Estimation and Inference for Grasping and Manipulation Tasks Using Vision and Kinesthetic Sensors

Thesis by

Paul Hebert

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2013

(Defended September 5, 2012)

To my love, Angela.

For my Mom and Dad.

Acknowledgments

First and foremost, I would like to thank my advisor Joel Burdick, who gave me the chance to return to Caltech, and ultimately giving me the freedom to explore a variety of projects. He served and will continue to serve as a role model for me both professionally and personally. He is without a doubt, a remarkable advisor, who is not only passionate about his student's research but also genuinely cares about them and wants them to excel.

In my beginnings in the Burdick group, I am grateful to have been under the knowledgeable wing of Michael Wolf, who gave me the right tools to start this thesis, his friendship, and the therapeutic time with Mowgli.

The rare simultaneous friendship and mentorship I have found in both Jeremy Ma and Nicolas Hudson during my PhD made this journey worthwhile and will provide lasting memories. Their encouragement along the way helped me to continue and ultimately finish this work.

I would like to thank Michael Mello, without whom, the first year at Caltech would have been impossible. I would also like to thank Christos Santis for his friendship, and the numerous breaks we took, especially getting beers and going to hockey games.

To my technical yet artistic colleague and bud Charles, I will forever cherish our time together during our formative McGill years. I am certain, that without your close friendship during that time, I could not have started down this road. You will forever remain my dear friend and ally.

To my confidant and oldest friend Fayez, whilst we shared too many memories to recount, our study times at Cafe Depot are my most fondest. I value and am grateful for our lasting friendship.

I want to thank my family for their love & support, especially my brother Glenn for his encouragement, his sage advice and our trip to California that ultimately opened the doors to Caltech.

The countless words of love, care, advice and support of my parents, their numerous sacrifices and provided opportunities, all beyond measure, made everything up to this point possible. I hope, in some small way, that this work makes up for all that they have given.

Last, but most important, my love Angela, who throughout the years and my struggles, believed in me, encouraged me and stood by me unwaveringly despite how far and how long we were apart. Above all, without her sacrifices and her continual support, for which I am infinitely grateful, I could not have completed this journey.

Abstract

This thesis presents a novel framework for state estimation in the context of robotic grasping and manipulation. The overall estimation approach is based on fusing various visual cues for manipulator tracking, namely appearance and feature-based, shape-based, and silhouette-based visual cues. Similarly, a framework is developed to fuse the above visual cues, but also kinesthetic cues such as force-torque and tactile measurements, for in-hand object pose estimation. The cues are extracted from multiple sensor modalities and are fused in a variety of Kalman filters.

A hybrid estimator is developed to estimate both a continuous state (robot and object states) and discrete states, called contact modes, which specify how each finger contacts a particular object surface. A static multiple model estimator is used to compute and maintain this mode probability.

The thesis also develops an estimation framework for estimating model parameters associated with object grasping. Dual and joint state-parameter estimation is explored for parameter estimation of a grasped object's mass and center of mass. Experimental results demonstrate simultaneous object localization and center of mass estimation.

Dual-arm estimation is developed for two arm robotic manipulation tasks. Two types of filters are explored; the first is an augmented filter that contains both arms in the state vector while the second runs two filters in parallel, one for each arm. These two frameworks and their performance is compared in a dual-arm task of removing a wheel from a hub.

This thesis also presents a new method for action selection involving touch. This *next best touch* method selects an available action for interacting with an object that will gain the most information. The algorithm employs information theory to compute an information gain metric that is based on a probabilistic belief suitable for the task. An estimation framework is used to maintain this belief over time. Kinesthetic measurements such as contact and tactile measurements are used to update the state belief after every interactive action. Simulation and experimental results are demonstrated using *next best touch* for object localization, specifically a door handle on a door.

The *next best touch* theory is extended for model parameter determination. Since many objects within a particular object category share the same rough shape, principle component analysis may be used to parametrize the object mesh models. These parameters can be estimated using the action selection technique that selects the touching action which best both localizes and estimates these

parameters. Simulation results are then presented involving localizing and determining a parameter of a screwdriver.

Lastly, the *next best touch* theory is further extended to model classes. Instead of estimating parameters, object class determination is incorporated into the information gain metric calculation. The best touching action is selected in order to best discern between the possible model classes. Simulation results are presented to validate the theory.

Contents

List of Figures

List of Tables

Notation

Estimation	
$\mathbf{\Theta}$	manipulator joint angles
θ	manipulator bias/error/offsets
$\mathbf{\Phi}$	neck joint angles
ϕ	neck bias/error/offsets
$G_{\mathcal{F}_1\mathcal{F}_2}$	homogenous transform from frame $\mathcal{F}_1$ to frame $\mathcal{F}_2$
$C_{f,F}$	contact mode between face f and finger F.
${}^f\boldsymbol{\beta}$	parametrized position of fingers on face f
$\mathcal{F}_A$	coordinate frame with label A
$\mathcal{M}$	polygonal mesh
f	face of mesh $\mathcal{M}$
v	vertex of mesh $\mathcal{M}$
n	face normal of mesh $\mathcal{M}$
X_k	state vector
P_k	state covariance
$\mathbf{f}(\cdot)$	state predictive model vector function
$\mathbf{F}_k$	state predictive matrix
$\mathbf{h}(\cdot)$	measurement model vector function
$\mathbf{H}_k$	measurement model matrix
Y_k	model parameter
u_k	command input
w_k	process noise
ζ_k	measurement noise
Q_k	process noise covariance matrix
R_k	measurement noise covariance matrix
$X_{k\|k-1}$	state prediction
$P_{k\|k-1}$	state prediction covariance
Z^k	measurement prediction
$bel(\cdot)$	belief
K_j^k	Kalman gain
$\chi_{i,k}$	sigma point state vector
$\mathcal{Z}_{i,k}$	sigma point measurement prediction
$\hat{z}_k$	mean sigma point measurement prediction
$\mathcal{X}_k$	state statistical Jacobian
$\mathbf{Z}_k$	measurement statistical Jacobian
P_{xx}	state-state covariance
P_{zx}	measurement-state cross covariance
$\mathcal{T}$	torque
$\mathbf{F}_g$	gravitational force
W	wrench

Next Best Touch	
a_i	action i
τ	discretized position along a particular action
IG	information gain
$\overline{IG}$	expected information gain
$E_z[\cdot]$	expected value function over z
$\hat{z}$	hypothetical measurement
$\mathcal{C}(\mathcal{M}_1, \mathcal{M}_2)$	collision function between mesh 1 and mesh 2
α	false positive probability in binary contact detection
β	true negative probability in binary contact detection
m	model parameter
M	model class

Chapter 1

Introduction

1.1 Motivation

Robots are more frequently asked to carry out complex tasks in unstructured environments. During tasks that involve grasping and manipulation of physical objects, accurate knowledge of the robot's and the grasped objects' pose is critical. Accurate knowledge of these poses is made difficult by common limitations of modern robotic systems. Kinematic errors due to joint sensor errors, tendon slack, calibration errors and perception errors in practical systems affect the quality of manipulation and grasping tasks, often leading to a poor grasp and task execution.

Particularly, when precise manipulation of objects is needed, a filtering and estimation framework is required to maintain accurate state estimates from a diverse set of sensors. Precise estimation requires a filtering and measurement system. Many available robots are equipped with multiple visual and kinesthetic sensor modalities. A framework to fuse the measurements from these multiple modalities is needed to provide an accurate state.

A common problem faced with grasping is the uncertainty in the position of the object in question. Visual pose estimation is often noisy and is not accurate enough for delicate grasping or manipulation. The use of touch before a grasp is made can be used to accurately locate the object relative to the hand. The choice of where to touch is an obvious problem.

Humans can easily identify objects they have not seen before, but such identification is currently still a difficult problem for machine vision. Similarly, grasping of unknown objects is very much an open research problem. Before grasping and manipulating these unknown objects, deliberate interaction with these objects such as touching and/or probing to build a geometric model may be used in order to find appropriate grasp-points for grasping.

This thesis develops a framework for estimation of both the manipulator and object state using multiple sensing modalities. The problem of touch is localizing with touch is also addressed.

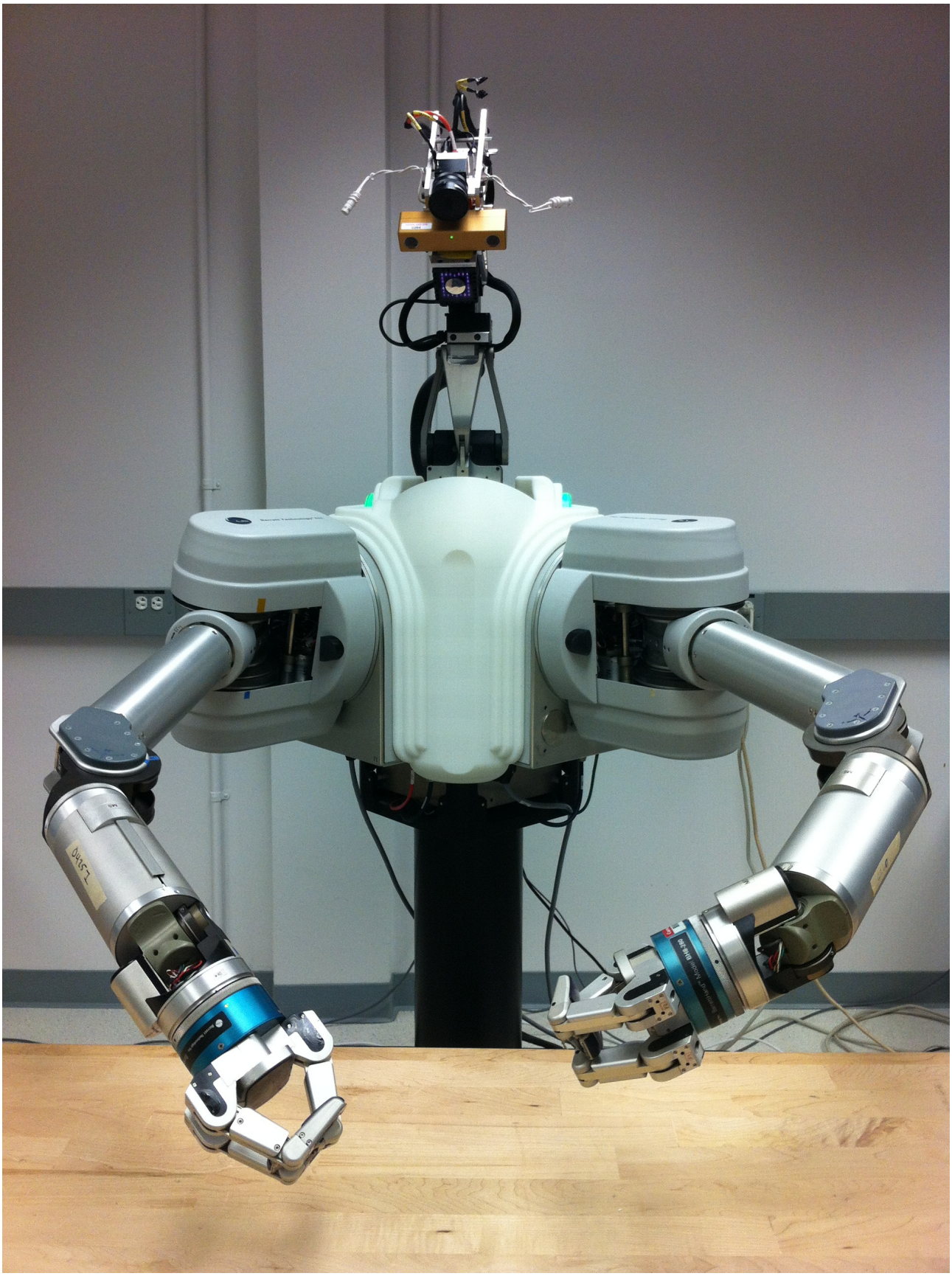


Figure 1.1: Dual-arm robot provided by DARPA for the autonomous robotic manipulation software (ARM-S) competition.

The above problems and issues were faced during the DARPA autonomous robotic manipulation software (ARM-S) challenge. The first phase of this challenge consisted of creating a software system composed of various algorithms thereby endowing the government furnished ARM-S robot, shown in Figure 1.1, with the ability to autonomously complete various tasks requiring precise manipulation, such as key insertion and drilling a point on a block of wood.

1.2 Review of Existing Literature

The ability to detect, localize, and track an object to be grasped or manipulated has been an important research thrust for some time. Earlier work focused on one primary sensor, namely vision [1–6]. Recently, a few works have considered the fusion of sensors which might naturally comprise a robotic grasping system. Most papers that discuss utilizing multiple sensors often use them at different stages of a grasp task. Allen et al. [7] used vision and tactile sensing to estimate the finger contact position and applied forces. In further work, Allen et al. [8] extends the sensor suite to include a force/torque sensor, however they do not develop a framework to combine the sensors synchronously. Prats [9] combines vision, force/torque and tactile sensors in a control scheme for manipulation tasks such as sliding a door open. Their work did not seek to accurately localize the object. Control is done via a virtual visual servoing method (VVS) in which they track an edge of the door. Fusion is not done in a traditional sense; first it done with only the vision and tactile sensor and then the results are then used as an input in an impedance force controller. Schmid et al. [10] also incorporates a multi-sensor control framework towards opening a door, however, the vision sensor is only used to detect the door handle.

In object localization work, most have not focused on using multiple sensors. Petrovskaya [11] uses only tactile sensors to localize an object using a novel particle filter approach. Similarly, Gadeyne et al. [12] uses a robot with force sensors only for object localization using Markov localization techniques. Corcoran [13] expanded the above work by proposing a model to incorporate hand-object tracking and does not incorporate any other sensors to the suite.

A hybrid system formulation is a natural approach in grasping and manipulation tasks. A large portion of the literature involves hybrid control in such tasks [14–18]. However, only a smaller portion concerns hybrid estimation. Gadeyne et al. [19] relates most to the contributions of this thesis - they incorporate a hybrid probabilistic framework for estimating various contact formations of a grasped object in compliant motion tasks. The approach in this thesis differ in that it applies a multiple model approach, and works well with an extended Kalman filter. While I implement a static estimator that is sufficient for the experiments presented in Chapters 3, a more sophisticated hybrid estimator such as a generalized pseudo-Bayesian estimator (GPB) or an interacting multiple model (IMM) [20] can be used for manipulation and tracking tasks.

In the field of computer vision research, shape and appearance have been a natural combination for both object modeling and object detection [21], [22], [23], as opposed to object tracking.

Appearance-based tracking, which typically refers to tracking in images, is a vast subject. Cootes [24] utilizes active appearance models, which are shape-free texture patches that are learnt from annotated training images and are iteratively matched in candidates images. These appearance models can only deform in ways which are expressed in the training set, and are inherited by active shape models [25]. Appearance based tracking of 2D blobs has also been popular, especially with the use of the mean-shift algorithm [26] and the extension through scale space [27]. Similarly, kernel-based approaches have also been implemented successfully [28]. The use of histograms have been widely used in visual tracking. [29] used color histogram in a probabilistic framework.

Similarly, the combination of shape and silhouettes has been used for model reconstruction [30], [31]. The use of silhouettes has had a large impact on people tracking [32], [33]. Using silhouettes as a method for tracking hand and manipulator have also been considered. Camarillo et al. [34] used silhouettes for tracking a flexible tendon manipulator to obtain its shape. Our experiments [35] on a tendon driven Barrett$^{\text{TM}}$WAM arm show a benefit from using silhouette information, the fusion of additional appearance, depth, and tactile information provides superior results. Sudderth et al. [36] estimated hand pose using a probabilistic framework based on nonparametric belief propagation. Their use of a Gibbs sampler may not lend itself to online tracking. Athitsos [37] estimated hand pose in a probabilistic manner as well, but uses an offline database of synthetic images to provide a ranked list of plausible poses, and is not a tracking framework.

Since vision can provide a large amount of information for accurate pose registration [38], many researchers have naturally used vision in one way or another for robotic grasping and manipulation. However most prior works have not combined multiple visual techniques and/or sensors. Kragic [2] used model-based pose estimation using vision by tracking edges to estimate grasp location of block objects. Saxena [3] learnt grasping points from 2D imagery using supervised learning on a training set of synthetic images. Multiple views are triangulated to provide grasp points to the objects. Saxena [4] also learnt orientation of objects using 2D imagery deemed useful for object grasping. Collet [39] used local descriptors from a set of training images to build a metric 3D model. Pose registration is done by matching the local descriptors and a novel combination of the RANSAC and mean-shift algorithm. Visual hand tracking has been quite popular for gesture recognition [36] but has also been researched for manipulation [40]. Prats [9] combined various sensors for multi-sensor controller used vision for pose estimation and used the method of virtual visual servoing. Our previous work [41] similarly used multiple sensors but for object pose estimation. The visual sensor tracked 3D SIFT features of objects in the hand to provide a pose-based measurement.

One of the most recent and comprehensive studies of object and manipulator pose estimation has been done by Krainin et al. [42], who estimate both the object's pose and the joint angles

of the manipulator for the application of object modeling. The approach in this thesis differs in that it combines multi-sensory data with different techniques for extracting information from the available images. We also fuse visual data with tactile data for accurate object pose estimation in manipulation tasks. To the author's knowledge, combining such information for tracking both object and manipulator has not been addressed previously.

The process of online manipulator tracking based on mechanism shape is a form of on-line calibration. There is a vast literature on calibration of robotic manipulators and is thought as a well-studied field. Early work dealt with parallel mechanisms and Stewart platforms. Visual based calibration is a natural form of calibration and has an extensive literature. Zhuang [43] attempted to calibrate both the visual sensor and the manipulator simultaneously to reduce error propagation. On-line calibration is often coined self-calibration in that there is no external measurements and/or ground truth data. A classic paper on self-calibration is by Ma [44] in which calibration is solved for the hand-eye (head-eye) geometry and the intrinsinc parameters of the camera. This only applies to one kinematic chain ending at the camera. More closely to our problem is the work by Bennett et al. [45] where they analyze the closed chain between the manipulator and the vision system. For an overview of robot calibration in general, the reader is directed to a reference [46].

There is a vast literature in kinesthetic and tactile estimation. Early works include Grimson [47], which used tactile sensors to identify and localize polyhedral objects, and early work of Allen [7, 8, 48], which combined vision with tactile sensors to identify and localize objects. Gadeyne [12] use a Markov based approach for force controlled robots and demonstrated localization on a box. Gorges [49] used tactile and force sensors to explore an object using skills of continuous and discrete movements (for example, following an edge). Recently, tactile localization has become increasingly prevalent with the advent of better sensors. Corcoran and Platt [13,50] introduce a particle filter and a novel measurement model for localizing an object in hand on Robonaut. Lastly, Petrovskaya [51] localizes a box based on tactile sensors and a novel particle filter to overcome the computational complexity - this work adopts their measurement model.

Action selection refers to choosing the best action from a set possible actions or behaviors which maximize a cost function. The problem of selecting actions is often encountered in the motion planning literature. While there are various ways to perform action selection, the approach in this work is based on expected information gain. Generally, information gain has been used extensively in the *next best view* problem, in which a robot actively chooses where to view or look next in order to gain the most knowledge of the problem at hand. In visual active search, Davison [52] used information theory with Gaussian uncertainty models for guided image processing, such as feature tracking. In the domain of active recognition, early work by Arbel [53] on gaze planning used an entropy map to create a trajectory that maximizes disambiguation of object recognition. In recent work, [54] optimizes a cost metric based on information gain to determine the best action for model

identification and pose estimation of a certain object. In terms of model generation, Krainin [42] uses the change in entropy to determine the next best view in order to build a 3D surface model of a grasped object.

Information gain has been extensively used in robotic motion planning and exploration. Thrun, Fox and Burgard [55] proposed to actively sense and navigate in a localization context of occupancy grids by minimizing entropy to determine the next best actions. A major difference between our work is that for each choice of action, we make no assumption that the action will complete. More recently, in the domain of exploration, Stachniss et al. [56] also uses the expected information gain to determine the best action to explore. Here, a Rao-Blackwellized particle filter is used to localize and map the environment. The novelty of this paper is computing the expected gain on two random variables, the pose of the robot and the map of the environment. However they too, also assume that each action will complete.

In the context of manipulation, Hsiao's [57] tactile exploration of objects implemented a decision-theoretic approach and an approximate POMDP, instead of information theory, to select actions for exploration. Schneider [58] uses a bag-of-features approach that combines tactile sensors with an information gain to determine a grasping strategy for object identification.

The contribution of this thesis is automating touch-based localization by exploiting information theory to determine the *next best touch* action. Also, unlike previous work with information-gain-based action selection, we consider actions as nondeterministic, since the interaction with the object is modeled to be stochastic due to the uncertainty in the state of both the object and robot.

1.3 Thesis Organization and Contributions

The remainder of this thesis is structured as follows. Chapter 2 reviews the technical background material and notation on traditional pose registration methods, camera models, and various filtering frameworks used to develop the thesis' contributions and experimental results.

Chapter 3 develops a novel framework for simultaneous object and manipulator tracking. First, the assumed system configuration used for experimental results is discussed, followed by the grasping and object models used for estimation. The general state representation used in this thesis is described. Next, the state predictive model for grasped objects is developed.

The measurement models used in the estimation process are broken down into their 3 parts: Section 3.5.1 visual manipulator tracking, Section 3.5.2 visual object tracking and Section 3.5.3 kinesthetic measurements. Visual manipulator tracking is further broken down into 3 sensory cues. The first is appearance-based that utilize features on the manipulator and hand. These features are found in left and right images of the stereo system using the technique described in Section 3.5.1.1 The disparity from the left and right images is computed and used to produce a 3D pose of these

features. The second cue is shape-based and exploits the sensory information from a 3D-RGB camera. Section 3.5.1.2 discusses how points are extracted from the cloud of points produced by 3D-RGB sensors that match the shape of the manipulator and are used in a variant of articulated iterated closest point. The corrected joint angles are then used to as the measurement. The last cue uses the silhouette of the manipulator. The silhouette of the manipulator's model is first rendered using an offscreen OpenGL^{TM}framework to produce a binary image. The contour of which is easily extracted and iteratively, matched using a technique described in Section 3.5.1.3, to edges in the real camera image.

Section 3.5.2, visual object tracking, is similarly broken down into 3 sensory cues. The first is also feature based, in which SIFT or SURF features are detected and matched to a learnt set of features for each object. The 3D locations of the features found from the stereo cameras are then used as measurements. The second cue is also shape-based but uses a one-step closest point matching, which is described in Section 3.5.2.2. The third is also silhouette-based in which the offscreen renderer, renders the object model at various sigma points and the contours of which are matched to the image of the real camera image. Section 3.5.2.3 describes this process in detail.

The kinesthetic measurement models are described in Section 3.5.3. First, the force-torque measurement model is derived for use with the force-torque sensor. Second, the tactile measurement models are also described that match the tactels location and normal to the object surface and surface normal in Section 3.5.3.2. Lastly, in Section 3.5.3.3, when tactile information is not available, finger position can be used as a pseudo-measurement of the joint measurements of the hand.

Hybrid estimation theory is developed for finger contact modes (how fingers contact the object's surface) and is described in Section 3.6. A state multiple model (SMM) estimator manages the discrete state of the finger contact modes and maintain a mode probability over time.

Next, model parameter estimation is developed for estimating non-dynamic quantities such as mass and center of mass of the grasped object in hand. Two frameworks are described in Section 3.7. First, joint estimation typically augments the state vector to include these model parameters and the same filter of choice is used. Second, dual estimation maintains another filter that run in parallel that updates only the model parameters.

Sections 3.8 develops a novel estimation framework for two arm manipulation. While preceding sections dealt with single-arm, single-object grasping, this section explores tasks that require two handed manipulation. One framework involves running two filters in parallel, one for each arm, and the second framework augments the state vector to include both arms. To use the predictive model, both frameworks maintain two estimates of the object's pose relative to each hand. Since these two states represent the same object, the states must be constrained to point at the same pose in space. The two independent filters compute a simple mean estimate the same location in space, while the augmented filter has the ability to include nonlinear constraint equations into the filter. Both of

these methods are described in Section 3.8.1 and Section 3.8.2, respectively.

Chapter 4 develops a novel method for action selection involving touch. This new algorithm, termed *next best touch*, selects the best touching action which attempts to gain the most information from interacting with an object. Using information theory and an estimation framework, an information metric is derived based on a probability belief that depends upon the task. This belief, representing the certainty, may be on the pose of the object, a model parameter or a model class of an object. Section 4.1 investigates the *next best touch* for localization. Section 4.4.1 extends this to include an unknown model parameter and Section 4.4.3 extend the theory to model classes.

Chapter 5 reviews the contributions of the thesis and discusses future research directions.

Chapter 2

Background

This chapter provides background and context for the contributions of this thesis. Since the thesis concerns the pose estimation of grasped objects and the pose of manipulators, standard pose estimation techniques are described in Section 2.1 and Section 2.2. Secondly, since many of the measurement models employed utilize visual information, the CAHVOR camera model used in the experiments is discussed in Section 2.3. Lastly, since the core of the thesis concerns estimation, the various filters used in the experimental results are described in detail in Section 2.4.

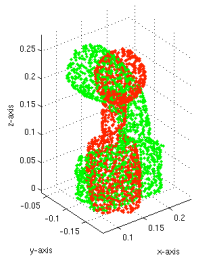
2.1 Iterated Closest Point

Iterated Closest Point (ICP) is a classical method for geometric alignment of two sets of 2D or 3D point data. ICP is often used as a pose registration tool to align point cloud data (typically in 3D). Pose registration of objects is common problem in robotics, particularly in manipulation. ICP matches two point clouds, one being the point cloud of the object model and the other being the sensed data of that object. Although many variants of ICP have been proposed (an overview of these variants may be found in [59]), the general aim is to minimize a cost function that describes the alignment between the two point clouds.

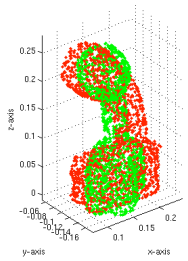
A brief overview of the ICP algorithm by Besl [60] is described. ICP attempts to align the model points $X = \{x_i\}, i = 1, \ldots, N_x$ and the sensed data points $P = \{p_i\}, i = 1, \ldots, N_p$ assuming $N_x = N_p$ where p_i corresponds in some way to a matching point x_i with the same index. The cost function typically takes a mean square form:

$$E(q) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - R(q)p_i - t\|^2 , \tag{2.1}$$

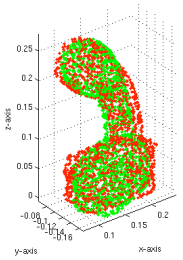
where q is the quaternion representing a rotation, between point cloud frames with $R(q)$ representing the SO(3) rotational matrix and t is the vector representing the translation. To minimize Equation 2.1, it will be required to calculate the center of masses of both the model points and the sensed



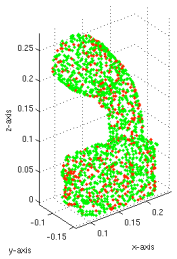
(a) Initial starting point.



(b) Intermediate matching point.



(c) Intermediate matching point.



(d) Final matched model.

Figure 2.1: Iterative closest point matching process of flashlight model points, x_i (red) to the data points, p_i (green)

data points:

$$\mu_x = \frac{1}{N_x}\sum_{i=1}^{N_x} x_i \quad \text{and} \quad \mu_p = \frac{1}{N_p}\sum_{i=1}^{N_p} p_i \,. \tag{2.2}$$

The cross covariance between the model point X and the data points P is:

$$\Sigma_{px} = \frac{1}{N_p}\sum_{i=1}^{N_p}[(p_i - \mu_p)(x_i - \mu_x)^T] \tag{2.3}$$

$$= \frac{1}{N_p}\sum_{i=1}^{N_p}\left[p_i x_i^T\right] - \mu_p\mu_x^T \,. \tag{2.4}$$

An anti-symmetric matrix $A_{ij} = (\Sigma_{px} - \Sigma_{px}^T)$ is formed and is used to form a column vector $\Delta = [A_{23}A_{31}A_{12}]^T$. The cross covariance and the column vector are used to form the symmetric 4 x 4 matrix $Q(\Sigma_{px})$:

$$Q(\Sigma_{px}) = \begin{bmatrix} tr(\Sigma_{px}) & \Delta^T \\ \Delta & \Sigma_{px} + \Sigma_{px}^T - tr(\Sigma_{px})I_3 \end{bmatrix} . \tag{2.5}$$

The unit eigenvector that corresponds to the maximum eigenvalue of $Q(\Sigma_{px})$ is the optimal quaternion, q. The optimal translation is thus:

$$t = \mu_x - R(q)\mu_p \,. \tag{2.6}$$

2.2 Articulated Iterated Closest Point

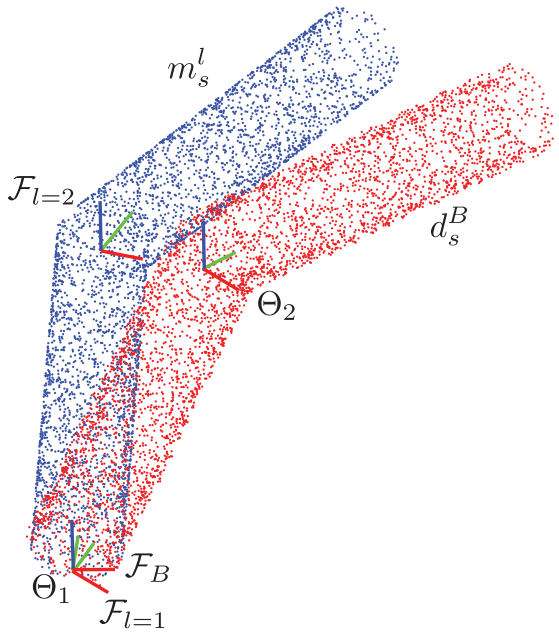


Figure 2.2: General formulation for articulated iterated closest point (AICP) diagram with two articulated cloud of points. The key variables and frames are shown

Articulated Iterated Closest Point (AICP) is simply an extension to the general ICP algorithm discussed in Section 2.1. Instead of minimizing an error function to solve for the 6DOF pose of an object, AICP aims at minimizing a similar error function but to solve for joint parameters of a rigid articulated body like a manipulator. Manipulators are commonly constructed from 1DOF rotational joints ($\mathbf{\Theta}$). In this case, one can formulate the error function as:

$$E(\mathbf{\Theta}) = \sum_{l=1}^{N_L} \sum_{s=1}^{N_{p_l}} \min_i \| G_{Bl}(\mathbf{\Theta})\, m_i^l - d_s^B \| \,. \tag{2.7}$$

This function is the sum of squared distances between model points m^l of l^{th} link to the closest sensed data point in the world frame $\mathcal{F}_B$. The minimization of this error function may be done via a gradient descent approach or an often used variant like the Levenberg-Marquand algorithm [61]. Section 3.5.1.2 discusses the derivation of such a gradient descent type approach as well as a fast matching approximation specialized for cylindrical links.

2.3 CAHVOR Camera Model

In utilizing monocular or stereo cameras as part of the estimation system hardware, a model must be used to calibrate the cameras both extrinsically and intrinsically. The simplest is the thin lens

model. The most common is the pinhole camera model, where the lens model's aperture is shrunk to zero or realistically infinitesimally small. Other camera projections often used are perspective projection and orthographic projection. Perspective projection preserves linearity and parallel lines do not stay parallel. On the contrast, orthographic projection becomes similar to a perspective projection when the distance to the scene is much larger than the focal length.

Two popular complex camera models are the Tsai model [62] and the family of CAHV models that was originally published by Gennery [63]. The CAHVOR models add corrections to lens distortions, which are not included in the CAHV model. The CAHVORE model is an "E"xtended version of the CAHVOR models and can handle very large distortions that are found in fish eye lenses.

Each letter in the model name, represents a parameter in the model. The C component specifies the vector that represents the translation of the origin of the world coordinate system to the focus point of the camera. In other words, it specifies the camera center of focus, i.e., the 3D coordinates of the pinhole focus point.

The A component specifies the vector that points in the normal direction of the sensor plane.

The H and V components specifies vectors about the Horizontal and Vertical information vectors. Although, these vectors seem arbitrary, they provide an efficient way to compute the 2D image point (x) from a 3D world point (P):

$$x = \frac{(P - C)H}{(P - C)A} \,. \tag{2.8}$$

The O component specifies the optical axis, which is used only for lens-distortion correction. This is typically nearly equal to A.

Finally, the R component specifies the radial lens distortion coefficients.

2.4 Estimation Frameworks

Sections in the remainder of this thesis uses different estimation frameworks and an overview of these frameworks are described below. These frameworks aim at computing a belief about the state of the system, which reflects a model of the robot's internal knowledge about the state of the environment.

The *Markov assumption* is commonly used in practice and is used throughout this thesis. The assumption postulates that if the state is known, then all past and future measurements are independent of the state. The assumption allows for simplification of the following frameworks.

In addition, some frameworks, particularly the Kalman variants, require a predicted measurement. To avoid confusion in the notation throughout this thesis, Z is used to denote predicted measurements and z is used to denote actual measurements.

2.4.1 Bayes' Filter

The Bayes' filter is the most general algorithm for calculating the state probability distribution (also referred to as the state belief). The Bayes' filter is discrete time and is recursive. The Bayes' filter possesses two essential steps. At each time step, k, the first step is to process the control input u_k by calculating a belief over the state X_k based on the prior belief over the state X_{k-1} using a predictive model, and is often called the *prediction* step. The second step is to process the sensory measurements by calculating the posterior belief based on the belief in the prediction step using a measurement model. This step is often called the *measurement update* step. The algorithm is simple - for all X_k compute the posterior probability $p(X_k|z_{1:k}, u_{1:k})$:

$$\text{Prediction:} \quad p(X_k|z_{1:k-1}, u_{1:k}) = \int p(X_k|X_{k-1}, z_{1:k-1}, u_k)\, p(X_{k-1}|z_{1:k-1}, u_{k-1})\, dX_{k-1} \tag{2.9}$$

$$\text{Update:} \quad p(X_k|z_{1:k}, u_{1:k}) = \frac{1}{\eta}\, p(z_k|X_k, z_{1:k-1}, u_{1:k})\, p(X_k|z_{1:k-1}, u_{1:k})\,, \tag{2.10}$$

where η is a normalizing factor, which need not be usually computed as the belief can be normalized after calculating the entire distribution. If η needs to be explicitly calculated:

$$\eta = p(z_k|z_{1:k-1}, u_{1:k}) = \int p(z_k|\hat{X}_k, z_{1:k-1}, u_{1:k})\, p(\hat{X}_k|z_{1:k-1}, u_{1:k})\, d\hat{X}_k. \tag{2.11}$$

$p(X_k|X_{k-1}, z_{1:k-1}, u_k)$ is the state transition probability, which is based on a predictive model. It specifies how the environment state evolves as a function of the control inputs, u_k. Robot environments are stochastic, and that is why the state transition is not deterministic, but rather probabilistic. The measurement probability, $p(z_k|X_k, z_{1:k-1}, u_{1:k})$, based on a measurement model which captures the probabilistic relation between z_k and the state, X_k.

2.4.2 Histogram Filter

The histogram filter is a discretized version of the Bayes' filter for continuous state spaces and is the simplest nonparametric filter. This filter quantizes and approximates the continuous state space, X_k, into a finite regions: $\text{dom}(X_k) = x_{1,k} \cup x_{2,k} \cup \ldots x_{N,k}$ and represents the posterior belief over each region as a single probability value at time t_k.

Here $x_{n,k}$ is the n^{th} bin of the N total bins that comprise the state space quantization. The histogram filter as an approximation can be formulated as the discrete Bayes' filter:

$$p(X_k = x_i|z_{1:k-1}, u_{1:k}) = \sum_{j=1}^{N} p(X_k = x_i|X_{k-1} = x_j, z_{1:k-1}, u_k)\, p(X_{k-1} = x_j|z_{1:k-1}, u_{k-1}) \tag{2.12}$$

$$p(X_k = x_i|z_{1:k}, u_{1:k}) = \frac{p(z_k|X_k = x_i, z_{1:k-1}, u_{1:k})\, p(X_k = x_i|z_{1:k-1}, u_{1:k})}{\sum_{j=1}^{N} p(z_k|\hat{X}_{k-1} = x_j, z_{1:k-1}, u_{1:k})\, p(\hat{X}_k = x_j|z_{1:k-1}, u_{1:k})}\,. \tag{2.13}$$

2.4.3 Kalman Filter

One the most widely used Bayes' filters is the Kalman filter [64] for both linear and gaussian type systems. The Kalman filter and its variants are used to compute the belief of continuous states and are not suitable for discrete state like the histogram filter. In addition, alone it cannot handle hybrid states spaces, but can be combined with other Kalman filters in parallel as discussed in Section 3.6.

The Kalman filter represents the belief at time k by a Gaussian PDF for X_k with mean $\overline{X}_k$ and a covariance P_k. The predictive dynamic model that leads to the state transition probability $p(X_k|u_k, X_{k-1})$ must be linear and with added Gaussian noise. Provided it is linear, the state transition can be expressed by:

$$X_k = F_k X_{k-1} + B_k u_k + w_k \,. \tag{2.14}$$

F_k and B_k are matrices of size $L \times L$ where L is the dimension of the state vector X_k. The random variable w_k is a zero mean Gaussian random variable which models the process uncertainty having covariance Q_k.

The measurement equation that leads to the probability $p(z_k|\hat{X}_k, z_{1:k-1}, u_{1:k})$ must be also linear with added zero mean Gaussian noise ζ_k, that is, it takes the following form:

$$Z_k = H_k X_k + \zeta_k \,. \tag{2.15}$$

H_k is the measurement matrix of size $k \times L$, where k is the dimension of the measurement vector. The random variable ζ_k describes the errors on the measurement models and is assumed to be a multivariate Gaussian with zero mean and covariance R_k.

The mean and covariance of the filter are propagated via Equations (2.14) in the following manner:

$$\overline{\mu}_k = F_k \mu_{k-1} + B_k u_k \tag{2.16}$$

$$\overline{\Sigma}_k = F_k \Sigma_{k-1} F_k^T + Q_k \,. \tag{2.17}$$

The update step uses Bayes Theorem with Equation 2.10 along with the measurement equation 2.15. The Kalman gain matrix K_k is a variable that determines how the filter should weight the measurements. Highly uncertain measurements will yield a low Kalman gain and vice versa,

$$K_k = \overline{\Sigma}_k H_k^T \left(H_k \overline{\Sigma}_k H_k^T + Q_k \right)^{-1} \tag{2.18}$$

$$\mu_k = \overline{\mu}_k + K_k \left(z_k - H_k \overline{\mu}_k \right) \tag{2.19}$$

$$\Sigma_k = \left(I - K_k H_k \right) \Sigma \,. \tag{2.20}$$

The optimal Kalman filter can illustrate the benefit of fusing complementary sources of data. Provided that the measurements are complementary and the underlying system is first order, the optimal Kalman filter fuses the measurements optimally according to the sensor's uncertainty characteristics [64, 65].

2.4.4 Extended Kalman Filter

The extended Kalman filter is an approximation for nonlinear state transitions and measurement update:

$$X_k = \mathbf{f}\left(X_{k-1}, u_k\right) + w_k \tag{2.21}$$

$$Z_k = \mathbf{h}(X_k) + \zeta_k\,. \tag{2.22}$$

The key is to linearize these function about the current state point. The Jacobian matrices produced from the linearization of functions $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are:

$$F_k = \frac{\partial \mathbf{f}}{\partial X_k}, \qquad B_k = \frac{\partial \mathbf{f}}{\partial u_k}, \qquad H_k = \frac{\partial \mathbf{h}}{\partial X_k}\,. \tag{2.23}$$

In this case, the mean and covariance of the filter are propagated in the prediction step in a similar manner:

$$\overline{\mu}_k = \mathbf{f}\left(\mu_{k-1}, u_k\right) \tag{2.24}$$

$$\overline{\Sigma}_k = F_k \Sigma_{k-1} F_k^T + Q_k\,. \tag{2.25}$$

In the updating step, the Kalman gain matrix is also computed a similar fashion, and the updated mean and covariance for the Gaussian PDF on X_k are given by:

$$K_k = \overline{\Sigma}_k H_k^T \left(H_k \overline{\Sigma}_k H_k^T + Q_k\right)^{-1} \tag{2.26}$$

$$\mu_k = \overline{\mu}_k + K_k\left(z_k - \mathbf{h}\left(\overline{\mu}_k\right)\right) \tag{2.27}$$

$$\Sigma_k = \left(I - K_k H_k\right)\Sigma\,. \tag{2.28}$$

2.4.5 Sigma Point Kalman Filter

The Sigma Point Kalman Filter (SPKF), a non-linear filter which uses the *unscented transform*, was first introduced by Uhlmann and Julier [66]. The SPKF represents the state probability distribution as Gaussian using a set of deterministic sample points. The points capture the mean and covariance of the gaussian random variable to the 3rd order (in Taylor series expansion) when propagated through a nonlinear system. The unscented transform is a mathematical method for recapturing the

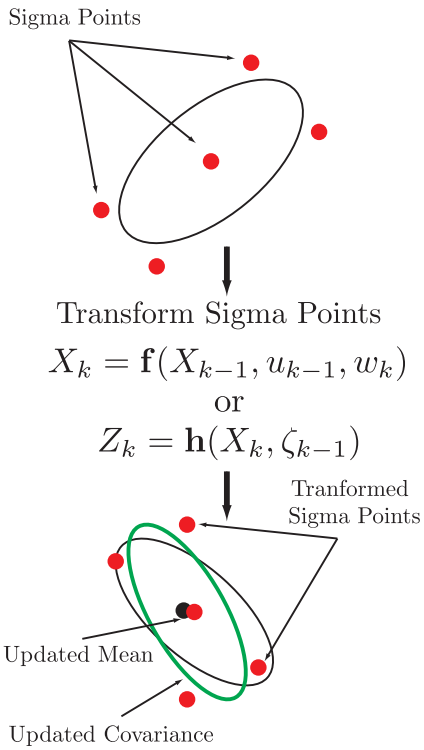


Figure 2.3: Unscented transformation of sigma points through prediction and measurement models.

lower-order statistical moments of a random variable after mapping by a nonlinear function. These deterministic sample points are referred to as *sigma points* and are sampled from the covariance of the state distribution.

In the same fashion as the discrete-time Bayes' filter, there exists *prediction* and *measurement update* steps. Assume there exists a nonlinear dynamic predictive model $\mathbf{f}$, which is a function of the system state X_k and control inputs u_k at time t_k. Similarly, there exists a nonlinear measurement model $\mathbf{h}$ relating measurements Z_k at time t_k to state X_k.

$$X_k = \mathbf{f}(X_{k-1}, u_{k-1}, w_k)\,, \qquad w_k \sim \mathcal{N}(0, Q_k) \tag{2.29}$$

$$Z_k = \mathbf{h}(X_k, \zeta_k)\,, \qquad \zeta_k \sim \mathcal{N}(0, R_k)\,. \tag{2.30}$$

The discrete time Gaussian white noise processes, w_k and ζ_k, respectively, model process and measurement uncertainty, while Q_k and R_k represent their respective covariances. An estimate of the state X_k and the belief of the state P_k are propagated as sigma points χ in two stages of *prediction* and *update*.

2.4.5.1 Prediction

As a first step to predict the state forward in time, the state and covariance are augmented to form the augmented state, X^a and its augmented covariance, P^a are constructed as follows:

$$X^a_{k-1} = \begin{bmatrix} X_{k-1} \\ \vec{0} \end{bmatrix}, \quad P^a_{k-1} = \begin{bmatrix} P_{k-1} & \mathbf{0} \\ \mathbf{0} & Q_k \end{bmatrix}. \tag{2.31}$$

The $2L+1$ *sigma points*, with $L = \dim X^a$, are constructed from the augmented state as follows:

$$\chi^a_{0,k-1} = X^a_{k-1} \tag{2.32}$$

$$\chi^a_{i,k-1} = X^a_{k-1} + \sqrt{(L+\lambda)P^a_{k-1}}^{\,i} \tag{2.33}$$

$$\chi^a_{i+L,k-1} = X^a_{k-1} - \sqrt{(L+\lambda)P^a_{k-1}}^{\,i}, \quad i = 1 \dots L \tag{2.34}$$

$$\text{with } \lambda = \alpha^2(L+\kappa) - L\,,$$

where $\chi^a_{l,m}$ is the l^{th} sigma point vector at time t_m, and the square root should be interpreted as the matrix square root. the notation $\sqrt{\cdot}^{\,i}$ implies the i^{th} column of the square root matrix. The variable α determines the spread of sigma points, and κ is a secondary scaling parameter usually set to zero.

As the state was initially augmented, separate the resulting augmented sigma points to obtain the state, χ_i, and noise, $\mathcal{W}_i$, portions:

$$\chi^a_{k-1} = \begin{bmatrix} \chi_{i,k-1} \\ \mathcal{W}_{i,k-1} \end{bmatrix}, \tag{2.35}$$

to pass the state sigma points through the dynamical model and obtain the *predicted sigma points*:

$$\chi_{i,k} = \mathbf{f}(\chi_{i,k-1}, \mathcal{W}_{i,k-1})\,. \tag{2.36}$$

From these predicted sigma points, the predicted state and belief can be appropriately combined:

$$X_{k|k-1} = \sum_{i=0}^{2L} \pi_i \, \chi_{i,k} \tag{2.37}$$

$$P_{k|k-1} = \sum_{i=0}^{2L} \pi_i \, [\chi_{i,k} - X_{k|k-1}][\chi_{i,k} - X_{k|k-1}]^{\mathsf{T}} \tag{2.38}$$

$$\text{where } \pi_i = \begin{cases} \frac{\lambda}{L+\lambda} & \text{if } i = 0 \\ \frac{1}{2(L+\lambda)} & \text{otherwise .} \end{cases}$$

2.4.5.2 Update/Correction

In the update step the sigma points are propagated through the measurement model to produce a set of predicted sigma measurements.

$$\mathcal{Z}_{i,k} = \mathbf{h}(\chi_{i,k}), \quad i = 1 \ldots L \, . \tag{2.39}$$

The mean predicted measurement and the innovation covariance may be calculated in the usual way:

$$\hat{z}_k = \sum_{i=0}^{2L} \pi_i \; \mathcal{Z}_{i,k} \tag{2.40}$$

$$P_{zz,k} = \sum_{i=0}^{2L} \pi_i \; (\mathcal{Z}_{i,k} - \hat{z}_k) \, (\mathcal{Z}_{i,k} - \hat{z}_k)^T + R_k \, . \tag{2.41}$$

In order to compute the Kalman gain K_k, the cross state-measurement covariance must be computed by:

$$P_{zx,k} = \sum_{i=0}^{2L} \pi_i \; \left(\chi_{i,k} - X_{k|k-1}\right) (\mathcal{Z}_{i,k} - \hat{z}_k)^T \tag{2.42}$$

$$K_k = P_{zx,k} \, P_{zz,k} \, . \tag{2.43}$$

The updated state and covariance may be found using the Kalman gain K_k:

$$X_k = X_{k|k-1} + K_k \, (z_k - \hat{z}_k) \tag{2.44}$$

$$P_k = P_{k|k-1} - K_k \, P_{zx,k}^T \, . \tag{2.45}$$

2.4.5.3 Update for High-Dimensional Measurement Vector

Since part of the sensor data generated by the stereo vision and 3D ranging sensors consists of "point clouds", the dimensions of the measurement vector can potentially be very large. In this case, the Kalman gain computation is costly due to the matrix inversion of a potentially large matrix.

Therefore we adopt the serial processing method of [67], which uses the Sherman-Morrison-Woodbury identity to alleviate the large matrix inversion. As with the standard UKF, the predicted sigma points are passed through the measurement model to produce a set of N predicted measurements, with N being the number of actual measurements:

$$\mathcal{Z}_{i,k}^{n} = \mathbf{h}(\chi_{i,k}), \quad n = 1 \dots N,\ i = 1 \dots L\,. \tag{2.46}$$

The mean predicted measurement, $\hat{z}_k^n$, and its statistical Jacobian factor, $\mathbf{Z}_k^n$, are found as:

$$\hat{z}_k^n = \sum_{i=0}^{2L} \pi_i\, \mathcal{Z}_{i,k}^{n} \tag{2.47}$$

$$\mathbf{Z}_k^n = \left[\dots \sqrt{\pi_i}\, (\mathcal{Z}_{i,k}^{n} - \hat{z}_k^n) \dots \right]. \tag{2.48}$$

Finally, the updated state and belief are obtained using the following serial computation by processing each actual measurement z_k^n and the associated noise covariance, $R_{n,k}$ sequentially (summation) as opposed to stacking all measurements:

$$\mathbf{C}_k = \sum_{n=1}^{N} (1 + \mathbf{Z}_k^{n\mathsf{T}} R_{n,k}^{-1} \mathbf{Z}_k^n) \tag{2.49}$$

$$\mathbf{d}_k = \sum_{n=1}^{N} Z_k^{n\mathsf{T}} R_{n,k}^{-1} (z_k^n - \hat{z}_k^n) \tag{2.50}$$

$$X_{k|k} = X_{k|k-1} + \mathcal{X}_k \mathbf{C}_k \mathbf{d}_k \tag{2.51}$$

$$P_{k|k} = \mathcal{X}_k \mathbf{C}_k \mathcal{X}_k^{\mathsf{T}} \tag{2.52}$$

where z_k^n is the n^{th} actual sensor measurement at time t_k, and $\mathcal{X}_k$ is the state statistical Jacobian:

$$\mathcal{X}_k = \left[\dots \sqrt{\pi_i}\, (\chi_{i,k} - X_{k|k-1}) \dots \right]. \tag{2.53}$$

Chapter 3

Estimation for Grasping and Manipulation

This chapter discusses the problem of state estimation for robotic grasping and manipulation. The assumed robotic system sensory configuration is first discussed, followed by corresponding object and state models used in the estimation process. Both the predictive and the various visual and kinesthetic measurement models implemented are presented in detail. A hybrid estimation scheme for estimating both continuous states (such as the grasped object pose) and discrete states (such as the discrete object contact modes), respectively, are presented. Experimental results are first shown for the case in which the robot grasps an object statically. Dual estimation principles for estimating both state and model parameters simultaneously are presented. Specifically, the estimation of the robot, the grasped object state and the center of mass parameter are discussed. Estimation for coordinated dual-arm grasping and manipulation is also presented. Specifically, details of the dual-arm predictive models, the measurement models and the use of non-linear constraints into the two arm estimation framework are presented. Lastly, visual arm tracking, dual estimation and dual-arm estimation experimental results are presented using the DARPA ARM-S robot described in Section 1.1.

3.1 Assumed System Configuration

The estimation approach developed in this thesis is based on the following assumptions and assumed general system configuration (see Figure 3.1). A serial chain manipulator is rigidly affixed to a base, or to a (possibly moving) torso. The manipulator is assumed to possess 6 or more internal degrees of freedom (though the technique could be adapted to a kinematically insufficient manipulator which possesses 5 or fewer DOF.).

- Let $\mathbf{\Theta}$ denote the joint variables of this mechanism.
- Let $\mathcal{F}_B$ denote a reference frame rigidly affixed to the base or torso which supports the arm.

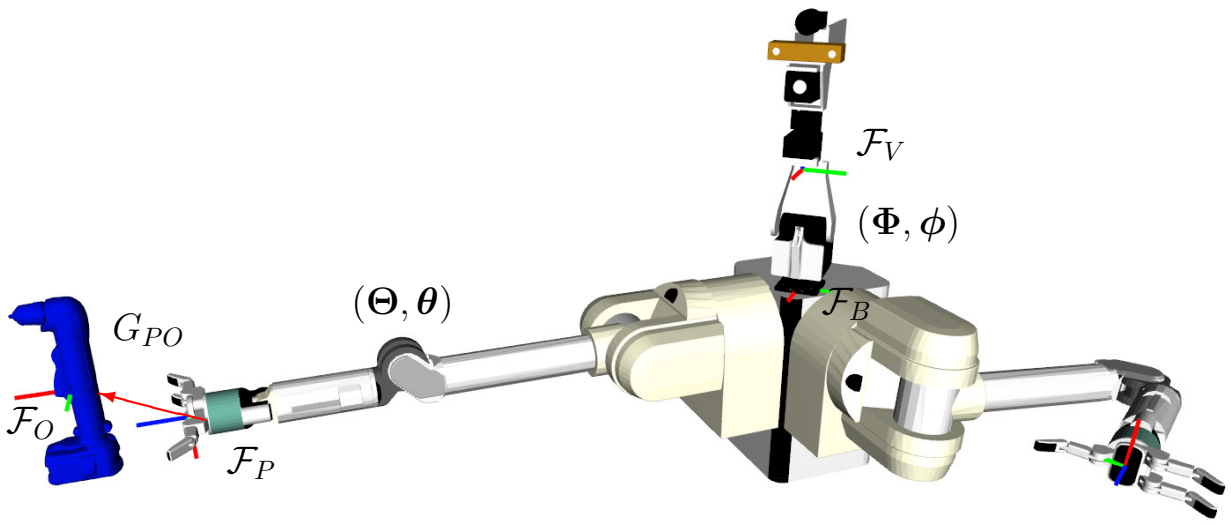


Figure 3.1: System configuration and state representation

- Let $\mathbf{\Phi}$ denote the joint variables of the neck mechanism.
- Let $\mathcal{F}_V$ denote a frame attached to the sensor suite.
- Let $\mathcal{F}_P$ denote a "palm" reference frame attached to the manipulator's distal end.

It is assumed that an articulated multi-fingered robot hand is attached to the manipulator, with a fixed offset relative to the palm frame. It is also assumed that a visual sensor suite (defined above) is rigidly attached to the end of an articulated (serial chain) pan-tilt or "neck" mechanism. The base of the neck is rigidly attached to the same base or torso as the manipulator. The robot shown in Figure 1.1, which is the system used in the experiments of Section 3.9, is representative of the assumptions just described. The geometry of this mechanism is shown in Figure 3.1. We further assume three commonly available sensors: namely a stereo camera, a lidar based (or RGD-D[1]) camera, which generates a dense set of range points, and a higher resolution monocular camera.

Let $G_{BP}(\mathbf{\Theta})$ denote the forward kinematic model relating the palm frame $\mathcal{F}_P$ to the base frame, $\mathcal{F}_B$. That is, G_{BP} defines the net spatial displacement between $\mathcal{F}_P$ and $\mathcal{F}_B$ as a function of the manipulator joint angles, Θ. Similarly, let $G_{BV}(\mathbf{\Phi})$ denote the neck mechanism's forward kinematic model, which relates the visual sensor frame, $\mathcal{F}_V$, to frame $\mathcal{F}_B$. The method developed in this thesis assumes that the measured joint angles may be in error due to low precision joint sensors. Similarly, flexibility in the joints (such has arises when using a tendon drive mechanisms). The model $G_{BP}(\mathbf{\Theta})$ represents the idealized model of the mechanism. Similarly, small flexibilities in the

[1]RGB-D cameras provide synchronized depth (D) and color (RGB) with each pixel. The Kinect^{TM}is an example of such a sensor.

links may contribute to mechanism modeling errors, which are approximated at small joint errors.

3.2 Object and Grasping Model

The geometry of the object to be manipulated, O, is assumed to be a priori known. Although this formulation allows for objects's geometry to be learnt adaptively, such work is not addressed in this section. The surface of object O is modeled using a standard polygonal mesh $\mathcal{M}$ consisting of a fixed number of faces $\{F_i\}$ $(i = 1, \ldots, n_F)$, edges, $\{E_j\}$ $(j = 1, \ldots, n_E)$, and vertices $\{\mathcal{V}_k\}$ $(k = 1, \ldots, n_\mathcal{V})$. A reference frame, $\mathcal{F}_O$, is rigidly attached to the object model (see Figure 3.2). Each link of the manipulator is endowed with its own link reference frame, $\mathcal{F}_l$ $(l = 1 \ldots n_\mathcal{L})$. The rigid link is similarly modeled as a separate polygonal mesh.

3.2.1 Contact Modes

I assume a prehensile grasping model where each finger of the robotic hand contacts the object surface at a single point (on the finger tip) on a unique object face. More than one finger is allowed to contact a single face. If the hand has $n_\mathbb{F}$ fingers, then there are $n_\mathbb{F} \cdot n_F$ possible pairings of fingers and object faces. Each pairing is termed a *contact mode*, where $C_{\boldsymbol{F},f}$ denotes the contact

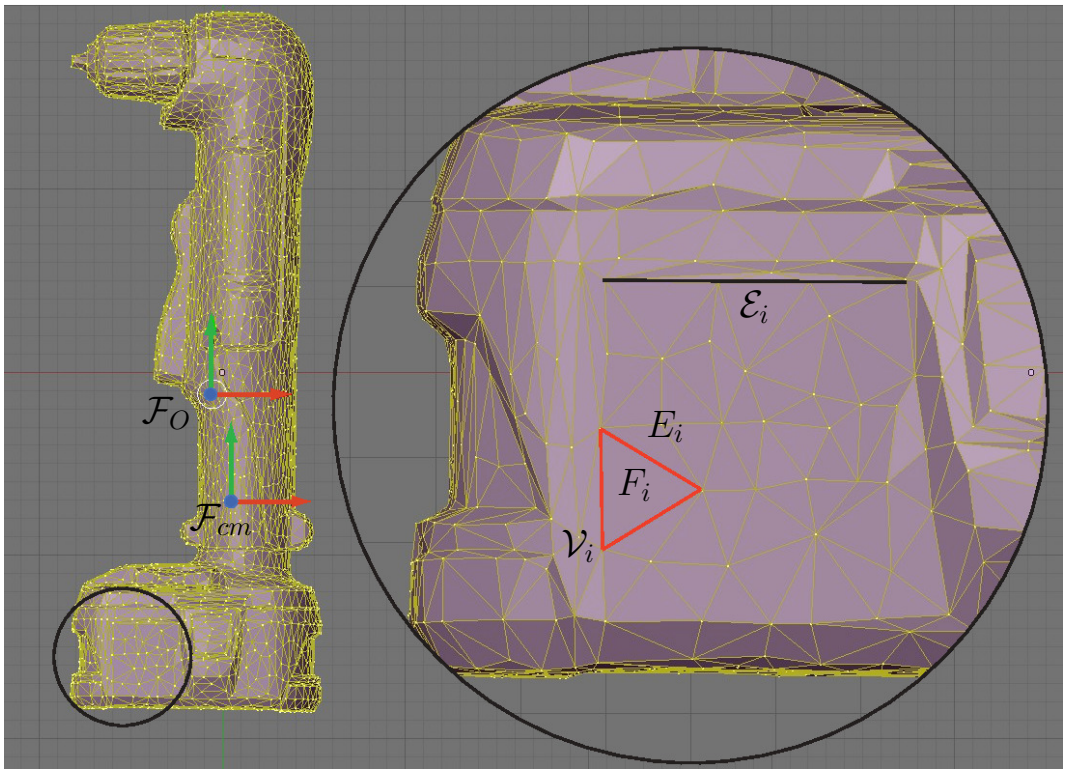


Figure 3.2: Polygonal object mesh model

mode which pairs the F^{th} finger with the f^{th} object face. If the F^{th} finger does not contact any object surface, it is assigned to the *null contact mode*, $C_{0,F}$. The contact modes define a discrete set of binary states (e.g., $C_{f,F} = 0$ if the F^{th} finger does not contact the f^{th} face, and $C_{f,F} = 1$ if the finger does contact the face) which must be estimated along with the continuous model states. Correct estimation of these contact modes constrains and improves the object's pose estimate.

3.3 State Representation

The aim is to accurately estimate and track the pose of the object, the state of the finger-object contact, and the state of the robot arm. As shown in Figure 3.1, for single-arm manipulation, the continuous estimator state, X, is chosen to be:

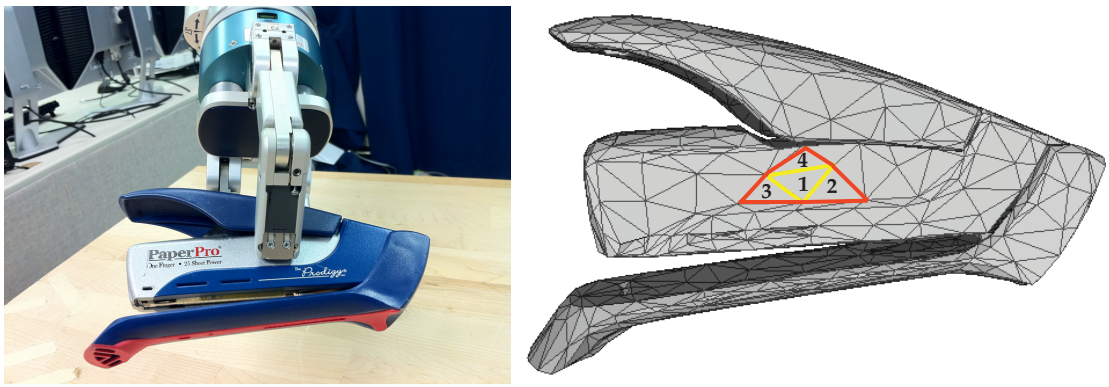
$$X = \{X_O, X_R\} = \{G_{PO}, \boldsymbol{\theta}, \boldsymbol{\phi}\}, \tag{3.1}$$

where X_O are the object-related states and X_R are the robot states:

- G_{PO} is the pose of the object relative to the manipulator's palm frame. This state directly measures the movement of O within the hand when the object is grasped.
- $\boldsymbol{\theta}$ is the set of "errors" in the arm angles.
- $\boldsymbol{\phi}$ is the set of "errors" in the neck angles.

Since I use a 3-parameter parameterization of $SO(3)$ (axis-angle representation), the estimator state vector has dimension $L = 17$ (13 without including the neck angles) for the robot shown in Figure 1.1.

When contact modes are to be estimated using sensory data, the position of the finger contacts on the object surface, ${}^{f}\boldsymbol{\beta}$, must also be estimated.



(a) Barrett hand grasping stapler (b) Nearby contact modes of finger grasping stapler

Figure 3.3: Polygonal mesh and highlighted contact modes of prehensile grasping of a stapler

- ${}^f\boldsymbol{\beta}$ are the parametrized positions of the finger contacts on the object face, F_f (The coordinate system is shown in Figure 3.17).

- The discrete state to be estimated consist of the contact modes, $\{C_{f,F}\}$ ($f = 0, \ldots, n_F$; $F = 1, \ldots, n_F$).

Estimation of the continuous states requires the specification of both dynamic and measurement models. The discrete state estimates are managed through a *static multiple model* (SMM) framework, which is described in section 3.6.

3.4 Predictive Dynamic Model

For the experiments described in Section 3.9, the following straight forward dynamic model, similar to a random walk model, is used when the object is grasped inside robot's hand:

$$\begin{bmatrix} G_{PO} \\ \boldsymbol{\theta} \\ \boldsymbol{\phi} \end{bmatrix}_{k+1} = \begin{bmatrix} I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{f}_m(\Delta\boldsymbol{\Theta}_k) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_n(\Delta\boldsymbol{\Phi}_k) \end{bmatrix} \begin{bmatrix} G_{PO} \\ \boldsymbol{\theta} \\ \boldsymbol{\phi} \end{bmatrix}_k + \mathbf{w}_k \,, \tag{3.2}$$

where $\Delta\boldsymbol{\Theta}_k$ and $\Delta\boldsymbol{\Phi}_k$ are the differences in the commanded manipulator and neck angles from time step $(k-1)$ to k and $\mathbf{f}_m$ and $\mathbf{f}_n$ are matrix functions that evaluates to $\mathbf{0} < \mathbf{f}_m(\Delta\boldsymbol{\Theta}_k) < I$ for $\Delta\boldsymbol{\Theta}_k > \mathbf{0}$ and $\mathbf{f}_m(\Delta\boldsymbol{\Theta}_k) = I$ otherwise. These matrix functions produce stable and attracting matrices, such that the eigenvalues of the matrices are less than 1, and are chosen by the user. These matrix functions are used as a forgetting factor, since the arm and neck states are highly dependent upon the current configuration of the object, arm, and neck. The matrix functions are used and activated when the $\Delta\boldsymbol{\Theta}_k$ and $\Delta\boldsymbol{\Phi}_k$ are above some threshold, which typically indicate dynamic movements to another configuration. Throughout this thesis, when the object is not grasped, the governing dynamics will depend upon the application.

3.5 Measurement Models

A key contribution of this thesis is the development of the measurement models which allow various visual cues to be integrated into an estimator. Models are developed for manipulator arm tracking as well as object tracking.

3.5.1 Visual Manipulator Tracking

In the following sections, three methods for visual manipulator tracking will be developed. Manipulator tracking entails the localization of the manipulator over time using visual sensors.

- Appearance-based tracking involves locating a template, or "patch", that is distinct on the arm and hand in stereo images, and then triangulating the template to obtain a 3D location.

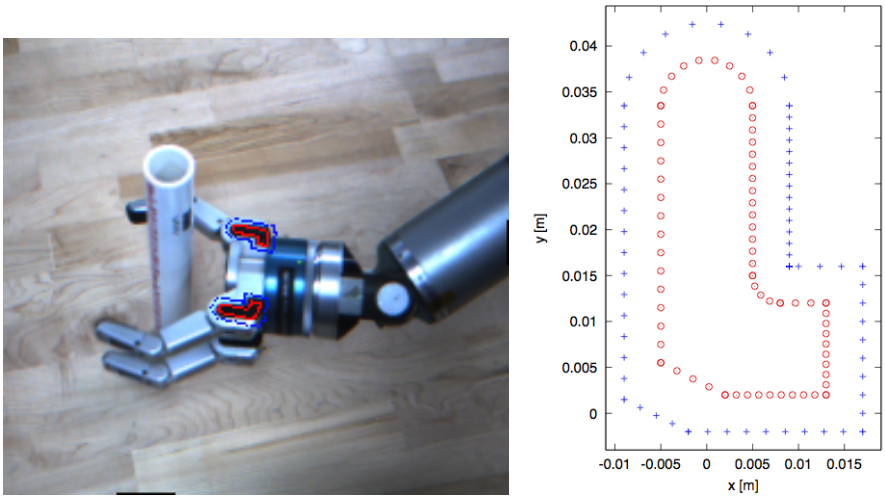
- Shape-based tracking is form of articulated iterated closest point that matches the shape of the sensed point cloud with the mesh model of the manipulator.

- Silhouette tracking involves rendering a 3D model of the manipulator into a virtual camera. The contour can easily be extracted from this rendering. The contour is then iteratively matched to an edge image of the original image.

Subsequently, these three methods will be combined into an estimator.

3.5.1.1 Appearance-Based

Many manipulators, such as the Barrett WAM arm, have distinct and recognizable visual features which can be used for tracking and localization. Alternatively, fiducial markers can be placed on arm/hand surfaces. For this appearance-based component of the tracking system, I adopt a technique of [68] to detect and then localize fiducial features on the manipulator body using a stereo camera. A *feature patch template* in a template-fixed reference frame $\mathcal{F}_T$, is parametrized by points $t_i = [t_{i,x}, t_{i,y}]$, (e.g., see Figure 3.4b), which includes the center of the template t_c. The template is first projected onto the left and right camera image planes using the nominal measured arm angles, $\boldsymbol{\Theta}$, and neck angles $\boldsymbol{\Phi}$.



(a) Barrett hand with matching templates

(b) Inner and outer templates

Figure 3.4: Appearance-based arm tracking template method

In detail, the projected template points p_i in the camera image (e.g., see Figure 3.4a) are com-

puted with the following projection:

$$p_i = \mathcal{P}_{VB}(\mathbf{\Phi}) G_{BT}(\mathbf{\Theta}, \mathbf{\Phi}) \begin{bmatrix} t_x \\ t_y \\ 0 \\ 1 \end{bmatrix}, \tag{3.3}$$

where, $\mathcal{P}_{CB}$ is the projection matrix which takes base frame ($\mathcal{F}_B$) coordinates into the visual camera frame ($\mathcal{F}_V$) using the neck joints $\mathbf{\Phi}$, and G_{BT} is the kinematic transformation of the template points in the template frame ($\mathcal{F}_T$) into the body frame ($\mathcal{F}_B$) using the nominal manipulator joint angles $\mathbf{\Theta}$ and the nominal neck joint angles $\mathbf{\Phi}$. The camera projection is computed using the CAHVOR model and method found in [63] and reviewed in Section 2.3. The projected template point locations, $\{p_i\}$, are then correlated across a square window (w) defined in pixels, in the image. The *correlation score* $\Delta\hat{p}$, is computed as the summation of directional derivatives (perpendicular to the edges of the chosen template) similarly as in [68] and the displacement, Δp, which maximizes the score represents the best match:

$$\Delta\hat{p} = \arg\max_{\Delta p} \sum_i \left[I\left(p_i^{\text{outer}} + \Delta p\right) - I\left(p_i^{\text{inner}} + \Delta p\right) \right]^2 \qquad \forall \Delta p \in \left[\frac{-w}{2}, \frac{w}{2} \right], \tag{3.4}$$

where p_i^{inner} and p_i^{outer} are the inner and outer template points used to query pixel values in the image. An example of these templates are shown in Figure 3.4b.

An exhaustive search is done over the window w and the image location of the highest correlation score is chosen. A disparity, as defined as the difference in columns of a matching feature in two images, is obtained by computing the best match in both the left and right images of a stereo camera. The disparity is then used to triangulate the 3D location of the template. The measured palm location $z^{\text{appearance}}$ is reported in the visual frame $\mathcal{F}_V$. Therefore, the measurement equation for this type of measurement is:

$$Z^{\text{appearance}} = G_{VP} + \zeta^{\text{appearance}}, \tag{3.5}$$

where G_{VP} is the homegeneous transform of the palm frame $\mathcal{F}_P$ as seen in the visual frame $\mathcal{F}_V$ and $\zeta^{\text{appearance}}$ is zero mean Gaussian noise, whose variance is scaled to match the uncertainty in the template matching process. Figure 3.4a shows the projected templates at their best matching location for the example of a Barrett$^{\text{TM}}$ hand knuckle features on the Barrett$^{\text{TM}}$ arm.

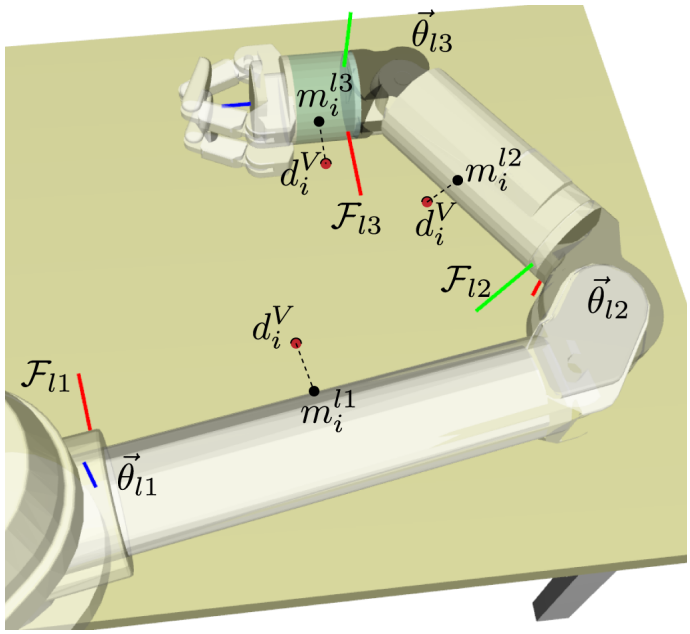


Figure 3.5: Articulated iterated closest point (aicp) diagram with key variables and frames

3.5.1.2 Shape-Based Measurements

Manipulator tracking via shape can utilize another common sensor available on many current robots. RGB-D sensors found in the Kinect$^{\text{TM}}$ sensor, in addition to flash lidar sensors (e.g., Swiss Ranger$^{\text{TM}}$), provide 3D coordinates of points on the manipulator mechanism surface. Our method to use these 3D measurements is a form of *articulated icp* (aicp) which others have used successfully. However, Pellegrini [69] uses a traditional ICP approach [60] on each link separately with selection policies and Krainin [42] uses a Kalman filter along with the Levenberg-Marquandt optimizer, that does not make explicit use of the manipulator's Jacobian.

In this aicp approach, estimates of the joint variables are obtained by minimizing the error between the model points of the manipulator (e.g. created by randomly sampling link mesh models offline) and the S sensed data points from the 3D sensor:

$$E = \sum_{s=1}^{S} (d_s^{l(s)} - m_s^{l(s)})^T (d_s^{l(s)} - m_s^{l(s)}) \tag{3.6}$$

$$= \sum_{s=1}^{N} [G_{Bl}(\mathbf{\Theta})(d_s^l - m_s^l)]^T [G_{Bl}(\mathbf{\Theta})(d_s^l - m_s^{l(s)})] \tag{3.7}$$

$$= \sum_{s=1}^{N} [G_{BC}(\mathbf{\Phi}) d_s^C - G_{Bl}(\mathbf{\Theta}) m_s^{l(s)}]^T [G_{BC}(\mathbf{\Phi}) d_s^C - G_{Bl}(\mathbf{\Theta}) m_s^{l(s)}], \tag{3.8}$$

where $d_s^{l(s)}$ and $m_s^{l(s)}$, respectively, are the 3D coordinates of the s^{th} data point and it's corresponding model point, as described in the l^{th} link frame. The data association variable $l(s)$ specifies how the

l^{th} link frame is associated with the s^{th} point. The vector d_s^C describes the 3D coordinates of the s^{th} data point in the camera reference frame $\mathcal{F}_C$. G_{BC} and G_{Bl} are the homogeneous transforms from the robot base frame to the camera frame and from the robot base frame to l^{th} link frame, respectively. Note that the equality between Equations (3.6) and (3.7) arises from the distance preserving property of the mapping G_{Bl}.

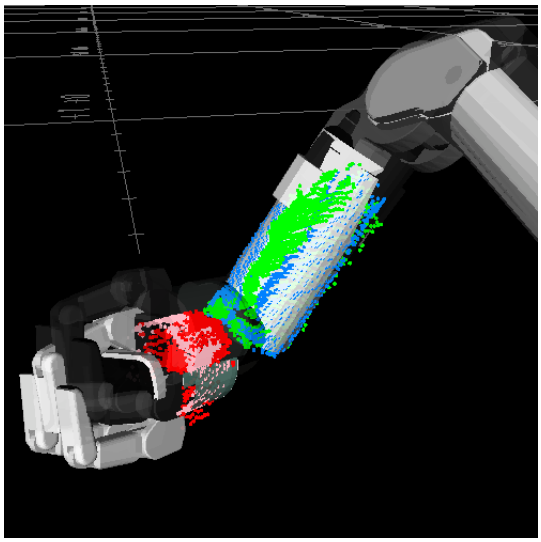


Figure 3.6: Point cloud (green and red) and matching model points (blue and pink). Optimized model (white) and original model (black).

The best joint angle estimates are derived by minimizing the collective error in Equation 3.8:

$$\boldsymbol{\Theta}^* = \arg\min_{\boldsymbol{\Theta}} E(\boldsymbol{\Theta})\,. \tag{3.9}$$

The experiments presented in this thesis use a gradient descent method to minimize the error function E. The iterative gradient descent procedure is as follows:

$$\begin{aligned}\hat{\boldsymbol{\Theta}}_{j+1} &= \hat{\boldsymbol{\Theta}}_j - \gamma \frac{\partial E(\hat{\boldsymbol{\Theta}}_j)}{\partial \hat{\boldsymbol{\Theta}}} \\ &= \hat{\boldsymbol{\Theta}}_j + 2\gamma \sum_{s=1}^{S} [\frac{\partial G_{Bl}(\hat{\boldsymbol{\Theta}}) m_s^{l(s)}}{\partial \hat{\boldsymbol{\Theta}}}]^T \cdot \\ &\quad [G_{BC} d_s^C - G_{Bl}(\hat{\boldsymbol{\Theta}}) m_s^{l(s)}]\,,\end{aligned} \tag{3.10}$$

where $\hat{\boldsymbol{\Theta}}_{j+1}$ is the estimate of the "correct" joint angles at iteration $j+1$, and γ is a positive constant, which controls the convergence process. The derivative may be further analyzed:

$$\frac{\partial G_{Bl}(\hat{\boldsymbol{\Theta}}) m_s^{l(s)}}{\partial \hat{\boldsymbol{\Theta}}} = \frac{\partial p_{Bl}(\hat{\boldsymbol{\Theta}})}{\partial \hat{\boldsymbol{\Theta}}} + \frac{\partial R_{Bl}(\hat{\boldsymbol{\Theta}})}{\partial \hat{\boldsymbol{\Theta}}} m_s^{l(s)}\,, \tag{3.11}$$

where p_{Bl} and R_{Bl} are the positional and rotational components of the homogenous transform, G_{Bl} and their derivatives can be found using the hybrid Jacobian of the manipulator mechanism. The matching model points can be found in two ways. The first method is the traditional method

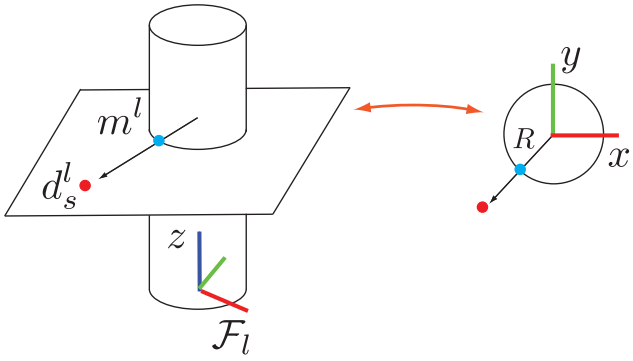


Figure 3.7: Cylindrical articulated iterative closest point matching points. The matching point m^l (blue) on the cylindrical surface of link l with radius R to the data point d_s^l (red).

of finding nearest point in the model or using a "nearest" metric to the model. This approach is unacceptably slow unless one uses a kD-tree or other approximate nearest neighbor algorithm. Our method exploits the primitive shapes of the manipulator links. In this case of a Barrett$^{\text{TM}}$ WAM arm, there are 3 cylinders in the arm mechanism profile. There is an analytic expression for the nearest point on a cylindrical surface. Without loss of generality, assuming that the z-axis of the link frame is collinear with the cylinder axis, the nearest model point will have the same z coordinate of the data point to which it will be matched. Mathematically, this can be written as $m_{s,z}^{l(s)} = d_{s,z}^{l(s)}$, which specifies the z component of the model and data point described in the l^{th} link frame, respectively. The matching point will have the following x and y coordinates:

$$\begin{bmatrix} m_{s,x}^{l(s)} \\ m_{s,y}^{l(s)} \end{bmatrix} = \frac{R}{\sqrt{(d_{s,x}^{l(s)})^2 + (d_{s,y}^{l(s)})^2}} \begin{bmatrix} d_{s,x}^{l(s)} \\ d_{s,y}^{l(s)} \end{bmatrix} . \tag{3.12}$$

A diagram of the matching point coordinates on the cylindrical surface is shown in Figure 3.7.

The iterative procedure in Equation 3.10 is terminated when the difference in joint angle updates reaches a tolerance set by the user. The shape measurement model is simply the direct observation of the state, the arm joint error variables, $\boldsymbol{\theta}$. The measurement is found by taking the difference of the joint variables from Equation 3.9 at the end of the gradient descent, $\boldsymbol{\Theta}^*$ and the current nominal joint angles, Θ, to form the state variable $\boldsymbol{\theta}$:

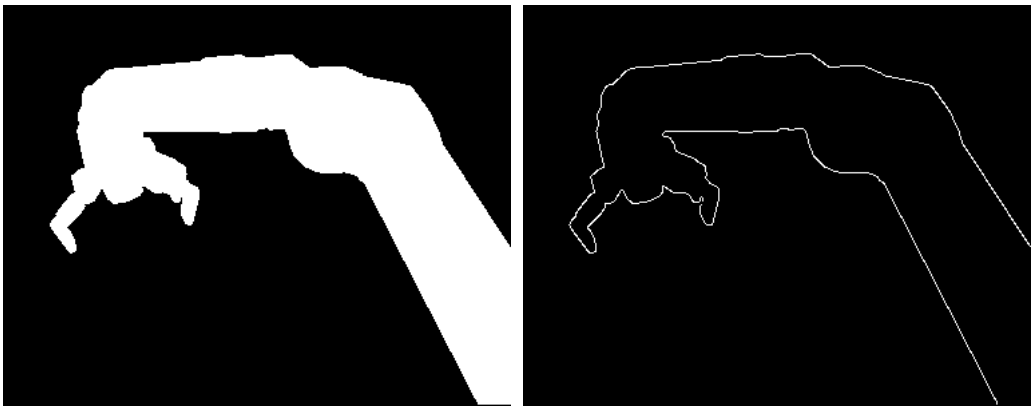
$$Z^{\text{shape}} = \boldsymbol{\theta} + \zeta^{\text{shape}} \tag{3.13}$$

$$= \boldsymbol{\theta}^* - \boldsymbol{\theta} + \zeta^{\text{shape}} , \tag{3.14}$$

where ζ^{shape} is zero mean Gaussian noise on the shape measurements.

Figure 3.6 shows the optimized match of the manipulator to a set of point cloud data points, where the data points are green and red, and the model points are blue and pink. The original manipulator configuration is shown in translucent black and the optimized configuration is shown in grey.

3.5.1.3 Silhouette-Based Visual Measurements



(a) Manipulator model rendered in OpenGL$^{\text{TM}}$ (b) Manipulator model edge image

Figure 3.8: OpenGL$^{\text{TM}}$ rendering of manipulator

The silhouette of an object or body is projected on the current viewing plane with the outline of the object and the interior being featureless. Silhouettes have been used extensively for tracking purposes, from object tracking to people tracking. Using a silhouette for manipulator tracking relies on fast rendering of the manipulator's geometric model. Using OpenGL$^{\text{TM}}$, a virtual camera is created using camera parameters from the calibrated CAHVOR model. Specifically, the CAHVOR parameters are converted into a more standard camera model such as the Tsai model [62]. Parameters such as focal length and center image point are then used to construct a viewing frustum, which is a region of space in the modeled world that will appear on the image plane or screen. Using the current nominal joint angle measurements of the manipulator and a geometric model of the manipulator, the arm and hand mechanism is rendered into the virtual camera. Using just a depth buffer, the rendered depth image in OpenGL$^{\text{TM}}$ is simply a binary image and the extraction of a silhouette is trivial using a simple edge-based detector. Figure 3.8 shows both the rendered manipulator and the subsequent edge image.

The idea is to minimize a distance metric between the rendered image silhouette and that of the silhouette in the actual image. Since the rendered model images are binary, a fast distance transform [70] can be used to quickly and efficiently compute an approximate distance metric. The transform computation is based on the convoluting of the edge image (EI) of the model with a

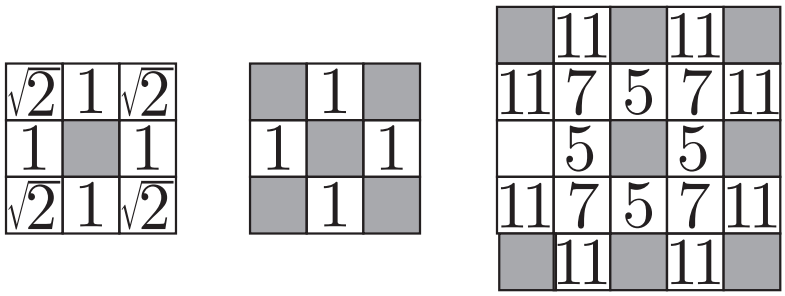
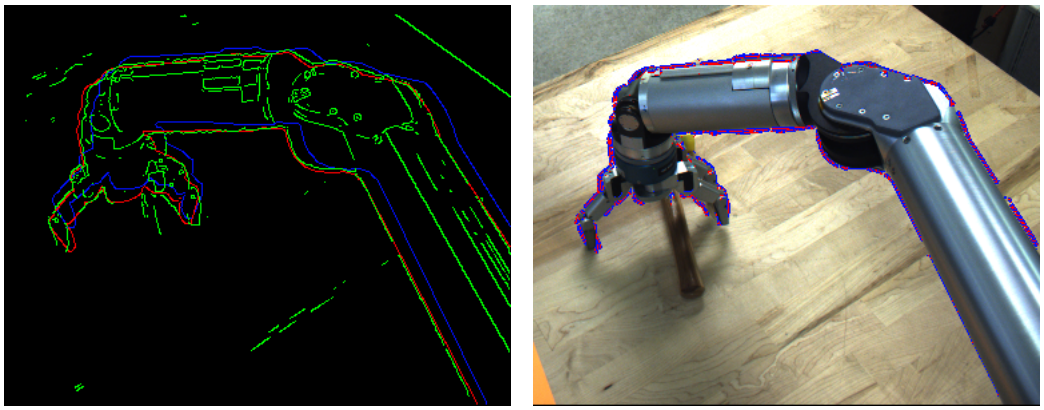


Figure 3.9: Distance masks for the distance transform. Left: Euclidean distance mask. Middle: manhattan distance mask. Right: chamfer 5-7-11 distance mask

distance mask. The distance mask is square patch that is often 3×3 or 5×5 and represent the various distance metrics as shown in Figure 3.9. The resulting convolution is the distance image (D).

Since an analytical gradient of this distance metric as a function of joint angles is difficult to calculate, a numerical difference-based optimizer is chosen to compute the manipulator's joint angles $\boldsymbol{\Theta}$, which best match the rendered silhouette with the visual silhouette. Specifically the Nelder Mead simplex algorithm [71] is used. The technique is a heuristic search method that converges to non-stationary points. It is based on evaluating a function at vertices of a simplex which iteratively shrinks until better points are found which satisfy a tolerance set by the user (tol). Algorithm 1 summarizes the computation of the measurement. The measurement model derived from this procedure is the direct observation of the state, the arm joint error angles, $\boldsymbol{\theta}$. The measurement is found by taking the difference of the joint angles at the end of the optimization, $\boldsymbol{\Theta}^*$ and the current



(a) Silhouettes of optimized model (red), initial model using nominal measured joint angles (blue), original edge image (green)

(b) Original image with matched silhouette.

Figure 3.10: Matching of the silhouette of the manipulator in the image

Algorithm 1 Silhouette-based Manipulator Tracking

Require: Image (I) and Joint Angles ($\mathbf{\Theta}$)
  Compute edge image (EI) on I
  Compute distance transform (D) of EI
  while $(\hat{\mathbf{\Theta}}_{j+1} - \hat{\mathbf{\Theta}}_j) > tol$ **do**
    $M = render\_model(\hat{\mathbf{\Theta}}_j)$
    $EM = compute\_edge\_image(M)$
    $Sum = D * EM$
    $\hat{\mathbf{\Theta}}_{j+1} = nelder\_mead\_simplex(Sum, \hat{\mathbf{\Theta}}_j)$
  end while
  return $\mathbf{\Theta}^* = \hat{\mathbf{\Theta}}_{j+1}$

nominal joint angles $\mathbf{\Theta}$ to form the state variable $\boldsymbol{\theta}$:

$$Z^{\text{silhouette}} = \boldsymbol{\theta} + \zeta^{\text{silhouette}}, \tag{3.15}$$

where $\zeta^{\text{silhouette}}$ is zero mean Gaussian noise on the silhouette measurement.

Figure 3.10a shows the initial contour using the nominal joint values, $\mathbf{\Theta}$ in blue, the optimized silhouette using Algorithm 1 shown in red and the edge segments shown in green. Figure 3.10b shows the optimized and matched silhouette overlayed on the original image.

3.5.2 Visual Object Tracking

It is valuable, and in some cases necessary, to simultaneously track both arm and object states particularly during the process of grasping. This section describes the analogous visual techniques I use for object tracking.

3.5.2.1 Feature-Based Object Tracking Measurements

For the purpose of object detection, localization, and ultimately tracking, the rigid body models described above are augmented by a set of visual features that can be observed by the stereo vision system. Visual appearance features such as SIFT features [72] or SURF features [73] are augmented by their 3D location in the camera frame so as to estimate the object's 6DOF pose. Each feature takes the form:

$$D_i = \{\mathbf{d}_i \, \tilde{\mathbf{d}}_i\}, \tag{3.16}$$

where $\mathbf{d}_i = [d_x \, d_y \, d_z]_i$ is the 3-D location of the feature and each feature $\tilde{\mathbf{d}}_i \in \mathbb{R}^{128}$ in the case of the 128-dimensional SIFT feature descriptor or $\tilde{\mathbf{d}}_i \in \mathbb{R}^{64}$ in the case of the 64-dimensional SURF feature descriptor. This descriptor captures color, texture, and orientation of the object's appearance around the feature point. The object model and features are learnt during a training phase where SIFT of SURF features of the object are selected from multiple camera viewpoints. The 3D position of these features is obtained using sparse stereo. A database, $\mathbf{D}$, is built of these 3D features (Cartesian

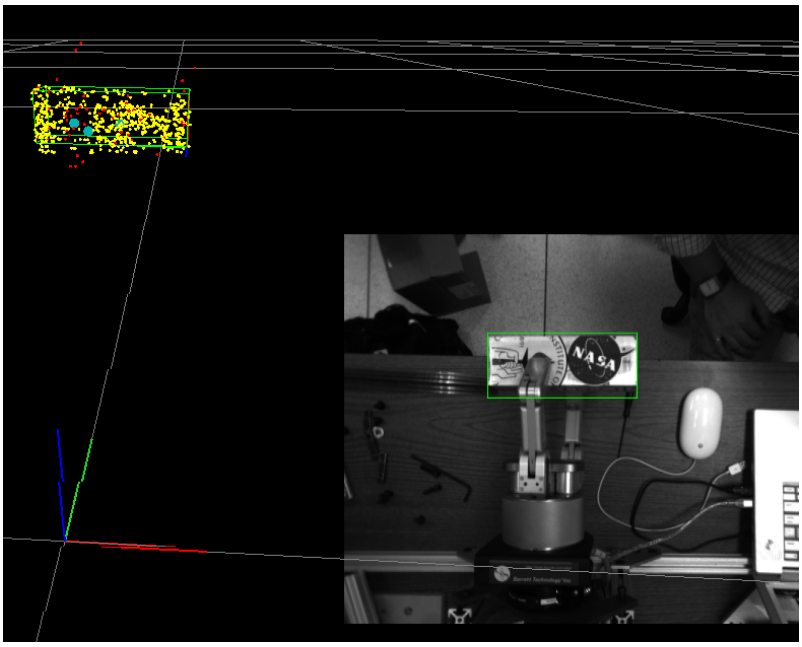


Figure 3.11: Example of learnt visual features (yellow) and the tracked object (in green box)

location and feature descriptor):

$$\mathbf{D} = [\{\mathbf{d}_0\, \tilde{\mathbf{d}}_0\}\, \{\mathbf{d}_1\, \tilde{\mathbf{d}}_1\}\, \ldots\, \{\mathbf{d}_N\, \tilde{\mathbf{d}}_N\}\,]\,. \tag{3.17}$$

To visually learn the object, the training phase may be carried out using a fixed camera and a turntable for rotating the object to obtain multiple viewpoints.

The vision measurements are the analagous 3D features $\mathbf{Z}_k = [\{\mathbf{z}_0\, \tilde{\mathbf{d}}_0\}\, \{\mathbf{z}_1\, \tilde{\mathbf{d}}_1\}\, \ldots\, \{\mathbf{z}_N\, \tilde{\mathbf{d}}_{n_k}\}]$ where $\mathbf{z}_i = [z_x\, z_y\, z_z]$ in the visual camera frame, $\mathcal{F}_V$. Lowe's [74] Best-Bin-First scheme is used to match the observed SIFT features with those in the database providing the correspondences $\mathbf{J} \in \mathbb{Z}^+$ so that $\mathbf{J}(i) = j$ links the database features $\mathbf{d}_j$ with observation $\mathbf{z}_i$.

Therefore, the measurement model consists of matching the position components of both the database and observed 3D SIFT features. Assuming at time t_k there are m_k matches from the Best-Bin-First algorithm, the measurement process can be modeled as:

$$\begin{aligned} \begin{bmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_{m_k} \end{bmatrix} &= \begin{bmatrix} G_{VO}(X_k)\, \mathbf{d}_{\mathbf{J}(0)} \\ G_{VO}(X_k)\, \mathbf{d}_{\mathbf{J}(1)} \\ \vdots \\ G_{VO}(X_k)\, \mathbf{d}_{\mathbf{J}(m_k)} \end{bmatrix} + \zeta_v \\ &\equiv \mathbf{H}_v(\boldsymbol{\phi},\, \boldsymbol{\theta},\, G_{PO}\,, \mathbf{J}) + \zeta_v\,, \end{aligned} \tag{3.18}$$

where ζ_v is a zero mean Gaussian noise associated with the stereo measurements.

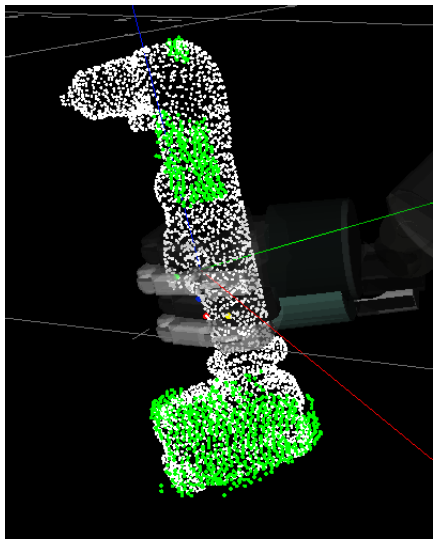


Figure 3.12: OpenGL™rendering of sensed point cloud (green points) from SwissRanger™camera matching to mesh model cloud (white).

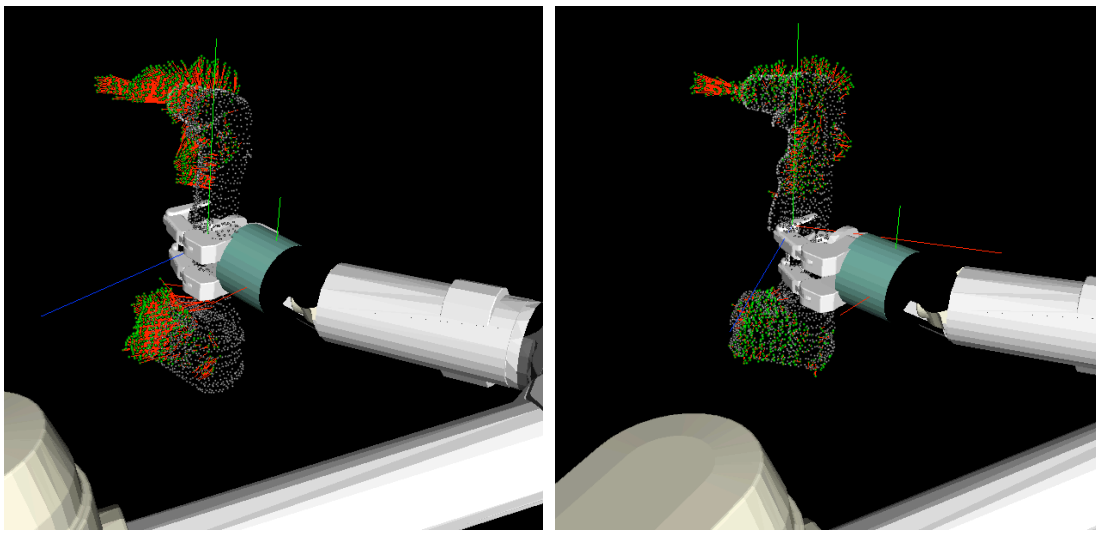
3.5.2.2 Shape-Based Object Visual Measurements

For object tracking I use a point cloud association procedure similar to the iterated closest point algorithm. Experiements have found that due to the large range in feature scale often experienced in manipulation tasks, classical features (e.g., SIFT and SURF) [41] do not perform effectively — descriptor matching often produces inconsistent matches, resulting in poor object tracking. In addition, lighting conditions affect performance especially when lights conditions are different from when features were first learnt. Therefore, I propose an approach to use more robust methods. Using point clouds for matching, our predicted measurements are simply:

$$Z_i^{\text{shape}} = x_{BO} + R_{BO} v_i^O + \zeta^{\text{shape}} \tag{3.19}$$

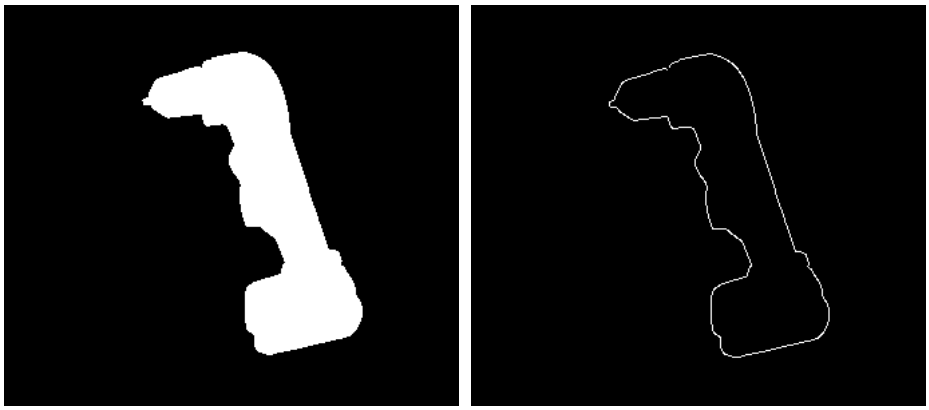
$$= G_{BP} G_{PO} \begin{bmatrix} v_i^O \\ 1 \end{bmatrix} + \zeta^{\text{shape}}, \tag{3.20}$$

which relates the matched point v_i^O in the object's mesh, $\mathcal{M}$, to the sensed data point z_i^{shape}, obtained from from a stereo vision system and flash lidar, and ζ^{shape} is a zero mean Gaussian noise on the shape measurement. The variance is determined by empirically measuring the noise of the camera used. Figure 3.12 shows the model (white points) at the current estimated position and the filtered stereo point cloud points (green points).



(a) Matchings points — large measurement innovation (b) Matching points — small measurement innovation

Figure 3.13: Depiction of the matching process used during shape matching. The red lines connect 3D shape data points (green dots) and their associated mesh model points (white dots).



(a) Object model rendered in OpenGL (b) Object model edge image

Figure 3.14: OpenGL object rendering for a specific sigma point.

3.5.2.3 Silhouette-Based Object Measurements

This silhouette approach used for object localization is similar to Algorithm 1, but the object model is now rendered at the various sigma points (Section 2.4.5 reviews the sigma point filter) instead at various values as determined by the Nelder Mead algorithm. The contour/edge points closest to the zero$^{\text{th}}$ sigma point in the actual image is approximated as the measurement in the UKF. As the actual image has many edge responses, selecting the edge associated with the silhouette can be a difficult process, which is not addressed in this thesis. One way to improve the selection of these edge measurements is to use optical flow, image differencing, and/or background subtraction. The sigma-point matching method produces reasonable measurements despite other edge responses, as shown in Figure 3.15b.

The measurement model of the object silhouette matching process consists of the contour image points:

$$Z^{\text{silhouette}} = \mathcal{C}(G_{VO}, O) + \zeta^{\text{silhouette}} \tag{3.21}$$

$$= \mathcal{C}(G_{VB}(\boldsymbol{\phi})G_{BP}(\boldsymbol{\theta})G_{PO}, O) + \zeta^{\text{silhouette}}\,, \tag{3.22}$$

where $\mathcal{C}$ is the computation of the contour points and G_{VO} is the transform relating the object's reference frame $\mathcal{F}_O$ in the visual reference frame $\mathcal{F}_V$ and $\zeta^{\text{silhouette}}$ is zero mean Gaussian noise on the silhouette measurement. $\mathcal{C}$ is computed by first rendering a model of the object, then projecting into a virtual camera frame and then computing an edge image. The pixels of this edge image constitute the contour points.

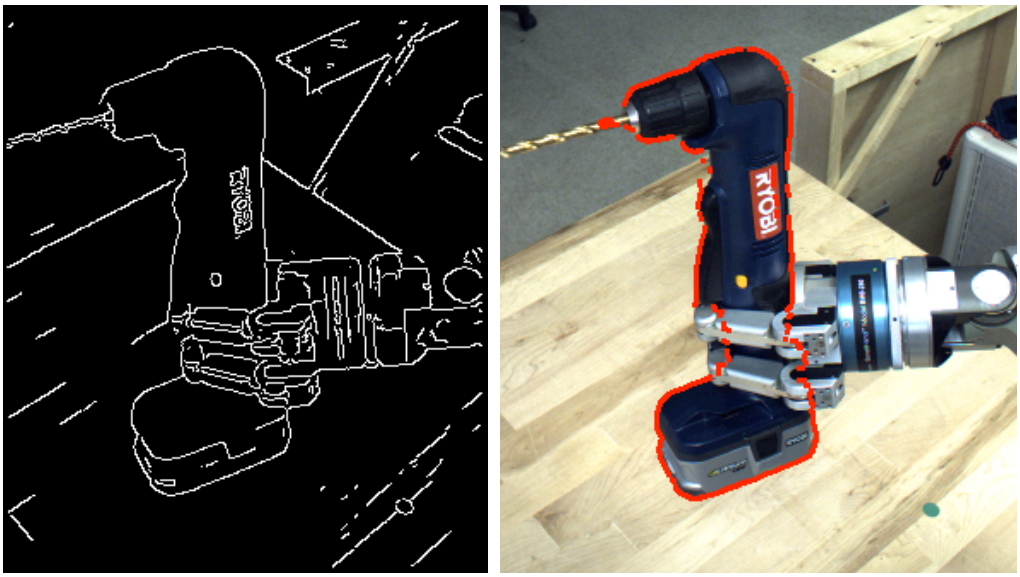
3.5.3 Kinesthetic Measurements

A wrist-mounted force sensor and tactile pads on the multi-fingered hand surfaces may be available. If so, this section describes the measurement models that can be associated to these sensors.

3.5.3.1 Wrist-Mounted Force-Torque Sensor Measurements

A wrist-mounted six-axis force-torque sensor will be sensitive to the gravitational forces acting on the object and the multi-fingered hand, which is mounted on the distal side of the sensor. Note that up to the first mass moment of the object, the 3 torque measurements are the only measurements which provide information about the object location. The net wrench measured at the wrist,

$$W_w = \begin{bmatrix} \mathbf{F}_w \\ \boldsymbol{\mathcal{T}}_w \end{bmatrix}, \tag{3.23}$$



(a) Original edge image (b) Original image with extracted contour (red)

Figure 3.15: Depiction of the silhouette-based object estimation showing the canny edge image and highlighted matched contour in red.

has the following relationship with the gravitational load on the object and hand mechanism. Neglecting dynamic loads on the hand and the object, these wrenches may be decomposed into:

$$\begin{aligned} W_w &= W_w^O + W_w^H \\ &= \begin{bmatrix} \mathscr{F}_{wO} \end{bmatrix} \begin{bmatrix} R_{O\mathcal{W}}\, {}^{\mathcal{W}}\mathbf{F}_g \\ \mathbf{0} \end{bmatrix} + W_w^H \\ &= \begin{bmatrix} R_{w\mathcal{W}}\, {}^{\mathcal{W}}\mathbf{F}_g \\ \hat{p}_{wO}\, R_{O\mathcal{W}}\, {}^{\mathcal{W}}\mathbf{F}_g \end{bmatrix} + W_w^H \,, \end{aligned} \tag{3.24}$$

where W_w^O and W_w^H are the measured wrenches due to the object and hand mechanism, respectively. ${}^{\mathcal{W}}\mathbf{F}_g$ is the gravitional force acting on the object, as measured in the world reference frame, $R_{w\mathcal{W}}$ denotes the orientation of the world frame with respect to the wrist frame, $\mathbf{p}_{wO}$ is the displacement of the object frame origin with respect to the wrist frame origin, and $\hat{p}_{wO}$ is the 3×3 skew symmetric matrix such that the vector $\hat{p}_{wO}\,\mathbf{v} = \mathbf{p}_{wO} \times \mathbf{v}$. $\mathscr{F}_{wO}$ is the force transformation or adjoint matrix that transforms forces in the object frame $\mathcal{F}_O$ into the wrist frame F_w. It is defined as:

$$\mathscr{F}_{wO} = \begin{bmatrix} R_{wO} & \mathbf{0} \\ \hat{p}_{wO}\, R_{wO} & R_{wO} \end{bmatrix} \tag{3.25}$$

Let W_w^H denote the gravitational load of the hand on the force-torque sensor. Practically, the gravitational load due to the hand mechanism is incorporated via a look-up table of the resting force-torque values at steady-state temperature found for various orientations of the hand and fingers.

Note that the torque measured at the wrist sensor, $\boldsymbol{\mathcal{T}}_w$, is the only portion of the force-torque measurement that is dependent on the object pose. Therefore, the measurement equation for relating force-torque sensor measurement to the object pose takes the form:

$$\begin{aligned} \boldsymbol{\mathcal{T}}_w &= \hat{p}_{wO}\, R_{O\mathcal{W}}\, {}^{\mathcal{W}}\mathbf{F}_g + \boldsymbol{\mathcal{T}}_w^H + \zeta_{\mathcal{T}} \\ &\equiv \mathbf{H}_{\mathcal{T}}(G_{PO}, \boldsymbol{\theta}) + \boldsymbol{\mathcal{T}}_w^H + \boldsymbol{\zeta}_{\mathcal{T}} \,, \end{aligned} \tag{3.26}$$

where $\boldsymbol{\mathcal{T}}_w^H$ represents gravitational torque due to the hand mass, $\zeta_{\mathcal{T}}$ represents measurement noise.

3.5.3.2 Tactile Sensor

There are many different technologies used for implementing tactile sensors [75–80]. This section uses a simple model of tactile sensing that should apply to many different devices.

In addition to visual information, a simple tactile measurement update can be integrated if it is available. Many robot hands, such as the Barrett$^{\text{TM}}$ hand used in the experiments described below contains tactile arrays on the fingers and palm. Here, a binary tactile measurement model is used

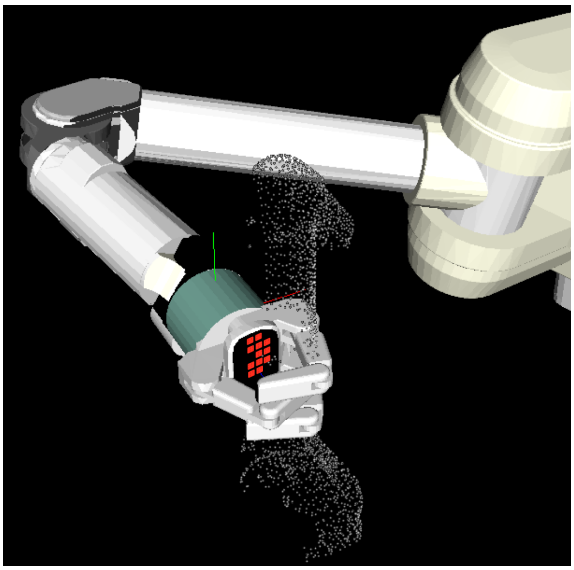


Figure 3.16: A process which matches the active tactels to the model provides valuable measurements. Activated tactels shown in red with mesh model in white.

to indicate whether or not each tactel is in contact with the object. The measurement model for the tactile sensor update is implicitly used to construct the pose measurement, meaning that the actual sensor measurement z^{tactile} is required to compute the predicted measurement Z^{tactile}. The modeling goal is to find the closest point/vertex, v^P, on the object mesh at the given state estimate, G_{PO}, to the tactel sensor position, z_i^{tactile}, in the palm ($\mathcal{F}_P$) reference frame:

$$Z_i^{\text{tactile}} = \arg\min_{v^P} \|v^P - z_i^{\text{tactile}}\| + \zeta^{\text{tactile}} \tag{3.27}$$

$$= G_{PO} \arg\min_{v^O} \|G_{PO} \begin{bmatrix} v^O \\ 1 \end{bmatrix} - z_i^{\text{tactile}}\| + \zeta^{\text{tactile}}, \tag{3.28}$$

where ζ^{tactile} is zero mean Gaussian noise on the tactel measurement. The noise variance is determined empirically by measuring noise over multiple measurements. Figure 3.16 depicts the measurement process. Similarly an active tactel can be used to infer the object surface normals. A second measurement model involves exploiting the mesh vertex normal, n^P, corresponding to the normal vector of the closest point/vertex, v^P, in the object's mesh at the given state, G_{PO}, to the tactile sensor position, z_i^{tactile}:

$$Z_i^{n,\text{tactile}} = \arg\min_{n^P} \|v^P - z_i^{\text{tactile}}\| + \zeta^{n,\text{tactile}} \tag{3.29}$$

$$= G_{PO} \arg\min_{n^O} \|G_{PO} \begin{bmatrix} v^O \\ 1 \end{bmatrix} - z_i^{\text{tactile}}\| + \zeta^{n,\text{tactile}}, \tag{3.30}$$

The actual measurement is the normal of the active tactel in the palm frame.

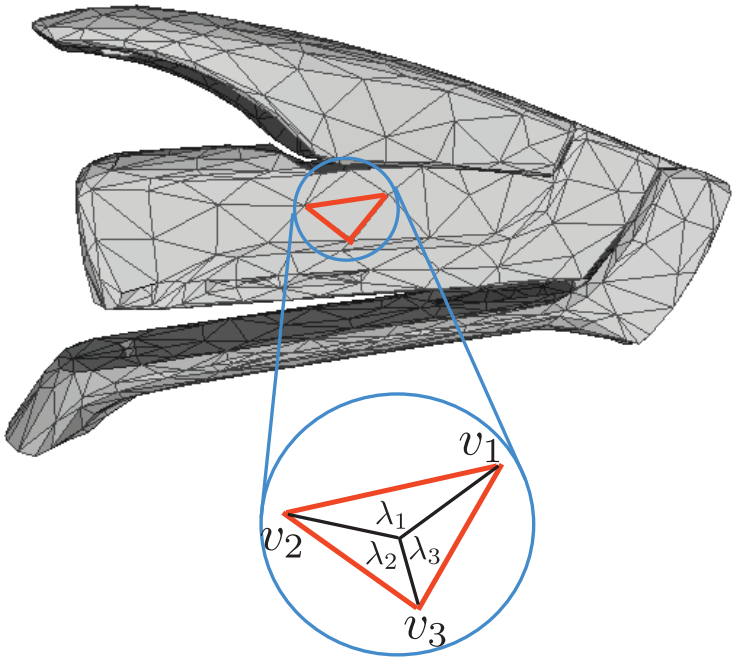


Figure 3.17: Barycentric coordinates.

3.5.3.3 Hand Mechanism and Manipulator Linkage Joint Measurements

Measurements of the state of the arm and hand linkages (e.g., rotations or displacements of joints in the hand or robot arm) are used, via the use of forward kinematic equations, to produce pseudo-measurements (since the actual measured value is the joint angle) of fingertip locations or wrist locations. I assume that the forward kinematic model is in error due to both joint sensor inaccuracy and unmodeled kinematic errors.

Measurements from the joints located in the finger linkages, $(\theta_{\mathcal{F}})$, may be used to provide a pseudo-measurement of the fingertip locations by means of forward kinematic equations.

To include the interaction between the fingers and the object, I include the finger-surface location into the measurement model. I assume a simple contact sensing process (e.g., using finger strain or active tactels) which can determine if the $\mathcal{F}^{\text{th}}$ finger is in contact with the object. This contact detection may be implemented via tactile sensor measurements, joint strain measurements, or by monitoring current drawn in the finger actuator. When the finger is in contact with a surface, I use a closure equation for the $\mathcal{F}^{\text{th}}$ finger in the wrist frame coordinate system, w, taking the form:

$$\begin{aligned} p_{w\mathcal{F}}(\theta_{\mathcal{F}}) &= p_{wO} + R_{wO}\, p_{O\mathcal{F}}(\mathcal{M}) + \zeta_{\mathcal{F}} \\ &\equiv \mathbf{H}_{\mathcal{F}}(\boldsymbol{\theta}\,, {}^{i}\boldsymbol{\beta},\, G_{PO}) + \zeta_{\mathcal{F}}\,, \end{aligned} \tag{3.31}$$

where $p_{O\mathcal{F}}(\mathcal{M})$ is the surface location of the finger $\mathcal{F}$ in the object's reference frame, and the coordinates of the contact location may be found using a surface parameterization. For example,

using barycentric coordinates as the surface parameters, ${}^f\boldsymbol{\beta} = \{\lambda_1, \lambda_2, \lambda_3\}$, and the three vertices $v_{1,2,3}$ that are defined by the contact mode $C_{f,F}$, the finger location can be parametrized as:

$$p_{OF}(\mathcal{M}) = \lambda_1 v_1 + \lambda_2 v_2 + \lambda_3 v_3 \,. \tag{3.32}$$

Barycentric coordinates are a form of homogeneous position coordinates that are quite suitable for parameterizing triangular meshes. In these triangular meshes, (see, e.g., figure 3.17), barycentric coordinates are essentially area coordinates. Each barycentric coordinate defines the areas formed by the rays joining the vertices of the triangle to a point within the triangle. Barycentric coordinates are frequently used in graphics engines since they can quickly determine whether a point is inside the triangle or not. Barycentric coordinates are subject to the constraint:

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \,. \tag{3.33}$$

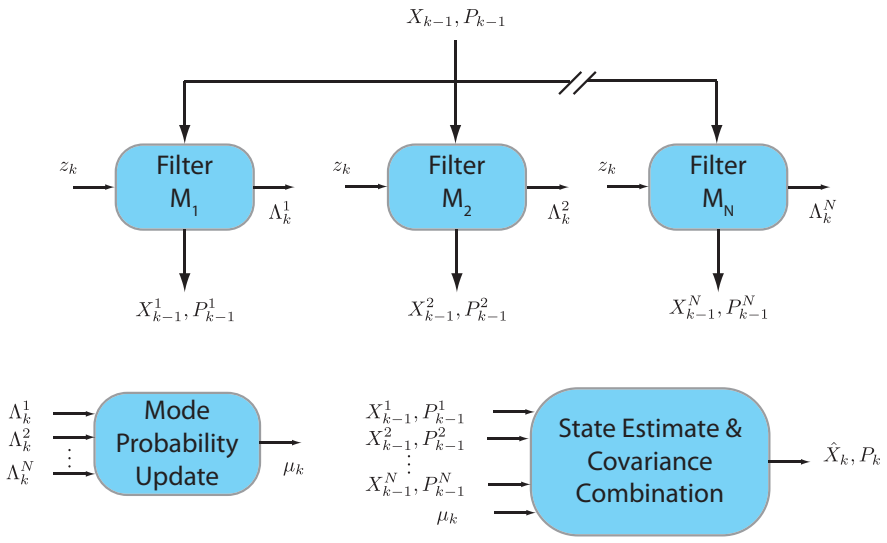


Figure 3.18: General static multiple model estimator.

3.6 Hybrid State Estimator

Since no assumptions are made regarding how the fingers contacts the object, the estimation filter must be able to estimate which surface is in contact with each finger (i.e., which contact modes are active), in addition to the continuous state variables. This combinatorial matching of finger to surface can be posed as a *hybrid* state estimation problem wherein both the continuous states and discrete contact modes must be jointly estimated. While a variety of hybrid system estimation algorithms and architectures have been proposed, the experiments in this thesis use a *static multiple model estimator* (SMM) [20]. Each possible finger-surface contact combination is associated to a model. Each model in turn has an associated measurement model to describe how each finger might contact the associated object face. The SMM estimator is based on a Bayesian framework which leverages prior probabilities of each model being correct to update and obtain posterior probabilities based on sensor measurements. The SMM estimator assumes that the system will obey only one of the possible models, such that there are no switching during the estimation process (i.e., the fingers are assumed to not change contact modes during the estimation interval). Although the particular mode that is in effect is assumed to stay fixed (static), each model can have its own dynamics, making the estimator essentially a dynamic process.

While the SMM assumes only one model to be in effect during the estimation process, the particular active contact mode is initially unknown. The prior probability on each of the possible contact modes can be selected by the user and for example, the prior might be provided by a grasp

planner. The ensuing experiments choose a uniform prior:

$$P\big(M_m|z^0\big) = \mu_0^m = \frac{1}{N}\,, \tag{3.34}$$

where z^0 is the available prior information (e.g. from grasp planners), μ_m is the m^{th} model probability and $\sum_{m=1}^{N} \mu_0^m = 1$. The m^{th} model here, specifies a set of contact mode for all fingers. Using Bayes' formula, the posterior probability of model m being correct given measurement data up to time k is given by:

$$\begin{aligned} \mu_m(k) &\equiv P\big(M_m|z^{1:k}\big) \\ &= \frac{p\big(z_k|z^{1:k-1}, M_m\big)P\big(M_m|z^{1:k-1}\big)}{\sum_{i=1}^{N} p\big(z_k|z^{1:k-1}, M_i\big)P\big(M_i|z^{1:k-1}\big)} \\ &= \frac{\Lambda_k^m\ \mu_{k-1}^m}{\sum_{i=1}^{N} \Lambda_k^i\ \mu_{k-1}^i}\,, \end{aligned} \tag{3.35}$$

where Λ_k^m is the likelihood function of mode m at time k. Assuming that the system uncertainty (P_k^m) under each model m is Gaussian with zero mean, the mode likelihood may be computed as:

$$\Lambda_k^m \equiv p(\nu_k^m) = \mathcal{N}(\nu_k^m; 0, P_k^m)\,, \tag{3.36}$$

where ν_k^m is the residual (innovation) caused by the discrepancy between the actual measurements and the predicted measurement values from mode m and P_k^m is the m^{th} mode measurement covariance.

The SMM maintains a separate filter for each candidate discrete mode. For more complex objects, the complexity of the SMM may be reduced by considering only local surface features near each estimated finger tip location. Figure 3.18 illustrates how the previous state estimate and covariance may be used to compute the next estimate and covariance given multiple modes and their filters. The filter associated to each mode produces a *mode-conditioned* state estimate, $\hat{X}^m$, and associated covariance, P^m, and the *mode likelihood*, Λ^m. Under the linear-Gaussian system assumption, the pdf of the object state is a Gaussian mixture using the Theorem of total probability:

$$p\big(X|Z^k\big) = \sum_{m=1}^{N} \mu_m(k)\,\mathcal{N}(X_k; \hat{X}_k^m, P_k^m)\,. \tag{3.37}$$

The optimal minimum mean square error (MMSE) of (3.37) provides the overall state estimate and covariance formulae:

$$\hat{X}_k = \sum_{m=1}^{N} \mu_k^m\ \hat{X}_k^m \tag{3.38}$$

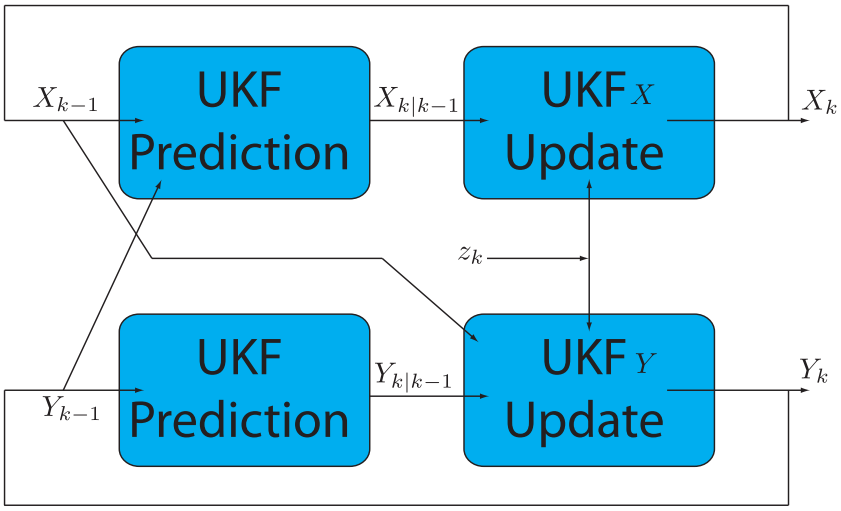


Figure 3.19: Dual unscented Kalman filter (UKF) framework.

$$P_k = \sum_{m=1}^{N} \mu_k^m \left\{ P_k^m + [\hat{X}_k^m - \hat{X}_k][\hat{X}_k^m - \hat{X}_k]^T \right\}. \tag{3.39}$$

The overall hybrid estimation process updates each continuous model state estimate, as well as the discrete mode estimator after each measurement.

3.7 Model Parameter Estimation

It is sometimes desirable to additionally estimate additional quantities such as model parameters. During object pose estimation, it is natural to estimate quantities such as the mass of the object and its center of mass location. There are two common approaches to simultaneously combine the state and parameter estimation processes. The first is called dual estimation in which the parameters Y_k are estimated in a separate filter using a Markov random walk model:

$$Y_k = Y_{k-1} + \xi \tag{3.40}$$

$$X_{k+1} = \mathbf{f}(X_k, Y_k, u_k, w_k), \tag{3.41}$$

where is X_k is the state, Y_k are the model parameters at time k, $\mathbf{f}$ is the dynamic predictive model, and ξ is zero mean Gaussian noise.

The second approach is called joint estimation, in which the parameters Y_k are concatenated

onto the state vector X_k to form a new augmented vector, X_k^a:

$$X_k^a = \begin{bmatrix} X_k \\ Y_k \end{bmatrix} . \tag{3.42}$$

Therefore one filter is used to jointly estimate both the state and the parameters. As discussed in [81], the joint filter approximates the MAP estimate by maximizing the joint density of the state and parameters. An augmented dynamic predictive model can now be formed by augmenting the original predictive model $\mathbf{F}$ with the predictive model of the parameters, which is identity:

$$X_{k+1}^a = \begin{bmatrix} \mathbf{f}(X_k, Y_k, u_k, w_k) \\ \mathbf{I} \cdot Y_k \end{bmatrix} . \tag{3.43}$$

With either dual or joint estimation, assuming that the parameters are constant over time, any type of filter (e.g. Kalman or particle) may be used, and the choice ultimately depends upon the application.

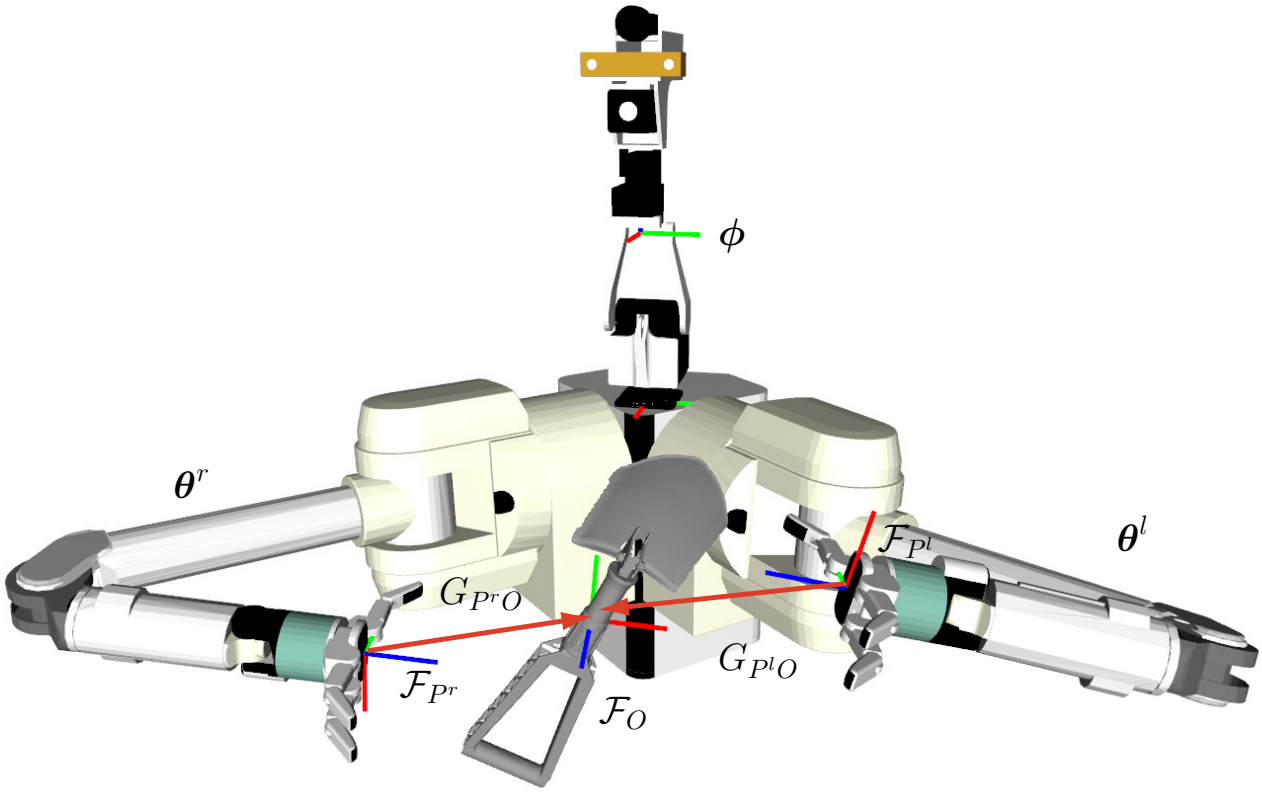


Figure 3.20: Dual-arm state representation.

3.8 Dual-Arm Estimation

I investigate multiple approaches to dual-arm grasping, which are described below. Dual-arm grasping requires both arms to interact with the object. During joint contact of the arms with the object, the state system must be augmented to estimate the state of both arms and to model the interaction from both arms. A predictive model must be augmented to include both additional states, the arm state and object state of the other arm:

$$\begin{bmatrix} G_{P^lO} \\ G_{P^rO} \\ \boldsymbol{\theta}^l \\ \boldsymbol{\theta}^r \\ \phi \end{bmatrix}_{k+1} = \begin{bmatrix} I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_m(\Delta\boldsymbol{\Theta}^l_k) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{f}_m(\Delta\boldsymbol{\Theta}^r_k) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{f}_n(\Delta\boldsymbol{\Phi}_k) \end{bmatrix} \begin{bmatrix} G_{P^lO} \\ G_{P^rO} \\ \boldsymbol{\theta}^l \\ \boldsymbol{\theta}^r \\ \phi \end{bmatrix}_k + \mathbf{w}_k \,, \tag{3.44}$$

This model generates two estimates for the object location in the base frame $\mathcal{F}_B$. The two estimates must be consistent and point to the same location in space. This requirement is handled through a

constraint that is described below.

3.8.1 Dual-Arm via Independent Filters

Since each robot arm is independent from the other arm until a firm dual-arm grasp is made, two filters may be run in parallel to estimate each arm's state. Each filter runs the single arm estimation described starting in Section 3.3 with the following left and right arm states:

$$X_k^l = \begin{bmatrix} \boldsymbol{\theta}^l \\ \boldsymbol{\phi}^l \\ G_{P^lO} \end{bmatrix} \qquad X_k^r = \begin{bmatrix} \boldsymbol{\theta}^r \\ \boldsymbol{\phi}^r \\ G_{P^rO} \end{bmatrix} \tag{3.45}$$

This allows for a multi-threading implementation, but also requires constraining both object state estimates to specify the same object frame in space. Therefore, a simple procedure to couple the two independent estimates can be obtained by fusing both left and right estimates using the uncertainty of each estimate, as weighed by the inverse of their covariances:

$$G_{BO}^{\dagger} = \frac{\Sigma_{G_{P^rO}}^{-1} G_{BP^r}(\boldsymbol{\theta}^r) G_{P^rO} + \Sigma_{G_{P^lO}}^{-1} G_{BP^l}(\boldsymbol{\theta}^l) G_{P^lO}}{\Sigma_{G_{P^rO}}^{-1} + \Sigma_{G_{P^lO}}^{-1}}\,, \tag{3.46}$$

where $\Sigma_{G_{PO}}$ is the square sub matrix pertaining to the object state G_{PO} of the full covariance matrix P:

$$P = \begin{bmatrix} \Sigma_{\boldsymbol{\theta}} & \Sigma_{\boldsymbol{\theta\phi}} & \Sigma_{\boldsymbol{\theta} G_{PO}} \\ \Sigma_{\boldsymbol{\phi\theta}} & \Sigma_{\boldsymbol{\phi}} & \Sigma_{\boldsymbol{\phi} G_{PO}} \\ \Sigma_{G_{PO}\boldsymbol{\theta}} & \Sigma_{G_{PO}\boldsymbol{\phi}} & \Sigma_{G_{PO}} \end{bmatrix} \tag{3.47}$$

To obtain the object state relative to each palm frame, another transformation is applied:

$$G_{P^rO}^{\dagger} = G_{BP^r}^{-1}(\boldsymbol{\theta}^r) G_{BO}^{\dagger} \tag{3.48}$$

$$G_{P^lO}^{\dagger} = G_{BP^l}^{-1}(\boldsymbol{\theta}^l) G_{BO}^{\dagger}\,. \tag{3.49}$$

Similarly, each filter constructs its own neck state estimate, $\boldsymbol{\phi}$. To ensure each estimate is consistent and to ensure that visual measurements are correctly and consistently processed, the neck states must be fused together similar to Equation 3.46:

$$\boldsymbol{\phi}^{\dagger} = \frac{\Sigma_{\boldsymbol{\phi}^r}^{-1} \boldsymbol{\phi}^r + \Sigma_{\boldsymbol{\phi}^l}^{-1} \boldsymbol{\phi}^l}{\Sigma_{\boldsymbol{\phi}^r}^{-1} \boldsymbol{\phi}^r + \Sigma_{\boldsymbol{\phi}^l}^{-1} \boldsymbol{\phi}^l}\,, \tag{3.50}$$

where $\Sigma_{\boldsymbol{\phi}}$ is the square sub matrix pertaining to the neck state $\boldsymbol{\phi}$ of the full covariance matrix P.

While independent filters provide a computational benefit of parallel execution, there is a major

disadvantage in that each filter does not know the state of the other arm. Hence, each filter cannot update the object state using kinesthetic data, like tactile and force-torque measurements, of the opposite arm.

3.8.2 Dual-Arm Augmented Filter

Instead of dual independent filters, another option is to augment the single arm filter with the second arm and object state:

$$X_k = \begin{bmatrix} \boldsymbol{\theta}^r \\ \boldsymbol{\theta}^l \\ \boldsymbol{\phi} \\ G_{P^rO} \\ G_{P^lO} \end{bmatrix} . \tag{3.51}$$

Similar to the independent filter approach, it is still necessary to produce a consistent estimate. Therefore, it is required to use the state estimate, Equation 3.46, in the filter.

Julier and LaViola [82] introduced an approach to incorporate these types of effects into a Sigma Point filter. See Chapter 2 for a review on sigma point filters. Their two step approach utilizes a sigma point transform that first constrains the probability distribution of the belief and then constrains the conditional mean of the belief distribution. An equality constraint between the state variables may be written in the form:

$$c(X_k) = 0\,, \tag{3.52}$$

and presumes the existence of a projection function $\mathbf{w}(X_k)$ such that

$$c(\mathbf{w}(X_k)) = 0 \qquad \forall X_k \in \mathbb{R}^n\,. \tag{3.53}$$

Assuming that the unconstrained state X_k^* and the covariance P_k^* have already been calculated, the projection operator is applied to every sigma point in the unconstrained state distribution. The additional constraints reduce the uncertainty of the state distribution and causes the covariance to decrease. Letting $\chi_k^\dagger = \mathbf{w}(\chi_k^*)$ be the constrained sigma points, the mean and covariance of the constrained state are given by:

$$\hat{X}_k^\dagger = E\left[\mathbf{w}(\chi_k^*)\right] \tag{3.54}$$

$$P_k^\dagger = E\left[(\chi_k^\dagger - \hat{X}_k^\dagger)(\chi_k^\dagger - \hat{X}_k^\dagger)^T\right] . \tag{3.55}$$

The mean constrained state and covariance estimates in Equations 3.54 and 3.55 are calculated

using the regular sigma point transform as presented in Section 2.4.5:

$$\hat{X}_k^\dagger = \sum_{i=0}^{2L} \pi_i \, \chi_k^\dagger \tag{3.56}$$

$$P_k^\dagger = \sum_{i=0}^{2L} \pi_i \, [\chi_{i,k}^\dagger - \hat{X}_k^\dagger][\chi_{i,k}^\dagger - \hat{X}_k^\dagger]^\mathsf{T} \,, \tag{3.57}$$

where π_i are the sigma point weights. For a review of the sigma point filter, please read Section 2.4.5.

The expectation of the constrained distribution may not lie on the constraint surface, therefore the projection operation is applied again to $\hat{X}_k^\dagger$ and similarly, the covariance is increased to account for the fact that the minimum squared estimate is adjusted:

$$\hat{X}_k = \mathbf{w}(\hat{X}_k^\dagger) \tag{3.58}$$

$$P_k = P_k^\dagger + (X_k^\dagger - \hat{X}_k)(X_k^\dagger - \hat{X}_k)^T \,. \tag{3.59}$$

For dual arm estimation, the projection function $\mathbf{w}(\chi_k^*)$ is the object frame fusion Equation 3.46, but using the unconstrained sigma points χ_k^*:

$$G_{BO}^\dagger = \frac{\Sigma_{G_{P^rO}}^{*-1} G_{BP^r}(\boldsymbol{\theta}^{*,r}) G_{P^rO}^* + \Sigma_{G_{P^lO}}^{*-1} G_{BP^l}^*(\boldsymbol{\theta}^{*,l}) G_{P^lO}^*}{\Sigma_{G_{P^rO}}^{*-1} + \Sigma_{G_{P^lO}}^{*-1}} \,, \tag{3.60}$$

to produce the full constrained state:

$$X_k^\dagger = \begin{bmatrix} \boldsymbol{\theta}^r \\ \boldsymbol{\theta}^l \\ \boldsymbol{\phi} \\ G_{P^rO}^\dagger \\ G_{P^lO}^\dagger \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}^r \\ \boldsymbol{\theta}^l \\ \boldsymbol{\phi} \\ G_{BP^r}^{-1}(\boldsymbol{\theta}^r) G_{BO}^\dagger \\ G_{BP^l}^{-1}(\boldsymbol{\theta}^l) G_{BO}^\dagger \end{bmatrix} . \tag{3.61}$$

The constraint equation for this projection function specifies that the object frame for both left and right estimates be at the same location. This is essentially a kinematic closure equation:

$$c(X_k^\dagger) = G_{P^rO}^\dagger - G_{P^lO}^\dagger = 0 \,. \tag{3.62}$$

It is important to note that the projection function and constraint equation be applied to the unconstrained sigma points χ_k^*.

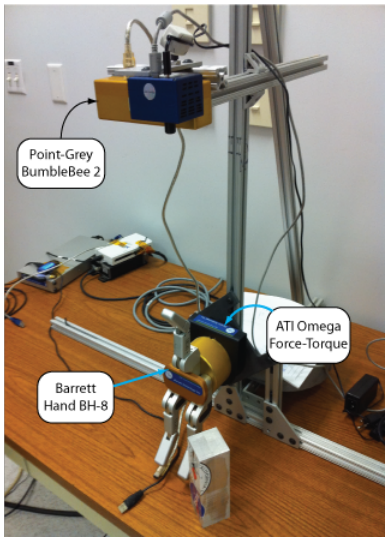


Figure 3.21: Experimental setup for hybrid estimation results with static grasping

3.9 Experimental Results

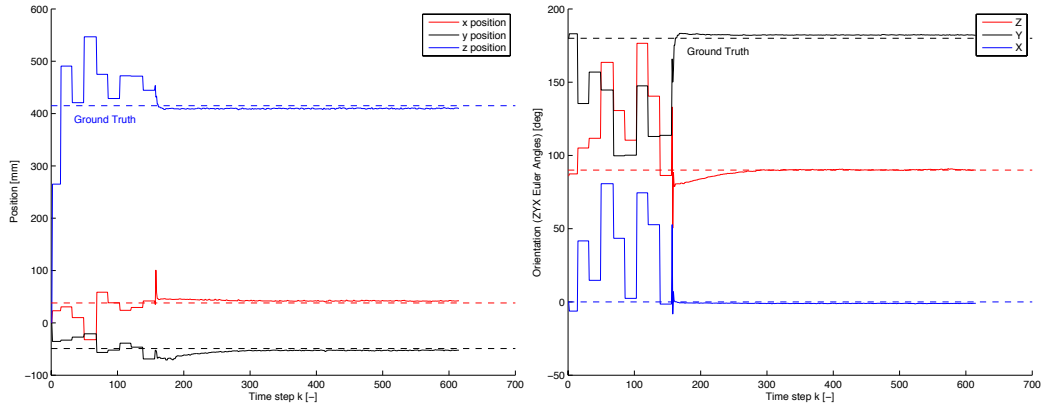
The following subsections details the experimental results of the preceding theory. First, experiments are shown using hybrid estimation theory on static grasps. Second, visual arm tracking results are presented. Lastly, single arm and object estimation is presented, which is followed by dual arm and object estimation.

3.9.1 Hybrid Estimation Results with Static Grasping

This experiment only verifies and demonstrates a subset of the algorithms described above. I implemented this subset on the following system. The grasper is a 3-fingered Barrett^{TM} hand, model BH-8, with internal strain gauges in each finger. These sensors allow one to infer if a finger contact is active. Joint encoders are located at the proximal joint. An ATI omega 6-axis force-torque sensor is mounted at the manipulator wrist. In this static experiment, the wrist is fixed rigidly to a frame, though the frame can be moved to simulate the motion of an arm. A Point-Grey Research Bumblebee 2 is mounted above the working area and is also fixed rigidly. The experimental object is an aluminum rectangular block (length: 15.24 cm, width: 5.08 cm, height: 5.08 cm) whose center of mass location is known accurately. The block is covered with stickers provides texture for the vision system. The block is placed at an appropriate height to be grasped by the Barrett Hand. At the outset, the vision system is already providing a vision estimate of the object pose. Once grasped, the combined filter begins to incorporate measurements from the fingers and the force-torque sensor.

The initial conditions are chosen to be at the camera sensor origin. Figure 3.11 illustrates a typical grasping scenario. The upper right hand corner is a rectified image of the stereo pair and the bounding box is displayed around the object. The lower left corner depicts the block with

learnt SIFT features (yellow dots), current detected SIFT features (red dots), and estimated finger locations on the object (large blue circles). Figure 3.22 shows the state estimate of the object pose

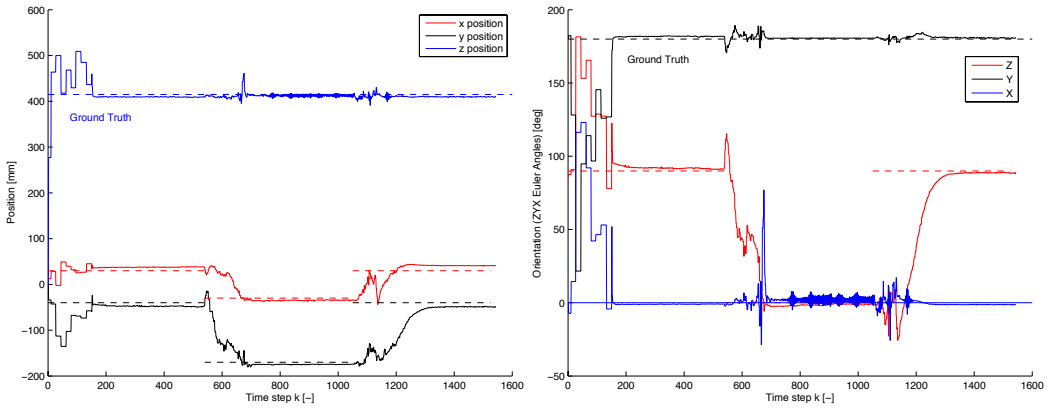


(a) XYZ position (b) ZYX Euler Angles

Figure 3.22: Estimate of object pose using a static multiple model estimator

in a static grasp. The ground truth position was measured with respect to a known location in the camera's frame. The mean error in the position estimate was 5.2 mm. The ground truth orientation was measured by visually aligning the axis of the block to a known orientation in the hand. The mean error in orientation was 1.51° assuming the ZYX angles were $(90, 180, 0)$.

A second experiment was performed to test the basic tracking capability of the system. The block was grasping in a similar position for a few moments, and then it was rotated manually in the grasp by 90° about the Z-axis. The block is then rotated back to the original position. Figure 3.23 shows the time varying state estimate of the object position, demonstrating this rotation inside the grasp. The mean error after a few seconds when the estimator steadies is 6–7 mm. The mean orientation



(a) XYZ position (b) ZYX Euler angles

Figure 3.23: Estimate of object pose, where object undergoes 90° planar rotation within grasp at $t_k = 540$ and $t_k = 1080$

error is again about 1.58°. A third and similar grasp is performed except that the block is rotated in increments of 45° up until 180°. Figure 3.24 shows the orientation estimate, with a mean error of 2.21°.

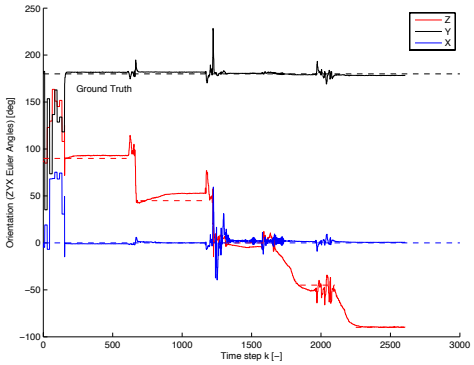
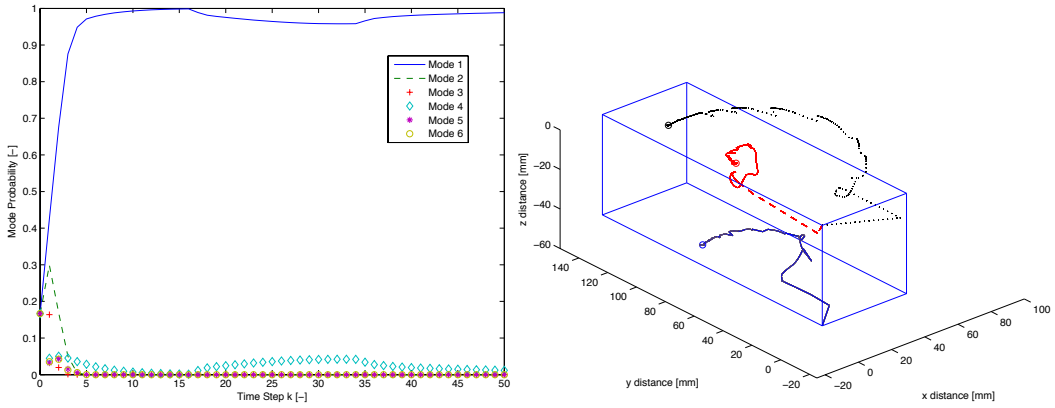


Figure 3.24: Orientation state where object undergoes 180° planar rotation in 45° increments within grasp



(a) Mode probability of grasping a block using SMM

(b) Finger positions assuming Mode 1

Figure 3.25: Contact mode and finger position estimation on a regular block

At the instant the object is grasped, an SMM is initialized to handle the possible finger contact states. To demonstrate the SMM, a subset of the possible contact modes are chosen, allowing all antipodal point grasps resulting in 6 possible contact modes. Antipodal grasps are a pair of grasp points on an object whose normal vectors are collinear and point in the opposite direction. Figure 3.25a illustrates a typical grasp in which the correct mode, mode 1, is quickly found. The mode estimate converges to a probability of 1 in less than a second, assuming a measurement rate of $\sim$25 Hz.

The various sensors used for this experiment were enabled in different combinations to additionally evaluate the contributions of each sensor to overall performance as shown in Figure 3.26. The visual estimate using feature-based measurements (SIFT) with a stereo camera produces a noisy, estimate of the pose with about 1cm jumps. When a force-torque sensor is enabled, the estimated

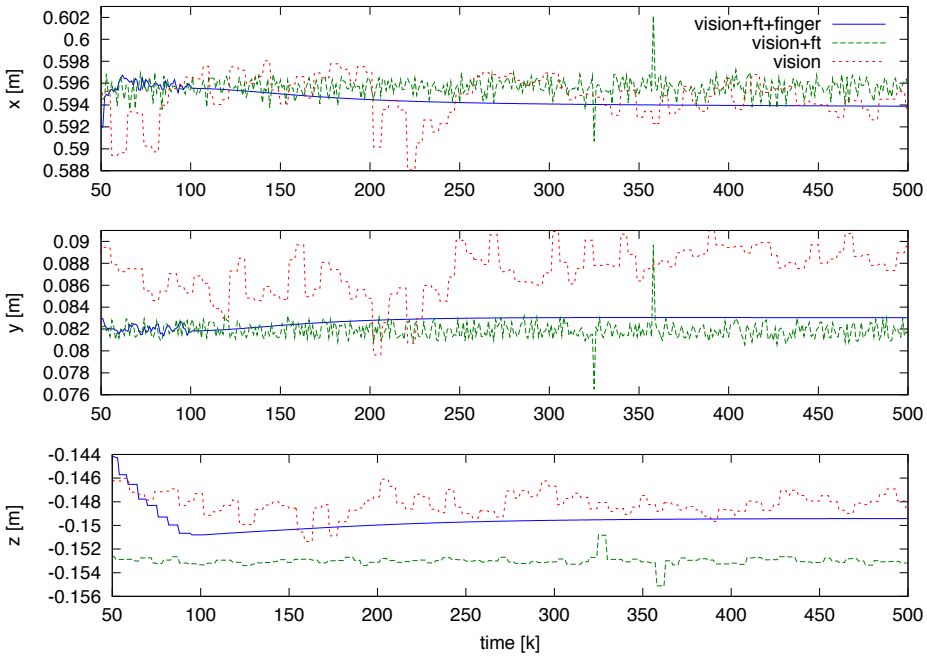
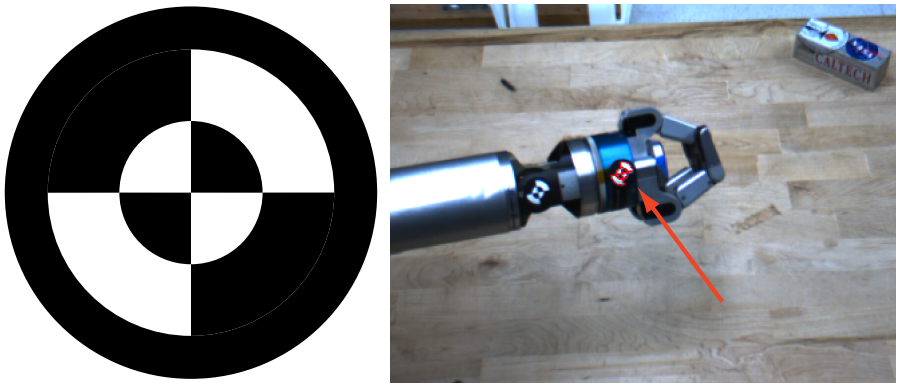


Figure 3.26: Contributions of multiple sensors to overall estimated signal. The red dashed line is vision alone. The green wide dashed line is vision and force-torque sensors. The blue solid line is vision, force-torque, and tactile sensors.

signal becomes slightly less noisy but with $\sim 1\,\text{cm}$ jumps. Lastly, pseudo-finger measurements were enabled with both the force-torque and visual measurements, the signal became quite steady, and was not subject to noise cause by stereo and force-torque sensors.

3.9.2 Experiment with Single-Arm Visual Tracking

The performance of the different visual cues used in manipulator tracking as discussed in Section 3.5.1, is assessed by evaluating each cue's contribution to the estimate of the manipulator

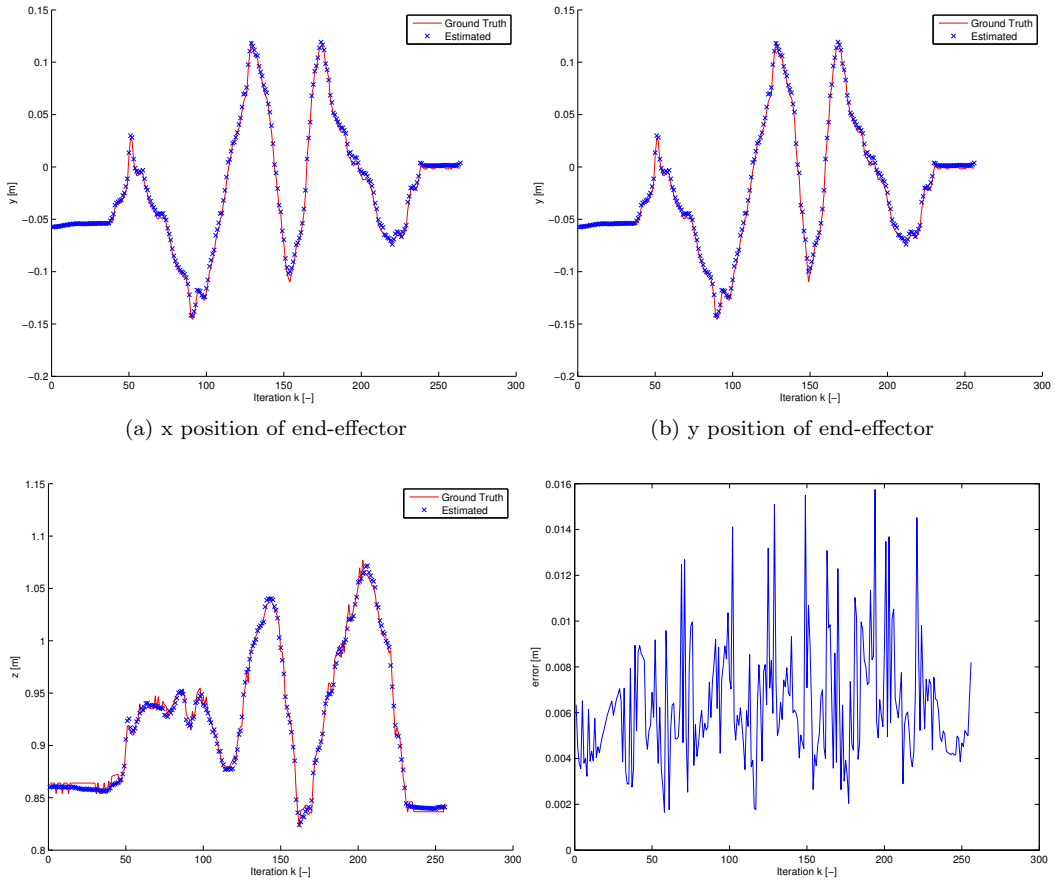


(a) Mars Science Laboratory fiducial (b) Detected fiducial on manipulator shown in red

Figure 3.27: Fiducial tracking for ground truth

configuration. The ground truth is computed by using a fiducial on the end-effector and tracked using the stereo camera. The fiducial was not used by the estimator. The fiducial is identical to the Mars Science Laboratory (MSL) fiducial as shown in Figure 3.27a. The system setup for this analysis was done using a multi-sensor head, a Barrett^{TM}WAM arm and hand. The sensor head is comprised of a high-resolution Prosilica^{TM}GC2450 (2448×2050 pixels) used for silhouette tracking, a Point-Grey^{TM}bumblebee2 color stereo camera (684×512 pixels) used for template tracking and lastly a Asus Xtion RGB-D camera (640×480 pixels) used in articulated ICP (shape tracking).

The ground truth fiducial detector works by matching gradients in the image to the gradients of the fiducial. The fiducial is matched against the inner and middle circles and the perpendicular edges. An initial estimate of the pose of the fiducial is required (which is computed using the forward kinematics of the manipulator) so that it can be projected into the image using a CAHVOR model (see Section 2.3). It is then matched in a region of interest in an area determined by the initial projection. The matching is done in the left and right images and consequently a disparity is calculated and finally from the disparity, a 3D location of the fiducial can be calculated.



(a) x position of end-effector

(b) y position of end-effector

(c) z position of end-effector

(d) Magnitude of error in x,y,z position of end effector with RMSE = 6.8 mm

Figure 3.28: Dynamic tracking of manipulator end-effector using articulated ICP and appearance

Table 3.1: Error in end-effector position from various visual cues

Visual Cue(s)	Error (cm)
Silhouette	1.13
Silhouette + Template	0.5
Silhouette + AICP	0.7
Template	0.5
Template + AICP	0.4
AICP	0.6
Silhouette + Template + AICP	0.3

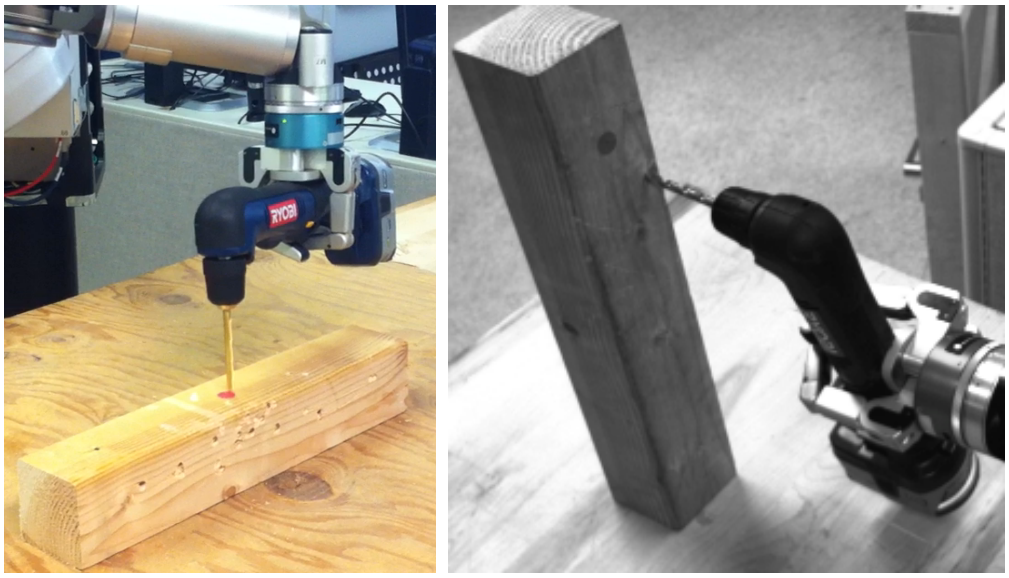
The first experiment tested dynamic tracking of a single arm using both shape (articulated ICP from the Xtion camera) and appearance (templates used in the stereo camera). The results are shown in Figure 3.28. The fusion of both cues was done using the Unscented Kalman Filter (UKF), which was discussed in Section 2.4.5. The x and y tracking is extremely accurate (RMSE of 3.9 mm) while there is some uncertainty and noise associated with depth (z). The depth accounts for the majority of the error. This can be attributed to a couple of reasons. First, the error might be due to an incorrect extrinsic calibration of the cameras. Secondly, the ground truth measurements are not filtered and hence have slight incorrect jumps in the output. The root mean square error (RMSE) of the combined AICP and Template tracker is 6.8 mm.

The second experiment is a comparison of the fusion of various visual cues of the same manipulator as experiment 1, but in a static scene, where the arm and neck are not moving. The performance of the cues is summarized in Table 3.1, which apportions contributions of each measurement type to overall system error. The visual arm tracking measurements are filtered and fused using the Unscented Kalman Filter as discussed in Section 2.4.5 and are compared against the ground truth measurements of the fiducial. This error is computed by compute the magnitude of the error vector between the 3D position of the end effector as determined by the fiducial and the visual arm tracking.

3.9.3 Results with Single-Arm and Object Manipulation

To further investigate and validate this approach, a manipulation task was chosen which involves autonomous drilling of a red circle on a block affixed to a table. In this task, the robot hand/arm must grasp the drill from an unknown position, find a red dot, and position the drill point on the circle before advancing the drill bit. The robot system is the DARPA ARM-S robot which consists of a multi-sensor head, a Barrett^{TM} WAM arm and Barrett^{TM} hand, equipped with a wrist-mounted force-torque sensor and tactile sensors on the palm and distal joints of the hand. The sensor head is comprised of a high-resolution Prosilica^{TM} GC2450 (2448×2050 pixels) , a Point-Grey^{TM} bumblebee2 color stereo camera (320×240 pixels) and a Mesa Imaging^{TM} Swiss Ranger SR4000 (176×144 pixels).

The drill and block are placed on the table. The objects are detected and their poses are initially determined, using vision, which leads to some uncertainty about their initial poses. Grasp points are then generated based on these poses. A plan is then generated to position the hand of the manipulator to a chosen grasp point which minimizes a particular cost function. Once grasped, a plan is then generated and executed blindly to bring the drill tip over the red dot. Also, at the time of grasping, the estimation process is started but it is not used as a visual servo correction. Once the drill is over the red dot, the process is stopped to allow for measuring the actual blind error. The estimated error is used as a visual servo Cartesian command. This error serves as a check on the estimator performance because the red dot location can be measured as a ground truth, and if a visual servo command was issued, there might be additional error associated with this motion.



(a) Drilling into block flat on table with red dot

(b) Drilling into block standing up on table with red dot

Figure 3.29: Drilling into block experiments.

Table 3.2 summarizes the results of the experiment.

The computation for this experiment was done on Dual Quad Core Intel Xeons (3.33 GHz) with 24 GB DDR3 RAM and a 1.5 GB NVIDIA Quadro FX4800 graphics card. The algorithm is implemented on two main threads. The first thread operates on the acquired imagery, computes stereo, acquires point clouds from the visual sensors, and computes the manipulator kinematics. The second thread implements the filter thread. With combined measurements for both object and manipulator running, the filter runs at 7–8 Hz.

Table 3.2: Estimated drill tip distance and contribution of different cues

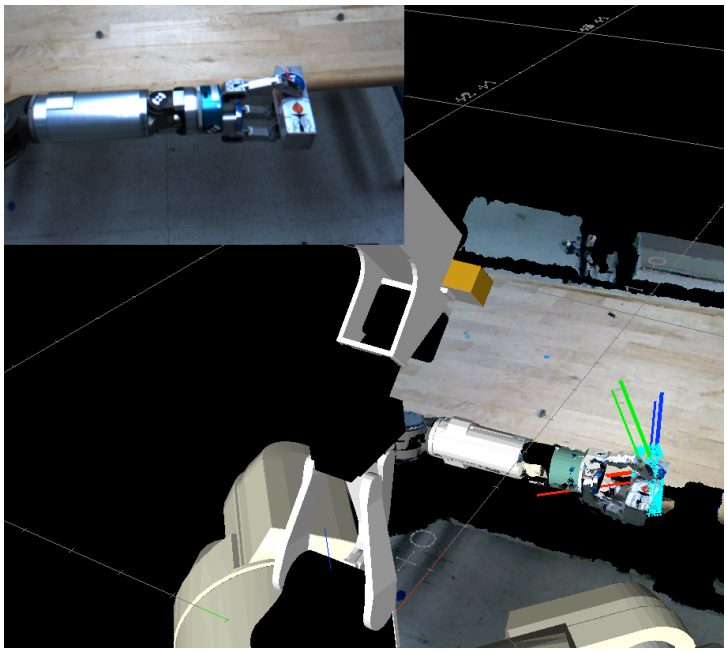
Trial	Blind Error		Shape		Shape+Tactile		Shape+Contour+Tactile	
	x (mm)	y (mm)	x (mm)	y (mm)	x (mm)	y (mm)	x (mm)	y (mm)
1	0.0	-39.8	8.6	2.2	8.4	2.2	6.8	0.2
2	-29.7	-48.6	-4.1	10.1	4.3	7.9	6.8	5.7
3	8.3	-22.0	12.4	6.7	11.9	4.1	6.5	3.7
4	-38.12	-57.1	-10.4	14.9	11.2	6.4	17.7	0.2
5	14.0	-46.8	1.9	12.5	2.5	7.3	2.1	5.0
RMSE	49.96 mm		12.8 mm		9.8 mm		6.9 mm	

Table 3.2 shows the drill tip position error realized with the various combinations of sensor cues, and the blind error (open loop). The blind error is the error without no visual corrections and is the largest at 4.9 cm. The shape cue using the Swiss Ranger camera produced an error of 1.28 cm and the shape and tactile fusion produced 0.98 cm. Using shape, tactile and the contour cue from the monocular camera, the error was down to 0.69 cm.

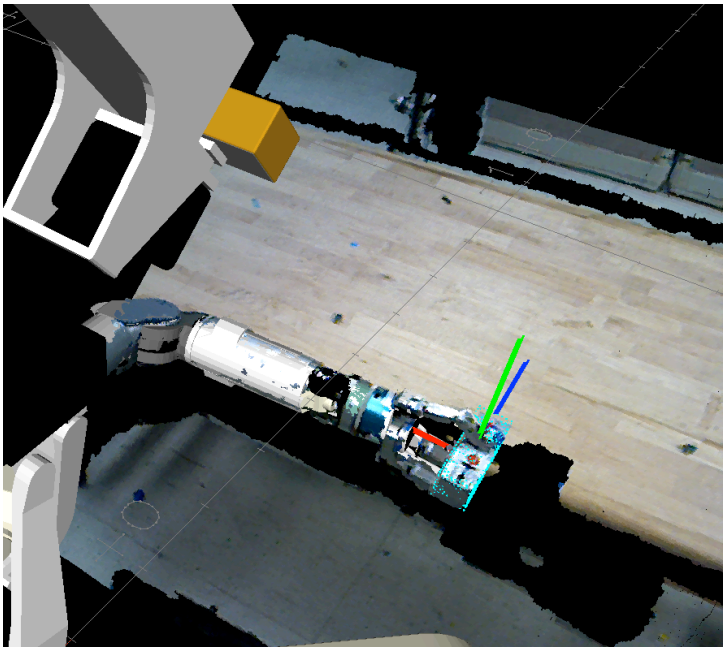
3.9.4 Experiments with Parameter Estimation

One crucial parameter that is often difficult to explicitly estimate is the grasped object's center of mass. This parameter play an equally important role to mass itself, as the center of mass affects the wrenches applied on the joints of the manipulator, and has a strong influence on grasp stability. Often times with heavier objects, proper planning of the trajectory is required so as to not over-torque the motors. Therefore, center of mass estimation is often required for objects with complicated geometry, especially with novel objects that the robot has not experienced.

The experimental validation involves using a Dual Unscented Kalman Filter which estimates both the robot state and the object state as described in Section 3.3. The state vector is $X = \{X_O, X_R\} = \{G_{PO}, \boldsymbol{\phi}, \boldsymbol{\theta}\}$, where the parameter estimated is the center of mass of the object, O, as seen from the object reference frame $\mathcal{F}_O$. Naturally, to be able to estimate such a quantity the force torque measurement model is used as described in Section 3.5.3.1. The initial state of the dual



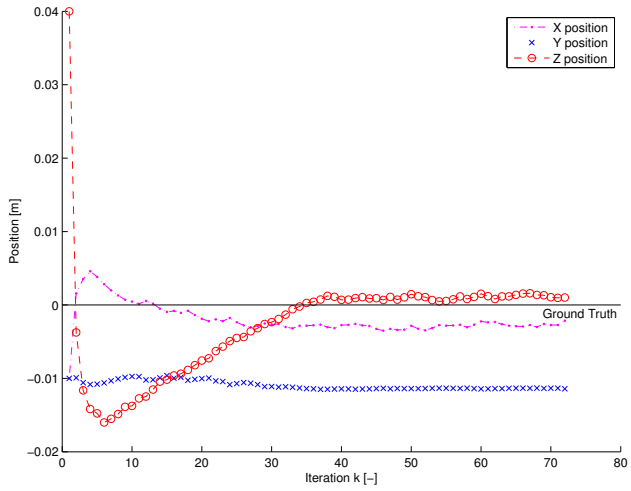
(a) Initial object state and center of mass frames



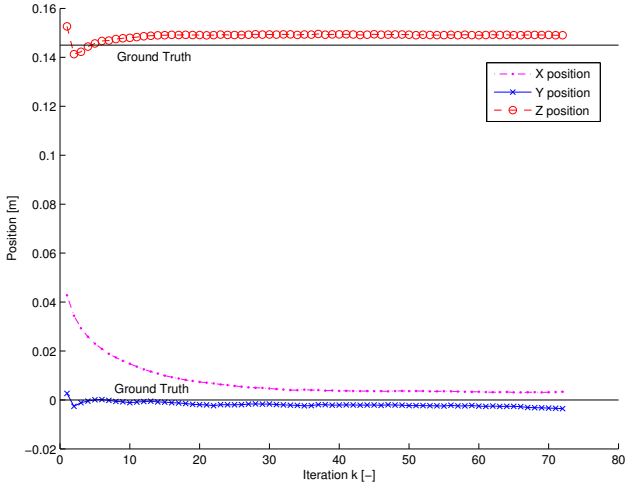
(b) Estimated object state and center of mass frames. Note that they are overlapping and appear as one

Figure 3.30: Before and after dual estimation of center of mass, object, and robot state

estimation is the rough location of the object in hand, which might be provided from a previous estimation sequence or an ICP fit as shown in Figure 3.30a. This initial estimated object location at the beginning of the estimation process is on the order of centimeters from the ground truth location. The initial starting location for the center of mass parameter is also on the order of centimeters from the ground truth location. The object to be estimated is the same block as used the experiments in



(a) Center of mass position relative to object frame and ground truth vs. filter iteration.

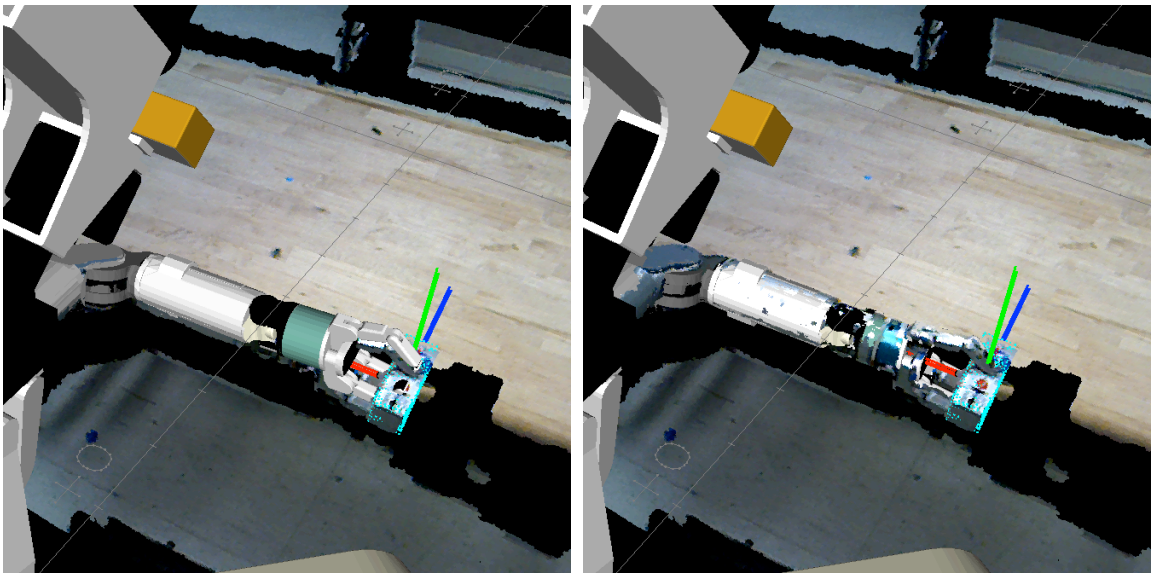


(b) Object position relative to the palm and ground truth vs. filter iteration

Figure 3.31: Center of mass position relative to object frame and object position relative to palm

Section 3.9.1, as the mass and the center of mass is crucial for the ground truth of this experiment. The estimated location and center of mass should naturally coincide as the chosen object frame $\mathcal{F}_O$ is at the center of mass. The two are shown overlapping with each other and the point cloud data in Figure 3.30b after iteration of the parameter estimation equations.

Since the force torque measurements and subsequent state updates rely heavily on accurate knowledge of the location of force-torque sensor, it is important to estimate the manipulator state accurately. Manipulating heavier objects tends to deflect the robotic arm, as in the case of tendon driven manipulators such as the Barrett manipulator, and hence visual arm tracking must be used in order to obtain meaningful estimates of the object and center of mass locations. Accurate arm location is also crucial for fusion of kinesthetic and visual sensors, and hence provides more evidence



(a) The nominal kinematic robot state. Note the hand and arm not lining up with the visual point cloud data.

(b) The estimated robot state. Note the matching of the model to the visual point cloud data.

Figure 3.32: The nominal robot state and the estimated robot state used in model parameter estimation.

for the need of visual arm tracking in manipulation. Figure 3.32 portrays the difference in the nominal robot state and the estimated robot state demonstrating the improvements realized. Note how the robot model overlays perfectly with visual point cloud data in Figure 3.32b, whereas the robot model that uses the nominal robot state is elevated away from the visual data, shown in Figure 3.32a.

The performance of the dual estimator is shown in Figure 3.31. The object position relative to the palm (position portion of G_{PO}) is shown in Figure 3.31b. The mean error of this estimate is 6.4 mm. The center of mass position is shown in Figure 3.31a. The mean error of this estimate is 11.5 mm. This is primarily due to the y estimate having an error of 11 mm. This is due to the fact that the y axis as seen in Figure 3.30b is parallel to the axis along which gravity acts. This represents a subspace in which the position cannot be estimated from the force torque sensor. A possible solution to fully estimate the center of mass is to move the object to multiple different orientations.

3.9.5 Two Arm Estimation Experiments

The system used in this experiment was similar to the system described in Section 3.9.3. This system consists of a multi-sensor head, two Barrett^{TM}WAM arms and hands each having a force-torque sensor at the wrist and tactile sensors on the palm and the distal joints of the hand. The sensor head is comprised of a high resolution Prosilica^{TM}GC2450 (2448 $\times$ 2050 pixels) , a Point-Grey^{TM}bumblebee2 color stereo camera (684 $\times$ 512 pixels) and a Asus Xtion (640 $\times$ 480 pixels)

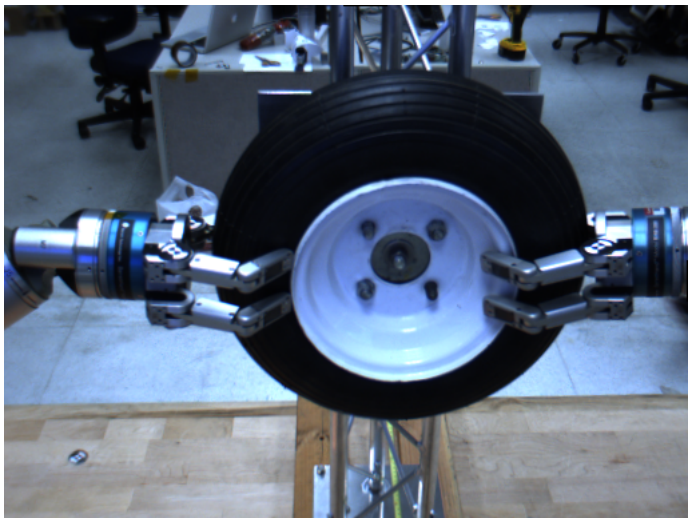


Figure 3.33: Dual-arm grasping of a wheel during wheel-hub disassembly

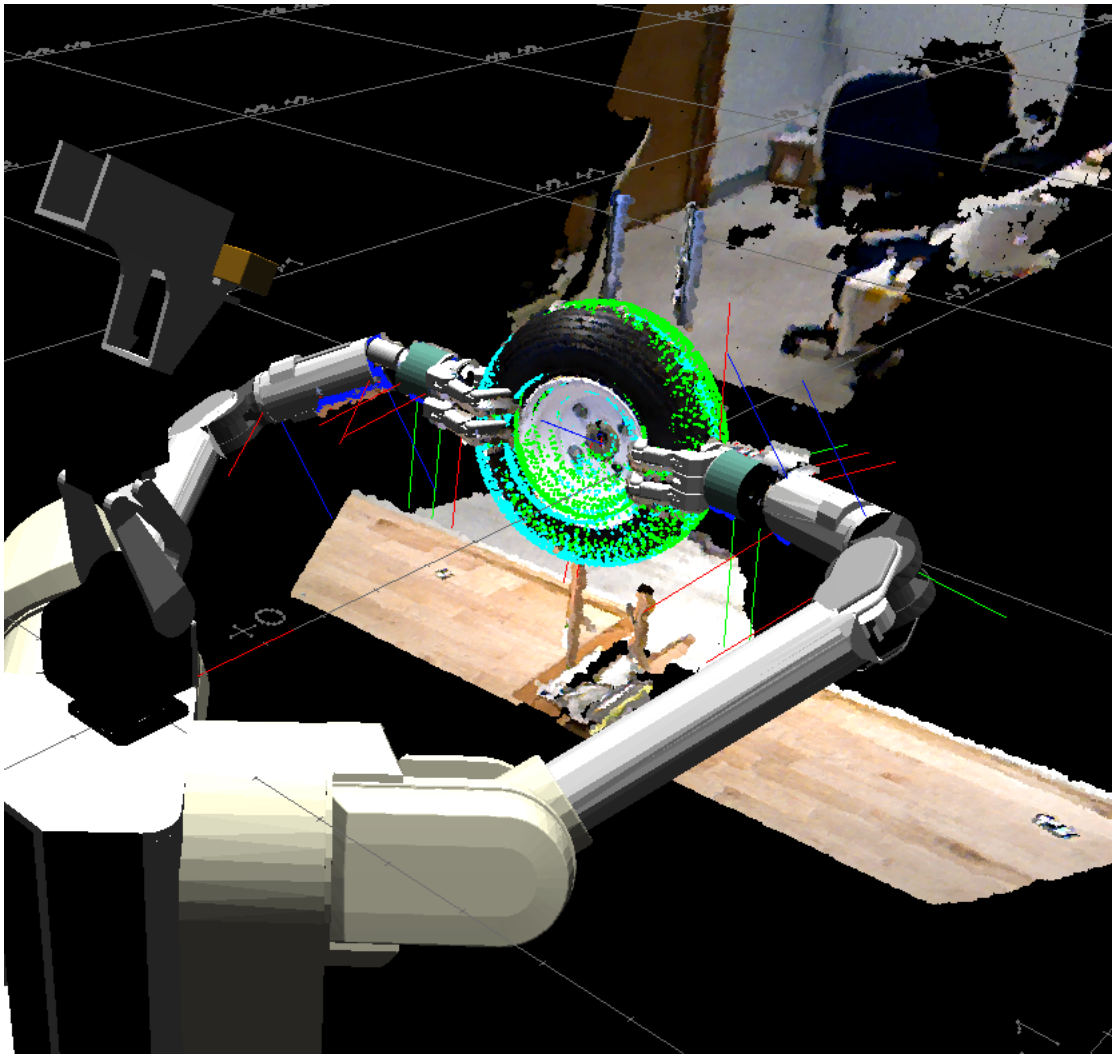
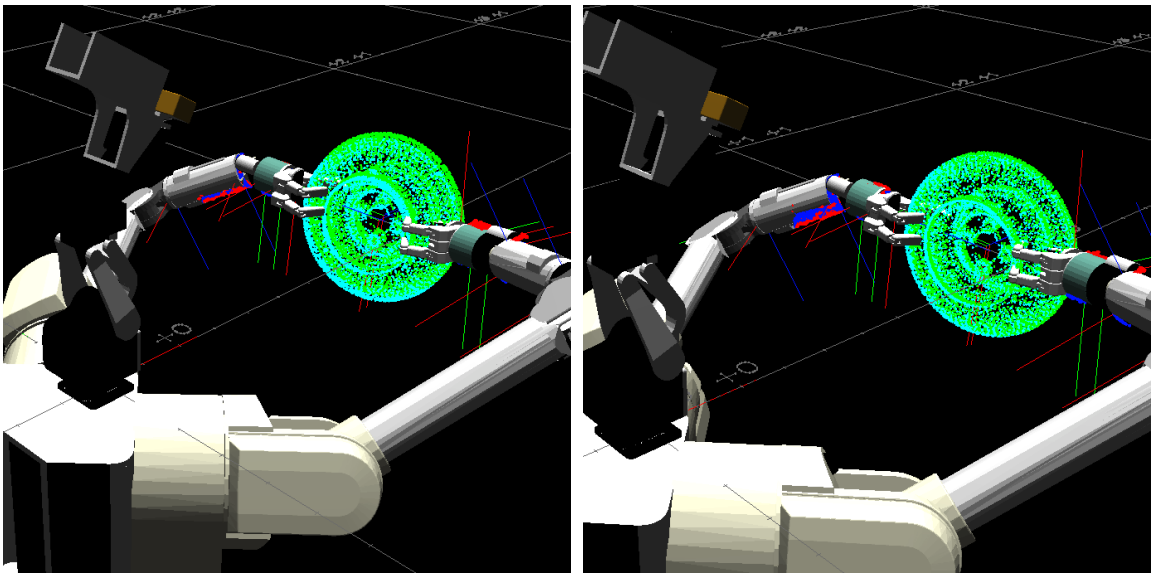


Figure 3.34: Fused wheel location estimate superimposed on Xtion sensor data and system model.



(a) Left and right estimates from two independent filters (b) Left and right estimates from augmented filter

Figure 3.35: Dual-arm grasping wheel assembly experiments.

RGB-D camera.

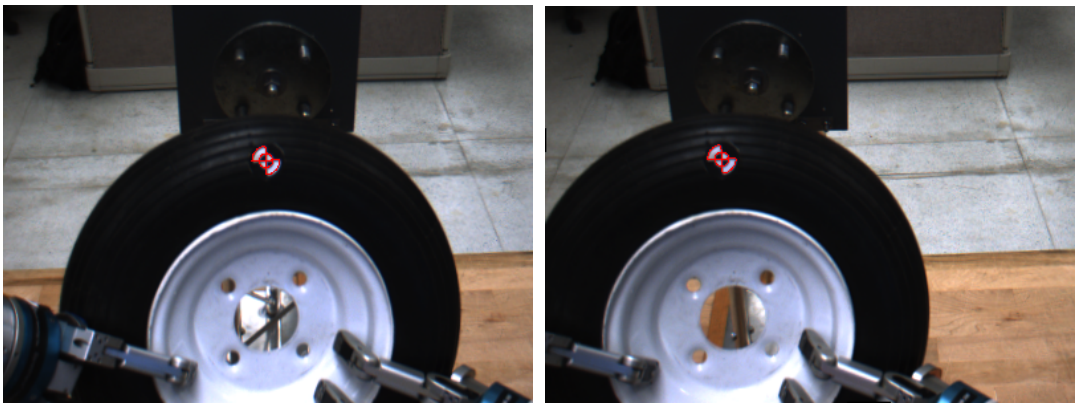
The experiments included the task of removing a wheel from a hub as shown in Figure 3.33. This is naturally a two-arm task since the object is too large and heavy for one arm to manipulate it. Also, the experimental setup is such that the wheel can be precisely attached to a frame, and hence its location can be measured externally to provide a ground truth.

The experiment begins as the robot identifies and roughly locates the wheel with ICP. Subsequently, a dual-arm trajectory plan is generated and executed to grasp the wheel. At this point, both estimators as described in Sections 3.8.1 and 3.8.2 are then executed, and their performance is compared using the ground truth measurements.

Table 3.3: Estimation errors for independent and augmented filters.

Trial	Independent			Augmented		
	x (cm)	y (cm)	z (cm)	x (cm)	y (cm)	z (cm)
1	91.7	2.6	6.0	92.1	2.7	6.2
2	91.7	2.6	6.0	92.1	2.7	6.2
3	91.7	2.9	5.9	92.1	2.3	6.3
4	91.7	2.7	5.9	92.1	2.7	6.1
5	91.9	3.5	5.9	92.1	2.8	6.2
Mean	91.7	2.8	5.9	92.1	2.66	6.2
Ground Truth	x: 91.7cm		y: 2.1cm		z: 6.2cm	
Error	0.0	0.7	0.3	0.4	0.5	0.0

The state being estimated is the robot state and object state as in Section 3.3: $X = \{X_O, X_R\} = \{G_{PO}, \boldsymbol{\phi}, \boldsymbol{\theta}\}$. The robot state is estimated using visual arm tracking as discussed in Section 3.5.1, specifically Section 3.5.1.2, in which the data from the Asus Xtion is used in an articulated ICP approach. The object measurement updates used the point cloud data from Asus Xtion and updates to the state are described in Section 3.5.2.2. The tactile locations were also used to provide contact location on the object and were used to update the state as described in Section 3.5.3.2. In both filters, the object state is estimated relative to each palm, that is: G_{P^rO} and G_{P^lO}. The left and right estimate of both filters are shown in Figure 3.35. Note that the left and right estimates of the augmented filter are slightly more in agreement with each other than the left and right estimates of the two independent filters. Both filter fuse these two estimates to provide one global estimate of G_{BO} relative to the body. The fused estimate can be seen superimposed on the point cloud data in Figure 3.34. This fused estimate is then compared against the ground truth. Both filters were compared over 5 trials and the RMSE error of the independent filter was 7.6 mm, while the augmented filter had an RMSE of 6.4 mm. Table 3.3 shows the position of the estimates, ground truth of this experiment and the errors of each filter. This modest performance improvement suggests that the extra complexity of the augmented filter does not increase performance enough to merit its use for static scenarios. This may prove different for more dynamic tracking situations, and this is future work to be investigated.



(a) Left stereo image of wheel removal task with fiducial detection

(b) Right stereo image of wheel removal task with fiducial detection

Figure 3.36: Left and right stereo images of wheel being removed from hub. Fiducials are detected in each image and are outlined in red.

A second experiment was done in a dynamic setting. The experiment involved removing a wheel from its hub and tracking the wheel as it is removed. The ground truth for this experiment was obtained using the fiducial as described in Section 3.9.2. The fiducial was attached to the wheel and it's location measured relative to the object frame $\mathcal{F}_O$. The augmented dual-arm filter (Section 3.8.2) was used for this experiment that included tactile and shape measurements. Arm

tracking did not have much impact on performance, as the majority of the arms were not in view during this experiment.

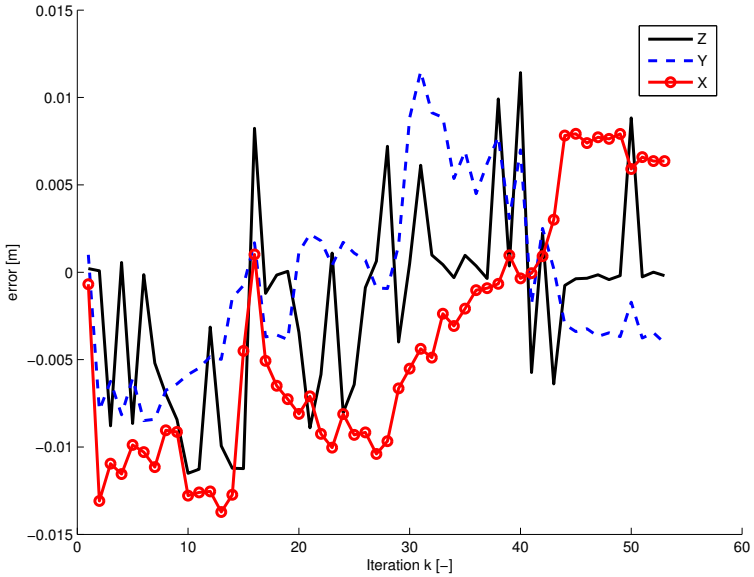


Figure 3.37: Estimation error of wheel tracking during the wheel removal task. X error shown in red with circles, Y error shown in blue dashed line, and Z error shown in black.

The tracking error in position between the filter estimate and the tracked fiducial is shown in Figure 3.37. The largest error in each axis is about 1cm. This might be due to the fact that wheel is rotationally symmetric and since the estimator used only consisted of shape measurements and not features, this error may be larger than in actuality. Overall, the RMSE for tracking the wheel as it is removed from its mounting hub is 3.9 mm.

Chapter 4

Next Best Touch

This chapter introduces a method for selecting the best action to touch/probe/interact with an object in the robot's environment in order to learn more about it. This paradigm, termed the *next best touch*, leverages information theory and an estimator to choose a touching/sensing action that would gain the most information about that object. This is useful when the robot's visual pose estimation is too poor for accurate manipulation.

First, the problem of localizing a known object using touch is considered. This approach is based on an information gain metric tailored to the application, and is computed using relative entropy. The next best touch is chosen as the action which maximizes the expected gain in the metric. The probability used in the relative entropy calculation requires state estimation. The algorithm uses a Bayes' filter, specifically a histogram filter, to update state estimates after each measurement. The measurement models incorporate the tactile sensing models described in Section 3.5.3.2 and the binary contact model described in Section 4.1.2. Simulation results with synthetic data are demonstrated for the task of localizing a door handle in Section 4.2. Experimental results obtained during the task of localizing and opening a door are presented in Section 4.3.

Section 4.4.1 explores extending the *next best touch* algorithm to select the next best action to not only localize the object, but also to estimate model parameters, associated with the object. For example, the robot may be presented with novel objects, but within a known object category or model class, and must be able to grasp and manipulate these new objects. Many object categories contain exemplars that roughly have the same shape. Principle component analysis (PCA) can be used to parametrize the mesh models of similarly shaped objects. A belief is then formed for the object state and these parameters. This belief is then used to compute the information gain metric. Simulation results using synthetic data are shown in Section 4.4.2 on a screwdriver.

Section 4.4.3 further extends the *next best touch* framework to include an unknown model class. This extension might be useful for situation in which the robot is unable to discern a particular object using vision. This might occur when the robot's viewpoints are limited, or when a robot reaches into a bag. A belief is derived for both the state and the discrete model class. Using this

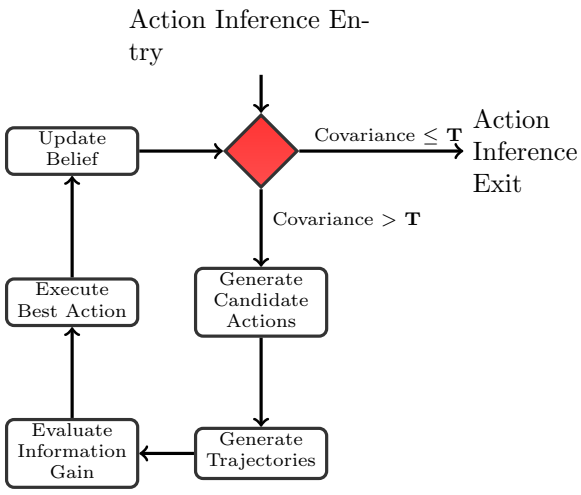


Figure 4.1: The next best touch algorithm flow diagram

belief, a similar information gain metric is computed. After each touch, the model class probability is updated until this probability exceeds a user-defined threshold.

The simulations used for the demonstration of the theory was developed using OpenGL models of the robot and objects and rendered in a viewer.

4.1 Next Best Touch (NBT) with Known Object Model

Object localization requires a state estimation framework which first forward predicts the state of world based on a new input action a, and then updates the state based on new measurements, $\hat{z}$. The general Bayes' filter maintains a state belief or posterior probability, $p(X_{t+1}|z_{1:t}, \hat{z}, u_{1:t}, a)$ and consists of two steps: *dynamic prediction* and *update*. The prediction step uses a generative dynamic model to propagate the state belief:

$$p(X_{t+1}|z_{1:t}, u_{1:t}, a) \quad = \int_{X_t} p(X_{t+1}|X_t, z_{1:t}, a)\, p(X_t|z_{1:t}, u_{1:t})\, dX_t\,, \tag{4.1}$$

where $p(X_{t+1}|X_t, z_{1:t}, a)$ is the dynamic or predictive model. The update step uses the measurement models described in Section 4.1.2, consisting of both tactile and contact measurements, which take positive and negative information ("positive information" indicates contact occurs, and "negative information" indicates lack of contact) into account. While this thesis explores contact and tactile sensors, this method maybe extended to use other visual sensors that have been used in the next best view literature. After a measurement, the state is then updated in the usual manner:

$$p(X_{t+1}|z_{1:t}, \hat{z}, u_{1:t}, a) \quad = \frac{p(\hat{z}|X_{t+1}, z_{1:t}, u_{1:t}, a)\, p(X_{t+1}|z_{1:t}, u_{1:t}, a)}{\int_{\hat{X}_{t+1}} p(\hat{z}|\hat{X}_{t+1}, z_{1:t}, u_{1:t}, a)\, p(\hat{X}_{t+1}|z_{1:t}, u_{1:t}, a) d\hat{X}_{t+1}}\,. \tag{4.2}$$

While this formulation does not require a specific type of filter, a discrete histogram filter is chosen for our implementation. In this histogram filter, the continuous state is binned into cells/divisions which form a partition of the state space and which are used in a discrete Bayes' filter. An advantage to using a histogram filter is that in the discretization of the state the implementer can specify the precision of the filter, however as a result the computational cost grows exponentially.

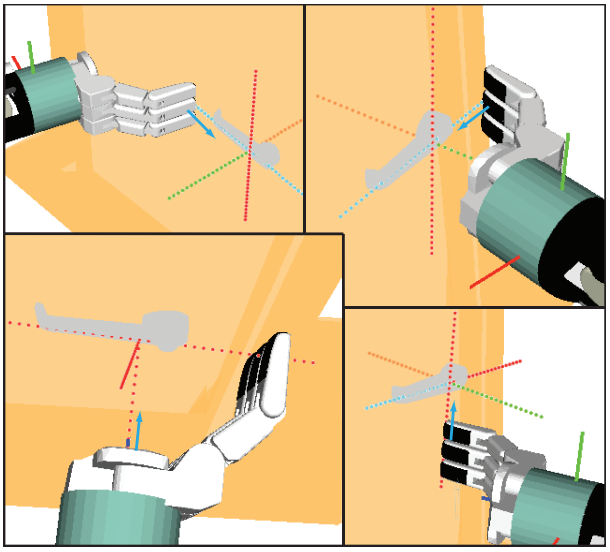


Figure 4.2: Candidate touching action directions and surfaces to contact. The Barrett^{TM} hand surfaces (palm tactels, finger tactels, side and back surfaces) contacting surfaces of door handle

4.1.1 Generate Candidate Actions

Definition 4.1. Candidate touching *actions* are comprised of a Cartesian motion direction $\vec{n}_O$ and a hand pose, namely the hand joint angles (Θ_H) and the hand (palm) transform (G_{BP^a}). ♦

The generation of candidate actions requires models of the object O and the robot R. Both are modeled using a standard polygonal mesh model $\mathcal{M}$ consisting of a fixed number of faces $\{F_i\}$ $(i = 1, \ldots, n_F)$, edges, $\{E_j\}$ $(j = 1, \ldots, n_E)$, and vertices $\{\mathcal{V}_k\}$ $(k = 1, \ldots, n_\mathcal{V})$. A reference frame F_o is rigidly attached to the object model. First, surfaces on the object model $\mathcal{M}_o$, are chosen (which may also be selected by the user) and the normals of each surface are chosen as candidate touching directions. The hand pose of the touching actions are chosen by specifying preferred surfaces of the robot's model, $\mathcal{M}_R$, which will contact the object (These surfaces will typically be the palm and/or the surfaces of the fingers). Conceivably, these surfaces can also lie on other parts of the robot structure, such as the forearm, etc. The normals of these surfaces are then aligned with the action directions $\vec{n}_O$ (shown in Figure 4.3). Since this construction does not lead to a unique hand pose G_{BP^a}, the minimum rotation from the hand's initial pose (at G_{BP^i})to the aligned action direction

is chosen, i.e.:

$$R_{P^i P^a} = \arg\min_R G_{P^i P^a} \,.$$

This minimum rotation may be computing the axis-angle ($\hat{\omega}$, Ω) rotation by aligning the r^{th} robot surface normal $\vec{n}_R^r$ (with the robot in the initial state) to the o^{th} object surface normal $\vec{n}_O^o$ both in the base frame $\mathcal{F}_B$:

$$\Omega = \cos^{-1}(\vec{n}_O^o \cdot \vec{n}_R^r) \tag{4.3}$$

$$\hat{w} = \vec{n}_O^o \times \vec{n}_R^r \,. \tag{4.4}$$

To provide a range of hand poses, since some may not be feasible, the rotation along the aligned axis $\vec{n}_O^o$ is sampled to provide a number of possible orientations about the touching action direction. Figure 4.2 illustrates some possible ways the hand may contact the object. Actions in which the hand collides with other objects are pruned out of the candidate touching actions.

In addition, human users may add extra information by specifying which surfaces/edges of the object and hand are advantageous to contact. Similarly, surfaces that are undesirable to contact may be specified.

4.1.2 Information Gain

The relative entropy of state X is the Kullback-Leibler divergence of the posterior and prior probabilities of X given new actions and measurements:

$$IG = \int_x p(X_{t+1}|z_{1:t+1}, u_{1:t+1}) \log \frac{p(X_{t+1}|z_{1:t+1}, u_{1:t+1})}{p(X_t|z_{1:t}, u_{1:t})} \,. \tag{4.5}$$

I.e., IG measures how much information is gained from the process of taking a specific measurement. In order to calculate the posterior belief, a new measurement is used to update the prior belief. Using the Bayes' filter equations (4.1 and 4.2), the information gain may be expanded using Bayes' rule and the Markov assumption:

$$IG = \int_x \frac{p(\hat{z}|X_{t+1})\, p(X_{t+1}|z_{1:t}, u_{1:t}, a_i)}{p(\hat{z})} \log \frac{p(X_{t+1}|z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t|z_{1:t}, u_{1:t})} \,. \tag{4.6}$$

Here, the new measurements are denoted by $\hat{z}$ and the possible action is denoted by a_i. However, in the context of finding the *next best action* a^*, these new measurements $\hat{z}$ are in the future and are always hypothetical. Since the specific value of these measurements cannot be known, the expectation of the relative entropy yields the expected gain in information due to a candidate action,

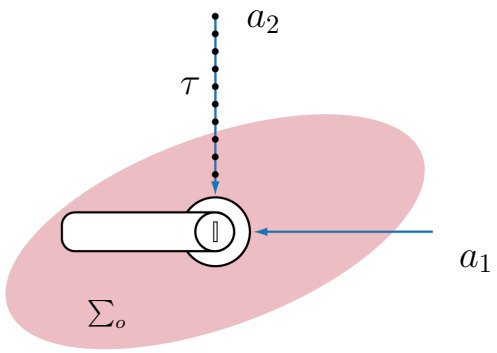


Figure 4.3: Touching action directions. Ellipse represents pose uncertainty and black dots represent the hypothetical points of contact with the object.

a_i:

$$
\begin{aligned}
& E_{\hat{z}}[IG(\hat{z}, a_i)] \qquad (4.7)\\
&= \int_{\hat{z}} p(\hat{z}) \int_x \frac{p(\hat{z}|X_{t+1})\, p(X_{t+1}|z_{1:t}, u_{1:t}, a_i)}{p(\hat{z})} \log \frac{p(X_{t+1}|z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t|z_{1:t}, u_{1:t})} \,.\\
&= \int_{\hat{z}} \int_x p(\hat{z}) \frac{p(\hat{z}|X_{t+1})\, p(X_{t+1}|z_{1:t}, u_{1:t}, a_i)}{p(\hat{z})} \log \frac{p(X_{t+1}|z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t|z_{1:t}, u_{1:t})}\\
&= \int_{\hat{z}} \int_x p(\hat{z}|X_{t+1})\, p(X_{t+1}|z_{1:t}, u_{1:t}, a_i) \log \frac{p(X_{t+1}|z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t|z_{1:t}, u_{1:t})} \,,
\end{aligned}
$$

where $p(\hat{z}|X_{t+1})$ is the likelihood of the corresponding measurement model, $p(X_{t+1}|z_{1:t}, u_{1:t}, a_i) = p(X_{t+1}|X_t, a_i)\, p(X_t|z_{1:t}, u_{1:t})$ is the intermediate belief after the state prediction of action a_i, and $p(X_{t+1}|X_t, a_i)$ is the prediction likelihood arising from the system's governing dynamics, in response to action a_i.

Similarly, the time and location of the end of an action (the moment of contact) is also unknown as the object's location is uncertain. We lump the time and location together and denote τ as the time/location at which collision occurs along the action direction. Hence the information gain formulation must take into account the uncertainty in the time of contact.

$$
\begin{aligned}
& E_{\tau,\hat{z}}\left[IG(\hat{z}, \tau, a_i)\right] \qquad (4.8)\\
&= \sum_\tau p(\tau) \int_{\hat{z}} \int_x p(\hat{z}|X_{t+\tau})\, p(X_{t+\tau}|z_{1:t}, u_{1:t}, a_i) \log \frac{p(X_{t+\tau}|z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t|z_{1:t}, u_{1:t})} \,,
\end{aligned}
$$

where $p(\tau)$ is the probability density of ending the action at time τ, and may be found using:

$$
p(\tau) = \frac{\sum_{X_{t+\tau}} \mathcal{C}(\tau)\, p(X_{t+\tau}|z_{1:t+\tau}, u_{1:t+\tau})}{\sum_\tau \sum_{X_{t+\tau}} \mathcal{C}(\tau) p(X_{t+\tau}|z_{1:t+\tau}, u_{1:t+\tau})}. \qquad (4.9)
$$

The function $\mathcal{C}$ simply indicates collision between the two bodies (represented by polygonal mesh

models, $\mathcal{M}$), namely the robot (R) manipulator when it lies at state X^R and the object (O) to touch positioned at state X^O at the time of contact τ can be determined by simulating contact using a computational contact detection algorithm. A binary function is constructed by determining intersections between two meshs or primitive shapes:

$$c = \mathcal{C}(\tau) = \mathcal{C}(\mathcal{M}_o(x^o_{t+\tau}), \mathcal{M}_R(x^R_{t+\tau})) = \begin{cases} 0 & \text{if } \mathcal{M}_o \cap \mathcal{M}_R = \emptyset \\ 1 & \text{if } \mathcal{M}_o \cap \mathcal{M}_R \neq \emptyset \end{cases} . \tag{4.10}$$

A world model with accompanying computational mechanics algorithms is used to support the collision detection algorithm. The simulation and experiments presented in this chapter use the third party Bullet[1] software. Polygonal meshes models of the objects and of the robot are used with Bullet to detect collisions.

The measurement, $\hat{z}$, is comprised of contact ($\hat{z}_1 \in \mathbb{R}^1$) and tactile ($\hat{z}_2 \in \mathbb{R}^3$) measurements which are typically used in touching. While a contact measurement could be determined from tactile measurements, force-torque sensors in the manipulator can also be used to infer contact. Therefore, the measurement likelihood can be described as:

$$\begin{aligned} p(\hat{z}|X_{t+\tau}) &= p(\hat{z}_1, \hat{z}_2|X_{t+\tau}) \\ &= p(\hat{z}_2|\hat{z}_1, X_{t+\tau}) p(\hat{z}_1|X_{t+\tau}) \,. \end{aligned} \tag{4.11}$$

The contact measurement likelihood can be described using the following binary detection model of an imperfect measurement process:

$$p(\hat{z}_1|X_{t+\tau}) = \begin{cases} P(\hat{z}_1 = 0|c = 1) = \beta \\ P(\hat{z}_1 = 1|c = 1) = 1 - \beta \\ P(\hat{z}_1 = 0|c = 0) = 1 - \alpha \\ P(\hat{z}_1 = 1|c = 0) = \alpha \end{cases} , \tag{4.12}$$

where α and β are the false positive and true negative error rates, respectively, and c is a binary variable that indicates an intersection/collision of the object and robot hand; this can be simulated using the contact detection method, $\mathcal{C}$.

Similarly, the tactile measurement likelihood is adopted from [51] and can be decomposed into the

[1] Bullet Physics Engine (`http://bulletphysics.org/wordpress/`)

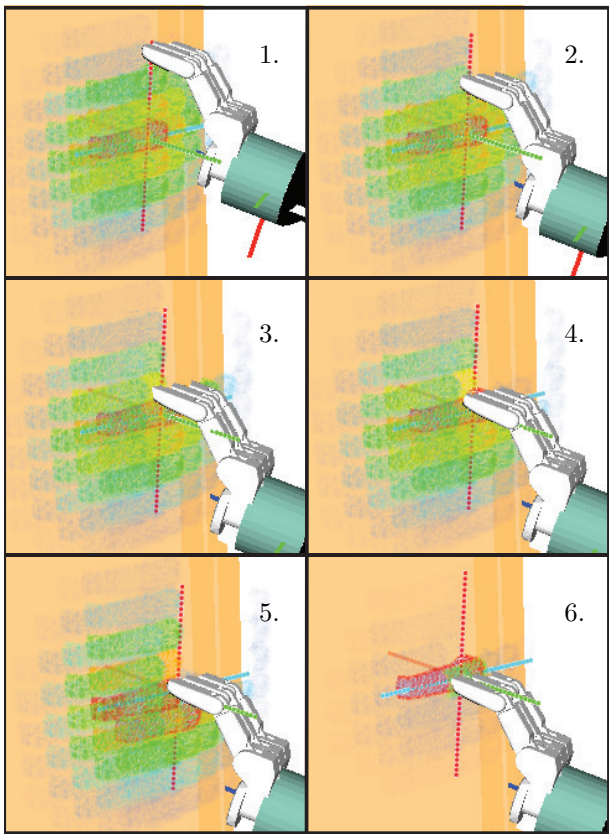


Figure 4.4: The belief distribution after a series of null contact measurements (1–5) and a positive contact at 6. Note the disappearance of possible states as the hand descends in 1–5, the change in states below the hand in 5 and the peaked distribution in 6. Blue indicates low belief and red indicates high belief.

contact position, $\hat{z}_2^p$ and the contact normal, $\hat{z}_2^n$ by independence $p(\hat{z}_2|\hat{z}_1, X_{t+\tau}) = p(\hat{z}_2^p|\hat{z}_1, X_{t+\tau})p(\hat{z}_2^n|\hat{z}_1, X_{t+\tau})$.

$$p(\hat{z}_2^p|\hat{z}_1{=}1, X_{t+\tau}) = \frac{1}{\sqrt{2\pi}\sigma_p}\exp\left(-\frac{1}{2}\frac{\mathrm{dist}(\hat{z}_p, F_i(\mathcal{M}_o(x^o_{t+\tau})))}{\sigma_p^2}\right) \tag{4.13}$$

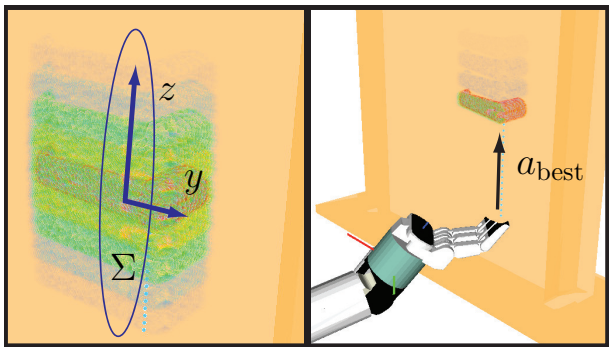
$$p(\hat{z}_2^n|\hat{z}_1{=}1, X_{t+\tau}) = \frac{1}{\sqrt{2\pi}\sigma_n}\exp\left(-\frac{1}{2}\frac{\|\hat{z}_n - n_i(\mathcal{M}_o(x^o_{t+\tau}))\|^2}{\sigma_n^2}\right), \tag{4.14}$$

where $F_i(\mathcal{M}_o(x^o_{t+\tau}))$ and $n_i(\mathcal{M}_o(x^o_{t+\tau}))$ are the face and normal most likely to cause the measurements at $x^o_{t+\tau}$ given the mesh $\mathcal{M}_o$.

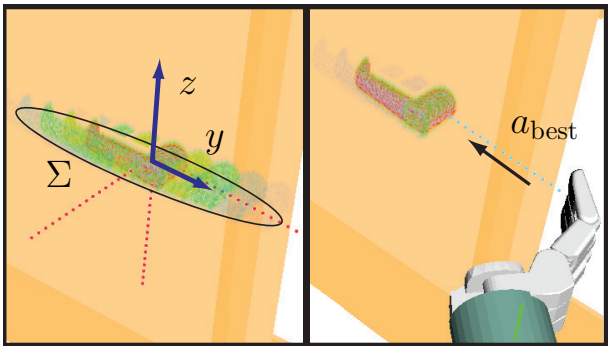
The best action a^* may be selected that causes the highest expected information gain $\overline{IG}$, while also minimizing some cost $C(a)$ incurred by taking that action:

$$a^* = \arg\max_a \left[\, \overline{IG}(a) - \gamma\, C(a) \,\right], \tag{4.15}$$

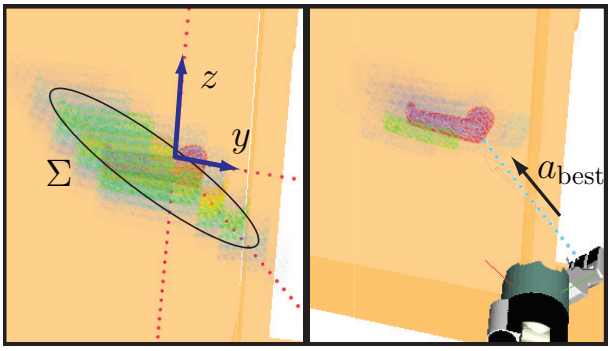
where γ is a constant to specify the relative importance of action costs to information gain.



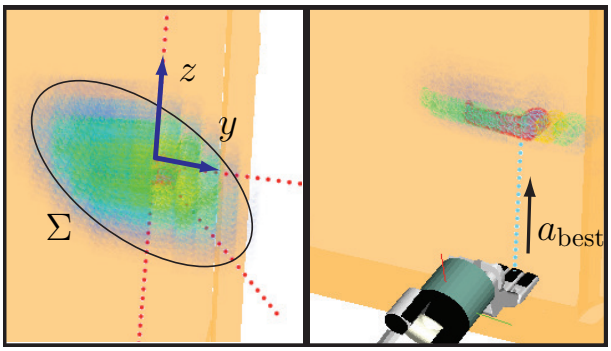
(a) Sharp prior along z axis



(b) Sharp prior along y axis



(c) Sharp prior along diagonal $z - y$ axis



(d) Wide prior along diagonal $z - y$ axis

Figure 4.5: Depiction of the Barrett$^{\text{TM}}$ hand and 4 actions touching the door handle located on a door. Four example choices of prior belief (left) and the corresponding best action (right) chosen by the method. The posterior belief (right) after a positive contact is also shown. Blue indicates low belief and red indicates high belief.

4.2 Simulation Results

In order to calculate the information gain of each candidate action, each action must be simulated. For each action, the belief of the object state is updated using a series of null contact measurements ($\hat{z}_{1:(\tau-1)} = 0$) occurring in the interval prior to the time/location (τ) at which contact is made and the information gain is to be calculated. I.e., as the robot progresses along the action direction at times $t+1, t+2, \ldots, t+\tau-1$, the measurements consist of null contacts, and the approach is continued until contact. At contact time τ, all possible measurements are used in Equation 4.8 in order to be integrated out of the expected information gain. Figure 4.4 illustrates the belief distribution changing as a result of null contact measurements (1–5) and finally (6) a positive contact (The *rainbow* color map is used with blue indicating low values and red indicating high values of belief) . Null contacts are used to update the belief (negative information) and is shown by the reduction of highly probably states in frames 1–5 of Figure 4.4. The belief changes shape, and change in belief is most evident in frame 5 as discretized states right below the hand turn to red (indicating high probability). Finally, when a positive contact between the hand surface and the object is made in frame 6 of Figure 4.4, the states in contact increase in belief and the states not in contact decrease in belief.

Secondly, Figure 4.5 shows the best action determined by the algorithm given various initial prior distributions. Given sharp beliefs (i.e. low uncertainty), approaches in the z and y axis directions, Figures 4.5a and 4.5b shows, as one might expect, the best action which provides the most gain of information about the pose of the door handle is the action that approaches the door handle while passing through the majority of the uncertainty ellipse. However, Figures 4.5c and 4.5d illustrate that the greatest information gain does not necessarily accrue in the direction of the largest uncertainty. As a result, both the shape of the object and the hand pose plays a role in the information gain calculation, as illustrated by the action chosen in Figure 4.5d.

4.3 Experimental Investigation

The next best touch algorithm described in previous sections was implemented on the DARPA ARM-S system and tested on the task of localizing a door handle and opening a door whose a priori location, while unknown, was located near the robotic torso. The robotic torso consists of a Barrett arm with seven joints, a 3-fingered Barrett hand with four actively controlled joints, and a four-jointet neck, totaling fifteen individual degrees of freedom. Objects are initially located in the environment using a suite of perception sensors that include a low-resolution stereo camera, flash lidar, and a high-resolution gigE camera. A six-axis force-torque sensor located at each hand-arm interface is used to detect contact with objects in the environment. Contact with the object or environment results in easily recognizable sensor signals. The computation is done on Dual Quad Core Intel Xeons (3.33 GHz) with 24 GB DDR3 RAM and a 1.5 GB NVIDIA Quadro FX4800 graphics card.

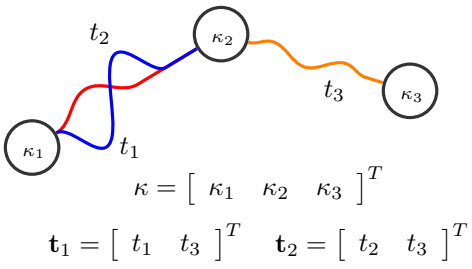
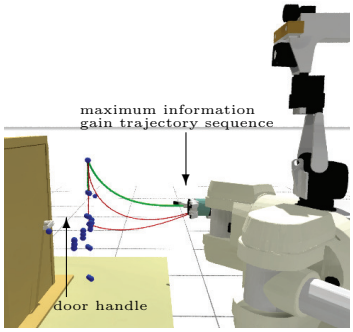


Figure 4.6: Action constraint and trajectory sequences for information gain planning. An action constraint sequence is an ordered list of state constraints (κ). A trajectory sequence is a continuous sequence of trajectories ($\mathbf{t}_1$, $\mathbf{t}_2$). κ_1 is the initial state of the robot, κ_2 is the end of the freespace maneuver, and κ_3 is the end of the interactive maneuver. In this example both $\mathbf{t}_1$ and $\mathbf{t}_2$ satisfy the state constraints in the constraint sequence κ.

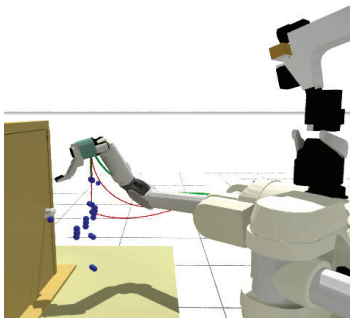
4.3.1 Motion Planning for the Next Best Touch

To demonstrate the *next best touch* framework, the technique is integrated with a motion planning algorithm that samples in the space of continuous parameterized trajectory sequences [83]. Each trajectory sequence is composed of a freespace motion $\kappa_1\kappa_2$ and an interactive motion $\kappa_2\kappa_3$. The freespace motion maneuvers the manipulator to the initial state of the interactive maneuver without interacting with the target object. The interactive motion is executed until collision with the surface of the target object is detected. To accurately detect collisions with the six-axis force torque sensor of the DARPA ARM robotic torso, the orientation of the hand is fixed during the interactive motion.

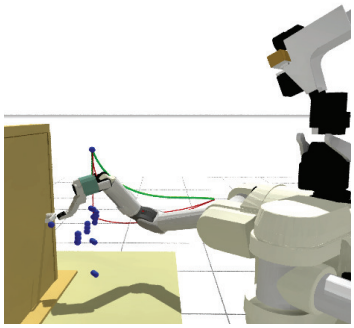
The motion planner first forms a set of action constraint sequences κ. The first constraint κ_1 in each constraint sequence is the initial state of the robot. The second, κ_2, and third, κ_3, constraints are the boundary states of the interactive motion. Additional constraints on the neck motion are developed by a view planner that minimizes occlusions between the manipulator and



(a) Three valid minimum-time trajectories with unique terminal wrist poses that intersect with the door handle



(b) The terminal state of the freespace maneuver for the minimum-time trajectory with the maximum predicted information gain



(c) The terminal state of the interactive maneuver for the minimum-time trajectory with the maximum predicted information gain

Figure 4.7: Motion planning for the *next best touch* is applied to the DARPA ARM-S robot. A set of trajectory sequences is generated and sorted to efficiently select a motion that will result in the maximum expected information gain of an object. Blue spheres indicate terminal and intermediate wrist poses. Red lines indicate two suboptimal information gain trajectories. Green line indicates the highest information gain trajectory.

the sensor head. An example of an entire sequence is shown in Figure 4.6. A family of trajectories are generated between each neighboring constraint pair by sampling the intermediate and terminal state space of redundant joint angles. A trajectory sequence is a set of trajectories that satisfy all of the constraints in a constraint sequence. The set of all trajectory sequences that satisfy constraint sequences in the constraint sequence set is called the trajectory sequence set.

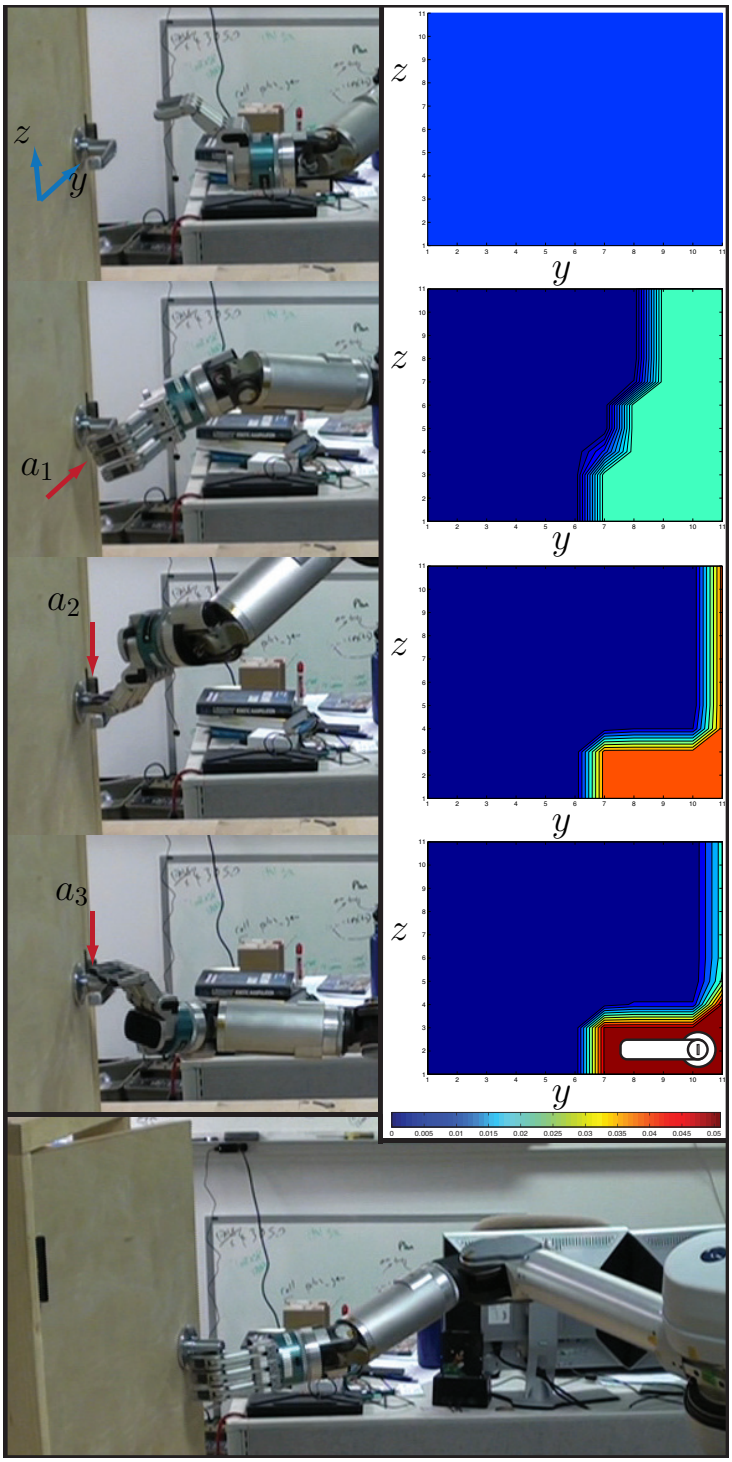


Figure 4.8: The evolution of belief distribution during a task to localize a door and door handle (Left column shows snapshots from the sequence of touching actions. Right column depicts the belief after each touching action. The belief map was marginalized over x for clarity. Regions of dark blue imply low belief and regions of dark red imply high belief.). The initial distribution is chosen as uniform. The first action touches the side of the door handle, and the next two actions touch the door handle from the top at different locations. At this point, the uncertainty is less than a small threshold, and the door is then opened.

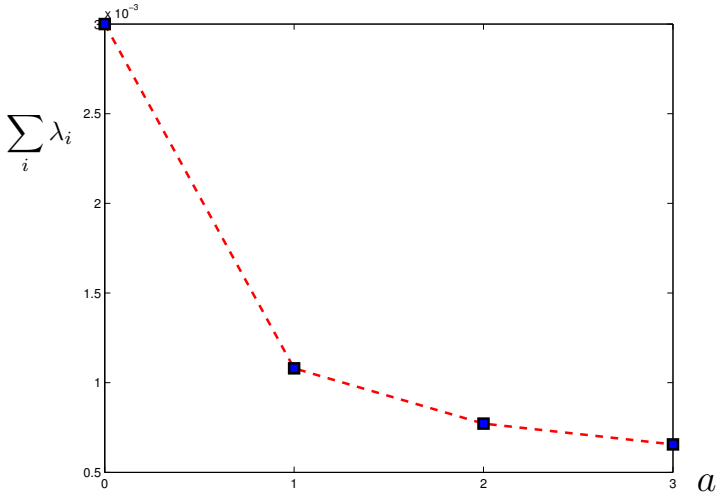


Figure 4.9: The covariance evolution. The sum of the eigenvalues of the covariance matrix is plotted at each action update. The uncertainty decreases until the threshold is met at which point the door is then opened.

4.3.2 Experimental Results

The algorithm was tested on the DARPA ARM-S robot with the task of opening a door. The robot planner was initialized with a model of the door and door handle. Before the touching sequence began, the door and door handle were both detected with their poses initially registered using vision. Unbeknownst to the robot, the door was then displaced manually after the initial registration in order to add additional error. The prior distribution in this experiment was chosen as uniform to add additional uncertainty in the location of the door handle. Note that the method does not preclude the use of other prior uncertainty models relating to the vision system used. The robot's hand motion was planned to an action starting location near the initially estimated door handle location. From there, action constraints and subsequent candidate actions were generated based on acquired sensory data. These actions were pruned for collisions with other objects in the scene.

Figure 4.8 illustrates the sequence of touching actions and the corresponding updated beliefs. A series of three touching actions was determined by the algorithm. The first action attempts to touch the door handle from the side. The subsequent two touching actions were initiated from above the door handle but in different locations. During each action, the belief is continually updated with contact measurements. The evolution of the belief after each action is also shown in the right column of Figure 4.8. The belief was marginalized over the state x for plotting clarity. As the belief peaked, the uncertainty of the door handle location decreased below the set threshold, at which point the door is then opened. Figure 4.9 plots the decrease in uncertainty (the sum of the eigenvalues of the covariance matrix) at each touching action.

4.4 Next Best Touch (NBT) with Unknown Model

In addition to touch-based localization, model and model parameter estimation may be necessary. For example, from certain views the robot maybe incapable of determining whether a particular object might be of a particular subclass of similar objects. A simple example is a travel thermos that may or may not have a handle, as shown in figure 4.16. As such, without knowing if the object has a handle or if it is not possible to view the handle, it might be particularly beneficial to touch the object before a grasp is selected and executed in order to detect presence of the handle.

Instead of exploiting the relative entropy of the belief of the object state for the next best touch in object localization, we exploit the relative entropy of the joint belief of the object state and model parameter m or model class M. This concept is developed and demonstrated in the remainder of this section. First, a parametrized model class is considered, followed by the case of unknown model classes.

4.4.1 NBT with Unknown Model Parameter

First consider the problem of estimating key parameters in a family of objects, e.g. screwdrivers. Similar mesh models, $\mathcal{M}$, within a object model class, M, has been shown to be parametrizable using an Active Shape Model (ASM) developed by Cootes [25]. An example of a parametrizable object such as a flathead screwdriver is shown in figure 4.10, where the parameters are shown. The basic principle of ASM is the use of principle component analysis (PCA). A PCA decomposition of a model class is computed in the following manner. The mesh points, $\mathbf{v}_i$, of each model $\mathcal{M}$, called

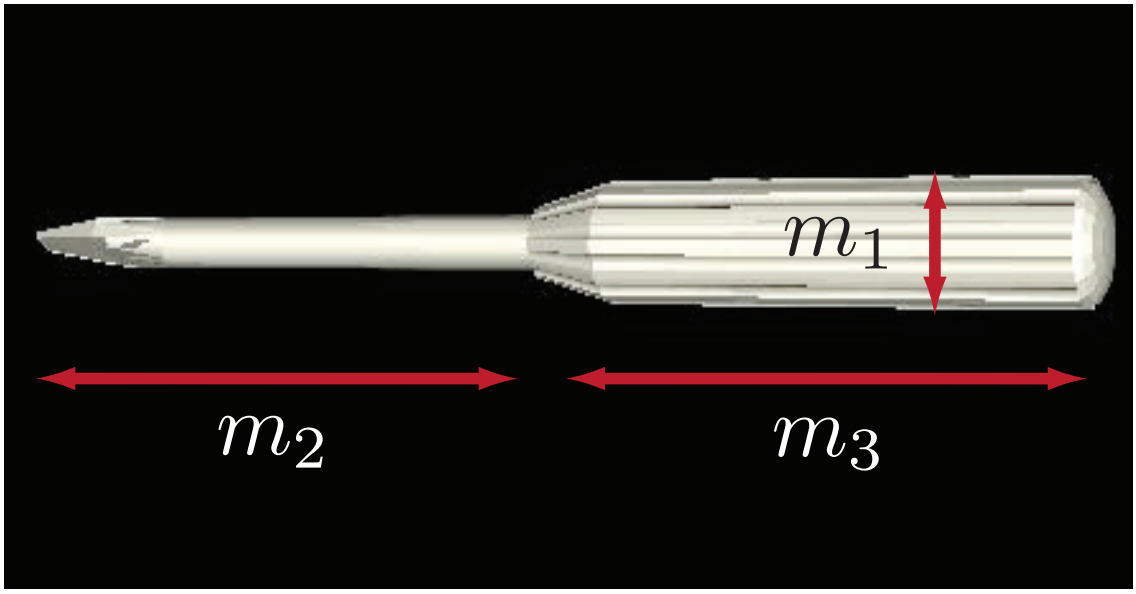


Figure 4.10: Similar objects that are parameterizable by multiple parameters.

shape vectors, are stacked to form a $3N$ vector, where each shape vector contains the same number of points N:

$$\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N]^T \ . \tag{4.16}$$

Assuming that we have K similar models, Procrustes analysis[2] is done to align the shape vectors. Next, a mean shape vector can be computed:

$$\overline{\mathbf{v}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{v}^{(k)} \, . \tag{4.17}$$

A semi-definite matrix representing the model covariance maybe computed a similar fashion:

$$P = \sum_{k=1}^{K} \left(\mathbf{v}^{(k)} - \overline{\mathbf{v}} \right) \left(\mathbf{v}^{(k)} - \overline{\mathbf{v}} \right)^T \ , \tag{4.18}$$

where $\mathbf{v}^{(k)}$ is the k^{th} model shape vector. A variability matrix V consists of the E eigenvectors of matrix P corresponding to the E largest eigenvalues. With this matrix, a shape of a new object in the class may be defined as:

$$\mathbf{v} = \overline{\mathbf{v}} + V \cdot \lambda \, , \tag{4.19}$$

where λ can be interpreted as the model parameters, m. These parameters may be unknown a priori and must be estimated. Therefore, a joint belief should be computed in order to estimate both the object state and the parameters. The joint estimate of object state and object parameters may be accomplished using a dual estimation technique or jointly by augmenting the state vector to include the model parameters.

The information gain of Equation 4.8 is modified to incorporate a joint belief, and an analogous derivation to Section 4.1.2 is carried out to develop an appropriate expression for the expected gain in information due to a candidate action, a_i, at time $t + 1$:

[2]Procrustes analysis is a statistical shape analysis used to analyze the distribution of a set of shapes.

$$
\begin{aligned}
&E_{\hat{z}}[IG(\hat{z}, a_i)] \\
&\quad = \int_{\hat{z}} p(\hat{z}) \int_x p(X_{t+1}, m | z_{1:t}, \hat{z}, u_{1:t}, a_i) \log \frac{p(X_{t+1}, m | z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t, m | z_{1:t}, u_{1:t})} \\
&\quad = \int_{\hat{z}} p(\hat{z}) \int_x \frac{p(\hat{z} | X_{t+1}, m)\, p(X_{t+1}, m | z_{1:t}, u_{1:t}, a_i)}{p(\hat{z})} \log \frac{p(X_{t+1}, m | z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t, m | z_{1:t}, u_{1:t})} \\
&\quad = \int_{\hat{z}} \int_x p(\hat{z}) \frac{p(\hat{z} | X_{t+1}, m)\, p(X_{t+1}, m | z_{1:t}, u_{1:t}, a_i)}{p(\hat{z})} \log \frac{p(X_{t+1}, m | z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t, m | z_{1:t}, u_{1:t})} \\
&\quad = \int_{\hat{z}} \int_x p(\hat{z} | X_{t+1}, m)\, p(X_{t+1}, m | z_{1:t}, u_{1:t}, a_i) \log \frac{p(X_{t+1}, m | z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t, m | z_{1:t}, u_{1:t})} \,.
\end{aligned}
\tag{4.20}
$$

As before, the end of the action is not known and the expectation must be taken over a distribution of stopping times:

$$
\begin{aligned}
&E_{\tau, \hat{z}} \left[IG(\hat{z}, \tau, a_i) \right] \\
&\quad = \sum_{\tau} p(\tau) \int_{\hat{z}} \int_x p(\hat{z}, m | X_{t+\tau})\, p(X_{t+\tau}, m | z_{1:t}, u_{1:t}, a_i) \log \frac{p(X_{t+\tau}, m | z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t, m | z_{1:t}, u_{1:t})} \,.
\end{aligned}
\tag{4.21}
$$

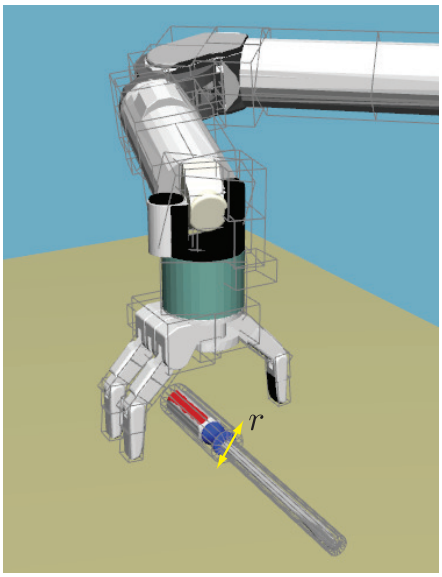
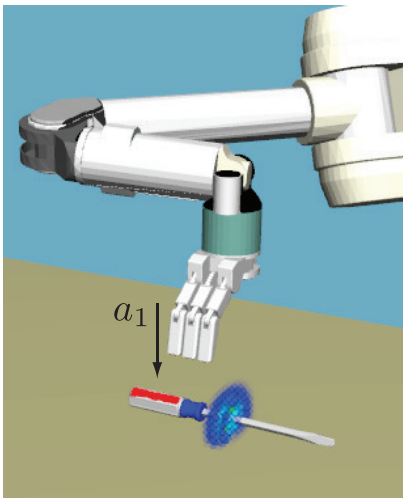


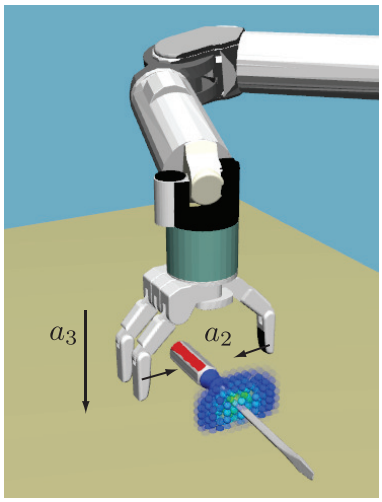
Figure 4.11: Primitive collision objects (rectangles, cubes, and cylinders) on robot and on object. Note the radius of the primitive cylinder on the screwdriver.

4.4.2 Simulation Results with Unknown Model Parameter

One task of the DARPA ARM-s program was to grasp a novel screwdriver that the robot had not seen before. As a result, knowledge of one of the key parameters, the handle radius, would have helped for accurate grasping of the screwdriver while it lies on the table. The visual system could not accurately ascertain this radius at the typical distances at which the screwdriver was located



(a) Screwdriver touching action a_1 — descending down in z



(b) Table touching action a_3 and pinching action a_2

Figure 4.12: Three feasible candidate actions to determine both the location and model parameter (screwdriver handle radius). The blue cloud represents the discrete belief of the screwdriver position and the handle radius of the screwdriver prior to any touching actions.

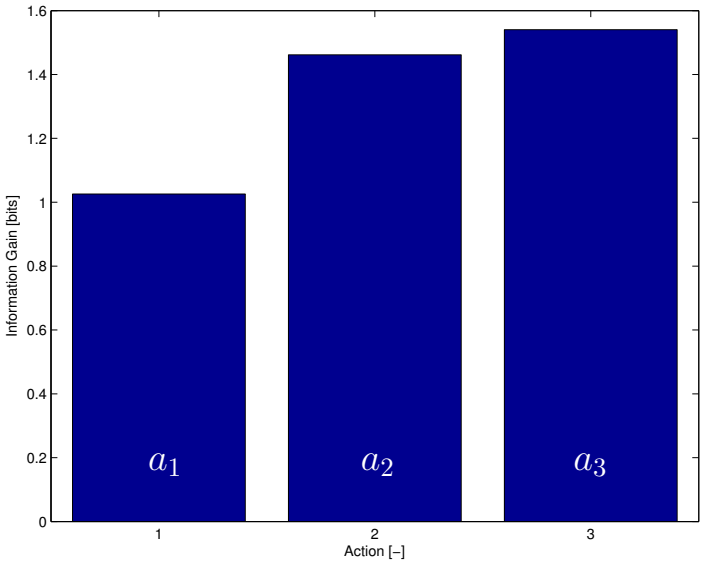
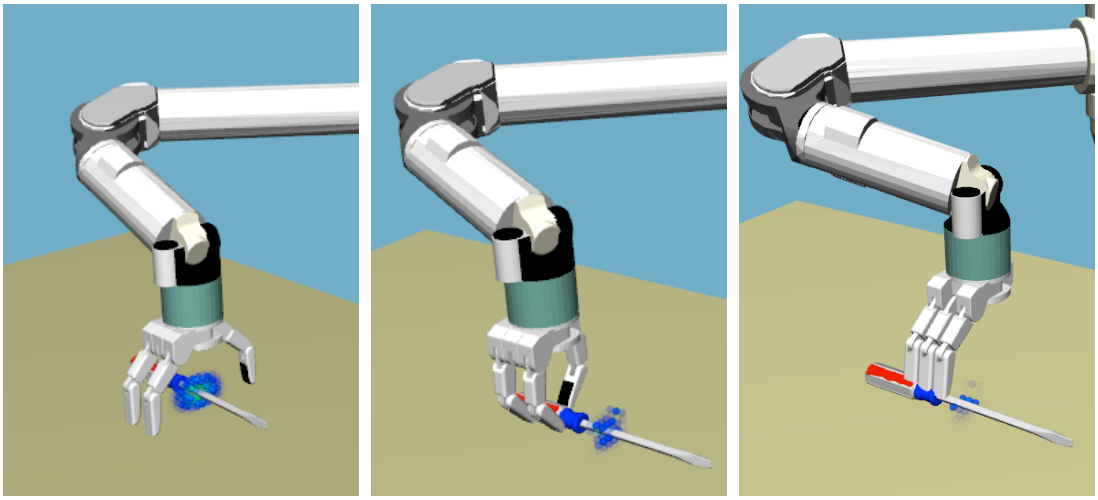


Figure 4.13: Information gain values for actions 1,2, and 3

relative to the cameras. As a result, this is an excellent example for touch-based parameter and state estimation.

As mentioned in Section 4.4.1, one can learn the parameters of the mesh using active shape models. In the case of a simple screwdriver example, one obvious parameter is the radius of the handle. For the calculation of collision detection, one can use a mesh model, or with simpler objects, such as the screwdriver, the objects can be decomposed into shape primitives. These primitives provide a fast and analytic way to compute intersection between objects. Figure 4.11 illustrates the screwdriver decomposed as cylinders, which can be parametrized by their radii.



(a) Belief after action a_3 (b) Belief after action a_2 (c) Belief after action a_1

Figure 4.14: Belief evolution after 3 consecutive touchings.

Figure 4.12 depicts 3 feasible candidates actions for the NBT with unknown model parameter, as well as initial uncertainty used in this simulation. The first action, a_1, is a straight down action in $z - direction$ to touch the handle, with fingertips. The second, a_2, is a pinching action (or action primitive) that is used on symmetric objects, The fingers close evenly until contact is made. The third action, a_3, is a table or surface touching action since the exact height of the table is equally uncertain. Touching the table is appropriate in this case as the relative positions of the table and screwdriver are constrained via with a planar constraint. The blue point cloud in Figure 4.12 surrounding the screwdriver depicts the discrete belief of the screwdriver position and handle radius parameter, with blue depicting low probability and red depicting high probability.

Figure 4.14 illustrates the belief distribution after each of the three candidate actions along the proposed action directions. Each panel shows the decrease of uncertainty after every positive measurement (i.e. contact is sensed). The simulation was able to infer the correct location (x: 0.74 m y: -0.35 m z: 0.49 m) and correct parameter (radius: 0.01 m) of the model screwdriver after an initial wrong x and z location (x: 0.75 m y: -0.35 m z: 0.48 m) and parameter values (radius: 0.02 m).

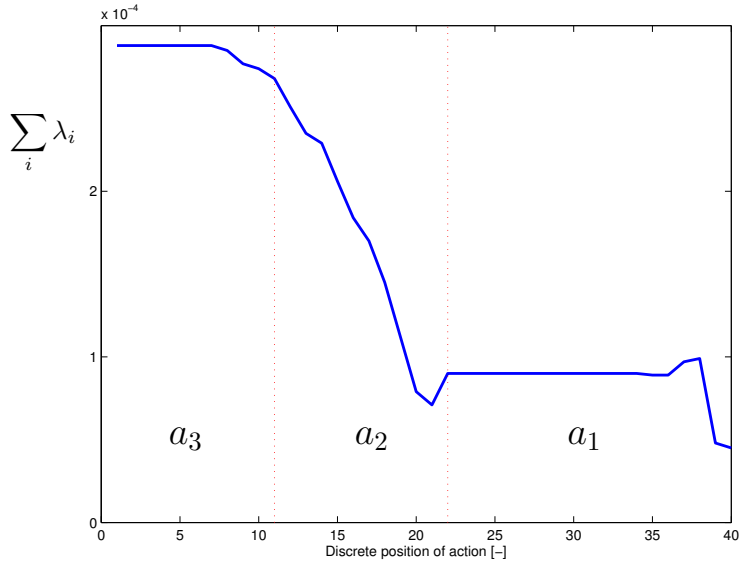


Figure 4.15: Covariance evolution of screwdriver belief after each action. The dotted lines indicate the start/end (boundary) of an action. Each discrete position in the plot is a waypoint (discretized motion) of the particular action within the boundary.

Figure 4.15 illustrates the decrease of uncertainty (calculated as the trace of the covariance matrix), after each discretized movement of the hand along each best action a_i. This shows the decrease of uncertainty after every negative measurement and positive measurement. At each red (dotted) line there was a positive measurement (contact occurred). After each red line, the next best action is computed and simulated. This figure shows considerable decrease of uncertainty of both the location and the parameter of the model.

Figure 4.16: Two similar objects are shown except one object has a feature only discernible from certain viewpoints.

4.4.3 NBT with Unknown Model Class

Now consider the problem of determining the category of an object (i.e. screwdriver or pipe, mug or glass). This is particularly useful if the robot has a limited view of an object or does not have the visual precision to discern the exact object class.

As before in Section 4.4.1, a joint probability must be computed consisting of the state X and the model class M, $p(X, M|z_{1:t}, u_{1:t})$. The joint expected information gain due to hypothetical action a_i, may then be computed as:

$$\begin{aligned} &E_\tau \left[IG(\tau, a_i)\right] \\ &= \sum_\tau p(\tau) \int_{X_{t+\tau}} \sum_M p(X_{t+\tau}, M|z_{1:t}, \hat{z}, u_{1:t}, a_i) \log \frac{p(X_{t+\tau}, M|z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t, M|z_{1:t}, u_{1:t})} \,. \end{aligned} \tag{4.22}$$

The joint belief may be broken into two probabilities using the product rule. The first is the state belief distribution over the object, and the second is the probability of the model.

$$p(X_{t+\tau}, M|z_{1:t}, \hat{z}, u_{1:t}, a_i) = p(X_{t+\tau}|M, z_{1:t}, \hat{z}, u_{1:t}, a_i)\ p(M|z_{1:t}, \hat{z}, u_{1:t}, a_i) \,. \tag{4.23}$$

The first term is simply the belief of the object state X under hypothetical model M. The second term is the model probability and can be found using Bayes' rule:

$$p(M|z_{1:t}, \hat{z}, u_{1:t}, a_i) = \frac{p(\hat{z}|M, z_{1:t}, u_{1:t}, a_i)\, p(M|z_{1:t}, u_{1:t}, a_i)}{\sum_M p(\hat{z}|M, z_{1:t}, u_{1:t}, a_i)\, p(M|z_{1:t}, u_{1:t}, a_i)} \,. \tag{4.24}$$

The first term in Equation 4.24 is the data likelihood and is the normalizer in the measurement update Equation 2.10 in the Bayes' filter, but under model class M. The data likelihood may be

found by marginalizing over the state X_t given the model M:

$$p(\hat{z}|M, z_{1:t}, u_{1:t}, a_i) = \int_{X_{t+\tau}} p(\hat{z}|X_{t+\tau}, M, z_{1:t}, u_{1:t}, a_i) p(X_{t+\tau}|M, z_{1:t}, u_{1:t}, a_i)\, dX_{t+\tau}\,. \tag{4.25}$$

The first term of Equation 4.25 is simply the measurement model likelihood by making the Markov assumption that past and future data is independent if the state is known. The second term is simply the intermediate probability after the prediction step of the Bayes' filter.

The second term in Equation 4.24 is the model probability from the previous time step and hence can be computed recursively. Lastly, the denominator of Equation 4.24 is a normalizer (abbreviated below by $p(z)$) and need not be computed explicitly for updating the model probability.

The expected information gain Equation 4.22 can be simplified into a form which can be easily computed. The joint belief in Equation 4.22 can be substituted with Equation 4.23 to yield:

$$\begin{aligned} E_\tau\left[IG(\tau, a_i)\right] = \sum_\tau p(\tau) \int_{X_{t+\tau}} \sum_M & \; p(X_{t+\tau}|M, z_{1:t}, \hat{z}, u_{1:t}, a_i)\; p(M|z_{1:t}, \hat{z}, u_{1:t}, a_i) \\ & \log \frac{p(X_{t+\tau}|M, z_{1:t}, \hat{z}, u_{1:t}, a_i)\; p(M|z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t, M|z_{1:t}, u_{1:t})}\,. \end{aligned} \tag{4.26}$$

Further breaking down this information gain, the belief conditional on the model class (the first term in Equation 4.23) can be substituted for the measurement update equation that is also conditional on the model class:

$$\begin{aligned} E_\tau\left[IG(\tau, a_i)\right] = \sum_\tau p(\tau) \int_{X_{t+\tau}} \sum_M & \; \frac{p(\hat{z}|X_{t+\tau}, M)\, p(X_{t+\tau}|M, z_{1:t}, u_{1:t}, a_i)}{p(\hat{z}|M)}\, p(M|z_{1:t}, \hat{z}, u_{1:t}, a_i) \\ & \log \frac{p(\hat{z}|X_{t+\tau}, M)\, p(X_{t+\tau}|M, z_{1:t}, u_{1:t}, a_i)}{p(\hat{z}|M)}\, \frac{p(M|z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t, M|z_{1:t}, u_{1:t})}\,. \end{aligned} \tag{4.27}$$

Similarly, the measurement z_τ is not known, it must be integrated out in the expectation of the information gain:

$$\begin{aligned} E_{\tau,\hat{z}}\left[IG(\hat{z}, \tau, a_i)\right] = \sum_\tau p(\tau) \int_{X_{t+\tau}} \sum_M \int_{\hat{z}} p(\hat{z}) & \frac{p(\hat{z}|X_{t+\tau}, M)\, p(X_{t+\tau}|M, z_{1:t}, u_{1:t}, a_i)}{p(\hat{z}|M)}\, p(M|z_{1:t}, \hat{z}, u_{1:t}, a_i) \\ & \log \frac{p(\hat{z}|X_{t+\tau}, M)\, p(X_{t+\tau}|M, z_{1:t}, u_{1:t}, a_i)}{p(\hat{z}|M)}\, \frac{p(M|z_{1:t}, \hat{z}, u_{1:t}, a_i)}{p(X_t, M|z_{1:t}, u_{1:t})}\,. \end{aligned} \tag{4.28}$$

To further simplify the above, the model probability may be substituted using Bayes' rule and conditioning variables that do not matter on the probability expressions ($z_{1:t}$ and $u_{1:t}$) are omitted

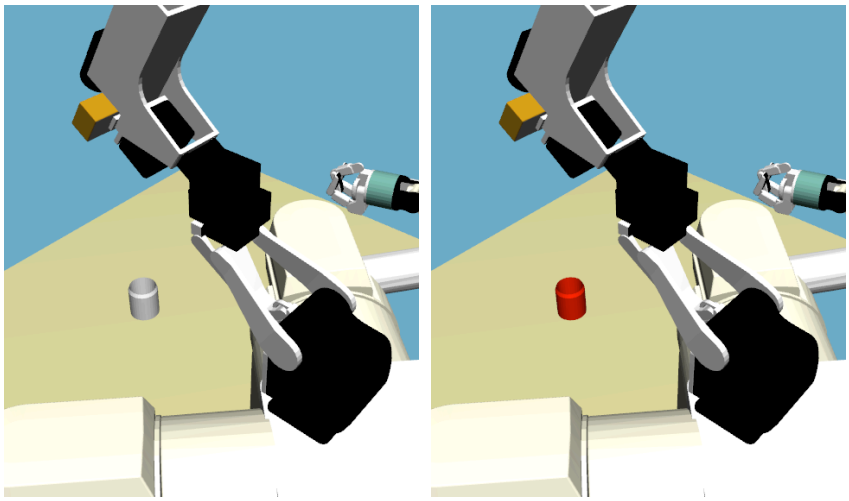
for clarity:

$$E_{\tau,\hat{z}}\left[IG(\hat{z},\tau,a_i)\right] = \int_{\tau} p(\tau) \int_{X_{t+\tau}} \sum_{M} \int_{\hat{z}} p(\hat{z}) \frac{p(\hat{z}|X_{t+\tau},M)\, p(X_{t+\tau}|M,z_{1:t},u_{1:t},a_i)}{p(\hat{z}|M)} \frac{p(\hat{z}|M)\, p(M)}{p(\hat{z})} \log \frac{p(\hat{z}|X_{t+\tau},M)\, p(X_{t+\tau}|M,z_{1:t},u_{1:t},a_i)}{p(\hat{z}|M)\, p(X_t,M|z_{1:t},u_{1:t})} \frac{p(\hat{z}|M)\, p(M)}{p(\hat{z})}\,. \tag{4.29}$$

Canceling out terms in the numerator and denominator, the final expected information gain is thus:

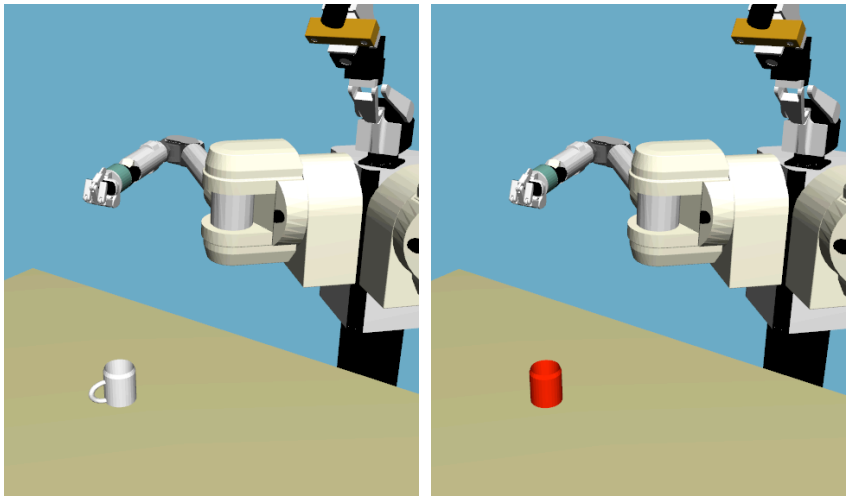
$$E_{\tau,\hat{z}}\left[IG(\hat{z},\tau,a_i)\right] = \int_{\tau} p(\tau) \int_{X_{t+\tau}} \sum_{M} \int_{\hat{z}} p(\hat{z}|X_{t+\tau},M)\, p(X_{t+\tau}|M,z_{1:t},u_{1:t},a_i)\, p(M) \log \frac{p(\hat{z}|X_{t+\tau},M)\, p(X_{t+\tau}|M,z_{1:t},u_{1:t},a_i)}{p(X_t,M|z_{1:t},u_{1:t})} \frac{p(M)}{p(\hat{z})}\,. \tag{4.30}$$

This equation can be computed using the measurement model $p(\hat{z}|X_{t+\tau},M)$, which may be the binary contact model given in Equation 4.12. The discrete prior model probability is denoted by $p(M)$ and the prior joint probability conditioned with the model is denoted by $p(X_t,M|z_{1:t},u_{1:t})$. The intermediate probability is also conditioned on the model M, denoted by $p(X_{t+\tau}|M,z_{1:t},u_{1:t},a_i)$ and $p(\hat{z})$ is the normalizer in the model probability calculation of Equation 4.24.



(a) Indiscernible view of mug

(b) Indiscernible view of cup



(c) Distinguishable view of mug

(d) Distinguishable view of cup

Figure 4.17: Particular views from the robot of the mug and cup in which both are visually indistinguishable

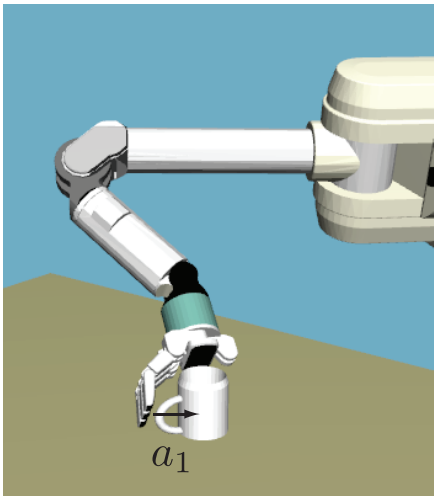
4.4.4 Simulation Results with Unknown Model Class

One challenging issue is to be able to discern objects when they are quite similar. Even when similar objects are in full view, disambiguating features may be too noisy, or only in partial view, to be used to discern the particular object. For example, while grasping objects within a bag, vision cannot be used. Therefore, it is sometimes valuable to use touch for identifying a particular object.

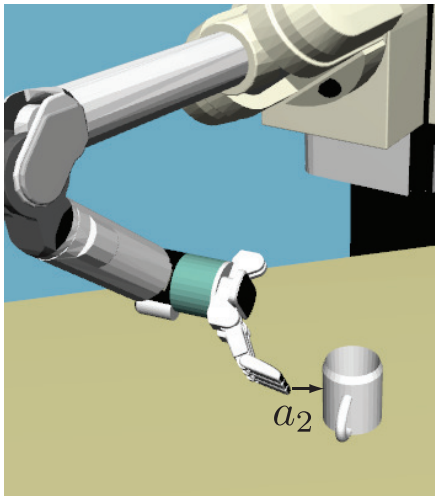
A simulation experiment was designed to demonstrate how this version of NBT might be used to distinguish between a mug and cup in the case they were placed at a pose in which the distinguishable feature (the handle) was not in view. As a result, in order to disambiguate between the two possible object, the robot must interact and touch the object.

The expected information gain calculation of Equation 4.30 is used to infer which is the best

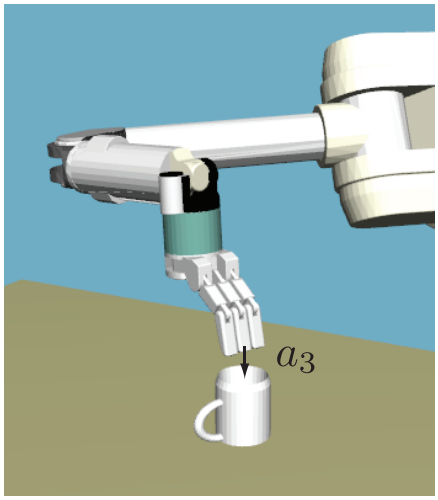
action to disambiguate between the two objects. For this demonstration, the prior model probability is assigned to be equal amongst both models: $P(M) = 0.5$. The geometry of this simulation is shown in Figure 4.17 in which the handle of the mug was out of view, thereby making the cup and mug visually indistinguishable by shape alone, as the object models share the same cylindrical shape by design.



(a) Action 1 touching the area near handle



(b) Action 2 touching the side of the mug/cup



(c) Action 3 touching the top of the mug/cup

Figure 4.18: Actions generated for next best touch with unknown model class with mug and cup

Actions are automatically generated based on the objects normals and surfaces and the choice of surfaces of the fingers and hand of the manipulator as described in Section 4.1.1. The three actions that satisfied all of the constraints are shown in Figure 4.18 and can be qualitatively described as (1) touching the handle, (2) touching the side of the mug/cup, and (3) touching the top of the mug/cup.

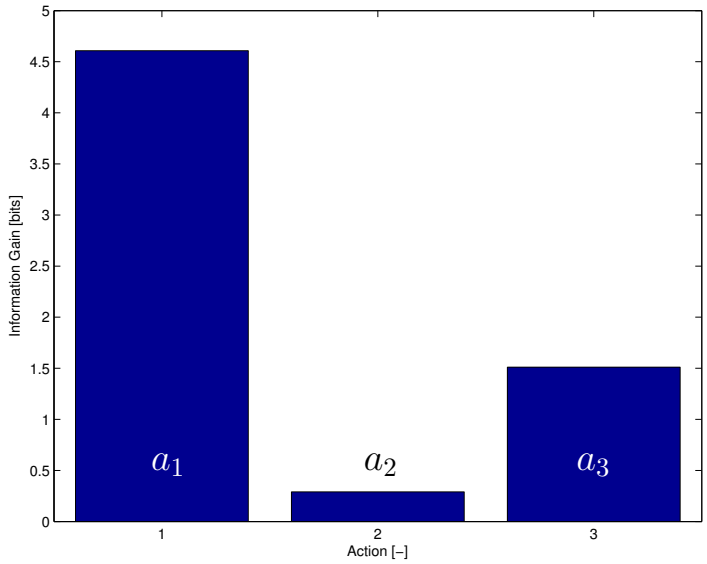


Figure 4.19: Information gain values for actions 1,2, and 3.

For each candidate action, the expected information gain in Equation 4.30 was determined using the binary contact model in Equation 4.12 and the model probability distribution was chosen as uniform (each model is equally likely). Given the actions in Figure 4.18, action a_1, which touches the handle, provides the most information gain as shown in Figure 4.19. While actions a_2 and a_3 do not touch the discriminating feature, there is some information gain due to the uncertainty in pose, since the information gain calculation is based on the joint belief between the model class M and object pose X.

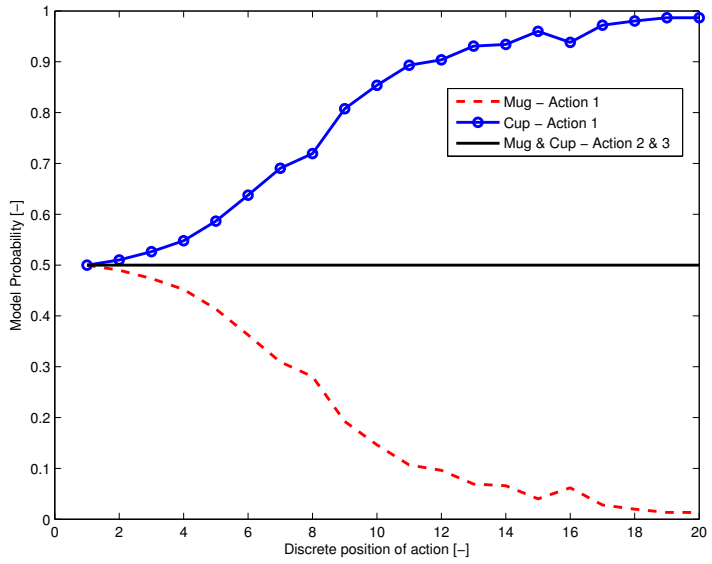


Figure 4.20: Model probability evolution for mug and cup for actions 1, 2, 3 given NULL contact measurements

As the information gain is computed while the robot moves along a particular action direction, it is important to note that in order for the action to continue, it assumed that the measurements are NULL contacts as described in Section 4.1. If there were to be a positive contact, then the action would end. With this in mind, Figure 4.20 shows the evolution of the model probability during each action 1, 2 and 3. Since there are no positive contact, the model probability for the cup gains favor in action 1. This is due to the fact that action 1 interacts with the handle and if there are no contact, then it is more likely that the object is the cup since the cup has no handle. The contrary is also true, in that since there is contact, it becomes less likely that the object is the mug. Finally note that actions 2 and 3 do not interact with the discernible feature (handle) and hence will not contribute to the model probability. For these actions, the model probability for each object remains at 0.5.

Chapter 5

Conclusion

5.1 Summary of Thesis Contributions

The are two primary contributions of this thesis, both of which relate to grasping and manipulation problems. The first contribution is an estimation framework for aspect of grasping and manipulation. The second relates to object interaction and involves action selection. This action inference theory, termed the *next best touch* uses kinesthetic inputs and information theory to choose the best touching action to gain the most information about the object.

The estimation framework in Chapter 3 combines multiple visual cue and multiple sensors provides more accurate results. There are several contributions from this estimation framework and be thought of in two parts: manipulator tracking and object tracking. The manipulator tracking involves the use and fusion of multiple visual cues from different sensors. The advantages of each sensor are leveraged. A stereo camera system is excellent at finding depth to individual features through point-wise/sparse stereo matching. Technologies such as 3D cameras, like the Kinect^{TM}or Xtion^{TM}provide remarkable accurate point clouds with little noise. Monocular cameras which are high resolution can discern small features that stereo cameras or RGB cameras from 3D cameras may not pick up or where depth is not important. A contribution of this thesis is fusing appearance features from stereo cameras, shape measurements from 3D cameras and contour/silhouette measurements from a high-resolution monocular camera to provide robust and accurate manipulator tracking. In addition, such a framework can be used as an online-calibration technique

The contribution to object tracking involves a similar fusion of the above cues and modalities: features, shape and silhouette. My contribution also fuses kinesthetic sensors such as tactile and force-torque sensors, since with grasping physical interaction is inherent. The tracking of both manipulator and object using a variety of sensors and cues is a novel contribution.

Another contribution is the implementation of a hybrid estimation framework for continuous states, such as the robot and object state but also discrete states, such as finger contact modes. These finger contact modes are useful for inferring how each finger is contacting the grasped object's

surface and valuable for grasping metrics. Theory and results are presented using sensor fusion and a static multiple model estimator.

In addition to state estimation, parameter estimation is also explored in the context of grasping and manipulation. Results in determining center of mass of a grasped object are presented using a dual estimation filtering framework. Another contribution is a filtering framework for dual-arm manipulation that utilizes nonlinear constraints. Two methods are compared and results are presented from a wheel changing task.

The last contribution is a new paradigm called *next best touch* which is an action selection method for object interaction through the use of touch. This method leverages information theory and selects the next best action to localize the object by touching using an information gain metric. Simulation and experiment are presented with the problem of locating a door handle on a door.

Another extension to this paradigm is that in addition to object localization, model parameter determination also becomes the goal. Many objects within an object class share the same rough shape. These similar object models can be parametrized to an extent. Simulation is presented to show that the method performs by localizing and determining a parameter of an unknown screwdriver.

Lastly, new theory is presented to determine the best action to touch in order to best localize and discern which class a particular object belongs to.

5.2 Opportunities for Future Work

The estimation frameworks used in this thesis employed various type of Kalman filters. A new direction might be determine whether a particle filter might provide added performance or the flexibility to employ more complicated measurement models. The moment of grasping of an object is complicated dynamic event and the flexibility of the particle filter formulation may allow for a more intelligent reasoning.

The methods used in manipulator tracking, particularly articulated iterated closest point and silhouette rendering are methods used independently and are unaware of the convergence of each other and other objects in the environment such as another arm or a grasped object. This can lead to mismatchings in all methods and ultimately poor performance. To prevent this and to possibly improve performance , the Iterated Sigma Point filter may be used to fuse these cues iteratively, rather than after convergence of the silhouette and aicp methods. This fusion may speed up convergence and improve the rate of the filter.

The tactile measurement models employed in this thesis are simple but effective models. However, there is more information to be exploited from these tactile sensors such center of pressure, total force, and features. Features such as edges and sharp points may provide increased performance

since they are more discriminative that the surface of models (vertices of the object mesh) employed in this thesis.

For parameter estimation, it was shown that for center of mass estimation that multiple object orientations were needed for accurate estimation. Future work will include placing the object in multiple poses in order to fully ascertain the center of mass.

There are several future directions for the *next best touch*. The first is naturally to extend the model class determination and demonstrating it on a physical robot. The second is to address objects that are articulated and to use information theory to select actions that both localize and learn the model parameters of the articulation.

Bibliography

[1] Peter K. Allen, Aleksandar Timcenko, Billibon Yoshimi, and Paul Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. on Robotics and Automation*, 1991.

[2] Danica Kragic, Andrew Miller, and Peter Allen. Real-time tracking meets online grasp planning. In *Proc. of IEEE ICRA*, 2001.

[3] Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. Robotic Grasping of Novel Objects using Vision. *IJRR*, 2008.

[4] Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. Learning 3-d object orientation from images. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009.

[5] Michael Krainin, Peter Henry, XXiaofeng Ren, and Dieter Fox. Manipulator and object tracking for in hand model acquisition. *Int. Conf. on Robotics & Automation (ICRA)*, 2010.

[6] Markus Ulrich and Christian Wiedemann. Cad-based recognition of 3d objects in monocular images. *ICRA*, 2009.

[7] Peter K. Allen, Andrew T. Miller, Paul Y. Oh, and Brian S. Leibowitz. Using tactile and visual sensing with a robotic hand. In *In Proc. of the 1997 IEEE Int. Conf. on Robotics and Automation*, pages 677–681, 1997.

[8] Peter K. Allen, Andrew T. Miller, Paul Y. Oh, and Brian S. Leibowitz. Integration of vision, force and tactile sensing for grasping. *Int. J. Intelligent Machines*, 1999.

[9] Angel Pobil Mario Prats, Pedro Sanz. Reliable nonprehensile door opening through the combination of vision, tactile and force feedback. *Autonomous Robots*, Jan. 2010.

[10] Andrew J. Schmid, Nicholas Gorges, Dirk Goger, and Heinz Worn. Opening a door with a humanoid robot using multi-sensory tactile feedback. In *Int. Conf. on Robotics and Automation (ICRA)*, 2008.

[11] Anya Petrovskaya, Oussama Khatib, Sebastian Thrun, and Andrew Y. Ng. Touch based perception for object manipulation. In *Proc. of Robotics: Science and Systems*, Atlanta, GA, USA, jun 2007.

[12] Klaas Gadeyne. Markov techniques for object localization with force-controlled robots. *International Conference on Advanced Robotics*, 2004.

[13] Craig Corcoran and Robert Platt. A measurement model for tracking hand-object state during dexterous manipulation. In *International Conference on Robotics and Automation (ICRA)*, Jan 2010.

[14] Lars Petersson, Magnus Egerstedt, and Henrik I. Christensen. A hybrid control architecture for mobile manipulation. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 1999.

[15] Yingjie Yin, Zhiwei Luo, M. Svinin, and S. Hosoe. Hybrid control of multi-fingered robot hand for dexterous manipulation. volume 4, oct. 2003.

[16] Tsuneo Yoshikawa and Xin-Zhi Zheng. Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object. *Int. J. of Robotics Research*, 1993.

[17] Masaru Uchiyama, Naotoshi Iwasawa, and Kyojiro Hakomori. Hybrid position/force control for coordination of a two-arm robot. In *Int. Conf. on Robotics and Automation (ICRA)*, 1987.

[18] Lars Blackmore and Steve Block. Control and estimation for cooperative manipulator tasks. MIT CSAIL Technical Report, 2005.

[19] Klaas Gadeyne, T. Lefebvre, and Herman Bruyninckx. Bayesian hybrid model-state estimation applied to simultaneous contact formation recognition and geometrical parameter estimation. *IJRR*, 2005.

[20] Yaakov. Bar-Shalom, X. Rong. Li, and Thiagalingam. Kirubarajan. *Estimation with applications to tracking and navigation*. J. Wiley & Sons, NY., 2001.

[21] A.C. Murillo, J. Kosecká, J.J. Guerrero, and C. Sagüés. Visual door detection integrating appearance and shape cues. *RAS*, 2008.

[22] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002.

[23] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV*, 1998.

[24] T.F Cootes, G.J Edwards, and C.J Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685, 2001.

[25] T.F Cootes, C.J Taylor, D.H Cooper, and J Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[26] Yizong Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence*, Jan 1995.

[27] Robert Collins. Mean-shift blob tracking through scale space. *Computer Vision and Pattern Recognition*, Jan 2003.

[28] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564 – 577, may 2003.

[29] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision — ECCV 2002*, volume 2350 of *Lecture Notes in Computer Science*, pages 661–675. Springer Berlin / Heidelberg, 2002.

[30] Schmitt Francis Esteban Carlos H. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 2004.

[31] Kampe Martin Tosovic Srdan, Sablatnig Robert. On combining shape from silhouette and shape from structured light. In *Proc. of the 7th Computer Vision Winter Workshop*, 2002.

[32] Quentin Delamarre and Olivier Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *Proc. of the Int. Conf. on Computer Vision (ICCV)*, 1999.

[33] Howe Nicholas R. Silhouette lookup for automatic pose tracking. In *Computer Vision and Pattern Recognition Workshop*, 2004.

[34] David Camarillo, Kevin .E. Loewke, Christopher R. Carlson, and J. Kenneth Salisbury. Vision based 3-d shape sensing of flexible manipulators. In *ICRA*, 2008.

[35] Paul Hebert, Nicolas Hudson, Jeremy Ma, Thomas Howard, Thomas Fuchs, Max Bajracharya, and Joel Burdick. Combined shape, appearance and silhouette for simultaneous manipulator and object tracking. In *Int. Conf. on Robotics and Automation (ICRA)*, may 2012.

[36] Erik .B. Sudderth, Michael I. Mandel, William T. Freeman, and Alan S. Willsky. Visual hand tracking using nonparametric belief propagation. In *Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

[37] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003.

[38] Irvin Rock and Jack Victor. Vision and touch: An experimentally created conflict between the two senses. *Science*, 1964.

[39] Alvaro Collet, Dmitry Berenson, Siddhartha S. Srinivasa, and Dave Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 48 –55, may 2009.

[40] Henning Hamer, Konrad Schindler, Esther Koller-Meier, and Luc Van Gool. Tracking a hand manipulating an object. *Computer Vision, 2009 IEEE 12th International Conference on*, 2009.

[41] Paul Hebert, Nicolas Hudson, Jeremy Ma, and Joel Burdick. Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location. In *International Conference on Robotics and Automation (ICRA)*, May 2011.

[42] Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. Manipulator and object tracking for in-hand 3d object modeling. *The International Journal of Robotics Research*, 2011.

[43] Hanqi Zhuang, Kuanchih Wang, and Z.S. Roth. Simultaneous calibration of a robot and a hand-mounted camera. *Robotics and Automation, IEEE Transactions on*, 11(5):649 –660, oct 1995.

[44] Sang De Ma. A self-calibration technique for active vision systems. *Robotics and Automation, IEEE Transactions on*, 12(1):114 –120, feb 1996.

[45] David J. Bennett, Davi Geiger, and John M. Hollerbach. Autonomous robot calibration for hand-eye coordination. *The International Journal of Robotics Research*, 10(5):550–559, 1991.

[46] Benjamin Mooring, Morris Driels, and Zvi Roth. *Fundamentals of Manipulator Calibration*. John Wiley & Sons, Inc., New York, NY, USA, 1991.

[47] W. Eric L. Grimson and Tomas Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *The International Journal of Robotics Research*, 3(3):3–35, 1984.

[48] Peter Allen and Ruzena Bajcsy. Object recognition using vision and touch. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, 1985.

[49] Nicolas Gorges, Stefan Gaa, and Heinz Worn. Object exploration with a humanoid robot using tactile and kinesthetic feedback. In *International Conference on Informatics in Control, Automation and Robotics*, 2008.

[50] Craig Corcoran and Robert Platt. Tracking object pose and shape during robot manipulation based on tactile information. In *International Conference on Robotics and Automation (ICRA)*, Jan 2010.

[51] Anya Petrovskaya, Oussama Khatib, Sebastian Thrun, and Andrew Y. Ng. Bayesian estimation for autonomous object manipulation based on tactile sensors. In *Robotics and Automation (ICRA)*, May 2006.

[52] Andrew .J. Davison. Active search for real-time vision. In *International Conference on Computer Vision (ICCV)*, October 2005.

[53] Tal Arbel and Frank P. Ferrie. Entropy-based gaze planning. In *Journal of Image and Vision Computing*, pages 779–786, 1999.

[54] Jeremy Ma and Joel Burdick. Dynamic sensor planning with stereo for model identification on a mobile platform. *International Conference on Robotics and Automation (ICRA)*, Jan 2010.

[55] Sebastian Thrun Dieter Fox, Wolfram Burgard. Active markov localization for mobile robots. *Journal of Robotics and Autonomous Systems*, Jan 1998.

[56] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[57] Kaijin Hsiao, Leslie Kaelbling, and Tomas Lozano-Perez. Task-driven tactile exploration. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[58] Alexander Schneider, Jurgen Sturm, Cyrill Stachniss, Marco Reisert, Hans Burkhardt, and Wolfram Burgard. Object identification with tactile sensors using bag-of-features. In *Int'l Conference on Intelligent Robots and Systems*, 2009.

[59] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Third International Conference on 3-D Digital Imaging and Modeling.*, 2001.

[60] Paul .J. Besl and Neil .D. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1992.

[61] Jorge Moré. The levenberg-marquardt algorithm: Implementation and theory. In G. Watson, editor, *Numerical Analysis*, volume 630 of *Lecture Notes in Mathematics*, pages 105–116. Springer Berlin / Heidelberg, 1978. 10.1007/BFb0067700.

[62] Roger Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation.*, pages 323 –344, 1987.

[63] Donald Gennery. Generalized camera calibration including fish-eye lenses. *IJCV*, 2006.

[64] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[65] Garry A. Einicke. *Smoothing, Filtering and Prediction - Estimating The Past, Present and Future*. InTech, 2012.

[66] Uhlmann Jeffrey K. Julier Simon J. A new extension of the kalman filter to nonlinear systems. In *Proceedings of the Int. Soc. for Optical Eng.*, 1997.

[67] Colin McManus and Timothy Barfoot. A serial approach to handling high-dimensional measurements in the sigma-point kalman filter. In *Proc. of Robotics: Science and Systems*, 2011.

[68] Max Bajracharya, Matthew DiCicco, Paul Backes, and Kevin Nickels. Visual end-effector position error compensation for planetary robotics. *Journal of Field Robotics*, 2007.

[69] Stefano Pellegrini, Konrad Schindler, and Daniele Nardi. A generalization of the icp algorithm for articulated bodies. In *BMVF*, 2008.

[70] Gunilla Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344 – 371, 1986.

[71] John A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[72] David G. Lowe. Object recognition from local scale-invariant features. *Computer Vision, IEEE International Conference on*, 2, 1999.

[73] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision – ECCV 2006*. 2006.

[74] Jeffrey S. Beis and David G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. *IEEE ICVPR*, 1997.

[75] G.M. Krishna and K. Rajanna. Tactile sensor based on piezoelectric resonance. *Sensors Journal, IEEE*, 4(5):691 – 697, oct. 2004.

[76] Makoto Shimojo, Akio Namiki, Masatoshi Ishikawa, Ryota Makino, and Kunihiko Mabuchi. A tactile sensor sheet using pressure conductive rubber with electrical-wires stitched method. *Sensors Journal, IEEE*, 4(5):589 – 596, oct. 2004.

[77] David Beebe, Denice Denton, Robert Radwin, and John Webster. A silicon-based tactile sensor for finger-mounted applications. *Biomedical Engineering, IEEE Transactions on*, 45(2):151 –159, feb. 1998.

[78] Robert S. Fearing and T.O. Binford. Using a cylindrical tactile sensor for determining curvature. *Robotics and Automation, IEEE Transactions on*, 7(6):806 –817, dec 1991.

[79] Bonnie L. Gray and Robert S. Fearing. A surface micromachined microtactile sensor array. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, pages 1 –6 vol.1, apr 1996.

[80] Jae S. Son, Eduardo A. Monteverde, and Robert D. Howe. A tactile sensor for localizing transient events in manipulation. In *ICRA*, pages 471–476, 1994.

[81] Eric A. Wan, Rudolph Van Der Merwe, and Alex T. Nelson. Dual estimation and the unscented transformation. In *in Neural Information Processing Systems*, pages 666–672. MIT Press, 2000.

[82] Simon J. Julier and Jeffrey J. LaViola. On kalman filtering with nonlinear equality constraints. *IEEE Transactions on Signal Processing.*, 2007.

[83] Nicolas Hudson, Thomas Howard, Jeremy Ma, Abhinandan Jain, Max Bajracharya, Steven Myint, Calvin Kuo, Larry Matthies, Paul Backes, Paul Hebert, Thomas Fuchs, and Joel Burdick. End-to-end dexterous manipulation with deliberate interactive estimation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2371 –2378, may 2012.