Optimization Techniques for Trustworthy 3D Object Understanding

by

Lorenzo Franceschini Shaikewitz

B.S. Mechanical Engineering, Caltech, 2023

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2025

Authored by: Lorenzo Franceschini Shaikewitz
Department of Aeronautics and Astronautics
May 16, 2025

Certified by: Luca Carlone
Associate Professor of Aeronautics and Astronautics, Thesis Supervisor

Accepted by: Jonathan How
Richard Cockburn Maclaurin Professor in Aeronautics and Astronautics
Chair, Graduate Program Committee

Optimization Techniques for Trustworthy 3D Object Understanding

by

Lorenzo Franceschini Shaikewitz

Submitted to the Department of Aeronautics and Astronautics
on May 16, 2025 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

ABSTRACT

Autonomous machines rely on reliable 3D object understanding to interpret and interact with their environment. We consider two tightly coupled 3D object understanding problems. Shape estimation seeks a consistent 3D model of an object given sensor data and some set of priors. Pose estimation seeks an estimate of the object's position and orientation relative to an invariant shape frame. In general, these problems are non-convex and thus difficult to solve accurately. We present algorithms which nonetheless solve shape and pose estimation efficiently and with assurances in the sense of optimality, uncertainty, or speed.

We begin in the multi-frame tracking setting, where we propose the certifiably optimal estimator CAST* for simultaneous shape estimation and object tracking. We use 3D keypoint measurements extracted from an RGB-D image sequence to phrase estimation as fixed-lag smoothing, and impose temporal constraints to enforce rigidity and smooth the motion model. Despite the non-convexity of this problem, we solve it to certifiable optimality using a small-size semidefinite relaxation. We also present compatibility-based outlier rejection scheme to handle outliers, and evaluate the proposed approach on synthetic and real data.

Next, we focus on object pose estimation with known shape from a single RGB image. Assuming only bounded noise on 2D keypoint measurements (*e.g.,* from conformal prediction), we derive an estimator for the most likely object pose which uses a semidefinite relaxation to initialize a local solver. We pair this with an efficient uncertainty estimation routinue which relies on a generalization of the S-Lemma to propagate keypoint uncertainty to high-probability translation and rotation bounds. The high-probability bounds hold regardless of the accuracy of the pose estimate, and are reasonably tight when tested on the LineMOD-Occluded dataset.

Lastly, we propose a sub-millisecond solution to simultaneous estimation of object shape and pose from a single RGB-D image. Our approach converts the first-order conditions of the non-convex optimization problem to a nonlinear eigenproblem in the quaternion representation of orientation. We use self-consistent field iteration to efficiently arrive at a local stationary point, finding solutions more than an order of magnitude faster than Gauss-Newton or on-manifold local solvers on synthetically generated data. The speed of the estimator ensures it can be run in real time on limited hardware.

Thesis supervisor: Luca Carlone
Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

Write your acknowledgments here.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Three-dimensional object understanding is a fundamental problem in computer vision and robotics. Objects themselves are an important abstraction for an autonomous machine to reason about and interact with its environment. In recent years, two-dimensional object understanding has seen significant advances. This class of problems includes classification, semantic segmentation, and feature detection. Benefitting from the abundance of image training data, learned classifiers and segmenters are effective and robust. It is now reasonably straightforward to train a classifier to identify an arbitrary object or category of objects or use a vision-language model to perform similar tasks. However, two-dimensional object understanding is not always enough. To navigate and interact with the environment, 3D information is crucial, especially when safety or efficiency are important factors.

In this thesis we consider two tightly coupled 3D object understanding problems. Shape estimation seeks to build a 3D model of an object under some prior information. We use category-level priors, assuming a library of 3D models for each category and representing arbitrary objects within the category using an *active shape model*. Importantly, category-level priors preserve semantic information about the location of key features. For example, the shape estimate of an arbitrary bottle would including annotations of the 3D positions of the bottle's cap, label, base, etc. The shape estimate also provides a consistent frame for the object's 6D pose (*i.e.,* position and orientation).

Pose estimation is the second fundamental problem we consider. Given an object's shape, the 6D pose completely describes the location of each part of the object. We seek estimates of the shape and pose from 2D images, potentially including depth information. These images can come from an RGB or RGB-D camera, which are common on robots and allow us to leverage advances in 2D object understanding. From another perspective, the shape and pose estimation problem is to de-noise the raw pixel and depth measurements of an object's points under some set of priors about the object.

Our general procedure will be to convert the high-dimensional sensor input into a sparse set of lower-dimensional *keypoints* with known correspondences to a 3D model or set of 3D models. Under some noise model, we formulate and solve an optimization problem for the desired estimates. In general, optimization over the manifold of orientations is non-convex and cannot be solved efficiently to global optimality. In the absence of global optimality solvers are prone to local solutions which correspond to bad estimates and highly dependent on an initial guess. Importantly, local solvers cannot distinguish between local and global

solutions, meaning they may produce bad estimates without warning.

We emphasize methods which come with assurances of a good estimate. In particular, Chapters 3 and 4 focus on certifiably globally optimal optimization, which reports a certificate that guarantees the estimate is globally optimal. In Chapter 5 we take a different approach, presenting a hyper-fast solver that allows high-rate estimation of shape and pose in real time.

Optimization with a sparse set of keypoints is not the only paradigm for 3D object understanding problems. Using a dense pixel-to-3D mapping is an increasingly popular approach [1]. However, it forgoes the guarantees we can achieve with sparse keypoints. We review this technique and other approaches in Section 1.2.

1.1 Summary of Contributions

TODO

1.2 Related Work

There is a significant body of work tackling the problems of shape and pose estimation with assurances. We review key paradigms in estimation from single images, object tracking across images, and pose estimation with uncertainty sets. We also review the growing field of certifiable optimization for robotics problems. The work in this thesis is most connected to [2] and [3].

1.2.1 Single-Frame Shape and Pose Estimation

The dominant paradigm for single-frame reasoning is the *two-stage* approach. In this approach, a network first estimates correspondences between an image and some set of priors. Then, an analytic or learned algorithm extracts a pose and shape estimate from the correspondence predictions. This approach has roots in classical point cloud registration algorithms, which efficiently find a pose given correspondences [4, 5] or iteratively refine correspondences [6, 7]. The rise of deep learning has substantially improved the correspondence estimation stage. Learned keypoint detectors trained on specific objects [8] or a category of objects [9–11] can efficiently detect a sparse set of correspondences. These are the front-ends relied upon in [2, 12]. More generally, deep learning enables regression to more abstract intermediate representations such as dense pixel-wise correspondences to a normalized frame [1] or probabilistic representations of regions of interest [13]. Recently, FoundationPose [14] shows training-free pose estimation given only a CAD model by exploiting a large pretrained foundation model for feature extraction. The tradeoff with more abstract representations is runtime; these methods require significant graphics and computational resources that inhibit real-time deployment on embedded systems.

Under these paradigms, the gap between estimating a pose given a known shape and simultaneously estimating shape and pose from a known prior is relatively small. Normalized object coordinates [1] paired with a neural-implicit shape decoder for shape estimation are particularly generalizeable to objects beyond a single category [15, 16]. Neural signed

distance fields can be learned from a series of images picturing a novel object [14, 17]. In Chapters 3, 5 we use an *active shape model* similar to [18–20].

In this thesis we use a keypoint-based front-end [8, 9, 21] and focus on analytical algorithms for pose and shape estimation. With few exceptions [2, 12], many of the dominant approaches reviewed here lack assurances. We use certifiable optimization to give certificates when the pose estimates are trustworthy, and propose substantially faster back-ends for real-time correction.

Pose Uncertainty

It is often desirable to have some measure of the uncertainty of a pose estimate. A simple lower bound on the covariance of the pose estimate can be obtained via the inverse Hessian at the optimal pose estimate. This is the Cramer-Rao lower bound [22], and is thus not an upper bound on uncertainty. Yang and Pavone [3] propose using conformal prediction [23] to obtain high-probability error bounds on measurements. Follow-up works focus on propagating these bounds to explicit uncertainty bounds on a pose estimate [24, 25] which are assumed to also hold with high probability. This propagation is difficult due to non-convexity of rotations and the implicit structure of a conformalized uncertainty set; the bounds given in [24, 25] are relatively loose and require high-order semidefinite relaxations, making them quite slow. A simpler approach might be to apply conformal prediction directly to a pose estimate, but this requires additional assumptions to obtain a good pose estimate. In Chapter 4 we build upon prior work [3, 24, 25] and propose an efficient algorithm to quickly obtain uncertainty bounds.

1.2.2 Object Tracking Across Frames

The key distinction between single-frame pose estimation and tracking is the additional information from multiple views. Traditional target tracking approaches circumvent shape estimation by assuming the object to be a point mass [26] or assuming full knowledge of the object shape [12, 13, 27–31]. Early approaches used handcrafted features, such as points, edges [32], or planes [33] to compute relative poses. The set of pose estimates could then be smoothed via Kalman filtering [27, 28]. More recently, the use of handcrafted features has given way to learned features [30] and edge detection [31], and new approaches based on point cloud registration [12], particle filtering [13], or unscented Kalman filters [29] have emerged.

In practical settings, instance-level information is rarely available. Recent approaches investigate pose and shape estimation for objects within a known category [34–38] or at least similar enough to the training data [17]. These approaches generally extract a sparse representation of the object to estimate relative motion between frames. Wang et al. [34] focus on an attention mechanism for extracting frame-to-frame keypoints in a self-supervised manner, leaving the work of relative pose estimation to point cloud registration, which is unable to use temporal information beyond two frames. Wen and Bekris [35] use a similar architecture but take a SLAM-inspired approach, using dense frame-to-frame feature correspondences and multi-frame pose graph optimization to refine the estimate. Other methods

use learned keypoint correspondences for the Iterative Closest Point (ICP) method [38] or learning-based regression to estimate relative motion in the small pose regime [36].

Even with keyframe selection [17], frame-to-frame back-ends require a separate tool to obtain object pose relative to a camera or world frame, which is often useful in applications. In contrast, we propose an optimization back-end that produces *certifiably optimal* shape and pose estimates from category-level keypoints without relying on local solvers. This gives useful world-frame poses directly and allows the use of a motion model to mitigate the impact of measurement noise.

1.2.3 Certifiable Algorithms

The work in this thesis extends the body of work on *certifiable perception algorithms*. A certifiable algorithm solves an optimization problem and either provides a certificate of optimality or a bound on the suboptimality of the produced solution [12]. Certifiable algorithms are typically derived using semidefinite relaxations, and are usually based on Shor's relaxation (see Section 2.1) of quadratically constrained quadratic programs (QCQPs) or Lasserre's relaxation of polynomial optimization problems [39–41]. Certifiable algorithms have been proposed for rotation averaging [42, 43], pose graph optimization [44, 45], 3D registration [12, 46], 2-view geometry [47, 48], perspective-n-point problems [49], and single-frame pose and shape estimation [2]. Recent work has extended certifiable solvers to cope with outliers [39] and anisotropic noise [50].

Chapter 2

Preliminaries

In this chapter we present mathematical preliminaries that will be useful in the succeeding chapters. First, Section 2.1 describes Shor's relaxation as applied to rotation matrices in SO(3). Section 2.2 details split conformal prediction and its coverage guarantees, and Section 2.3 gives properties of unit quaternions and their connection to rotation matrices. These sections are self-contained but not comprehensive, providing references for a more detailed perspective.

2.1 Rotation Matrices and Shor's Relaxation

Here we review the quadratic structure of SO(3) and Shor's semidefinite relaxation [41] which can be used to solve rotation-constrained problems in polynomial time.

2.1.1 Special Orthogonal Group as Quadratic Constraints

The set of rotation matrices form the special orthogonal group. A matrix $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is *orthogonal* if $\mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{R}\mathbf{R}^\mathsf{T} = \mathbf{I}_3$. This allows matrices of determinant ± 1. The special orthogonal matrix additional constrains the determinant to $+1$. This is equivalent to the *right-hand rule* constraint: columns of $\mathbf{R}$ obey the right hand rule. Denoting columns by $\mathbf{r}_i$, the right hand rule is:

$$\det(\mathbf{R}) = 1 \iff \begin{cases} \mathbf{r}_1 \times \mathbf{r}_2 = \mathbf{r}_3 \\ \mathbf{r}_2 \times \mathbf{r}_3 = \mathbf{r}_1 \\ \mathbf{r}_3 \times \mathbf{r}_1 = \mathbf{r}_2 \end{cases} \tag{2.1}$$

Together, the orthogonality and right-hand rule constraints make up quadratic inequality constraints. We give the explicit forms of these 15 quadratic inequality constraints in Appendix B.1.

2.1.2 Shor's Semidefinite Relaxation

Quadratic equality constraints such as SO(3) constraints are non-convex. Fortunately, they lend themselves to a convex semidefinite relaxation known as Shor's relaxation [41].

Consider the following quadratically-constrained quadratic program (QCQP) for a given set of symmetric matrices $\mathbf{A}$, $\mathbf{B}_i$, and $\mathbf{C}_j$:

$$\begin{aligned} f^\star = \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}^\mathsf{T}\mathbf{B}_i\mathbf{x} \leq 0, \quad i = 1, ..., N, \\ & \mathbf{x}^\mathsf{T}\mathbf{C}_j\mathbf{x} = 0, \quad j = 1, ..., M \end{aligned} \tag{2.2}$$

Eq. (2.2) is *non-convex* due to (i) quadratic equality constraints, (ii) quadratic inequality constraints which may not be positive semidefinite, and (iii) a quadratic objective which may not be positive semidefinite.

Notice that $\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} = \text{tr}(\mathbf{A}\mathbf{x}\mathbf{x}^\mathsf{T})$ for any quadratic. Thus, we can rewrite (2.2) in terms of $\mathbf{X} \triangleq \mathbf{x}\mathbf{x}^\mathsf{T}$, where the matrix $\mathbf{X} \succeq 0$ and has rank 1. With this reparameterization, the QCQP (2.2) is equivalent to:

$$\begin{aligned} f^\star = \min_{\mathbf{X} \in \mathbb{R}^{n \times n}} \quad & \text{tr}(\mathbf{A}\mathbf{X}) \\ \text{s.t.} \quad & \text{rank}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq 0, \\ & \text{tr}(\mathbf{B}_i\mathbf{X}) \leq 0, \quad i = 1, ..., N, \\ & \text{tr}(\mathbf{C}_j\mathbf{X}) = 0, \quad j = 1, ..., M \end{aligned} \tag{2.3}$$

Eq. (2.3) is a semidefinite program with a linear objective and linear constraints. The only non-convex piece is the rank constraint. Dropping the rank constraint gives a convex semidefinite program known as *Shor's relaxation*.

Theorem 2.1.1 (Shor's Semidefinite Relaxation)**.** *Consider the following optimization problem:*

$$\begin{aligned} f^{\text{SDP}} = \min_{\mathbf{X} \in \mathbb{R}^{n \times n}} \quad & \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{X} \succeq 0, \\ & \mathbf{x}^\mathsf{T}\mathbf{B}_i\mathbf{x} \leq 0, \quad i = 1, ..., N, \\ & \mathbf{x}^\mathsf{T}\mathbf{C}_j\mathbf{x} = 0, \quad j = 1, ..., M \end{aligned} \tag{2.4}$$

The solution to (2.4) is a lower bound on the optimal objective of the non-convex QCQP (2.2). That is, $f^{\text{SDP}} \leq f^\star$.

Proof. The SDP (2.4) is the dual of the dual of the QCQP (2.2). We omit details here and refer the interested reader to [51]. □

Shor's relaxation is useful in practice because the duality gap is often small or near zero. Further, it is the basis of the well-studied moment-sum-of-squares hierarchy, which gives higher-order relaxations with tighter duality gaps [52].

2.2 Split Conformal Prediction

Split conformal prediction [23] provides formal statistical guarantees on uncertainty under relatively mild conditions. The key idea is to test a prediction algorithm on annotated data which is similar enough to the expected test-time input. The predictor's performance on this data will inform its performance at test time. In this section we review split conformal prediction, including assumptions and guarantees. For a comprehensive perspective see [53].

Given a predictor $f : \mathcal{X} \to \mathcal{Y}$ which predicts labels $y \in \mathcal{Y}$ from input $x \in \mathcal{X}$, we wish to quantify the uncertainty of its prediction. We are free to choose an uncertainty metric $s(x, y)$ that maps predictions and labels to the real line. The key idea of split conformal prediction is to evaluate this metric on new calibration data annotated with ground truth (x_i, y_i), $i = 1, ..., N$ which is *independent and identically distributed* with the expected test data. The prediction sets are quantiles of this evaluation at confidence α, plus a minor numerical correction. Importantly, N may be small, on the order of $500 - 1000$ points, to get good coverage [54]. See Algorithm 2.1 for the full algorithm.

Data: Score s, confidence $\alpha \in [0, 1]$, and calibration data (x_i, y_i), $i = 1, ..., N$.
Result: Prediction set $\mathcal{C}(x) = \{y \in \mathcal{Y} : s(x, y) \leq q\}$
for $i \leftarrow 1$ **to** N **do**
  $S_i \leftarrow s(x_i, y_i)$
end
$q \leftarrow \text{Quantile}\left(\mathbf{S}, \frac{\lceil (1-\alpha)(N+1) \rceil}{N}\right)$

Algorithm 2.1: Standard split conformal prediction.

Useful score functions generally quantify the deviation of a prediction from its label. For example, consider the *scaled residual score* [53] which we use in Section 4.4. Given a predictor f and an uncertainty estimate $\sigma(x)$, the scaled residual score is:

$$s(x, y) = \frac{\|y - f(x)\|}{\sigma(x)} \tag{2.5}$$

This score is useful for keypoint detection, when the predictor outputs a pixel detection and an uncertainty estimate. Importantly, we make no assumptions about the accuracy of the predictor or its uncertainty estimate. Despite the mild assumptions, we can still obtain tight uncertainty bounds.

Theorem 2.2.1 (Conformal Coverage)**.** *Consider calibration data* (x_i, y_i) $i = 1, ..., N$ *which is* exchangeable *with test data* (x_{N+1}, y_{N+1})*. Then, the prediction set* $\mathcal{C}(x)$ *given by Algorithm 2.1 satisfies:*

$$1 - \alpha \leq \mathbb{P}\left(y_{N+1} \in \mathcal{C}(x_{N+1})\right) \leq 1 - \alpha + \frac{1}{N+1} \tag{2.6}$$

assuming the conformal scores S_i *are unique (no ties) with probability* 1*.*

For a proof of Theorem 2.2.1 see [53]. Notice that split conformal prediction has a quite general guarantee of tight coverage. The key assumption is *exchangeability*, which is a light

relaxation of independent and identically distributed (the latter implies exchangeability). Informally, exchangeability means that the data is indistinguishable if reordered. For example, sampling from a finite set without replacement produces an exchangeable sequence which is not independent.

2.3 Unit Quaternions

In this section, we review the rules of quaternion arithmetic for rigid rotations. For a more complete discussion see [55]. For a historical perspective, [56] is relevant.

A unit quaternion $\mathbf{q} \in \mathbb{S}^3$ is a unit vector $\mathbf{q} = [q_1, \mathbf{q}_v^\mathsf{T}]^\mathsf{T}$. The term q_1 is called the *scalar part* and the vector $\mathbf{q}_v \in \mathbb{R}^3$ is the *vector part*. We consider quaternions for their connection to rigid rotations. Given an axis $\boldsymbol{\omega}$ and angle θ, the quaternion representation is:

$$\mathbf{q} = \begin{bmatrix} \cos(\theta/2) \\ \boldsymbol{\omega} \sin(\theta/2) \end{bmatrix} \tag{2.7}$$

Two properties are immediately apparent from (2.7). First, to undo a rotation by θ simply negate the vector part: $\mathbf{q}^{-1} = [q_1, -\mathbf{q}_v^\mathsf{T}]^\mathsf{T}$. Second, quaternions have *double coverage*: $-\mathbf{q}$ and $\mathbf{q}$ represent the same rotation. Applying a rotation to a vector requires quaternion algebra. To rotate a point $\mathbf{y} \in \mathbb{R}^3$:

$$\mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \otimes \mathbf{q}^{-1} = \begin{bmatrix} 0 \\ \mathbf{R}\mathbf{y} \end{bmatrix} \tag{2.8}$$

where $\otimes$ denotes the *quaternion product* and $\mathbf{R} \in \mathrm{SO}(3)$ is the rotation corresponding to the quaternion $\mathbf{q}$. The vector $\tilde{\mathbf{y}} \triangleq [0, \mathbf{y}^\mathsf{T}]^\mathsf{T}$ is called the homogeneous form of $\mathbf{y}$. In this thesis we consider quaternion products as matrix-vector products. Given $\mathbf{a} \in \mathbb{R}^4$ and $\mathbf{b} \in \mathbb{R}^4$,

$$\mathbf{a} \otimes \mathbf{b} = \boldsymbol{\Omega}_1(\mathbf{a})\mathbf{b} = \boldsymbol{\Omega}_2(\mathbf{b})\mathbf{a} \tag{2.9}$$

which defines the following product matrices:

$$\boldsymbol{\Omega}_1(\mathbf{a}) = \begin{bmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & -a_4 & a_3 \\ a_3 & a_4 & a_1 & -a_2 \\ a_4 & -a_3 & a_2 & a_1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Omega}_2(\mathbf{a}) \begin{bmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & a_4 & -a_3 \\ a_3 & -a_4 & a_1 & a_2 \\ a_4 & a_3 & -a_2 & a_1 \end{bmatrix} \tag{2.10}$$

The matrices $\boldsymbol{\Omega}_1$ and $\boldsymbol{\Omega}_2$ have several useful properties. We summarize these in the Lemma below, inspired by [57].

Lemma 2.3.1 (Quaternion Product Properties)**.** *The quaternion product matrices* (2.10) *satisfy the following properties.*

(i) Inverse*: for any* $\mathbf{a} \in \mathbb{R}^4$ *and any* $\mathbf{y} \in \mathbb{R}^3$*:*

$$\boldsymbol{\Omega}_1(\mathbf{a}^{-1}) = \boldsymbol{\Omega}_1(\mathbf{a})^\mathsf{T} \tag{2.11}$$

$$\boldsymbol{\Omega}_2(\mathbf{a}^{-1}) = \boldsymbol{\Omega}_2(\mathbf{a})^\mathsf{T} \tag{2.12}$$

$$\boldsymbol{\Omega}_1(-\tilde{\mathbf{y}}) = \boldsymbol{\Omega}_1(\tilde{\mathbf{y}}^{-1}) = \boldsymbol{\Omega}_1(\tilde{\mathbf{y}})^\mathsf{T} = -\boldsymbol{\Omega}_1(\tilde{\mathbf{y}}) \tag{2.13}$$

$$\boldsymbol{\Omega}_2(-\tilde{\mathbf{y}}) = \boldsymbol{\Omega}_2(\tilde{\mathbf{y}}^{-1}) = \boldsymbol{\Omega}_2(\tilde{\mathbf{y}})^\mathsf{T} = -\boldsymbol{\Omega}_2(\tilde{\mathbf{y}}) \tag{2.14}$$

(ii) Linearity*: for any* $\alpha \in \mathbb{R}$, $\mathbf{a} \in \mathbb{R}^4$, $\mathbf{b} \in \mathbb{R}^4$*:*

$$\mathbf{\Omega}_1(\mathbf{a}+\mathbf{b}) = \mathbf{\Omega}_1(\mathbf{a}) + \mathbf{\Omega}_1(\mathbf{b}) \tag{2.15}$$
$$\mathbf{\Omega}_2(\mathbf{a}+\mathbf{b}) = \mathbf{\Omega}_2(\mathbf{a}) + \mathbf{\Omega}_2(\mathbf{b}) \tag{2.16}$$
$$\mathbf{\Omega}_1(\alpha\mathbf{a}) = \alpha\mathbf{\Omega}_1(\mathbf{a}) \tag{2.17}$$
$$\mathbf{\Omega}_2(\alpha\mathbf{a}) = \alpha\mathbf{\Omega}_2(\mathbf{a}) \tag{2.18}$$

(iii) Commutative*: for any* $\mathbf{a} \in \mathbb{R}^4$ *and* $\mathbf{b} \in \mathbb{R}^4$*, the matrices* $\mathbf{\Omega}_1(\mathbf{a})$ *and* $\mathbf{\Omega}_2(\mathbf{b})$ *commute:*

$$\mathbf{\Omega}_1(\mathbf{a})\mathbf{\Omega}_2(\mathbf{b}) = \mathbf{\Omega}_2(\mathbf{b})\mathbf{\Omega}_1(\mathbf{a}) \tag{2.19}$$

(iv) Orthogonality*: for any unit quaternion* $\mathbf{q} \in \mathbb{S}^3$*,*

$$\mathbf{\Omega}_1(\mathbf{q})\mathbf{\Omega}_1(\mathbf{q})^\mathsf{T} = \mathbf{I}_4 \tag{2.20}$$
$$\mathbf{\Omega}_2(\mathbf{q})\mathbf{\Omega}_2(\mathbf{q})^\mathsf{T} = \mathbf{I}_4 \tag{2.21}$$

These properties may be checked by substitution. We conclude this section with a quadratic form identity for rotations. Given $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$ and $\mathbf{R} \in \mathrm{SO}(3)$, we rewrite the inner product $\mathbf{x}^\mathsf{T}\mathbf{R}\mathbf{y}$ in terms of its quaternion $\mathbf{q} \in \mathbb{S}^3$. Using (2.8),

$$\mathbf{x}^\mathsf{T}\mathbf{R}\mathbf{y} = \tilde{\mathbf{x}}^\mathsf{T}(\mathbf{q} \otimes \tilde{\mathbf{y}} \otimes \mathbf{q}^{-1}) \tag{2.22}$$

From (2.9) we can rewrite the quaternion products as matrix-vector products:

$$\mathbf{q} \otimes \tilde{\mathbf{y}} \otimes \mathbf{q}^{-1} = \mathbf{q} \otimes \mathbf{\Omega}_1(\mathbf{q}^{-1})\tilde{\mathbf{y}} = \mathbf{\Omega}_2(\mathbf{q})\mathbf{\Omega}_1(\mathbf{q})^\mathsf{T}\tilde{\mathbf{y}} \tag{2.23}$$

Now use commutation and (2.9) to write this as a quadratic form:

$$\mathbf{x}^\mathsf{T}\mathbf{R}\mathbf{y} = (\mathbf{\Omega}_1(\mathbf{q})\tilde{\mathbf{x}})^\mathsf{T}(\mathbf{\Omega}_2(\mathbf{a})\tilde{\mathbf{y}}) = \mathbf{q}^\mathsf{T}\mathbf{\Omega}_2(\tilde{\mathbf{x}})^\mathsf{T}\mathbf{\Omega}_1(\tilde{\mathbf{y}})\mathbf{q} \tag{2.24}$$

In summary, an inner product with a rotated vector can be rewritten as a quadratic form of unit quaternions.

$$\mathbf{x}^\mathsf{T}\mathbf{R}\mathbf{y} = -\mathbf{q}^\mathsf{T}\mathbf{\Omega}_2(\tilde{\mathbf{x}})\mathbf{\Omega}_1(\tilde{\mathbf{y}})\mathbf{q} \tag{2.25}$$

Chapter 3

Globally Optimal Shape Estimation and Object Tracking

This chapter details the Certifiable Algorithm for Shape and pose Estimation (CAST*). Part of this work was published in Robotics and Automation Letters in 2024 as *A Certifiable Algorithm for Simultaneous Shape Estimation and Object Tracking* [58] and the source code is publicly available[1].

Section 3.1 introduces the category-level shape and pose tracking problem. We use 3D semantic keypoint measurements extracted from an RGB-D image sequence by an external front-end, such as a neural network [9, 59]. Assuming Gaussian random noise, we phrase the estimation as a fixed-lag smoothing problem. Temporal constraints enforce the object's rigidity and smooth motion according to a constant pseudo-world-frame or body-frame motion model. The solutions to this problem are the estimates of the object's state (poses, velocities) and shape (parameterized according to the *active shape model*) over the smoothing horizon. Our key contribution is to show that despite the non-convexity of the fixed-lag smoothing problem, we can solve it to *certifiable optimality* using a small-size semidefinite relaxation in the outlier-free case (Section 3.2). Under a body-frame velocity model we marginalize out the shape estimation problem, while the world-frame model allows us to marginalize shape, position, and velocity, leading to significant speedups.

For robustness to incorrect keypoint detections, Section 3.3 describes a fast outlier rejection scheme. We use compatibility tests based on the rigid-body assumption and active shape model for fast outlier rejection, and wrap our solver in a graduated non-convexity scheme [60]. For a high-level overview of the outlier-robust method, denoted CAST#, see Fig. 3.1a. In Section 3.4 we evaluate the proposed approach on synthetic and real data, showcasing its performance in a table-top manipulation scenario and a drone-based vehicle tracking application.

3.1 Problem Formulation

This section formalizes the *category-level shape estimation and pose tracking* problem. Given a sequence of RGB-D images picturing an object of known category (*e.g.,* a car), and as-

[1] https://github.com/MIT-SPARK/certifiable_tracking

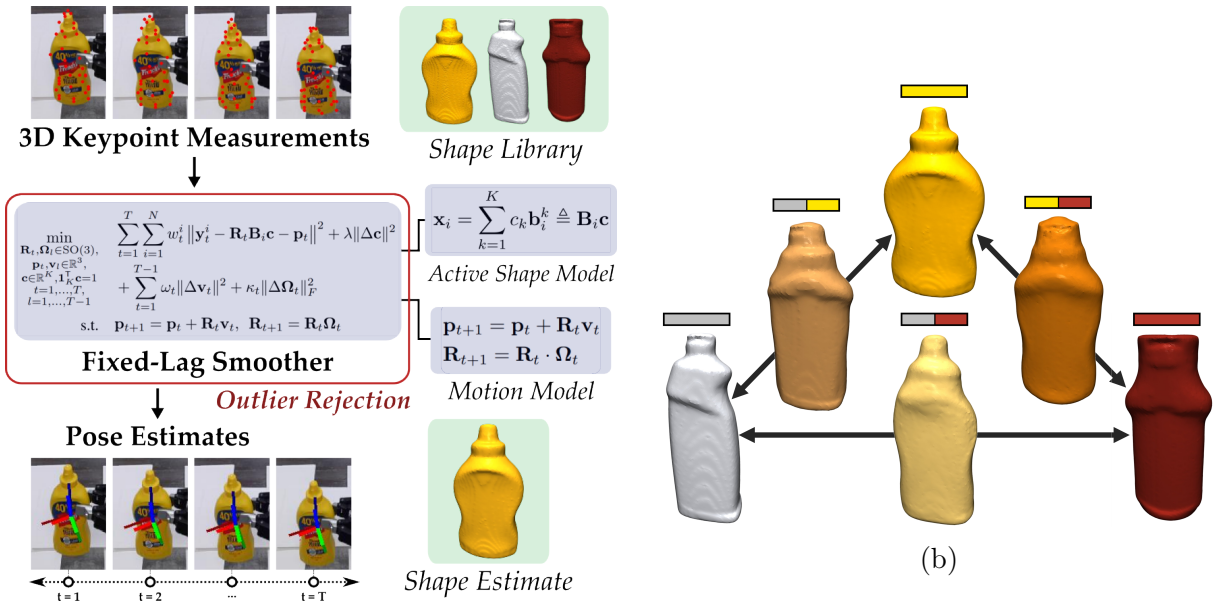


Figure 3.1: (a) **Overview of CAST#.** Given 3D keypoint measurements obtained via a learning-based detector, we formulate a non-convex fixed-lag smoothing problem. We solve this problem via a tight and small-size semidefinite relaxation and wrap the method in an outlier rejection scheme to robustly estimate shape and pose over a fixed time horizon. (b) **Active Shape Model.** Known 3D models in the *bottle* category and their averages computed according to the active shape model. Vertices are the original models and edges are the average of the two vertices. The active shape model can represent any 3D geometry in the convex hull of its shape library through a point-wise weighted average.

suming the availability of a 3D semantic keypoint detector, we seek an estimate of the time-independent shape and time-dependent pose (position and orientation) of the object. Below we describe our choice of motion models, shape representation, and measurement model.

3.1.1 Object State and Motion Models

We represent the target object's state using its pose and velocity at a particular time t. Denote the position and orientation of the target object in the world frame as $\mathbf{p}_t \in \mathbb{R}^3$ and $\mathbf{R}_t \in \mathrm{SO}(3)$, respectively. We consider a *body-frame* model, a bilinear model which generalizes the non-holonomic motion of ground vehicles such as cars, and a linear *pseudo-world-frame* model which is less realistic but allows substantial computational speedup. We first introduce the common framework and then specialize to each model.

Denote the target's change in rotation between time steps with $\mathbf{\Omega}_t \in \mathrm{SO}(3)$ and some change in position with $\mathbf{v}_t \in \mathbb{R}^3$. These state variables are the discrete time analog to velocity and rotation rate. Any object's motion obeys the following discrete-time first-order dynamics:

$$\mathbf{p}_{t+1} = \mathbf{f}(\mathbf{p}_t, \mathbf{R}_t, \mathbf{R}_{t+1}, \mathbf{v}_t), \quad \mathbf{R}_{t+1} = \mathbf{R}_t \cdot \mathbf{\Omega}_t \tag{3.1}$$

where f is at most linear in $\mathbf{v}_t$ and will be defined separately for the world and body-frame models. The model (3.1) is quite general, since by choosing suitable values for $\mathbf{v}_t$, $\boldsymbol{\Omega}_t$ we can produce arbitrary trajectories.

Now, we assume that the velocities' dynamics are approximately constant; *i.e.,* the velocity and the rotation rate are constant during short time intervals up to random perturbations $\boldsymbol{v}_t^\epsilon \in \mathbb{R}^3$ and $\mathbf{R}_t^\epsilon \in \mathrm{SO}(3)$:

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \boldsymbol{v}_t^\epsilon, \quad \boldsymbol{\Omega}_{t+1} = \boldsymbol{\Omega}_t \cdot \mathbf{R}_t^\epsilon \tag{3.2}$$

When $\mathbf{v}_t$ and $\boldsymbol{\Omega}_t$ are exactly constant the dynamical system in equations (3.1)-(3.2) models perfectly first-order dynamics. The random noise terms allow small deviations from this assumption in the observed trajectory. We now specialize to the two motion models.

Body-Frame Model

Let $\mathbf{v}_t$ be the target's *body-frame* change in position. The body-frame motion model is:

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \mathbf{R}_t \mathbf{v}_t \triangleq \mathbf{f}_b(\mathbf{p}_t, \mathbf{R}_t, \mathbf{R}_{t+1}, \mathbf{v}_t) \tag{3.3}$$

Together with (3.2), this assumes the velocities' dynamics are approximately *constant twist*. When $\mathbf{v}_t$ and $\boldsymbol{\Omega}_t$ are exactly constant the dynamical system models 3D spiral-shaped trajectories, including the corner cases of a straight line, circular trajectory, or in-place rotation. The random noise terms model small deviations from these assumptions in the observed trajectory. The proposed body-frame model is a 3D version of the popular constant-turn-rate model [28], generalizing it to allow an arbitrary axis of rotation. Such a model is expressive enough to capture the non-holonomic motion of a car and the unpredictable motions of a manipulated object. Although realistic, we will see that the bilinear constraints introduce additional constraints which cannot be analytically eliminated, resulting in a larger estimation problem.

Pseudo-World-Frame Model

The pseudo-world-frame model is a reformulation of the body-frame model which sacrifices interpretability for computational speed. Let $\mathbf{v}_t$ be the change in position between time steps as defined by the following first-order dynamics:

$$\mathbf{p}_{t+1} = \mathbf{R}_{t+1}(\mathbf{R}_t^\mathsf{T} \mathbf{p}_t + \mathbf{v}_t) \triangleq \mathbf{f}_w(\mathbf{p}_t, \mathbf{R}_t, \mathbf{R}_{t+1}, \mathbf{v}_t) \tag{3.4}$$

We call this model together with (3.2) a *pseudo-world-frame* model because it is a linear update in the rotated frames: $\mathbf{R}_{t+1}^\mathsf{T} \mathbf{p}_{t+1} = \mathbf{R}_t^\mathsf{T} \mathbf{p}_t + \mathbf{v}_t$. While there is no intuitive physical interpretation, we expect any first-order approximation to be accurate for sufficiently small time step between frames. The random noise terms in (3.2) allow the model to remain expressive to small deviations between time steps. The key utility of this model is in simplifying the fixed lag smoother, described in Section 3.2.

In the following, we assume that the velocity noise follows an isotropic zero-mean Gaussian distribution: $\boldsymbol{v}_t^\epsilon \sim \mathcal{N}(0, \boldsymbol{\Sigma}_t^v)$ and that the relative rotation noise follows an isotropic Langevin distribution about the identity, following standard practice [44] for distributions over $\mathrm{SO}(3)$: $\mathbf{R}_t^\epsilon \sim \mathcal{L}(\mathbf{I}_3, \kappa_t)$. In this equation, κ_t is the *concentration parameter* of the Langevin distribution (intuitively, this plays a similar role as the inverse of the variance).

3.1.2 Shape Parameterization

We use the *active shape model* to represent intra-category shape variations. Given an object category (*e.g.,* bottle), we assume a library of 3D models (*e.g.,* specific bottle shapes) that span the category, where the objects in the library are denoted as $\mathcal{B}_k$, $k = 1, \ldots, K$. Any instance, then, is just a pointwise linear combination of the models in the shape library (see Fig. 3.1b). More formally, let $\mathbf{x}_i$ be a point on the instance object corresponding to the point $\mathbf{b}_i^k \in \mathcal{B}_k$ in each library shape. The active shape model is:

$$\mathbf{x}_i = \sum_{k=1}^{K} c_k \mathbf{b}_i^k \triangleq \mathbf{B}_i \mathbf{c} \tag{3.5}$$

where $c_k \in [0, 1]$ and $\sum_k c_k = 1$. Thus, the shape of the target object is fully specified by its shape coefficient $\mathbf{c} = [c_1, \ldots, c_K]$ and the shape library for each point $\mathbf{B}_i = [\mathbf{b}_i^1, \ldots, \mathbf{b}_i^K]$. This representation is simple and expressive: it captures any object in the convex hull of the shape library (including the library shapes themselves) via a linear combination described by a single vector of coefficients [61, 62]. Further, measurements of a small number of semantic *keypoints* are enough to resolve the dense object shape.

3.1.3 Measurement Model

The inputs to our estimator are measurements of the 3D positions of *semantic keypoints* on the target object. These keypoints correspond to semantically meaningful features common to a specific object category, and are typically produced by a learning-based detector, as in [9, 59]. For instance, a set of keypoints on a bottle might be the locations of the cap, center-point of the base, label, etc. Such keypoints may be detected by a model trained on a category of bottles, not just a particular instance.

At each time t we are given the 3D position of N keypoints denoted $\mathbf{y}_t^1, \ldots, \mathbf{y}_t^N$. These measurements obey the following generative model:

$$\mathbf{y}_t^i = \mathbf{R}_t \cdot (\mathbf{B}_i \mathbf{c}) + \mathbf{p}_t + \boldsymbol{\epsilon}_t^i \tag{3.6}$$

Each measurement $\mathbf{y}_t^i$ is a rigid transformation $(\mathbf{R}_t, \mathbf{p}_t)$ of the keypoint's location in the object's frame $\mathbf{B}_i \mathbf{c}$ (expressed according to the active shape model) plus measurement noise $\boldsymbol{\epsilon}_t$. For now, we assume the measurement noise obeys an isotropic zero-mean Gaussian distribution: $\boldsymbol{\epsilon}_t^i \sim \mathcal{N}(0, \boldsymbol{\Sigma}_t^i)$.

3.1.4 Simultaneous Shape Estimation and Tracking

We now state the primary problem we tackle in this chapter.

Problem 3.1.1. *Consider an object of known category moving according to the dynamics in eqs.* (3.1)-(3.2) *and a choice of motion model* (3.3) *or* (3.4)*. Given measurements of N keypoints in the form* (3.6) *taken over T time steps, estimate the time-varying state $(\mathbf{R}_t, \mathbf{p}_t, \mathbf{v}_t, \boldsymbol{\Omega}_t)$ and time-independent shape $\mathbf{c}$ of the object for $t = 1, \ldots, T$.*

Problem 3.1.1 may be interpreted as a fixed-lag smoother, where our primary goal is to estimate the state at time T using also the most recent $T - 1$ measurements.

3.2 CAST*: Certifiable Shape Estimation and Tracking in the Outlier-Free Setting

This section presents CAST*, a certifiably optimal estimator solving Problem 3.1.1 in the outlier-free setting. CAST* is also the basis for our outlier-robust extension in Section 3.3.

We adopt a *maximum a posteriori* estimation framework that represents Problem 3.1.1 as an optimization problem. This framework minimizes the residual errors of the measurement and motion models over the time horizon T, possibly including priors. In our case, the only prior is that shape coefficients $\mathbf{c}$ are distributed according to a Gaussian with covariance $\frac{1}{\lambda}\mathbf{I}_3$ about the mean shape $\bar{\mathbf{c}} \triangleq \frac{1}{K}\mathbf{1}_K$. In practice, this prior regularizes the problem when the shape library is larger than the number of keypoints ($K > N$); see e.g. [2].

The maximum a posteriori estimator takes the form:

$$\begin{aligned} \min_{\substack{\mathbf{R}_t,\boldsymbol{\Omega}_l\in\mathrm{SO}(3),\\ \mathbf{p}_t,\mathbf{v}_l\in\mathbb{R}^3,\\ \mathbf{c}\in\mathbb{R}^K,\mathbf{1}_K^\mathsf{T}\mathbf{c}=1\\ t=1,\dots,T,\\ l=1,\dots,T-1}} \quad & \sum_{t=1}^{T}\sum_{i=1}^{N} w_t^i \left\|\mathbf{y}_t^i - \mathbf{R}_t\mathbf{B}_i\mathbf{c} - \mathbf{p}_t\right\|^2 + \lambda\|\Delta\mathbf{c}\|^2 \\ & + \sum_{t=1}^{T-1} \omega_t\|\mathbf{v}_{t+1}-\mathbf{v}_t\|^2 + \kappa_t\|\boldsymbol{\Omega}_{t+1}-\boldsymbol{\Omega}_t\|_F^2 \\ \text{s.t.} \quad & \mathbf{p}_{t+1} = \mathbf{f}(\mathbf{p}_t,\mathbf{R}_t,\mathbf{R}_{t+1},\mathbf{v}_t),\ \ \mathbf{R}_{t+1} = \mathbf{R}_t\boldsymbol{\Omega}_t \end{aligned} \tag{3.7}$$

In the previous expression, we used the shorthand $\Delta\mathbf{c} \triangleq \mathbf{c} - \bar{\mathbf{c}}$ and assumed isotropic covariances $\boldsymbol{\Sigma}_t^i \triangleq \frac{1}{w_t^i}\mathbf{I}_3$ and $\boldsymbol{\Sigma}_t^v \triangleq \frac{1}{w_t}\mathbf{I}_3$. We also relaxed the constraint $c_k \geq 0$. We observe that the objective is the sum of the shape prior with the negative log-likelihoods of the measurements (3.6) and dynamics (3.2). The constraints enforce the domains of the variables (e.g. $\mathbf{R}_t \in \mathrm{SO}(3)$ or $\mathbf{1}_K^\mathsf{T}\mathbf{c} = 1$) and the dynamics (3.1), where $\mathbf{f}$ may be either the body-frame model (3.3) or world-frame model 3.4. Eq. (3.7) is a maximum a posteriori estimator; see Appendix A.1 for proof.

Notice that (3.7) is non-convex due to the constraint set SO(3) and the quadratic equality constraints. Thus, local search methods such as gradient descent or Gauss-Newton are prone to local minima that result in bad estimates.

In the following we present our approach to solving (3.7) to certifiable optimality via a semidefinite relaxation. In Section 3.2.1 we simplify the problem by analytically solving for the optimal shape coefficient. Using a change of variables, we rewrite (3.7) as a non-convex quadratically constrained quadratic program (QCQP) (which can be simplified further in the case of the pseudo-world-frame motion model) in Section 3.2.2 and apply a semidefinite relaxation in Section 3.2.3. This relaxed problem can be solved using traditional convex optimization techniques and is shown to be empirically *tight* (*i.e.,* the relaxation solves (3.7) to optimality) in Section 3.4.

3.2.1 Closed-Form Solution for Shape

Observe that (3.7) is a linearly constrained convex quadratic program in the variable $\mathbf{c}$. Thus, we can solve for the optimal shape coefficient $\mathbf{c}^\star$ in closed form as a function of the other unknown variables. We formalize this observation below.

Proposition 3.2.1 (Optimal Shape)**.** *For any positions and rotations* $(\mathbf{p}_t, \mathbf{R}_t)$*, the optimal shape coefficient solving* (3.7) *is*

$$\mathbf{c}^\star = 2\mathbf{G}\left(\mathbf{B}^\mathsf{T}\sum_{t=1}^{T}\mathbf{W}_t\begin{bmatrix}\mathbf{R}_t^\mathsf{T}(\mathbf{y}_t^1-\mathbf{p}_t)\\ \vdots \\ \mathbf{R}_t^\mathsf{T}(\mathbf{y}_t^N-\mathbf{p}_t)\end{bmatrix}+\lambda\bar{\mathbf{c}}\right)+\mathbf{g} \tag{3.8}$$

where we defined the following symbols:

$$\begin{array}{lll}\mathbf{W}_t & \triangleq \operatorname{blkdiag}(w_1^t\mathbf{I}_3,\dots,w_N^t\mathbf{I}_3) & \in\mathbb{R}^{3N\times 3N}\\ \mathbf{B} & \triangleq [\mathbf{B}_1^\mathsf{T},\dots,\mathbf{B}_T^\mathsf{T}]^\mathsf{T} & \in\mathbb{R}^{3N\times K}\\ \mathbf{H} & \triangleq \frac{1}{2}\left(\mathbf{B}^\mathsf{T}\left(\sum_{t=1}^{T}\mathbf{W}_t\right)\mathbf{B}+\lambda\mathbf{I}_K\right)^{-1} & \in\mathbb{R}^{K\times K}\end{array} \tag{3.9}$$

$$\mathbf{G}\triangleq\mathbf{H}-\frac{\mathbf{H}\mathbf{1}_K\mathbf{1}_K^\mathsf{T}\mathbf{H}}{\mathbf{1}_K^\mathsf{T}\mathbf{H}\mathbf{1}_K},\qquad \mathbf{g}\triangleq\frac{\mathbf{H}\mathbf{1}_K}{\mathbf{1}_K^\mathsf{T}\mathbf{H}\mathbf{1}_K} \tag{3.10}$$

Proof. See Appendix A.2. □

3.2.2 Change of Variables to Quadratic Program

Problem (3.7) remains non-convex in the state variables $(\mathbf{R}_t, \mathbf{p}_t, \mathbf{v}_t, \mathbf{\Omega}_t)$ due to quadratic equality constraints. We aim to relax this problem into a convex semidefinite program. Towards this goal, we show how (3.7) can be rewritten as a quadratically constrained quadratic program (QCQP). When $\mathbf{f} = \mathbf{f}_b$ (the body-frame model) this is simply a change of variables. In the world-frame model, further simplification similar to Section 3.2.1 significantly reduces the size of the QCQP.

In the objective the squared norm of $\mathbf{R}_t\mathbf{B}_i\mathbf{c}^\star$ is quartic. We use the rotational invariance of the ℓ_2 norm to reparametrize position as $\mathbf{s}_t \triangleq \mathbf{R}_t^\mathsf{T}\mathbf{p}_t$, turning the objective into a quadratic function. Under this transformation the constraint set is quadratic: the dynamics (3.1) become quadratic equalities, and the SO(3) constraints on rotations can be written as quadratic equality constraints, see, *e.g.,* [63]. The result is summarized below.

Proposition 3.2.2 (QCQP Formulation)**.** *Let* $\mathbf{c}$ *be defined as in* (3.8)*, and note that it is a linear function of* $\mathbf{R}_t$ *and* $\mathbf{s}_t$*. The shape estimation and tracking problem can be equivalently written as a quadratically constrained quadratic program:*

$$\begin{aligned}\min_{\substack{\mathbf{R}_t,\mathbf{\Omega}_l\in\mathrm{SO}(3),\\ \mathbf{s}_t,\mathbf{v}_l\in\mathbb{R}^3,\\ t=1,\dots,T,\\ l=1,\dots,T-1}}\quad & \sum_{t=1}^{T}\sum_{i=1}^{N}w_t^i\left\|\mathbf{R}_t^\mathsf{T}\mathbf{y}_t^i-\mathbf{B}_i\mathbf{c}-\mathbf{s}_t\right\|^2+\lambda\|\Delta\mathbf{c}\|^2\\ & +\sum_{t=1}^{T-1}\omega_t\|\mathbf{v}_{t+1}-\mathbf{v}_t\|^2+\kappa_t\|\mathbf{\Omega}_{t+1}-\mathbf{\Omega}_t\|_F^2\\ s.t.\quad & \mathbf{D}\mathbf{s}_{t+1}=\mathbf{s}_t+\mathbf{v}_t,\ \ \mathbf{R}_{t+1}=\mathbf{R}_t\mathbf{\Omega}_t\end{aligned} \tag{3.11}$$

where $\mathbf{D}=\mathbf{\Omega}_t$ *in the body-frame model and* $\mathbf{D}=\mathbf{I}_3$ *for the world-frame model.*

Proof. See Appendix A.3. □

Body-Frame Motion Model

We rewrite (3.11) in canonical form, separating the quadratically constrained variables $(\mathbf{s}, \mathbf{R}, \boldsymbol{\Omega})$ from the linearly constrained ones $(\mathbf{v})$:

$$\begin{aligned} f_b^\star = \min_{\substack{\mathbf{x}\in\mathbb{R}^{21T-8} \\ \mathbf{v}\in\mathbb{R}^{3T-3}}} \ & \mathbf{x}^\mathsf{T}\mathbf{Q}\mathbf{x} + \mathbf{v}^\mathsf{T}\mathbf{P}\mathbf{v} \\ \text{s.t.} \quad & \mathbf{x}^\mathsf{T}\mathbf{A}_i\mathbf{x} + \mathbf{d}_i^\mathsf{T}\mathbf{v} + f_i = 0,\ i = 1, \ldots, m \end{aligned} \tag{3.12}$$

In this equation, $\mathbf{x}$ is a vector in homogeneous form stacking all the unknowns in (3.11) except for $\mathbf{v}_t$ terms, which are stacked in $\mathbf{v}$. Specifically:

$$\mathbf{x} \triangleq [1, \operatorname{vec}(\mathbf{R}_1), \ldots, \operatorname{vec}(\mathbf{R}_T), \operatorname{vec}(\boldsymbol{\Omega}_1), \ldots, \operatorname{vec}(\boldsymbol{\Omega}_{T-1}), \mathbf{s}_1, \ldots, \mathbf{s}_T]^\mathsf{T} \tag{3.13}$$

The matrices $\mathbf{Q}$, $\mathbf{P}$, and $\mathbf{A}_i$ are known symmetric matrices governing the quadratic objective and constraints, and the vectors $\mathbf{d}_i$ and scalars f_i capture the linear and constant portions of the m constraints, respectively.

Pseudo-World-Frame Motion Model

With $\mathbf{D} = \mathbf{I}_3$, (3.11) is *convex* in $\mathbf{s}_t$ and $\mathbf{v}_t$. Crucially, the constraints are now linear functions of $\mathbf{v}_t$ and $\mathbf{s}_t$ with no bilinear constraints involving rotations. Before rewriting the QCQP in canonical form, we can use the same principle in Section 3.2.1 to analytically solve for the optimal positions and velocities as a function of the rotations.

Proposition 3.2.3 (Optimal Position and Velocity)**.** *Stack rotations in* $\mathbf{r}$*, rotated positions in* $\mathbf{s}$*, and velocities in* $\mathbf{v}$*, each vectors. For any rotations* $\mathbf{r}$*, the optimal rotated positions and velocities solving* (3.7) *with* $\mathbf{D} = \mathbf{I}_3$ *are the solutions to the following linear system:*

$$\begin{bmatrix} 2\mathbf{A}_s^\mathsf{T}\mathbf{A}_s & \mathbf{0} & \mathbf{D}_s^\mathsf{T} \\ \mathbf{0} & 2\mathbf{A}_v^\mathsf{T}\mathbf{A}_v & -\mathbf{I}_{3T-3} \\ \mathbf{D}_s & -\mathbf{I}_{3T-3} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{v} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} -2\mathbf{A}_s^\mathsf{T}\mathbf{A}_g - 2\mathbf{A}_s^\mathsf{T}\mathbf{A}_r\mathbf{r} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{3.14}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{3T-3}$ *is a dual variable. The expressions for* $\mathbf{A}_s$*,* $\mathbf{A}_g$*, and* $\mathbf{A}_r$ *are given in Appendix A.4. We note that the leftmost matrix is invertible given at least 3 non-colinear keypoints per time step and with* $T \geq 3$*.*

Proof. See Appendix A.4. □

Notice that the position and velocity terms are *linear* in rotations. Thus, (3.11) remains a QCQP with the substitution of $\mathbf{p}_t^\star$ and $\mathbf{v}_t^\star$ according to Proposition 3.2.3. We rewrite the QCQP in canonical form:

$$\begin{aligned} f_w^\star = \min_{\mathbf{x}\in\mathbb{R}^{18T-8}} \ & \mathbf{x}^\mathsf{T}\mathbf{Q}'\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}^\mathsf{T}\mathbf{A}_i'\mathbf{x} = 0,\ i = 1, \ldots, m' \end{aligned} \tag{3.15}$$

Here, $\mathbf{x}$ is a homogeneous vector stacking all rotations and rotation rates:

$$\mathbf{x} \triangleq [1, \operatorname{vec}(\mathbf{R}_1), \ldots, \operatorname{vec}(\mathbf{R}_T), \operatorname{vec}(\boldsymbol{\Omega}_1), \ldots, \operatorname{vec}(\boldsymbol{\Omega}_{T-1})]^\mathsf{T} \tag{3.16}$$

The matrices $\mathbf{Q}'$ and $\mathbf{A}_i'$ are known symmetric constant matrices describing the quadratic objective and constraints. Compared to (3.12), the decision variable $\mathbf{x}$ is smaller and constraints between positions and velocities are dropped.

3.2.3 Convex Semidefinite Relaxation

While the QCQP in (3.12) is still non-convex in the variable $\mathbf{x}$, it admits a standard semidefinite relaxation [44, 64, 65]. Instead of solving for $\mathbf{x}$ directly, we reparametrize the problem using $\mathbf{X} = \mathbf{x}\mathbf{x}^\mathsf{T}$ (a rank-1 positive semidefinite matrix), and drop the rank-1 constraint on $\mathbf{X}$ to obtain a convex problem that may be solved by off-the-shelf solvers such as MOSEK [66]. This is the well-known Shor's relaxation [41]. It takes slightly different forms for the world and body-frame models.

Corollary 3.2.4 (Shor's Relaxation)**.** *The following semidefinite program (SDP) is a convex relaxation of the body-frame formulation (3.12):*

$$\begin{aligned} f^\star_{\text{SDP}} = \min_{\substack{\mathbf{X}\in\mathbb{S}^{21T-8}\\ \mathbf{v}\in\mathbb{R}^{3T-3}}} & \operatorname{trace}(\mathbf{Q}\mathbf{X}) + \mathbf{v}^\mathsf{T}\mathbf{P}\mathbf{v} \\ s.t. \quad & \operatorname{trace}(A_i\mathbf{X}) + \mathbf{d}_i^\mathsf{T}\mathbf{v} + f_i = 0, \\ & \mathbf{X} \succeq 0, \quad i = 1,\dots,m \end{aligned} \tag{3.17}$$

Also, the following SDP is a convex relaxation of the world-frame QCQP (3.15):

$$\begin{aligned} f^\star_{\text{SDP}} = \min_{\mathbf{X}\in\mathbb{S}^{18T-8}} & \operatorname{trace}(\mathbf{Q}'\mathbf{X}) \\ s.t. \quad & \operatorname{trace}(\mathbf{A}'_i\mathbf{X}) = 0, \\ & \mathbf{X} \succeq 0, \quad i = 1,\dots,m' \end{aligned} \tag{3.18}$$

Further, when the solution $\mathbf{X}^\star$ of (3.17) or (3.15) is rank-1 we can recover exactly the solution to the non-convex QCQP (3.12) by factorizing $\mathbf{X}^\star = \mathbf{x}^\star(\mathbf{x}^\star)^\mathsf{T}$.

Similar to relaxations derived in related work [2, 44, 67] the rank of $\mathbf{X}^\star$ is a *certificate* for the optimality of the solution. Moreover, we can bound the suboptimality of a feasible solution to (3.12) obtained from (3.17) or (3.18) using the objective. Given a feasible solution $(\hat{\mathbf{x}}, \hat{\mathbf{v}})$ achieving objective $\hat{f}$ in (3.12), we bound its suboptimality using $\hat{f} \geq f^\star \geq f^\star_{\text{SDP}}$. The condition $\hat{f} = f^\star_{\text{SDP}}$ also certifies the optimality of the solution. The scalar $\hat{f} - f^\star_{\text{SDP}}$ is called the *suboptimality gap*.

The SDP relaxation is relevant in practice because we observe it to be empirically tight in the case of low-to-moderate noise and no outliers; hence it can produce optimal solutions without needing an initial guess. Moreover, the size of the SDP is independent of the size of the shape library, hence the relaxation is relatively efficient to solve. The key difference between the body-frame SDP (3.17) world-frame SDP (3.18) is the number of decision variables and constraints. The world-frame model has a smaller set of decision variables allowing faster computation.

We name the resulting approach CAST*: *Certifiable Algorithm for Shape estimation and Tracking.*

3.3 Adding Outlier Robustness

Real-world measurements are often corrupted by outliers. In particular, sparse keypoints are vulnerable to misdetections and incorrect depth measurements. Without modifications,

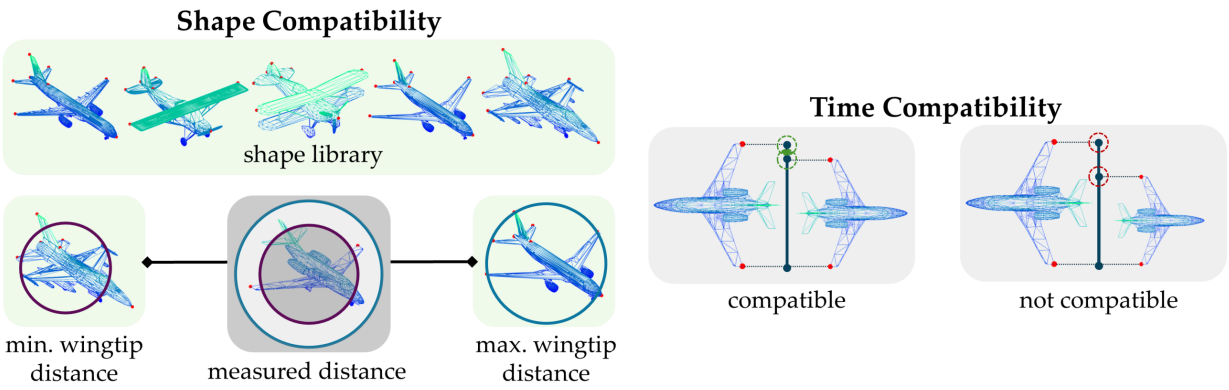


Figure 3.2: **Outlier compatibility tests.** Most outliers are easy to identify via shape or time compatibility tests. Shape compatibility retains keypoints that are mutually within the convex hull of the known shape library. Time compatibility compares keypoint pairs over multiple observations and retains groups that preserve 3D distance over time, up to a tolerance ϵ. We determine the largest set of compatible measurements via a mixed integer linear program.

outliers degrade the result of CAST*. To tackle this problem, we propose a preprocessing step in which we quickly identify and prune gross outliers, and a wrapper for CAST* that iteratively converges to the inlier set. We name this approach CAST# and show empirical robustness to 50-60% outliers.

3.3.1 Compatibility Checks to Remove Gross Outliers

Inspired by [2], we introduce compatibility tests to identify gross outliers (Fig. 3.2). These tests rely on the assumptions of rigid-body motion of the object and the active shape model. The most likely inlier set is thus the largest set of compatible measurements, found via a fast mixed-integer linear program.

Shape Compatibility

Recall that any observed object must lie within the convex hull of the shape library by assumption. Framed as pairwise compatibility, the true distance between any two keypoints i and j must lie somewhere between the minimum and maximum distance between i and j in the shape library models. Therefore, any two keypoint measurements that are outside this bound cannot simultaneously be inliers; one or both must be outliers. Allowing for keypoint noise expands these bounds as summarized in Proposition 3.3.1. Refer to [67] for a full proof.

Proposition 3.3.1 (Shape Compatibility Test)**.** *Let ϵ be the maximum error for a measurement to be considered an inlier. If a pair of measurements $\mathbf{y}_t^i$ and $\mathbf{y}_t^j$ are both inliers, then:*

$$b_{ij}^{\min} - 2\epsilon \leq \|\mathbf{y}_t^i - \mathbf{y}_t^j\| \leq b_{ij}^{\max} + 2\epsilon \tag{3.19}$$

where $b_{ij}^{\{\min,\max\}}$ are the minimum and maximum distances between keypoints i and j in the

shape library:

$$b_{ij}^{\{\min,\max\}} \triangleq \underset{\mathbf{c}\geq 0,\mathbf{1}^T\mathbf{c}=1}{\{\min,\max\}} \|(\mathbf{B}_i - \mathbf{B}_j)\mathbf{c}\| \tag{3.20}$$

Time Compatibility

For a rigid body the distance between two points is constant over time. This forms the basis for a compatibility test between pairs of points at two times.

Proposition 3.3.2 (Time Compatibility Test)**.** *Let ϵ be the maximum error for a measurement to be considered an inlier. Consider the measurements of keypoints i and j at times l and m. If these measurements are all inliers then:*

$$\left|\|\mathbf{y}_l^i - \mathbf{y}_l^j\| - \|\mathbf{y}_m^i - \mathbf{y}_m^j\|\right| \leq 4\epsilon \tag{3.21}$$

Proof. Compare the distance between keypoints i and j at each time, rotating to align coordinates with the body frame:

$$\left|\|\mathbf{R}_l^\mathsf{T}(\mathbf{y}_l^i - \mathbf{y}_l^j))\| - \|\mathbf{R}_m^\mathsf{T}(\mathbf{y}_m^i - \mathbf{y}_m^j)\|\right| \tag{3.22}$$

Bound this using the reverse and forward triangle inequalities, noting that noise is isotropic ($\mathbf{R}\boldsymbol{\epsilon} = \boldsymbol{\epsilon}$):

$$\begin{aligned}(3.22) &\leq \|(\mathbf{B}_i - \mathbf{B}_j + \boldsymbol{\epsilon}_l^i - \boldsymbol{\epsilon}_l^j) - (\mathbf{B}_i - \mathbf{B}_j + \boldsymbol{\epsilon}_m^i - \boldsymbol{\epsilon}_m^j)\| \\ &= \|\boldsymbol{\epsilon}_y^{i,l} - \boldsymbol{\epsilon}_y^{i,l} - \boldsymbol{\epsilon}_y^{i,m} + \boldsymbol{\epsilon}_y^{i,m}\| \leq 4\epsilon\end{aligned} \tag{3.23}$$

Since the 2-norm is invariant to rotations, we can remove the rotations from (3.22) and obtain the result. □

Outlier Pruning

Any set of inliers must satisfy the compatibility conditions presented above. To prune gross outliers, we select the largest set of compatible measurements. Finding this set can be cast as a mixed-integer linear program which we solve using the commercial solver COPT [68].

Proposition 3.3.3 (Largest Set of Compatible Measurements)**.** *Let $\mathcal{S}$ be the set of measurement pairs that do not satisfy the shape compatibility condition (3.19) and $\mathcal{T}$ be the set of groups of four measurements that do not satisfy the time compatibility condition (3.21). The largest set of measurements that satisfy both shape and time compatibility is given by the following mixed integer linear program:*

$$\begin{aligned}\underset{\boldsymbol{\theta}\in\{0,1\}^{N\times T}}{\operatorname{argmax}} \quad & \sum_{t=1}^{T}\sum_{i=1}^{N}\theta_t^i \\ s.t. \quad & \theta_t^i + \theta_t^j \leq 1 \quad \forall\ (t,i,j) \in \mathcal{S} \\ & \theta_l^i + \theta_l^j + \theta_m^i + \theta_m^j \leq 3 \quad \forall\ (l,m,i,j) \in \mathcal{T}\end{aligned} \tag{3.24}$$

where $\theta_t^i = 1$ denotes including measurement $\mathbf{y}_t^i$ in the set.

The proof of Proposition 3.3.3 follows from Propositions 3.3.1 and 3.3.2.

3.3.2 Graduated Non-Convexity for Robustness

While consistency checks can remove a significant proportion of outliers, they may miss a number of difficult-to-detect outliers. To remove these remaining outliers we use CAST* as a non-minimal solver for *graduated non-convexity* (GNC) [60]. We use the truncated least squares loss in GNC and follow the implementation and parameter choices of [60]. In our experiments, we show the combination of GNC and our compatibility checks is robust to 50-60% of outliers.

3.4 Experiments

This section characterizes CAST* and CAST#. Synthetic experiments (Section 3.4.1) show the semidefinite relaxation in CAST* is empirically tight and returns accurate estimates in the presence of noise, while CAST# is robust to 50-60% outliers. Sections 3.4.2, 3.4.3, and 3.4.4 show CAST# is competitive with other category-level approaches on two public datasets and a real-world drone-based vehicle tracking scenario. We also compare the world and body-frame motion models in the synthetic and drone experiments, showing the world-frame model has similar accuracy but substantial computational benefit.

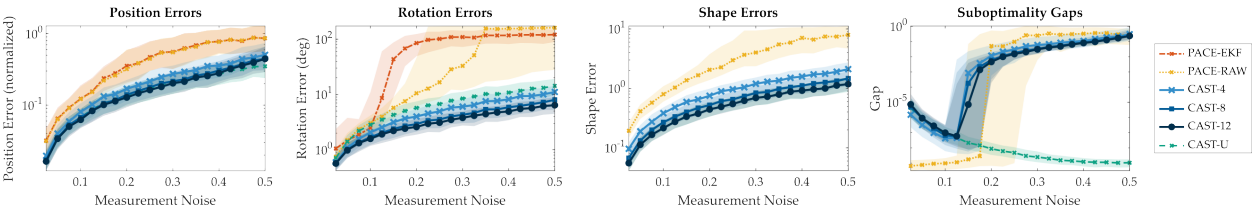
Notation

Throughout this section we default to the *body-frame* motion model, which is better physically motivated. CAST* refers to the outlier-free algorithm discussed in Section 3.2, and CAST# refers to the outlier-robust algorithm in Section 3.3, both with the body-frame model. Hyphens after CAST* or CAST# indicate perturbations of the algorithm. For example, CAST*-8 indicates CAST* with $T = 8$ frames in the fixed lag smoother. To denote the world-frame motion model, we use CAST*-W and CAST#-W. Other perturbations are CAST# with ground truth 3D keypoints that include occluded points (CAST#-GT), CAST# with ground truth pixel keypoints (depth from RGB-D image) that do not include occlusion (denoted CAST#-GTK), and CAST* with no motion model (CAST*-U). CAST*-U uses the body-frame formulation and drops velocity and angular velocity objective terms.

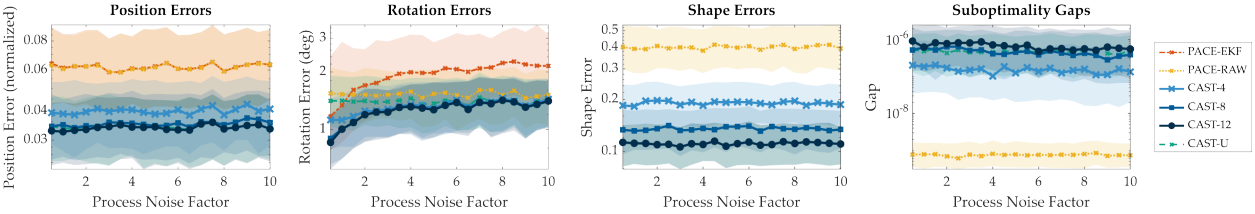
3.4.1 Optimality and Robustness in Synthetic Dataset

Dataset

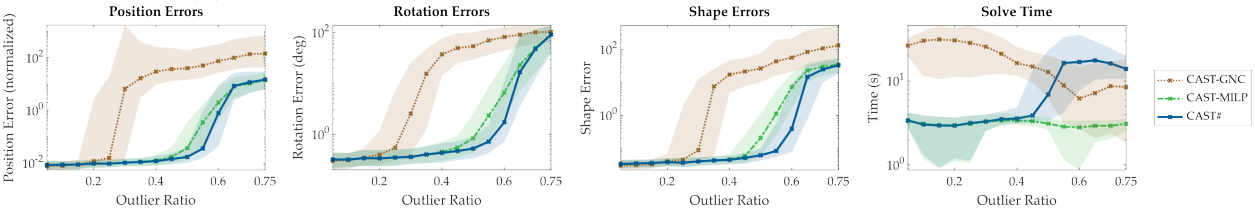
We generate keypoint measurements according to the measurement model in Section 3.1 for the body-frame motion model. Ground truth trajectories follow the corresponding motion model (3.1) and (3.3) with Gaussian velocity noise and Langevin rotation rate noise (process noise). The ground truth trajectory and randomly generated shape determine the measured keypoint positions without regard for occlusion, subject to Gaussian perturbations (measurement noise) and outliers. We use the realistic PASCAL3D+ aeroplane shape library [69] (with characteristic length $l = 0.2$ m) to generate a ground truth shape vector. In each experiment, we fix measurement noise to 5% of the characteristic length, and the process noise to 0.01 m and 0.01 rad. For the measurement noise experiment, we set the velocity weights



(a) Measurement noise robustness of CAST⋆ with fixed process noise.



(b) Process noise robustness of CAST⋆ with fixed measurement noise.



(c) Outlier robustness of CAST# and ablations.

Figure 3.3: **Performance of CAST⋆ and CAST# in synthetic experiments.** Using the PASCAL3D+ aeroplane shape library, we generate synthetic measurements to test the robustness of CAST⋆ and CAST# to measurement noise, process noise, and outliers. Plots show median and IQR of 500 runs.

$\omega_t = 1$ to improve tightness; results with standard weights for the body-frame motion model and results for the world-frame motion model are provided in Appendix A.5.

Baselines

We compare CAST⋆ against PACE [2], a certifiably optimal solver for single-frame pose estimation, and PACE+EKF, an approach that filters the pose estimate from PACE at each time using an extended Kalman filter (EKF) while using a constant-twist motion model. We test CAST⋆ with a time horizon of 4, 8, or 12 frames and label the corresponding results as CAST⋆-4, CAST⋆-8, and CAST⋆-12; we also report CAST⋆-U, which is a variant of CAST⋆-12 with no velocity or rotation smoothing ($\omega_t = 0$, $\kappa_t = 0$). Finally, to test CAST#, we replace a fraction of the measurements with random outliers normally distributed about the centroid of the object with a standard deviation equal to the characteristic length. For the tests with outliers, we compare against ablations of CAST# with only GNC or only compatibility-based (MILP) outlier rejection without GNC and use $T = 12$. In each experiment we show the median and interquartile range of the error of the last estimated state over 500 runs.

Results

Fig. 3.3 reports the median position error (as a percent of length scale), rotation error (in degrees), shape error (l_2 distance between predicted and ground truth shape vector $\mathbf{c}$), and suboptimality gap or solve time for increasing measurement noise (normalized by length scale), process noise (reported as a multiple of 5%), and outlier ratio. In all experiments, CAST* and CAST# achieve the lowest median position, rotation, and shape error. In Figs. 3.3(a) and 3.3(b), CAST* is consistently tight (suboptimality gap $< 10^{-4}$) in low to moderate noise, and still gives an accurate estimate when not tight. Interestingly, while CAST* outperforms its unsmoothed variant CAST*-U, the latter remains tight for higher measurement noise. CAST*-U benefits over PACE from access to additional measurements, and CAST* benefits over CAST*-U from additional information about the object's motion. The primary cost of CAST* compared to PACE is its runtime, which ranged from 0.1 to 7 Hz depending on the time horizon; see Appendix A.5 for detailed runtimes. Synthetic results are nearly identical for the world-frame version of CAST* and given in Appendix A.5.

We also note the poor performance of PACE-EKF in both experiments. The EKF provides some benefit for very low noise but quickly diverges for higher noise as the distribution of PACE measurements deviates from Gaussian and the dynamics are nonlinear. We provide a comparison to an EKF using perturbed ground truth poses in Appendix A.5.

The outlier experiment in Fig. 3.3(c) shows robustness to 50-60% of outliers using CAST#. Compatibility tests alone are robust to 40-50% of outliers, while GNC only tolerates 20-30% of outliers. The data show GNC and MILP-based outlier rejection are complimentary, with the fast MILP solve time being unaffected by GNC in the low outlier regime.

3.4.2 YCBInEOAT Dataset

Dataset

The YCBInEOAT dataset [30] includes 9 RGB-D videos of a robotic manipulator interacting with 5 YCB objects [70]. It includes in-hand manipulation, pick-and-place, and handovers. We train a simple RGB keypoint detector for each object using their CAD models and manually-defined semantic points. The detector has a ResNet18 backbone [71] and is trained on the BOP YCB-V synthetic dataset [72]. We report the ADD and ADD-S area under the precision-accuracy curve (AUC) scores at 0.1 m with estimated poses applied to ground truth CAD models; see [73].

Baselines

We compare against state-of-the-art instance and category-level tracking approaches for the cracker, sugar, and mustard objects. We omit the small soup object and bleach because it matches the background and gripper colors and our simple keypoint detector is unable to detect reasonable keypoints. TEASER++ [12] is the only instance-level approach and uses the same keypoints given to CAST#.

For CAST#, we group mustard and bleach into the "bottle" category along with a CAD model of a ketchup bottle [74] (3 shapes, 65 keypoints). Similarly, we group cracker and sugar into the "box" category (2 shapes, 52 keypoints).

Table 3.1: Comparison of Methods on YCBInEOAT Dataset

Method	Cracker		Sugar		Mustard		Reconst.
	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	CD (cm)
6-PACK	-	-	-	-	34.49	80.76	-
TEASER++	84.76	92.14	83.26	91.27	86.02	93.43	-
MaskFusion	79.74	88.28	36.18	45.62	11.55	13.11	-
BundleTrack	85.07	89.41	85.56	90.22	**92.26**	95.35	2.81
BundleSDF	81.44	90.76	**86.55**	**92.85**	90.83	**95.48**	1.16
CAST#-8	**86.93**	**93.14**	81.97	89.45	84.67	92.41	**0.16**
CAST#-GTK	89.00	94.09	91.05	95.27	92.18	96.17	-

Results from 6-PACK [34], MaskFusion [37], and BundleTrack [35] are taken from the results reported in [35]. BundleSDF [17] results were replicated using the open source implementation with ground truth segmentation masks. BundleSDF is the only category-free method. Since our keypoint detector is fairly simple, we also report CAST# evaluated on ground truth pixel keypoints with true depth and occlusions. For all methods we compute scores using the ground truth shape and initialize with the first frame ground truth pose. For CAST# and TEASER we do not need any initialization. We also report the chamfer distance between the tightest final shape estimate (a dense 3D model) and the true shapes averaged over all 5 objects [17].

Results

CAST# clearly outperforms baselines for the cracker object but underperforms for sugar and mustard (Table 3.1). The results are encouraging: even with a simple keypoint detector, CAST# outperforms elaborate learning-based methods. The sugar and mustard results are not far behind baselines and mostly reflect the quality of the keypoint detector, which struggles with smaller objects (see the video[2]). Given ground truth pixel keypoints, CAST# outperforms virtually all baselines, despite the low-quality depth data in the dataset. CAST# achieves a near-perfect reconstruction chamfer distance; note that the shape library includes the true model.

3.4.3 NOCS Dataset

Dataset and Baselines

We evaluate our system on NOCS-REAL275 [1] on the camera (2561 frames) and mug (2615 frames) object categories. We use a keypoint detector based on YOLOv8-pose [75] trained on synthetic images of 5 mugs and 3 cameras compiled from 3D scans because the NOCS training data does not include precise ground truth. Different from YCBInEOAT, the detector and shape library do not include the ground truth object models; this experiment

[2] https://youtu.be/eTIlVD9pDtc

Table 3.2: Comparison of Methods on NOCS Dataset

Method		6Pack	CAPTRA	BundleTrack	iCaps	CAST#-8	CAST#-GT
Initialization		Pert GT	Pert GT	Pert GT	2D seg.	**2D det.**	2D det.
camera	5°5cm	10.1	0.41	**85.8**	9.32	<u>20.68</u>	88.42
	R_{err} (deg)	35.7	17.82	**3.0**	13.69	<u>11.07</u>	2.33
	p_{err} (cm)	5.6	35.53	**2.1**	<u>2.72</u>	3.81	2.55
mug	5°5cm	24.1	**55.17**	<u>53.6</u>	21.82	35.22	97.71
	R_{err} (deg)	21.3	<u>5.36</u>	**5.2**	10.69	8.42	1.45
	p_{err} (cm)	2.3	**0.79**	2.2	<u>1.31</u>	2.46	2.18

is truly category-level. We drop the laptop due to bad training data and the symmetric objects, which CAST is not designed to handle. We report **5°5cm**: the percent of estimates within 5° and 5 cm of the ground truth, $\mathbf{R_{err}}$: the mean orientation error (degrees), and $\mathbf{p_{err}}$: the mean position error (cm). Consistent with [34], we exclude measurements with high position error (0.1 m) from R_{err} and p_{err}. We compare category-level pose tracking with 6-PACK [34], CAPTRA [36], BundleTrack [35], and iCaps [76]. iCaps and CAST# consider the more difficult problems of initializing tracking with a segmentation mask or 2D detection, respectively. Baseline results are from Table 1 in [76].

Results

CAST# achieves state of the art for the 5°5cm metric among methods that do not initialize via ground truth (Table 3.2). There is still substantial gap between CAST# and methods that rely on the less practical assumption of ground truth initialization. This reflects the quality of the keypoint detector; with ground truth 3D keypoints ignoring occlusions, CAST# is nearly perfect and outperforms all baselines except in p_{err}, which is due to the size of the shape library.

3.4.4 Drone-based Vehicle Tracking

We use the soft drone platform described in [77] to evaluate CAST# under dynamic real-world conditions, see the video[3].

During the experiment we remotely piloted a mini racecar in an elliptical trajectory while the soft drone autonomously followed using the centroid and heading derived from raw keypoints estimated at 30 Hz. Our keypoint detector, like the YCBInEOAT experiments, used a ResNet architecture [71] with 7 keypoints, and was trained on images of a similar racecar. Offline, we used motion capture to transform the 3D position of each keypoint to the world frame (to compensate for the known motion of the drone) and ran CAST# to estimate the racecar's shape and pose at each time step. For the category-level shape library we used scaled PASCAL3D+ car shapes and the racecar instance.

[3] https://youtu.be/eTIlVD9pDtc

Table 3.3: Quantitative Results of Drone Experiment

Method	ADD ↑	R_{err} (deg) ↓	p_{err} (cm) ↓	c_{err} ↓	FPS ↑
TEASER++ [12]	57.0	9.6 ± 23.2	4.3 ± 3.8	-	**39.1**
PACE [67]	52.0	12.1 ± 32.0	3.2 ± 2.4	0.79	3.94
CAST#-B4	56.6	7.6 ± 4.5	$\mathbf{2.7} \pm 1.3$	0.84	3.65
CAST#-W4	56.8	7.7 ± 4.8	$\mathbf{2.7} \pm 1.3$	0.83	18.8
CAST#-B8	58.0	7.0 ± 4.3	$\mathbf{2.7} \pm 1.4$	0.76	1.44
CAST#-W8	58.0	7.0 ± 4.3	$\mathbf{2.7} \pm 1.4$	0.76	5.02
CAST#-B12	58.8	$\mathbf{6.5} \pm 3.8$	$\mathbf{2.7} \pm 1.4$	**0.71**	0.67
CAST#-W12	**59.0**	6.7 ± 4.2	$\mathbf{2.7} \pm 1.4$	**0.71**	1.95
CAST#-U	58.2	8.8 ± 15.1	4.6 ± 20.0	**0.71**	0.91

Quantitative results of TEASER, PACE, CAST#, and variants (CAST#-B denotes the body-frame motion model and CAST#-W denotes the world-frame model) are given in Table 3.3. TEASER and PACE operate on the same raw keypoint data as CAST# and are tuned for optimal performance. Across metrics, CAST# achieves the highest accuracy and lowest mean errors. In particular, the batch approach with motion priors significantly decreases the standard deviation of rotation and position errors. While the frames per second (FPS) of CAST#-B are not competitive with TEASER, CAST#-W4 is not significantly slower than TEASER and outperforms PACE with only tradeoff in rotation error. In practice, the frame parameter can be used to tune the desired tradeoff between accuracy (using more frames) and speed (using fewer frames). CAST#-W largely outperforms CAST#-B and is significantly faster. The advantage of the more realistic motion model in CAST#-B manifests in the average rotation error. We note that the PACE implementation is a comparatively slow python-based implementation which accounts for its slower runtime.

Chapter 4

Conformalized Monocular Pose and Uncertainty Estimation

In the previous chapter we assumed the measurement noise followed an approximately Gaussian distribution. The certificate of global optimality for the maximum likelihood estimator is therefore contingent on this assumption. Further, obtaining reasonable estimates required careful handling of outliers which sacrificed the maximum likelihood guarantees. In this chapter we consider instead *bounded* measurement noise in the setting of known shape and monocular pixel measurements (*i.e.,* no depth). High probability bounds on the measurement noise are easy to obtain under mild assumptions via conformal prediction. Propagating them to a pose estimate with uncertainty, however, requires careful consideration. We propose a semidefinite relaxation to find the most likely pose estimate and an elegant approach to convert the implicit pose uncertainty set into an explicit ellipse with angular and translational uncertainty bounds. Experimental results confirm the accuracy of this method against leading learning-based approaches and previous conformalized pose and uncertainty estimation procedures.

4.1 Pose Estimation from Conformalized Pixel Keypoints

In this section we consider the *conformalized monocular pose and uncertainty estimation* problem. From a single RGB image picturing an object of known shape we seek an estimate of its 6D pose (position and orientation) with angular and translational uncertainty bounds. We assume measurements in the form of conformalized pixel prediction sets with known correspondences to the 3D object model. This section details the conformal uncertainty set measurements and propagates them to a pose uncertainty set. Distinct from [3], we explicitly propagate the coverage probability to the pose uncertainty set. Considering the pose uncertainty set as a random quantity motivates a statistically meaningful estimate of the most likely pose, as we will explore in Section 4.2. Further, we introduce bounds on the ∞-norm of the pose uncertainty set that are linear in position and orientation, improving computation.

Figure 4.1: **Conformal calibration sets.** We use conformal prediction with 2-norm uncertainty (left) or ∞-norm uncertainty (right) to obtain uncertainty sets which contain the ground truth keypoint with probability at least $1-\alpha$ (the same for every keypoint). The radius of the uncertainty sets is determined by the $1-\alpha$ quantile of calibration errors. To calibrate, we measure the distance between the detected and ground truth keypoints in pixel space using some p-norm weighted by the confidence score. The calibration images are assumed to be exchangeable with the test data (*i.e.,* independent draws from the same distribution).

4.1.1 Measurement Model: Pixel Uncertainty Sets

Our measurements are 2D pixel detections of known 3D object *keypoints*, with bounded noise in pixel space. For each of N 3D object points, we assume a front-end detects their approximate pixel location in the image frame. We obtain high-probability norm-ball uncertainty bounds on the pixel measurements (see Fig. 4.1) using conformal calibration data from exchangeable images. This leads to the following bounded uncertainty model.

Denote the known 3D object points in the object frame by $\mathbf{b}_i \in \mathbb{R}^3$ for $i = 1, ..., N$ and their corresponding pixel measurements by $\mathbf{y}_i = [u_i, v_i, 1]^\mathsf{T}$. For an object with position $\mathbf{t} \in \mathbb{R}^3$ and orientation $\mathbf{R} \in \mathrm{SO}(3)$, and a camera with known intrinsics $\mathbf{K}$, the keypoint measurement model is:

$$\mathbf{y}_i = \frac{\mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t})}{\hat{\mathbf{e}}_3 \cdot \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t})} + \boldsymbol{\epsilon}_i \tag{4.1}$$

where $\boldsymbol{\epsilon}_i \in \mathbb{R}^2$ is some random measurement noise in homogeneous form ($(\boldsymbol{\epsilon}_i)_3 = 0$) and $\hat{\mathbf{e}}_3 \triangleq [0, 0, 1]^\mathsf{T}$.

We make no distributional assumptions for $\boldsymbol{\epsilon}_i$. Instead, we assume its p-norm is bounded with probability at least $1 - \alpha_i$:

$$\mathbb{P}(\|\boldsymbol{\epsilon}_i\|_p \leq r_i(\alpha_i)) \geq 1 - \alpha_i \tag{4.2}$$

This type of bound is easy to obtain with relatively small amounts of calibration data through *split conformal prediction* [53]. For now, take $r_i(\alpha_i)$ as a given constant for any choice of α_i; we explain our split conformal prediction procedure in Section 4.4.

4.1.2 Pose Uncertainty Set

From the measurement model (4.1) and noise bound (4.2) we seek a *pose uncertainty set* that contains the true pose with high probability. Begin by rephrasing the measurements. Eqs. (4.1, 4.2) bound the maximum error of the keypoint measurement. Thus, the following reprojection constraint holds with probability at least $1 - \alpha_i$ for all $i = 1, ..., N$:

$$\left\| \mathbf{y}_i - \frac{\mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t})}{\hat{\mathbf{e}}_3 \cdot \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t})} \right\|_p = \|\boldsymbol{\epsilon}_i\|_p \leq r_i(\alpha_i) \tag{4.3}$$

Since we assume the object is in front of the camera, the projection of $\mathbf{b}_i$ must have positive depth. This is the *chirality* (front-of-camera) constraint for keypoint i.

$$\hat{\mathbf{e}}_3 \cdot \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) > 0 \tag{FoC}$$

To proceed we convert the rational reprojection constraint (4.3) to a polynomial backprojection constraint and specialize to the cases $p = \infty$ and $p = 2$. As pictured in Fig. 4.1, the ∞-norm bound is equivalent to square axis-aligned uncertainty sets with side length $2r_i(\alpha_i)$, while the 2-norm bound gives circular uncertainty sets of radius $r_i(\alpha_i)$.

∞-Norm Backprojection Constraints

When $p = \infty$, (4.3) reduces to two inequality constraints which hold with probability at least $1 - \alpha_i$ for all $i = 1, ..., N$:

$$\begin{cases} \left| \left[\mathbf{y}_i \hat{\mathbf{e}}_3^\mathsf{T} \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) - \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) \right] \cdot \hat{\mathbf{e}}_1 \right| \leq r_i(\alpha_i) \hat{\mathbf{e}}_3^\mathsf{T} \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) \\ \left| \left[\mathbf{y}_i \hat{\mathbf{e}}_3^\mathsf{T} \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) - \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) \right] \cdot \hat{\mathbf{e}}_2 \right| \leq r_i(\alpha_i) \hat{\mathbf{e}}_3^\mathsf{T} \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) \end{cases} \tag{BP$_\infty$}$$

To derive this equation we multiplied (4.3) by the depth and used the chirality constraint (FoC) to drop its absolute value on the right hand side. The infinity norm constrains each coordinate independently; we drop the third coordinate since $(\mathbf{y}_i)_3 = 1$, so the left hand side would always be zero. Note that (BP$_\infty$) may be written as four inequality constraints which are linear in $\mathbf{R}$ and $\mathbf{t}$.

2-Norm Backprojection Constraints

For $p = 2$ we square the inequality (4.3) and multiply through by depth. This gives the following inequality constraint which holds with probability at least $1 - \alpha_i$ for all $i = 1, ..., N$:

$$\left\| \mathbf{y}_i \hat{\mathbf{e}}_3^\mathsf{T} \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) - \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) \right\|_2^2 \leq (r_i(\alpha_i) \hat{\mathbf{e}}_3^\mathsf{T} \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}))^2 \tag{BP$_2$}$$

We observe that (BP$_2$) is a quadratic function of $\mathbf{t}$ and $\text{vec}(\mathbf{R})$. In the following sections we refer to (BP$_\infty$) or (BP$_2$) as (BP$_p$), where $p = 2$ or $p = \infty$.

Pose Uncertainty Set

We are now ready to construct the pose uncertainty set. Combining the backprojection and chirality constraints for each keypoint gives the following proposition.

Proposition 4.1.1 (Pose Uncertainty Set)**.** *Assume measurements of the form (4.1) with noise bounded in p-norm by $r_i(\alpha_i)$ with probability at least $1-\alpha_i$ for $i = 1, .., N$ (that is, satisfying eq. 4.2). The true position $\mathbf{t}_{\text{gt}}$ and orientation $\mathbf{R}_{\text{gt}}$ are contained in the following pose uncertainty set:*

$$\left\{ \begin{array}{l} \mathbf{t} \in \mathbb{R}^3 \\ \mathbf{R} \in \text{SO}(3) \end{array} \;\middle|\; (\text{FoC}) \text{ and } (\text{BP}_p) \text{ for } i = 1, ..., N \right\} \tag{$\mathcal{P}_p$}$$

with probability at least β.

In particular, for arbitrary dependence among $\boldsymbol{\epsilon}_i$ we have $\beta \geq 1 - \sum_{i=1}^{N} \alpha_i$. In the extreme where $\boldsymbol{\epsilon}_i$ is independent from $\boldsymbol{\epsilon}_j$ for all $i \neq j$ then $\beta = \prod_{i=1}^{N}(1-\alpha_i)$. When the $\boldsymbol{\epsilon}_i$ are perfectly positively correlated and all $\alpha_i \equiv \alpha$, we have $\beta = 1 - \alpha$.

Proof. For perfect correlation the probability of (BP_p) for all i is the same as the probability for any choice of i. Under independence, the probability of the intersection of (BP_p) for each i reduces to a product. For arbitrary dependence, a union bound gives:

$$\mathbb{P}\left[\bigcap_{i=1}^{N} (\text{BP}_p)_i\right] = 1 - \mathbb{P}\left[\bigcup_{i=1}^{N} (\text{BP}_p)_i\right] \geq 1 - \sum_{i=1}^{N} \mathbb{P}\left[(\text{BP}_p)_i\right] = 1 - \sum_{i=1}^{N} \alpha_i \tag{4.4}$$

□

Several remarks are in order. The set $(\mathcal{P}_p)$ is an *implicit* representation of the set of poses which are consistent with the conformalized keypoint measurements (*i.e.,* the keypoint measurements satisfying eq. 4.2). It is not immediately clear how to generate poses which satisfy $(\mathcal{P}_p)$ beyond sampling [3]. It is also unclear how to obtain explicit angular or translation bounds from the set of inequalities.

Lastly, we comment on the statistical guarantee of coverage of the ground truth pose. We note that the worst case confidence can quickly become uninformative. For example, if $\alpha_i = 0.1$ for all i and $N = 10$, the set $(\mathcal{P}_p)$ is guaranteed to contain the ground truth with probability at least 0. In practice, however, we expect some positive correlation; thus, the coverage should exceed the independent case. For our choice of $\alpha_i = 0.1$ and $N = 10$, we expect the set $(\mathcal{P}_p)$ to contain the ground truth pose at least 35% of the time. We provide empirical coverage results in Section 4.4, but note that coverage is difficult to verify due to unreliable ground truth pose annotation in real data.

We now state the *conformalized monocular pose and uncertainty estimation* problem.

Problem 4.1.1. *Given 2D keypoint uncertainty sets of the form (4.1), compute an estimate of the ground truth object pose ($\mathbf{t}$, $\mathbf{R}$) with angular and translational bounds that hold with probability at least β, where β is given in Proposition 4.1.1.*

4.2 Obtaining a Pose Estimate

In the previous section we reformulated explicit high-probability bounds on keypoint measurement noise into an implicit pose uncertainty set of the form ($\mathcal{P}_p$). Our primary goal in this section is to compute an accurate estimate of the ground truth pose from these high-probability bounds. Yang and Pavone [3] take the mean of poses sampled from ($\mathcal{P}_p$) as their estimate, which requires a computationally-intensive sampling procedure. In contrast, we search for the mode: the most likely pose given uncertainty bounds. It is natural that this mode is independent of the choice of confidence $\boldsymbol{\alpha}$, unlike the mean sampling procedure in [3]. This handles the case where ($\mathcal{P}_p$) is empty and can be solved from a single optimization problem.

To find the mode of the distribution we first allow confidence $\boldsymbol{\alpha}$ to vary. Explicitly writing the dependence on $\boldsymbol{\alpha}$, recall that $\mathcal{P}_p(\boldsymbol{\alpha})$ denotes the family of pose uncertainty sets parameterized by confidence $1-\alpha_i$ for each keypoint i. Intuitively, lower confidence (larger α_i) gives tighter keypoint uncertainty sets, which in turn contain fewer poses. The mode of the distribution is contained in the tightest keypoint uncertainty sets which still contain at least one pose. The following proposition formalizes this intuition.

Proposition 4.2.1 (Most Likely Pose)**.** *The most likely pose under the pose uncertainty sets $\mathcal{P}_p(\boldsymbol{\alpha})$ is given by the solution to the following optimization problem.*

$$\begin{aligned} \min_{\substack{\mathbf{t}\in\mathbb{R}^3,\,\mathbf{R}\in\mathrm{SO}(3)\\ \boldsymbol{\alpha}\in[0,1]^N}} \quad & \sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & (\mathbf{t},\mathbf{R})\in\mathcal{P}_p(\boldsymbol{\alpha}) \end{aligned} \tag{4.5}$$

Proof. For ground truth translation $\mathbf{t}_{\text{gt}}$ and orientation $\mathbf{R}_{\text{gt}}$, Proposition 4.1.1 gives:

$$\mathbb{P}\left[(\mathbf{t}_{\text{gt}},\mathbf{R}_{\text{gt}})\in\mathcal{P}_p(\boldsymbol{\alpha})\right]\geq 1-\sum_{i=1}^{N}\alpha_i \tag{4.6}$$

Allowing α_i to vary between 0 and 1 for $i=1,...,N$ and maintaining the constraints $\mathcal{P}_p(\boldsymbol{\alpha})$, the most likely pose (*i.e.,* the mode) maximizes the quantity $1-\sum_{i=1}^{N}\alpha_i$. Dropping constants in the objective, we arrive at the minimization problem (4.5). □

While compact, (4.5) is difficult to solve in practice because it requires explicitly computing uncertainty bounds for each keypoint at a variety of confidences. Instead, we approximate the $\boldsymbol{\alpha}$ dependence as a linear multiplier (see Fig. 4.2 for intuition). That is, for fixed confidence $\tilde{\alpha}$:

$$r_i(\alpha_i)\approx\alpha_i r_i(\tilde{\alpha})\triangleq\alpha_i r_i \tag{4.7}$$

By adopting this model we sacrifice statistical guarantees for our estimate of the most likely pose. However, the model is not unreasonable. Larger α_i leads to larger sets, and smaller α_i leads to smaller sets. Importantly, this model has some dependence on the choice of fixed confidence α. Adopting this model, we can rewrite the most likely pose problem as a polynomial optimization problem, which is amenable to both local optimization [78] and semidefinite relaxation [79]. We call the most likely pose problem with approximated confidence dependence the *central pose* problem.

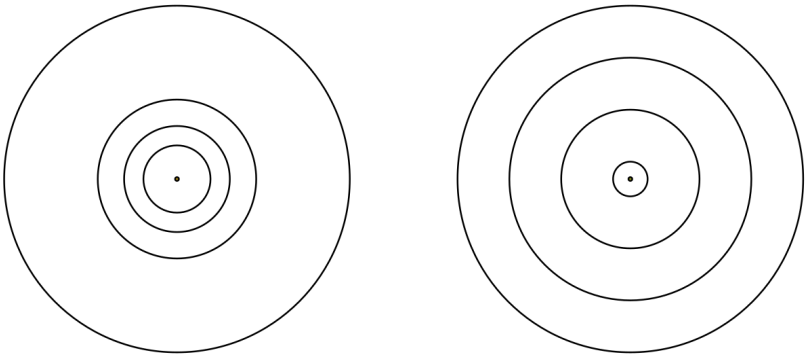


Figure 4.2: **Multiplicative approximation of conformal sets.** We approximate the conformal prediction uncertainty bounds, which vary by confidence α, by multiplying by the radius at fixed confidence by a constant $\gamma > 0$. Left, the true conformal sets about a keypoint at confidences $0.1, 0.3, 0.5,$ and 0.7. Right, the conformal set at confidence 0.1 multiplied by the equally-spaced constants $1, 0.7, 0.4,$ and 0.1. The approximation is crude, but captures the behavior for moderate α and signficantly simplies computation. It is particularly bad as α approaches 0 or 1.

4.2.1 Convex Relaxation

As in Chapter 3, we solve the central pose problem using a semidefinite relaxation. To proceed, we consider the $p = \infty$ and $p = 2$ cases separately. The $p = \infty$ case reduces to a quadratically-constrained linear program which may be solved using Shor's relaxation as reviewed in Section 2.1. The $p = 2$ case is a quartically-constrained linear program and thus requires more advanced tools. In practice, we solve both using the sparse moment sum-of-squares hierarchy [79]. Although the relaxation is not tight in practice, the relaxation often provides a good initial guess to a local solver.

For $p = \infty$, the central pose is given by the solution to the following optimization problem:

$$\begin{aligned} \min_{\substack{\mathbf{t}\in\mathbb{R}^3,\, \mathbf{R}\in\mathrm{SO}(3) \\ \boldsymbol{\gamma}\in[l,u]^N}} \quad & \sum_{i=1}^{N} \gamma_i \\ \text{s.t.} \quad & \left\|(\mathbf{y}_i\hat{\mathbf{e}}_3^\mathsf{T} - \mathbf{I}_3)\mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t})\right\|_\infty \leq \gamma_i r_i \hat{\mathbf{e}}_3^\mathsf{T}\mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}), \\ & \hat{\mathbf{e}}_3 \cdot \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) > 0, \quad i = 1, ..., N \end{aligned} \tag{4.8}$$

Note that eq. (4.8) has a linear objective with quadratic inequality constraints ($\gamma_i\mathbf{R}$ and $\gamma_i\mathbf{t}$) and quadratic equality constraints ($\mathbf{R} \in \mathrm{SO}(3)$). Thus, it is a non-convex problem which directly admits a convex semidefinite relaxation as described in Section 2.1. In practice, we first solve the relaxed convex program with a first-order relaxation formulated in TSSOS [79]. After obtaining a solution to the relaxed problem (a lower bound f_{sdp} on the optimal objective $f_\star$), we use Ipopt [78], a local nonlinear solver, to obtain an upper bound f_{local} and feasible solution. This gives a *suboptimality gap* g:

$$g = \frac{f_{\text{local}} - f_{\text{sdp}}}{\max(1, |f_{\text{local}}|)} \tag{4.9}$$

When the gap is small (less than 10^{-3}) it serves as a certificate of global optimality. When the gap is large we have converged to a local stationarity point which may or may not be globally optimal. In practice, we rarely observe a small optimality gap. To bound γ, we find $l = 0.01$ and $u = 10$ to work well in practice.

We use a similar approach for $p = 2$, except we must resort to a second-order relaxation. The central pose is given by the solution to the following optimization:

$$\begin{aligned} \min_{\substack{\mathbf{t}\in\mathbb{R}^3,\, \mathbf{R}\in\mathrm{SO}(3) \\ \boldsymbol{\gamma}\in[l,u]^N}} \quad & \sum_{i=1}^{N} \gamma_i \\ \text{s.t.} \quad & \left\|(\mathbf{y}_i\hat{\mathbf{e}}_3^\mathsf{T} - \mathbf{I}_3)\mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t})\right\|_2^2 \leq \left(\gamma_i r_i \hat{\mathbf{e}}_3^\mathsf{T}\mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t})\right)^2, \\ & \hat{\mathbf{e}}_3 \cdot \mathbf{K}(\mathbf{R}\mathbf{b}_i + \mathbf{t}) > 0, \quad i = 1, ..., N \end{aligned} \tag{4.10}$$

Eq. (4.10) has biquadratic inequality constraints in addition to the quadratic equality constraints $\mathbf{R} \in \mathrm{SO}(3)$. Thus, we require a second-order convex relaxation. We use TSSOS [79] to formulate the relaxation and exploit term sparsity for speed. As before, we use Ipopt for a local solution and to obtain an optimality gap. The solution to this problem is rarely tight in practice, making the relaxation largely useful for obtaining a good initial guess for the local solver.

4.3 Pose Uncertainty Bounds

In addition to a pose estimate we seek a numerical representation of pose uncertainty in the form of explicit angular and translational bounds. In this section, we convert the implicit pose uncertainty set ($\mathcal{P}_p$) into a single ellipsoidal constraint centered at the pose estimate from Section 4.2. This ellipse is amenable to a simple projection scheme for translation bounds and a tight semidefinite relaxation for axis-aligned angular bounds. The simplicity of this approach stands in contrast with the sampling-based inner approximation in [24] and the computationally expensive outer approximation which uses a third-order sum-of-squares relaxation in [25].

Throughout this section we will consider only the $p = 2$ case. We find the $p = \infty$ case fails to solve without explicitly including additional implied quadratic constraints; we leave exploration of this to future work.

4.3.1 Quadratic Form of Pose Uncertainty Set

Unlike the previous section, we seek uncertainty bounds at a fixed confidence $\boldsymbol{\alpha} = \tilde{\alpha}\mathbf{1}_N$. We begin by writing the pose uncertainty set ($\mathcal{P}_2$) as a set of quadratic constraints. Let $\mathbf{r} \triangleq \mathrm{vec}(\mathbf{R}) \in \mathbb{R}^9$, where the vec operator stacks the columns of $\mathbf{R}$. The constraint $\mathbf{R} \in \mathrm{SO}(3)$ may be written as 15 equality constraints which are quadratic in $\mathbf{r}$; see Section 2.1 and Appendix B.1. Observe that the product $\mathbf{K}\mathbf{R}\mathbf{b}_i = (\mathbf{b}_i^\mathsf{T} \otimes \mathbf{K})\mathbf{r}$, where $\otimes$ denotes a Kronecker product. Thus, the chirality constraints (FoC) are linear in $(\mathbf{r}, \mathbf{t})$ and the 2-norm backprojection constraints (BP_2) are quadratic in the variables. Below we derive the explicit quadratic forms.

Let $\mathbf{x} \triangleq [\mathbf{r}, \mathbf{t}, 1]^\mathsf{T} \in \mathbb{R}^{13}$. The chirality constraints (FoC) can be written as:

$$(\text{FoC})_i \iff \mathbf{x}^\mathsf{T} \begin{bmatrix} \mathbf{0} & \mathbf{d}_i \\ \mathbf{d}_i^\mathsf{T} & 0 \end{bmatrix} \mathbf{x} < 0 \tag{4.11}$$

where the vector $\mathbf{d}_i$ is given by $\mathbf{d}_i \triangleq -\begin{bmatrix} \hat{\mathbf{e}}_3^\mathsf{T}(\mathbf{b}_i^\mathsf{T} \otimes \mathbf{K}) & \hat{\mathbf{e}}_3^\mathsf{T}\mathbf{K} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{12}$ for $i = 1, ..., N$.

Similarly, the 2-norm backprojection constraints (BP_2) can be rewritten as:

$$(\text{BP}_2)_i \iff \mathbf{x}^\mathsf{T} \begin{bmatrix} \mathbf{C}_r^\mathsf{T}\mathbf{C}_r & \mathbf{C}_r^\mathsf{T}\mathbf{D}_t & \mathbf{0} \\ \mathbf{D}_t^\mathsf{T}\mathbf{C}_r & \mathbf{D}_t^\mathsf{T}\mathbf{D}_t & \\ \mathbf{0} & & 0 \end{bmatrix} \mathbf{x} \leq 0 \tag{4.12}$$

where we define the following constant matrices, recalling $r_i \triangleq r_i(\tilde{\alpha})$:

$$\mathbf{C}_r \triangleq \begin{bmatrix} (\mathbf{I}_3 - \mathbf{y}_i\hat{\mathbf{e}}_3)(\mathbf{b}_i \otimes \mathbf{K}) \\ r_i\hat{\mathbf{e}}_3^\mathsf{T}(\mathbf{b}_i \otimes \mathbf{K}) \end{bmatrix} \in \mathbb{R}^{6\times 9} \quad \text{and} \quad \mathbf{D}_t \triangleq \begin{bmatrix} (\mathbf{I}_3 - \mathbf{y}_i\hat{\mathbf{e}}_3)\mathbf{K} \\ r_i\hat{\mathbf{e}}_3^\mathsf{T}\mathbf{K} \end{bmatrix} \in \mathbb{R}^{6\times 3} \tag{4.13}$$

Thus, we can rewrite ($\mathcal{P}_2$) with constraints in their quadratic form:

$$\left\{ \begin{array}{c|lc} \mathbf{x} \in \mathbb{R}^{13} & \mathbf{x}^\mathsf{T}\mathbf{Q}_k\mathbf{x} = 0, \;\; k = 1, ..., 15 & (\mathbf{R} \in \text{SO}(3)) \\ x_{13} = 1 & \mathbf{x}^\mathsf{T}\mathbf{A}_j\mathbf{x} \leq 0, \;\; j = 1, ..., N & (\text{BP}_p) \\ & \mathbf{x}^\mathsf{T}\mathbf{B}_i\mathbf{x} \leq 0, \;\; i = 1, ..., N & (\text{FoC}) \end{array} \right\} \tag{4.14}$$

The matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ can be read from (4.12) and (4.11), respectively.

4.3.2 Reduction to a Single Bounding Ellipse

We now reduce (4.14) into a single ellipsoidal constraint centered at the central pose estimate. Let $\bar{\mathbf{x}} \triangleq \begin{bmatrix} \bar{\mathbf{r}}^\mathsf{T} & \bar{\mathbf{t}}^\mathsf{T} & 1 \end{bmatrix}^\mathsf{T}$ denote the central pose estimate obtained in Section 4.2. We seek a bounding ellipse of the form:

$$\begin{bmatrix} \mathbf{r} - \bar{\mathbf{r}} \\ \mathbf{t} - \bar{\mathbf{t}} \end{bmatrix}^\mathsf{T} \mathbf{H} \begin{bmatrix} \mathbf{r} - \bar{\mathbf{r}} \\ \mathbf{t} - \bar{\mathbf{t}} \end{bmatrix} \leq 1 \tag{4.15}$$

The size and shape of the ellipse is defined by the unknown matrix $\mathbf{H} \succ 0$. Notice that (4.15) can also be written in quadratic form similar to (4.14):

$$(4.15) \iff \mathbf{x}^\mathsf{T} \begin{bmatrix} \mathbf{H} & -\mathbf{H}\bar{\mathbf{x}} \\ \bar{\mathbf{x}}^\mathsf{T}\mathbf{H} & \bar{\mathbf{x}}^\mathsf{T}\mathbf{H}\bar{\mathbf{x}} - 1 \end{bmatrix} \mathbf{x} \leq 0 \tag{4.16}$$

To reduce the pose uncertainty set into a single ellipsoidal constraint, we seek $\mathbf{H}$ such that $(\mathbf{r}, \mathbf{t}) \in (\mathcal{P}_2) \implies (4.15)$. Ideally, we could find $\mathbf{H}$ using a convex program and without simultaneously solving for a value of $\mathbf{x}$. The following lemma is a generalization of the celebrated S-Lemma [80] and allows us to do just that.

Lemma 4.3.1 (Generalized S-Lemma)**.** *Let $\mathbf{x} \in \mathbb{R}^n$ be a vector and $\mathbf{W}$, $\mathbf{Y}_i$, $\mathbf{Z}_j \in \mathcal{S}^n$ be symmetric matrices for $i = 1, ..., N$ and $j = 1, ..., M$. For the following statements,*
(i) $\implies$ (ii):

(i) There exists $\boldsymbol{\mu} \in \mathbb{R}^M$ *and* $\boldsymbol{\lambda} \in \mathbb{R}^N$, $\lambda_i \geq 0 \; \forall \; i$ *such that* $\mathbf{W} \preceq \sum_{i=1}^N \lambda_i \mathbf{Y}_i + \sum_{j=1}^M \mu_j \mathbf{Z}_j$.

(ii) $\mathbf{x}^\mathsf{T}\mathbf{Y}_i\mathbf{x} \leq 0$ *for* $i = 1, ..., N$ *and* $\mathbf{x}^\mathsf{T}\mathbf{Z}_j\mathbf{x} = 0$ *for* $j = 1, ..., M \implies \mathbf{x}^\mathsf{T}\mathbf{W}\mathbf{x} \leq 0$.

Proof. Take the quadratic form of (i) with $\mathbf{x}$:

$$\mathbf{x}^\mathsf{T}\mathbf{W}\mathbf{x} \leq \sum_{i=1}^N \lambda_i \mathbf{x}^\mathsf{T}\mathbf{Y}_i\mathbf{x} + \sum_{j=1}^M \mu_j \mathbf{x}^\mathsf{T}\mathbf{Z}_j\mathbf{x} \tag{4.17}$$

For non-negative $\boldsymbol{\lambda}$ and under the conditions of the left hand side of (ii), the result holds. □

To obtain a bounding ellipse, we only need to find $\mathbf{H}$ which satisfies statement (i) of Lemma 4.3.1. Matching to the lemma, let $\mathbf{W}$ be the constant matrix in (4.16) which includes the unknown matrix $\mathbf{H}$. Let the inequality matrices $\mathbf{Y}_i$ correspond to the chirality and backprojection constraints $\mathbf{A}_i$ and $\mathbf{B}_i$, $i = 1, ..., N$. Lastly, let the equality matrices $\mathbf{Z}_j$ be the 15 equalities $\mathbf{Q}_j$ which constrain $\mathbf{R} \in \mathrm{SO}(3)$. This result is summarized in the following proposition.

Proposition 4.3.2 (Bounding Ellipsoid)**.** *An outer bounding ellipsoid for the set* (4.14) *is given by the solution to the following convex optimization problem.*

$$\begin{aligned} \max_{\substack{\mathbf{H}\in\mathcal{S}^{12},\,\mathbf{H}\succeq 0 \\ \boldsymbol{\lambda}\in\mathbb{R}^{2N},\,\boldsymbol{\mu}\in\mathbb{R}^{15}}} \quad & \log\det(\mathbf{H}) \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{H} & -\mathbf{H}\bar{\mathbf{x}} \\ \bar{\mathbf{x}}^\mathsf{T}\mathbf{H} & \bar{\mathbf{x}}^\mathsf{T}\mathbf{H}\bar{\mathbf{x}} - 1 \end{bmatrix} \preceq \sum_{i=1}^N \lambda_i \mathbf{A}_i + \sum_{i=1}^N \lambda_{N+i} \mathbf{B}_i + \sum_{k=1}^{15} \mu_k \mathbf{Q}_k, \\ & \lambda_i \geq 0, \quad i = 1, ..., 2N \end{aligned} \tag{4.18}$$

The maximizer $\mathbf{H}^\star$ *defines an ellipse centered at* $\bar{\mathbf{x}}$ *as in* (4.15).

The proof follows directly from Lemma 4.3.1. Notice that the bounding ellipse is a *relaxation* of the implicit pose uncertainty set ($\mathcal{P}_2$). Crucially, the problem is convex and thus may be solved efficiently. The objective $\log\det(\mathbf{H})$ in (4.18) seeks the minimum volume ellipse which encloses ($\mathcal{P}_2$). When solved to optimality (4.18) gives the tightest bounding ellipse which also satisfies statement (i) of Lemma 4.3.1. This is not the same as the tightest bounding ellipse, since that ellipse may not have a representation consistent with statement (i). In Section 4.4 we show the ellipse generated by Proposition 4.3.2 is empirically accurate and reasonably tight.

4.3.3 Explicit Rotation and Translation Bounds

Recall that the primary motivation for reducing ($\mathcal{P}_2$) to a single bounding ellipsoid constraint was to obtain explicit uncertainty bounds in rotation and translation. The ellipse (4.15) is still difficult to interpret as an explicit uncertainty set because it is joint in the rotation and translation vectors. Fortunately, its compactness makes it amenable to marginalization via projection into a translation and rotation ellipse which are each interpretable as explicit uncertainty bounds.

To obtain translation bounds we project $\mathbf{H}$ onto its last three coordinates via orthogonal projection. Define the projection matrix $\mathbf{P}_t = \begin{bmatrix}\mathbf{0}_{3\times 9} & \mathbf{I}_3\end{bmatrix}$. The set of translations $\mathbf{t} \in \mathbb{R}^3$ satisfying ($\mathcal{P}_2$) also satisfy the following ellipse.

$$(\mathbf{t} - \bar{\mathbf{t}})^\mathsf{T} \mathbf{H}_t (\mathbf{t} - \bar{\mathbf{t}}) \leq 1 \quad \text{with} \quad \mathbf{H}_t \triangleq \left(\mathbf{P}_t \mathbf{H}^{-1} \mathbf{P}_t^\mathsf{T}\right)^{-1} \tag{4.19}$$

We observe that the ellipse (4.19) is directly interpretable as Euclidean error bounds on the translation. This marginalization scheme is simple but somewhat crude; it does not enforce SO(3) constraints on the first 9 coordinates of the original ellipse $\mathbf{H}$. In Section 4.4 we observe that this set is particularly loose along the optical axis of the camera and expands to include the camera origin.

Projecting the full ellipse onto the rotation vector proceeds similarly. Let $\mathbf{P}_r = \begin{bmatrix}\mathbf{I}_9 & \mathbf{0}_{9\times 3}\end{bmatrix}$ be the projection matrix onto the first 9 coordinates of $\mathbf{H}$. Any rotation $\mathbf{r} = \text{vec}(\mathbf{R})$ in ($\mathcal{P}_2$) obeys the following ellipsoidal constraint.

$$(\mathbf{r} - \bar{\mathbf{r}})^\mathsf{T} \mathbf{H}_r (\mathbf{r} - \bar{\mathbf{r}}) \leq 1 \quad \text{with} \quad \mathbf{H}_r \triangleq \left(\mathbf{P}_r \mathbf{H}^{-1} \mathbf{P}_r^\mathsf{T}\right)^{-1} \tag{4.20}$$

Unlike the translation ellipse (4.19), it is not clear how to interpret this ellipse over rotations as explicit angular bounds for the maximum rotation about the x, y, and z axes. A simple solution is to pose three additional rotation-constrained optimization problems that solve for explicit bounds. This result is summarized in the following proposition.

Proposition 4.3.3 (Explicit Angular Bounds)**.** *Let $\mathbf{R}_j(\theta_j)$ be the rotation by angle θ_j about the x, y, or z axes for $j = 1, 2, 3$ respectively. Assume $\theta_j < \pi/2$ for $j = 1, 2, 3$ under (4.20). The maximum angular deviation from $\bar{\mathbf{R}}$ under the constraint (4.20) about axis j is given by the solution to the following problem:*

$$\begin{aligned} \min_{\theta_j \in [-\frac{\pi}{2}, \frac{\pi}{2}]} \quad & \cos(\theta_j) \\ \text{s.t.} \quad & \left[\text{vec}(\mathbf{R}_j(\theta_j)\bar{\mathbf{R}} - \bar{\mathbf{R}})\right]^\mathsf{T} \mathbf{H}_r \left[\text{vec}(\mathbf{R}_j(\theta_j)\bar{\mathbf{R}} - \bar{\mathbf{R}})\right] \leq 1, \\ & \mathbf{R}_j(\theta_j) \in \text{SO}(3) \end{aligned} \tag{4.21}$$

We remark that Proposition 4.3.3 is a non-convex optimization problem. Fortunately, it admits a convex relaxation through a simple reparameterization. Noting that $\cos(\theta_j) \geq 0 \implies \theta_j \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, replace the angular dependence with sines and cosines. For axis-aligned rotations, the matrix $\mathbf{R}_j(\theta_j)$ requires only two variables and is subject to a single constraint. The result is a linear objective with quadratic constraints, where only the rotation constraints introduce non-convexity. We write this explicitly for $j = 1$ below.

$$\begin{aligned} \min_{\substack{\cos(\theta_1) \geq 0 \\ \sin(\theta_1)}} \quad & \cos(\theta_1) \\ \text{s.t.} \quad & \left[\text{vec}(\mathbf{R}_1(\theta_1)\bar{\mathbf{R}} - \bar{\mathbf{R}})\right]^\mathsf{T} \mathbf{H}_r \left[\text{vec}(\mathbf{R}_1(\theta_1)\bar{\mathbf{R}} - \bar{\mathbf{R}})\right] \leq 1, \\ & \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & -\sin(\theta_1) \\ 0 & \sin(\theta_1) & \cos(\theta_1) \end{bmatrix} = \mathbf{R}_1(\theta_1), \\ & \cos^2(\theta_1) + \sin^2(\theta_1) = 1 \end{aligned} \tag{4.22}$$

We solve this reduced problem using Shor's relaxation (see Section 2.1) for $j = 1, 2, 3$, giving three symmetric angular uncertainty bounds. The semidefinite relaxation is fast and empirically tight; thus, it gives guaranteed upper bounds on the maximum angular deviation from $\bar{\mathbf{R}}$ satisfying ($\mathcal{P}_2$). Due to the relaxation to a bounding ellipse we have no guarantees of tightness.

We conclude with two remarks. To obtain explicit angular bounds we require the assumption $\theta_j \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. We argue this assumption is not too limiting, since angular bounds greater than 90 degrees are not very useful in practice. In particular, the bounding approach we propose is not designed to handle multimodal pose distributions. Secondly, the reader may question the need for a bounding ellipse. The key utility lies in simplifying the pose uncertainty constraints *before* solving for a rotation and translation. In our experience, solving for explicit bounds with just the constraints ($\mathcal{P}_2$) either does not solve to optimality or does not achieve tightness. Either scenario results in an uncertainty bound which may not capture the full extent of the set.

4.4 Experiments

In this section we evaluate our pose estimation and uncertainty algorithms on the LineMOD-Occluded (LM-O) dataset [81]. We begin by describing our measurements and the conformal prediction procedure for obtaining high-probability uncertainty bounds. Section 4.4.1 also validates the empirical coverage of keypoints and poses. Next, we compare our central pose algorithms at two different confidences against other monocular pose estimation procedures. Finally, in Section 4.4.3 we compare the tightness of our angular and translational uncertainty bounds against prior work [25]. We also give a breakdown of the runtimes of each component of our method.

4.4.1 Keypoint Bounds and Empirical Coverage Results

Our algorithms take as input a set of pixel keypoint detections $\mathbf{y}_i$ and associated uncertainty bounds $r_i(\alpha)$. To detect keypoints, we use the heatmap-based convolutional neural network from [8] and a manually-defined 3D keypoint library. This network was trained on the BOP synthetic image split [82]. Like [3], we use the highest-confidence pixel in the heatmap as a detection and the associated confidence as confidence score.

The keypoint uncertainty bounds $r_i(\alpha)$ come from split conformal prediction, reviewed in detail in Section 2.2. We calibrate on the 200 real images comprising the BOP subset [83]. Although this violates the assumptions of Theorem 2.2.1 since the test images include calibration images, we use this calibration for fairness with [3]. For each keypoint i in each image, denote the detection $\mathbf{y}_i$ and the confidence $c_i \in [0, 1]$. Let $\mathbf{z}_i \triangleq \frac{\mathbf{K}(\mathbf{R}\mathbf{b}_i+\mathbf{t})}{\hat{\mathbf{e}}_3 \cdot \mathbf{K}(\mathbf{R}\mathbf{b}_i+\mathbf{t})}$ be the ground truth pixel coordinate as in (4.1). For conformal prediction, we use a confidence-weighted score function:

$$s((\mathbf{y}_i, c_i), \mathbf{z}_i) = c_i \|\mathbf{y}_i - \mathbf{z}_i\|_p \tag{4.23}$$

Let $\tilde{r}_i(\alpha)$ be the $1-\alpha$ quantile of the scores of keypoint i on all the calibration images (note that $\tilde{r}_i(\alpha)$ is only computed once per keypoint). The keypoint uncertainty bound is given by

Table 4.1: Coverage Percentages for 2-Norm Pose Uncertainty Set

		$\alpha = 0.1$					$\alpha = 0.4$				
	N	keypts		($\mathcal{P}_2$)		ellipse	keypts		($\mathcal{P}_2$)		ellipse
Calibration		S	R	S	R	R	S	R	S	R	R
ape	9	46.1	90.5	2.2	62.5	87.3	22.8	59.2	0.1	10.3	36.1
can	8	67.4	90.6	10.5	59.2	91.0	24.7	59.6	0	6.4	37.3
cat	10	67.2	91.0	8.9	57.9	83.4	36.7	61.4	0.1	5.0	30.3
duck	9	36.5	90.7	1.5	71.1	88.5	15.2	60.3	0	15.3	46.5
driller	11	71.3	89.9	16.2	58.7	85.6	27.9	58.9	0.2	6.8	32.6
eggbox	9	22.3	90.6	1.4	58.4	96.3	9.7	61.6	0.2	22.7	66.8
glue	10	47.3	88.8	1.6	56.8	85.6	15.8	56.0	0.6	7.2	32.8
holepuncher	10	50.1	87.8	1.1	41.0	79.6	21.2	57.9	0.1	3.8	19.4
average		51.0	90.0	5.7	58.0	87.2	21.8	59.4	0.1	9.6	37.6

$r_i(\alpha) \triangleq \tilde{r}_i(\alpha)/c_i$. Under Theorem 2.2.1, the high-probability error bound (4.2) holds when calibration data is *exchangeable* with evaluation data.

As a heuristic for evaluating exchangeability, we compute the coverage of the ground truth keypoints in each keypoint uncertainty set. We also compute coverage of the ground truth pose in both the propagated pose uncertainty set ($\mathcal{P}_2$) and the bounding ellipsoid (4.15) (more details in Section 4.4.3). The former roughly quantifies the probability β in Proposition 4.1.1. Coverage is defined as the percentage of keypoints (or poses) with uncertainty sets that contain the ground truth keypoint (or pose). Due to the inaccuracy of hand-labeled ground truth poses, this is not a perfect measure of exchangeability.

The results are given in Table 4.1 for $p = 2$. We compare coverage for calibration on freshly generated synthetic data (S) from physically-based rendering [72] and real calibration (R) at confidence 0.1 and 0.4. For each object we report the mean over all frames. It is immediately clear that the synthetic data is not exchangeable with the real LM-O images. While the real calibration data achieves near-perfect keypoint coverage, coverage of synthetic data varies widely and is far below 90% or 60%. We caution that LM-O ground truth poses are known to be imperfect [3]. Small errors in pose are magnified when propagated to keypoints.

More interestingly, Table 4.1 shows the pose uncertainty set suffers a $30 - 45\%$ drop in coverage compared to keypoints. There are $8 - 11$ keypoints per object, rendering the worst-case bound $1 - N\alpha$ (Proposition 4.1.1) useless or nearly useless. The pose coverage is slightly better than the independence case, suggesting our bounds benefit from some correlation between the keypoints. We provide keypoint and pose uncertainty set calibration results for $p = \infty$ in Appendix B.2.

Lastly, we note that the ellipse uncertainty set is not overly conservative. For $\alpha = 0.1$, it achieves just under 90% coverage. The ellipse is a little tighter for $\alpha = 0.4$, but in both cases coverage does not vary too much across objects. This suggests the ellipse is a useful measure of uncertainty. We provide quantitative and qualitative uncertainty bound results in Section 4.4.3.

Table 4.2: Percentage of 2D Projection Errors Under 5 Pixels (LM-O Dataset)

					$\alpha = 0.1$			$\alpha = 0.4$		
	Tekin [84]	PoseCNN [73]	Oberweger [85]	PVNet [86]	RANSAG [3]	OURS$_2$	OURS$_\infty$	RANSAG [3]	OURS$_2$	OURS$_\infty$
ape	7.01	34.6	69.6	69.1	77.7	77.6	77.0	79.5	76.6	76.4
can	11.2	15.1	82.6	86.1	73.4	19.2	82.2	75.4	69.9	81.3
cat	3.62	10.4	65.1	65.1	87.4	77.8	71.1	90.6	77.8	73.7
duck	5.07	31.8	61.4	61.4	82.7	80.7	79.0	83.1	80.5	76.9
driller	1.40	7.4	73.8	73.1	79.3	65.3	64.6	82.5	66.2	64.3
eggbox		1.9	13.1	8.43	0	0	4.8	0	0.1	4.5
glue	4.70	13.8	54.9	55.4	56.5	1.5	55.9	71.1	8.0	55.7
holepuncher	8.26	23.1	66.4	69.8	81.7	7.0	76.0	82.9	48.5	74.0
average	6.16	17.2	60.9	61.1	67.3	41.1	63.8	70.7	53.5	63.4

4.4.2 Central Pose Results

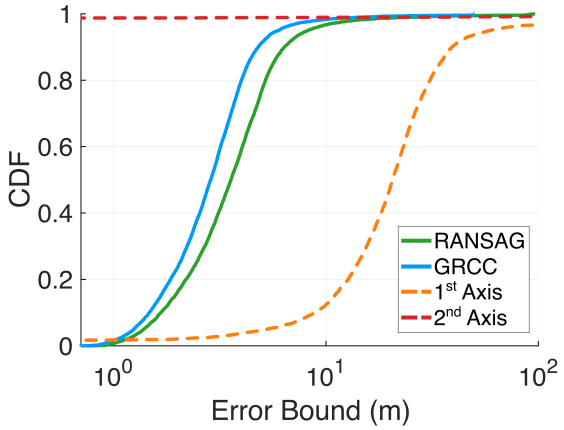
We evaluate the accuracy of our central pose estimate (Section 4.2) against four learning-based pose estimators and the averaged pose estimate from RANSAG [3]. Recall that none of these estimates come with statistical guarantees; at best, our estimate and the RANSAG pose estimate inherit guarantees from their respective uncertainty bounds. For each method we report the *five-pixel 2D projection error* metric [81]. The 2D projection error first projects the vertices of the relevant 3D model into pixel space, transforming according to the ground truth pose or estimated pose. The 2D projection error for each frame is the average ℓ_2 distance (in pixel space) between these projected vertices. The five-pixel metric is the percentage of images for which 2D projection error is less than 5 pixels.

The results are shown in Table 4.2. Our approach with $p = \infty$ outperforms the learning-based methods on average, but falls of short of conclusively outperforming RANSAG [3] except for a few objects[1]. This reflects the poor worst-case performance of our central pose algorithms. When the local solver fails to solve, we return a bad estimate of the pose of the object. In contrast, the RANSAG algorithm [3] uses the minimal solver P3P [87] to sample from the pose uncertainty set and simply averages these samples, even if it fails to find any points with the pose uncertainty set. Despite the simplicity of our approach, it is not significantly worse than RANSAG [3].

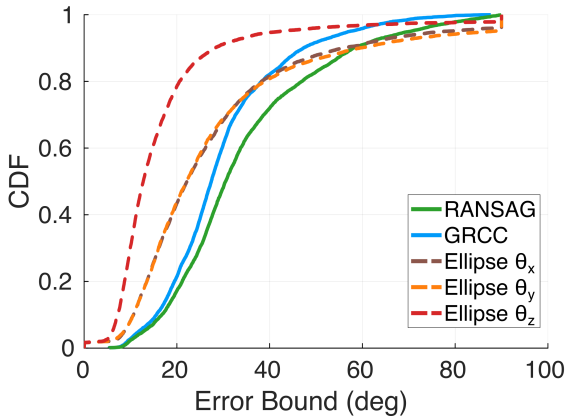
Table 4.2 also highlights a discrepancy between the 2-norm approach and ∞-norm approach for finding the central pose. The ∞-norm has consistently better 2D projection error, and the 2-norm can give very poor results when the uncertainty sets are large ($\alpha = 0.1$). This is likely the result of the quartic constraints on the central pose problem with $p = 2$, which require a second-order relaxation and are more difficult for the local solver to refine. Additionally, the ∞-norm results are similar at each confidence level. This suggests our multiplicative approximation of keypoint uncertainty sets at different confidences (Section 4.2) is reasonably accurate for $p = \infty$.

We note that our central pose approach is real time, achieving about 100 ms runtimes on

[1]We note that the eggbox object was observed to have particularly bad keypoint detections and ground truth annotations, leading to poor performance [3].



(a) CDF of translation bounds on LM-O. We show the half-length of the first and second principal axes of our marginalized ellipse.



(b) CDF of rotation bounds on LM-O. For our approach, we give angular bounds about the x, y, and z axes.

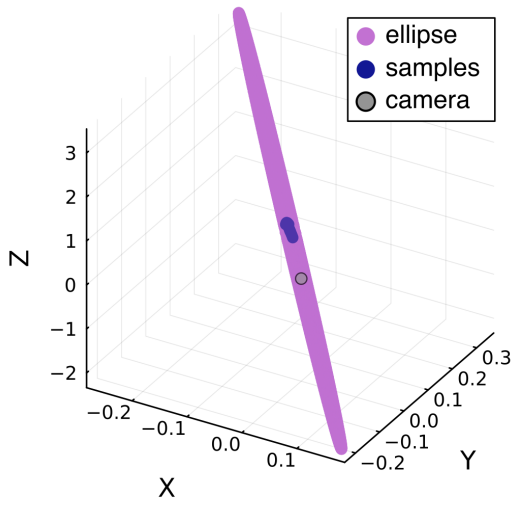
Figure 4.3: **Tightness of angular and translational bounds.** We plot the cumulative distribution function (CDF) of each approach over all objects excluding the eggbox. Our translation bounds are looser than baselines along one axis but significantly tighter along the others. Our rotation bounds are tighter and more interpretable, since they break out the three principal angles. RANSAG and GRCC results are from [25].

average. We provide more runtime details in Appendix B.2. We also provide additional 2D projection error results, including calibration on synthetic data, central pose with only the local solver, and a maximum margin formulation which is fast but biased towards estimates further from the camera. The central pose approach proposed in Section 4.2 with initialization via convex relaxation attains the best 2D projection error and outperforms the local solver in runtime.

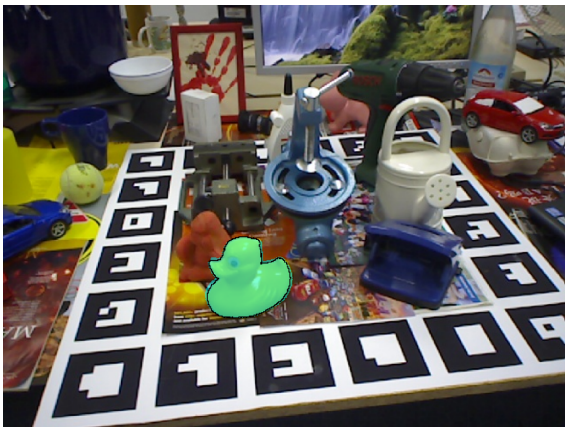
4.4.3 Uncertainty Bound Results

We use the central pose estimates ($p = \infty$) from the previous section as center for the bounding ellipse procedure proposed in Section 4.3. Unlike the central pose, the bounding ellipse is guaranteed to outer bound the conformal uncertainty set ($\mathcal{P}_2$) at a constant confidence. Here we show results for $\alpha = 0.1$ (more conservative) and leave results for $\alpha = 0.4$ to Appendix B.2. We compute translation and rotation bounds for every feasible pose estimate excluding the eggbox object (7586 frames in total).

Fig. 4.3 shows our quantitative results. We plot a cumulative distribution function (CDF) of the translation and rotation bounds, comparing with results from [25]. The translation bounds (Fig. 4.3a) are a much looser spherical approximation of uncertainty but are much tighter along the second and third principal axes (see Appendix B.2 for a CDF which spans the entire domain of the second axis). This general behavior is expected in the absence of depth information; however, our bounds along optical axis are overly conservative. Fig. 4.4a shows a translation ellipse for a particular frame; the ellipse is very tight except for the optical axis, where it expands to include some translation behind the the camera.



(a) A projection of the uncertainty ellipse into translation space.



(b) A sample pose estimate projected onto an image with rotational uncertainty at constant translation highlighted in green.

Figure 4.4: **Qualitative bounding sets in rotation and translation space.** The projected translation ellipse (a) is very tight along axes perpendicular to the optical axis but very loose along optical axis, expanding to include the origin and non-physical translation behind the camera. The rotations (b) show the relative tightness of the angular bounds (between 5 and 7 degrees for this frame). Both plots are for the duck object at frame 352 of the LM-O dataset.

In contrast, our angular bounds (Figure 4.3b) are clearly tighter than prior work. In the 5 to 30 degree uncertainty range all three axes are tighter than RANSAG [3] and GRCC [25]. This is the critical region where angular uncertainty bounds can be useful. Our angular bounds are also more specific, giving bounds for rotations about each axis. The tighter z-axis bounds reflect additional certainty for orientation about the optical axis. Projected onto the image plane, a typical rotation uncertainty set is quite tight. We give an example of this in Fig. 4.4b and provide more examples in Appendix B.2.

Note that, in contrast to the baselines, we have some angular uncertainty bounds near zero. This is not a breakdown of tightness; rather, it reflects the conditional coverage property of conformal prediction. Some keypoint uncertainty sets are very small and admit only a few set of possible rotations, even if this does not cover the ground truth pose. The angular problems incorporate a semidefinite relaxation which was consistently tight and fast. We provide a runtime breakdown in the next section, and a CDF of suboptimality gaps in Appendix B.2.

Runtime Breakdown

Lastly, we provide a runtime breakdown of our approach to generate uncertainty bounds in Table 4.3. The main bottleneck is finding a central pose; both the bounding ellipse and rotation marginalization problems are quite fast in practice. Table 4.3 shows the average

Table 4.3: Breakdown of Mean Runtimes for Pose Estimation and Uncertainty

	$\alpha = 0.1$ (ms)	$\alpha = 0.4$ (ms)
Keypoint Detection	48.8	48.8
Central Pose (∞)	116.4	90.7
Bounding Ellipse	33.6	49.4
Rotation Marginal	12.1	12.8
Total	210.9	201.7

runtimes over the same 7586 feasible frames used in the previous section. We omit the translation marginalization since it is sub-millisecond. With the exception of the keypoint detector, all algorithms are implemented in Julia and run on a single CPU thread with a clock speed clock speed of 4.2 GHz. The keypoint detector is implemented in PyTorch and runs on an M3 Mac.

These runtimes are significantly faster than prior work. In particular, the pose estimate from RANSAG [3] is reported to take less than a second. The authors do not report the runtime for finding a bounding ellipse using GRCC [25], but the third order semidefinite relaxation suggests a runtime of greater than one second per frame. Our method is much faster, finding a bounding ellipse and reducing it to rotation and translation bounds almost as fast as keypoint detection itself.

4.5 Discussion

This chapter presented a set of algorithms to convert keypoint uncertainty bounds from conformal prediction into a pose estimate with high-probability uncertainty bounds. Our pose estimation approach is a relaxation of the mode of the pose uncertainty sets and uses a semidefinite relaxation to initialize a local solver. To obtain uncertainty bounds, we first compute an outer bounding ellipse at fixed confidence and marginalize this to translation bounds via projection and rotation bounds via a tight semidefinite relaxation. In this section we discuss the limitations of our approach and give directions for future work.

We begin with a comment on the statistical motivation for this work. The relaxation from a Gaussian noise assumption to bounded noise is primarily a useful way to obtain high-probability pose uncertainty sets of the form ($\mathcal{P}_2$). It is less of a relaxation for pose estimation because bounded noise is *subGaussian* [88], meaning tail bounds can be given using a Gaussian upper bound. For a single confidence bounded noise is both less precise and less robust than Gaussian noise, completely removing the tails of the distribution and not capturing concentration phenomena. Noise bounds obtained via conformal prediction do little to counter the effect of outliers or bad pose estimates.

In light of this, it is not surprising that the pose estimation stage is the weakest part of our approach. Our formulation is relatively brittle in the presence of outliers and is highly dependent on the accuracy of keypoints. This is most clear in the difference between the $\alpha = 0.1$ and $\alpha = 0.4$ results in Table 4.2; the method struggles with larger uncertainty sets ($\alpha = 0.1$) even though it should be, to first order, invariant to changes in scale of the sets.

For better pose estimates, future work should include explicit consideration of outliers (or at least some tail probabilities). It may also be possible to derive an estimator that maximizes the subGaussian distribution bounds instead of the union bound probability.

In contrast, our approach for propagating uncertainty from keypoints to poses is much more effective. The bounding ellipse problem is fast and marginalizing to rotation bounds is empirically tight. The propagated bounds also maintain the high-probability keypoint uncertainty bounds from conformal prediction, although they may not be as tight; future work may quantify the relative volume of the pose uncertainty set and the reduced ellipse. In particular, the translation bounds are very loose along the optical axis. More work is needed to obtain a tight scale estimate in the absence of depth information.

It is important to mention that our pose bounds hold with some unknown probability less than $1 - \alpha$, where α is the keypoint conformal confidence. Empirically they satisfy the independence bounds and not just the worst-case union bounds, but this is an additional assumption. An alternative approach would be to conformalize some pose estimator directly using a conformal score. This would immediately yield high-probability bounds, and be much more independent of the pose estimate. However, we caution that conformalizing directly is not a panacea, and propagating uncertainty sets as we do in Section 4.3 can be useful. In some problem instances there may only be ground truth for some measurement and not the quantity of interest.

Chapter 5

Sub-Millisecond Solutions to Category-Level Shape and Pose Estimation

The certifiably optimal algorithms presented in the previous sections are fast enough to run near real-time, but they cannot be run at extremely high rates, especially on compute-limited hardware. In some cases it is more beneficial to have a fast local solver. This is the primary focus of this chapter. We explore a fast local solver for the single-frame shape and pose estimation problem under category-level priors. We begin with the problem formulation in Section 5.1, which is a single-frame version of the problem considered in Chapter 3. In Section 5.2 we reformulate the problem into a quartic objective using the quaternion representation of rotations. This admits first order conditions which are a nonlinear eigenvalue problem with eigenvector nonlinearities. We solve this eigenproblem efficiently using self-consistent field iteration [89], which requires only finding the eigenvector corresponding to the minimum eigenvalue at each step. Section 5.3 gives the self-consistent field iteration algorithm. In Section 5.4 we show this approach has a significant speed advantage compared to other local solvers including Gauss-Newton [90] and Manopt [91] in synthetically generated problems, achieving an order of magnitude speedup. We conclude with a discussion of theoretical gaps and extensions to similar problems.

5.1 Category-Level Shape and Pose Estimation Problem

Given detections of 3D keypoints on an object of known category, the *category-level shape and pose estimation* problem is to estimate the shape and pose (position and orientation) of the object. This section describes the problem formulation, including our choice of shape representation and measurement model. The problem formulation is identical to the problem considered in [2], although our solution strategy is substantially different.

5.1.1 Active Shape Model

We use the same active shape model introduced in Section 3.1, summarized here for clarity. For each category, we assume a library of K 3D shapes that span the category in the following sense. For each point $\mathbf{x}_i$ on an arbitrary category object, $\mathbf{x}_i$ may be expressed as a linear combination of corresponding points $\mathbf{b}_k$ on the objects in the 3D shape library. Mathematically:

$$\mathbf{x}_i = \sum_{k=1}^{K} c_k \mathbf{b}_{ik} \triangleq \mathbf{B}_i \mathbf{c} \tag{5.1}$$

where $\mathbf{B}_i$ stacks each $\mathbf{b}_{ik}$ as rows and $\mathbf{c}$ defines a linear combination: $c_k \in [0, 1]$ and $\sum_{k=1}^{K} c_k = 1$. It is useful to think of these points as semantically related. For example, within the *bottle* category, a point on each shape could be the position of the center of its bottlecap. Refer to Fig. 3.1b for a visualization of this model.

5.1.2 Measurement Model

We consider the measurements of a sparse set of 3D keypoints $\mathbf{y}_i$ with known associations to set of points $\mathbf{B}_i$ in the shape library, $i = 1, ..., N$. These measurements may come from pixel detections by a learned keypoint detector combined with depth information, and are typically semantically meaningful.

Denoting the object's position $\mathbf{p}$ and orientation $\mathbf{R}$ with respect to some fixed reference frame (*i.e.,* the camera frame), the detected keypoints $\mathbf{y}_i$, $i = 1, ..., N$ obey the following generative model:

$$\mathbf{y}_i = \mathbf{R}\mathbf{B}_i\mathbf{c} + \mathbf{p} + \boldsymbol{\epsilon}_i \tag{5.2}$$

We assume the measurement noise $\boldsymbol{\epsilon}_i$ follows an isotropic Gaussian distribution with zero mean and known isotropic covariance: $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, w_i^{-1}\mathbf{I}_3)$. This assumption can be related to a truncated normal distribution for outlier rejection using graduated non-convexity [60] or another outlier-robust wrapper such as RANSAC [92].

We now state the *category-level shape and pose estimation* problem:

Problem 5.1.1. *Estimate the shape* $\mathbf{c}$ *and pose* $(\mathbf{R}, \mathbf{p})$ *of an object from* N *3D keypoint measurements with known category-level associations.*

5.2 Nonlinear Eigenproblem for Local Minima

In this section we write Problem 5.1.1 as a maximum a posteriori (MAP) optimization problem. We show the optimization can be reduced to unconstrained optimization on the SO(3) manifold via convex marginalization. We conclude by expressing the first order optimality conditions of the rotation estimation problem as a nonlinear eigenproblem with eigenvector nonlinearities. This reformulation is novel compared to [2], and requires a slightly different marginalization.

Under model (5.2) the measurements $\mathbf{y}_i$ are generated according to the following likelihood:

$$\mathbb{P}(\mathbf{y}_i|\mathbf{R},\mathbf{c},\mathbf{p}) \propto \exp\left(-\frac{w_i}{2}\|\mathbf{y}_i - \mathbf{R}\mathbf{B}_i\mathbf{c} - \mathbf{p}\|^2\right) \tag{5.3}$$

Let us also introduce a prior on the shape coefficient $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I}_K)$, which will regularize the problem when there are more shapes than measurements:

$$\mathbb{P}(\mathbf{c}) \propto \exp\left(-\frac{\lambda}{2}\|c\|^2\right) \tag{5.4}$$

In this setting we arrive at the following estimator.

Proposition 5.2.1 (MAP Shape and Pose Estimator)**.** *The maximum a posteriori estimator that solves Problem 5.1.1 is given by:*

$$\begin{aligned} \min_{\substack{\mathbf{R}\in\mathrm{SO}(3)\\ \mathbf{p}\in\mathbb{R}^3,\,\mathbf{c}\in\mathbb{R}^K}} \; & \sum_{i=1}^{N} w_i\|\mathbf{y}_i - \mathbf{R}\mathbf{B}_i\mathbf{c} - \mathbf{p}\|^2 + \lambda\|\mathbf{c}\|^2 \\ s.t. \quad & \mathbf{1}^\mathsf{T}\mathbf{c} = 1,\ \mathbf{c} \in [0,1]^K \end{aligned} \tag{5.5}$$

Proof. The objective minimizes the negative log likelihood, given by:

$$\ell(\mathbf{R},\mathbf{c},\mathbf{p}|\mathbf{y}_i) = -\log\left(\mathbb{P}(\mathbf{y}_i|\mathbf{R},\mathbf{c},\mathbf{p})\mathbb{P}(\mathbf{c})\right) \tag{5.6}$$

Where we used Bayes rule and removed the constant in the denominator. The expressions for the conditional and prior probabilities are given by (5.3) and (5.4) respectively. □

In the following we will relax the problem by dropping the constraint $\mathbf{c} \in [0,1]^K$.

5.2.1 Reduction to Rotation Estimation Problem

We now perform convex analytic elimination to solve for position $\mathbf{p}^\star$ and shape $\mathbf{c}^\star$ in (5.5) as closed form functions of the optimal rotation. Unlike [2], we avoid vectorizing the rotation $\mathbf{R}$.

Holding $\mathbf{R}$ and $\mathbf{c}$ constant, (5.5) is an unconstrained convex quadratic problem in $\mathbf{p}$. Thus, first order conditions give the optimal position as a function of $\mathbf{R}$ and $\mathbf{c}$.

Proposition 5.2.2 (Optimal Position)**.** *The optimal position solving (5.5) is given by:*

$$\mathbf{p}^\star(\mathbf{R},\mathbf{c}) = \bar{\mathbf{y}} - \mathbf{R}\bar{\mathbf{B}}\mathbf{c} \tag{5.7}$$

where $\bar{\mathbf{y}}$ and $\bar{\mathbf{B}}$ are weighted averages of $\mathbf{y}_i$ and $\mathbf{B}_i$ as below:

$$\bar{\mathbf{y}} \triangleq \frac{\sum_{i=1}^N w_i\mathbf{y}_i}{\sum_{i=1}^N w_i} \quad \textit{and} \quad \bar{\mathbf{B}} \triangleq \frac{\sum_{i=1}^N w_i\mathbf{B}_i}{\sum_{i=1}^N w_i} \tag{5.8}$$

Proof. The first order optimality conditions are necessary and sufficient. In the unconstrained case the first order conditions are $\mathbf{0} = \nabla_{\mathbf{p}}(\sum_{i=1}^{N} w_i \|\mathbf{y}_i - \mathbf{R}\mathbf{B}_i\mathbf{c} - \mathbf{p}\|^2)$. Simplifying,

$$\sum_{i=1}^{N} w_i \mathbf{p}^\star = \sum_{i=1}^{N} w_i (\mathbf{y}_i - \mathbf{R}\mathbf{B}_i\mathbf{c}) \tag{5.9}$$

The result follows from solving for $\mathbf{p}^\star$. □

We now plug in the optimal value for $\mathbf{p}$ as given by (5.7). Letting $\bar{\mathbf{y}}_i \triangleq \mathbf{y}_i - \bar{\mathbf{y}}$ and $\bar{\mathbf{B}}_i \triangleq \mathbf{B}_i - \bar{\mathbf{B}}$, eq. (5.5) simplifies to:

$$\min_{\substack{\mathbf{R}\in\mathrm{SO}(3) \\ \mathbf{c}\in\mathbb{R}^K \, \mathbf{1}^\mathsf{T}\mathbf{c}=1}} \sum_{i=1}^{N} \|\bar{\mathbf{y}}_i - \mathbf{R}\bar{\mathbf{B}}_i\mathbf{c}\|^2 + \lambda\|\mathbf{c}\|^2 \tag{5.10}$$

We now solve for the optimal shape vector as a function of the rotation $\mathbf{c}^\star(\mathbf{R})$. Eq. (5.10) is convex in $\mathbf{c}$ because the objective is quadratic and the constraints are linear equalities. Thus, the KKT conditions are necessary and sufficient for the optimal shape vector. The result is summarized in Proposition 5.2.3 below.

Proposition 5.2.3 (Optimal Shape)**.** *The shape vector that optimally solves* (5.10) *is:*

$$\mathbf{c}^\star(\mathbf{R}) = \mathbf{C}_1 \sum_{i=1}^{N} \left(\bar{\mathbf{B}}_i^\mathsf{T}\mathbf{R}^\mathsf{T}\bar{\mathbf{y}}_i\right) + \mathbf{c}_2 \tag{5.11}$$

where we use the following symbols:

$$\begin{array}{lll} \hat{\mathbf{B}}^2 & \triangleq \sum_{i=1}^{N} \bar{\mathbf{B}}_i^\mathsf{T}\bar{\mathbf{B}}_i & \in \mathbb{R}^{K\times K} \\ \mathbf{H} & \triangleq \hat{\mathbf{B}}^2 + \lambda\mathbf{I}_K & \in \mathbb{R}^{K\times K} \\ \mathbf{C}_1 & \triangleq \mathbf{H}^{-1} - \mathbf{H}^{-1}\mathbf{1}_K \left(\mathbf{1}_K^\mathsf{T}\mathbf{H}^{-1}\mathbf{1}_K\right)^{-1} \mathbf{1}_K^\mathsf{T}\mathbf{H}^{-1} & \in \mathbb{R}^{K\times K} \\ \mathbf{c}_2 & \triangleq \mathbf{H}^{-1}\mathbf{1}_K \left(\mathbf{1}_K^\mathsf{T}\mathbf{H}^{-1}\mathbf{1}_K\right)^{-1} & \in \mathbb{R}^{K} \end{array} \tag{5.12}$$

We note that $\hat{\mathbf{B}}^2$ is invertible as long as there are $N \geq 3$ non-colinear keypoints and $N > K$. The λ term regularizes the problem to ensure invertibility when the latter condition is violated, i.e., *when $N \leq K$.*

The proof of Proposition 5.2.3 is algebraically involved and postponed to Appendix C.1. The key idea is to use the KKT conditions for (5.10) to write a linear system for $\mathbf{c}$ where only a known constant matrix needs to be inverted.

We now state the rotation estimation problem which, together with the shape (5.11) and position (5.7) formula, solves Problem 5.1.1.

$$\min_{\mathbf{R}\in\mathrm{SO}(3)} \sum_{i=1}^{N} \left\| \bar{\mathbf{y}}_i - \mathbf{R}\bar{\mathbf{B}}_i \left(\mathbf{C}_1 \sum_{j=1}^{N} \left(\bar{\mathbf{B}}_j^\mathsf{T}\mathbf{R}^\mathsf{T}\bar{\mathbf{y}}_j\right) + \mathbf{c}_2 \right) \right\|^2 + \lambda \left\| \mathbf{C}_1 \sum_{j=1}^{N} \left(\bar{\mathbf{B}}_j^\mathsf{T}\mathbf{R}^\mathsf{T}\bar{\mathbf{y}}_j\right) + \mathbf{c}_2 \right\|^2 \tag{5.13}$$

Note that (5.13) is an optimization problem over only a single rotation with no constraints beyond the SO(3) manifold. Further, it is *quadratic* in the unknown matrix $\mathbf{R}$. In Section 5.3 we present a method which finds first order stationary points of (5.13) that relies on both of these properties.

5.2.2 First Order Conditions in Terms of Quaternions

In contrast to [2] we refrain from writing (5.13) as a quadratic program in vec($\mathbf{R}$), where vec vectorizing a matrix by stacking its columns. Instead, we rewrite the problem as a quartic objective with a quadratic equality constraint using quaternions. As we will see, this quaternion formulation leads to first order conditions which are a nonlinear eigenproblem. Throughout this section we use the quaternion algebra reviewed in Section 2.3.

Begin by expanding the objective of (5.13), grouping terms by their dependency on $\mathbf{R}$.

$$\begin{aligned}(5.13) = \min_{\mathbf{R}\in\mathrm{SO}(3)} &\left\{\sum_{i=1}^{N}(\bar{\mathbf{y}}_i^\mathsf{T}\bar{\mathbf{y}}_i) + \mathbf{c}_2^\mathsf{T}\hat{\mathbf{B}}^2\mathbf{c}_2 + \lambda\mathbf{c}_2^\mathsf{T}\mathbf{c}_2\right\} \\ &+ \left\{\sum_{i=1}^{N} 2\bar{\mathbf{y}}_i^\mathsf{T}\mathbf{R}\bar{\mathbf{B}}_i\left(-\mathbf{I}_3 + \mathbf{C}_1\hat{\mathbf{B}}^2 + \lambda\mathbf{C}_1\right)\mathbf{c}_2\right\} \\ &+ \left\{\sum_{i=1}^{N}\bar{\mathbf{y}}_i^\mathsf{T}\mathbf{R}\bar{\mathbf{B}}_i\mathbf{C}_1\left(-2\mathbf{I}_3 + \hat{\mathbf{B}}\mathbf{C}_1 + \lambda\mathbf{C}_1\right)\sum_{j=1}^{N}\bar{\mathbf{B}}_j^\mathsf{T}\mathbf{R}^\mathsf{T}\bar{\mathbf{y}}_j\right\}\end{aligned} \tag{5.14}$$

Now, drop the constant terms and rewrite the objective in terms of the unit quaternion $\mathbf{q}$ which denotes the rotation represented by $\mathbf{R}$. We use the identity (2.25) and drop the tilde notation for homogeneous vectors; all vectors in $\mathbb{R}^3$ are assumed to be homogenized with 0 as the scalar part; see Section 2.3.

$$\begin{aligned}(5.13) = \min_{\mathbf{q}\in\mathbb{S}^3} &\left\{ 2\mathbf{q}^\mathsf{T}\underbrace{\sum_{i=1}^{N}\mathbf{\Omega}_2(\bar{\mathbf{y}}_i)\mathbf{\Omega}_1\left[\bar{\mathbf{B}}_i\left(\mathbf{I}_3 - \mathbf{C}_1\hat{\mathbf{B}}^2 - \lambda\mathbf{C}_1\right)\mathbf{c}_2\right]}_{\triangleq\mathbf{D}}\mathbf{q}\right\} \\ &+ \left\{\mathbf{q}^\mathsf{T}\underbrace{\sum_{i=1}^{N}\mathbf{\Omega}_2(\bar{\mathbf{y}}_i)\mathbf{\Omega}_1\left[\bar{\mathbf{B}}_i\left(2\mathbf{I}_3 - \mathbf{C}_1\hat{\mathbf{B}}^2 - \lambda\mathbf{C}_1\right)\mathbf{C}_1\sum_{j=1}^{N}\bar{\mathbf{B}}_j^\mathsf{T}\mathbf{R}^\mathsf{T}(\mathbf{q})\bar{\mathbf{y}}_j\right]}_{\triangleq\mathbf{A}(\mathbf{q})}\mathbf{q}\right\}\end{aligned} \tag{5.15}$$

More compactly,

$$(5.13) = \min_{q\in\mathbb{S}^3}\mathbf{q}^\mathsf{T}\mathbf{A}(\mathbf{q})\mathbf{q} + 2\mathbf{q}^\mathsf{T}\mathbf{D}\mathbf{q} \tag{5.16}$$

We observe the following properties of (5.16). First, the objective contains a quadratic and a quartic term. The constant matrix of the quadratic term $\mathbf{D}$ is symmetric by the inverse property given in Lemma 2.3.1. The quartic term is also symmetric in the sense that the matrix to quaternion identity (2.25) gives the same result when applied to either rotation. Both $\mathbf{D}$ and $\mathbf{A}(\mathbf{q})$ have zero trace.

Under the unit norm constraint eq. (5.16) trivially satisfies the linear independence constraint qualification. Thus, the set of stationary points are given by the first order conditions. We use the product rule for objective terms and $\mu\in\mathbb{R}$ as a dual variable for the unit norm constraint. Stationary points of (5.16) occur when:

$$\mathbf{0} = 4\mathbf{A}(\mathbf{q})\mathbf{q} + 4\mathbf{D}\mathbf{q} - \mu\mathbf{q} \tag{5.17}$$

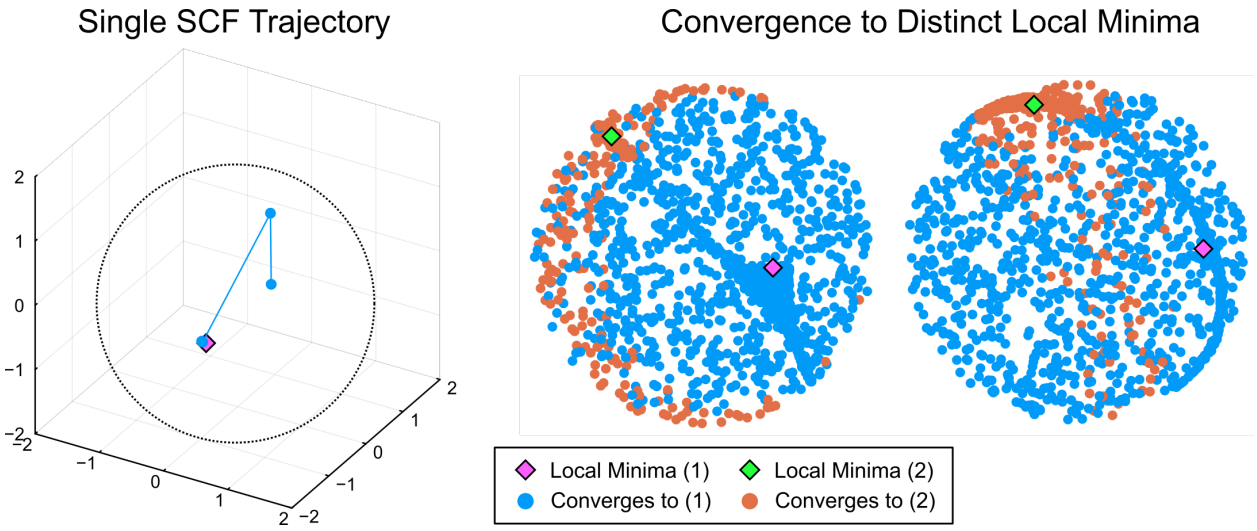


Figure 5.1: **Stereographic projections of self-consistent field iterates.** Beginning from a unit quaternion $\mathbf{q}_0 \in \mathbb{S}^3$, SCF rapidly converges to a local stationary point. Left, a single SCF trajectory. Right, two views of unit quaternions stereographically projected into the volume of the 3-dimensional unit ball (see Appendix C.2) and colored by the local minimum SCF converges to. Nearby starting points tend to converge to the same local minimum except at the distinct boundary. Plots show synthetic data with high measurement noise ($\sigma_m = 5$).

Eq. (5.17) resembles an eigenvalue problem, with one summand $\mathbf{A}(\mathbf{q})$ having eigenvector dependence. This result is summarized in the following proposition.

Proposition 5.2.4 (Eigenproblem for Local Solutions)**.** *All local minima* $\mathbf{q}$ *of* (5.13) *satisfy the following nonlinear eigenproblem for some* $\mu \in \mathbb{R}$*:*

$$(\mathbf{A}(\mathbf{q}) + \mathbf{D})\,\mathbf{q} = \mu\mathbf{q} \tag{5.18}$$

Several remarks are in order. While it is not immediately clear how to solve (5.18), Section 5.3 will develop a fast iterative solver that only requires computing $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^4$ and its smallest eigenvalue-eigenvector pair at each iteration. When $\lambda = 0$, computing the matrices $\mathbf{A}(\mathbf{q})$ and $\mathbf{D}$ requires even less computation. Apart from the obvious simplifications, $\lambda = 0$ implies $\mathbf{C}_1\hat{\mathbf{B}}^2\mathbf{c}_2 \equiv 0$. Lastly, in the interest of global solutions, observe the objective value at a given stationary point is a perturbation of μ by the constant matrix $\mathbf{D}$:

$$f_{\text{local}} = \mu + \mathbf{q}^{\mathsf{T}}\mathbf{D}\mathbf{q} \tag{5.19}$$

We discuss global optimality further in Section 5.5.

5.3 Iterative Method for Fast Local Solutions

Proposition 5.2.4 gives a compact necessary condition for local minima. In this section we propose a fast solution strategy for (5.18) using self-consistent field iteration [93–95].

5.3.1 Self-Consistent Field Iteration

Self-consistent field (SCF) iteration is a solution strategy for nonlinear eigenproblems such as (5.18). SCF starts from an initial guess and computes the corresponding data matrix $\mathbf{A}(\mathbf{q}) + \mathbf{D}$. Then, the estimated $\mathbf{q}$ is updated to an eigenvector of the data matrix. The algorithm terminates when it converges to a stationary point–a unit vector $\mathbf{q}$ which exactly satisfies (5.18). In practice, we terminate using the angle between the current and next iterate.

Data: $\mathbf{A}(\mathbf{q})$ and $\mathbf{D}$ from (5.15)
Result: $\mathbf{q}$ satisfying (5.18)
initialize $\mathbf{q}_0$
for $t \leftarrow 1$ **to** T **do**
  $\mathbf{q}_{t+1} \leftarrow \arg\min_{\mathbf{q}\in\mathbb{S}^3} \mathbf{q}^\mathsf{T}(\mathbf{A}(\mathbf{q}_t) + \mathbf{D})\mathbf{q}$
  `/* termination condition */`
  if $\sin \measuredangle(\mathbf{q}_t, \mathbf{q}_{t+1}) < \epsilon$ **then**
    $\mathbf{q} \leftarrow \mathbf{q}_t$
    break
  end
end

Algorithm 5.1: Self-consistent field iteration for local solutions to (5.16).

The full algorithm is given in Algorithm 5.1 and illustrated in Fig. 5.1. We update $\mathbf{q}$ according to the eigenvector corresponding to the *minimum* eigenvalue. Although we could pick any of the eigenvectors, picking the smallest has several desirable properties. First, it is likely to be a local minima (rather than a saddle point or local maxima) since the objective at stationary points is dominated by the eigenvalue, see (5.19). Second, it has strong computational benefits. In particular, the minimum eigenvalue is often the eigenvalue with largest magnitude at optimality (recall the matrices $\mathbf{A}(\mathbf{q})$ and $\mathbf{D}$ are not positive semidefinite and have zero trace). This property enables fast convergence, often less than 5 iterations. For faster convergence, the termination condition may be relaxed to a near-constant objective value. This is especially useful in high-noise situations when the objective landscape is very flat near optima.

The key advantage of our approach is its speed. A single iteration of SCF requires only computing a 4×4 matrix and its minimum eigenvector. The termination condition requires only checking the value of an inner product. In practice, these steps take less than 10 μs on a single CPU thread. Starting with different initial conditions is easily parallelized across GPU resources. In Section 5.4 we show the entire algorithm takes about 150 μs with random data.

5.4 Empirical Performance in Synthetic Dataset

We evaluate the accuracy and computational speed of our approach using synthetic data in the local and global settings. We generate synthetic data according to the measurement

model (5.2). First, we generate a mean shape with N points drawn from a standard normal and centered about their mean. The shape library $\mathbf{B}$ is formed by adding zero-mean Gaussian noise to each point with fixed uniform standard deviation $r = 0.2$. Then, we generate a ground truth shape vector $\mathbf{c}$ by normalizing a K-dimensional uniformly random vector. Ground truth position is drawn from a standard normal with mean 1, and ground truth rotation is drawn uniformly from SO(3). The measurements $\mathbf{y}_i$ follow model (5.2) with fixed isotropic variance $\sigma^2 = w_i^{-1}$. We adopt r as our length scale and normalize all length-dependent quantities (σ, positions) accordingly. We test with a small shape library ($N = 10$, $K = 4$, $\lambda = 0$) and provide some results with a larger shape library in Appendix C.5; performance is not significantly different. All benchmarks are run on a single CPU thread with a clock speed of 4.2 GHz.

5.4.1 Baselines

We compare against the local solvers Gauss–Newton (G-N), Levenburg–Marquardt (L-M), and Manopt [91]. Manopt is an off-the-shelf general-purpose solver for unconstrained problems on a manifold. We use it to directly solve (5.13). The G-N and L-M solvers are specialized to the shape and pose estimation problem (5.13) with analytical gradients for a fair comparison (derived in Appendix C.4). We also compare against SDP, which converts the QCQP (5.13) into a convex semidefinite program by taking the dual of the dual (reviewed in Section 2.1). This provides a lower bound on the optimal objective which is empirically tight for low-noise problems [2], and a zero duality gap certifies the global optimality of the solution.

We also compare against two ablations of self-consistent field iteration. The SCF-Obj approach is identical to Algorithm 5.1 except it terminates when the objective value is stationary. Computing the objective slows down each iteration but may also allow significantly fewer iterations. Lastly, the Power solver replaces the eigenvector step in SCF with a power method, a product-based strategy which converges to the dominant eigenvector [1] . At each iteration, Power updates the estimate by normalizing the product between the data matrix and the previous estimate, terminating when it reaches a stationary point. The full algorithm is given in Appendix C.3.

All methods are implemented in Julia and runtimes do not include precompilation time. In each experiment all methods are tested with the same set of 1000 synthetically-generated problems for each measurement noise value σ_m.

5.4.2 Local Solutions

Runtimes

Table 5.1 compares the mean and 90th percentile (p90) of runtimes for each method across noise scales. The vanilla version of self-consistent field iteration is the fastest method, averaging around 150 μs to find a solution across noise scales. This is more than an order of magnitude faster than the next-fastest local solver. Performance is similar to SCF-Obj and

[1] We gratefully acknowledge Aaron Ray for suggesting the power method as a solution strategy.

Table 5.1: Mean and 90th Percentile of Solver Runtimes

	$\sigma_m = 0.25$		$\sigma_m = 2.5$	
Method	Mean (ms)	p90 (ms)	Mean (ms)	p90 (ms)
SCF	**0.146**	**0.149**	**0.157**	**0.169**
Manopt	3.211	3.791	2.585	3.672
G-N	2.366	2.887	3.938	4.946
L-M	2.417	2.881	3.790	4.701
SDP	5.687	5.915	5.696	5.907
SCF-Obj	0.179	0.208	0.186	0.211
Power	0.401	0.541	0.383	0.536

more than twice as fast as Power. The sub-millisecond runtime of SCF across noise scales enables a real-time perception loop with a 1 kHz update or on compute-limited hardware.

Runtime is a function of both number of keypoints N and size of shape library K. We expect all methods to scale similarly in number of keypoints, but it is not clear how SCF scales in K. Empirically, we find runtime when K is large to be highly dependent on the choice of regularization λ. For completeness we provide an analog to Table 5.1 for $K > N$ in Appendix C.5.

Rotation Estimation Performance

Fig. 5.2 compares SCF against the certifiable global solver SDP and the ablation with objective termination SCF-Obj. All three methods exhibit similar performance across noise scales, with SDP achieving slightly lower rotation error at higher noise. The key advantage of SDP over SCF is the certificate of global optimality. While SCF does not always converge to the global minimizer (see *e.g.,* Fig. 5.1), it frequently finds a reasonable estimate. Similarly, terminating self-consistent field iteration using the objective (SCF-Obj) does not lead to significantly different rotation estimates.

For brevity, we omit a direct comparison of performance between the other local solvers. Apart from L-M they are all guaranteed to reach a first order stationary point. We provide the equivalent to Fig. 5.2 for a large shape library $K > N$ in Appendix C.5.

5.5 Discussion and Extensions

Section 5.4 gives empirical assurances regarding the behavior of self-consistent field iteration in solving Problem 5.1.1. However, there are still theoretical gaps regarding its local and global convergence. Cai et al. [93] consider a matrix-valued eigenvector-dependent eigenproblem and show each eigenvector has a region of attraction; that is, if SCF starts close enough it will converge to that eigenvector. This theory is presented under a set of assumptions that are difficult to check without prior knowledge of the stationary points. The authors also show SCF converges in finite time to the eigenvectors corresponding to the smallest eigenvalues under the milder condition of a non-zero eigenvalue gap [93]. Some of this theory can be

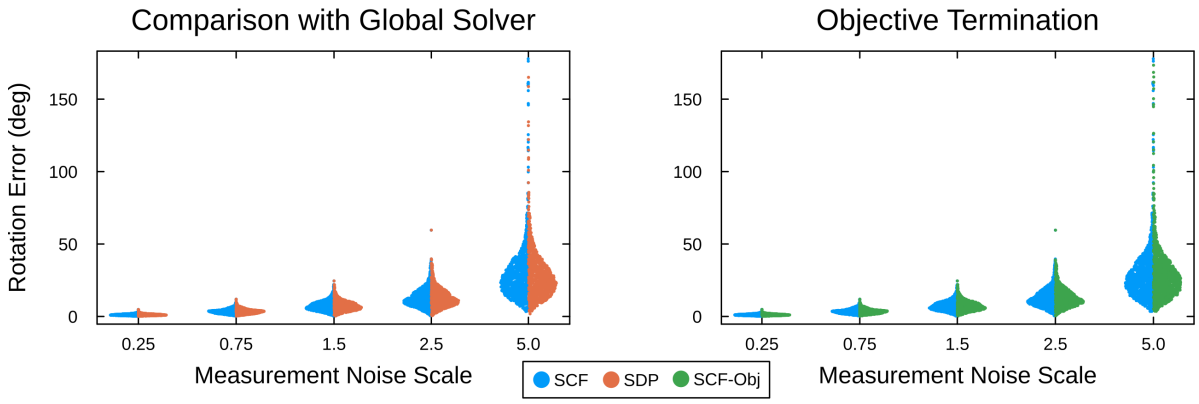


Figure 5.2: **Distribution of rotation errors for SCF, SDP, and SCF-Obj.** We plot the distributions of rotation error at selected noise scales expressed as multiples of the object radius. Left, SDP and SCF achieve similar performance across noise scales. Especially at higher noise scales, SDP performs slightly better on average. Right, SCF-Obj and SCF have near-identical performance, suggesting objective termination is an effective signal of reaching a local minimizer.

directly applied to Problem 5.1.1, but more work is needed to give compact conditions for local convergence.

Additionally, we note that reformulating a quadratic rotation-constrained problem into a nonlinear eigenproblem for local extrema is not unique to Problem 5.1.1. It may be applied to similar rotation-constrained problems such as the backprojection form of perspective-n-point [96].

Chapter 6

Conclusion

TODO

Appendix A

Additional Results and Proofs for CAST*

A.1 Maximum A Posteriori Derivation

Here we show that (3.7) is a maximum a posteriori (MAP) estimator. We first restate the problem:

$$\begin{aligned}\min_{\substack{\mathbf{R}_t,\boldsymbol{\Omega}_l\in\mathrm{SO}(3),\\ \mathbf{p}_t,\mathbf{v}_l\in\mathbb{R}^3,\\ \mathbf{c}\in\mathbb{R}^K,\mathbf{1}_K^\mathsf{T}\mathbf{c}=1\\ t=1,\dots,T,\\ l=1,\dots,T-1}} \quad & \sum_{t=1}^{T}\sum_{i=1}^{N} w_t^i \left\|\mathbf{y}_t^i - \mathbf{R}_t\mathbf{B}_i\mathbf{c} - \mathbf{p}_t\right\|^2 + \lambda\|\Delta\mathbf{c}\|^2 \\ & + \sum_{t=1}^{T-1} \omega_t\|\Delta\mathbf{v}_t\|^2 + \kappa_t\|\Delta\boldsymbol{\Omega}_t\|_F^2 \\ \text{s.t.} \quad & \mathbf{p}_{t+1} = \mathbf{f}(\mathbf{p}_t,\mathbf{R}_t,\mathbf{R}_{t+1},\mathbf{v}_t),\ \ \mathbf{R}_{t+1} = \mathbf{R}_t\boldsymbol{\Omega}_t \\ & \Delta\mathbf{v}_t = \mathbf{v}_{t+1} - \mathbf{v}_t,\ \ \Delta\boldsymbol{\Omega}_t = \boldsymbol{\Omega}_{t+1} - \boldsymbol{\Omega}_t\end{aligned} \tag{A.1}$$

where we introduced auxiliary variables $\Delta\mathbf{v}_t$ and $\Delta\boldsymbol{\Omega}_t$ for the velocity changes. We now show that the first summand in (A.1) corresponds to the likelihood of our keypoint measurements (3.6), while the other terms describe our priors on the shape, velocity, and rotation rates. Denote the quantities to estimate by $\mathbf{z} \triangleq [\mathbf{c}, \{\mathbf{p}_t,\mathbf{R}_t\}_{t=1}^T, \{\mathbf{v}_t,\boldsymbol{\Omega}_t\}_{t=1}^{T-1}, \{\Delta\mathbf{v}_t,\Delta\boldsymbol{\Omega}_t\}_{t=1}^{T-1}]$ belonging to the domain $\mathbb{Z}$ which includes all relevant constraints in (A.1). The MAP estimator takes the form:

$$\arg\max_{z\in\mathbb{Z}}\ \mathbb{P}(\mathbf{z}\,|\,\{\mathbf{y}_t^i\}_{i,t=1}^{N,T}) = \arg\max_{z\in\mathbb{Z}}\ \mathbb{P}(\{\mathbf{y}_t^i\}_{i,t=1}^{N,T}\,|\,\mathbf{z})\mathbb{P}(\mathbf{z}) \tag{A.2}$$

where we expanded using Bayes rule. Assuming independent measurements, shape independence, and Markovian time-independence, we can rewrite (A.2) as:

$$\arg\max_{z\in\mathbb{Z}} \prod_{i,t=1}^{N,T} \mathbb{P}(\mathbf{y}_t^i\,|\,\mathbf{z}) \prod_{t=1}^{T-1} \mathbb{P}(\Delta\mathbf{v}_t)\mathbb{P}(\Delta\boldsymbol{\Omega}_t)\mathbb{P}(\Delta\mathbf{c}) \tag{A.3}$$

For the posterior $\mathbb{P}(\mathbf{y}_t^i\,|\,\mathbf{z})$ we assume a zero-mean Gaussian with covariance $\boldsymbol{\Sigma}_t^i = \frac{1}{w_t^i}\mathbf{I}_3$. Hence, using (3.6):

$$\mathbb{P}(\mathbf{y}_t^i\,|\,\mathbf{z}) = \alpha_t^i \exp\left(-\frac{w_t^i}{2}\left\|\mathbf{y}_t^i - \mathbf{R}_t\mathbf{B}_i\mathbf{c} - \mathbf{p}_t\right\|^2\right) \tag{A.4}$$

with normalization constant α_t^i.

Similarly, for velocity and shape we assume a zero-mean Gaussian prior with covariance $\frac{1}{\omega_t}\mathbf{I}_3$ and $\frac{1}{\lambda}\mathbf{I}_3$ respectively:

$$\mathbb{P}(\Delta\mathbf{v}_t) = \alpha_t^v \exp\Big(-\frac{\omega_t}{2}\left\|\Delta\mathbf{v}_t\right\|^2\Big) \tag{A.5}$$

$$\mathbb{P}(\Delta\mathbf{c}) = \alpha_c \exp\left(-\frac{\lambda}{2}\left\|\Delta\mathbf{c}\right\|^2\right) \tag{A.6}$$

We also assume that the rotation rate follows a Langevin distribution with concentration parameter κ_t:

$$\mathbb{P}(\Delta\mathbf{\Omega}_t) = \alpha_t^o \exp\big(-\kappa_t \left\|\Delta\mathbf{\Omega}_t\right\|_F^2\big) \tag{A.7}$$

where α_t^v, α_c, and α_t^o are suitable normalization constants.

Replacing the maximum of the posterior with the minimum of the negative logarithm of the posterior and dropping multiplicative and additive constants, we arrive at the result. □

A.2 Proof of Proposition 3.2.1: Closed-Form Optimal Shape

Holding all other variables constant, (3.7) is a linearly constrained least squares problem in $\mathbf{c}$. Thus, the minimum with respect to $\mathbf{c}$ is convex and admits a unique solution via the KKT conditions. If we drop objective terms that do not depend on $\mathbf{c}$ in (3.7), we get:

$$\min_{\substack{\mathbf{c}\in\mathbb{R}^K,\\ \mathbf{1}_K^\mathsf{T}\mathbf{c}=1}} \quad \sum_{t=1}^{T}\sum_{i=1}^{N} w_t^i \left\|\mathbf{y}_t^i - \mathbf{R}_t\mathbf{B}_i\mathbf{c} - \mathbf{p}_t\right\|^2 + \lambda\|\Delta\mathbf{c}\|^2 \tag{A.8}$$

Expanding the summation over keypoint indices i and moving the weights into the norm:

$$\sum_{t=1}^{T}\left\|\underbrace{\begin{bmatrix}\sqrt{w_t^1}\mathbf{I}_3 & & \\ & \ddots & \\ & & \sqrt{w_t^N}\mathbf{I}_3\end{bmatrix}}_{\triangleq\mathbf{W}_t}\left(\underbrace{\begin{bmatrix}\mathbf{R}_t^\mathsf{T}(\mathbf{y}_t^1-\mathbf{p}_t)\\ \vdots \\ \mathbf{R}_t^\mathsf{T}(\mathbf{y}_t^N-\mathbf{p}_t)\end{bmatrix}}_{\triangleq\mathbf{h}_t} - \underbrace{\begin{bmatrix}\mathbf{B}_1\\ \vdots\\ \mathbf{B}_N\end{bmatrix}}_{\triangleq\mathbf{B}}\mathbf{c}\right)\right\|^2 + \lambda\|\Delta\mathbf{c}\|^2 \tag{A.9}$$

where we used the rotational invariance of the 2-norm to move $\mathbf{R}^\mathsf{T}$.

We'll use stationarity and primal feasibility. The Lagrangian is:

$$L(\mathbf{c},\mu) = \sum_{t=1}^{T}\|\mathbf{W}_t(\mathbf{h}_t - \mathbf{B}\mathbf{c})\|^2 + \lambda\|\Delta\mathbf{c}\|^2 + \mu(1-\mathbf{1}^\mathsf{T}\mathbf{c}) \tag{A.10}$$

The stationarity condition is:

$$0 = \nabla_\mathbf{c} L = 2\mathbf{B}^\mathsf{T}\left(\sum_{t=1}^{T}\mathbf{W}_t^2\right)\mathbf{B}\mathbf{c} - 2\mathbf{B}^\mathsf{T}\left(\sum_{t=1}^{T}\mathbf{W}_t^2\mathbf{h}_t\right) + 2\lambda\Delta\mathbf{c} + \mathbf{1}\mu \tag{A.11}$$

Putting this together with primal feasibility, we arrive at the following linear system:

$$\begin{bmatrix} \mathbf{H}^{-1} & \mathbf{1}_K \\ \mathbf{1}_K^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mu \end{bmatrix} = \begin{bmatrix} 2\left(\mathbf{B}^\mathsf{T} \sum_{t=1}^T \mathbf{W}_t^2 \mathbf{h}_t + \lambda \bar{\mathbf{c}}\right) \\ 1 \end{bmatrix} \tag{A.12}$$

where $\mathbf{H} \triangleq \frac{1}{2}\left(\mathbf{B}^\mathsf{T}\left(\sum_{t=1}^T \mathbf{W}_t^2\right)\mathbf{B} + \lambda \mathbf{I}_K\right)^{-1}$. The matrix on the left hand side can be inverted using the Schur complement rule.

Solving for $\mathbf{c}$ and substituting the definitions of $\mathbf{G}$ and $\mathbf{g}$, we arrive at the result. Crucially, notice that the matrix we must invert to get $\mathbf{H}$ is made up only of constants. Notice that $\mathbf{H}$ is invertible so long as $\lambda > 0$ or the number of shapes K is less than the number of keypoints N. □

A.3 Proof of Proposition 3.2.2: Quadratically Constrained Quadratic Program

We focus on the measurement terms, the constraints, and the variable $\mathbf{c}$. The remaining objective terms contain norms of single-degree variables and are thus quadratic. The key idea is to let $\mathbf{s}_t \triangleq \mathbf{R}_t^\mathsf{T}\mathbf{p}_t$. Then, the measurement term of the objective may be rotated without changing its norm:

$$\left\|\mathbf{y}_t^i - \mathbf{R}_t\mathbf{B}_i\mathbf{c} - \mathbf{p}_t\right\|^2 = \left\|\mathbf{R}_t^\mathsf{T}\mathbf{y}_t^i - \mathbf{B}_i\mathbf{c} - \mathbf{s}_t\right\|^2 \tag{A.13}$$

Similarly, the optimal solution $\mathbf{c}^*$ may be rewritten as:

$$\mathbf{c}^\star = 2\mathbf{G}\left(\mathbf{B}^\mathsf{T}\sum_{t=1}^T \mathbf{W}_t \begin{bmatrix} \mathbf{R}_t^\mathsf{T}\mathbf{y}_t^1 - \mathbf{s}_t \\ \vdots \\ \mathbf{R}_t^\mathsf{T}\mathbf{y}_t^N - \mathbf{s}_t \end{bmatrix} + \lambda\bar{\mathbf{c}}\right) + \mathbf{g} \tag{A.14}$$

to complete the changes needed for the objective. Notice that $\mathbf{c}$ is a linear function of $\mathbf{R}$ and $\mathbf{s}$. Thus, every term of the objective is quadratic in the new variables $\mathbf{s}_t$, $\mathbf{R}_t$, $\mathbf{v}_t$, $\mathbf{\Omega}_t$.

Body-Frame Motion Model

For the body-frame motion model, the variable $\mathbf{p}$ still remains in the constraint $\mathbf{p}_{t+1} = \mathbf{p}_t + \mathbf{R}_t\mathbf{v}_t$. Multiplying both sides by $\mathbf{R}_t^\mathsf{T}$:

$$\mathbf{R}_t^\mathsf{T}\mathbf{p}_{t+1} = \mathbf{s}_t + \mathbf{v}_t \tag{A.15}$$

From the rotation rate constraint, $\mathbf{R}_{t+1} = \mathbf{R}_t\mathbf{\Omega}_t \Rightarrow \mathbf{R}_t^\mathsf{T} = \mathbf{\Omega}_t\mathbf{R}_{t+1}^\mathsf{T}$. Plugging this in gives the constraint $\mathbf{\Omega}_t\mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{v}_t$ as desired.

World-Frame Motion Model

For the world-frame motion model, the change of variables makes the constraint $\mathbf{p}_{t+1} = \mathbf{p}_t + \mathbf{R}_t\mathbf{v}_t$ linear. Multiplying both sides by $\mathbf{R}_{t+1}^\mathsf{T}$:

$$\mathbf{R}_{t+1}^\mathsf{T}\mathbf{p}_{t+1} = \mathbf{R}_t^\mathsf{T}\mathbf{p}_t + \mathbf{v}_t \implies \mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{v}_t \tag{A.16}$$

□

A.4 Proof of Proposition 3.2.3: Closed-Form Optimal Position and Velocity

Holding $\mathbf{R}_t$ constant and with the world-frame velocity model, (3.11) is a linearly constrained least squares problem in rotated position and velocity. Thus, the minimum with respect to position and velocity is convex and the KKT conditions give a unique solution. The derivation that follows involves significant algebraic manipulation, but the main ideas follow the proof in Appendix A.2.

Dropping objective terms and constraints that do not depend on position of velocity, we have:

$$\begin{aligned} \min_{\substack{\mathbf{R}_t,\in \mathrm{SO}(3),\\ \mathbf{s}_t,\mathbf{v}_t\in\mathbb{R}^3,\\ t=1,\ldots,T,\\ l=1,\ldots,T-1}} \quad & \sum_{t=1}^{T}\sum_{i=1}^{N} w_t^i \left\|\mathbf{R}_t^\mathsf{T}\mathbf{y}_t^i - \mathbf{B}_i\mathbf{c} - \mathbf{s}_t\right\|^2 + \lambda\|\Delta\mathbf{c}\|^2 \\ & + \sum_{t=1}^{T-1} \omega_t \|\mathbf{v}_{t+1} - \mathbf{v}_t\|^2 \\ \text{s.t.} \quad & \mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{v}_t \end{aligned} \tag{A.17}$$

We rewrite (A.17) in terms of the vectorized forms of each variable, one at a time. Define:

$$\begin{aligned} \mathbf{s} &\triangleq [\mathbf{s}_1, \ldots, \mathbf{s}_T]^\mathsf{T} && \in \mathbb{R}^{3T} \\ \mathbf{v} &\triangleq [\mathbf{v}_1, \ldots, \mathbf{v}_T]^\mathsf{T} && \in \mathbb{R}^{3T-3} \\ \mathbf{r} &\triangleq [\mathrm{vec}(\mathbf{R}_1), \ldots, \mathrm{vec}(\mathbf{R}_T)]^\mathsf{T} && \in \mathbb{R}^{9T} \end{aligned} \tag{A.18}$$

We seek to write the Lagrangian of the problem as:

$$L(\mathbf{r}, \mathbf{s}, \mathbf{v}, \boldsymbol{\mu}) = \|\mathbf{A}_r\mathbf{r} - \mathbf{A}_s\mathbf{s} + \mathbf{A}_g\|^2 + \|\mathbf{A}_v\mathbf{v}\|^2 + \boldsymbol{\mu}^\mathsf{T}(\mathbf{D}_s\mathbf{s} - \mathbf{v}) \tag{A.19}$$

Algebraic Manipulation

Begin with the optimal shape coefficient $\mathbf{c}$. Proposition 3.2.1 gives:

$$\mathbf{c}^\star = 2\mathbf{G}\left(\mathbf{B}^\mathsf{T}\sum_{t=1}^{T}\mathbf{W}_t\begin{bmatrix}\mathbf{R}_t^\mathsf{T}\mathbf{y}_t^1-\mathbf{s}_t\\ \vdots\\ \mathbf{R}_t^\mathsf{T}\mathbf{y}_t^N-\mathbf{s}_t\end{bmatrix}+\lambda\bar{\mathbf{c}}\right)+\mathbf{g} \tag{A.20}$$

$$= 2\mathbf{G}\mathbf{B}^\mathsf{T}\sum_{t=1}^{T}\mathbf{W}_t\underbrace{\begin{bmatrix}\mathbf{R}_t^\mathsf{T} & & \\ & \ddots & \\ & & \mathbf{R}_t^\mathsf{T}\end{bmatrix}}_{=\mathbf{I}_N\otimes\mathbf{R}_t^\mathsf{T}}\underbrace{\begin{bmatrix}\mathbf{y}_1^t\\ \vdots\\ \mathbf{y}_N^t\end{bmatrix}}_{\triangleq\mathbf{y}^t} - 2\mathbf{G}\mathbf{B}^\mathsf{T}\sum_{t=1}^{T}\mathbf{W}_t\begin{bmatrix}\mathbf{s}_t\\ \vdots\\ \mathbf{s}_t\end{bmatrix}+\underbrace{2\mathbf{G}\lambda\bar{\mathbf{c}}+\mathbf{g}}_{\triangleq\bar{\mathbf{g}}} \tag{A.21}$$

$$= \underbrace{2\mathbf{G}\mathbf{B}^\mathsf{T}\begin{bmatrix}\mathbf{W}_1\left((\mathbf{y}^1)^\mathsf{T}\otimes\mathbf{I}_3\right) & \dots & \mathbf{W}_T\left((\mathbf{y}^T)^\mathsf{T}\otimes\mathbf{I}_3\right)\end{bmatrix}\mathbf{P}}_{\triangleq\mathbf{C}_r}\mathbf{r} - \underbrace{2\mathbf{G}\mathbf{B}^\mathsf{T}\begin{bmatrix}w_1^1\mathbf{I}_3 & \dots & w_1^T\mathbf{I}_3\\ \vdots & \ddots & \vdots\\ w_N^1\mathbf{I}_3 & \dots & w_N^T\mathbf{I}_3\end{bmatrix}}_{\triangleq\mathbf{C}_s}\mathbf{s}+\bar{\mathbf{g}} \tag{A.22}$$

$$= \mathbf{C}_r\mathbf{r}-\mathbf{C}_s\mathbf{s}+\bar{\mathbf{g}} \tag{A.23}$$

where we used the "vec trick" to rewrite the Kronecker product and $\mathbf{P}$ is the permutation matrix to convert $\mathrm{vec}(\mathbf{R}_t^\mathsf{T})$ to $\mathbf{r}=\mathrm{vec}(\mathbf{R}_t)$.

Now we rewrite the objective. The first term is:

$$f_1 = \sum_{t=1}^{T}\sum_{i=1}^{N} w_t^i\left\|\mathbf{R}_t^\mathsf{T}\mathbf{y}_t^i-\mathbf{B}_i\mathbf{c}^\star-\mathbf{s}_t\right\|^2 \tag{A.24}$$

$$= \sum_{t=1}^{T}\left\|\sqrt{\mathbf{W}_t}((\mathbf{y}^t)^\mathsf{T}\otimes\mathbf{I}_3)\mathrm{vec}(\mathbf{R}_t^\mathsf{T})-\sqrt{\mathbf{W}_t}\mathbf{B}\mathbf{c}^\star-\underbrace{\begin{bmatrix}w_1^t\mathbf{I}_3\\ \vdots\\ w_N^t\mathbf{I}_3\end{bmatrix}}_{\triangleq\bar{\mathbf{w}}_t}\mathbf{s}_t\right\|^2 \tag{A.25}$$

$$= \left\|\mathrm{diag}(\{\sqrt{\mathbf{W}_t}((\mathbf{y}^t)^\mathsf{T}\otimes\mathbf{I}_3)\}_{t=1}^T)\mathbf{P}\mathbf{r}-\mathrm{diag}(\{\mathbf{w}_t\}_{t=1}^T)\mathbf{s}-\begin{bmatrix}\sqrt{\mathbf{W}_1}\mathbf{B}\\ \vdots\\ \sqrt{\mathbf{W}_T}\mathbf{B}\end{bmatrix}(\mathbf{C}_r\mathbf{r}-\mathbf{C}_s\mathbf{s}+\bar{\mathbf{g}})\right\|^2 \tag{A.26}$$

$$\triangleq \|\tilde{\mathbf{A}}_r\mathbf{r}-\tilde{\mathbf{A}}_s\mathbf{s}+\tilde{\mathbf{A}}_g\|^2 \tag{A.27}$$

It remains to rewrite the second and third terms of the objective and the constraints. We continue with the objective's second term $\lambda\|\mathbf{c}^\star-\bar{\mathbf{c}}\|^2$:

$$\lambda\|\mathbf{c}^\star-\bar{\mathbf{c}}\|^2 = \|\sqrt{\lambda}\mathbf{C}_r\mathbf{r}-\sqrt{\lambda}\mathbf{C}_s\mathbf{s}+\sqrt{\lambda}(\bar{\mathbf{g}}-\bar{\mathbf{c}})\|^2 \tag{A.28}$$

The third term of the objective is also simple:

$$\sum_{t=1}^{T} \omega_t \|\mathbf{v}_{t+1} - \mathbf{v}_t\|^2 = \left\| \begin{bmatrix} \sqrt{\omega_1}(\mathbf{v}_2 - \mathbf{v}_1) \\ \vdots \\ \sqrt{\omega_{T-1}}(\mathbf{v}_T - \mathbf{v}_{T-1}) \end{bmatrix} \right\|^2 \tag{A.29}$$

$$= \left\| \begin{bmatrix} -\sqrt{\omega_1}\,\mathbf{I}_3 & \sqrt{\omega_1}\,\mathbf{I}_3 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & -\sqrt{\omega_{T-1}}\,\mathbf{I}_3 & \sqrt{\omega_T}\,\mathbf{I}_3 \end{bmatrix} \mathbf{v} \right\|^2 \tag{A.30}$$

$$\triangleq \|\mathbf{A}_v \mathbf{v}\|^2 \tag{A.31}$$

Lastly, we rewrite the set of constraints $\mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{v}$:

$$\mathbf{0} = \mathbf{s}_{t+1} - \mathbf{s}_t - \mathbf{v} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_3 & \mathbf{I}_3 & \dots & \mathbf{0} & \mathbf{0} \\ & & & \ddots & & \\ & & & & -\mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix} \mathbf{s} - \mathbf{v} \triangleq \mathbf{D}_s \mathbf{s} - \mathbf{v} \tag{A.32}$$

Optimal Position and Velocity

As with shape, use stationarity and primal feasibility. Let $\mathbf{A}_r \triangleq [\tilde{\mathbf{A}}_r^\mathsf{T}, \sqrt{\lambda}\mathbf{C}_r^\mathsf{T}]^\mathsf{T}$, $\mathbf{A}_s \triangleq [\tilde{\mathbf{A}}_s^\mathsf{T}, \sqrt{\lambda}\mathbf{C}_s^\mathsf{T}]^\mathsf{T}$, and $\mathbf{A}_g \triangleq [\tilde{\mathbf{A}}_g^\mathsf{T}, \sqrt{\lambda}(\bar{\mathbf{g}} - \bar{\mathbf{c}})^\mathsf{T}]^\mathsf{T}$ be the coefficients of the rotations, positions, and scalar parts respectively. The Lagrangian is:

$$L(\mathbf{r}, \mathbf{s}, \mathbf{v}, \boldsymbol{\mu}) = \|\mathbf{A}_r \mathbf{r} - \mathbf{A}_s \mathbf{s} + \mathbf{A}_g\|^2 + \|\mathbf{A}_v \mathbf{v}\|^2 + \boldsymbol{\mu}^\mathsf{T}(\mathbf{D}_s \mathbf{s} - \mathbf{v}) \tag{A.33}$$

The stationarity conditions for $\mathbf{s}$ and $\mathbf{v}$ are:

$$0 = \nabla_\mathbf{s} L = 2\mathbf{A}_s^\mathsf{T}\mathbf{A}_g + 2\mathbf{A}_s^\mathsf{T}\mathbf{A}_r \mathbf{r} + 2\mathbf{A}_s^\mathsf{T}\mathbf{A}_s \mathbf{s} + \mathbf{D}_s^\mathsf{T}\boldsymbol{\mu} \tag{A.34}$$

$$0 = \nabla_\mathbf{v} L = 2\mathbf{A}_v^\mathsf{T}\mathbf{A}_v \mathbf{v} - \boldsymbol{\mu} \tag{A.35}$$

Putting this together with the constraint $\mathbf{0} = \mathbf{D}_s \mathbf{s} - \mathbf{v}$, we arrive at the linear system in (3.14). □

A.5 Additional Experimental Results

A.5.1 Additional Synthetic Results

In Section 3.4.1 we showed the robustness of CAST⋆ to measurement and process noise, and the robustness of CAST# to outliers. Here we give the runtimes of each method (Table A.1), show results for the choice of weights resulting from MAP estimation (Fig. A.1), and address the EKF results (Fig. A.3). Having established similar performance of the world and body-frame motion models, this appendix considers the body-frame motion model apart from run times.

Table A.1: Synthetic Experiment Runtimes

Runtime (s)	PACE	CAST⋆-B			CAST⋆-W			CAST⋆-U
		4	8	12	4	8	12	
Meas. Noise	0.0028	0.483	2.15	5.49	0.0786	0.427	1.23	5.25
Proc. Noise	0.0040	0.857	4.05	10.6	0.0315	0.171	0.507	10.2

From Table A.1 we observe CAST⋆ is the slower than the single-frame method PACE, which is expected. While the results are obtained with an unoptimized MATLAB implementation, the world-frame version is still fast enough for real-time use. This aside, the variable horizon length allows a trade-off between computational speed and accuracy. As computation improves, the benefits of certifiable optimality and increased accuracy make CAST⋆ an attractive choice of tracking algorithm.

Recall that in the tests in Section 3.4 we chose the velocity weights to be $\omega_t = 1$ instead of setting them as prescribed by MAP estimation (where they should be taken as the inverse of the variance of the prior). This is equivalent to increasing the standard deviation of the velocity noise by a factor of 10; in other words, it reduces the effect of motion smoothing. Fig. A.1 shows that using the true velocity covariance degrades tightness, although it does not have any visible effect on the accuracy results.

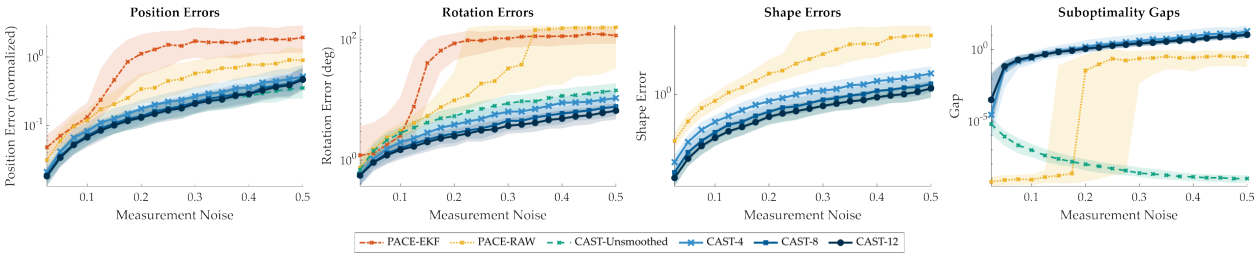
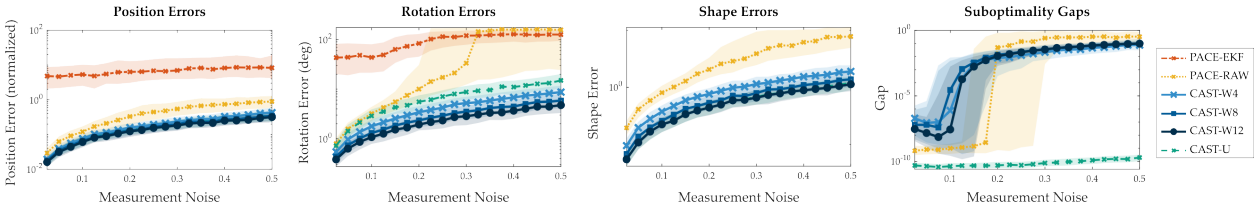


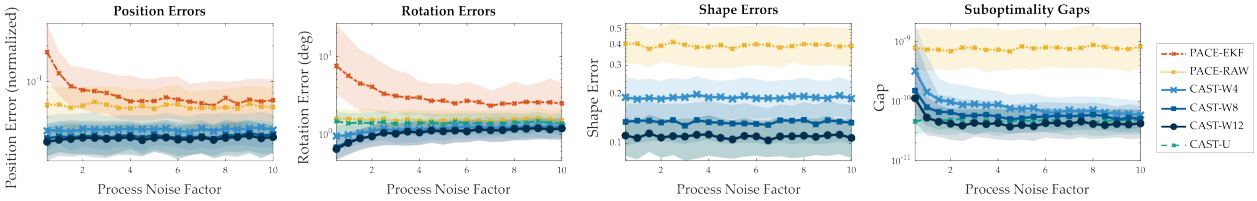
Figure A.1: **Performance of CAST⋆ in synthetic experiments with increasing measurement noise.** Robustness to measurement noise with CAST⋆ using the inverse of the simulated velocity covariance for the velocity weights ω_t. The key difference between this plot and Fig. 3.3(a) lies in the suboptimality gap figure, where CAST⋆ loses tightness quickly. Despite losing its optimality certificate, CAST⋆ maintains the lowest position, rotation, and shape errors.

We also present synthetic results for the world-frame motion model, an analog to the body-frame results in Fig. 3.3. Here, we generate synthetic data according to the world-frame motion model (3.4) and perturb it with Gaussian and Langevin random noise. These results are nearly identical to the synthetic results for the body-frame model; increasing the number of frames improves accuracy, the motion model is effective for small number of frames, and the compatibility tests (MILP) handle the majority of outliers quickly. One key difference with Fig. 3.3 is outlier robustness: CAST# remains accurate up to and just beyond 60% random outliers. This result may be partly an artifact of how the same outlier generation process manifests differently for different motion models.

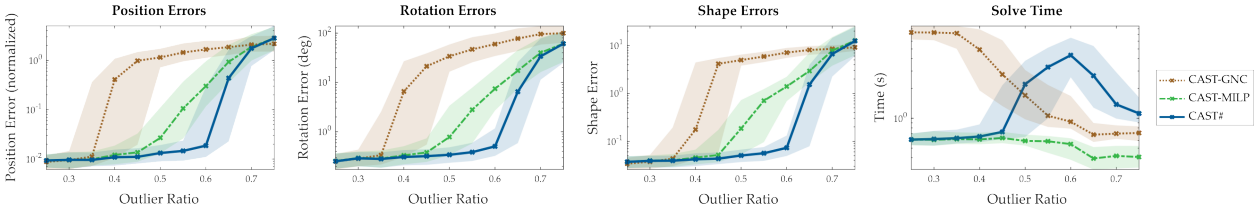
Lastly, we present additional results showing the performance of the EKF using perturbed ground truth data instead of PACE. Specifically, we perturb the ground truth poses according



(a) Measurement noise robustness of CAST⋆ with fixed process noise.



(b) Process noise robustness of CAST⋆ with fixed measurement noise.



(c) Outlier robustness of CAST# and ablations.

Figure A.2: **Performance of CAST⋆-W and CAST#-W in synthetic experiments**

to a zero-mean Gaussian with standard deviation equal to 1/25th of the measurement noise for position and 1/50th for rotation (arbitrarily chosen as realistic values). Fig. A.3 shows the median EKF estimate consistently beats the perturbed ground truth value (results are averaged over 500 independent trials for each noise value). The large interquartile range is likely because of errors due to linearization, particularly of the constant twist motion model. The EKF likely struggled when using PACE's poses in the measurement update because of the high variance and heavy-tailed distribution of the estimates.

Table A.2: Additional YCBInEOAT Results

Method	Bleach		Soup	
	ADD	ADD-S	ADD	ADD-S
6-PACK	4.18	18.00	12.82	60.32
TEASER++	35.39	46.40	65.85	81.53
MaskFusion	29.83	43.31	5.65	6.45
BundleTrack	**89.34**	**94.72**	**86.00**	95.13
BundleSDF	85.59	93.11	80.54	**96.47**
CAST#-8	47.53	45.82	27.61	41.70
CAST#-GT	62.19	75.14	37.07	63.29

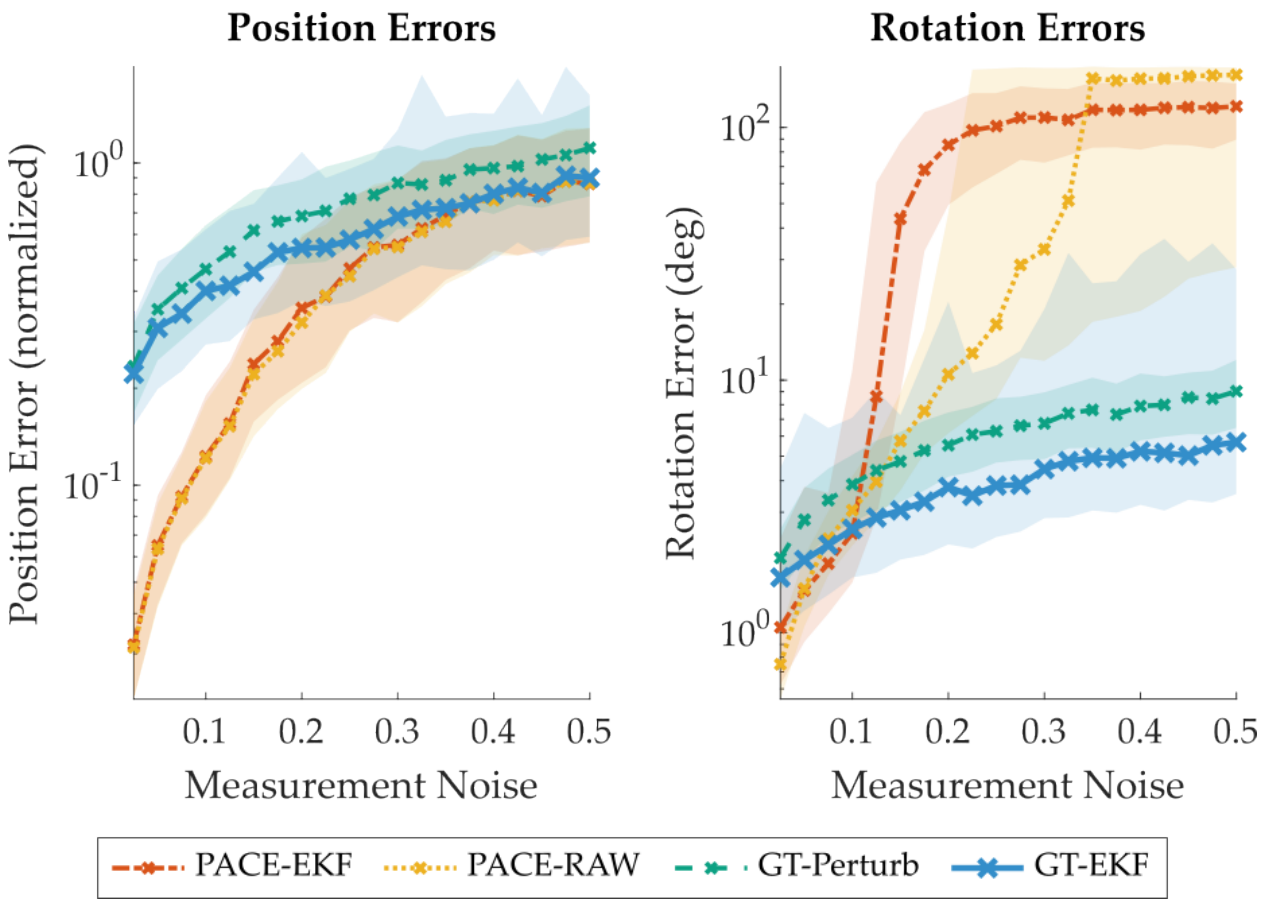


Figure A.3: **Extended Kalman Filter with perturbed ground truth measurements.** With Gaussian-perturbed ground truth measurements, the extended Kalman filter outperforms the raw measurements in median error across measurement noise values. This supports our claim that the EKF performs poorly using pose estimate from PACE, likely due to the high variance and heavy-tailed distribution of the estimates.

A.5.2 Results for Bleach and Soup on YCBInEOAT

Table A.2 shows scores for all tested methods on the "soup" and "bleach" objects. As mentioned in the text, the soup object is particularly difficult because it is very small and cylindrically symmetric, which CAST* is not designed to handle (other approaches also achieve low scores, compared to the other objects). The bleach object is larger but matches the background color, making keypoint detection difficult.

Appendix B

Additional Results and Proofs for Conformalized Pose and Uncertainty Estimation

B.1 Explicit Form of Rotation Equality Constraints

For completeness, we provide explicit forms of the equality constraint matrices $\mathbf{Q}_k$ used in Chapter 4. We write the constraint matrices in sparse form, specifying only the nonzero elements. The notation (i, j, l) means $\mathbf{Q}_{i,j} = l$. Let $\mathbf{R} \in \mathrm{SO}(3)$ and denote its columns by $\mathbf{r}_i$, $i = 1, 2, 3$.

Three constraint matrices enforce that the columns of $\mathbf{R}$ are unit norm:

$$\begin{array}{lll} \|\mathbf{r}_1\|^2 = 1 & \Longrightarrow & \mathbf{Q}_1 : (1,1,1), (2,2,1), (3,3,1), (13,13,-1) \\ \|\mathbf{r}_2\|^2 = 1 & \Longrightarrow & \mathbf{Q}_2 : (4,1,1), (5,2,1), (6,3,1), (13,13,-1) \\ \|\mathbf{r}_3\|^2 = 1 & \Longrightarrow & \mathbf{Q}_3 : (7,1,1), (8,2,1), (9,3,1), (13,13,-1) \end{array} \tag{B.1}$$

Three constraint matrices enforce the columns are orthogonal (have zero dot product). Although we don't explicitly specify, all $\mathbf{Q}_k$ are symmetric.

$$\begin{array}{lll} \mathbf{r}_1 \cdot \mathbf{r}_2 = 1 & \Longrightarrow & \mathbf{Q}_4 : (1,4,1), (2,5,1), (3,6,1) \\ \mathbf{r}_1 \cdot \mathbf{r}_3 = 1 & \Longrightarrow & \mathbf{Q}_5 : (1,7,1), (2,8,1), (3,9,1) \\ \mathbf{r}_3 \cdot \mathbf{r}_2 = 1 & \Longrightarrow & \mathbf{Q}_6 : (4,7,1), (5,8,1), (6,9,1) \end{array} \tag{B.2}$$

The final 9 constraint matrices enforce the right hand rule cross products between the

Table B.1: Coverage Percentages for ∞-Norm Pose Uncertainty Set

		$\alpha = 0.1$				$\alpha = 0.4$			
	N	Keypts		($\mathcal{P}_\infty$)		Keypts		($\mathcal{P}_\infty$)	
Calibration		S	R	S	R	S	R	S	R
ape	9	41.2	87.3	2.0	60.2	19.6	53.1	0.1	9.9
can	8	61.8	87.8	9.6	59.4	21.1	53.8	0	6.6
cat	10	62.7	88.8	8.6	57.3	32.3	56.8	0.1	5.1
duck	9	31.7	88.4	1.2	70.2	13.2	54.8	0.1	16.5
driller	11	66.5	87.5	17.4	56.7	23.6	53.4	0.2	6.8
eggbox	9	20.6	87.7	1.6	58.8	8.8	58.5	0	23.5
glue	10	42.3	86.3	2.0	54.7	13.2	50.1	0.6	8.0
holepuncher	10	45.1	85.2	1.1	42.2	18.2	52.2	0.1	4.1
average		46.5	87.4	5.7	57.3	18.7	54.2	0.1	10.0

columns of $\mathbf{R}$, element-wise.

$$\begin{aligned}
\mathbf{r}_1 \times \mathbf{r}_2 = \mathbf{r}_3 \quad &\Longrightarrow \quad \begin{cases} \mathbf{Q}_7 : (2,6,1),(3,5,-1),(13,7,-1) \\ \mathbf{Q}_8 : (3,4,1),(1,6,-1),(13,8,-1) \\ \mathbf{Q}_9 : (1,5,1),(2,4,-1),(13,9,-1) \end{cases} \\
\mathbf{r}_2 \times \mathbf{r}_3 = \mathbf{r}_1 \quad &\Longrightarrow \quad \begin{cases} \mathbf{Q}_{10} : (5,9,1),(6,8,-1),(13,1,-1) \\ \mathbf{Q}_{11} : (6,7,1),(4,9,-1),(13,2,-1) \\ \mathbf{Q}_{12} : (4,8,1),(5,7,-1),(13,3,-1) \end{cases} \\
\mathbf{r}_3 \times \mathbf{r}_1 = \mathbf{r}_2 \quad &\Longrightarrow \quad \begin{cases} \mathbf{Q}_{13} : (8,3,1),(9,2,-1),(13,4,-1) \\ \mathbf{Q}_{14} : (9,1,1),(7,3,-1),(13,5,-1) \\ \mathbf{Q}_{15} : (7,2,1),(8,1,-1),(13,6,-1) \end{cases}
\end{aligned} \tag{B.3}$$

B.2 Additional Experimental Results

B.2.1 Calibration for ∞-Norm

For completeness, we provide the analog to Table 4.1 for the ∞-norm case. As in Section 4.4.1 we compare calibration on 200 independent synthetically-generated images (Synthetic) and calibration on the 200 real images selected by BOP [83]. The results, shown in Table B.1, are not significantly different from the 2-norm case. Across all calibration data the coverage results are slightly worse. The same coverage gap of $30-40\%$ holds between keypoints and poses.

B.2.2 Extra Central Pose Results

In this section we provide feasibility, tightness, and runtime results (Table B.2) and additional central pose results (Table B.3). For both tables, we compare against two ablations of our

method. The ablation Loc_p omits the semidefinite relaxation and exclusively runs the local solver using Ipopt [78]. To initialize, we give a random rotation in SO(3) and zero for the margins and translations. We rerun the local solver up to 25 times, terminating at the first iteration where it converges to a solution. This ablation is designed to show the value of the semidefinite initialization, which generates an initial guess and only runs the local solver once.

We also compare against a *maximum margin* formulation (labeled as "Margin"). The key idea is to find the pose most consistent with the backprojection constraints for a single confidence α. In particular, we solve the following optimization problem:

$$\begin{aligned} \max_{\substack{\mathbf{t}\in\mathbb{R}^3,\,\mathbf{R}\in\text{SO}(3)\\ \boldsymbol{\gamma}\in\mathbb{R}^N}} \quad & \sum_{i=1}^{N}\gamma_i \\ \text{s.t.} \quad & \left\|(\mathbf{y}_i\hat{\mathbf{e}}_3^\mathsf{T}-\mathbf{I}_3)\mathbf{K}(\mathbf{R}\mathbf{b}_i+\mathbf{t})\right\|_2^2 \leq \left(r_i\hat{\mathbf{e}}_3^\mathsf{T}\mathbf{K}(\mathbf{R}\mathbf{b}_i+\mathbf{t})\right)^2-\gamma_i, \\ & \hat{\mathbf{e}}_3\cdot\mathbf{K}(\mathbf{R}\mathbf{b}_i+\mathbf{t})>0, \quad i=1,...,N \end{aligned} \tag{B.4}$$

Note that we only implement the maximum margin for the case $p=2$. Eq. (B.4) simply maximizes the distance from the boundary of each backprojection constraint (BP_2) subject to the chirality constraint (FoC). Observe that (B.4) is a quadratically-constrained quadratic problem in $\mathbf{R}$ and $\mathbf{t}$, but both the objective and constraints are linear in the margin $\boldsymbol{\gamma}$. Thus, we can use a first-order semidefinite relaxation to solve the problem (see Section 2.1). As shown in Table B.2, this relaxation is empirically tight. The key issue with the maximum margin formulation is its bias towards estimates which are further away from the camera. This significantly limits its performance when keypoint uncertainty sets are large.

Feasibility, Tightness, and Runtime Results for Central Pose

Table B.2 provides feasibility and tightness results for our central pose algorithms under real calibration. Feasibility measures the percentage of pose estimates which satisfy the chirality (FoC) and backprojection (BP_p) constraints for the returned value of $\boldsymbol{\gamma}$. Our tightness threshold is 10^{-3}. Runtime results reflect performance in Julia with precompilation time excluded. Results are means across all objects.

As previously mentioned, the maximum margin is fast and an empirically tight semidefinite relaxation for the majority of cases. Interestingly, OURS_2 is tight for just over 40% of frames; although the results are generally worse than OURS_∞, this suggests $p=2$ can find good solutions when the measurements are reasonable (note that tightness does not imply a reasonable pose estimate; rather, it implies we found the global solution to the optimization problem (4.10)). The poor pose estimation performance when $p=2$ may suggest bad outlier handling. Lastly, we note that the local solver ablations are much slower than simply using a semidefinite relaxation. This reflects the underlying non-convexity of the problem; the solver often fails to solve with a random initial guess.

Central Pose Ablations

We also give extra pose estimation results for our method and ablations in Table B.3. In the table, R refers to calibration on real data and S refers to calibration on synthetic data. The

Table B.2: Feasibility and Tightness Results

	$\alpha = 0.1$			$\alpha = 0.4$		
	Feasible (%)	Tight (%)	Time (ms)	Feasible (%)	Tight (%)	Time (ms)
Loc_2	53.0		153.3	54.3		158.6
Loc_∞	99.7		478.1	99.7		362.7
OURS_2	55.7	43.6	88.1	57.5	41.1	106.3
OURS_∞	95.3	0	116.4	97.8	0	90.7
Margin	98.6	83.5	19.8	97.8	75.6	26.2

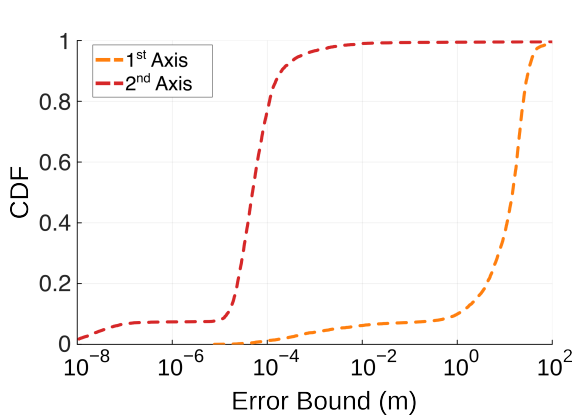
Table B.3: Percentage of 2D Projection Errors Under 5 Pixels with Additional Methods

	$\alpha = 0.1$								$\alpha = 0.4$							
	Loc_2	Loc_∞	OURS_2		OURS_∞		Margin		Loc_2	Loc_∞	OURS_2		OURS_∞		Margin	
	R	R	R	S	R	S	R	S	R	R	R	S	R	S	R	S
ape	76.9	41.0	77.6	76.8	77.0	59.3	61.1	75.8	76.6	58.0	76.6	76.7	76.4	51.8	75.4	76.9
can	16.2	63.9	19.2	53.4	82.2	76.5	72.0	82.0	69.9	77.0	69.9	84.0	81.3	65.3	82.3	85.6
cat	77.2	44.7	77.8	77.3	71.1	62.4	65.0	74.1	77.8	63.8	77.8	77.0	73.7	62.3	75.5	76.3
duck	80.6	45.4	80.7	80.5	79.0	57.4	57.0	79.9	80.5	52.5	80.5	80.6	76.9	48.6	78.4	80.3
driller	65.5	42.0	65.3	65.9	64.6	56.3	53.3	62.5	66.2	55.7	66.2	66.7	64.3	38.6	64.6	66.0
eggbox	0	1.6	0	0.2	4.8	1.7	0	2.6	0.1	1.7	0.1	3.1	4.5	2.9	0	4.7
glue	1.7	25.0	1.5	32.9	55.9	45.4	29.0	55.2	8.0	39.5	8.0	55.2	55.7	43.6	53.3	56.4
holepuncher	7.4	49.2	7.0	62.0	76.0	61.2	69.2	75.3	48.5	65.0	48.5	74.1	74.0	50.5	73.9	74.0
average	40.7	39.1	41.1	56.1	63.8	52.5	50.8	63.4	53.5	51.7	53.5	64.7	63.4	45.5	62.9	65.0

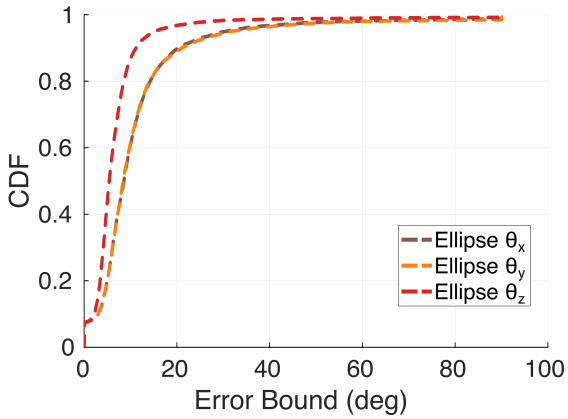
main text contains results only from real calibration due to the exchangeability observation in Table 4.1. However, including the calibration data in evaluation data is statistically dubious. The synthetic uncertainty sets tend to be significantly tighter than the ones from real calibration. We report the percentage of 2D projection errors less than 5 pixels for each object; see [81].

For real calibration, our approach with $p = \infty$ achieves the best 2D projection error and conclusively outperforms the local solver. The performance improvement from $\alpha = 0.1$ to $\alpha = 0.4$ or from real to synthetic data reflects behavior with tighter uncertainty sets. The local solvers, OURS_2, and maximum margin all perform significantly better with smaller uncertainty sets. This suggests that the keypoint detections are generally reasonably accurate (maximum margin is biased towards pose estimates within the fixed-confidence keypoint uncertainty bounds). It also suggests the semidefinite relaxation produces better results when given smaller keypoint uncertainty radii.

Lastly, it is clear from Table B.3 that the local solver is outperformed by initializing with a semidefinite relaxation. This is expected in the case $p = 2$ when the relaxation is tight, but somewhat surprising for $p = \infty$ when the relaxation is essentially never tight.



(a) CDF of translation bounds on LM-O. We show the half-length of the first and second principal axes of our marginalized ellipse.



(b) CDF of rotation bounds on LM-O. For our approach, we give angular bounds about the x, y, and z axes.

Figure B.1: **Tightness of angular and translational bounds for $\alpha = 0.4$.** The tighter keypoint uncertainty sets are reflected in tighter translation bounds (a) and rotation bounds (b). In particular, the translation bounds are loose along the optical axis but very tight elsewhere. The rotational bounds are largely below 10 degrees.

B.2.3 Extra Bounding Ellipse Results

This section provides additional quantitative and qualitative results for the bounding ellipse. In Fig. B.1 we show the cumulative distribution function (CDF) of our rotation and translation bounds for $\alpha = 0.4$. We do not compare against the baselines GRCC [25] do not release $\alpha = 0.4$ results. For this figure, we consider the 7316 frames where the central pose estimate ($p = \infty$) returned a feasible point and exclude the eggbox object. The rotation results are much tighter, largely achieving bounds under 20 degrees for all axes. We also show the full domain of the translation error bounds. While the first principal axis is relatively loose, it is clear that the second and third principal axes are quite tight. As before, this reflects overly conservative uncertainty along the optical axis but high confidence in perpendicular directions.

Fig. B.2 provides two additional qualitative examples of the rotation uncertainty set at fixed rotation for $\alpha = 0.1$. For the ape object, the pose estimate was slightly off, but the uncertainty set correctly covers the entire object. We also show a failure case (Fig. B.2b) where the pose uncertainty set does not cover the ground truth pose. As reported in Table 4.1, this occurs about 10% of the time for $\alpha = 0.1$.

(a) A pose estimate of the ape object with rotational uncertainty at constant translation.

(b) A pose estimate of the duck object with rotational uncertainty. The uncertainty set clearly does not include the true duck.

Figure B.2: **Extra examples of qualitative rotation sets.** The ape uncertainty set (a) covers the entire ape, in contrast to the pose estimate (outlined in black), which is slightly off. The duck pose estimate and uncertainty set (b) are both wrong, in an example of a case where the pose uncertainty set does not cover the true pose. Both images show frame 1099 of the LM-O dataset.

Appendix C

Additional Results and Proofs for SCF

C.1 Proof of Proposition 5.2.3: Closed Form Solution for Optimal Shape

TODO

C.2 Stereographic Projection of Unit Quaternions

To visualize unit quaternions in Fig. 5.1 we stereographically project them from the 4-sphere onto the volume of the unit 3-ball. The projection is simple. Let $\mathbf{q} \in \mathbb{S}^3$ denote a unit quaternion. Recall that $-\mathbf{q}$ represents the same rotation. When the scalar part is positive, the vector part can be understood as coordinates within the 3-ball. When the scalar part is negative, we simply negate the quaternion and use the vector part. Thus, we project by taking the vector part times the sign of the scalar part. For a quaternion $\mathbf{q} = [q_1, \mathbf{q}_v^\mathsf{T}]^\mathsf{T}$,

$$\mathbf{q}_{\text{proj}} = -\text{sign}(q_1)\mathbf{q}_v \tag{C.1}$$

As with any projection this is an imperfect representation of the space. Points which are on the boundary and on opposite sides of the unit ball are actually quite close to each other. The projection also makes the quaternion $\mathbf{q}$ indistinguishable from its inverse $\mathbf{q}^{-1} = [q_1, -\mathbf{q}_v^\mathsf{T}]^\mathsf{T}$.

C.3 Power Method Ablation

The Power solver we compare against in Section 5.4 is a simple ablation of SCF (Algorithm 5.1). Instead of computing the eigenvalues at each iteration, Power simply multiplies the data matrix by the quaternion from the previous iteration and renormalizes. The algorithm terminates at a stationary point, which also corresponds to an eigenvector. See Algorithm C.1.

Data: $\mathbf{A}(\mathbf{q})$ and $\mathbf{D}$ from (5.15)
Result: $\mathbf{q}$ satisfying (5.18)
initialize $\mathbf{q}_0$
for $t \leftarrow 1$ **to** T **do**
  $\mathbf{q}_{t+1} \leftarrow (\mathbf{A}(\mathbf{q}_t) + \mathbf{D})\mathbf{q}_t$
  $\mathbf{q}_{t+1} \leftarrow \mathbf{q}_{t+1}/\|\mathbf{q}_{t+1}\|$
  `/* termination condition */`
  if $\sin \measuredangle(\mathbf{q}_t, \mathbf{q}_{t+1}) < \epsilon$ **then**
    $\mathbf{q} \leftarrow \mathbf{q}_t$
    break
  end
end

Algorithm C.1: A power method for solving (5.16).

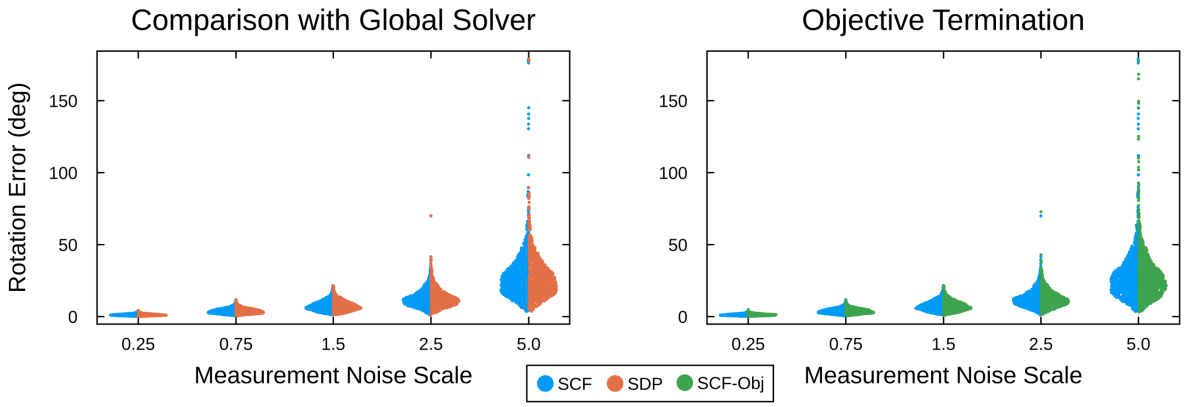


Figure C.1: **Comparison with Global Solver and Objective Termination.** For $K > N$ the performance depends heavily on choice of regularization λ. For $\lambda = 1.0$, we plot histograms of rotation error at selected noise scales. As in Fig. 5.2, all three methods achieve similar rotation accuracy across noise scales.

C.4 Gauss-Newton and Levenburg-Marquardt Solvers

TODO GN and LM are based on axis-angle linearization specialized to (5.13).

C.5 Extra Experimental Results with Larger Shape Library

Performance with a larger shape library strongly depends on the choice of regularization constant λ. For $\lambda \to 0$ the problem becomes ill-conditioned and all solvers slow significantly. Large λ greatly reduces shape ambiguity by imposing a strong prior. For these results, we test with $K = 25$, $N = 10$, and $\lambda = 1.0$. This moderate choice of λ produces results which are similar to Section 5.4.

Table C.1: Mean and 90th Percentile of Solver Runtimes (Large K)

	$\sigma_m = 0.25$		$\sigma_m = 2.5$	
Method	Mean (ms)	p90 (ms)	Mean (ms)	p90 (ms)
SCF	0.673	0.584	0.718	0.693
Manopt	3.779	4.456	3.526	4.334
G-N	3.491	4.220	5.816	7.436
L-M	3.642	4.418	5.655	7.452
SDP	13.715	14.543	13.936	14.780
SCF-Obj	**0.586**	**0.510**	**0.587**	**0.555**
Power	0.645	0.736	0.727	0.836

We compare the estimation accuracy of SCF with SDP and SCF-Obj in Fig. C.1. As before, the three methods achieve virtually the same performance across noise scales. More interestingly, the runtimes in Table C.1 are slightly different than the low K case. In particular, SCF runs at around 700 μs on average, which is near the performance of Power and is outperformed by early objective termination SCF-Obj. For all three of these methods based on first order conditions the 90th percentile of runtimes (p90) is below the mean, suggesting a small concentration of problems for which many iterations were required. SCF is still significantly faster than the other methods, which do not suffer as big of a performance drop for larger K. The exception is SDP, which is more than 4 times slower than before.

References

[1] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. Guibas. "Normalized object coordinate space for category-level 6d object pose and size estimation". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2642–2651.

[2] J. Shi, H. Yang, and L. Carlone. "Optimal and Robust Category-level Perception: Object Pose and Shape Estimation from 2D and 3D Semantic Keypoints". *IEEE Trans. Robotics* 39.5 (2023). (pdf), pp. 4131–4151.

[3] H. Yang and M. Pavone. *Object Pose Estimation with Statistical Guarantees: Conformal Keypoint Detection and Geometric Uncertainty Propagation*. 2023. arXiv: 2303.12246 [cs.CV].

[4] K. Arun, T. Huang, and S. Blostein. "Least-Squares Fitting of Two 3-D Point Sets". *IEEE Trans. Pattern Anal. Machine Intell.* 9.5 (Sept. 1987), pp. 698–700.

[5] J. Gower and G. Dijksterhuis. "Procrustes Problems". *Procrustes Problems, Oxford Statistical Science Series* 30 (Jan. 2005).

[6] G. Sharp, S. Lee, and D. Wehe. "ICP registration using invariant features". *IEEE Trans. Pattern Anal. Machine Intell.* 24.1 (Jan. 2002), pp. 90–102.

[7] A. Myronenko and X. Song. "Point set registration: Coherent point drift". *IEEE Trans. Pattern Anal. Machine Intell.* 32.12 (2010), pp. 2262–2275.

[8] K. Schmeckpeper, P. Osteen, Y. Wang, G. Pavlakos, K. Chaney, W. Jordan, X. Zhou, K. Derpanis, and K. Daniilidis. "Semantic keypoint-based pose estimation from single RGB frames". *arXiv preprint arXiv:2204.05864* (2022).

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. "Mask R-CNN". In: *Intl. Conf. on Computer Vision (ICCV)*. 2017, pp. 2980–2988.

[10] S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi. "Discovery of latent 3d keypoints via end-to-end geometric reasoning". *arXiv preprint arXiv:1807.03146* (2018).

[11] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. "PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4561–4570.

[12] H. Yang, J. Shi, and L. Carlone. "TEASER: Fast and Certifiable Point Cloud Registration". *IEEE Trans. Robotics* 37.2 (2020). extended arXiv version 2001.07715 (pdf), pp. 314–333.

[13] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox. "PoseRBPF: A Rao-Blackwellized Particle Filter for 6D Object Pose Tracking". In: *Robotics: Science and Systems (RSS)*. 2019.

[14] B. Wen, W. Yang, J. Kautz, and S. Birchfield. " FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects ". In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2024, pp. 17868–17879. DOI: 10.1109/CVPR52733.2024.01692. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01692.

[15] J. Shi, R. Talak, H. Zhang, D. Jin, and L. Carlone. "CRISP: Object Pose and Shape Estimation with Test-Time Adaptation". *arXiv preprint: 2412.01052* (2024).

[16] M. Tian, M. H. Ang, and G. H. Lee. "Shape prior deformation for categorical 6d object pose and size estimation". In: *European Conf. on Computer Vision (ECCV)*. Springer. 2020, pp. 530–546.

[17] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield. "Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 606–617.

[18] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. "Use of active shape models for locating structures in medical images". *Image and vision computing* 12.6 (1994), pp. 355–365.

[19] Y. Zhang and J. Leonard. *ShapeICP: Iterative Category-level Object Pose and Shape Estimation from Depth*. 2024. arXiv: 2408.13147 [cs.CV]. URL: https://arxiv.org/abs/2408.13147.

[20] A. P. Fard, H. Abdollahi, and M. Mahoor. "ASMNet: a Lightweight Deep Neural Network for Face Alignment and Pose Estimation". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, June 2021, pp. 1521–1530. DOI: 10.1109/cvprw53098.2021.00168. URL: http://dx.doi.org/10.1109/CVPRW53098.2021.00168.

[21] D. Reis, J. Kupec, J. Hong, and A. Daoudi. *Real-Time Flying Object Detection with YOLOv8*. 2024. arXiv: 2305.09972 [cs.CV]. URL: https://arxiv.org/abs/2305.09972.

[22] T. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[23] G. Shafer and V. Vovk. "A Tutorial on Conformal Prediction". *J. of Machine Learning Research* (2008), p. 51.

[24] Y. Gao, Y. Tang, H. Qi, and H. Yang. "CLOSURE: Fast Quantification of Pose Uncertainty Sets". In: *Robotics: Science and Systems (RSS)*. 2024.

[25] Y. Tang, J.-B. Lasserre, and H. Yang. "Uncertainty quantification of set-membership estimation in control and perception: Revisiting the minimum enclosing ellipsoid". In: *Proceedings of the 6th Annual Learning for Dynamics and Control Conference*. Ed. by A. Abate, M. Cannon, K. Margellos, and A. Papachristodoulou. Vol. 242. Proceedings of Machine Learning Research. PMLR, July 2024, pp. 286–298. URL: https://proceedings.mlr.press/v242/tang24a.html.

[26] Y. Bar-Shalom. *Multitarget multisensor tracking: Advanced applications*. Norwood, MA: Artech House, 1992.

[27] V. Lepetit and P. Fua. *Monocular Model-Based 3D Tracking of Rigid Objects: A Survey*. Now Foundations and Trends, 2005. DOI: 10.1561/0600000001.

[28] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House radar library. Artech House, 1999. ISBN: 9781580530064. URL: https://books.google.com/books?id=lTIfAQAAIAAJ.

[29] N. A. Piga, F. Bottarel, C. Fantacci, G. Vezzani, U. Pattacini, and L. Natale. "Maskukf: An instance segmentation aided unscented kalman filter for 6d object pose and velocity tracking". *Frontiers in Robotics and AI* 8 (2021), p. 594583.

[30] B. Wen, C. Mitash, B. Ren, and K. E. Bekris. "se(3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 10367–10373.

[31] L. Wang, S. Yan, J. Zhen, Y. Liu, M. Zhang, G. Zhang, and X. Zhou. "Deep Active Contours for Real-time 6-DoF Object Tracking". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

[32] G. Simon and M.-O. Berger. "A two-stage robust statistical method for temporal registration from features of various type". In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 261–266. DOI: 10.1109/ICCV.1998.710728.

[33] G. Simon and M.-O. Berger. "Pose estimation for planar structures". *IEEE Computer Graphics and Applications* 22.6 (2002), pp. 46–53. DOI: 10.1109/MCG.2002.1046628.

[34] C. Wang, R. Martín-Martín, D. Xu, J. Lv, C. Lu, L. Fei-Fei, S. Savarese, and Y. Zhu. "6-pack: Category-level 6d pose tracker with anchor-based keypoints". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 10059–10066.

[35] B. Wen and K. Bekris. "Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 8067–8074.

[36] Y. Weng, H. Wang, Q. Zhou, Y. Qin, Y. Duan, Q. Fan, B. Chen, H. Su, and L. J. Guibas. "CAPTRA: CAtegory-level Pose Tracking for Rigid and Articulated Objects from Point Clouds". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2021. DOI: 10.1109/iccv48922.2021.01296. URL: http://dx.doi.org/10.1109/ICCV48922.2021.01296.

[37] M. Rünz, M. Buffier, and L. Agapito. "MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects". In: *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2018, pp. 10–20.

[38] J. Sun, Y. Wang, M. Feng, D. Wang, J. Zhao, C. Stachniss, and X. Chen. "ICK-Track: A Category-Level 6-DoF Pose Tracker Using Inter-Frame Consistent Keypoints for Aerial Manipulation". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2022. DOI: 10.1109/iros47612.2022.9982183. URL: http://dx.doi.org/10.1109/IROS47612.2022.9982183.

[39] H. Yang and L. Carlone. "Certifiably Optimal Outlier-Robust Geometric Perception: Semidefinite Relaxations and Scalable Global Optimization". *IEEE Trans. Pattern Anal. Machine Intell.* (2022). (pdf).

[40] J. Lasserre. *Moments, positive polynomials and their applications*. Vol. 1. World Scientific, 2010.

[41] N. Shor. "Quadratic optimization problems". *Izv. Akad. Nauk SSSR Tekhn. Kibernet.* 1 (1987), pp. 128–139.

[42] L. Brynte, V. Larsson, J. P. Iglesias, C. Olsson, and F. Kahl. "On the Tightness of Semidefinite Relaxations for Rotation Estimation". *Journal of Mathematical Imaging and Vision* 64.1 (Oct. 2021), pp. 57–67. ISSN: 1573-7683. DOI: 10.1007/s10851-021-01054-y. URL: http://dx.doi.org/10.1007/s10851-021-01054-y.

[43] J. Saunderson, P. Parrilo, and A. Willsky. "Semidefinite relaxations for optimization problems over rotation matrices". In: *IEEE Conf. on Decision and Control (CDC)*. May 2014.

[44] D. Rosen, L. Carlone, A. Bandeira, and J. Leonard. "SE-Sync: a certifiably correct algorithm for synchronization over the Special Euclidean group". *Intl. J. of Robotics Research* (2018). arxiv preprint: 1611.00128, (pdf).

[45] L. Carlone and F. Dellaert. "Duality-based Verification Techniques for 2D SLAM". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. (pdf) (code). 2015, pp. 4589–4596.

[46] J. Briales and J. Gonzalez-Jimenez. "Convex Global 3D Registration with Lagrangian Duality". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[47] J. Zhao, W. Xu, and L. Kneip. "A Certifiably Globally Optimal Solution to Generalized Essential Matrix Estimation". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[48] M. Garcia-Salguero, J. Briales, and J. Gonzalez-Jimenez. "Certifiable relative pose estimation". *Image and Vision Computing* 109 (2021), p. 104142.

[49] L. Sun and Z. Deng. "Certifiably Optimal and Robust Camera Pose Estimation from Points and Lines". *IEEE Access* (2020).

[50] C. Holmes, F. Dümbgen, and T. D. Barfoot. *On Semidefinite Relaxations for Matrix-Weighted State-Estimation Problems in Robotics*. 2024. arXiv: 2308.07275 [cs.RO].

[51] H. Yang. *Semidefinite optimization and relaxation*. Apr. 2024. URL: https://hankyang.seas.harvard.edu/Semidefinite/.

[52] J. B. Lasserre. "Global optimization with polynomials and the problem of moments". *SIAM J. Optim.* 11.3 (2001), pp. 796–817.

[53] A. N. Angelopoulos, R. F. Barber, and S. Bates. *Theoretical Foundations of Conformal Prediction*. Nov. 18, 2024. DOI: 10.48550/arXiv.2411.11824. arXiv: 2411.11824[math]. URL: http://arxiv.org/abs/2411.11824 (visited on 11/27/2024).

[54] A. N. Angelopoulos and S. Bates. "A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification". *arXiv* arXiv:2107.07511 (Sept. 2022).

[55] S. Altmann. *Rotations, Quaternions, and Double Groups*. Dover Books on Mathematics. Dover Publications, 2013. ISBN: 9780486317731. URL: https://books.google.com/books?id=50DDAgAAQBAJ.

[56] S. L. Altmann. "Hamilton, Rodrigues, and the Quaternion Scandal". *Mathematics Magazine* 62.5 (1989), pp. 291–308. ISSN: 0025570X, 19300980. URL: http://www.jstor.org/stable/2689481 (visited on 04/12/2025).

[57] H. Yang and L. Carlone. "A Quaternion-based Certifiably Optimal Solution to the Wahba Problem with Outliers". In: *Intl. Conf. on Computer Vision (ICCV)*. (Oral Presentation, accept rate: 4%), Arxiv version: 1905.12536, (pdf). 2019.

[58] L. Shaikewitz, S. Ubellacker, and L. Carlone. "A Certifiable Algorithm for Simultaneous Shape Estimation and Object Tracking". *IEEE Robotics and Automation Letters (RA-L)* (2024). (pdf), (video), (code).

[59] G. Pavlakos, X. Zhou, A. Chan, K. Derpanis, and K. Daniilidis. "6-DoF Object Pose from Semantic Keypoints". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2017.

[60] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone. "Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection". *IEEE Robotics and Automation Letters (RA-L)* 5.2 (2020). arXiv preprint:1909.08605 (with supplemental material), (pdf), pp. 1127–1134.

[61] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. "Active shape models - their training and application". *Comput. Vis. Image Underst.* 61.1 (Jan. 1995), pp. 38–59.

[62] X. Zhou, S. Leonardos, X. Hu, and K. Daniilidis. "3D shape reconstruction from 2D landmarks: A convex formulation". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[63] R. Tron, D. Rosen, and L. Carlone. "On the Inclusion of Determinant Constraints in Lagrangian Duality for 3D SLAM". In: *Robotics: Science and Systems (RSS), Workshop "The problem of mobile sensors: Setting future goals and indicators of progress for SLAM"*. (pdf). 2015.

[64] J. Briales and J. Gonzalez-Jimenez. "Fast global optimality verification in 3D SLAM". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Oct. 2016, pp. 4630–4636. DOI: 10.1109/IROS.2016.7759681.

[65] A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin. "Rotation averaging and strong duality". *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2018).

[66] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 8.1.* 2017. URL: http://docs.mosek.com/8.1/toolbox/index.html.

[67] J. Shi, H. Yang, and L. Carlone. "Optimal Pose and Shape Estimation for Category-level 3D Object Perception". In: *Robotics: Science and Systems (RSS)*. arXiv preprint: 2104.08383, (pdf), (video). 2021.

[68] D. Ge, Q. Huangfu, Z. Wang, J. Wu, and Y. Ye. *Cardinal Optimizer (COPT) user guide*. https://guide.coap.online/copt/en-doc. 2022.

[69] Y. Xiang, R. Mottaghi, and S. Savarese. "Beyond pascal: A benchmark for 3d object detection in the wild". In: *IEEE Winter Conf. on Appl. of Computer Vision*. IEEE. 2014, pp. 75–82.

[70] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. "The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research". In: *Intl. Conf. on Advanced Robotics (ICAR)*. July 2015, pp. 510–517.

[71] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition" (2016), pp. 770–778.

[72] M. Denninger, D. Winkelbauer, M. Sundermeyer, W. Boerdijk, M. Knauer, K. H. Strobl, M. Humt, and R. Triebel. "BlenderProc2: A Procedural Pipeline for Photorealistic Rendering". *Journal of Open Source Software* 8.82 (2023), p. 4901. DOI: 10.21105/joss.04901. URL: https://doi.org/10.21105/joss.04901.

[73] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes". In: *Robotics: Science and Systems (RSS)*. 2018.

[74] A. Smit. *Heinz Ketchup Bottle*. Oct. 2020. URL: https://grabcad.com/library/heinz-ketchup-bottle-1.

[75] G. Jocher, A. Chaurasia, and J. Qiu. *Ultralytics YOLO*. Version 8.0.0. Jan. 2023. URL: https://github.com/ultralytics/ultralytics.

[76] X. Deng, J. Geng, T. Bretl, Y. Xiang, and D. Fox. "iCaps: Iterative Category-level Object Pose and Shape Estimation". *IEEE Robotics and Automation Letters* (2022).

[77] S. Ubellacker, A. Ray, J. Bern, J. Strader, and L. Carlone. "High-speed aerial grasping using a soft drone with onboard perception". *Nature Robotics* (2024). (pdf),(video),(web).

[78] A. Wächter and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". *Mathematical Programming* 106.1 (Apr. 2005), pp. 25–57. ISSN: 1436-4646. DOI: 10.1007/s10107-004-0559-y. URL: http://dx.doi.org/10.1007/s10107-004-0559-y.

[79] J. Wang, V. Magron, and J.-B. Lasserre. "TSSOS: A Moment-SOS hierarchy that exploits term sparsity". *SIAM Journal on Optimization* 31.1 (2021), pp. 30–58.

[80] I. Pólik and T. Terlaky. "A Survey of the S-Lemma". *SIAM Review* 49.3 (2007), pp. 371–418. DOI: 10.1137/S003614450444614X. eprint: https://doi.org/10.1137/S003614450444614X. URL: https://doi.org/10.1137/S003614450444614X.

[81] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. "Learning 6d object pose estimation using 3d object coordinates". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 536–551.

[82] T. Hodaň et al. "BOP: Benchmark for 6D Object Pose Estimation". In: *European Conf. on Computer Vision (ECCV)*. 2018, pp. 19–35.

[83] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas. "BOP Challenge 2020 on 6D Object Localization". *European Conference on Computer Vision Workshops (ECCVW)* (2020).

[84] B. Tekin, S. N. Sinha, and P. Fua. "Real-time seamless single shot 6d object pose prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 292–301.

[85] M. Oberweger, M. Rad, and V. Lepetit. "Making deep heatmaps robust to partial occlusions for 3d object pose estimation". In: *European Conf. on Computer Vision (ECCV)*. 2018, pp. 119–134.

[86] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. "Pvnet: Pixel-wise voting network for 6dof pose estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4561–4570.

[87] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. "Complete solution classification for the perspective-three-point problem". *IEEE Trans. Pattern Anal. Machine Intell.* 25.8 (2003), pp. 930–943.

[88] P. Rigollet and J.-C. Hütter. *High-Dimensional Statistics*. 2023. arXiv: 2310.19244 [math.ST]. URL: https://arxiv.org/abs/2310.19244.

[89] R. M. Martin. *Electronic Structure: Basic Theory and Practical Methods*. Cambridge University Press, 2004.

[90] P. Lindstrom and P. Wedino. *Gauss-Newton based algorithms for constrained nonlinear least squares problems*. Tech. rep. UMINF-901.87. Institute of Information Processing, University of Umea, Sweden, 1988.

[91] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. "Manopt, a Matlab Toolbox for Optimization on Manifolds". *Journal of Machine Learning Research* 15.42 (2014), pp. 1455–1459. URL: https://www.manopt.org.

[92] M. Fischler and R. Bolles. "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography". *Commun. ACM* 24 (1981), pp. 381–395.

[93] Y. Cai, L.-H. Zhang, Z. Bai, and R.-C. Li. "On an Eigenvector-Dependent Nonlinear Eigenvalue Problem". *SIAM Journal on Matrix Analysis and Applications* 39.3 (Jan. 2018), pp. 1360–1382. ISSN: 0895-4798, 1095-7162. DOI: 10.1137/17M115935X. URL: https://epubs.siam.org/doi/10.1137/17M115935X (visited on 03/28/2025).

[94] L.-H. Zhang, W. H. Yang, C. Shen, and J. Ying. "An Eigenvalue-Based Method for the Unbalanced Procrustes Problem". *SIAM Journal on Matrix Analysis and Applications* 41.3 (Jan. 2020), pp. 957–983. ISSN: 0895-4798, 1095-7162. DOI: 10.1137/19M1270872. URL: https://epubs.siam.org/doi/10.1137/19M1270872 (visited on 03/08/2025).

[95] R.-C. Li. *A Theory of the NEPv Approach for Optimization On the Stiefel Manifold*. Oct. 19, 2024. DOI: 10.48550/arXiv.2305.00091. arXiv: 2305.00091[math]. URL: http://arxiv.org/abs/2305.00091 (visited on 03/14/2025).

[96] G. Terzakis and M. Lourakis. "A consistently fast and globally optimal solution to the perspective-n-point problem". In: *European Conf. on Computer Vision (ECCV)*. Springer. 2020, pp. 478–494.