Robust Object Pose Estimation with Point Clouds from Vision and Touch

by

Gregory Izatt

B.S., California Institute of Technology (2014)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

© Massachusetts Institute of Technology 2017. All rights reserved.

Author .....
Department of Electrical Engineering and Computer Science
May 19, 2017

Certified by .....
Russ Tedrake
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Robust Object Pose Estimation with Point Clouds from Vision and Touch

by

Gregory Izatt

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2017, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

We present a study of object pose estimation performed with hybrid visuo-tactile sensing in mind. We propose that a tactile sensor can be treated as a source of dense local geometric information, and hence consider it to be a point cloud source analogous to an RGB-D camera. We incorporate the tactile geometric information directly into a conventional point-cloud-based articulated object tracker based on signed-distance functions. This tracker runs at 12 Hz using an online depth reconstruction algorithm for the GelSight tactile sensor and a modified second-order update for the tracking algorithm. The tracker provides robust pose estimates of small objects throughout manipulation, even when the objects are occluded by the robot's end effector. To address limitations in this tracker, we additionally present a formulation of the underlying point-cloud correspondence problem as a mixed-integer convex program, which we efficiently solve to optimality with an off-the-shelf branch and bound solver. We show that reasoning about object pose estimation in this way allows natural extension to point-to-mesh correspondence, multiple object estimation, and outlier rejection without losing the ability to obtain a globally optimal solution. We probe the extent to which rich problem-specific formulations typically tackled with unreliable nonlinear optimization can be rigorously treated in a global optimization framework.

Thesis Supervisor: Russ Tedrake
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I'm very happy to get to thank my advisor, Russ, for his brilliant guidance and support. In this thesis, I refer frequently to the intractability of global optimization. However, I suspect Russ to be personally immune to this, in his ability to pierce to the heart of any research problem and see the fundamentally fruitful directions to take, instead of the clutter of stopgap solutions and low-hanging fruit. Russ also deserves enormous credit for bringing together the people, robots, and countless resources that make this lab great. I'd also like to thank him for his patience as I've taken the time to explore some uniquely *MIT* side projects... and for keeping me on course when I have gotten lost.

I can't thank the MIT DARPA Robotics Challenge team and the Robot Locomotion Group enough for granting me such a unique and inspiring place to work, and for continuing to teach me so much. Every person in this lab is brilliant and sharp – and critically, they are wonderfully generous with those attributes. Thanks to everyone for many late nights on robots, hours of deep conversation, and insightful suggestions at group meetings.

Working with the MIT Hyperloop Team was an incredible pleasure. I am proud to have worked with such an amazingly skilled group of engineers, and honored to have been a part of the endeavor. Flying that pod was a dream of mine come true.

Even more thanks to my friends at Sidney Pacific. SP is truly home, and has given me a refuge during an intense couple of years. SPEC, officers, Heads of House, helpers, residents, and the tea party crowd – thanks for being wonderful friends and comrades in our many endeavors.

Thank you to innumerable other friends around MIT and Boston, in LA, and everywhere else – and my family. I eagerly await my next adventure with each of you – be it costume prop engineering, climbing, late-night Skype calls, flying, or an unexpected box of far too many chocolates.

Contents

Chapter 1

Introduction and Motivation

The ability to perceive and control objects through contact is a fundamental skill for any robot interacting with the world. Laser scanners, RGB-D cameras, and stereo vision have become the primary means for modern autonomous systems to sense their environment. Techniques and tools for detecting and localizing objects in the image and point cloud data from these sensors have become very mature. As a testament to the power of these methods, Team MIT relied almost exclusively on point cloud geometry from an onboard laser scanner to perform navigation and manipulation tasks during the DARPA Robotics Challenge Finals (Figure 1-1).

This reliance on a fundamentally visual modality becomes a curse during manipulation. During manipulation, it is guaranteed that the robot's manipulator will occlude some or all of the manipuland from vision. Fortunately, precisely when visual sensors fail, tactile sensors can fill in the gap by providing information from the area that is being occluded. Tactile sensors come in many forms, with most sensors specializing in binary contact discrimination or contact force estimation at a single point, or at multiple points across the sensor surface.

The algorithmic challenges facing object pose estimation from tactile sensing are diverse and depend on the tactile sensor modality. However, they share a common underlying difficulty in that the information from tactile sensors is sparse compared to vision sensors. This sparsity exists in both space – due to the limited surface area of a contact patch between sensor and object – and time – due to intermittent

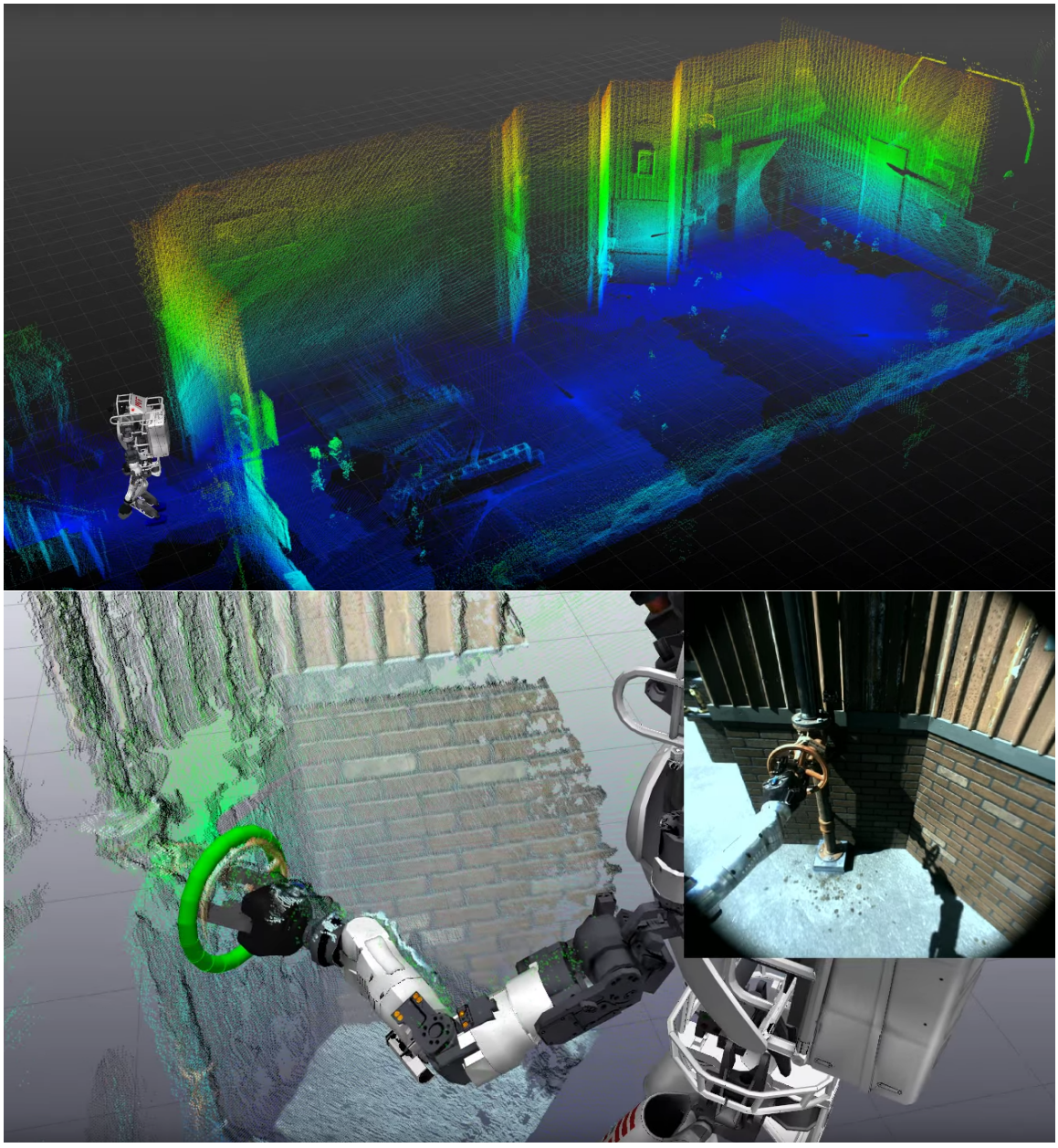

Figure 1-1: Illustrations of point cloud information gathered from the laser scanner on MIT's robot at the DARPA Robotics Challenge Finals in 2015. **Top:** Point cloud geometry of the indoor section of the challenge course. **Bottom:** Point cloud geometry from the valve task, with the RGB camera image, robot pose, and valve geometry overlaid. Point cloud information was critical for both navigating the course and avoiding obstacles, as well as for localizing objects for manipulation.

contact between sensor and object. This sparsity poses a significant challenge to the adaptation of state-of-the-art techniques from robotic perception to the problem of object pose estimation from vision *and* touch.

1.1 Problem Specification

We wish to address the broadest possible form of the pose estimation problem, such that we can address the unique problems that arise when considering tactile information in the most fundamental way.

We consider the *general object pose estimation problem* to be the problem of finding the best parameters of a model to explain the data available from the sensors. Given some sensor observations of the scene z, a model parameterization x, and a sensor model $p(z|x)$, we wish to find a pose estimate $\hat{x}$:

$$\hat{x} = \underset{x}{\operatorname{argmin}}\, p(z|x).$$

This problem is most frequently addressed with rigid body models and point cloud observations. In the single-object rigid body model, model parameters are a rotation $R \in SO(3)$ and a translation $T \in \mathbb{R}^3$, in which the model has well-defined surface which can be defined implicitly as the zero-level-set of a distance function $\mathbb{DF}_M(p)$ for $p \in \mathbb{R}^3$. These parameters can be repeated to support multiple independent rigid bodies. The most common sensor model assumes the sensor samples a set of N_s points $z = \{s_i\}$ from the geometry of the world.

Note that using this simplified model discards important information about the nature of real cameras – for example, the fact that a point being recovered from a depth sensor indicates that the ray between the camera origin and observed point is unobstructed. Some techniques (especially those from the SLAM community) recover this information with a more complex model, but as we will review, many more do not.

Using this point cloud model, we define the *point-cloud object pose estimation problem* as

$$\min_{R,T} \sum_{i \in [0,N_s]} \mathbb{DF}_M(Rs_i + T). \tag{1.1}$$

This objective is reflected in a majority of the pose estimation techniques in the literature. The key difference between techniques lies in the model representation and distance function used, as well as the method for optimizing Equation (1.1). Many techniques represent the model as a collection of N_m point features, and penalizes a norm:

$$\min_{R,T} \sum_{i \in [0,N_s]} \min_{j \in [0,N_m]} \|Rs_i + T - m_j\| .$$

Critically, this form of the distance function is not convex, and is difficult to directly optimize over. An equivalent form of this optimization that better reveals its convex subproblems is

$$\min_{R,T,C} \sum_{i \in [0,N_s]} \left\|Rs_i + T - m_{C(i)}\right\| , \tag{1.2}$$
$$C(i) = \operatorname*{argmin}_{j \in [0,N_m]} \|Rs_i + T - m_j\| ,$$

where $C(i)$ corresponds each scene point to the closest model feature according to the desired norm.

A critical feature of this problem is that the correspondences C and transformation R, T are independently sufficient to specify a solution to this problem. Given the correspondences, the optimal transformation can be computed in closed form [1]. Given the transformation, correspondences can be backed out if desired via, e.g., closest point lookups on the model.

1.2 Related Work

Here, we review object pose estimation from an incremental, or *tracking*, perspective; as well as from a *global* perspective. We will review the state of the art in visual perception with depth sensors, pose estimation with tactile sensing in the loop, and global optimization for pose estimation.

1.2.1 Local Optimization and Object Tracking

A widespread method for tackling the general pose estimation problem is to rely on local optimization methods. A well-known method for performing local search of Equation (1.2) is the Iterative Closest Point (ICP) algorithm [2, 3]. Given a guess for the model pose $\{R_k, T_k\}$ at iteration k, ICP performs an update in two steps: first, it performs closest point lookups on the model in configuration $\{R_k, T_k\}$ to find the current correspondences C_k. By fixing the correspondences to C_k, a new pose $\{R_{k+1}, T_{k+1}\}$ can be computed in closed form, with the guarantee that the new transform will be at least as good as the old. By alternating between correspondence and transform updates, ICP functionally performs expectation-maximization on Equation (1.2) and converges to a local optimum.

Techniques extending ICP have reached a significant level of maturity. Because ICP must be initialized with a reasonable guess of the model pose, ICP has been particularly popular in the subfield of object *tracking*. The object tracking problem extends the pose estimation objective (Eq 1.1), but is solved repeatedly as new data arrives from the sensor using the solution from the previous timestep as a prior:

$$x_k = \underset{x}{\operatorname{argmin}}\, p(z|x, x_{k-1}).$$

Modern object tracking methods are successful at tracking complex articulated objects including the human hand [4] and clothing [5]. ICP-based tracking techniques have been successfully deployed, for example, in the DARPA ARM-S manipulation competition [6, 7].

Almost all local optimization methods for tracking rely on this core idea of sepa-

rating the correspondence and pose update steps, but many add additional terms to reflect more complex measurement models and enforce physical constraints. Schulman et. al track models of highly-articulated soft bodies by performing forward physical simulation of the models with additional fictitious forces added to drive the estimated objects towards nearby points in the measured point cloud at each timestep [5]. Schmidt et. al, on the other hand, optimize more directly on the measurement model in their Dense Articulated Real-Time Tracking (DART) framework by leveraging the signed distance function (SDF) to directly align their object model to the incoming point cloud data, with additional terms in their optimization adding respect for free-space information implicit in the depth returns and contact constraints [4,8]. Like all methods based on ICP, these iterative tracking techniques show excellent performance when fed sufficiently rich data and a good initial guess. However, they are vulnerable to occlusion, which limits the amount of information sampled from the model. Occlusion is particularly likely when the model undergoes manipulation, because the robot's hand is likely to cover the object being manipulated.

It is important to acknowledge that not all methods are based on an ICP-like objective. Parts of the shape matching literature, for example, emphasize objectives that encourage matching of model parts with local similar geometry [9,10]. Instead of explicitly searching for good alignment transformations, these techniques emphasize the search for good correspondences between model features, where a good correspondence minimizes distortion of inter-feature distances. These techniques have proven very successful for deformable shape matching, and are relevant as an alternative form of pose estimation.

1.2.2 Tactile Sensing for Object Tracking

Tactile sensing presents an alternative solution to the occlusion problem in traditional point cloud tracking, by promising to provide data exactly when visual sensors are occluded. However, while tactile sensors take a wide variety of forms, few allow for recovery of dense local geometry – modular commercial sensors, including the SynTouch BioTac [11] and the RightHandRobotics Takktile sensor [12], typically

focus on providing either discriminative or scalar pressure signals.

Object tracking methods that address contact sensors correspondingly concentrate on taking advantage of discriminative contact signals. These signals often enter trackers as manifold constraints in which a positive contact signal constrains pose estimates to the set of poses that specify an object being in contact with the sensor. This set of poses is described as the contact manifold for the sensor. A diverse set of techniques have arisen to implement this constraint with projection methods based on the signed distance function [8] or physical modeling [13, 14], alongside other methods that sample pose candidates directly from a parameterization of the contact manifold in a particle filter [15, 16]. These methods are reviewed in greater detail in Chapter 2.

We observe that the GelSight sensor is uniquely able to address these localization challenges. The GelSight sensor is capable of producing a rich contact depth map in the vicinity of a contact, which is ideal for small-scale geometric localization of objects [17–20]. In Chapter 2, we present a object estimation pipeline which supports arbitrarily articulated models and non-planar contact surfaces by incorporating the dense geometric information from a GelSight sensor with an state-of-the-art ICP-based articulated object tracker. However, the interaction of the local iterative optimization with the sometimes ambiguous local contact geometry makes the tracker brittle in certain situations, and motivates further exploration into more robust pose estimation.

1.2.3 Global Search for Object Pose Estimation

The broader problem of from-scratch object pose estimation and detection has also been addressed over the years. While object tracking algorithms are iterative and hence require a previous guess for object position, from-scratch pose estimation algorithms do not. Unsurprisingly, this kind of global estimation problem is more difficult to solve reliably and efficiently due to the size of the search space.

Sampling approaches

When dealing with point clouds, many object detection or pose estimation algorithm tackle the correspondence problem in Equation (1.2) to correspond sample points to object models. This correspondence problem poses a significant problem for global optimization. Because exhaustive search of correspondences is usually intractable, modern techniques seek to accelerate or sidestep the correspondence search in some way. One popular class of approaches to this problem employs sampling to search directly over the raw point cloud [21–23]. Some solutions seek to make extremely efficient local search methods – primarily ICP – more robust. (See [24] for a brief review.) Each of these techniques are popular and can be very efficient, but struggle in ambiguous, occluded, and very noisy cases.

Local-feature approaches

Another broad class of approaches downsamples the original point cloud through local feature extraction, similar to how popular object detection techniques for RGB images extract 2D features [25]. These features typically reflect some aspect of the surface geometry in that area of the point cloud; examples include point pairs [26], spin images [27], and SHOT descriptions [28]. Features can also be learned using machine learning techniques (e.g. [29]). These reduced-but-higher-dimensional features are designed to be amenable to efficient exhaustive matching and grouping. These techniques can be much less sensitive to initial conditions, but again fall prey to situations with limited or ambiguous features, and ambiguous object signatures.

Template-matching approaches

Template matching methods have also grown in popularity. LINEMOD and its extensions leverage binning of geometric and image features to enable efficient template matching for object detection and coarse pose estimation, which can be post-processed with ICP to generate fine alignment [30]. If segmentation has already been performed, object signatures can be calculated and compared in the segmented point clouds to

efficiently produce pose estimates without initial guesses. (See [31] for an extensive overview.) These techniques struggle under significant occlusion due to their reliance on whole-object signatures.

Machine learning approaches

Machine learning techniques have demonstrated remarkable performance on tasks throughout computer vision. While a significant amount of the computer vision literature and its benchmarks focuses on bounding box detection in RGB images [32, 33] and pixelwise segmentation [34], these techniques have begun being expanded to 3D object pose estimation. Our collaborators at Draper, for example, have leveraged learned pixelwise segmentation to assign point-object correspondences, which are refined by an online ICP-based tracker to produce robust pose estimates [35]. In this technique, the deep convolutional neural network provides coarse initializations and high-level guidance to a local tracker. End-to-end prediction of pose from RGB and RGB-D data is beginning to be explored – e.g., for the related camera pose estimation problem [36]. These learning techniques are remarkably efficient and robust when properly trained, but are notoriously unpredictable in situations beyond their training set. This problem is compounded by practical limitations on the collection of labeled training data.

Global optimization

A final class of pose estimation and object detection techniques attempts complete global search of the solution space. Olsson et al. provided a routine for branch-and-bound search over $SO(3)$ [37], which was extended to $SE(3)$ in the GO-ICP algorithm to produce certifiably globally optimal solutions to the object pose estimation problem in a point cloud [24]. Their technique is efficient enough to run on practically-sized point sets, with the restriction that it assumes only a single object is present, and that outliers can be accounted for by rejecting the worst $N\%$ of point correspondences, for some pre-specified N. SDP relaxations of the closely related shape matching problem have also become popular, and can have great practical performance [38].

However, these techniques search over transforms in $O(3)$ instead of $SE(3)$, and again often assume a single object. A more detailed review of these techniques continues in Chapter 3.

1.3 Contributions

Many tactile sensors measure *force* at a single point or patch of contact, providing potentially rich dynamic information but limited geometric information. However, we are investigating the application of a tactile sensor – GelSight – which provides the dense *local geometry* of objects that come in contact with its surface. This sensor can thus directly supply the part of scene geometry that is otherwise lost due to occlusion of distant optical depth sensors. GelSight can provide exceptionally fine geometric information, capturing surface features as fine as 2 microns [39], and it can simultaneously measure shear and slip [40].

In Chapter 2, we utilize this fine contact geometry to enable precise object localization for small manipulands during occluded manipulation [41]. We achieve significant improvement in the tracking of small objects by treating the dense contact geometry information from GelSight as a point cloud in order to fuse it with large-scale geometric information from a visual RGB-D sensor. We combine both point clouds in a real-time ICP-based object tracker based on the recent object tracking work of Schmidt et. al [4, 8].

A limitation of this tracking technique is its reliance on purely local search methods. We experienced that the combination of local search with the limited volume of the contact sensor causes the tracker to frequently become trapped in incorrect local minima during tracking. While this issue is mitigated by the presence of a larger but less precise point cloud from an RGB-D camera, the tracking technique is unreliable when the initial guess for the tracker is sufficiently bad. This approach is too unreliable for use in feedback control systems that seek to understand and respond to unexpected contact.

Hence, in Chapter 3, we discuss a complementary approach to object pose esti-

mation that emphasizes global optimization. By approximating point-cloud-to-point-cloud and point-cloud-to-mesh registration as mixed-integer convex programs, we can leverage branch and bound techniques to efficiently solve registration problems to global optimality. We will show that this formulation can be used to perform object pose estimation in point clouds. Further, we show how to extend our formulation to explicitly classify points as outliers; how to assign points to one of multiple models that are being simultaneously considered; and how to leverage candidate poses from other pose estimation techniques as upper bounds to accelerate the global optimization.

Finally, in Chapter 4, we conclude and remark on future directions suggested by this work.

Chapter 2

Tracking Objects with Point Clouds from Vision and Touch

2.1 Introduction

Established approaches to manipulation tasks rely primarily on cameras and optical depth sensors to track object state. However, it is precisely when a robot's manipulator approaches an object that vision sensors are likely to be limited by occlusion. Incorporating tactile sensing into the pose tracker seems natural, but requires continued progress in both the tactile sensors and the algorithms that take advantage of their properties.

Many tactile sensors measure force at a single point or patch of contact, providing potentially rich dynamic information but limited geometric information. In this Chapter, we investigate the application of a tactile sensor—GelSight—which provides dense geometric information of objects that come in contact with its surface and can thus provide the localization data that is otherwise lost due to occlusion of distant optical depth sensors. Because of the smaller view area, GelSight can provide exceptionally fine geometric information, capturing surface features as fine as 2 microns [39], and it can simultaneously measure shear and slip [40]. We focus on utilizing this precise contact geometry to enable precise object localization for small manipulands.

Stereographic, structured light, and LIDAR sensors have spurred fundamentally geometric point-cloud based approaches to robotic perception. Many variants of the Iterative Closest Point (ICP) algorithm have been developed to locate and track objects in point clouds [42]. In conjunction with ICP, signed distance functions (SDF) have proven a valuable tool for reasoning in a continuous and smooth way about the geometries of objects and scenes, and have proven invaluable in object tracking, and simultaneous localization and mapping (SLAM) [4, 43–45].

We show that the contact geometry information from the GelSight contact sensor is compatible with traditional ICP-based tracking techniques. By constructing an object tracker that utilizes both precise, local contact geometry from GelSight, and large-scale point-cloud data, we achieve contact-aware object tracking that utilizes tactile data to output greatly refined pose estimates. We provide experimental results showing quantitative improvement to estimator performance when contact geometry information is added, and demonstrate the use of our system to track the grasping and manipulation of a small tool.

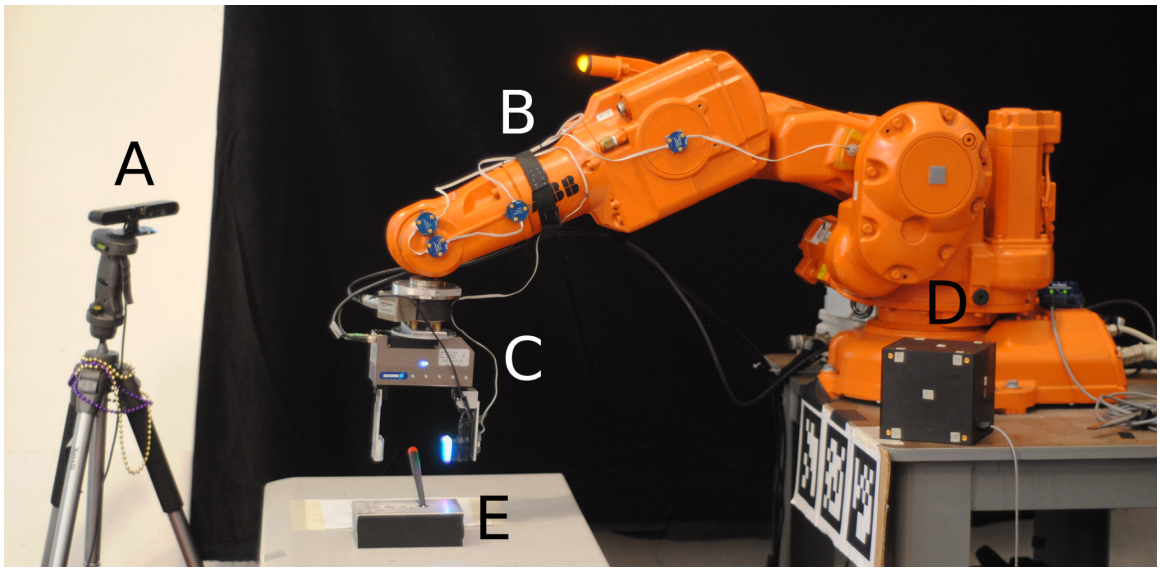

Figure 2-1: The setup used in our experiments consists of an Asus Xtion RGB-D camera (**A**) observing a 6-DOF ABB IRB-140 arm (**B**). The end effector is a Schunk WSG-50 parallel gripper with a GelSight-enabled custom set of fingers (**C**). A cube with attached optical markers (**D**), and a small screwdriver and rectangular holster (**E**) are used as manipulands.

2.2 Related Work

ICP informs many modern visual tracking methods. These methods have been extended to support articulated object collections with internal joints. Klingensmith, et al. demonstrate closed-loop servoing using articulated ICP for online pose estimation [7], and Hebert, et al. utilize articulated ICP for simultaneous manipulator and manipuland tracking [6]. Following a similar path, the Dense Articulated Real-Time Tracking (DART) framework [4] performs articulated object tracking from dense depth data in real time by leveraging the signed distance function (SDF) to efficiently align an articulated object model to an incoming stream of point cloud data, while balancing a free space term. These online tracking techniques show excellent performance when fed sufficiently rich data. However, they are vulnerable to occlusion during manipulation, when the robot's hand is likely to cover the object being manipulated, and are likely to lose tracking without an additional sensing modality or physically-derived constraint.

To address this issue, DART was extended to include nonpenetration and binary contact constraints, which are made continuous and efficiently enforceable via further application of the SDF [8]. A parallel body of work employs particle filters (PFs) to tackle exactly the ambiguity and nonlinearity often inherent in contact state estimation. Koval, et al. take advantage of the manifold structure of the state space of contact to greatly reduce the critical particle starvation issue facing PFs during contact events by resampling directly from the contact manifold [15]. Zhang and Trinkle tackle the same problem by using a constraint-based physical model to enforce that particle updates stay physically feasible with respect to nonpenetration and contact forces [13]. Li, et al. instead use a PF to track discrete contact modes in the contact graph, while performing continuous state estimation at each particle with a Kalman filter using a process update derived from the particle's contact mode [14]. While particle filters have better theoretical ability to represent the nonlinear and potentially multimodal state distributions that arise through contact events, they face difficult scaling issues even under these optimizations. Klingensmith, et al. make progress on

this scaling by leveraging SDFs to avoid expensive explicit parameterization of the contact manifold [16].

Tactile sensors take a wide variety of forms, but few allow for recovery of dense local geometry. Modular sensors designed to be used as fingertips include the SynTouch BioTac [11], which discriminates contact over the entire sensor surface, and the RightHandRobotics Takktile sensor [12]. Both of these sensors output a single pressure signal, though the Takktile sensor can be purchased in an 8mm tiled layout to sense rough contact location. Sensors with greater ability to resolve geometry are under active development. Jamali, et al. discusses the design of sensing skin for the iCub robot's fingertips, which utilize tiled force sensors at approximately 1mm spacing on a flexible PCB [46]. Patel and Correll present an alternate sensor design that combines distance and force elements [20].

We observe that there is a bias towards contact discrimination rather than recovery of dense contact geometry in the majority of these sensors, though cutting-edge sensing skins blur this distinction. The tracking algorithms surveyed above were tailored to these discriminatory sensors. Schmidt, et al. support a binary contact detection signal as an input, and estimates contact locations that explain the binary contact detections [8]. The contact manifold [15, 16] and contact-mode switching [14] approaches are natural when used with a discriminative sensor, but do not extend as naturally to dense geometric contact information.

A key component of our solution to these localization challenges is the GelSight touch sensor. This sensor is capable of producing a rich contact depth map in the vicinity of a contact, which is ideal for small-scale geometric localization of objects [17, 19, 20]. Li, et al. demonstrate a GelSight texture recognition pipeline for localizing objects in-hand, which they use to accomplish a precise peg-in-hole task [18]. By incorporating the dense geometric information from a GelSight sensor with an ICP-based articulated object tracker, we build upon this work by offering a more general estimation pipeline which supports arbitrarily articulated models and non-planar contact surfaces.

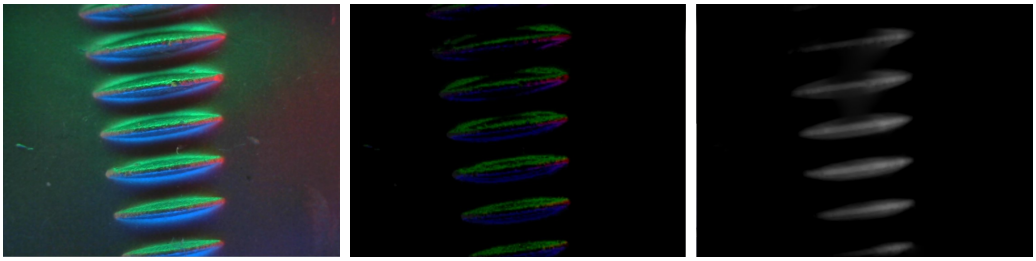

Figure 2-2: **Top:** Raw GelSight image of threads on a bolt. **Middle:** Gradient image generated by lookup table after calibration. **Bottom:** Final depth map.

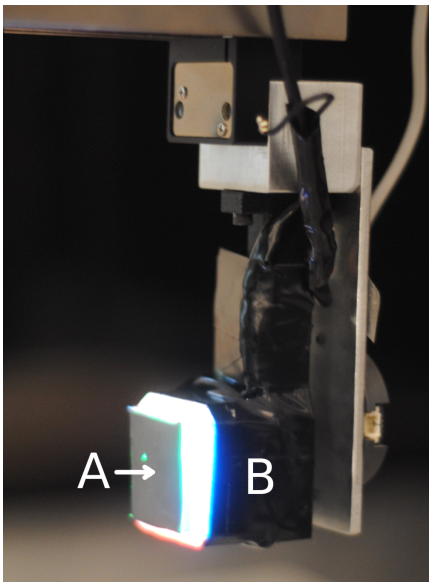

Figure 2-3: A GelSight sensor mounted as a finger on a gripper. The elastomer surface (**A**) deforms when pushed against other objects. Deformations of the surface are illuminated from multiple sides and captured with a webcam inside of the sensor housing (**B**).

2.3 Sensor Overview

Our tracker takes input from two primary sensors: a structured-light dense RGB-D camera, and a GelSight contact geometry sensor.

The GelSight sensor (Figure 2-3) consists of a thin elastomer observed by a conventional color camera. The camera captures deformations of the elastomer when the elastomer is pressed against an object. The GelSight sensor produces an RGB image which gives geometric information within a $11.5 \times 15 \times 2$ mm volume. The sensor surface is backlit by a different color of light from three sides, such that different slopes of the sensor surface correspond to different colors on the RGB image. By collecting the raw RGB images formed by contacting the sensor with known calibration surfaces, we learn a mapping from RGB points $\in \Re^3$ to depth map gradients $\in \Re^2$.

Following the technique of Li, et. al for producing a depth map from the GelSight images, we use a $16 \times 16 \times 16$ binned lookup table which maps the color of background-subtracted pixels to their corresponding gradient values [18]. For training data, we roll a 2.5mm-diameter ball bearing around the sensor and use machine vision to detect its location automatically in each image. This technique allows us to easily generate enough ground truth data to learn the mapping. At runtime, we use this lookup table to determine the gradients in the full-resolution image, then use a modified Poisson integration on a downscaled gradient image to obtain the depth map.

Poisson integration of the gradient image $\mathbf{g}^{h \times w \times 2}$ into the final depth map $\mathbf{p}^{h \times w \times 1}$ is performed via a large but sparse unconstrained least squares optimization over the $h \times w$ pixels $\mathbf{p}$ of the final depth map. For every horizontally neighboring pair of points $p_{x,y}$ and $p_{x+1,y}$, we add a horizontal gradient violation penalty $||(p_{x+1,y} - p_{x,y}) - g_{x,y,1}||^2$. Similarly, for every vertically neighboring pair of points $p_{x,y}$ and $p_{x,y+1}$, we add a vertical gradient violation penalty $||(p_{x,y+1} - p_{x,y}) - g_{x,y,2}||^2$. An additional set of terms enforces boundary conditions by penalizing $k_e \times ||p_{x,y}||^2$ for all x, y on the image boundary, using a gain k_e to weight this penalty against the integration penalties.

We perform this conversion in real time using OpenCV [47] and Eigen [48]. We have empirically found $k_e = 1.0$ to yield good results. Our pipeline can attain a

resolution of 256-by-186 tactels at a rate of 12 Hz.

2.4 Tracking Algorithm

Our tracking algorithm takes as input a continuous stream of RGB-D images from an off-the-shelf dense depth sensor, and depth images from the GelSight sensor. We take inspiration from the DART tracking system of Schmidt, et al. [4, 8] and construct a single-hypothesis tracker based on an EKF. We use the same formulation, but offer novel optimization strategy. The formulation that follows in this section is repeated from the original presentation of DART in order to motivate the subsequent description of our optimization strategy.

2.4.1 Modified EKF Formulation

At a time step k, we estimate the state x_k and its variance Σ_k. x_k collects positions and velocities, including floating base translations and rotations and joint angles.

Following a standard EKF formulation, we can use a dynamic model of the scene to generate a *predicted* state and variance $\bar{x}_k, \bar{\Sigma}_k$, using the previous estimated state x_{k-1} and any relevant control inputs u_{k-1}:

$$\bar{x}_k = f(x_{k-1}, u_{k-1}),$$
$$\bar{\Sigma}_k = J(x_{k-1}, u_{k-1})\Sigma_{k-1}J(x_{k-1}, u_{k-1})^\top + W.$$

Here, $f(x_{k-1}, u_{k-1})$ is the process update, J its Jacobian, and W additive process error. The simplest model would be to assume the state never changes and the variance slowly increases; this corresponds to using $f(x_{k-1}, u_{k-1}) = x_{k-1}$ and W nonzero.

A standard EKF would call for the measurement update to be performed by computing a predicted measurement $h(\bar{x}_k)$ and the measurement residual $\tilde{y}_k = z_k - h(\bar{x}_k)$ using forward measurement models. However, the forward measurement model $h(\bar{x}_k)$ is discontinuous in the case of a camera, and would yield poor gradients and an

ineffective approximate Kalman gain. Thus, instead of the standard form, we write the measurement update as a direct optimization of system state over measurement probabilities derived from our sensors. Using θ our decision variable, we write:

$$x_k = \underset{\theta}{\operatorname{argmin}} \Big[-\log(p(z_k|\theta)) + (\theta - \bar{x}_k)^\top \bar{\Sigma}_k^{-1} (\theta - \bar{x}_k) \Big],$$
$$\Sigma_k = H(x_k)^{-1}.$$

Here, generalized sensor readings for time step k are written z_k. Since x_k is a maximum likelihood estimate given the negative log-likelihood function above, the variance update takes the form of the inverse of the Hessian $H(x_k)^{-1}$ of that negative log-likelihood function.

The specific optimization problem we will solve depends on the construction of $p(z_k|\theta)$ for our particular set of sensors.

2.4.2 Measurement Model for Point Clouds: Positive Returns

A depth sensor produces a list of pixels $im^{meas} = \{pixel_i^{meas} \in \Re\}$, and from each we can calculate a point in space $pt_i^{meas} \in \Re^3$ using the camera calibration.

Following DART, we will suppose that the likelihood of a point is normally distributed with respect to the signed distance to the closest surface (signed distance function, *SDF*), which depends on the system state θ. We assign a variance of σ reflecting the depth sensor noise characteristics:

$$p(pt_i^{meas}|\theta) = Ke^{-SDF(pt_i^{meas};\theta)^2/\sigma^2},$$
$$K = \frac{1}{\sqrt{2\pi\sigma^2}}.$$

The likelihood of the complete image combining all pixels (indexed by i) is

$$p(image|\theta) = \prod_i p(pt_i|\theta).$$

After taking the negative log likelihood, the expression simplifies:

$$-\log\Big(\prod_i p(pt_i^{meas}|\theta)\Big) = \sum_i -\log\Big(p(pt_i^{meas}|\theta)\Big) =$$
$$\sum_i -\log\Big(Ke^{-SDF(pt_i^{meas};\theta)^2/\sigma^2}\Big) =$$
$$\frac{1}{\sigma^2}\sum_i SDF(pt_i^{meas};\theta)^2 + \log(K).$$

We drop the constant term $\log(K)$, as it has no dependence on our optimization variable θ.

2.4.3 Measurement Model for Point Clouds: Free Space

As pointed out by Ganapathi, et al. [49], for each point pt_i in the point cloud, we know that there must be clear line-of-sight between the camera origin and that point. Thus, we know that no surface on the proposed model can lie between that point and the camera. If, for a given proposed model, we produce a simulated depth image as a collection of depths $im^{sim} = \{pixel_i^{sim}\}$, then we want to constrain $pixel_i^{sim} \geq pixel_i^{meas}$. Directly constraining this value yields poor performance, as the simulated depth returns have sharp discontinuities around object edges which hinder optimization. As such, Ganapathi, et al. and Schmidt, et al. instead partition $\Re^3$ into space known to be free, and space out of sight of the camera, and constrain all points on the surface of the proposed model to lie in the second partition [4, 49]. The partitioning surface, S_{obs}, is defined by the points in the measured point cloud; free space lies in front of S_{obs}, and out-of-sight space lies behind it.

We suppose that the probability of simulated depth point $pixel_i^{meas}$ is constant in out-of-sight space, and decreases with distance to the out-of-sight space. To calculate this, we will create another distance function, DF_{obs}, which yields the distance a given point has to move for it to leave free space.

Given DF_{obs}, we specify the probability density to be

$$p(pt_i^{sim}|im^{meas}) \propto Ke^{-DF_{obs}(pt_i^{sim})^2/\sigma^2}.$$

Following similar steps as for the positive return case, computing probability over all points in the simulated depth image, taking the negative log likelihood, and dropping the constant term gives

$$-\log\Big(\prod_i p(pt_i^{sim}|im^{meas})\Big) = \frac{1}{\sigma^2}\sum_j DF_{obs}(pt_j^{sim};\theta)^2.$$

2.4.4 Likelihood Model for Nonpenetration

During manipulation experiments, we observed a need to further constrain estimates to be physically feasible with respect to penetration. Inspired by DART, we implement this using a very similar distance-function-based penalty to that used in the free space constraint. For each point pt_i^{surf} sampled from the surface of an object, we suppose that the probability density falls off with the penetration distance DF_{pen} into the surfaces of the rest of the robot in configuration θ.

$$p(pt_i^{surf}|\theta) \propto Ke^{-DF_{pen}(pt_i^{surf};\theta)^2/\sigma^2}.$$

As before, computing probability over all points sampled from the surfaces of objects of interest, taking the negative log likelihood, and dropping the constant term gives

$$-\log\Big(\prod_i p(pt_i^{surf}|\theta)\Big) = \frac{1}{\sigma^2}\sum_j DF_{pen}(pt_j^{surf};\theta)^2.$$

2.5 Optimization

At each step, we compute a measurement update given the latest depth image from the RGB-D sensor, a depth image from the GelSight sensor, and the last estimated

state θ. Written out in full, this update is:

$$\begin{aligned} x_k = \underset{\theta}{\operatorname{argmin}} \quad & \frac{1}{\sigma^2_{kinect}} \sum_{\text{kinect pts}} SDF(pt_i^{meas};\theta)^2 \\ & + \frac{1}{\sigma^2_{kinect}} \sum_{\text{kinect pts}} DF(pt_i^{sim};\theta)^2 \\ & + \frac{1}{\sigma^2_{gelsight}} \sum_{\text{gelsight pts}} SDF(pt_i^{meas};\theta)^2 \\ & + \frac{1}{\sigma^2_{gelsight}} \sum_{\text{gelsight pts}} DF(pt_i^{sim};\theta)^2 \\ & + \frac{1}{\sigma^2_{nonpen}} \sum_{\text{surface pts}} DF_{pen}(pt_i^{surf};\theta)^2 \\ & + (\theta - \bar{x}_k)^\top \bar{\Sigma}_k^{-1} (\theta - \bar{x}_k). \end{aligned}$$

As written, this optimization is nonlinear. It is particularly tough because changing θ changes $SDF(pt_i^{meas};\theta)$, pt_j^{sim}, $DF_{obs}(pt_i^{sim})$, and $DF_{pen}(pt_i^{surf})$ in complex ways depending on the shape of the object surface. We solve this problem by iteratively constructing and solving approximating unconstrained quadratic programs (QPs).

2.5.1 Approximate Minimization of $SDF(pt_i^{meas};\theta)^2$

In every iteration, we calculate the closest point $\hat{pt}_i$ and corresponding body $body_i$ to pt_i^{meas} in our model in configuration x_{k-1}. These closest point calculations are performed directly via the Bullet collision library, working on convex decompositions of the robot's collision geometry. We observe that locally, $SDF(pt_i^{meas};\theta)^2 \approx ||pt_i^{meas} - \hat{pt}_i||^2$. Using the Jacobians J_{cam} at the camera and J_{body_i} at $body_i$ computed via forward kinematics, we can minimize that term by finding

$$\underset{\theta}{\operatorname{argmin}} \, ||(pt_i^{meas} - \hat{pt}_i) + (J_{cam} - J_{body_i})(\theta - x_{k-1})||^2.$$

The minimizer for this expression corresponds to θ that moves both the camera and $body_i$ to place the measured point pt_i^{meas} on the body's surface.

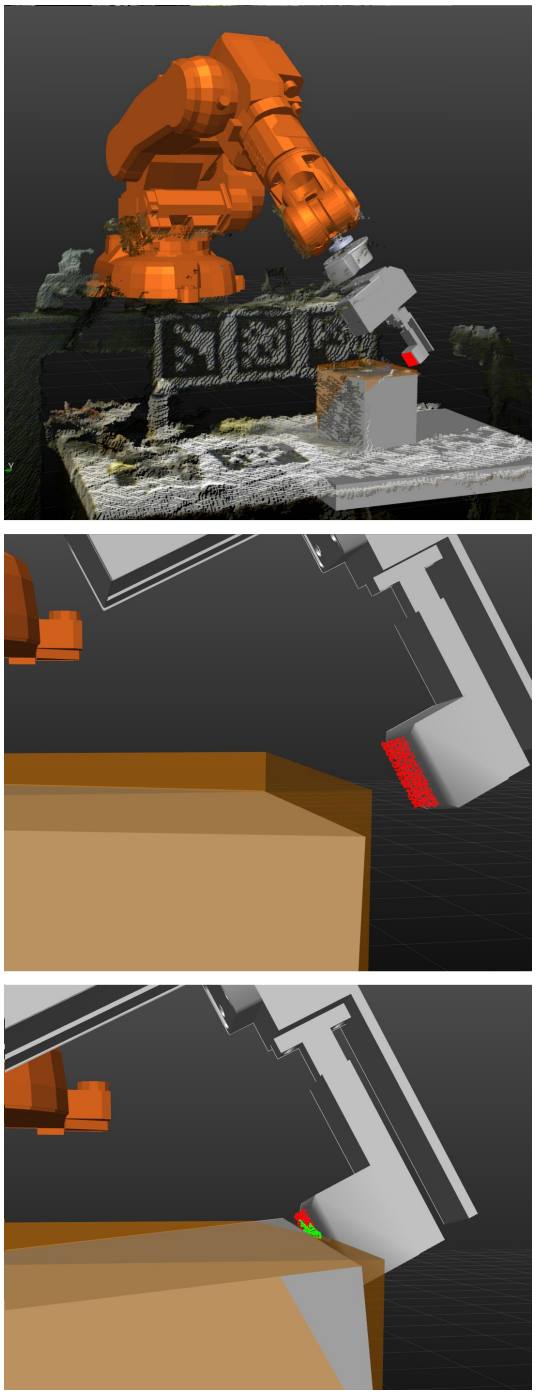

Figure 2-4: **Top:** We use a GelSight sensor mounted on the end of a 6-DOF arm to manipulate a simple object. The benchtop is observed by an RGB-D sensor, and ground truth manipulator and object positions are provided by an external motion capture system. **Middle:** The object pose estimate (solid gray) from RGB-D data, alongside the ground truth object pose (transparent orange). A 1-cm vertical bias is injected into the RGB-D data for demonstrative purposes, causing the object pose estimate to be approximately 1cm low. **Bottom:** When the GelSight sensor is brought in contact with the object, the dense contact geometry information is used to improve the object pose estimate, correcting the 1cm bias in the vicinity of the contact.

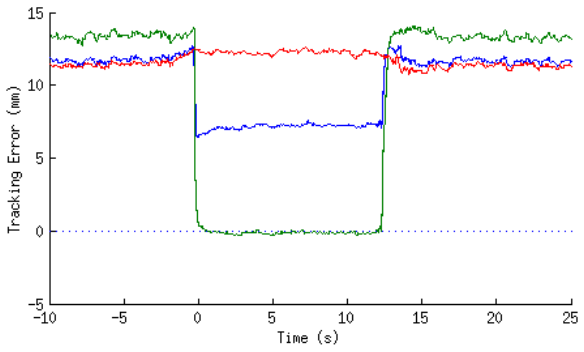

Figure 2-5: Tracking error relative to the hand pose, in the vertical axis, as judged against ground truth, of the simple object during manipulation by the GelSight manipulator. **Red**: relative tracking error with no GelSight data used. **Blue**: relative tracking error to the object's centroid when GelSight data is used. **Green**: relative tracking error to the near edge of the object when GelSight data is used. A 1cm vertical bias was injected into the RGB-D point cloud data, causing the object tracking to exhibit a consistent, approximately 1cm bias in the absence of additional information. When the GelSight is brought in contact with the object (at $t = 0$), the contact geometry information counteracts this bias and improves tracking performance significantly. The tracking error returns when the contact is removed (at $t = 13$). The improvement is strongest in the vicinity of the contact sensor.

2.5.2 Approximate Minimization of $DF_{obs}(pt_i^{sim};\theta)^2$

For pt_i^{sim} on the surface of $body_i$, we calculate the closest point $\hat{pt}_i$ to pt_i^{sim} that is on or behind S_{obs}. Taking inspiration from Ganapathi et al. [49], we compute DF_{obs} and find this closest point efficiently by decomposing DF_{obs} into components perpendicular and parallel to the camera view ray. This decomposition allows us to avoid computing the full 3D distance function to S_{obs}. We calculate a 2D distance function finding the nearest pixel of the image for which $pixel_j^{sim} \geq pixel_j^{meas}$, which indicates how far we would have to move $body_i$ laterally for it to leave free space; and a 1D distance function, which is simply $pixel_j^{meas} - pixel_i^{sim}$, which indicates how far back we would have to push $body_i$ for it to leave free space. We set $\hat{pt}_i$ to the shorter of these two correspondences. We observe that locally, $DF_{obs}(pt_i^{sim};\theta)^2 \approx ||pt_i^{sim} - \hat{pt}_i||$. Again using the Jacobians J_{cam} and J_{body_i} at the camera and $body_i$, we can minimize

that term by finding

$$\underset{\theta}{\operatorname{argmin}} ||(pt_i^{sim} - \hat{pt}_i) - (J_{cam} - J_{body_i})(\theta - x_{k-1})||^2.$$

The minimizer for this expression corresponds to θ that moves both the camera and $body_i$ to place the simulated depth point pt_i^{sim} out of the measured free space.

2.5.3 Approximate Minimization of $DF_{pen}(pt_i^{surf}; \theta)^2$

For pt_i^{surf} on the surface of $body_i$, we calculate the closest point $\hat{pt}_i$ to pt_i^{surf} outside of all other bodies. Locally, $DF_{pen}(pt_i^{surf}; \theta)^2 \approx ||pt_i^{surf} - \hat{pt}_i||$. Again using the Jacobians J_{cam} and J_{body_i} at the camera and $body_i$, we can minimize that term by finding

$$\underset{\theta}{\operatorname{argmin}} ||(pt_i^{surf} - \hat{pt}_i) - (J_{cam} - J_{body_i})(\theta - x_{k-1})||^2.$$

The minimizer for this expression corresponds to θ that moves $body_i$ to move the point pt_i^{surf} to the surface of the object it is penetrating.

2.5.4 Solution

All of these approximate minimizers are unconstrained QPs in θ. We solve this QP online by solving the necessary and sufficient conditions for optimality as a linear system using QR Factorization with column pivoting in Eigen [48].

2.6 Experimental Results

We have implemented this tracking framework, and present experimental results of the tracker running on the testbed documented in Figure 2-1. We employ a 6-DOF manipulator with the GelSight sensor mounted as finger on a parallel gripper. We observe the scene with an Asus Xtion PRO LIVE RGB-D camera, and use an additional optical motion capture system to recover ground truth for one experiment. The

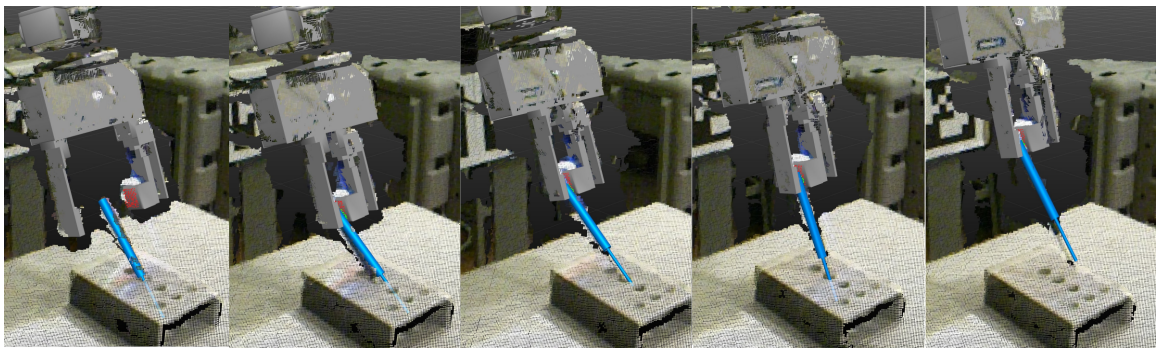

Figure 2-6: A GelSight sensor on a parallel gripper on the end of a 6-DOF arm is used to manipulate a small screwdriver via teleoperation of the arm. Rendered robot and object pose estimates are overlayed with the point cloud information from the RGB-D camera. The GelSight sensor surface is shown in red where the depth is less than a threshold (indicating no contact), and green where depth is above the threshold (indicating contact). Our system successfully tracks the position of the screwdriver throughout a sequence of manipulations to remove the screwdriver from a holster. This manipulation involved sigificant contact between the screwdriver and holster, and caused the grasp of the screwdriver to shift significantly.

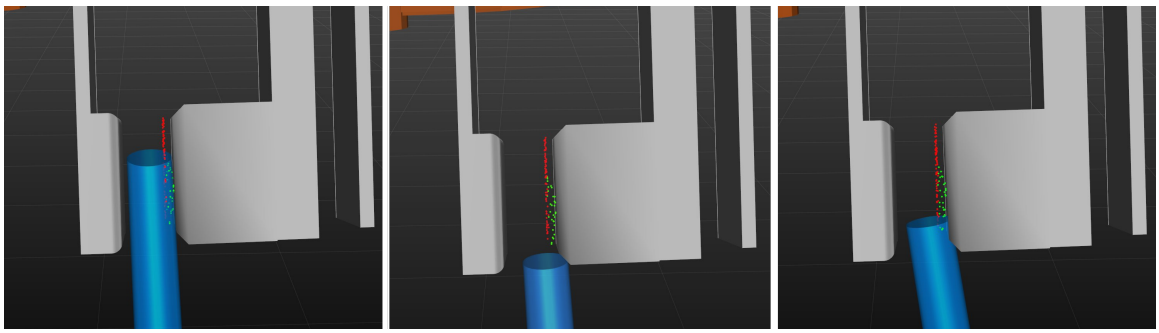

Figure 2-7: The use of precise contact geometry information from the GelSight sensor enabled a significantly more precise pose estimation (**Left**) than was achieved by the same tracker with GelSight disabled (**Middle**), or by the same tracker with GelSight data and nonpenetration constraints disabled (**Right**). When the screwdriver was in the gripper, the grasped section was occluded from the RGB-D camera, preventing an RGB-D-only tracker from producing an accurate fit. All three estimates pictured are the stable final pose produced by the tracker after the same manipulation sequence.

tracker runs as a single thread on a high-end desktop computer, and updates between 10 and 30 Hz depending on the complexity of the scene. To increase the tracking performance, we calibrate the RGB-D sensor position with an AprilTag of known global position, and do not estimate the RGB-D sensor position online. The collective state of the robot arm, attached GelSight sensor, table surface, and manipulated objects are estimated online by our tracker during teleoperation. This state includes multiple pin joint articulations within the arm, and floating base articulations for every object.

We present two experiments. The first demonstrates that contact information from the GelSight sensor can be used to quantitatively improve tracking performance for manipulation. The second employs our tracking framework to track manipulation of a small tool, and demonstrates qualitative improvement of tool pose during the manipulation. Both experiments run the same code, with the only difference being minor parameter changes made between experiments to increase the stability of the tracking of the small tool.

2.6.1 Demonstrating Quantitative Tracking Improvement

There are many sources of bias when fitting an object to a point cloud, including the use of inaccurate object models and poor camera calibration. However, as argued in Schmidt, et al. [8] it is not global object tracking accuracy that matters, but rather the accuracy of the estimated transform between the robot's hand and the object. A tactile sensor mounted directly on the hand has the potential to dramatically reduce this relative error. To demonstrate this, we use our tracker to estimate the pose of a robot arm and simple object before and after contacting the object with the GelSight sensor. To make the effect of the GelSight data more clear, we induce a 1cm vertical error in the hand-object relative pose by injecting a 1cm vertical bias in the estimated RGB-D camera position. When the GelSight reports contact with the box, the tracker updates the estimated object pose to better explain the detected contact geometry (Figure 2-4). This correction dramatically reduces the hand-object relative tracking error (Figure 2-5).

2.6.2 Demonstrating Small Tool Manipulation

Grasping and using small tools is a difficult task for many robots due to the difficulty of pose estimation of the tool in the robot's hand. When the hand closes on an object, the object is occluded from the external high-resolution vision sensors that could otherwise have been used to estimate its pose. This occlusion poses a serious limitation to the use of tools like screwdrivers, whose precise in-hand pose is critical for effective use. We demonstrate our tracker providing a pose estimate for a small screwdriver while the robot is teleoperated through extraction of the screwdriver from a holster. This maneuver involves significant contact between the screwdriver and holster, which causes the in-hand pose of the screwdriver to shift continuously. Our tracker maintains an accurate pose estimate of the tool throughout the procedure (Figure 2-6). This experiment utilizes the same code as the qualitative tracking experiment, with the only parameter tweaks being adjustment of the weighting of the dynamics model and RGB-D camera data. These parameters were reduced in this experiment to improve robustness to transient outliers in the RGB-D data. We provide, for comparison, tracking results acquired by running the same tracker without GelSight data, and by running the same tracker with neither GelSight data nor nonpenetration constraints (Figure 2-7). Tracking fails in both cases due to a combination of occlusion and the tight fit of the screwdriver in the hand.

2.7 Discussion

Existing contact-aware object trackers work hard to incorporate sparse tactile data from discriminative contact sensors. Koval, Klingensmith, and others calculate contact manifolds on which poses must lie when a binary contact sensor is active [15, 16], and Schmidt et. al were forced to estimate the contact position as an additional state to incorporate a binary contact detector into DART [8]. Object tracking algorithms that operate on point clouds, in comparison, have proven more scalable and mature. The accurate, dense, and fundamentally geometric output of the GelSight sensor affords an opportunity to apply these point cloud algorithms directly to a tactile sensor,

thus circumventing issues associated with sparse contact sensing.

Our quantitative tracking experiment demonstrates that the inclusion of GelSight data into an articulated object tracker can significantly improve relative hand-object pose estimates. Because the tactile sensor sits directly at the interface between end effector and the object being manipulated, it is expected to greatly decrease pose tracking error during contact. Our experimental data meets this expectation: the GelSight-enabled tracker is able to recover from significantly inaccurate point cloud data once contact is made, with the relative pose error at the contacted edge falling from greater than one centimeter to below one millimeter. The relative pose error at the object's centroid is also reduced, but unlike the edge being contacted, the error does not fall to zero, because the contact geometry information is local to the contact location and provides too little additional information about the pose of the rest of the object to overpower the biased RGB-D data. The small tool manipulation example demonstrates a practical use of this tracking technique, and highlights the importance of dense geometric tactile sensing as a tool for fighting occlusion. The control cases which ignored the GelSight data were consistently unable to accurately localize the tool in the hand, because the gripper occluded the part of the tool inside of the grasp.

While our system benefits from the simplicity of treating dense geometric tactile data as a point cloud, our approach has limitations that will require further work to resolve. Principal among these limitations is that the contact geometry information available is typically small in volume. During a grasp on an arbitrary object, the contact volume is likely to encompass only a small fraction of the object's total surface. As many small regions on the object's surface are likely to look similar to one another, it is easy for the tracker to fall into local minima. Thus, in the experiments in this paper, we rely on the RGB-D data to provide a strong enough prior to provide adequate initial guesses for our tracker to converge when contact is made. There are many potential solutions to this problem, including the extension to external initialization by a single-shot pose estimator, multi-hypothesis tracking, increasing sensor depth and size, and the introduction of texture as an additional

element in the measurement model [18].

2.8 Conclusion

We extend the state-of-the-art articulated object tracker DART to fuse point cloud information from an RGB-D camera with accurate and dense geometric contact data from a GelSight sensor. By focusing on a contact sensor as a source of geometric data, we can leverage dense tactile information identically to conventional point cloud data within the articulated object tracker. The application of our tracking system to fine manipulation tasks shows that the inclusion of dense and accurate tactile information is effective at solving occlusion problems. We believe that geometric sensors like GelSight used in combination with point cloud object tracking techniques will enable the execution of previously unachievable tasks ranging from small parts assembly to grasping of soft and novel objects.

Chapter 3

Global Object Pose Estimation

3.1 Introduction

Pose estimation is a prerequisite for any robot to interact with the world. However, existing techniques struggle in the clutter, heavy occlusion, and ambiguity typical in real-world scenarios. In the case of RGB-D cameras, tightly packed objects occlude each other, forcing the robot to make inferences about world state from partial observations. In the case of tactile sensing, the situation is even worse: object observations are both intermittent and limited to the contact patch of the sensor by the nature of the sensing modality.

As demonstrated in Chapter 2, pose estimation given tactile geometry can be treated identically to pose estimation given dense depth sensing from an RGB-D camera. However, classic local approaches like ICP are likely to fail on tactile point clouds unless they are combined with an accurate prior. This limitation proved practically significant during the execution of the experiments shown in Chapter 2. A common failure mode was for the pose estimates of small objects being manipulated to diverge before the tactile sensor made contact. Because the RGB-D camera provides only a small number of sample points on small objects, the objects were frequently recruited to explain nearby unmodeled points (e.g. from the robot's wiring harness, or from extra clutter on the desk top). In an ideal system, the object's pose would be reacquired as soon as the tactile sensor is brought into contact. However, in many cases,

the pose estimate diverged enough that by the time contact was made and rich data was available, the pose estimates were out of the basin of attraction of the global solution.

Such failure cases would be remedied by a solver capable of searching beyond the local solution space to find a globally optimal solution. Global optimization is extremely hard in general, as it suffers from a curse of dimensionality. However, certain classes of problems admit efficient search algorithms to produce globally optimal solutions with practically useful runtime. In this Chapter, we show that the general point-cloud pose estimation problem can be well-approximated with mixed-inter convex programming (MICP) formulations that admits one such efficient algorithm for global optimization. We show that these MICP forms are easily extensible, and enable explicit outlier handling and multiple simultaneous object fitting. We further show that our formulation enables us to leverage candidate poses from those other pose estimation techniques as upper bounds to accelerate the branch-and-bound search process. Our MICP formulation has significant practical value for certifying the *optimality* and quantifying the *ambiguity* of solutions to the pose estimation problem in difficult situations.

3.2 Related Work

Due to the enormous volume of point cloud data available from modern dense depth sensors, many approaches for performing pose estimation from that data rely entirely on local search; others fall back on downsampling to make global optimization more tractable; and finally, a handful of techniques attempt global optimization over the solution space for the full problem. To preface this Chapter, we will review a handful of techniques from the latter two categories.

3.2.1 Downsampling

Many techniques opt to compress the point cloud data to a more manageable size in order to decrease the volume of data that enters the optimization. This class of

techniques aims to take advantage of the massive redundancy inherent in a raw point cloud: a point cloud may contain many thousands of points, but there are ultimately only a handful of free variables to determine. A perfect engineered downsampling strategy would ideally extract a minimal set of features from the point cloud to perfectly recover the optimal solution to the general pose estimation problem.

The 4PCS algorithm [23] takes this concept to its extreme by sampling four coplanar points from the scene. Given that dramatic compression of the scene, the 4PCS algorithm efficiently finds the optimal four model point correspondences for those scene points. The optimal correspondence of the chosen four points generates a hypothesis for a pose, which is utilized in a Monte Carlo consensus sampling algorithm (i.e. RANSAC [21]) to reliably find good object pose estimates.

A great wealth of geometric feature extractors provide a means to collapse point cloud information into fewer, higher-dimensional features [26–29]). The higher dimension of these features makes it much easier to detect correspondence by simple nearest neighbor searches; and the lower number of features makes global search for optimal correspondence more tractable. Zhou et al. demonstrated a pseudo-global search method over point cloud feature correspondences leveraging graduated convexification [50]. Gelfand et al. build up a branch and bound algorithm to optimally correspond a handful of descriptors [51]; this technique is discussed in more detail below.

3.2.2 Global Optimization

With semidefinite programming

One strategy for tackling global optimization of this problem is to use a semidefinite programming (SDP) relaxation of the rotation and correspondence constraints to constrain R to be within a convex hull of $SO(3)$, and allow a continuous relaxation of C [52]. This relaxation transforms the difficult nonlinear problem to a much easier convex one. This technique has proven very powerful for solving the Procrustes Matching (PM) problem, which is equivalent to our general pose estimation problem

with fixed translations (adapted from [38]):

$$
\begin{aligned}
\underset{R,C}{\text{minimize}} \quad & \sum_{i \in [0,N_s]} \left\| R s_i - M C_{i,:}^T \right\|_2^2 \\
\text{subject to} \quad & R^T R = R R^T = I, \\
& \sum_{j \in [1,N_m]} C_{i,j} = 1, \ \forall i, \\
& \sum_{i \in [1,N_s]} C_{i,j} = 1, \ \forall j, \\
& C_{i,j} \in [0,1], \ \forall i, j.
\end{aligned}
$$

The SDP relaxation of the PM problem operates by relaxing $R^T R = R R^T = I$ to $R^T R = R R^T \preceq I$. Maron et al. leverage a reformulation of this relaxation to correspond tens to hundreds of points in tens of seconds [38].

Chaudhury et al. apply a similar SDP relaxation to the pose estimation problem in both rotation and translation for aligning multiple point clouds simultaneously [53]. Their method boasts tightness up to a quantified noise threshold, and is demonstrated aligning 800 points across 30 overlapping point clouds. However, their method assumes given correspondences are known a priori.

With branch and bound

Operating on an expanded form of the PM problem that generalizes to non-point models, Olsson et al. provided an alternative means for global optimization of the pose estimation problem with pre-specified correspondences via branch-and-bound search over $SO(3)$ [37]. Olsson's algorithm performs its search over quaternions by branching in the ambient space of $q \in \mathbb{R}^4$. Similar methods from Hartley and Li search instead over the space of angle-axis parameterized rotations [54, 55].

This idea has recently been extended further: in the Globally Optimal ICP (GO-ICP) algorithm, Yang et al. [24] instead perform a branch-and-bound search over the space of angle-axis parameterized rotations $r \in \mathbb{R}^3$, where $\|r\| \leq \pi$ parameterizes all

valid rotations; as well as over translations $T \in \mathbb{R}^3$. Given bounds on the members of r and T, Yang derives corresponding bounds on the registration error of every scene point. Given the center of a search region $\{r_0, T_0\}$, one can calculate the optimal error for the i^{th} scene point as $e_i(r_0, T_0)$. This calculation can be made very efficient by precomputing the distance function to the model. Given a measure of the search region width $\{\gamma_r, \gamma_T\}$, Yang derives a lower bound for the improved error $\underline{e_i}(r, T)$ possible for any $\{r, T\}$ within the bounding box:

$$\underline{e_i}(r, T) = e_i(r_0, T_0) - (\gamma_r + \gamma_T).$$

A significant but subtle advantage of this approach is that it tackles the full pose estimation problem but avoids explicitly searching over correspondences, which allows the algorithm to scale to much larger scenes and more complicated models. Yang presents results fitting 1,000 scene points to 40,000 model points in tens of seconds. Rules for handling outliers – for example, rejecting the farthest N% of correspondences as outliers – can be encoded in the calculation of $e_i(r_0, T_0)$.

While GO-ICP accomplishes our broad goal of providing globally optimal pose estimates, it does not explicitly reason about correspondences. This manifests itself most clearly in the handling of outliers: a user of GO-ICP must specify the expected fraction of outliers ahead of time, and setting the parameter incorrectly may result in invalid results. (Setting the parameter too high will force the algorithm to consider less than the full number of points sampled from the object; setting the parameter too low will force the algorithm to always include some outlier points in the error calculation, which will compromise the final pose estimates.)

Other techniques have tackled correspondences. Enqvist et al. demonstrated a solution for 3D-3D registration with a branch-and-bound algorithm based by relating the problem to the vertex cover problem [56]. They apply their algorithm to correspondences of high-dimensional extracted features, and are able to correspond hundreds of points in seconds. Gelfand et al. demonstrated a similar branch-and-bound algorithm relying on distance matrix comparisons [51]. Both of these algorithms take

advantage of the property that it is easy to detect inconsistencies in small sets of correspondences in order to prune branches in the search tree.

The transform and correspondence information are tightly coupled in the pose estimation problem. Thus, a formulation that reasons about point correspondences and model transformations *simulataneously* stands to benefit from this interplay. In this Chapter, we will present such a formulation.

3.2.3 Mixed-Integer Programming

Mixed-integer programming (MIP) provides a formalism for optimization with the constraint that some variables take binary or integer values [57]. While even restricted class of mixed-integer linear programs is itself NP-hard, MIPs that are convex in their continuous and integer variables are amenable to branch and bound search that can be very efficient, given the right problem structure. These algorithms are implemented by powerful off-the-shelf solvers capable of solving problems with millions of variables and constraints [58]. In particular, we will focus our attention on the class of mixed-integer linear programs (MILP) with binary variables:

$$\begin{aligned} \underset{x,y}{\text{minimize}} \quad & c^T \begin{bmatrix} x \\ y \end{bmatrix} \\ \text{subject to} \quad & A \begin{bmatrix} x \\ y \end{bmatrix} \geq 0, \\ & y \in \{0,1\}. \end{aligned}$$

MILPs (and, more broadly, MICPs) can be solved with a branch and bound algorithm [57]. While the worst-case time complexity of this algorithm is still exponential, in practice it results in dramatic speed improvements by focusing computational effort on the right regions of the solution space. The convergence of the algorithm can be separated into two components. An *upper bound* on the global optimal cost is improved by searching for better integer-feasible solutions. A *lower bound* on the global

optimal cost is improved by solving partial LP relaxations of the original problem, in which some integer variables are assigned fixed integer values while the rest are allowed to vary continuously. Broad sets of solutions can be shown to not contain the optimal solution by showing that the set's LP relaxation is either infeasible, or has optimal cost worse than the current upper bound. By continuously branching the space of solutions into increasingly specific partial relaxations, a lower bound on the global optimal cost is established and improved. These two searches are tightly coupled, in that better upper bound estimates make it easier to prune away bad relaxations; and the process of pruning bad solution regions and refining good ones provides guidance for finding better feasible solutions. It is possible for the full optimal solution to be known well before its global optimality is proven, if the convergence of the lower bound is slower than the convergence of the upper bound.

Mixed-integer convex programming has seen occasional use in robotics: for example, in planning trajectories for UAVs [59], planning footstep locations for humanoid robots [60], and for trajectory optimization for legged robots [61]. Each of these techniques utilizes integer variables to encode discrete choices that need to be made by the planner, while further continuous variables and convex constraints to encode the dynamics of the robot. Of these techniques, Deits and Valenzuela both include models of rotation in their motion planning formulation, and address the nonconvexity of rotations with piecewise linear approximation.

General nonlinear functions can be approximated in mixed-integer convex frameworks by adding integer variables to control which piece of a piecewise convex approximation is active. This approximation involves a disjunction over (e.g.) linear constraints:

$$\begin{aligned} \forall i: \; & a_i^T x \leq b_i + \mathbb{M}(1 - z_i), \\ & \sum_i z_i = 1, \\ \forall i: \; & z_i \in \{0, 1\}, \end{aligned}$$

in which $z_i = 1 \implies z_{j \neq i} = 0$, and $z_i = 0$ deactivates constraint i given sufficiently large $\mathbb{M}$. An equivalent formulation implements the same disjunction as a convex hull of the constituent constraints. (See [62] for a more extensive treatment of these methods.) The primary tool for mixed-integer convex relaxation of nonconvex functions that is relevant to our method is the McCormick Envelope, which assembles a piecewise linear outer approximation of bilinear constraints of the form $w = x \times y$ using these disjunctive constraint tools [63, 64].

3.3 Mixed-Integer Problem Formulation

We present formulations of the point cloud pose estimation problem using the framework of mixed-integer programming. We present two formulations, which differ in the model used to represent objects. We begin with a simple but inefficient object model that represents objects as a set of points sampled from the object surface. We then generalize our formulation to a more precise object model that describes objects with a set of vertices and planar faces.

3.3.1 Pose Estimation of Sampled Point Models

Given a set of scene points $S = \{s_i\}, i \in [0, N_s]$ and a set of model points $M = \{m_j\}, j \in [0, N_m]$, the generic pose estimation problem is equivalent to finding a rotation matrix R, a translation matrix T, and a correspondence matrix $C \in \{0, 1\}^{N_s \times N_m}$ that satisfy the following:

$$
\begin{aligned}
\underset{R,T,C}{\text{minimize}} \quad & \frac{1}{N_s} \sum_{i \in [0, N_s]} \left\| R s_i + T - M C^T_{i,:} \right\|_2^2 \\
\text{subject to} \quad & R \in SO(3), \\
& \sum_j C_{i,j} = 1, \ \forall i, \\
& C_{i,j} \in [0, 1], \ \forall i, j.
\end{aligned}
$$

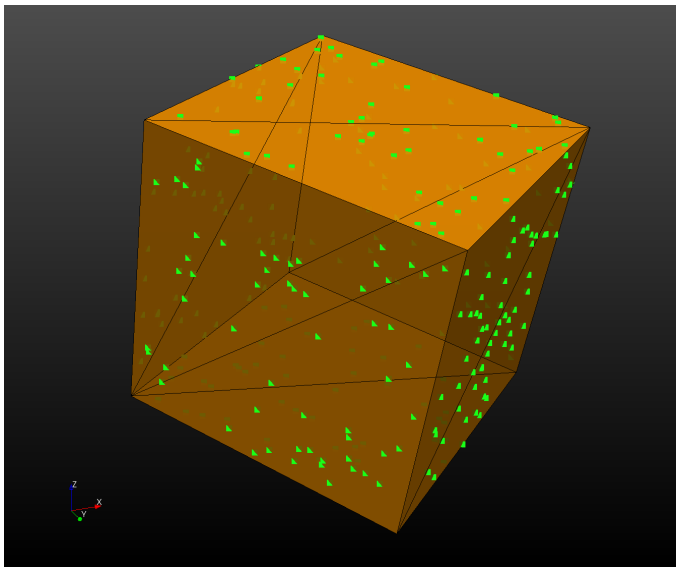

Figure 3-1: Sampled point (green) and mesh (orange) representations of a cube model. The sampled point model is generated by sampling N points from the surface of the model. Here, they were sampled randomly with N=300.

The exact layout of M and $C_{i,:}$ are chosen such that $MC_{i,:}^T$ takes the value of the model point m_j when $C_{i,j} = 1$. Rows of C summing to 1 enforces that a scene point must be corresponded to exactly one model point. A single model point is allowed to explain multiple scene points.

When employing this formulation, models to be fit in the scene cloud are represented via a set of points sampled from their surface (Figure 3-1). This formulation has the property that a valid assignment $\hat{C}$ can be used to compute an optimal translation $\hat{T}$ and rotation $\hat{R} \in SO(3)$ in closed form [65, 66]. This is the same property at the heart of the ICP algorithm [2].

Unfortunately, in this formulation, the optimal pose estimate is guaranteed to have nonzero error as scene and model points are sampled independently from the true scene and model geometry. This best-case error decreases as the model point sampling density increases, but requires quadratically more model sample points to achieve linear improvements in optimal cost. As we will see, this can cause inefficiencies for branch-and-bound while certifying optimality.

3.3.2 Pose Estimation of Mesh Models

The sampled point model representation can be naturally extended to a richer mesh model with minor modifications. Using this mesh model, the optimal pose estimate will have zero error in the absence of noise. Here, we represent the model with a collection of vertices and faces, where each face is constructed as an affine combination of a coplanar subset of vertices (Figure 3-1).

Given a model defined by N_m vertices and N_f faces, where each face is defined as an affine combination of a subset of coplanar vertices, as well as

- scene points $S = \{s_i\}, i \in [0, N_s]$,
- model vertices $M = \{m_j\}, j \in [0, N_m]$,
- a binary face membership map $F \in \{0,1\}^{1 \times N_f}$,

the generic pose estimation problem is equivalent to finding a rotation matrix R, a translation matrix T, a combination matrix $C \in \mathcal{R}^{N_s \times N_m}$, and a face correspondence matrix $f \in \{0,1\}^{N_s \times N_f}$ that satisfy the following.

$$
\begin{aligned}
\underset{R,T,C,f}{\text{minimize}} \quad & \frac{1}{N_s} \sum_{i \in [0,N_s]} \left\| Rs_i + T - MC_{i,:}^T \right\|_2^2 \\
\text{subject to} \quad & R \in SO(3), \\
& \sum_{j \in [1,N_m]} C_{i,j} = 1, \ \forall i, \\
& \sum_{k \in [1,N_f]} f_{i,k} = 1, \ \forall k, \\
& 0 \leq C_{i,j} \leq Ff_i, \ \forall i, j, \\
& f_{i,j} \in [0,1], \ \forall i, j.
\end{aligned}
$$

This formulation is fundamentally similar to the point-model form, but searches over scene-point-to-face correspondences instead of scene-point-to-model-point correspondences. Affine combination coefficients for the i^{th} scene point $C_{i,:}$ are constrained to be inactive unless one of their parent faces is active. Scene points can only correspond to a single model face.

3.4 Approximation of $R \in SO(3)$

The above formulations are nonconvex due to the binary set constraints on correspondence variables, and due to the constraint $R \in SO(3)$. Here, we discuss applicable relaxations of the rotation constraint in the context of this problem.

The techniques listed here rely on the equivalence of $R \in SO(3)$ with the set of constraints $\{R^T R = RR^T = I,\ det(R) = +1\}$. $R^T R = I$ and $RR^T = I$ equivalently encode that R is an orthogonal matrix, and $det(R) = +1$ disallows R to take determinant -1 and thus excludes reflections.

3.4.1 Convex Outer Approximations

SDP relaxations of $SO(3)$ functionally constrain the decision variables R to be within the convex hull of $SO(3)$ [52]. Critically, $R = \mathbb{0}$ is a feasible point under this relaxation. While this is not an issue for the PM problem, our problem allows for simultaneous optimization of translations, and does not constrain columns of C. Because of this, $R = \mathbb{0}$, $T = m_1$, $C = [\mathbb{1}, \mathbb{0}, ..., \mathbb{0}]$ is a trivial solution with zero error for our problem under such a liberal relaxation of $SO(3)$. This trivial solution collapses all scene points to the origin and corresponds each scene point to the same model point. Thus, for our problem, this relaxation cannot be used without additional constraints.

3.4.2 Domain Restrictions

Due to the trivial solution at $R = \mathbb{0}$, any reformulation and relaxation should not include that point. Given a valid rotation R_0, a linear constraint to accomplish this is the constraint

$$\begin{aligned} \|R - R_0\|_1 \leq & \epsilon, \\ \epsilon \leq & \min_{v \in \text{Rows and columns of } R_0} \|v\|_1 . \end{aligned}$$

Note that the L-1 norm can be implemented as a set of linear constraints with the

introduction of one slack variable per scalar absolute value term.

This approximation does not contain the entirety of $SO(3)$, and thus is only useful in select circumstances. This could be the case, for example, when the approximate object orientation is known (e.g. for a rotationally symmetric object on a tabletop). This could also be the case when searching over a finite number of partitions of $SO(3)$, as in GO-ICP; one instance of the pose estimation problem, with this form of constraint and a different R_0, would be solved per partition.

3.4.3 Piecewise Linear Envelopes of Orthogonality Constraints

This approximation builds piecewise convex outer approximations of $SO(3)$, in the spirit of the McCormick Envelope [63]. This work has been pioneered by Dai and Tedrake, and is presented in more detail in [67].

For each member of the rotation matrix $R_{i,j}$, we introduce new binary variables to assign $R_{i,j}$ to one of N_k partitions of $[-1, 1]$:

$$b^+_{i,j,k},\ k \in [1...N_k],\ b^+_{i,j,k} = 1 \implies R_{i,j} \geq k/N_k,$$
$$b^-_{i,j,k},\ k \in [1...N_k],\ b^-_{i,j,k} = 1 \implies R_{i,j} \leq -k/N_k.$$

These binary variables intentionally indicate membership in overlapping regions, with the intent that partial assignments that assign indicator variables of lower k provide information about other partial assignments that assign variables of higher k.

To begin, we enumerate relationships between $b^+_{:,:,0}$ and $b^-_{:,:,0}$ to enforce the constraint that no row (or column) of R should be within the same or opposite orthant of another row (or column) of R.

We construct simple expressions $c_{i,j,k}$ that are linear in these binary variables, which are active if and only if $R_{i,j}$ is within the corresponding partition of $[-1, 1]$.

For the row or column vector v of R, let the other row or column decision variables be v_1 and v_2. We can the expression c to construct piecewise convex outer approximations of the constraints constructing $SO(3)$. That is, for each bounding box within $[-1, 1]^3$ in which v may lie, we generate the linear expression $\gamma = A\, c_{i,:,:}$ that sums

the three membership indicators for this bounding box, and constrain:

- If this box doesn't intersect the unit sphere at all, add the constraint $\gamma \leq 2$, which constraints v from being in this orthant.
- If the bounding box intersects at a single point u, then if $\gamma = 3$, then $v = u$. Populate the orthogonality conditions $u^T v_1 = 0$, $u^t v_2 = 0$, $u \times v_1 = v_2$.
- If the bounding intersects at multiple points, then we can calculate an outer convex hull that v must lie within. We translate this hull into convex outer approximations of constraints $v^T v_1 = 0$ and $v^T v_2 = 0$ and $v \times v_1 = v_2$ based on the max possible angle between v and the orthant normal vector.

We compare this formulation to a spiritually-similar alternative that utilizes McCormick envelopes to approximate the bilinear products within the constraints $v^T v = 1$, $v^T v_1 = v^T = 0$, and $v \times v_1 = v_2$, for rows or columns v, v_1, v_2 of R, directly. Logarithmic encoding schemes can be used to make this bilinear products very efficient in terms of the number of binary variables added [68, 69]. This alternative formulation is attractive in its efficiency in terms of number of binary variables, and excellent performance in other problems, and is included as a comparison for the above formulation.

3.5 Extensions

A core strength of the mixed-integer convex formulation of pose estimation is its extensibility. In this section, we describe how to extend the model to handle the cases of outliers and multiple simultaneous objects; and how to take advantage of feasible solutions generated by other pose estimation methods.

3.5.1 Handling Outliers

Correct outlier handling is critical for object pose estimation algorithms, as point clouds in the wild invariably include unmodeled points from nearby objects and support surfaces in the scene. When the error metric in the standard pose estimation

objective (Equation 1.2) is an L-2 norm, outlier points affect the final pose estimate disproportionately. This issue is partially alleviated by moving to lower norms than L-2 [70], which we will take advantage of; but tweaking the norm used still does not address that outliers are fundamentally different than inliers.

A standard trick used in practice to identify and discount outliers is to discard the farthest N% of correspondences when computing the registration error (as used in, e.g. [24]). However, this requires estimating the *expected fraction of outliers*, which might be inconsistent across and within experiments. Our mixed-integer form gives us the flexibility to instead consider outlier rejection as an additional explicit correspondence choice.

To support outliers, we first switch from the L-2 to the L-1 norm in our error metric, so that we can include the distance to each point in the set of linear constraints. We introduce an intermediate variable ϕ_i for each scene point s_i storing the L-1 distance from s_i to the matched point on the model. Additional slack variables $\alpha^{i,l}$ are introduced to implement the $l \in 1...3$ absolute values within the L-1 norm for each scene index i. We bound ϕ_i with a constant *maximum allowed L-1 distance* ϕ_{max} as a threshold (and penalty) for classifying points as outliers. Finally, we add a new binary variable o_i for each scene point indicating that that scene point is being considered an outlier.

In the mesh-model case, we now solve (for arbitrarily large $\mathbb{M}$):

$$
\begin{aligned}
\underset{R,C}{\text{minimize}} \quad & \min \frac{1}{N_s} \sum_{i \in [0, N_s]} \phi_i \\
\text{subject to} \quad & \textit{Relaxed } R \in SO(3), \\
& \phi_i \geq \mathbb{1}^T \alpha_{i,l}, \\
& \phi_i \geq \phi_{max}\, o_i, \\
& \alpha_i \geq + \left(Rs_i + T - MC^T_{i,:}\right) - \mathbb{M} o_i, \qquad (3.1) \\
& \alpha_i \geq - \left(Rs_i + T - MC^T_{i,:}\right) - \mathbb{M} o_i, \qquad (3.2) \\
& \sum_{j \in N_m} C_{i,j} + o_i = 1, \\
& \sum_{k \in N_f} f_{i,k} + o_i = 1, \\
& 0 \leq C_{i,j} \leq F\, f_i, \\
& \phi_i,\ \alpha_i \geq 0, \\
& f_{i,j},\ o_i \in \{0, 1\}.
\end{aligned}
$$

3.5.2 Handling Multiple Objects

Using similar machinery to that employed to correspond to outliers, we can extend our formulation further to support multiple objects. We can extend the formulation to simultaneously optimize over multiple rotations and translations $\{R_1, T_1\}, ..., \{R_{N_b}, T_{N_b}\}$ for N_b separate bodies. Given a map $B \in \{0, 1\}^{N_b \times N_f}$, where the $(i, j)^{th}$ entry indicates if face j is a member of body i, we can replace constraints (3.1) and (3.2) with the disjunction

$$
\begin{aligned}
\forall i \in N_m, k \in N_b : \qquad & \alpha_i \geq + \left(R_k s_i + T - MC^T_{i,:}\right) - \mathbb{M}(1 - B_{k,:} f^T_{i,:}), \\
& \alpha_i \geq - \left(R_k s_i + T - MC^T_{i,:}\right) - \mathbb{M}(1 - B_{k,:} f^T_{i,:}).
\end{aligned}
$$

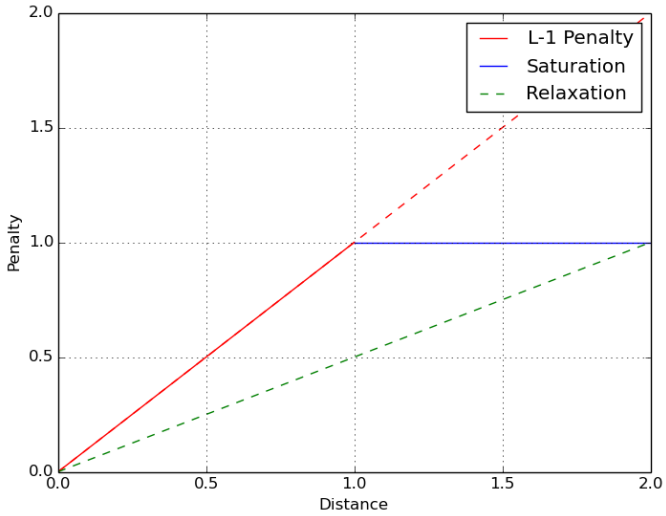

Figure 3-2: Effective error landscape under the outlier formulation, which takes the minimum of the L-1 distance (red) and a constant penalty ϕ_{max} (blue) at optimality. Because the switching between the L-1 distance and constant penalty is controlled by a big-$\mathbb{M}$ formulation, the effective error landscape when integrality constraints on o_i are relaxed is the lower convex hull of the L-1 penalty and the saturation on the domain $[0, \mathbb{M}]$. In this illustration, $\phi_{max} = 1$ and $\mathbb{M} = 2$.

where the expression $\mathbb{M}(1 - B_{k,:} f_{i,:}^T)$ deactivates the constraint if the current assignment f does not assign scene point i to a face on body k.

3.5.3 Using Other Pose Estimation Methods as a Heuristic

A benefit of optimizing directly over the fundamental problem addressed by a wide class of pose estimation methods is that we can take advantage of solutions generated by those other methods by consuming them as candidate feasible solutions. The branch and bound algorithm (and solvers that implement it) is able to asynchronously consume feasible solution guesses as nodes in the search tree. These new feasible solutions provide upper bounds on the global optimal cost, which are used to prune bad nodes. Because a significant amount of search time is spent finding better feasible solutions (as can be seen in the results in e.g. Figure 3-8), getting better feasible solutions from faster but less-consistent pose estimation methods can improve the runtime of the global optimization. This ability also means that this formulation

can be used to post-process the output of notoriously unpredictable methods, like neural networks, in order to guarantee stable results without completely discarding the efficiency of the original method.

Given the mesh model MILP formulation described above and a candidate pose $\{R_0, T_0\}$ generated by any method, one can extract C, f, ϕ, and α via closest-point queries against the mesh models. The means to extract the variables in the rotation approximation vary, but in this case of the piecewise-linear convex approximations, the value of R directly determines which binary variables should be active.

3.6 Characterization

The convergence rate of MILP branch and bound is dependent on both how quickly good feasible solutions can be found, and on the tightness of the continuous relaxation of the MILP. Quickly finding good feasible solutions is equivalent to rapidly providing a tight upper bound on the global optimal cost. Correspondingly, tight continuous relaxations reduce the depth of the search tree of partial relaxations when proving a lower bound. Myriad local and approximate global techniques provide means for rapidly producing good feasible solutions – but analysis of the convergence of the lower bound must be more fundamental. Here we present several quantitative experiments intended to characterize the quality of the continuous relaxation of our formulation.

3.6.1 Partial Assignment of Correspondences

The pose estimation problem has a well-known property that fully specifying the correspondences is equivalent to finding the transformation, as one can be derived from the other. To investigate the degree to which this property manifests in our formulation, we performed quantitative experiments in which we removed all constraints on R, and solved the problem with a varying number of correspondences being correctly specified via equality constraints on the appropriate variables. Leaving R unconstrained simulates early relaxations in the branch and bound process, before binary variables in the MILP relaxation of $R \in SO(3)$ are assigned.

Quantitative experiments reveal that correct assignment of a handful of correspondences is sufficient for the relaxation to converge to the correct solution, *even in the absence of constraints on R*. As shown in Figure 3-3, the number of correspondence assignments requires appears scales much more slowly than the complexity of the rest of the problem.

This behavior makes sense when the dimensionality of the core pose estimation problem is considered. While the problem we solve is very high-dimensional, it is ultimately an overparameterization of the original problem that searches over just R and T (Eq. 1.1). Thus, specifying $size(R) + size(T) = 12$ correct correspondences should certainly completely specify the solution. However, this relationship is broken by the presence of outliers – because an assignment as an outlier comes with a constant penalty, there is a break-even point at which matching the partially assigned correspondences becomes better than maintaining an all-zero rotation matrix that increases with both the total number of scene points and fraction of outliers. Shown in Figure 3-3, our experimental results support these claims, given small error arising from problem-specific degeneracies.

Just as specifying correct correspondences cause the relaxations to quickly converge to the correct solution, specifying *incorrect* correspondences must constrain the relaxations to have high optimal values for branch and bound to be efficient. (Otherwise, many correspondences would need to be assigned before branches had sufficiently high lower bounds that they could be discarded.) We repeated the same experiment as above, instead specifying an incremental number of randomly chosen *incorrect* correspondences. The results (Figure 3-4) show a linear scaling of relaxation optimal value with bad correspondence assignments.

In combination, these results indicate that much of the attractive structure of the pose estimation problem is maintained in our formulation. These results indicate that we should expect branch and bound search trees to terminate after an approximately constant depth – a critical property that lends hope to the idea of tackling this combinatorial problem in its full form.

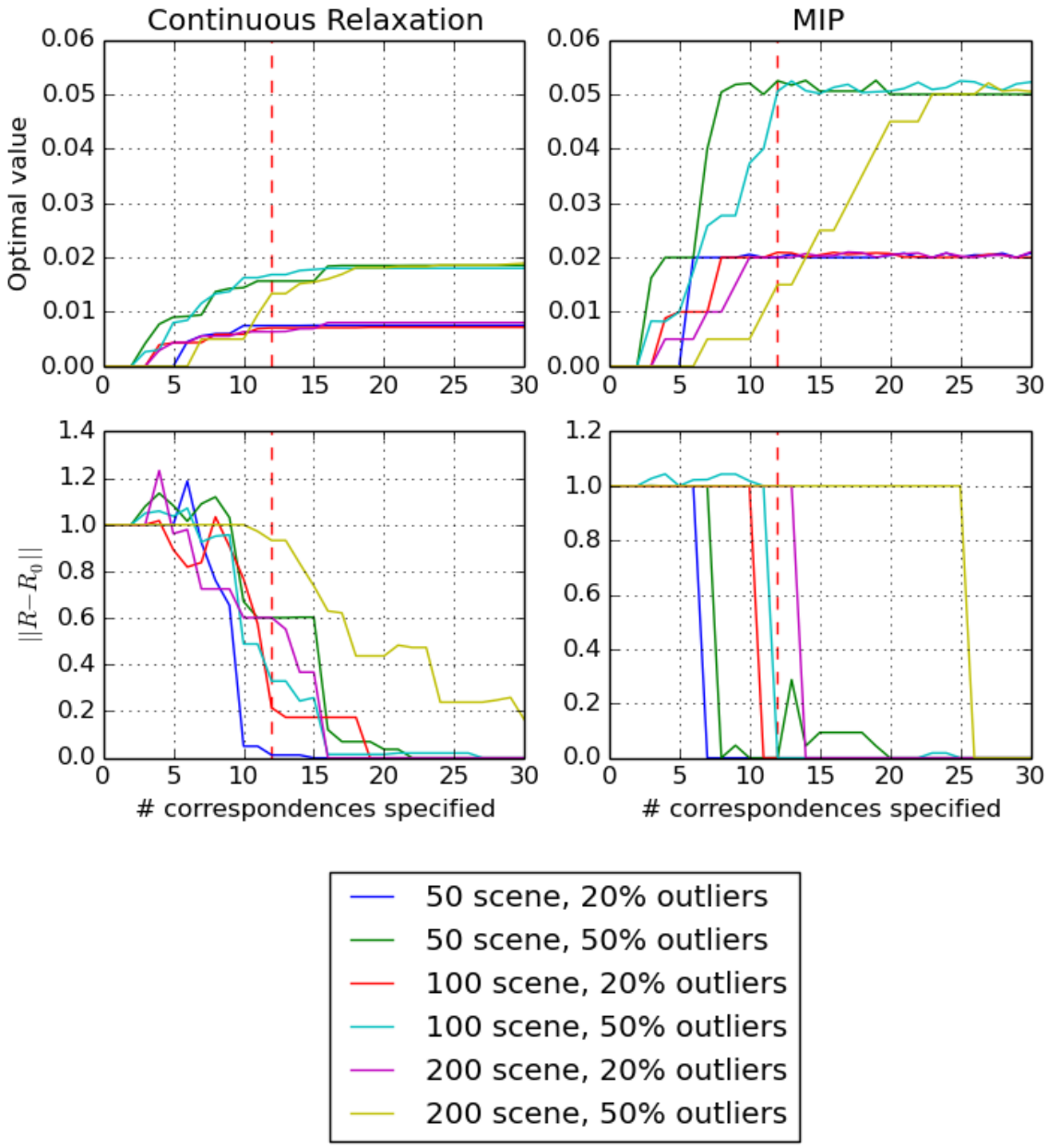

Figure 3-3: Optimal value (first row) and difference of rotation from the from ground truth solution R_0 (second row) versus the number of correct correspondence assignments pre-specified. Each line reports the same experiment run on a different problem size and complexity, in terms of number of scene points and percent outliers. Results are reported for the continuous relaxation (left), and full MIP optimal solution (right). The vertical dashed red line indicates the hypothesized break point at which we hypothesize the solution should become fully specified. We hypothesize lower bounds are consistently a multiplicative factor below upper bounds due to the relaxation gap created by the outlier formulation shown in Figure 3-2.

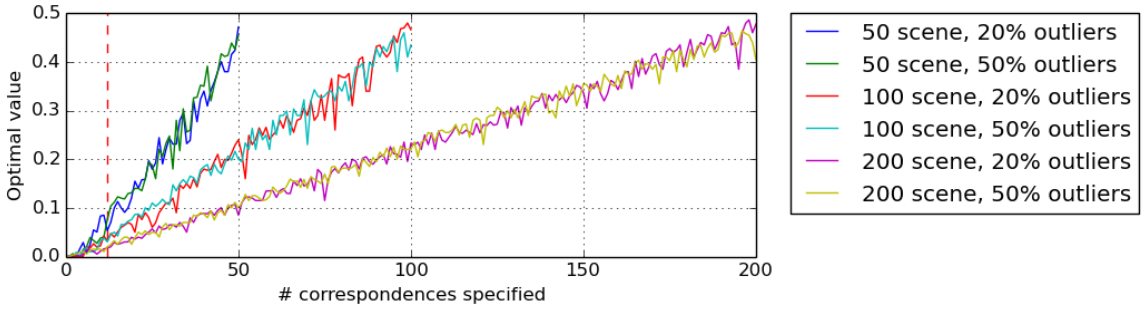

Figure 3-4: Optimal value versus the number of incorrect correspondence assignments pre-specified, for the continuous relaxation of the problem. Each line reports the same experiment run on a different problem size and complexity, in terms of number of scene points and percent outliers. The vertical dashed red line indicates the hypothesized break point at which we hypothesize the solution should become fully specified.

3.6.2 Partial Assignment of Rotation

Given the assumption that a significant fraction of the scene points are inliers and are accurately explained by the model, one might hope that the true solution would remain the optimal solution under very lenient relaxations of the complete problem. To verify whether this is the case, we construct a model problem using the mesh-model MILP formulation with a *domain-restricted* rotation R. In this model, members of R are free, but rows and columns of R are constrained within an L-1 norm ϵ of the ground truth R_0. By varying ϵ, we demonstrate that the integer feasible optimal solution is tight until ϵ is sufficiently large that the feasible region includes the trivial solution that allows a row of R to take a value of 0. The continuous relaxation does not share this properly. This result is replicated in the case of injected noise with no outliers, and in the presence of outliers. (See Figure 3-5.)

3.7 Results

We present several experiments in which we verify our formulation on synthetic data. To perform these experiments, we implemented both formulations in both C++ and Julia, relying on the Drake [71] and JuMP [72] symbolic optimization libraries respectively. We used Gurobi 7.0.2 [58] as a backend to solve the resulting mixed-integer

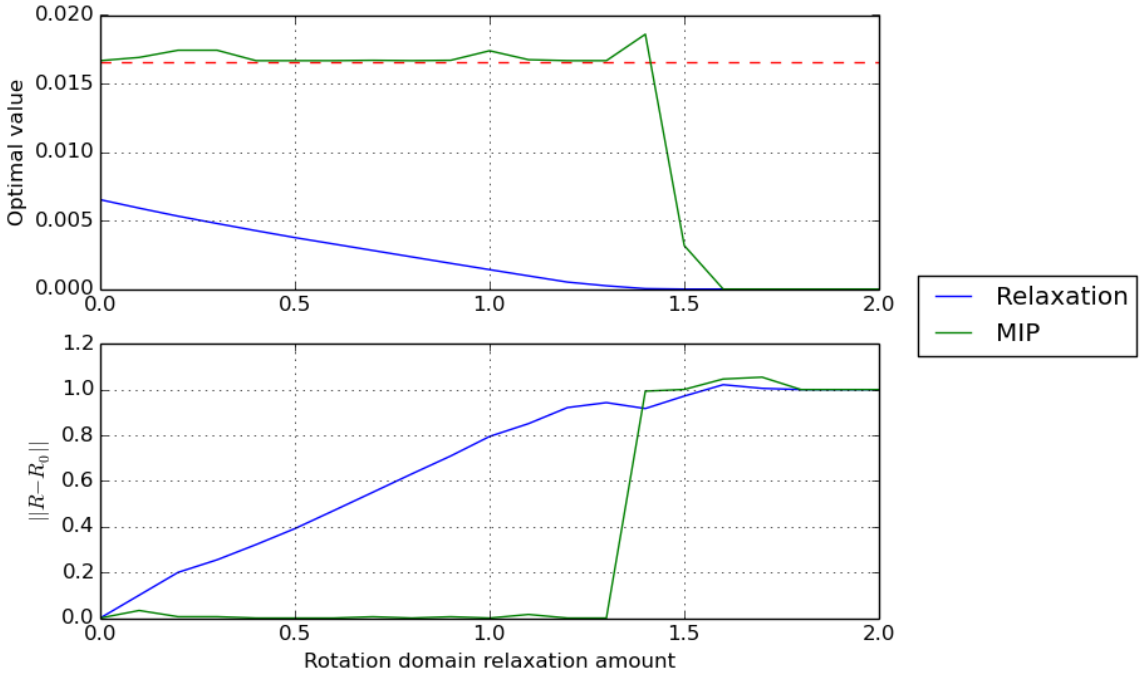

Figure 3-5: Optimal value (top) and difference from ground truth solution (bottom) versus the maximum allowed deivation of R from R_0, in terms of row- and column-wise L-1 norm. The integer optimal solution remains close to the true solution until the relaxation includes the trivial solution when at least one row of $R = 0$. However, the continuous relaxation does not share this properly. These plots were generated from a problem with 24 scene points and 4 outliers.

programs, as well as their relaxations.

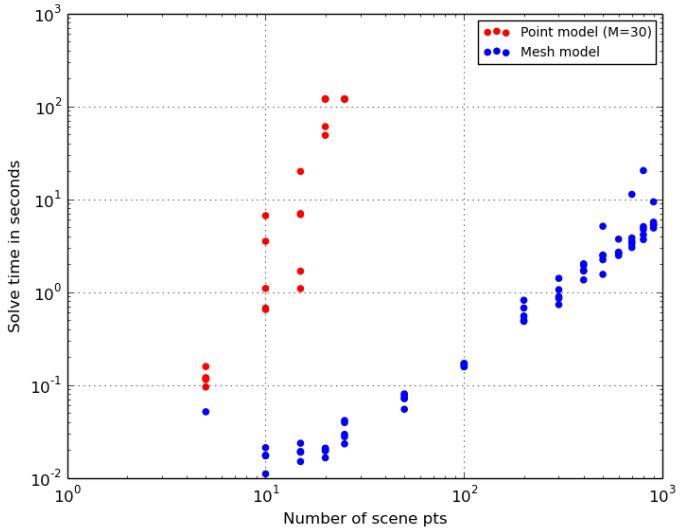

Figure 3-6: Point and mesh model formulation solve times for different numbers of scene points. The scene points were sampled uniformly from the surface of a cube, with no outliers. The point model sampled 30 points randomly from the surface of the same cube model. The mesh model consisted of 12 nonoverlapping triangles completely representing the cube. Each point is a sample solve time.

In order to use the point model formulation, we would need to sample a large number of points from the model surface to generate an accurate representation. However, this presents an unfortunate tradeoff: more model points leads to lower optimal registration errors and hence less time moving the lower bound, but more model points also increases the problem complexity. Figure 3-6 illustrates that this effect causes the point model formulation to be far slower than the mesh model formulation for all scene complexities. For that reason, in all following experiments, we used the MILP mesh model formulation.

3.7.1 Comparison of Rotation Approximations

To compare the two rotation approximations we considered, we compared performance on a simple single-object test case for the two methods. We generated a synthetic point cloud from a cube model with a side length of 1 unit. We generated 15 scene points, with 10 sampled randomly from the surface of the cube at its ground

truth pose, and 5 more generated randomly in the area around the cube. We included outliers in this test case to illuminate how the effect of the rotation approximation on the progress of the upper *and lower* bounds – otherwise, the optimal error would be close to the trivial lower bound of 0. An optimal fit in this configuration has an optimal average saturated L-1 error of 0.033: the $\frac{2}{3}$ of points that are inliers have L-1 error of 0, and the $\frac{1}{3}$ of points that are outliers have L-1 error of $\geq \phi_{max} = 0.1$ by construction.

The MILP mesh model formulation converged to the optimal solution and certified its global optimality to within a MIP gap of 5% under both rotation approximations (Figures 3-7 and 3-8). The logarithmic McCormick envelope approximation used 3 bins (2 binary variables) per bilinear term in $R^T R = I$ and $R_1 \times R_2 = R_3$. The per-element linear binning used 4 binary variables per element of R.

In the logarithmic McCormick envelope approximation, the largest elementwise infeasibility of $R^T R = I$ was 0.165 and $det(R)$ was 1.031. In the per-element linear binning, the largest elementwise infeasibility of $R^T R = I$ was 0.020 and $det(R)$ was 1.002. The per-element linear binning is a clear winner in these experiments, in terms of both the accuracy of the approximation, and convergence time of the resulting formulation.

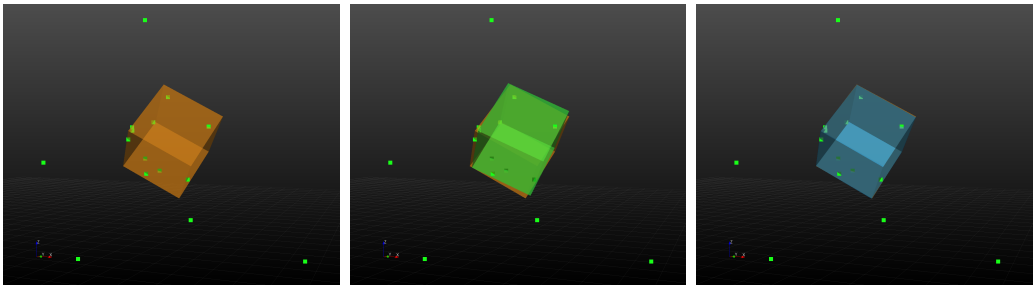

Figure 3-7: Pose estimates produced by our MILP mesh model formulation for a cube model of 12 triangular faces, given 15 scene points with 5 outliers. Both solutions shown here have optimal cost that matches the optimal cost of the ground truth solution. Optimality of these solutions were certified to a MIP gap of 5%. **Left:** Ground truth pose. **Middle:** Pose estimate in green using 2D logarithmic binning for approximation of $R \in SO(3)$, shown over the ground truth pose in orange. **Right:** Pose estimate using per-element linear binning for approximation of $R \in SO(3)$.

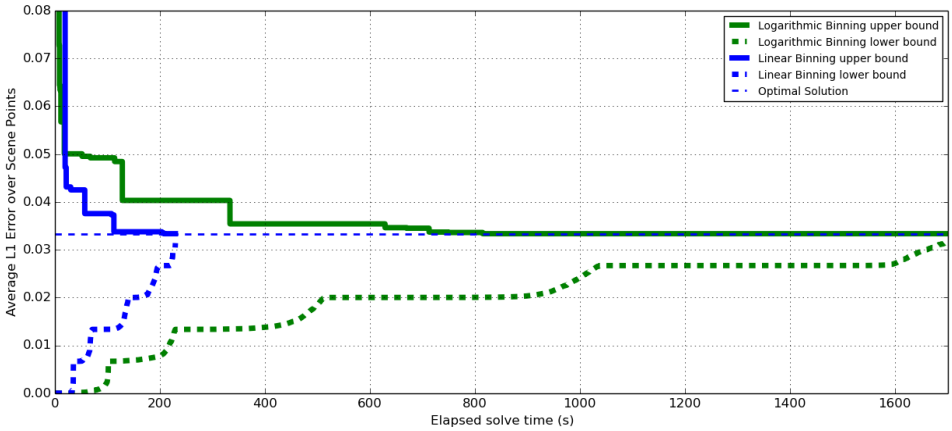

Figure 3-8: Convergence times of the upper and lower bounds across time for the solutions shown in Figure 3-7.

3.7.2 Outlier Rejection

To highlight the outlier rejection capability of our formulation, we generated synthetic point clouds from a cube model with a side length of 1 unit. We generated 100 scene points, with only 50%, 20%, and 10% of them sampled randomly from the surface of the cube at its ground truth pose in three test cases. We used the mesh model MILP formulation; however, to avoid unreasonably long runtimes, we had to constrain rotations and limit the search to be over translations and correspondences. R was thus constrained to take the value of the ground truth rotation.

In each of the three test cases, the optimization converged on the correct optimal solution and certified its optimality within an MIP gap of 5% (Figures 3-9 and 3-10).

3.7.3 Multiple Models

To highlight the extension of our formulation to multiple models, we generated a synthetic point cloud from two differently shaped box models at different poses. We generated 100 scene points with no outliers. We used the mesh model MILP formulation; however, to avoid unreasonably long runtimes, we had to constrain rotations and limit the search to be over translations and correspondences. R_k was thus constrained to take the value of the ground truth rotations of model k.

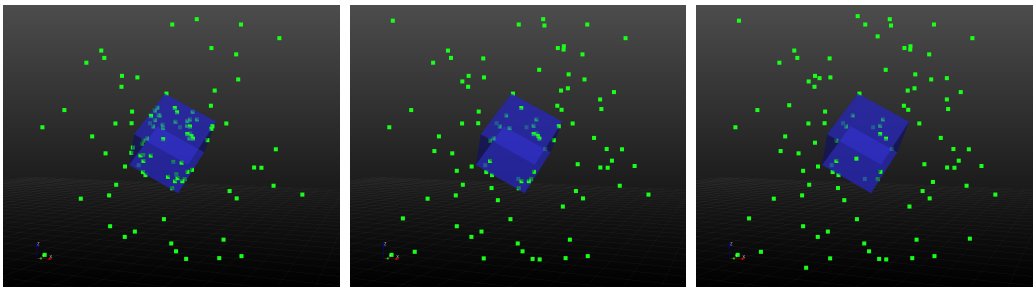

Figure 3-9: Pose estimates produced by our MILP mesh model formulation for a cube model of 12 triangular faces, given 100 scene points, with a varying number of them being outliers: **Left:** 50% outliers, **Middle:** 80% outliers, and **Right:** 90% outliers. Rotations were frozen to the ground truth rotation in order to produce these solutions in reasonable time. All solutions shown here have optimal cost that matches the optimal cost of the ground truth solution and align with the ground truth pose. Optimality of these solutions were certified to a MIP gap of 5%.

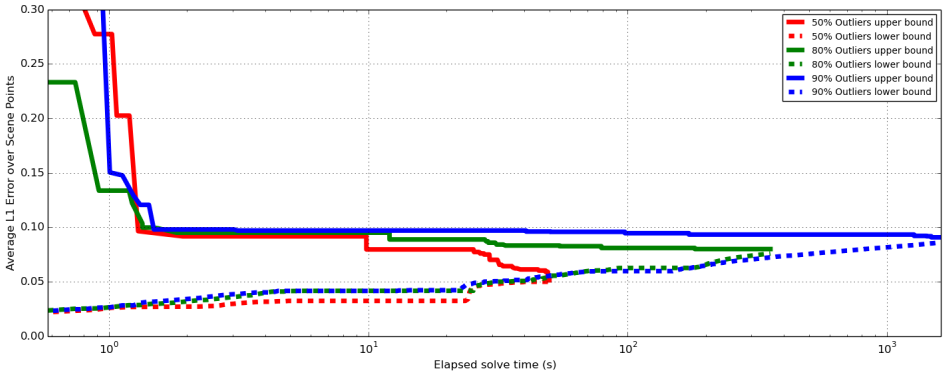

Figure 3-10: Convergence times of the upper and lower bounds across time for the solutions shown in Figure 3-9. Note log scaling on the x-axis.

The optimization converged in 1038s to the optimal solution. Because the L-1 error at this solution is 0, optimality is certified trivially.

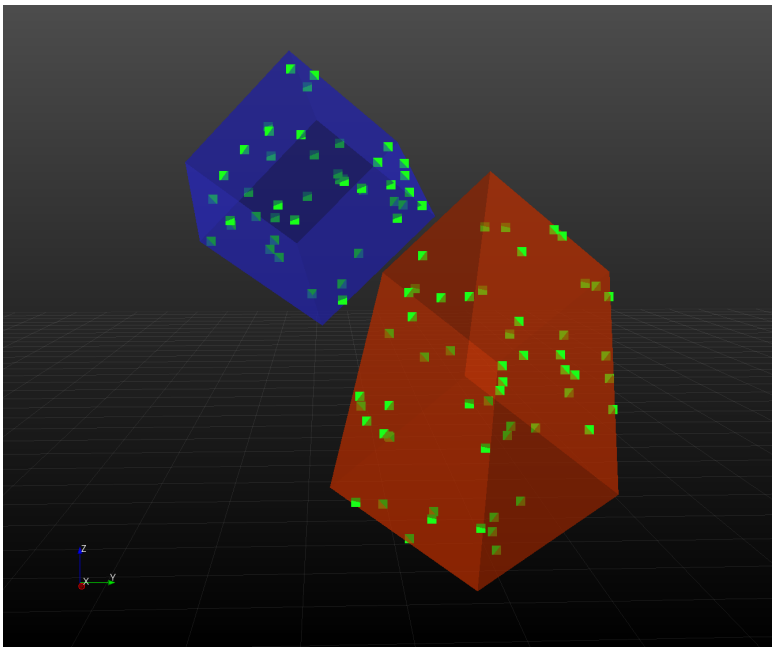

Figure 3-11: Pose estimates produced by our MILP mesh model formulation simultaneously fitting two box models to 100 scene points with no outliers. Rotations were frozen to the ground truth rotations in order to produce these solutions in reasonable time.

3.7.4 Upper Bounds from ICP

To demonstrate that solutions generated from other efficient but non-global methods can be leveraged to make our global optimization faster, we implemented an ICP-based heuristic for generating candidate feasible solutions online during the optimization. This procedure is directly inspired by GO-ICP [24]. Our solver maintains a queue of feasible solutions found by the branch and bound algorithm, and runs point-to-plane ICP with proportional outlier rejection on each feasible solution in a parallel thread alongside the global optimization solver. If the ICP produces a solution better than the best currently held by the solver, the ICP solution is handed to the solver as a heuristically-derived feasible solution. This procedure significantly improves runtime, as is shown in Figure 3-12.

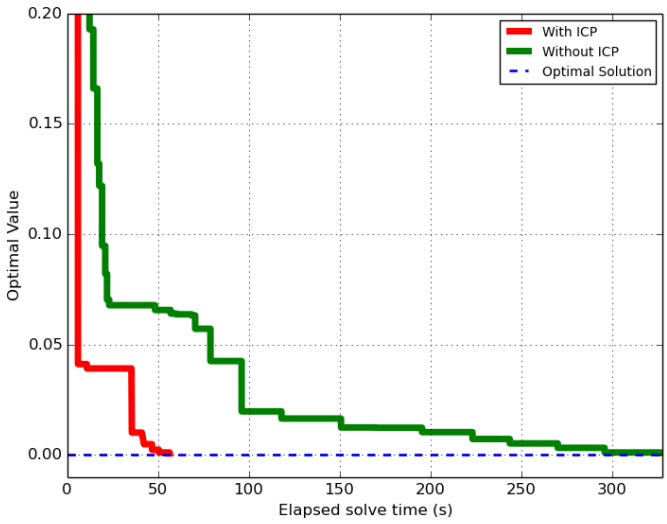

Figure 3-12: Comparison of the upper bound convergence behavior of the MILP mesh formulation with 50 scene points and 0 outliers fitting a box model, with and without an ICP algorithm generating novel feasible solutions in parallel. The lower bound is omitted, as it is trivially 0 for the 0 outlier case.

3.8 Discussion

The formulations we present can be used to find certifiably globally optimal solutions for small numbers of scene points and outliers, even in the face of combinatorial complexity. The solver is capable of finding and certifying the right solution, even in very high outlier ratios, and can optimize with multiple objects seamlessly. Convergence history results demonstrate that, even if the search gets stuck in poor feasible solutions for long periods of time, they are always eventually escaped.

However, the complexity of the mixed-integer program ultimately grows too quickly for this technique to scale to full point clouds from experimental data. While the roughly constant depth of the search tree with respect to partial correspondence assignments is hopeful, the breadth of the tree still grows with the number of possible assignments. While good feasible solutions relatively are easy to find for this problem, the certification of global optimality relies on verifying that all possible branches of the search tree will have optimal value worse than the incumbent optimal feasible solution. Even if the depth of the search tree is fixed at N_d, this requires exploring

$\binom{N_s}{N_d}$ nodes, which is polynomial but high-order in N_s ($O(N_s^{N_d})$). This lower bound search is harder for larger amounts of noise or larger numbers of outliers – a problem which is replicated in GO-ICP and other branch-and-bound based solvers.

The regime of tens of points appears to be the sweet spot for this algorithm. While this is far fewer points than are present in a raw point cloud, it is a reasonable number of features to extract from that point cloud using any number of standard geometric feature extractor. A clear direction for extension of these results would be to apply it to find globally optimal correspondences of high-dimensional features; the higher dimension features may additionally improve the relaxations by allowing higher penalties on outliers.

That this technique can so easily incorporate hypotheses from other methods makes it a candidate for being an offline *verification* technique for the results from other efficient but inconsistent pose estimation methods. This functionality is critical when considering the kinds of highly ambiguous point clouds that result from highly cluttered scenes, and from tactile sensing. The formulation and its branch and bound solution has the significant advantage that, by examining partial relaxations of the problem, it either verifies that a solution is globally optimal, or provides a search region that may contain a better solution. GO-ICP is founded on exactly this principle of pairing efficient local search with high-level guidance from branch and bound [24]. Our formulation provides a direction to extend that powerful search strategy to a more general and extensible formulation of the point cloud pose estimation problem.

Chapter 4

Discussion and Future Work

In this thesis, we have presented work focused on advancing the state of the art in pose estimation of objects with tactile sensing in the loop. The object tracker presented in Chapter 2 provides accurate pose estimates with a sufficiently general object model to be applied for a wide range of manipulation tasks. Many other techniques for object tracking with tactile sensors consider the tactile sensing as a source of discriminative or dynamic information (as in [8, 14, 15]), which necessitates a complex and often discontinuous measurement model. We have demonstrated that considering this problem from a purely geometric perspective is sufficient to handle complex contact interactions through interplay of geometric free space, nonpenetration, and positive return models described in Chapter 2; but remains simple enough to be well-approximated as a mixed-integer convex program, as described in Chapter 3.

Further, our investigation of the global optimization of the general point-cloud pose estimation problem illuminates the notion that there is very strong structure in very loose relaxations of the problem. We are hopeful that further study of the properties of these partial relaxations might allow them to be leveraged to give global guidance to the many powerful but inconsistent local methods for pose estimation.

4.1 Directions Forward

4.1.1 Different Object Models

One significant limitation of the work presented in this thesis is that the rigid body object model used in both the tracker and global optimization precludes extension to non-rigid bodies. Many of the toughest unsolved problems in robotics relate to interaction with either novel or deformable object instances; in both cases, pre-specified rigid body models are a bad fit. This is fundamentally a modeling problem: given an object model that can scalably encode the state of deformable bodies in a way that allows efficient signed distance calculations to the non-rigid surfaces, our tracking framework could be fairly simply extended to this case. Possible models include highly-actuated finite element models [5] or implicit surface representations [73]. Global optimization over these non-rigid models may be more difficult, though results from the non-rigid registration and shape matching community (e.g. [38]) are hopeful.

4.1.2 Scaling of Global Optimization

While our results in Chapter 3 indicate that our global optimization strategy for pose estimation works on small problems, it suffers from serious scaling issues and is not yet practical for use on raw point cloud data. As discussed in Chapter 3, one possible avenue for deployment would be to globally optimize over downsampled features from the raw point cloud data, and corresponding features from the models, as in [51] and [56].

From a more fundamental perspective, a significant factor in the runtime of branch and bound search is the depth of the search tree – or equivalently, how many integer assignments must be made before the continuous relaxation of the solution is either integer feasible, or worse than another known integer feasible solution. In this problem, there are at least two avenues for decreasing the required search depth: addressing the trivial solutions at $R = \mathbb{0}$, and reducing the slack in the outlier formulation.

As observed in Chapter 3, sufficiently loose relaxations of the point-cloud pose

estimation problem have an invalid trivial solution that assigns an all-zero rotation matrix. This trivial solution relies on corresponding all scene points to the same model point $\hat{m}$, and setting $T = \hat{m}$. Our results from Figure 3-5 suggest that removing this trivial solution leaves the true optimal solution as the next best solution (though we expect there to be an additional relationship depending on noise and outliers). It is possible that supplying additional constraints to make such degenerate solutions infeasible might produce a much better continuous relaxation. We have experimented with several types of constraints that constrain the number of correspondences allowed to each model point, as well as geometrically-derived mutual exclusion constraints between far away scene points, but have made only limited progress. Significantly more exploration is needed to fully map the space of possible additional constraints in this spirit.

As demonstrated in Figure 3-2 and evident in the results in Figure 3-5, our outlier formulation produces a fairly loose continuous relaxation that grows worse as the big-$\mathbb{M}$ constant is increased. Given a formulation that requires explicit correspondence decisions for every scene points, this may be unavoidable. Alternative formulations with different underlying objectives – for example, ones motivated by shape matching [38] or vertex-cover [56] approaches – may sidestep this problem.

4.1.3 Multi-Hypothesis Tracking

Our object tracking method focuses on maintaining a single hypothesis of object pose, which changes gradually and is updated as frequently as possible. The global optimization method we provide, as well as the numerous others we have surveyed, can produce dramatically different pose estimates from those coming out of the tracker, depending on the basin of attraction that the tracker is operating in. To unify these two approaches, careful design is required to construct a system capable of marrying occasional pose hypotheses from the global optimizers with the powerful local refinement of a pose tracker.

We propose a multi-hypothesis tracking (MHT) framework [74] as a good fit for this problem. MHT focuses on maintaining a fairly small number of hypotheses to

approximate an underlying multimodal distribution of possible poses. Intentional modeling of the history of each pose track, along with this smaller number of simultaneous hypotheses, sets MHT apart from a particle filter. The small number of hypotheses would make the parallel execution of our pose tracker, which runs efficiently but not sufficiently quickly for massive parallelization, tractable. However, careful decisions must be made concerning which pose tracks are worth keeping, and where new tracks should be initialized – questions that could be informed by our results in global optimization.

4.2 Conclusion

Pose estimation must take advantage of both the power of local approaches, and the consistency of exhaustive global optimization, in order to produce reliable pose estimates from the hardest cluttered, occluded, and partial scenes. These challenges are particularly critical when attempting to leverage the sparse information available from tactile sensors. We have demonstrated progress in both practical local tracking of objects given tactile information, and fundamental understanding of what it means to locate an object in a point cloud, in order to advance the dream of robots courageously, intentionally, and safely interacting with any object in the world.

Bibliography

[1] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: A comparison of four major algorithms. *Mach. Vision Appl.*, 9(5-6):272–290, March 1997.

[2] Paul J. Besl and Neil D. McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[3] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

[4] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Dart: dense articulated real-time tracking with consumer depth cameras. *Autonomous Robots*, 39(3):239–258, 2015.

[5] John Schulman, Alex Lee, Jonathan Ho, and Pieter Abbeel. Tracking deformable objects with point clouds. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1130–1137. IEEE, 2013.

[6] 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 *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2405–2412. IEEE, 2012.

[7] Matthew Klingensmith, Thomas Galluzzo, Christopher M Dellin, Moslem Kazemi, J Andrew Bagnell, and Nancy Pollard. Closed-loop servoing using real-time markerless arm tracking. 2013.

[8] Tanner Schmidt, Katharina Hertkorn, Richard Newcombe, Zoltan Marton, Michael Suppa, and Dieter Fox. Depth-based tracking with physical constraints for robot manipulation. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 119–126. IEEE, 2015.

[9] Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):417–487, 2011.

[10] Justin Solomon, Gabriel Peyré, Vladimir G Kim, and Suvrit Sra. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG)*, 35(4):72, 2016.

[11] SynTouch. Biotac. http://www.syntouchllc.com/Products/BioTac/.

[12] Right Hand Robotics. Takktile products. http://www.takktile.com/product:all.

[13] Li Zhang and Jeffrey C Trinkle. The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3805–3812. IEEE, 2012.

[14] Shuai Li, Siwei Lyu, and Jeff Trinkle. State estimation for dynamic systems with intermittent contact. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3709–3715. IEEE, 2015.

[15] Michael C Koval, Nancy S Pollard, and Siddhartha S Srinivasa. Pose estimation for planar contact manipulation with manifold particle filters. *The International Journal of Robotics Research*, 34(7):922–945, 2015.

[16] Matthew Klingensmith, Michael C Koval, Siddhartha S Srinivasa, Nancy S Pollard, and Michael Kaess. The manifold particle filter for state estimation on high-dimensional implicit manifolds. *arXiv preprint arXiv:1604.07224*, 2016.

[17] Micah K Johnson and Edward H Adelson. Retrographic sensing for the measurement of surface texture and shape. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1070–1077. IEEE, 2009.

[18] Rui Li, Robert Platt, Wenzhen Yuan, Andreas ten Pas, Nathan Roscup, Mandayam A Srinivasan, and Edward Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3988–3993. IEEE, 2014.

[19] Rui Li et al. *Touching is believing: sensing and analyzing touch information with GelSight*. PhD thesis, Massachusetts Institute of Technology, 2015.

[20] Radhen Patel and Nikolaus Correll. Integrated force and distance sensing using elastomer-embedded commodity proximity sensors. In *Proceedings of Robotics: Science and Systems*, 2016.

[21] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[22] Chavdar Papazov and Darius Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. *Computer Vision–ACCV 2010*, pages 135–148, 2011.

[23] Nicolas Mellado, Dror Aiger, and Niloy J Mitra. Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*, volume 33, pages 205–215. Wiley Online Library, 2014.

[24] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: a globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2016.

[25] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[26] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 998–1005. Ieee, 2010.

[27] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.

[28] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European Conference on Computer Vision*, pages 356–369. Springer, 2010.

[29] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[30] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012.

[31] Ceyhun Burak Akgül, Bülent Sankur, Yücel Yemez, and Francis Schmitt. 3d model retrieval using probability density-based shape descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1117–1133, 2009.

[32] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[34] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[35] Jay M Wong, Vincent Kee, Tiffany Le, Syler Wagner, Gian-Luca Mariottini, Abraham Schneider, Lei Hamilton, Rahul Chipalkatty, Mitchell Hebert, David

Johnson, et al. Segicp: Integrated deep semantic segmentation and pose estimation. *arXiv preprint arXiv:1703.01661*, 2017.

[36] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[37] Carl Olsson, Fredrik Kahl, and Magnus Oskarsson. Branch-and-bound methods for euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):783–794, 2009.

[38] Haggai Maron, Nadav Dym, Itay Kezurer, Shahar Kovalsky, and Yaron Lipman. Point registration via efficient convex relaxation. *ACM Transactions on Graphics (TOG)*, 35(4):73, 2016.

[39] Micah K Johnson, Forrester Cole, Alvin Raj, and Edward H Adelson. Microgeometry capture using an elastomeric sensor. In *ACM Transactions on Graphics (TOG)*, volume 30, page 46. ACM, 2011.

[40] Wenzhen Yuan, Rui Li, Mandayam A Srinivasan, and Edward H Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 304–311. IEEE, 2015.

[41] Gregory Izatt, Geronimo Mirano, Edward Adelson, and Russ Tedrake. Tracking objects with point clouds from vision and touch. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017.

[42] François Pomerleau, Francis Colas, Roland Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, 2015.

[43] Sturm J. Kerl C. Kahl F. & Cremers D. Bylow, E. Real-time camera tracking and 3d reconstruction using signed distance functions. In *Robotics: Science and Systems (RSS) Conference 2013*, volume 9, 2013.

[44] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015.

[45] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J Leonard, and John McDonald. Real-time large-scale dense rgb-d slam with volumetric fusion. *The International Journal of Robotics Research*, 34(4-5):598–626, 2015.

[46] Nawid Jamali, Marco Maggiali, Francesco Giovannini, Giorgio Metta, and Lorenzo Natale. A new design of a fingertip for the icub hand. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2705–2710. IEEE, 2015.

[47] G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000.

[48] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[49] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. Real-time human pose tracking from range data. In *European conference on computer vision*, pages 738–751. Springer, 2012.

[50] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision*, pages 766–782. Springer, 2016.

[51] Natasha Gelfand, Niloy J Mitra, Leonidas J Guibas, and Helmut Pottmann. Robust global registration. In *Symposium on geometry processing*, volume 2, page 5, 2005.

[52] Mica Arie-Nachimson, Shahar Z Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 81–88. IEEE, 2012.

[53] Kunal N Chaudhury, Yuehaw Khoo, and Amit Singer. Global registration of multiple point clouds using semidefinite programming. *SIAM Journal on Optimization*, 25(1):468–501, 2015.

[54] Richard I Hartley and Fredrik Kahl. Global optimization through rotation space search. *International Journal of Computer Vision*, 82(1):64–79, 2009.

[55] Hongdong Li and Richard Hartley. The 3d-3d registration problem revisited. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[56] Olof Enqvist, Klas Josephson, and Fredrik Kahl. Optimal correspondences from pairwise constraints. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1295–1302. IEEE, 2009.

[57] George L Nemhauser and Laurence A Wolsey. Integer programming and combinatorial optimization. *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, 20:8–12, 1988.

[58] Gurobi Optimization Inc. Gurobi optimizer reference manual, 2016.

[59] Arthur Richards, John Bellingham, Michael Tillerson, and Jonathan How. Coordination and control of multiple uavs. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 4588, 2002.

[60] Robin Deits and Russ Tedrake. Footstep planning on uneven terrain with mixed-integer convex optimization. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 279–286. IEEE, 2014.

[61] Andrés Klee Valenzuela. *Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain*. PhD thesis, Massachusetts Institute of Technology, 2016.

[62] Arthur Richards and Jonathan How. Mixed-integer programming for control. In *American Control Conference, 2005. Proceedings of the 2005*, pages 2676–2683. IEEE, 2005.

[63] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part iâĂŤconvex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

[64] Scott Kolodziej, Pedro M Castro, and Ignacio E Grossmann. Global optimization of bilinear programs with a multiparametric disaggregation technique. *Journal of Global Optimization*, 57(4):1039–1063, 2013.

[65] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987.

[66] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.

[67] Hongkai Dai and Russ Tedrake. Global inverse kinematics. 2017.

[68] Juan Pablo Vielma and George L. Nemhauser. *Modeling Disjunctive Constraints with a Logarithmic Number of Binary Variables and Constraints*, pages 199–213. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[69] Joey Huchette and Juan Pablo Vielma. Small independent branching formulations for unions of v-polyhedra. *arXiv preprint arXiv:1607.04803*, 2016.

[70] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse iterative closest point. In *Computer graphics forum*, volume 32, pages 113–123. Wiley Online Library, 2013.

[71] Russ Tedrake and the Drake Development Team. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems, 2016.

[72] Iain Dunning, Joey Huchette, and Miles Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[73] Jonathan C Carr, Richard K Beatson, Jon B Cherrie, Tim J Mitchell, W Richard Fright, Bruce C McCallum, and Tim R Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM, 2001.

[74] Roy L Streit and Tod E Luginbuhl. Probabilistic multi-hypothesis tracking. Technical report, DTIC Document, 1995.