work

Other vision researchers have incorporated some form of geometry information with their local descriptor representation (Fergus et al., 2003; Lazebnik et al., 2006; Cao et al., 2010). Our approach differs fundamentally from these methods, since ours not only encodes geometric statistics but also supports recovery of the object pose. There are two areas of prior research that formed the basis for the method presented in this chapter and thus merit a thorough comparison. First, we discuss the seminal work of Grimson and Lozano-Perez (1984); then the use of techniques from geometric hashing is covered.

5.1.1 Comparison to Interpretation Trees

Grimson and Lozano-Perez (1984) presents a recognition algorithm that uses tactile sensor readings to identify and localize unknown objects. In their formulation of the problem, each sensor reading provides a contact point and a surface normal. The surface points are assumed to have an uncertainty that is bounded by a given volume, while the surface normals have a corresponding cone of uncertainty. Objects are modeled as polyhedra, and the recognition process consists of generating interpretation trees that are consistent with the set of sensor readings that have been received so far. The interpretation tree is essentially a mapping between sensor readings and faces of the polyhedral object model with which the sensor may have been in contact. This tree is pruned using a set of inexpensive geometric tests based on each of the individual sensor readings before a consistent object identity and pose is computed for each remaining interpretation.

The approach presented in this chapter shares many characteristics with this work, but also has a couple very important differences. Though its structure is rather different than that of Grimson and Lozano-Perez, we also maintain a mapping between sensor readings and object identities and poses. As sensor readings are received, the set of consistent hypothesized object identities and poses is computed and pruned. Arbitrary 3D objects are handled, and they are represented as polygonal models in our simulator. The process by which this mapping is computed is very different in the SVA approach, though:

- The SVA method incorporates information about tactile appearance. The mapping from sensor readings to possible poses uses associations to regions on the object surface defined by their appearance properties, rather than associations to faces of a polygonal model. This additional source of information results in a significant reduction in the number of potential matches.

- The constraints imposed in the SVA approach are based on pairs of sensor readings throughout the process.

- All of these associations are probabilistic in nature, thus allowing for distinctions between the likelihoods of different hypotheses and, importantly, facilitating explicit modeling of noise and inconsistencies in the sensing process.

As a result of these differences, the SVA approach is able to apply more information to the problem, more discriminatingly distinguish between multiple possible (but perhaps unlikely) hypotheses, and more robustly deal with real-world complications.

5.1.2 Geometric Hashing

Our approach was inspired in part by the geometric hashing algorithm. A review of geometric hashing techniques provided in Wolfson and Rigoutsos (1997). Geometric hashing provides a fast way of matching and identifying the pose of a set of points taken from a previously seen point cloud with an unknown rigid body transformation applied. The main idea of this approach is to transform the set of points making up the current object model to be relative to a basis composed of some of the points themselves.

5.1.2.1 Standard Geometric Hashing

In the 3D case, the training process consists of selecting, in turn, each set of three points from the explored object, $\mathbf{O}_i$, and constructing a basis, B, for the 3D space from them (assuming they are not all collinear). A tuple $T_{\mathbf{O}_i,B}$ is then generated, containing the set of points comprising this basis and a reference to the object model of which it is a part. The 3D space is quantized into a 3D grid of appropriate resolution, and a binning function $b(p)$ is defined in the usual way to assign a 3D point to the corresponding bin of the grid. After being projected into this basis, each object point, p_j, places an instance of the tuple, $T_{\mathbf{O}_i,B}$ into the bin $b(p_j)$. So, for a model consisting of N points, all N points will be binned for each of the $\binom{N}{3}$ bases that can be formed, and the same procedure is applied to each object to be recognized. The resulting database can be quite large, particularly for detailed models.

The trade-off for this space complexity comes in the speed of matching an unknown object. During the recognition stage, three points of the explored model may be chosen at random to form a basis as before. Once again, all remaining points in the query model are projected into this basis and binned. Then a Hough-transform-like vote-casting process ensues. For each model point, p_i, every tuple in bin $b(p_i)$ casts a vote for its model and basis. These votes are accumulated and the model-basis tuples receiving the most votes are taken as the best candidate matches. The positions of the basis points in the original model coordinate system provide a set of correspondences to compute a rough rigid body transformation to align the point sets, and then the match is passed on to a more discriminative method to verify or refine the match.

5.1.2.2 Applied to the Tactile Case

In our situation, our measurements are much more informative than just contact point locations, so we can take advantage of this extra information to greatly improve upon the efficiency of the standard geometric hashing approach. Though two contact points are not sufficient to define a basis in a three-dimensional space, they can be used to constrain a transformation up to one degree of freedom of uncertainty; our contact points also have associated surface normal estimates, which can in many cases be used to constrain this last degree of freedom. Additionally, we have tactile images associated with each point, giving it an appearance signature that can be used to distinguish individual points. We therefore extend the geometric hashing algorithm by incorporating this additional information as probabilistic constraints, maintaining the advantage of fast lookup times while reducing space requirements from $\binom{N}{3}$ to $\binom{N}{2}A^2$, where A is a factor of the ambiguity of appearance of a surface patch, described in Section 5.3.2.

5.2 Spatially Varying Appearance (SVA) and Bayes Filters

The geometric method of Chapter 3, as described in Section 3.1.1.2 and then in detail in Section 3.3.2, requires sampling the function $\Pr(z_t \mid x_t, \mathbf{M})$ for many hypothesized values of x_t. As we move to full three-dimensional models, however, the state space of x_t grows to six continuous dimensions, plus the discrete dimension for object identity. Because of the so-called "curse of dimensionality", the number of particles or histogram bins needed to adequately cover this space quickly becomes unmanageable, particularly if precise localization is required.

For this reason, we instead model the related probability $\Pr(x_t \mid z_t, \mathbf{M})$ by leveraging some of our new-found knowledge of tactile appearance. Instead of estimating, for every possible pose and object identity, what the likelihood of a given measurement is, we estimate what set of poses and object identities could explain the sensor reading. We are able to greatly decrease the complexity of modeling the space of all possible sensor measurements by making use of the appearance classes introduced in Chapter 4.

5.2.1 Belief Update

It is useful at this point to introduce a bit more notation to explain how this new quantity fits in to the standard Bayes filter update equations. Traditionally, (e.g., in Thrun et al., 2005) a Bayes filter's estimate of state is denoted $bel(x)$ for "belief", and it is updated as

$$\overline{bel}(x_t) = \int \Pr(x_t \mid u_t, x_{t-1}, \mathbf{M}) bel(x_{t-1}) dx_{t-1} \tag{5.1}$$

$$bel(x_t) = \eta \Pr(z_t \mid x_t, \mathbf{M}) \overline{bel}(x_t) \tag{5.2}$$

where η is a normalization constant. Note that the same η will be freely used in different equations in this manner where it may represent different values. Applying Bayes rule to Equation 5.2, we get the new update rule

$$bel(x_t) = \eta \frac{\Pr(x_t \mid z_t, \mathbf{M})}{\Pr(x_t, \mathbf{M})} \overline{bel}(x_t) \tag{5.3}$$

with η once again as a (different) normalization constant. Now consider this denominator. The object identity and pose can be considered independent of the maps, allowing us to break this term into $\Pr(x_t, \mathbf{M}) = \Pr(x_t)\Pr(\mathbf{M})$. As we have set up the problem in this application, the prior on object identity and pose, $\Pr(x_t)$ is static, since the object is fixed, so the time subscript can be ignored. Moreover, $\Pr(x_t)$ is assumed to be uniform, which allows both these terms to be folded into the normalization constant to get

$$bel(x_t) = \eta \Pr(x_t \mid z_t, \mathbf{M}) \overline{bel}(x_t) \tag{5.4}$$

So comparing Equations 5.2 and 5.4, one can see that the updates using $\Pr(x_t \mid z_t)$ are made in the same way as in the original formulation. In order to take this last step in the case of changing state (i.e., when manipulation may change the object pose), one must also assume that these motions themselves are not informative; e.g., the object must not be able to be pushed into a corner. Otherwise, an estimate of $\Pr(x_t, \mathbf{M})$ must be maintained.

The net effect of these manipulations is that the belief can be updated through modeling and evaluating $\Pr(x_t \mid z_t, \mathbf{M})$ instead of the usual $\Pr(z_t \mid x_t, \mathbf{M})$. This change is computationally advantageous since traditional means of modeling the latter require its evaluation for each observed z_t, for every plausible hypothesized x_t; the former need only be evaluated for the observed values of z_t, and the resulting values of x_t for which $\Pr(x_t \mid z_t, \mathbf{M})$ is significantly nonzero is sparse. As will be shown, finding these sparse values, much like in geometric hashing, amounts to a set of table lookups and computations of rigid body transformations with associated uncertainty.

5.2.2 Measurement to Pose Mapping

Now we can break down the $\Pr(x_t \mid z_t, \mathbf{M})$ term to make use of both appearance and geometry information simultaneously. We will do that by considering the constraints imposed by pairs of sensor readings we have seen so far. With this approach, the mapping from measurements to states can be evaluated as

$$\Pr(x_t \mid z_t, \mathbf{M}) = \Pr(x_t \mid \{\text{pair}(z_i, z_t), \text{pair}(z_t, z_i);\ i = 1, .., t-1\}, \mathbf{M}) \tag{5.5}$$

Without loss of generality, let the $n_P = 2(t-1)$ pairs in the set in Equation 5.5 be numbered pair_1 to pair_{n_P} (ordering does not matter). Then we can rewrite Equation 5.5 as

$$\Pr(x_t \mid z_t, \mathbf{M}) = \Pr(x_t \mid \{\text{pair}_i;\ i = 1,..,n_P\}, \mathbf{M}) \tag{5.6}$$

$$= \Pr(x_t \mid \text{pair}_{n_P}, \{\text{pair}_i;\ i = 1,..,n_P - 1\}, \mathbf{M}) \tag{5.7}$$

$$= \frac{\Pr(\text{pair}_{n_P} \mid x_t, \{\text{pair}_i;\ i = 1,..,n_P - 1\}, \mathbf{M}) \Pr(x_t, \{\text{pair}_i;\ i = 1,..,n_P - 1\}, \mathbf{M})}{\Pr(\text{pair}_{n_P}, \{\text{pair}_i;\ i = 1,..,n_P - 1\}, \mathbf{M})} \tag{5.8}$$

$$= \frac{\Pr(\text{pair}_{n_P} \mid x_t, \mathbf{M}) \Pr(x_t \mid \{\text{pair}_i;\ i = 1,..,n_P - 1\}, \mathbf{M}) \Pr(\{\text{pair}_i;\ i = 1,..,n_P - 1\}, \mathbf{M})}{\Pr(\{\text{pair}_i;\ i = 1,..,n_P\}, \mathbf{M})} \tag{5.9}$$

Equation 5.8 follows by applying Bayes' rule. Equation 5.9 asserts that pairs are conditionally independent given the state and expands the joint probability $\Pr(x_t, \{\text{pair}_i;\ i = 1,..,n_P - 1\}, \mathbf{M})$. Bayes' rule can be applied again to get

$$\begin{aligned} &\Pr(x_t \mid z_t, \mathbf{M}) = \\ &\quad \frac{\Pr(\text{pair}_{n_P} \mid \mathbf{M}) \Pr(\{\text{pair}_i;\ i = 1,..,n_P - 1\}, \mathbf{M})}{\Pr(x_t \mid \mathbf{M}) \Pr(\{\text{pair}_i;\ i = 1,..,n_P\}, \mathbf{M})} \cdot \\ &\quad \Pr(x_t \mid \text{pair}_{n_P}, \mathbf{M}) \Pr(x_t \mid \{\text{pair}_i;\ i = 1,..,n_P - 1\}, \mathbf{M}) \end{aligned} \tag{5.10}$$

Finally, constant terms can be folded into a single constant, η, and this equation can be expanded recursively to get

$$\Pr(x_t \mid z_t, \mathbf{M}) = \eta \Pr(x_t \mid \text{pair}_{n_P}, \mathbf{M}) \underbrace{\Pr(x_t \mid \{\text{pair}_i;\ i = 1,..,n_P - 1\}, \mathbf{M})}_{\text{recurse}} \tag{5.11}$$

$$= \eta \prod_{i=1}^{n_P} \Pr(x_t \mid \text{pair}_i, \mathbf{M}) \tag{5.12}$$

$\Pr(\text{pair}_{np} \mid \mathbf{M})$ is observed, and $\Pr(x_t \mid \mathbf{M}) = \Pr(x_t)$ was discussed in Section 5.2.1. By applying the recursive expansion of Equations 5.6 through 5.12 to Equation 5.5, we get

$$\Pr(x_t \mid z_t, \mathbf{M}) = \eta \prod_{i=1}^{t-1} \Pr(x_t \mid \text{pair}(z_t, z_i), \mathbf{M}) \prod_{i=1}^{t-1} \Pr(x_t \mid \text{pair}(z_i, z_t), \mathbf{M}) \tag{5.13}$$

Note that the ordering of pairings is significant; $\text{pair}(z_i, z_j)$ is not the same as $\text{pair}(z_j, z_i)$ Estimation of the individual probabilities in Equation 5.13 is driven by our maps **M** in this approach containing information about surface patch pairs acquired during training. As in the approaches of Chapters 3 and 4, the training phase consists of collecting a large number of sensor readings that cover the surface of each object to be recognized. In this approach, the map will contain characterizations of pairs of surface patches from each object along with the identity of that object and their location on its surface. During testing, the identity and pose of the unknown object are then constrained by matching observed pairs of surface regions to those in the map (denoted $\mathbf{m}_{[\cdot]}$), giving

$$\begin{aligned} &\Pr(x_t \,|\text{pair}(z_{a1}, z_{b1}), \mathbf{M}) = \\ &\sum_{\mathbf{m}_{a2}, \mathbf{m}_{b2} \in \mathbf{M}} \underbrace{\Pr\left(x_t \mid \text{match}\left(\text{pair}(z_{a1}, z_{b1}), \text{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2})\right)\right)}_{\text{match constraint}} \cdot \\ &\qquad \underbrace{\Pr\left(\text{match}\left(\text{pair}(z_{a1}, z_{b1}), \text{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2})\right)\right)}_{\text{match likelihood}} \end{aligned} \tag{5.14}$$

First we will discuss the match likelihood term in Section 5.3, then the distributions imposed on the state space by the match constraint term will be covered in Section 5.4.

5.3 Mapping Spatially-Varying Appearance

The purpose of the SVA map is to evaluate $\Pr(\text{match}(\text{pair}(z_{a1}, z_{b1}), \text{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2})))$, the probability that an observed pair of sensor readings corresponds to the same pair of regions on an object surface represented by a pair of entries in the map. It is important to realize that these are regional correspondences rather than point correspondences, since sampling exactly the same point multiple times is unlikely without specialized exploration. This matching, since it makes use of appearance information, therefore depends on the characterization of local appearance to be consistent over a local region; locations nearby a given point on the object surface are assumed to have a similar appearance characterization. This effectively means that the object surface must be sampled densely enough during training for the appearance characterization used to be consistent across regions whose size corresponds to the sampling resolution in the neighborhood. It also implicitly requires that any regions of the object surface that may be encountered in testing must be completely explored (i.e., densely sampled, the requirements of which are discussed in Section 5.6.3) during map-building. The implications of this requirement on appropriate appearance descriptors are analyzed in Section 5.3.3.

5.3.1 Using Surface Patch Pair Statistics

Each sensor reading consists of a tactile image and a 3D translation and orientation of the sensor in the robot frame. Readings are collected in the manner described in Sec-

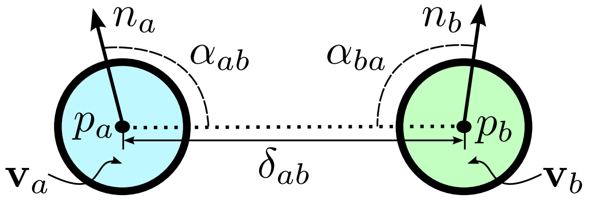


Figure 5.1: Illustration of the relevant geometry for dealing with a pair of surface patches, labeled *a* and *b*: Each has a centroid $p_{[\cdot]}$, appearance feature $\mathbf{v}_{[\cdot]}$ (visualized by color), and surface normal estimate $n_{[\cdot]}$. δ_{ab} denotes the distance between the patch centroids. The angle between each patch's surface normal and the vector between the patches (dotted line) is marked as $\alpha_{[\cdot]}$.

tion 4.2.2, so rotation about the sensor normal is not controlled for. The geometry of a pair of surface patches *a* and *b* is therefore described by the positions of the two patches, p_a and p_b, and their estimated surface normals, n_a and n_b, shown in Figure 5.1.

As in Chapter 4 (Section 4.3.1.1), let us describe appearance through association with appearance classes in the form of clusters $\mathbf{c}_i$ in the space of the appearance descriptors of Section 4.3.2. Each cluster corresponds to an appearance class of physical surfaces that gives rise to measurements with certain characteristics picked out by the descriptor being used. We can then evaluate the likelihood of a measurement being drawn from appearance class *i* as $\Pr(z_t \mid \mathbf{c}_i)$. $\mathbf{c}(\mathbf{v}_j)$ will be taken to represent the set of those likelihoods for the appearance feature $\mathbf{v}_j$ extracted from measurement z_j.

Unknown objects may be encountered in any pose, so the map of surface patch pairs is indexed by quantities independent of the pose. Let $\mathbf{v}_a$ and $\mathbf{v}_b$ be the features describing each patch's appearance. Regardless of the pose of the object, these values and the distance between the points, $\delta_{ab} = ||p_b - p_a||^2$, should remain constant. Since we wish to distinguish pair(z_a, z_b) from pair(z_b, z_a), pairs are indexed by $\mathbf{c}(\mathbf{v}_a)$, $\mathbf{c}(\mathbf{v}_b)$, and δ_{ab}.We originally also included consideration of how well-aligned the surface normal estimates of the patch pairs were. Because of the potential for large uncertainty in these normal estimates though (discussed later in Section 5.4.2), this portion of the matching proved computationally intensive for little gain and was therefore dropped.

5.3.2 Appearance Class Likelihoods

Although Gaussian mixture models proved to be a very effective clustering method for the bag-of-features recognition approach of Chapter 4, it is undesirable in this case to always have associations between a feature and every appearance class encoded in the map. A hard clustering method is excessively restrictive though, as the appearance class of many inputs may be legitimately ambiguous. We therefore opt for a soft clustering approach that only associates a feature with the most likely clusters.

Let the affinity between features $\mathbf{v}_a$ and $\mathbf{v}_b$, $aff(\mathbf{v}_a, \mathbf{v}_b)$ be given by their inner product, $< \mathbf{v}_a, \mathbf{v}_b >$. We use Partitioning Around Medoids (Kaufman and Rousseeuw, 1990) to form n_C clusters from the set of all features acquired in training using $aff(\cdot, \cdot)$, each represented by a medoid med_i, such that each feature $\mathbf{v}_j$ is associated with the nearest medoid by its membership m_j. Affinities of members of a cluster to the medoid were assumed to be distributed roughly as a Gaussian with mean one[1] and variance ψ_i, computed for each

[1] A mean of 1 is used rather than the empirical mean in order to guarantee that likelihood always increases with affinity up to the maximum value of 1.

cluster $\mathbf{c}_i$ as

$$\psi_i = \frac{\sum_j \left(1 - aff(\mathbf{v}_j, med_i)\right)^2 \operatorname{Ind}(m_j, i)}{\sum_j \operatorname{Ind}(m_j, i)} \tag{5.15}$$

where $\operatorname{Ind}(i, j)$ is an indicator function equal to 1 if $i = j$ and 0 otherwise. Then the appearance class likelihoods of each feature are given initially by

$$\Pr(\mathbf{v}_j \mid \mathbf{c}_i) = \frac{1}{\sqrt{\pi \psi_i}} exp\left(-\frac{(1 - aff(\mathbf{v}_j, med_i))^2}{2\psi_i}\right) \tag{5.16}$$

Unlikely matches are then pruned away by setting

$$\text{best}_{i,j} = \max_i \Pr(\mathbf{v}_j \mid \mathbf{c}_i) \tag{5.17}$$

$$\text{Prune}(\mathbf{v}_j \mid \mathbf{c}_i) = \begin{cases} \Pr(\mathbf{c}_i \mid \mathbf{v}_j) & \Pr(\mathbf{c}_i \mid \mathbf{v}_j) > T_a \text{best}_{i,j} \\ 0 & \text{otherwise} \end{cases} \tag{5.18}$$

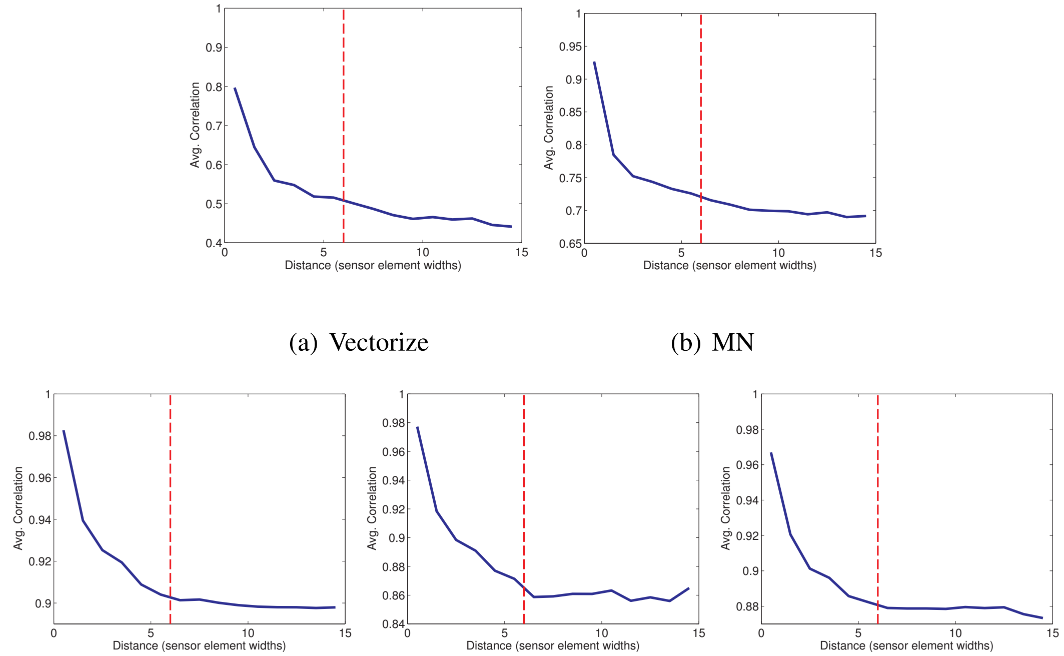
$$\Pr(z_j \mid \mathbf{c}_i) = \eta \text{Prune}(\mathbf{v}_j \mid \mathbf{c}_i) \tag{5.19}$$

In our experiments, T_a was set to 0.75. In the worst case, e.g. if all appearance classes have the same likelihood, this procedure can produce a match for each class, making the appearance ambiguity mentioned in Section 5.1.2.2, A, equal n_C in the worst case; in practice, however, many fewer matches are common. The average number of expected matches can be adjusted through the choice of T_a, though at the risk of pruning away true matches.

5.3.3 Appearance Consistency

Some simple tests were conducted to evaluate the consistency of the novel appearance descriptors from Chapter 4 in nearby locations on the surface of the object. We would like to be confident that the appearance descriptor for a patch of surface represents the appearance not just at that point (which we generally take to be given by the sensor centroid), but also to some degree in a region around that point. Figure 5.2 shows appearance features' correlation as a function of the distance between them. Distance is measured in sensor element widths, using a sensor of size 6-x-6, so distances greater than 6 correspond to comparing non-overlapping portions of the object surface. Since non-overlapping regions by definition do not image any of the same portion of the object surface, one would not expect them to be any more similar in appearance than arbitrarily-far-apart surface patches, except due to smoothness in the variation of appearance across an object. Such smoothness of appearance characteristics varies, of course, from one object to another. For objects without major symmetries or other self-similar regions, though, it is reasonable to expect to see similarities drop for distances above this threshold when using a well-performing appearance descriptor; repeating patterns on an object surface can, however, contribute to this function not decreasing monotonically across all comparators.

A novel appearance comparator, Exhaustive, is introduced here for comparison. It is intended to provide an upper-bound on consistency for correlation-based comparisons. Exhaustive compares tactile image $\mathbf{I}_A$ to tactile image $\mathbf{I}_B$ by computing correlation of a sub-window of $\mathbf{I}_A$, $\mathbf{W}_A$, with the best-matching part of $\mathbf{I}_B$. Cross-correlation is computed between 16 rotated versions of $\mathbf{W}_A$ and $\mathbf{I}_B$, to give invariance to both rotation and translation, and the maximum value is taken as the Exhaustive correlation. In the case of a 6-x-6 sensor, the sub-window was size 4-x-4.



(a) Vectorize (b) MN

(c) MNTI (d) PF (e) Exhaustive

Figure 5.2: Correlation of appearance features as a function of distance for different appearance descriptors. Exhaustive gives an upper bound on consistency that MNTI and PF are quite close to. The dashed red line marks the maximum distance at which sensor readings may image an overlapping area.

The graphs in Figure 5.2 were generated from the training data (1000 sensor readings) for the Dragon model (see Figure 4.6(h)), selected for its wide variety of surface textural properties. The final data point (at position 15) actually averages all points in the model with a distance of at least 15 sensor elements. In order for a descriptor to be both consistent and discriminative, we would like correlations to remain relatively high for small translations, thus ensuring that nearby points are grouped into consistent appearance classes, but smaller for far-away points, promoting a diversity of appearance classes. Although correlations drop significantly from a distance of 0 to a distance of about 10 sensor elements for all comparators, the drops in correlation for Vectorize and MN are distinctly more gradual than the rest, with a long tail that extends well beyond the area of overlap. MNTI and PF can be seen to have a very similar correlation profile (up to scaling of the range of observed correlations) to Exhaustive, all of which exhibit correlation values that largely level off after the non-overlap distance (marked in red); this indicates that these descriptors do not change much when the sensor is translated over small distances (i.e., when sensor readings overlap), but they are still able to discriminate non-overlapping regions, as desired. This behavior is consistent with the design of the MNTI descriptor (see Section 4.3.2.5); interestingly, PF shares the same characteristics, despite its design rationale being different (see Section 4.3.2.5). This phenomenon suggests that it may simply be the degree of invariance built into both descriptors that gives them this behavior, despite the invariance designed for being translational in one case and rotational in the other.

5.3.4 Matching Distances

Distances were matched using a maximum likelihood estimate of the distribution of expected observed distances between points in two surface patches. This estimate was made from from a single observation using kernelized histograms. The range of possible distance values was discretized into a set of n_{DB} uniform regions with distance bin centers $\{\text{bin}D_i\}$ at

$$\text{bin}D_i = \text{minDist} + (\text{maxDist} - \text{minDist})\frac{i+0.5}{n_{DB}} \tag{5.20}$$

and δ_{ab} was associated with the nearest bins through applying a step kernel with diameter the width of one bin to give the likelihood of observing a distance measurement in each range, $\Pr(\delta_{ab}, \mathrm{bin}D_i)$, for each pair of surface patches. This kernel also effectively imposes a distribution on the expected sizes of surface patches.

5.3.5 Putting it all Together

An overall match quality was computed as the joint probability of the observed measurement and map data. Let CA and CB be random variables corresponding to the appearance classes of surface patches a and b respectively for both the prospectively matching pairs and D be another random variable corresponding to the distance between points in the pairs. Marginalizing over appearance and distance classes gives

$$\begin{aligned}&\Pr(\mathrm{match}(\mathrm{pair}(z_{a1}, z_{b1}), \mathrm{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2})))\\&\quad = \sum_{i,j,k} \Pr(\mathrm{pair}(z_{a1}, z_{b1}), \mathrm{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2}), CA = \mathbf{c}_i, CB = \mathbf{c}_j, D = \mathrm{bin}D_k)\end{aligned} \tag{5.21}$$

Then we can convert the joint to a condition on the appearance and distance characteristics to get

$$\begin{aligned}&\Pr(\mathrm{match}(\mathrm{pair}(z_{a1}, z_{b1}), \mathrm{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2})))\\&\quad = \sum_{i,j,k} \Pr(\mathrm{pair}(z_{a1}, z_{b1}), \mathrm{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2}) \mid CA = \mathbf{c}_i, CB = \mathbf{c}_j, D = \mathrm{bin}D_k) \cdot\\&\qquad \Pr(CA = \mathbf{c}_i, CB = \mathbf{c}_j, D = \mathrm{bin}D_k)\end{aligned} \tag{5.22}$$

Since the observed measurements are independent of the map measurements given the appearance and distance classes, we can rewrite this as

$$\begin{aligned}
&\Pr(\mathrm{match}(\mathrm{pair}(z_{a1}, z_{b1}), \mathrm{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2}))) \\
&\quad = \sum_{i,j,k} \Pr(\mathrm{pair}(z_{a1}, z_{b1}) \mid CA = \mathbf{c}_i, CB = \mathbf{c}_j, D = \mathrm{bin}D_k) \cdot \\
&\qquad\quad \Pr(\mathrm{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2}) \mid CA = \mathbf{c}_i, CB = \mathbf{c}_j, D = \mathrm{bin}D_k) \cdot \\
&\qquad\quad \Pr(CA = \mathbf{c}_i) \Pr(CB = \mathbf{c}_j) \Pr(D = \mathrm{bin}D_k)
\end{aligned} \tag{5.23}$$

Now we consider the pairs to be fully described by their appearance and distance characteristics, each of which is independent of the others:

$$\begin{aligned}
&\Pr(\mathrm{match}(\mathrm{pair}(z_{a1}, z_{b1}), \mathrm{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2}))) \\
&\quad = \sum_{i,j,k} \Pr(z_{a1} \mid CA = \mathbf{c}_i) \Pr(z_{b1} \mid CB = \mathbf{c}_j) \Pr(\delta_{a1b1} \mid D = \mathrm{bin}D_k) \cdot \\
&\qquad\quad \Pr(\mathbf{m}_{a2} \mid CA = \mathbf{c}_i) \Pr(\mathbf{m}_{b2} \mid CB = \mathbf{c}_j) \Pr(\delta_{a2b2} \mid D = \mathrm{bin}D_k) \cdot \\
&\qquad\quad \Pr(CA = \mathbf{c}_i) \Pr(CB = \mathbf{c}_j) \Pr(D = \mathrm{bin}D_k)
\end{aligned} \tag{5.24}$$

Alternatively, terms can be rearranged and grouped by characteristic into the form

$$\begin{aligned}
&\Pr(\mathrm{match}(\mathrm{pair}(z_{a1}, z_{b1}), \mathrm{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2}))) \\
&\quad = \sum_{i=1}^{n_C} (\Pr(z_{a1} \mid CA = \mathbf{c}_i) \Pr(\mathbf{m}_{a2} \mid CA = \mathbf{c}_i) \Pr(CA = \mathbf{c}_i) \cdot \\
&\qquad \sum_{j=1}^{n_C} (\Pr(z_{b1} \mid CB = \mathbf{c}_j) \Pr(\mathbf{m}_{b2} \mid CB = \mathbf{c}_j) \Pr(CB = \mathbf{c}_j) \cdot \\
&\qquad \sum_{k=1}^{n_{DB}} \Pr(\delta_{a1b1} \mid D = \mathrm{bin}D_k) \Pr(\delta_{a2b2} \mid D = \mathrm{bin}D_k) \Pr(D = \mathrm{bin}D_k)))
\end{aligned} \tag{5.25}$$

This provides a way to evaluate the probability of each pair of observed points corresponding to pairs of regions in the object maps. The individual summation terms[2] evaluate the probability that the measurements being matched belong to the same appearance and distance classes, summed over all possible class combinations. The appearance likelihoods can be computed by evaluating Equation 5.19, and the distance likelihoods can be obtained from the joint distribution of Section 5.3.4 as $\Pr(\delta_{ab}, \text{bin}D_k) = \Pr(\delta_{ab} \mid \text{bin}D_k)\Pr(\text{bin}D_k)$. In our experiments, the class priors, $\Pr(\mathbf{c}_i)$ for CA and CB (these distributions are taken to be equal), were assumed to be uniform; this assumption may not hold depending on the clustering method used, in which case the distribution must be estimated from the clustering result. A maximum likelihood estimate of $\Pr(\mathbf{c}_i)$ can be obtained from the available measurements by counting the weighted membership of each cluster. $\Pr(\text{bin}D_i)$ can be similarly estimated from the map data.

5.3.6 Implementation Details

Practical use of the SVA mapping approach requires a data structure to support efficient lookup of sets of possible matches. Our implementation begins by discretizing the range of inter-patch distances into a set of bins and combining them with bins for appearance cluster membership for each of the surface patches to form a 3-dimensional discrete space into which patch pairs can be classified. Pairs can be classified into a number of different bins with weights given by the probabilities calculated as described in the previous sections.

[2] Note that Equation 5.25 is still effectively one large summation, as in Equation 5.24. It has just been broken down to show the structure of the inner parts.

This mapping was implemented as a hash multi-map, indexed by bin numberings of a virtual lookup table (most of whose entries may be empty), to support fast lookups without using excessive space when the space is sparsely covered (particularly, e.g., when there are many appearance clusters). This data structure can then be searched for potential matches for each pair by assembling all the pairs in the virtual bins corresponding to each 3-tuple with significant probability, consisting of (1) an appearance bin associated with the first patch, (2) an appearance bin associated with the second patch, and (3) a distance bin for the patch separation. This defines the set of values of i, j, and k that must be evaluated in Equation 5.25, since the observed measurement probabilities are considered zero for all other combinations; then the entries in the map give the product $\Pr(\mathbf{m}_{a2} \mid CA = \mathbf{c}_i)\Pr(\mathbf{m}_{b2} \mid CB = \mathbf{c}_j)\Pr(\delta_{a2b2} \mid \text{bin}D_k)$.

5.4 Recognition and Localization from SVA Maps

Given a matched pair of surface patches, $\text{pair}(z_{a1}, z_{b1})$ and $\text{pair}(\mathbf{m}_{a2}, \mathbf{m}_{b2})$ we now wish to estimate the set of rigid transformations that would align them.

5.4.1 Initial Alignment of Surface Patch Pairs

We begin with a version of the method of Arun et al. (1987) to align 3D point clouds, simplified to the case of two points. Continuing with the notation of Section 5.3, this procedure gives a rotation, R_1 that is effective for aligning the points of contact, p_{a1} with p_{a2} and p_{b1} with p_{b2} but it leaves rotations about the axis between the points in the pair, $\mathrm{axis}_{a,b} = p_b - p_a / ||p_b - p_a||^2$, unconstrained. The surface normals associated with each patch can next be used to further constrain the aligning transformations. The normals are limited in their ability to be used in this respect, though, in two ways:

- Each normal only constrains rotation about $\mathrm{axis}_{a,b}$ if it is not collinear with $\mathrm{axis}_{a,b}$.
- The observed sensor surface normals themselves may be unconstrained if the object surface normal in the area is not well-defined, e.g. in the case of an edge or corner.

Because of these factors, we have considerably more confidence in the point locations than in the surface normals, so we are comfortable parameterizing the aligning transformation as a rigid transformation based on point locations followed by a rotation about $\mathrm{axis}_{a1,b1}$ by an angle AA with uncertainty. We will first discuss our handling of the second issue above in Section 5.4.2, then this will be incorporated with the first issue into our full estimate of the axis-angle rotation portion of the transformation with its associated uncertainty in Section 5.4.3.

5.4.2 Estimating Constraints on Sensor Normals

The surface normal at a surface patch can only be used to constrain rotation if it is itself constrained, so our goal here is to estimate the level of constraint the object surface imposed upon the sensor normal when a reading z_i was taken by looking at the associated tactile image, $\mathbf{I}_i$. Our approach is to infer a rough set of contact points of the sensor with the object surface from sensor elements with non-zero responses. In order for the sensor normal to be well-constrained, the surface should make contact with the sensor in at least three well-separated, non-collinear locations. Algorithm 5.4.2 quantifies a way of measuring the degree of fulfillment of this requirement, returning $normConf(n_i)$.

A set of 3D contact points are estimated from $\mathbf{I}_i$. Their centroid is subtracted off, giving a set of relative positions, which are assembled into a matrix, A, whose singular value decomposition is computed. In the case of a completely planar contact, we would expect two large singular values and one zero value. For surface contacts with four or more observable non-coplanar points of contact (which could still be well- constrained), one could get three significant singular values. For edge or point contacts, however, one would expect less than two significantly non-zero singular values, and in this case the function returns a confidence of zero. Otherwise it returns a value greater than zero that approaches one as the two largest singular values approach each other, i.e. as the contact type approaches fully planar.

Algorithm 2 Estimate Normal Constraint

```
$pts \leftarrow \emptyset$
$avgPt \leftarrow point3D(0,0,0)$
**for all** sensor elements $i$ **do**
    **if** $val(i) > contactThresh$ **then**
        $p \leftarrow point3D(getX(i), getY(i), estimateDepth(val(i))$
        add $p$ to $pts$
        $avgPt \leftarrow avgPt + p$
    **end if**
**end for**
**if** $sizeOf(pts) < 3$ **then**
    **return** 0
**end if**
$avgPt \leftarrow avgPt/sizeOf(pts)$
$r \leftarrow 1$
$A \leftarrow matrix(sizeOf(pts), 3)$
**for all** points $p$ in $pts$ **do**
    $rowOf(A, r) \leftarrow p - avgPt$
    $r \leftarrow r + 1$
**end for**
$S = svd(A)$ {Returns sorted vector of singular values}
**return** $S[2]/S[1]$ {Ratio of two largest singular values}
```

5.4.3 Formulating Axis-Angle Uncertainty

Finally, we incorporate the constraint imposed by the normals on about-axis rotation with the uncertainty in the normals themselves to estimate a distribution over possible axis-angle rotations to complete the alignment of our two patch pairs.

5.4.3.1 Estimating the Axis-Angle Rotation

First the surface normals associated with the first patch pair, n_{a1} and n_{b1}, and the axis between them, $\text{axis}_{a1,b1}$ are rotated according to the transformation obtained in Section 5.4.1 to be in the same coordinate system as n_{a2} and n_{b2}. Then a projection $Proj$ is computed to project each pair's normals onto the plane normal to $rax = R_1 \text{axis}_{a1,b1}$.

$$pn_{a1} = Proj(R_1 n_{a1}) \tag{5.26}$$

$$pn_{b1} = Proj(R_1 n_{b1}) \tag{5.27}$$

$$pn_{a2} = Proj(n_{a2}) \tag{5.28}$$

$$pn_{b2} = Proj(n_{b2}) \tag{5.29}$$

Next, a rotation is computed to align these projected normals once again based on the method of Arun et al. (1987):

$$H = pn_{a1} pn_{a2}^T + pn_{b1} pn_{b2}^T \tag{5.30}$$

$$USV^T = \text{svd}(H) \tag{5.31}$$

$$R_{AA} = VU^T \tag{5.32}$$

The projection has the effect of scaling each vector by the degree to which it is perpendicular to $\text{axis}_{a1,b1}$ in the rotated space, thereby also scaling its contribution to the least squares error being minimized in the fit. If the determinant of R_{AA} is negative, this generally means S is rank deficient and the sign of one column of U is unconstrained, so R_{AA} is reset to $V\,diag(1,-1)\,U^T$, giving a valid rotation.

5.4.3.2 Converting to a Distribution

Finally, the angle of rotation is extracted from R_{AA} to get $\hat{AA}$, our estimate of AA. The overall confidence in this value, ranged zero to one, is estimated as

$$q_a = ||pn_{a1}||\ normConf(n_{a1})\ ||pn_{a2}||\ normConf(n_{a2}) \tag{5.33}$$

$$q_b = ||pn_{b1}||\ normConf(n_{b1})\ ||pn_{b2}||\ normConf(n_{b2}) \tag{5.34}$$

$$alignConf = \frac{q_a + q_b}{2} \tag{5.35}$$

The distribution of possible true values of AA is then conservatively estimated as a Gaussian with mean $\hat{AA}$ and standard deviation given by $\sigma_{\text{init}}/alignConf$. In our evaluations, σ_{init} was set to 0.1.

5.4.4 Maintaining State Estimates

At each time-step, a new sensor reading is received, which is paired with previous sensor readings to form a set of surface patch pairs as in Equation 5.13. Then Equation 5.14 shows how these pairs are used to update our estimates of the object state, x_t. The SVA map is queried for a list of surface patch pairs that may match each of the observed pairs, the object with which they are associated, and their match likelihoods. These are computed as described in Section 5.3. Then for each of these possible matches, a distribution is computed over possible aligning transformations as described in the previous section and associated with the object identity from the map to give a distribution over x_t.

In practice, the distribution over x_t is maintained as a sparse histogram. Except at initialization, when the distribution can be implicitly assumed equal to the prior (e.g., uniform), the likelihood in most bins will be zero. The data structure can be efficiently implemented using a hash map indexed by the histogram bin numbers.

5.5 Evaluations

Recognition was performed on the same sets of objects as in Chapters 3 and 4, so the performance of this method could be easily compared to that of the previous methods. Simulation evaluations using the models from the Princeton set of Chapter 4 are covered in Section 5.5.1, and experiments on the raised letters of Chapters 3 and 4 are in Section 5.5.2.

5.5.1 3D Simulation Evaluations

Recognition evaluations using the SVA approach in simulation were conducted using the same 3D models as in Section 4.4.2. This was a set of 10 diverse objects selected from the Princeton shape benchmark (Shilane et al., 2004) (See Figure 4.6).

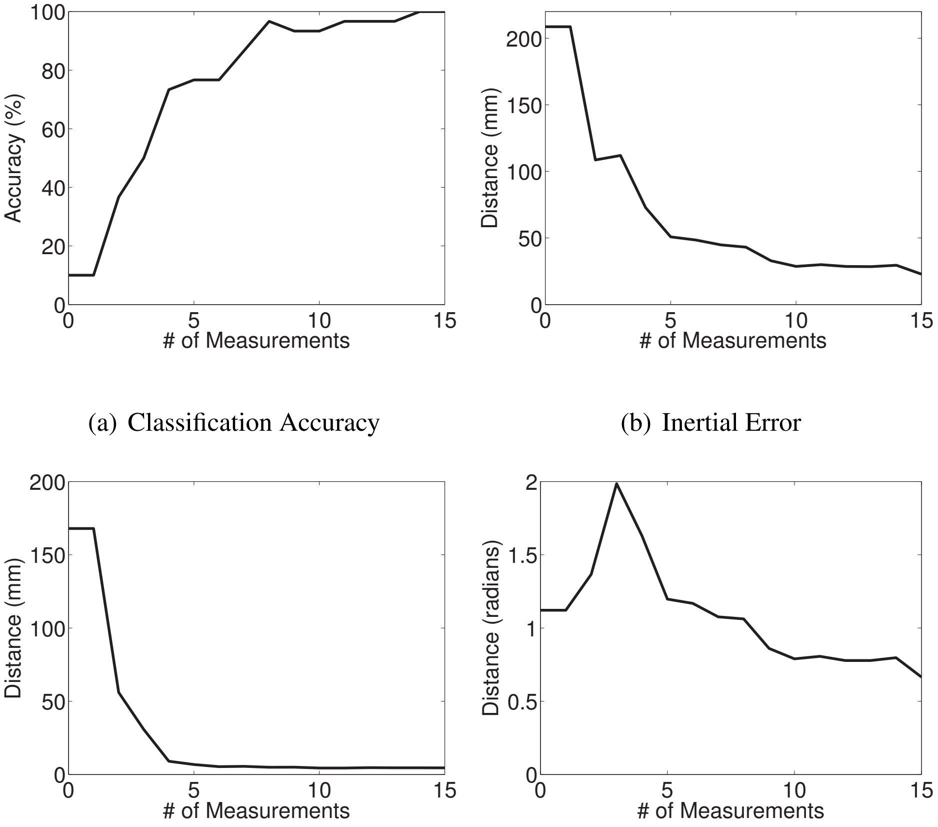
A set of 1000 sensor readings of each object was collected for training, and a separate 100 readings of each object were collected for testing. All sensor readings were collected using the sampler of Section 4.4.1.2.

5.5.1.1 Map-Building

As described in Section 5.3, appearance descriptors were extracted from each sensor reading and these were grouped into 25 clusters using *k*-medoids, then an SVA map was built of all the objects. The map used 40 bins to discretize the space of inter-patch distances that covered a range from 30 mm to 160 mm. The objects themselves were scaled 80 mm in their largest dimension.

5.5.1.2 Testing

Recognition performance was measured as a function of the number of sensor readings seen so far by averaging results over a number of trials. In each trial, readings were selected uniformly at random from the set of test readings for the selcted unknown object. An object pose was generated and the pose of each sensor reading was transformed according to this unknown pose before it was presented to the recognition algorithm. This pose consisted of an arbitrary 3D rotation and a translation in the range $[-200, 200]$ mm in each direction. One test repetition consisted of one trial of recognition on each object from the set. Performance was averaged over three test repetitions to get the final results shown in Figure 5.3.



(a) Classification Accuracy

(b) Inertial Error

(c) Translational Error

(d) Angular Error

Figure 5.3: SVA recognition and localization results on Princeton set.

The virtual histogram used to maintain pose estimates used 50 bins for each dimension of translation and represented rotation by a 3-dimensional vector in Rodrigues form (with magnitude encoding angle of rotation) divided into 9 bins for each dimension. At each time step, the hypothesized object identity was taken as the object with the most probability weight, summed over all possible poses. The hypothesized object pose was taken as the centroid of the histogram bin with the highest weight.

5.5.1.2.1 Performance Measures

As in previous evaluations, performance was measured in terms of classification accuracy and of distance from the estimated pose, $[\hat{R}\hat{T}]$, to the true pose, $[RT]$, where a pose of $[\mathbf{I0}]$ corresponds to the object located at the origin in its canonical pose (that observed in training). Classification accuracy was taken as the percent of the time the hypothesized object identity was the true identity over all trials. Error in the pose was recorded only when the mode corresponded to the true object identity, and it was measured in three ways:

Translational error was measured as the distance between the translational components, $||\hat{T} - T||$.

Angular error was taken as the angle of rotation, ϕ_e, required to align the estimate with the true pose:

$$\phi_e = ||unskew(logm(R\hat{R}^T))|| \tag{5.36}$$

where *logm* denotes the matrix logarithm and *unskew* extracts the vector v from $sk(v)$, its corresponding skew-symmetric matrix.

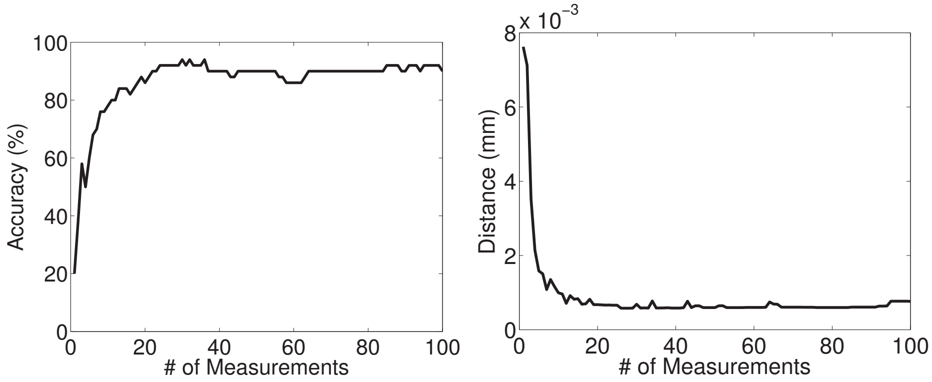
Inertial distance was measured according to the metric of (Chirikjian and Zhou, 1998, Equation 4), where each object's mass and moment of inertia was approximated by a solid sphere of radius 40 mm. This metric combines translational and angular error into a measurement of the energy required to align the two transformations.

5.5.1.2.2 RESULTS

The MNTI descriptor (See Section 4.3.2.5) was used to characterize appearance due to its invariance properties' robustness to small translations. Figure 5.3 graphs all of the error metrics above: Figure 5.3(a) shows recognition accuracy. Figures 5.3(b), 5.3(c), and 5.3(d) show inertial, translational, and angular error respectively. Classification accuracy climbs to 100% with ten sensor readings. Translational error drops to slightly above 4 mm, the lowest expected attainable error using histogram bins of width 8 mm. Angular error remains high throughout, however, most likely due to symmetries in the objects, which are not being specially handled. The glass, tire, and screwdriver models, for instance, all have an axis of rotational symmetry that prevents any algorithm with the available information from constraining one degree of freedom of the pose. The increase in angular error at small numbers of readings is a result of objects with poor orientation estimates being correctly identified for the first time; recall that error in pose is computed only for objects that are correctly identified, so the sharp increase in recognition accuracy during this period results in more models contributing to the pose accuracy. As a result of the angular error, inertial error decreases substantially, but it does so more slowly than translational, and it does not reach its minimum expected value.

5.5.2 2D Physical Sensor Experiments

The SVA mapping approach was also tested on the raised letter set used in Chapters 3 and 4. The experimental setup was the same as in Section 3.4. The same set of sensor read-



(a) Classification Accuracy

(b) Inertial Error

Figure 5.4: SVA recognition and localization results on raised letters

ings, collected with the mechanical apparatus of Figure 3.6, were used. This set consisted of two readings taken at each of 192 poses. The first set of readings at each pose was used for training, while the second set was used for testing.

As in previous experiments, readings from the unknown object were transformed according to a randomly selected object pose for each trial. This pose consisted of a translation in x and y in the range $[-10, 10]$ mm in each direction and an arbitrary rotation in the plane. This pose space was discretized using 21 bins for each dimension of translation and 9 bins for each dimension of the Rodrigues vector representing rotation. The map used 100 bins for distance, covering a range of 3 mm to 20 mm. Since there was a discrete set of contact locations, invariance to translation was less of a concern, so the Moment-Normalized descriptor was used.

Classification accuracy and inertial distance are shown in Figure 5.4. Classification accuracy quickly climbs above 90% within about 30 sensor readings. Inertial distance, which considers symmetries in the letters as described in Section 3.4, drops down below 1 mm within about 20 sensor readings. This is once again close to the expected optimum given the virtual histogram resolution.

5.6 Discussion

In this chapter, we have presented a method that makes use of both of the appearance content of a set of tactile force sensor readings and the geometric information associated with each. The method was demonstrated on both the simulated and real data sets of Chapters 3 and 4, exhibiting strong performance both in recognition accuracy and pose estimation in all cases. Performance on real sensor readings was not perfect, however, so we provide some analysis of why that may be, ideas for improvement, and guidance on how to apply the method in different situations.

5.6.1 Analysis of Failure Modes

One can see from the experiments of Sections 5.5.2 and 3.4 that the SVA approach achieves higher pose accuracy than the geometric method but does not quite reach the same level of classification accuracy. Higher pose accuracy is achievable because a higher-resolution histogram can be maintained using a forward mapping from sensor readings to pose and a sparse representation of the probability space. One question that might naturally arise is why the SVA method does not achieve 100% classification accuracy when the geometric method does. This question gets to the heart of how the approach works in practice; it is instructive to address it in some detail, as it gives guidance as to what considerations should be made when applying the method in other situations.

A classification failure must result from the true pose (or a pose that maps to the same virtual histogram bin with significant probability) not being among those that evaluated as able to explain a surface patch pair. If at any point the bin of the true pose is not assigned significant probability (i.e., it is taken to be zero), then the optimal solution will not be found unless all probabilities go to zero and the distribution is re-seeded. It is still possible for classification to succeed, however, if another pose hypothesis with the same object identity maintains significant weight. This situation is most likely when the object has symmetries, as additional measurements are otherwise likely to eliminate this hypothesis. So then why would the true pose be assigned zero probability? There are a few ways this might occur:

1. A sensor reading may be of a portion of the object surface that was not observed during training. In this case, there would be no valid match in the map for any pair that included that point, so only erroneous matches would remain.

2. A patch pair may not be matched to the nearest corresponding regions represented in the map because their estimated parameters (appearance and distance) do not match well enough.

3. The distribution of aligning transformations computed for a correct (or close) patch pair match may not place significant probability on the true object pose. This might be due to a mis-estimation of the surface patch's locations or (more likely) their surface normals.

4. A combination of the factors above, each acting in part, could push the probability of the true pose below machine precision.

The first situation would not occur with the raised letter data, since sensor readings were taken at set locations, and those locations were the same in training as in testing. The third failure mode is also not likely on the letter data for the same reason, and since the surface normals were all known and equal in this case. The fourth failure mode did not seem to come into play in our experiments with our chosen parameters; an implementation that represented probabilities as log probabilities (thereby significantly extending the range of representable probabilities) did not appear to affect performance. The most likely source of classification failures therefore seems to be the second item, in the appearance classification of surface patch pairs.

As described in Section 2.6.2, having a deformable covering that is not permanently affixed to the sensors makes the sensor response less consistent. It appears that the thresholding steps in the geometric approach are more robust these effects. Although these steps discard some information, thresholded sensor responses are evidently still quite informative. Either thresholding, e.g. using an approach similar to that of maximally-stable extremal regions (Matas et al., 2004), or some other method of increasing the robustness of appearance classifications could be helpful when dealing with real sensor readings.

5.6.2 Parameter Selection

The SVA approach relies on the choice of some parameters. Our choices for each were given in the text, but here we elaborate a bit on the rationale one might take for making such choices in other situations.

Appearance Similarity Threshold The selectivity of appearance matching can be somewhat regulated by the choice of the threshold T_a in Equation 5.18. Lowering this value decreases the chances of a true match being erroneously pruned away, but it also increases the number of expected false positive matches. In the extreme, T_a could be set to 0; in that case, matches would still be weighted according to their likelihood, but the number of matches would depend only on surface patch distances, resulting in a very large increase in that number and therefore in the computation required in subsequent steps and the size of the map data structure.

Minimum and Maximum Distances In general, the minimum and maximum distances used in the patch pair map, minDist and maxDist from Equation 5.20, should be set according to the size of the objects being mapped (for maxDist) and the required baseline between points to be able to stably estimate a transformation and the range of distances of which appearance characterizations are consistent (for minDist). minDist can be increased, however, not only to further guarantee stable transformation estimates, but also to simply reduce the number of matches that must be stored in the map or processed curing recognition. This obviously involves discarding some information in the form of closer-together patches, but these sacrifices may be worthwhile when computational resources are limited. Such tuning would, for instance, have a particularly large influence on mitigating the increase in computation mentioned above resulting from setting T_a to zero.

Number of Clusters In the appearance-only approach of Chapter 4, increasing the number of clusters (mixture components in the GMM) never hurt performance in our experiments. Although the appearance class modeling in the SVA approach is based on that one and it still makes soft associations, the pruning of low-association matches (essential to limiting the number of hypotheses that need to be evaluated) leads to more clusters not always being better in this case. Too many clusters (in conjunction with a high value for T_a) increases the risk of readings not being associated with their true match, whereas too few clusters can lead to poor discrimination and an unnecessarily large number of matches; care must therefore be taken with the choice, and determining the appropriate value is somewhat of a dark art, as with most clustering algorithms.

5.6.3 Sampling Density Requirements

In this work, we have not focused on the effects of the exploration process either in training or in testing. In training, we have allowed allowed the robot to collect a very large number of sensor readings to build its recognition model, under the assumption that it would be more than sufficient. In the SVA approach, since the size of this data structure grows with the square of the number of sensor readings used, it would be desirable in practical applications to use only as much data is necessary.

Determining the sampling density required to map a particular set of objects would be an interesting direction for future research. The sampling density required is effectively determined by the number of distinct regions an object's surface must be divided into to support the desired recognition precision. This would depend on the distance classes (which in turn depends on the requirements for localization accuracy) and appearance classes used. Appearance features must be consistent over the entire region, so, according to the analysis of Section 5.3.3, the density needed would vary widely based on the descriptor used. Different sets of objects could also have very different requirements, depending on the scale of surface features as observed by the sensors and the smoothness with which appearance varies across the object surfaces.

5.6.4 Scalability

One drawback to the modeling of the forward association between measurement and state inherent in the SVA approach is the amount of memory required. Since the SVA map of each object grows with the square of the number of sensor readings used to build it, each object's map can be quite large. Since each object's map is independent, though, the size of the entire database grows linearly with the number of objects, assuming they are sampled consistently. In the current implementation, the total memory usage of the application reaches approximately 4.6 GB when doing recognition in the simulations of Section 5.5.1, so recognition on larger databases would require high-memory machines. Note that this is without applying any techniques to compress the memory usage, and the modifications of Section 5.6.3 have the potential to significantly reduce the database size. Nonetheless, recognition on very large sets of objects would probably require modification for out-of-core processing of the map database.

5.6.5 Comparison of Performance to Geometric Approach

It is worthy of note that the performance of the SVA approach in experiments on the letter set was not as strong as in simulation. In fact, the classification accuracy on the letter set was lower than that of the geometric method (though localization error was lower). We suspect that this decrease in performance is the result of two factors:

- The requirements on appearance features are pulled in two conflicting directions; appearance classes must be general enough to describe all surface patches that may be encountered, so it is difficult to simultaneously support discrimination of fine-level detail.
- The current formulation of appearance features is less robust to inconsistencies in the sensor response than the discretized form used by the geometric method through its thresholding contact classifier. Particularly in the context of the relatively hard association of appearance features to appearance classes[3], deriving appearance features from binarized images like those used in the geometric approach may increase robustness on real sensors.

As a result, a hybrid approach may be most suitable. The computational limitations of the geometric approach apply particularly at the very beginning of the exploration of the object, when the set of plausible state hypotheses spans the entire state space. Later in the recognition process, the number of plausible hypotheses has typically been greatly reduced, potentially allowing the geometric approach to be feasible again. At the same time, the computation required for the SVA approach increases throughout the recognition process as more pairs become available (assuming all available pairs are used, which is not mandatory), so a crossover point would occur in most cases[4] where switching to the geometric method would become less computationally expensive. Since this method may also

[3] That is, features are associated with only a small number of clusters, as opposed to the soft clustering in the appearance-only approach, where features had some association to every appearance class.

[4] For certain types of objects, the uncertainty in pose may not be able to be resolved. For example, the orientation of a homogeneously-textured sphere could not be determined from any number of sensor readings.

be able to better discriminate fine-level detail, there is potential to improve accuracy as well as efficiency.

5.6.6 Parallelization

Like the geometric method, for which each particle or histogram bin calculation is independent, this approach has great potential for parallelization. With this method, the computations for each surface patch pair can be carried out in parallel. Additionally, the computations for each prospective match for a surface patch pair are also independent, leading to another level of parallelization.

5.6.7 Other Domains

Although this method was developed with tactile force sensors as the intended source of information, it should be noted that it is equally applicable to other sensing modalities. For instance, a stereo vision system could also be used to acquire surface patch information comprising location, surface normal estimates, and appearance. A particularly interesting extension would be to examine what appearance properties can be characterized both by vision and touch; then cross-modality models could be built using one modality and then used in another or with both modalities.

Chapter 6

Conclusions

Tactile force sensors offer great opportunities to sense the physical properties of objects being manipulated, and they are becoming increasingly available on the end-effectors of modern robots. Object recognition is a good benchmark task to drive the development of techniques for estimating and representing an object's physical characteristics, and so it was the problem addressed in this work. We began with a model of tactile force sensors as imagers in Chapter 2 and a method to implement that model as a tactile force sensor simulator. The chapters that followed presented different approaches to the object recognition problem, using either an entirely geometric model of the object (Chapter 3), a model based only on local texture or "tactile appearance" (Chapter 4), or a combination of the two in the form of "spatially-varying appearance" characteristics (Chapter 5). These algorithms were validated both using the simulator of Chapter 2 and using data from physical sensors. The geometric approach provided a method to generate occupancy grid maps of objects from tactile force data. A method was also presented to use such maps for recognition and localization of unknown objects from tactile data; these maps might also be used for other purposes, though, and the maps used for recognition could be generated with other sensing modalities. The appearance-based approach explored the idea of local appearance as applied to tactile sensor readings. Finally, the combination of this notion of tactile appearance with the sequential state estimation techniques of the geometric approach and some ideas from geometric hashing formed the basis for the first combined geometry and appearance tactile recognition method.

In all, we have presented three different approaches to the haptic object recognition problem, each appropriate for use in a particular set of circumstances. The SVA recognition algorithm of Chapter 5 builds upon ideas of the other two, so it is the most general-purpose of the three. The appearance-only algorithm of Chapter 4 is more suitable, though, in situations where the position information associated with tactile sensor readings is unreliable or the object moves in unknown ways as it is being explored. The geometric approach of Chapter 3 may be most appropriate for cross-modality work, since it allows almost direct use of a geometric model of the object (e.g., from vision) for tactile recognition. It also provides a method for producing such geometric models, which could in turn be used in other sensing modalities.

6.1 Future Work

In addition to the opportunities for extending each of the individual methods, this work suggests several promising future research directions in the domain.

6.1.1 Implementation on Physical Robot

Because a robot was not available with integrated tactile sensing, our ability to conduct experiments using real sensors was limited to those involving data that could be acquired without such a robot, e.g., using a mechanical system instead to measure the sensor position. Analysis of the full 3D robotic exploration and manipulation task was therefore limited to simulation. Applying the techniques in this dissertation to a robot with integrated tactile force sensors would enable a richer set of experiments to help point out and then resolve the inevitable difficulties that arise in applying algorithms to the real world.

6.1.2 Skin-like Sensors

The work in this dissertation has been focused on sensors with sensing elements laid out in a planar array. A number of groups have also developed tactile sensors with less structured, often deformable geometries, typically to enable skin-like contact detection over the entire surface of a robot, rather than explicitly to inform manipulation (e.g. Tajima et al., 2002; Kerpa et al., 2003; Hoshi and Shinoda, 2006; Ohmura et al., 2006; Dahiya et al., 2009). Despite differences between the geometries of such sensors and those of the systems used in this work, some of the techniques presented here could be applied to these sensor arrays to provide another source of information about a robot's environment. The geometric approach of Chapter 3 could be applied to skin-like tactile sensors almost without modification, provided that the position of each sensor element can be accurately estimated. Even if the element positions are not known precisely, the method already provides a framework for explicitly handling this uncertainty. The methods that make use of appearance information would require a generalization of the notion of appearance to this type of sensor and the types of sensor-surface contact one would expect to encounter. This would probably involve a more explicit modeling of physical surface characteristics, as opposed to the tactile images they give rise to.

6.1.3 Informed Exploration

In this work, we have avoided placing any more constraints upon the exploration process than necessary, but an active and informed exploration process could also greatly improve recognition. Our simulation framework provides a method to estimate expected information gain from particular future actions. This could be used to optimize the exploration process, e.g. to decrease the number of sensor readings required or to minimize energy expenditures.

6.1.4 Interest points

The notion of interest points from the computer vision domain, such as the well-known SIFT features (Lowe, 1999), could be applied to the tactile domain. Well-localized distinctive points on the surface of an object could greatly reduce the search space for object recognition or localization. The SVA approach already makes use of point-matching similar to that which might be applied to detected interest points. The increased distinctiveness of interest points would greatly reduce the number of possible matches that need to be considered though, constraining the object with many fewer measurements.

The major trade-off is that distinctive points are observed much less frequently. Preliminary work suggests that straightforward controllers could be used to guide a sensor across the surface of an object to converge upon distinctive points, but this obviously requires changing the exploration algorithm. Interest points could certainly be used whenever they happen to be encountered, even if the exploration process does not actively seek them out, but certain classes of objects may not even have any interest points (consider, e.g., a sphere). These considerations dissuaded us from pursuing methods that rely on interest points in the work presented here, but they nonetheless offer opportunities to supplement the methods presented here with significant additional information.

6.1.5 Tactile Image Sequences

The ability to simulate sensor responses in real time enables a number of interesting potential extensions, including the potential to simulate active sensing of properties that may require motion to sense, such as roughness or stiffness. Though Chapter 4 presents a method for characterizing surface texture from static touch, humans extract much more information from deriviative information associated with sliding contact. Temporal derivative information could be used to enable more human-like sensation. Image sequences could be mosaicked together to produce larger local surface patches for analysis. Finally, dynamic viscoelastic properties could be estimated through active touch. This type of exploration leads to some slightly different engineering constraints on the system (such as the ability to deal with shear forces), but it also offers a rich source of new information. In our work thus far, we have ignored effects such as friction and other forces in directions orthogonal to the sensor surface. Additional modeling would be necessary to incorporate these effects.

6.1.6 Incorporation with Other Sensors

In this dissertation, we have focused entirely on the use of tactile force sensors to inform the recognition process. Many robotic systems would have other ways of gathering information though, such as from vision or range sensing, which could also be put to use. Just as the experiments of Chapters 3 and 4 informed the design of the algorithm in Chapter 5 that had more information available, so could multi-sensor algorithms make use of this work. The probabilistic nature of the methods presented here also make them particularly well-suited to being combined with other sources of information; e.g., by explicitly modeling uncertainty, one can use the most reliable information from each sensor. In many cases, the likelihoods of object identiy and pose given just visual information could produce a small number of high-confidence hypotheses, where the remaining ambiguities could be resolved through touch.

Bibliography

1001 Free Fonts. Corpulent caps font, 2010. URL `http://www.1001freefonts.com/CorpulentCaps.php`. [3]

P. K. Allen and P. Michelman. Acquisition and interpretation of 3-d sensor data from touch. *IEEE Transactions on Robotics and Automation*, 6(4):397–404, 1990. ISSN 1042-296X. [1.3, 1.3.3]

P. K. Allen and K. S. Roberts. Haptic object recognition using a multi-fingered dextrous hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 342–347, Scottsdale, AZ, 1989. [1.3]

P. K. Allen, A. T. Miller, P. Y. Oh, and B. S. Leibowitz. Integration of vision, force and tactile sensing for grasping. *International Journal of Intelligent Machines (IJIM)*, 4(1): 129–149, 1999. [1.3]

D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-898716-24-5. [4.4.2.1]

K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9:698–700, September 1987. [5.4.1, 5.4.3.1]

R. Bajcsy. What can we learn from one finger experiments? In *International Symposium on Robotics Research (ISRR)*, pages 509–527, Bretton Woods, NH, 1984. [1.3]

R. Bajcsy and G. Hager. Tactile information processing – the bottom up approach. *7ICPR*, pages 809–811, 1984. [1.3]

J. Bay. Tactile shape sensing via single- and multifingered hands. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 290–295, Scottsdale, AZ, 1989. [1.3]

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 509–522, 2002. [4.3.2.5]

P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, 1992. [1.3, 4.1.1.2]

A. Bosch, A. Zisserman, and X. Munoz. Scene classification via pLSA. In *European Conference on Computer Vision (ECCV)*, pages 517–530, 2006. [4.3.2.2]

M. Briot et al. The utilization of an artificial skin sensor for the identification of solid objects. In *International Symposium on Industrial Robots*, pages 13–15, 1979. [1.3.3]

M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1704–1711, Los Alamitos, CA, USA, 2010. IEEE Computer Society. [4.1.1.1]

G. Canepa, M. Morabito, D. De Rossi, A. Caiti, and T. Parisini. Shape estimation with tactile sensors: a radial basis functions approach. In *IEEE Conference on Decision and Control*, pages 3493–3495 vol.4, 1992a. [2.1]

G. Canepa, M. Morabito, D. de Rossi, A. Caiti, and T. Parisini. Shape from touch by a neural net. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2075–2080 vol.3, May 1992b. [2.1]

Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial-bag-of-features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3352–3359, June 2010. [4.1.1.1, 5.1]

S. Casselli, C. Magnanini, and F. Zanichelli. On the robustness of haptic object recognition based on polyhedral shape representations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, page 2200, Los Alamitos, CA, USA, 1995. IEEE Computer Society. [1.3]

S. R. Chhatpar and M. S. Branicky. Localization for robotic assemblies using probing and particle filtering. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1379–1384, July 2005. [3.1.1.2]

G. S. Chirikjian and S. Zhou. Metrics on motion and deformation of solid models. *Journal of Mechanical Design*, 120:252, 1998. [3.4.2, 5.5.1.2]

H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005. [4.2.1]

C. S. Chua and R. Jarvis. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision (IJCV)*, 25(1):63–85, 1997. [1.3.1]

G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision (ECCV)*, volume 1, page 22, Prague, Czech Republic, 2004. [4.1.1.1]

R. Dahiya, G. Metta, M. Valle, A. Adami, and L. Lorenzelli. Piezoelectric oxide semiconductor field effect transistor touch sensing devices. *Applied Physics Letters*, 95(3): 034105–034105, 2009. [1.1, 6.1.2]

J. Dargahi and S. Najarian. Advances in tactile sensors design/manufacturing and its impact on robotics applications–a review. *Industrial Robot: An International Journal*, 32(3): 268–281, 2005. [1.1]

F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1322–1328. Citeseer, 1999. [3.1.1.2, 3.3.1, 3.3.3]

A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249 –265, jun. 1987. [3.1.1.1, 3.2.1, 2]

R. Ellis and M. Qin. Singular-value and finite-element analysis of tactile shape recognition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2529–2535 vol.3, May 1994. [2.1]

E. Faldella, B. Fringuelli, D. Passeri, and L. Rosi. A neural approach to robotic haptic recognition of 3-d objects based on a kohonen self-organizing feature map. *IEEE Transactions on Industrial Electronics*, 44(2), April 1997. [1.3]

R. Fearing. Tactile Sensing Mechanisms. *International Journal of Robotics Research (IJRR)*, 9(3):3–23, 1990a. [1.3]

R. Fearing. Tactile sensing for shape interpretation. In *Dextrous robot hands*, pages 209–238. Springer-Verlag New York, Inc., 1990b. [1.3]

R. Fearing and T. Binford. Using a cylindrical tactile sensor for determining curvature. *IEEE Transactions on Robotics and Automation*, 7(6):806–817, Dec 1991. ISSN 1042-296X. [2.1]

J. Fehr, A. Streicher, and H. Burkhardt. A bag of features approach for 3d shape retrieval. *Advances in Visual Computing*, pages 34–43, 2009. []

R. Fergus, P. Perona, A. Zisserman, and O. P. U. K. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 264–271, 2003. [4.1.1.1, 5.1]

R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning Object Categories from Google's Image Search. In *International Conference on Computer Vision (ICCV)*, volume 2, 2005. [4.3.2.2]

D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *National Conference on Artificial Intelligence*, pages 343–349. John Wiley & Sons Ltd., 1999. [3.1.1.2, 3.3.1]

W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(9):891–906, 1991. [4.3.2]

A. Frome, D. Huber, R. Kolluri, and T. Bülow. Recognizing objects in range data using regional point descriptors. In *European Conference on Computer Vision (ECCV)*, pages 224–237, 2004. [4.3.2.5]

K. Gadeyne and H. Bruyninckx. Markov techniques for object localization with force-controlled robots. In *International Conference on Advanced Robotics (ICAR)*, pages 91–96, 2001. [3.1.1.2]

R. Gal and D. Cohen-or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics*, 25:130–150, 2006. [1.3.1]

J. M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12(8):938–943, 2003. [4.3.2.3]

N. Gorges, P. Fritz, and H. W

”orn. Haptic object exploration using attention cubes. In *KI 2010: Advances in Artificial Intelligence*, pages 349–357. Springer, 2010. [1.3, 4.1.1.2]

W. Grimson and T. Lozano-Perez. Model-based recognition and localization from tactile data. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 248–255, Atlanta, GA, 1984. [1.3, 5.1, 5.1.1]

J. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *1998 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, 1998. [3.4.1]

W. D. Hillis. A High-Resolution Imaging Touch Sensor. *The International Journal of Robotics Research (IJRR)*, 1(2):33–44, 1982. [1.3.3]

T. Hoshi and H. Shinoda. Robot skin based on touch-area-sensitive tactile element. In

IEEE International Conference on Robotics and Automation (ICRA), pages 3463–3468. IEEE, 2006. ISBN 0780395050. [6.1.2]

M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962. ISSN 0096-1000. [4.3.2.4]

A. Iggo and A. R. Muir. The structure and function of a slowly adapting touch corpuscle in hairy skin. *Journal of Physiology*, 200(3):763–796, February 1969. [2.5.2]

A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997. [4.3.2.5]

A. E. Johnson and M. Hebert. Recognizing objects by matching oriented points. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 684–689, 1997. [1.3.1]

M. K. Johnson and E. H. Adelson. Retrographic sensing for the measurement of surface texture and shape. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1070–1077, Miami, FL, 2009. [1.1]

F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *IEEE International Conference on Computer Vision*, volume 1, pages 604–610, Beijing, China, 2005. [4.1.1.1, 4.3.1]

R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960. [3.3.1]

L. Kaufman and P. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 5. Wiley Online Library, 1990. [5.3.2]

O. Kerpa, K. Weiss, and H. Worn. Development of a flexible tactile sensor system for a humanoid robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 1–6. IEEE, 2003. ISBN 0780378601. [6.1.2]

D. Kraft, A. Bierbaum, M. Kjaergaard, J. Ratkevicius, A. Kjaer-Nielsen, C. Ryberg, H. Petersen, T. Asfour, R. Dillmann, and N. Kruger. Tactile object exploration using cursor navigation sensors. In *World Haptics*, pages 296 –301, March 2009. [4.1.1.2]

K. Kuchenbecker, J. Romano, and W. McMahan. Haptography: Capturing and recreating the rich feel of real surfaces. *Robotics Research*, pages 245–260, 2011. [1.3]

S. Kullback and R. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. [4.3.1.2]

C. Lampert, M. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(12):2129–2142, 2009. [4.3.2.2]

S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, MA, 2006. [4.2.1]

S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research (IJRR)*, 20(5):378–400, 2001. [4.2.1]

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178. IEEE, 2006. [4.1.1.1, 5.1]

S. Lederman and R. Klatzky. Hand movements: A window into haptic object recognition* 1. *Cognitive psychology*, 19(3):342–368, 1987. [1.3]

D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999. [3.5.1, 4.3.2.2, 6.1.4]

K. MacLean. The 'haptic camera': A technique for characterizing and playing back haptic properties of real environments. In *Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, Atlanta, GA, November 1996. [1.3]

Magenta and G. Triantafyllakos. Bpreplay font, 2008. URL `http://www.fontspace.com/backpacker/bpreplay`. [3]

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004. [5.6.1]

T. McGregor, R. Klatzky, C. Hamilton, and S. Lederman. Haptic classification of facial identity in 2d displays: Configural vs. feature-based processing. *IEEE Transactions on Haptics*, 3:48 – 55, 2010. [4.1.1.1]

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence (PAMI), 27(10):1615–1630, 2005. [4.1.1.1, 4.3.2.2]

H. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Mag.*, 9(2):61–74, 1988. [3.1.1.1, 3.2.1]

D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5, pages 2161–2168, New York, NY, 2006. [4.1.1.1]

E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*, pages 490–503, Graz, Austria, 2006. [4.1.1.1]

Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo. Conformable and scalable tactile sensor skin for curved surfaces. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1348–1353. IEEE, 2006. ISBN 0780395050. [6.1.2]

A. Okamura and M. Cutkosky. Haptic exploration of fine surface features. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 2930–2936. IEEE, 1999. [1.3.3]

A. M. Okamura. *Haptic Exploration of Unknown Objects*. PhD thesis, Stanford University, Department of Mechanical Engineering, Stanford, CA, June 2000. [1.3.2]

D. Pai, J. Lang, J. Lloyd, and R. Woodham. ACME, a telerobotic active measurement facility. *Experimental Robotics VI*, pages 391–400, 2000. [1.3]

E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, pages 1065–1076, 1962. [3.3.3]

V. Patoglu and R. Gillespie. The haptic probe: mechanized haptic exploration and automated modeling. In *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 117–124, March 2003. [1.3]

E. Petriu, S. Yeung, S. Das, A. Cretu, and H. Spoelder. Robotic tactile recognition of pseudorandom encoded objects. *IEEE Transactions on Instrumentation and Measurement*, 53(5):1425–1432, 2004. [1.3.3]

A. Petrovskaya, O. Khatib, S. Thrun, and A. Ng. Touch Based Perception for Object Manipulation. In *Robotics Science and Systems (RSS), Robot Manipulation Workshop*, 2007. [3.1.1.2]

Z. Pezzementi, E. Jantho, L. Estrade, and G. D. Hager. Characterization and simulation of tactile sensors. In *Haptics Symposium*, pages 199–205, Waltham, MA, USA, 2010. [1, 2]

Z. Pezzementi, E. Plaku, C. Reyda, and G. D. Hager. Tactile object recognition from appearance information. *IEEE Transactions on Robotics*, 2011a. [1]

Z. Pezzementi, C. Reyda, and G. D. Hager. Object mapping, recognition, and localization from tactile geometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011b. [1]

J. Phillips and K. Johnson. Tactile spatial resolution. III. A continuum mechanics model of skin predicting mechanoreceptor responses to bars, edges, and gratings. *Journal of Neurophysiology*, 46(6):1204, 1981. [2.5.2]

R. Platt, F. Permenter, and J. Pfeiffer. Inferring hand-object configuration directly from tactile data. In *Mobile Manipulation Workshop, IEEE Conference on Robotics and Automation (ICRA)*, 2010. [3.1.1.2]

PPS. DigiTacts II$^{\mathrm{TM}}$, tactile array sensor evaluation kit with digital output. Pressure Profile Systems, 2008. URL `http://www.pressureprofile.com/UserFiles/File/DigiTactsII%20Evaluation%20Specification%20Sheet.pdf`. [2.1.1, 4.4.4]

M. Schaeffer and A. M. Okamura. Methods for intelligent localization and mapping during haptic exploration. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 3438–3445, 2003. [1.3.2]

C. Schmid. Constructing models for content-based image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 39–45, Kauai, HI, 2001. [4.3.2]

A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard. Object identification with tactile sensors using bag-of-features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 243 –248, October 2009. [1.3.3, 4.1.1.1, 4.1.1.2, 4.3.2.1]

P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Shape Modeling International (SMI)*, pages 167–178, Genova, Italy, June 2004. [4.6, 4.4.2, 5.5.1]

M. Shimojo. Spatial filtering characteristic of elastic cover for tactile sensor. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 287–292 vol.1, May 1994. [2.3]

R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 14, pages 1080–1087, 1995. [3.1.1.2, 3.3.1]

J. Son and R. Nowe. Tactile sensing and stiffness control with multifingered hands. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3228–3233 vol.4, Apr 1996. [1.3]

S. A. Stansfield. *Visually-guided haptic object recognition*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 1987. [1.3.3]

P. Swerling. A proposed stagewise differential correction procedure for satellite tracking and prediction. Technical report, RAND Corporation, 1958. [3.3.1]

R. Tajima, S. Kagami, M. Inaba, and H. Inoue. Development of soft and distributed tactile sensors and the application to a humanoid robot. *Advanced Robotics*, 16(4):381–397, 2002. [6.1.2]

J. W. Tangelder and R. C. Veltkamp. A survey of content based 3d shape retrieval methods. In *Shape Modeling International (SMI)*, pages 145–156, 2004. [1.3.1]

J. Tegin and J. Wikander. Tactile sensing in intelligent robotic manipulation–a review. *Industrial Robot: An International Journal*, 32(1):64–70, 2005. [1.1]

TekScan. Sensor map #5027. TekScan Inc., 2011. URL `http://www.tekscan.com/5027-pressure-sensor`. [2.5.2, 4.4.2.6]

S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, 15(2):111–127, 2003. [2]

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005. ISBN 0262201623. [3.1.1.2, 3.2.1, 3.3.1, 3.3.2, 5.2.1]

A. . Vallbo and R. Johansson. Properties of cutaneous mechanoreceptors in the human hand related to touch sensation. *Human Neurobiology*, 3:3–14, 1984. [2.5.2]

M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 691–698, Madison, WI, 2003. [4.3.2.1]

M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision (IJCV): Special Issue on Texture Analysis and Synthesis*, 62(1):61–81, 2005. [4.3.2.3]

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. URL `http://www.vlfeat.org/`. [4.3.2.2]

Willow Garage. Grasping/manipulation — willow garage. `http://www.willowgarage.com/pages/research/grasping-manipulation`, 2011. [1.2]

H. J. Wolfson and I. Rigoutsos. Geometric hashing: an overview. *IEEE Computational Science and Engineering*, 4:10–21, 1997. [5.1.2]

Y. Yanagida, M. Kakita, R. Lindeman, Y. Kume, and N. Tetsutani. Vibrotactile letter reading using a low-resolution tactor array. In *International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 400–406, March 2004. [1.3.3]

J. Zhang, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision (IJCV)*, 73:213–238, 2007. [4.1.1.1, 4.3.2.2]

Symbols Index

Recognition

$\mathbf{O}_i$	**Identity** of object i
$\mathbf{M}$	**Model** (e.g., set of maps) acquired in training and used for object recognition
$\mathbf{m}_i$	Individual **map element** i. Its form depends on the type of model being used.
$\mathbf{I}$	Tactile **image**

Distributions

$\mathbf{G}_{\mu,\sigma}$	**Gaussian distribution** with mean μ and standard deviation σ
η	A **normalization constant** to enforce a distribution summing to 1

Bayes Filters

z_t, $z_{1:t}$	**Measurement** (sensor reading) collected at time t, at times 1 through t
u_t, $u_{1:t}$	**Control** sent to the robot at time t, at times 1 through t
x_t, $x_{1:t}$	**State** (object identity and pose) at time t, at times 1 through t

Vita

Zachary Pezzementi received the B. S. degree in Engineering and B. A. degree in Computer Science from Swarthmore College in 2005, and enrolled in the Computer Science Ph.D. program at Johns Hopkins University the same year. He was inducted into the Sigma Xi honor society in 2003, received the M. S. E. degree in Computer Science in 2007, and was awarded a Link Foundation Fellowship for Simulation and Training in 2009. His research is in robotics and machine perception, with focuses on vision and touch sensing.

Starting in June 2011, Zachary will be working on vision systems for agricultural robots at Carnegie Mellon University's National Robotics Engineering Center.