2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Category-Level Object Shape and Pose Estimation in Less Than a Millisecond

Lorenzo Shaikewitz[1], Tim Nguyen[2], and Luca Carlone[1]

***Abstract*— Object shape and pose estimation is a foundational robotics problem, supporting tasks from manipulation to scene understanding and navigation. We present a fast local solver for shape and pose estimation which requires only category-level object priors and admits an efficient *certificate* of global optimality. Given an RGB-D image of an object, we use a learned front-end to detect sparse, category-level semantic *keypoints* on the target object. We represent the target object's unknown shape using a linear *active shape model* and pose a maximum a posteriori optimization problem to solve for position, orientation, and shape simultaneously. Expressed in unit quaternions, this problem admits first-order optimality conditions in the form of an eigenvalue problem with eigenvector nonlinearities. Our primary contribution is to solve this problem efficiently with *self-consistent field iteration*, which only requires computing a 4×4 matrix and finding its minimum eigenvalue-vector pair at each iterate. Solving a linear system for the corresponding Lagrange multipliers gives a simple global optimality certificate. One iteration of our solver runs in about 100 microseconds, enabling fast outlier rejection. We test our method on synthetic data and a variety of real-world settings, including two public datasets and a drone tracking scenario.**

I. Introduction

A diverse set of robotics applications benefits from object shape and pose estimation. Autonomous cars, for example, need to locate obstacles and other cars [1], while household manipulators need to locate objects to interact with [2]. In many of these applications the object shape is not known exactly but its *category* is available (*e.g.,* from a semantic segmentation method). We consider this setting and derive a shape and pose estimator using category-level priors.

The work of Shi et al. [3] established a certifiably optimal approach for category-level shape and pose estimation using a semidefinite relaxation. We consider a similar setup but emphasize both *speed* and *certifiability*. A fast estimator allows quick reaction to new inputs, performance with limited compute, and comprehensive outlier rejection [4], [5]. Certifiability provides an *a posteriori* guarantee that the estimate returned is statistically optimal. When the certificate fails, the user can decide to trust the output, try a different initialization, or acquire a new batch of measurements.

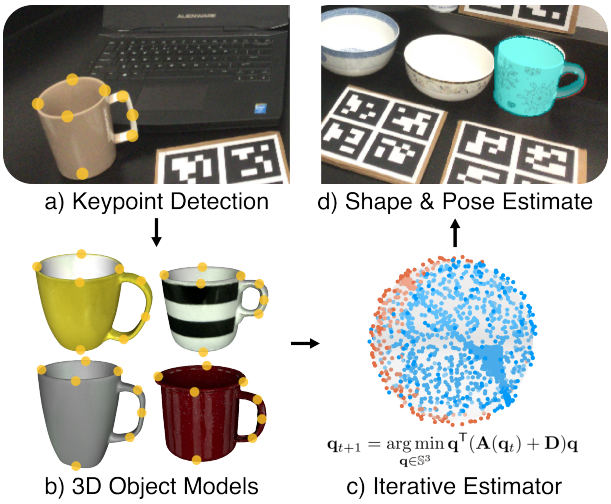

Fig. 1: Given 3D keypoint detections (a) on an RGB-D image, and a category-level shape library (b), we use self-consistent field iteration (c), to estimate an object's shape and pose (d).

Our algorithm relies on the eigenvalue structure of the first-order optimality conditions written in the quaternion representation of rotations. It returns local solutions which are often globally optimal. To verify this, we introduce a fast global optimality certifier based on Lagrangian duality. Specifically, our contributions are:

- A fast local solver for category-level shape and pose estimation using self-consistent field iteration [6].
- A fast *a posteriori* certificate of global optimality for our local solutions.
- Experimental evaluation of runtime and accuracy on synthetic data, a drone tracking scenario, and two large-scale datasets.

The remainder of the paper is organized as follows. We begin with a literature review (Section II) and quaternion preliminaries (Section III). Then, we describe the problem in Section IV and reformulate it with quaternions in Section V. To solve the quaternion form, we use self-consistent field iteration for local solutions and SDP

This work was supported by the AFOSR "Certifiable and Self-Supervised Category-Level Tracking" program, Carlone's NSF CAREER award, and the ONR RAPID program. L. Shaikewitz is supported by an NSF graduate research fellowship. L. Carlone holds concurrent appointments at MIT and as an Amazon Scholar. This paper describes work performed at MIT and is not associated with Amazon.

[1]L. Shaikewitz and L. Carlone are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA. Emails: {lorenzos, lcarlone}@mit.edu.

[1]T. Nguyen is with Boston University, Boston, MA. Work was completed at MIT. Email: timnguyen737@gmail.com.

optimality conditions to certify global optimality in Section VI. Section VII shows our method is significantly faster than other local solvers and learned baselines.

II. Related Work

Solvers for Rotation Estimation Problems. Rotation estimation encompasses a large class of non-convex problems beyond pose estimation. When posed in a least-squares form, they can be solved with Gauss-Newton [7] or Levenberg-Marquardt [8], which approximate Newton's method by linearizing the residuals. A more specialized set of approaches use the manifold structure of $\mathrm{SO}(3)$ and Riemannian counterparts to Gauss-Newton or gradient descent [9], [10], [11], [12]. These approaches generate *local* solutions. Global optimality is largely achieved via a posteriori optimality *certificates* using Shor's semidefinite relaxation [13] or the Burer-Monteiro method [14]. This approach has been applied to a variety of rotation estimation problems including rotation averaging [15], pose graph optimization [16], and pose estimation [3], [17].

A notable exception to these paradigms is point cloud registration, which can be solved globally via eigenvalue decomposition [18], [19]. In this paper, we show the shape and pose estimation problem admits a similar (albeit nonlinear) eigenvalue structure, and leverage that structure to construct a fast certifiable solver.

Category-Level Shape and Pose Estimation. Category-level shape and pose estimation largely reduces to correspondence and alignment stages. At the most explicit, [3] uses a learned keypoint detector [20] to estimate pixel-wise correspondences and optimize for the maximum likelihood object shape and pose. Other methods use local features [21], [22] or semantic features [23] to build and align a shape estimate. To avoid explicit features, normalized object coordinates [24] compute pixel-wise correspondences in a normalized frame and then regress for object pose [24], [25], [26]. We refer the reader to [27] for a comprehensive survey. In this paper, we adopt the problem setup from [3], using a keypoint detector and focusing on the alignment stage.

III. Preliminaries on Quaternion Arithmetic

In this section, we review quaternion arithmetic for rigid rotations (for more detail, see [28]). A rotation about axis $\boldsymbol{\omega} \in \mathbb{R}^3$ by angle $\theta \in \mathbb{R}$ admits the following representation as a unit quaternion $\mathbf{q}$:

$$\mathbf{q} = \begin{bmatrix} \cos(\theta/2) \\ \boldsymbol{\omega} \sin(\theta/2) \end{bmatrix}. \tag{1}$$

From this definition, it is clear that negating the last three elements (the *vector part* of the quaternion) gives the inverse rotation: $\mathbf{q}^{-1} = [q_1, -\mathbf{q}_v^{\mathsf{T}}]^{\mathsf{T}}$. The first element q_1 is called the *scalar part*. We also observe that there are two unit quaternions for every rigid rotation: $-\mathbf{q}$ and $\mathbf{q}$ represent the same rotation. To rotate a point $\mathbf{y} \in \mathbb{R}^3$, we use a *quaternion product* $\circ$. Specifically,

$$\mathbf{q} \circ \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \circ \mathbf{q}^{-1} = \begin{bmatrix} 0 \\ \mathbf{R}\mathbf{y} \end{bmatrix}, \tag{2}$$

where $\mathbf{R} \in \mathrm{SO}(3)$ is the rotation matrix corresponding to the quaternion $\mathbf{q}$. Quaternion products may be written as matrix-vector products. For quaternions $\mathbf{a}, \mathbf{b} \in \mathbb{R}^4$,

$$\mathbf{a} \circ \mathbf{b} = \boldsymbol{\Omega}_l(\mathbf{a})\mathbf{b} = \boldsymbol{\Omega}_r(\mathbf{b})\mathbf{a}. \tag{3}$$

In (3), $\boldsymbol{\Omega}_l$ and $\boldsymbol{\Omega}_r$ are the product matrices:

$$\boldsymbol{\Omega}_l(\mathbf{a}) \triangleq \begin{bmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & -a_4 & a_3 \\ a_3 & a_4 & a_1 & -a_2 \\ a_4 & -a_3 & a_2 & a_1 \end{bmatrix}, \boldsymbol{\Omega}_r(\mathbf{a}) \triangleq \begin{bmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & a_4 & -a_3 \\ a_3 & -a_4 & a_1 & a_2 \\ a_4 & a_3 & -a_2 & a_1 \end{bmatrix}. \tag{4}$$

In this paper we allow $\boldsymbol{\Omega}_l$ and $\boldsymbol{\Omega}_r$ to take vectors $\mathbf{y} \in \mathbb{R}^3$ by implicitly homogenizing them with a leading 0. A little algebra shows that a Euclidean inner product involving a rotation matrix $\mathbf{R}$ can be written as a quadratic form in a corresponding quaternion $\mathbf{q}$.

Lemma 1 ([29]): Let the unit quaternion $\mathbf{q} \in \mathbb{S}^3$ represent the same rotation as the matrix $\mathbf{R} \in \mathrm{SO}(3)$. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$ vectors: $\mathbf{x}^{\mathsf{T}}\mathbf{R}\mathbf{y} = -\mathbf{q}^{\mathsf{T}}\boldsymbol{\Omega}_l(\mathbf{x})\boldsymbol{\Omega}_r(\mathbf{y})\mathbf{q}$.

IV. Category-Level Shape and Pose Estimation Problem

Given detections of 3D keypoints on an object of known category, we estimate its shape and pose (position and orientation). This section describes the problem formulation, including our choice of shape representation and measurement model. We adopt the same problem as [3], rephrased here for clarity.

A. Active Shape Model

For each category, we assume a library of K representative 3D point clouds that span the category according to an *active shape model*. For each point $\mathbf{x}_i \in \mathbb{R}^3$ on an arbitrary object in the category, $\mathbf{x}_i$ may be expressed as a linear combination of corresponding points $\mathbf{b}_k^i \in \mathbb{R}^3$ on the objects in the 3D shape library. Mathematically:

$$\mathbf{x}_i = \sum_{k=1}^{K} c_k \mathbf{b}_k^i \triangleq \mathbf{B}_i \mathbf{c}, \tag{5}$$

where $\mathbf{B}_i \in \mathbb{R}^{3 \times K}$ stacks each $\mathbf{b}_k^i$ as columns 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 center of its bottle cap. The active shape model can represent any object in the convex hull of its 3D shape library. For more than two objects this becomes quite expressive; in the bottle example, the active shape model could represent any cap between the shortest and tallest in the library.

B. Measurement Model

Given an object's category, we estimate its shape and pose from a sparse set of 3D *keypoints* $\mathbf{y}_i \in \mathbb{R}^3$, $i = 1, ..., N$, with known associations to a point in each of the library shapes $\mathbf{B}_i$. These measurements may come from pixel detections by a learned keypoint detector [30] combined with depth information, and are typically semantically meaningful (see Fig. 1).

Let $\mathbf{c}$ be the object's shape vector. Denoting the object's position $\mathbf{p} \in \mathbb{R}^3$ and orientation $\mathbf{R} \in \mathrm{SO}(3)$ 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:

$$\mathbf{y}_i = \mathbf{R}\mathbf{B}_i\mathbf{c} + \mathbf{p} + \boldsymbol{\epsilon}_i. \tag{6}$$

Eq. (6) models the keypoint measurement as a linear combination of shape library points that is rotated and translated before being perturbed by measurement noise $\boldsymbol{\epsilon}_i$. We assume the measurement noise follows an isotropic Gaussian distribution with zero mean and known covariance: $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, w_i^{-1}\mathbf{I}_3)$. Our goal is to estimate the object pose and shape vector from noisy measurements.

Problem 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.

V. Nonlinear Eigenproblem for Local Solutions

In this section, we rephrase Problem 1 as a non-convex optimization problem and develop its first-order optimality conditions into a nonlinear eigenvalue problem. We recall from [3] that the optimal position and shape solving Problem 1 can be computed in closed form, leading to a rotation-only maximum a posteriori (MAP) estimation problem. While [3] solves this non-convex problem with a semidefinite relaxation, we re-express the rotation estimation problem with quaternions and show its first-order optimality conditions form a nonlinear eigenproblem. Our solver exploits this structure.

Under model (6) with Gaussian noise and a shape prior $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I}_K)$, the MAP estimator solving Problem 1 takes the following form [3]:

$$\begin{aligned} \min_{\substack{\mathbf{R}\in\mathrm{SO}(3)\\ \mathbf{p}\in\mathbb{R}^3,\,\mathbf{c}\in\mathbb{R}^K}} \; & \sum_{i=1}^{N} w_i \|\mathbf{y}_i - \mathbf{R}\mathbf{B}_i\mathbf{c} - \mathbf{p}\|^2 + \lambda\|\mathbf{c}\|^2 \\ \text{s.t.} \quad & \mathbf{1}^\mathsf{T}\mathbf{c} = 1,\ \mathbf{c} \in [0,1]^K. \end{aligned} \tag{7}$$

Following [3], we drop the constraint $\mathbf{c} \in [0,1]^K$ for the remainder of the paper. While (7) has a convex quadratic objective, the constraint $\mathbf{R} \in \mathrm{SO}(3)$ introduces non-convexity [16]. Fortunately, (7) is convex in $\mathbf{p}$ and $\mathbf{c}$ as a function of $\mathbf{R}$. This allows analytic elimination of the shape and position variables via the first-order optimality conditions, summarized below.

Proposition 2 (Optimal Shape and Position [3]): Given a rotation estimate $\mathbf{R}$ and shape vector $\mathbf{c}$, the optimal position solving (7) is:

$$\mathbf{p}^\star(\mathbf{R}, \mathbf{c}) = \bar{\mathbf{y}} - \mathbf{R}\bar{\mathbf{B}}\mathbf{c}, \tag{8}$$

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

$$\bar{\mathbf{y}} \triangleq \frac{\sum_{i=1}^N w_i\mathbf{y}_i}{\sum_{i=1}^N w_i} \quad \text{and} \quad \bar{\mathbf{B}} \triangleq \frac{\sum_{i=1}^N w_i\mathbf{B}_i}{\sum_{i=1}^N w_i}. \tag{9}$$

The optimal shape vector solving (7) can be recovered from a rotation estimate:

$$\mathbf{c}^\star(\mathbf{R}) = \mathbf{C}_1 \sum_{i=1}^{N} \left(\bar{\mathbf{B}}_i^\mathsf{T}\mathbf{R}^\mathsf{T}\bar{\mathbf{y}}_i\right) + \mathbf{c}_2, \tag{10}$$

where we use the following symbols:

$$\begin{aligned} \hat{\mathbf{B}}^2 &\triangleq \textstyle\sum_{i=1}^N \bar{\mathbf{B}}_i^\mathsf{T}\bar{\mathbf{B}}_i, \\ \mathbf{H} &\triangleq \hat{\mathbf{B}}^2 + \lambda\mathbf{I}_K, \\ \mathbf{C}_1 &\triangleq \mathbf{H}^{-1} - \mathbf{H}^{-1}\mathbf{1}_K\left(\mathbf{1}_K^\mathsf{T}\mathbf{H}^{-1}\mathbf{1}_K\right)^{-1}\mathbf{1}_K^\mathsf{T}\mathbf{H}^{-1}, \\ \mathbf{c}_2 &\triangleq \mathbf{H}^{-1}\mathbf{1}_K\left(\mathbf{1}_K^\mathsf{T}\mathbf{H}^{-1}\mathbf{1}_K\right)^{-1}. \end{aligned} \tag{11}$$

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

Substituting the optimal position (8) and shape (10) into (7), we rephrase object shape and pose estimation as a rotation-only estimation problem.

Problem 2: Estimate the object's MAP rotation:

$$\min_{\mathbf{R}\in\mathrm{SO}(3)} \sum_{i=1}^{N} \left\|\bar{\mathbf{y}}_i - \mathbf{R}\bar{\mathbf{B}}_i\mathbf{c}^\star(\mathbf{R})\right\|^2 + \lambda\|\mathbf{c}^\star(\mathbf{R})\|^2. \tag{12}$$

Note that (12) is an optimization problem over only a single rotation with no constraints beyond $\mathbf{R} \in \mathrm{SO}(3)$. Although it is not immediately clear from the formulation, it is also *quadratic* in the unknown matrix $\mathbf{R}$. In the next section we rewrite (12) as a quartic problem for the quaternion rotation representation and derive its first-order optimality conditions.

A. First-Order Conditions in Terms of Quaternions

Here we depart from [3] to derive a fast local solver. Instead of vectoring the rotation matrix $\mathbf{R}$, we rewrite the problem in terms of the unit quaternion rotation representation. The quaternion formulation has a quartic objective and a quadratic equality constraint, leading to a nonlinear eigenproblem for first-order stationary points.

We begin by expanding the objective of (12). Grouping terms by $\mathbf{s} \triangleq \sum_{j=1}^{N} \bar{\mathbf{B}}_j^{\mathsf{T}} \mathbf{R}^{\mathsf{T}} \bar{\mathbf{y}}_j$, we obtain a quadratic:

$$\begin{aligned} \min_{\mathbf{R}\in\mathrm{SO}(3)} \sum_{i=1}^{N} (\bar{\mathbf{y}}_i^{\mathsf{T}} \bar{\mathbf{y}}_i) + \mathbf{c}_2^{\mathsf{T}} \hat{\mathbf{B}}^2 \mathbf{c}_2 + \lambda \mathbf{c}_2^{\mathsf{T}} \mathbf{c}_2 \\ + 2\mathbf{s}^T \left(-\mathbf{I}_3 + \mathbf{C}_1 \hat{\mathbf{B}}^2 + \lambda \mathbf{C}_1 \right) \mathbf{c}_2 \\ + \mathbf{s}^{\mathsf{T}} \left(-2\mathbf{I}_3 + \mathbf{C}_1 \hat{\mathbf{B}}^2 + \lambda \mathbf{C}_1 \right) \mathbf{C}_1 \mathbf{s}. \end{aligned} \tag{13}$$

Now, we rewrite (13) in terms of a unit quaternion $\mathbf{q} \in \mathbb{S}^3$ which represents the same orientation as $\mathbf{R}$. Using Lemma 1 and dropping the constant terms in (13), we arrive at the following optimization problem:

$$\min_{\mathbf{q}\in\mathbb{S}^3} \mathbf{q}^{\mathsf{T}} (2\mathbf{D} + \mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}}))\mathbf{q}, \tag{14}$$

where

$$\mathbf{C}_\delta \triangleq \mathbf{I}_3 - \mathbf{C}_1 \hat{\mathbf{B}}^2 - \lambda \mathbf{C}_1, \tag{15}$$

$$\mathbf{D} \triangleq \sum_{i=1}^{N} \mathbf{\Omega}_l(\bar{\mathbf{y}}_i) \mathbf{\Omega}_r(\bar{\mathbf{B}}_i \mathbf{C}_\delta \mathbf{c}_2), \tag{16}$$

$$\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}}) \triangleq \sum_{i=1}^{N} \mathbf{\Omega}_l(\bar{\mathbf{y}}_i) \mathbf{\Omega}_r[\bar{\mathbf{B}}_i (\mathbf{I}_3 + \mathbf{C}_\delta) \mathbf{C}_1 \mathbf{s}(\mathbf{q})]. \tag{17}$$

The matrix $\mathbf{D}$ rewrites the linear term in (13) into a quadratic quaternion form using Lemma 1 and absorbs the negative into $\mathbf{C}_\delta$. In contrast, $\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}})$ uses Lemma 1 on only the first rotation. We write $\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}})$ to emphasize its remaining quadratic dependence on $\mathbf{q}$.

We now write the first-order optimality conditions for (14), which are necessary but not sufficient for globally optimal solutions. Introducing the dual variable μ for the constraint $\mathbf{q} \in \mathbb{S}^3$, the Lagrangian is:

$$\mathcal{L}(\mathbf{q}, \mu) = \mathbf{q}^{\mathsf{T}} (2\mathbf{D} + \mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}}))\mathbf{q} + \mu(\mathbf{q}^{\mathsf{T}}\mathbf{q} - 1). \tag{18}$$

The first-order conditions are $\nabla_{\mathbf{q}} \mathcal{L} = \mathbf{0}$. We differentiate each term with the product rule, noting $\mathbf{D}$ and $\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}})$ are symmetric. The $\mathbf{q}^{\mathsf{T}}\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}})\mathbf{q}$ term requires some care. Numbering each quaternion, (17) gives:

$$\mathbf{q}_1^{\mathsf{T}} \mathbf{A}(\mathbf{q}_2 \mathbf{q}_3^{\mathsf{T}}) \mathbf{q}_4 = \mathbf{q}_2^{\mathsf{T}} \mathbf{A}(\mathbf{q}_1 \mathbf{q}_4^{\mathsf{T}}) \mathbf{q}_3. \tag{19}$$

Thus, the derivative of $\mathbf{q}^{\mathsf{T}}\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}})\mathbf{q}$ term yields four copies. The first-order conditions for (14) are:

$$\mathbf{0} = \nabla_{\mathbf{q}} \mathcal{L}(\mathbf{q}, \mu) = 4\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}})\mathbf{q} + 4\mathbf{D}\mathbf{q} - 2\mu\mathbf{q}. \tag{20}$$

Eq. (20) resembles an eigenvalue problem, with one summand $\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}})$ having eigenvector dependence.

Proposition 3 (Eigenproblem for Local Solutions): All local minima[1] $\mathbf{q}$ of (12) satisfy the following nonlinear eigenproblem for some $\mu \in \mathbb{R}$:

$$\left(\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}}) + \mathbf{D}\right) \mathbf{q} = \mu\mathbf{q}. \tag{21}$$

[1]Notice that both $+\mathbf{q}$ and $-\mathbf{q}$ are valid eigenvectors, consistent with the double coverage property of quaternions.

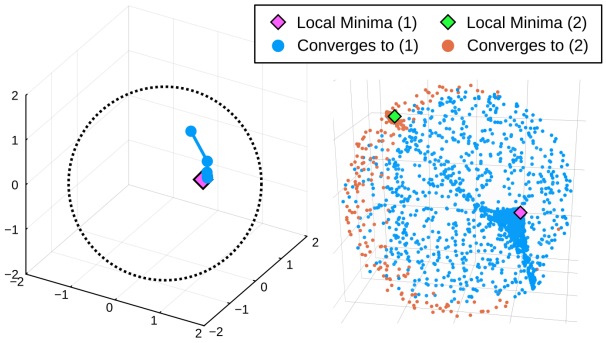

Fig. 2: Stereographic projections of self-consistent field iterates. Beginning from a unit quaternion $\mathbf{q}_0 \in \mathbb{S}^3$, SCF rapidly converges to a local stationary point. Left, a single SCF trajectory. Right, unit quaternion iterates stereographically projected into the volume of the 3-dimensional unit ball and colored by which of the two local minima 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 ($\sigma_m = 5$).

While it is not immediately clear how to solve (21), the weak dependence on $\mathbf{q}$ suggests standard numerical eigenvalue solvers such as power iteration [31] may yield good solutions. Similar eigenproblems have been studied in math and physics [6], [32], [33], [34], but have found less applications in robotics so far. In Section VI, we apply these results to develop a fast iterative solver that only requires computing $\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}}) \in \mathbb{R}^4$ and its smallest eigenvalue-eigenvector pair at each iteration.

VI. Iterative Method for Fast Shape and Pose Estimates

In this section, we propose a fast solution strategy for (21) using self-consistent field iteration [6], [33], [34]. At each iterate, the dominant computational cost is computing the smallest eigenvector-eigenvalue pair of a 4×4 matrix. While the solutions are only guaranteed to be local stationary points, we also present a fast certificate of global optimality based on Shor's relaxation [13].

A. Self-Consistent Field Iteration

We use self-consistent field (SCF) iteration to solve the nonlinear eigenproblem (21). Starting from an initial guess, SCF computes the data matrix $\mathbf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}}) + \mathbf{D}$ and updates $\mathbf{q}$ to one of its normalized eigenvectors. The algorithm terminates when it converges to a stationary point—a unit vector $\mathbf{q}$ which exactly satisfies (21). In practice, we terminate by a numerical tolerance on the angle between consecutive iterates.

The full algorithm is given in Algorithm 1 and illustrated in Fig. 2. At each iterate, 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

Data: $\mathbf{A}(\mathbf{q}\mathbf{q}^\mathsf{T})$ from (17) and $\mathbf{D}$ from (16)
Result: $\mathbf{q}$ satisfying (21)
initialize $\mathbf{q}_0 \in \mathbb{S}^3$
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 1: SCF iteration for local solutions to (14).

saddle point or local maxima) since the objective at stationary points is dominated by the eigenvalue. To see this, notice that the objective of (14) can be written as $\mathbf{q}^\mathsf{T}\left(\mathbf{A}(\mathbf{q}\mathbf{q}^\mathsf{T})+\mathbf{D}\right)\mathbf{q} + \mathbf{q}^\mathsf{T}\mathbf{D}\mathbf{q}$. At a stationary point, (21) implies the first term is just the eigenvalue. That is, $f_{\text{local}} = \mu + \mathbf{q}^\mathsf{T}\mathbf{D}\mathbf{q}$. Thus, the minimum eigenvalue is a good guess for the local minimum. It also has strong computational benefits. In particular, we observe the minimum eigenvalue (guaranteed to be non-positive) is often the eigenvalue with largest magnitude at optimality. This property enables fast convergence similar to power iteration [31], often in less than 5 iterations.

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. For 10 keypoints, these steps take less than 10 μs on a single CPU thread. Although we do not do so here, starting with different initial conditions could be easily parallelized across GPU resources. In Section VII we show the entire algorithm takes about 100 μs.

B. SDP KKT Conditions for Global Optimality

While the previous sections focused on local solutions to Problem 2, we now develop a tool to *certify* the global optimality of a local solution. Certification is essential for reliability; a certificate guarantees Algorithm 1 converged to a statistically optimal estimate.

To certify a local solution of (14), we check if it is a KKT point of the problem's convex semidefinite program (SDP) relaxation. To avoid a second-order relaxation, we use the rotation matrix form (12). Our certificate relies on dual variables, but the matrix form of the $\mathrm{SO}(3)$ constraint does not satisfy the linear independence constraint qualification (necessary for unique duals [35]). Thus, we relax (12) to an *orthogonal* matrix constraint [36]:

$$\min_{\mathbf{R}\in\mathrm{O}(3)} \sum_{i=1}^{N} \left\|\bar{\mathbf{y}}_i - \mathbf{R}\bar{\mathbf{B}}_i\mathbf{c}^\star(\mathbf{R})\right\|^2 + \lambda\|\mathbf{c}^\star(\mathbf{R})\|^2. \quad (22)$$

	QCQP (23)	SDP (24)
Stationarity	$\mathbf{S}\mathbf{x} = \mathbf{0}$	$\mathbf{S} = \mathbf{C} - \sum_{i=1}^{7}\lambda_i\mathbf{A}_i$
Slackness		$\langle \mathbf{S}, \mathbf{X}\rangle = 0$
Primal feas.	$\langle \mathbf{A}_i, \mathbf{x}\mathbf{x}^\mathsf{T}\rangle = b_i$	$\langle \mathbf{A}_i, \mathbf{X}\rangle = b_i$
Dual feas.		$\mathbf{S} \succeq 0$

TABLE I: KKT optimality conditions for the QCQP (23) and its SDP relaxation (24).

Noting that (22) is *quadratic* in $\mathbf{R}$, we define the homogeneous variable $\mathbf{x} \triangleq [1, \mathrm{vec}(\mathbf{R})^\mathsf{T}]^\mathsf{T}$. The $\mathrm{vec}(\cdot)$ operator stacks the columns of its argument into a vector. In standard form, (22) is:

$$\min_{\mathbf{x}\in\mathbb{R}^{10}} \mathbf{x}^\mathsf{T}\mathbf{C}\mathbf{x} \quad \text{s.t. } \mathbf{x}^\mathsf{T}\mathbf{A}_i\mathbf{x} = b_i,\ i = 1, ..., 7, \quad (23)$$

for the constraints $x_1 = 1$ and orthogonality, $\mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{I}_3$.

Now, we further relax (23) to an SDP using Shor's relaxation [13]. Observe $\mathbf{x}^\mathsf{T}\mathbf{C}\mathbf{x} = \langle \mathbf{C}, \mathbf{x}\mathbf{x}^\mathsf{T}\rangle$. Relaxing $\mathbf{x}\mathbf{x}^\mathsf{T}$ to a matrix $\mathbf{X} \in \mathbb{S}^{10}$, we arrive at an SDP:

$$\min_{\mathbf{X}\succeq 0} \langle \mathbf{C}, \mathbf{X}\rangle \quad \text{s.t. } \langle \mathbf{A}_i, \mathbf{X}\rangle = b_i,\ i = 1, ..., 7. \quad (24)$$

The KKT conditions for (23) and (24) are given in Table I [37]. For a local solution $\mathbf{x}_l$ satisfying the QCQP KKT conditions, the candidate SDP solution $\mathbf{X} = \mathbf{x}_l\mathbf{x}_l^\mathsf{T}$ satisfies primal feasibility and complementarity slackness from QCQP stationarity. Thus, we use the dual feasibility condition as a global optimality certificate. To compute $\mathbf{S}$, we only need the Lagrange multipliers $\boldsymbol{\lambda}$. From QCQP stationarity, this is a linear system:

$$\sum_{i=1}^{7} \lambda_i\mathbf{A}_i\mathbf{x} = \mathbf{C}\mathbf{x}. \quad (25)$$

Thus, to certify optimality of a local solution $\mathbf{x}$ we solve the linear system (25) for $\boldsymbol{\lambda}$ and check if $\mathbf{S} \succeq 0$.

VII. Experiments

In this section, we evaluate the computational speed and estimation accuracy of our approach against other category-level shape and pose estimators. In Section VII-A we demonstrate that self-consistent field iteration (SCF) produces good local solutions when our Gaussian noise priors are satisfied (*i.e.,* no outliers). Next, Section VII-B extends this evaluation to the real-world drone tracking scenario in [17], where we use graduated non-convexity for outlier robustness [4]. Lastly, we compare against learning-based methods on the NOCS-REAL275 dataset [24] (Section VII-C) and the Apollo-Car3D dataset [38]. These experiments are summarized in Fig. 3. All benchmarks run on a single CPU thread with clock speed 4.2 GHz.

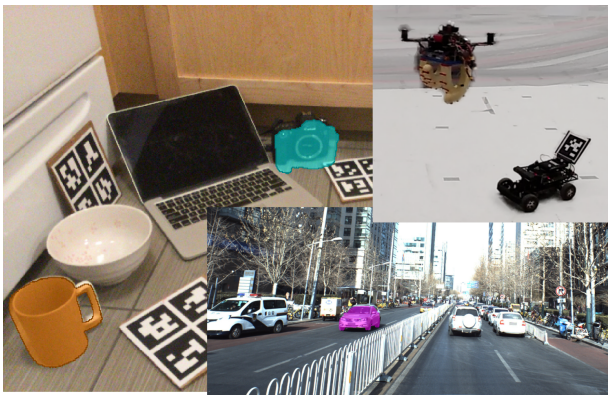

Fig. 3: Overview of experiments. We test on a variety of datasets and synthetic data (not pictured). Left, NOCS-REAL275 [24] contains common household categories including mugs and cameras. Upper right, the CAST drone dataset [17] includes pictures from an aerial quadcopter following a small racecar. Lower right, ApolloCar3D [38] has real-world autonomous driving. Sample pose estimates are highlighted in color.

A. Empirical Performance in Synthetic Dataset

We begin by evaluating SCF in a synthetic environment, where measurements are generated according to the generative model (6) with known Gaussian noise. Specifically, we generate a mean shape with $N = 10$ points drawn from a standard normal. The shape library $\mathbf{B}$ adds zero-mean Gaussian noise to each point with fixed standard deviation $r = 0.2$ m. We generate the ground truth shape by normalizing a $K = 4$ dimensional vector uniformly random in $[0, 1]^K$. The ground truth position is drawn from a standard normal with mean 1 and ground truth rotation is randomly sampled from $\mathrm{SO}(3)$. We set $\lambda = 0$. Results are reported in terms of a normalized measurement standard deviation $\sigma_m \triangleq r/w$, with $w \triangleq w_1 = \ldots = w_N$ the same for all keypoints.

Baselines. We compare against other solution strategies for (12). Manopt [9] is an off-the-shelf local solver for unconstrained manifold optimization. Gauss–Newton (G-N) [7] and Levenberg–Marquardt (L-M) [8] are local solvers with optimized implementations and analytic Jacobians for (12). Lastly, PACE* [3] uses a semidefinite relaxation to find and certify global solutions to (12). We denote by SCF* our approach (SCF) with optimality certificate checking. All methods are implemented in Julia and runtimes do not include precompilation time. For each measurement noise value, each method solves the same 10,000 problems with the same initial guess.

Computational Speed. Table II compares the mean and 90th percentile (p90) of runtimes for each method at small and large noise scales. SCF is by far the fastest method, and more than twice as fast as G-N or L-M. Certificate checking for SCF* adds only a small computational penalty.

Estimation Performance. We also compare the rotation error distributions of G-N, SCF, SCF*, and PACE* in Fig. 4 (we omit the other methods for clarity). G-N, SCF, and PACE* exhibit similar performance across noise scales, with the local methods (G-N and SCF) having slightly more outliers throughout. In view of Table II, we conclude SCF is much faster without sacrificing accuracy. For this plot only, SCF* and PACE* results show only *globally optimal* estimates. For PACE*, all estimates reported a global optimality certificate (using tolerance 10^{-4}). SCF* certified global optimality for 62%, 60%, 55%, 45%, and 19% of estimates at noise scales 0.25, 0.75, 1.5, 2.5, and 5.0, respectively. SCF* reports fewer certificates because of the relaxation from $\mathrm{SO}(3)$ to $\mathrm{O}(3)$, but the certified estimates are, on average, more accurate.

	$\sigma_m = 0.25$		$\sigma_m = 2.5$	
	Mean (ms)	p90 (ms)	Mean (ms)	p90 (ms)
SCF	**0.104**	**0.117**	**0.108**	**0.126**
G-N	0.217	0.253	0.299	0.366
L-M	0.211	0.244	0.285	0.345
Manopt	1.028	1.155	1.039	1.164
SCF*	0.120	0.126	0.126	0.138
PACE*	1.655	1.601	1.667	1.617

TABLE II: Runtimes on synthetic dataset.

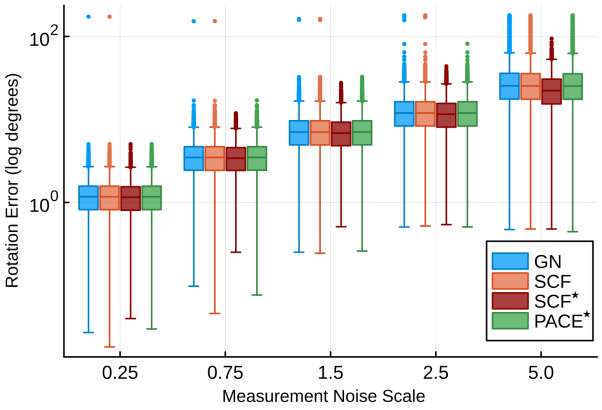

Fig. 4: Distribution of rotation errors for Gauss-Newton, SCF, SCF*, and PACE*. GN, SCF, and PACE have nearly identical performance although SCF runs significantly faster. SCF* and PACE* show only certifiably optimal estimates. SCF* consistently filters out the worst estimates.

B. Performance in Drone Tracking Scenario

We also test SCF in the real-world drone tracking scenario from [17] (CAST). In the CAST dataset, an autonomous quadcopter follows a remote-controlled racecar around a track. There are 1897 RGB-D images taken in sequence from the quadcopter. For this and following experiments, real-world keypoint measurements are corrupted by outliers. Thus, we first run compatibility tests [39] and wrap SCF in graduated non-convexity

	Runtimes		R_{err}	GNC Iters.
	Mean (ms)	p90 (ms)	Mean (deg)	Mean
SCF	**0.456**	**0.774**	9.4	7.1
G-N	1.817	3.480	9.5	7.1
L-M	5.200	12.701	9.5	7.1
Manopt	6.062	11.783	9.5	7.1
SCF*	0.606	1.029	9.4	7.1
PACE*	10.826	21.514	9.5	7.1

TABLE III: Performance on CAST drone dataset [17].

(GNC) [4]. Both use $\bar{\mathbf{c}}^2 = 0.005$. We compare with the same baselines as Section VII-A and use identical outlier rejection. We use the resnet-based [40] keypoint detections ($N = 7$) and shape library ($K = 10$) from [17] with $\lambda = 0.1$.

Results. We report total runtime (which includes GNC and compatibility tests but not keypoint detection), mean rotation error (R_{err}), and mean GNC iterations (number of times the solver is called) in Table III. Recall that Problem 2 is a rotation estimation problem, so rotation error is a good proxy for overall estimation error. As before, SCF is substantially faster than other local approaches and nearly five times faster than G-N. The runtime per iteration is faster than synthetic experiments because of fewer keypoints ($N = 7$). All methods achieve very similar rotation estimation performance and require the same number of GNC iterations.

C. Performance on NOCS-REAL275 Dataset

The NOCS-REAL275 dataset [24] contains real-world RGB-D video sequences of common objects within 6 categories. We test on the camera (2561 frames, $N = 50$, $K = 3$) and mug (2615 frames, $N = 43$, $K = 5$) categories. We drop the other objects due to keypoint detector availability. We use the same YOLOv8 [41] detector as [17], which was trained using synthetically-generated images. We also use the same shape library, which is composed of representative CAD models from 3D scans and BOP datasets [42] (note that these are distinct from the objects in NOCS). We compare against the same set of baselines and the tracking method BundleTrack [21]. BundleTrack numbers are from [21].

Results. In Table IV, we report the mean runtime across categories and accuracy for each category. For accuracy, we report the percentage of estimates within 5° and 5 cm of ground truth ($5^\circ 5\text{cm}$) and the mean orientation error in degrees (R_{err}) and position error in cm (p_{err}). For position and orientation error we exclude measurements with position error above 10 cm [21]. We exclude keypoint detection runtime. For reference, the keypoint detector runs in about 50 ms per image. SCF is significantly faster than other methods, running in just over a millisecond (the majority of this runtime is from compatibility tests). The two-stage methods

	camera			mug			Time
	$5^\circ 5\text{cm}$	R_{err}	p_{err}	$5^\circ 5\text{cm}$	R_{err}	p_{err}	mean (ms)
SCF	7.8	19.5	3.4	24.0	12.7	**1.0**	**1.26**
G-N	7.8	18.4	3.4	23.6	12.8	1.1	1.81
L-M	6.5	21.6	3.4	21.6	12.5	**1.0**	49.1
Manopt	7.8	18.2	3.4	23.5	12.5	**1.0**	2.45
BundleTrack	**85.8**	**3.0**	**2.1**	**99.9**	**1.5**	2.2	100
SCF*	7.8	18.9	3.4	23.4	12.9	**1.0**	1.34
PACE*	7.7	15.0	3.4	20.2	11.9	**1.0**	3.90

TABLE IV: NOCS-REAL275 [24] Performance.

	A3DP-Rel ↑			GNC Iters.	Time
	mean	c-l	c-s	mean	mean (ms)
SCF	17.1	35.7	**28.4**	5.8	**4.4**
G-N	17.1	35.5	**28.4**	5.6	10.3
L-M	17.1	35.7	28.3	5.4	8.0
Manopt	17.1	35.7	**28.4**	6.9	17.7
GSNet	**20.2**	**40.5**	19.9	-	450
SCF*	17.1	35.7	**28.4**	5.8	4.8
PACE*	17.2	35.7	**28.4**	5.3	11.2

TABLE V: ApolloCar3D [24] Performance.

achieve similar performance, although they are significantly worse than BundleTrack except for mug position error. This poor performance is largely due to the low-quality keypoint detector [17].

D. Performance on ApolloCar3D

Lastly, we evaluate SCF on the autonomous driving dataset ApolloCar3D [38]. This dataset has real-world stereo images taken from cars driving in four cities in China. We test on the 200-image validation split and use $K = 79$ car models with $N = 66$ semantic keypoint annotations. For this larger shape library, we set $K = 1.5 \cdot 10^4$. As in [3], we use the keypoint detections from GSNet [20] with stereo depths. For GNC and compatibility tests, we set $\bar{\mathbf{c}}^2 = 0.15$ m. We compare with GSNet [20] and the same set of solvers as in synthetic experiments.

Results. The estimation performance and runtimes are shown in Table V. For estimation, we use the A3DP-Rel metric as defined in [38]. This metric jointly measures translation, rotation, and shape similarity between the estimated and ground truth cars using relative translation thresholds. "Mean" averages across 10 thresholds, while "c-l" reports performance under a loose threshold and "c-s" reports performance under a strict threshold. We also report the mean number of GNC iterations and the mean runtime of each approach. We exclude keypoint detector time and use GSNet results from [20]. SCF is conclusively faster than the other approaches, and significantly outperforms GSNet on the strict (c-

s) criterion. It is not far behind in the mean and loose criteria. The GSNet runtime is difficult to compare with because it includes keypoint detection and was evaluated on different hardware. Although we attempted to exactly replicate [3], our accuracy results are slightly worse.

VIII. Conclusion

In this paper, we revisited the shape and pose estimation problem in [3] with an emphasis on solver speed. Our local solver, based on self-consistent field iteration, can estimate an object's shape and pose in about 100 microseconds. The solver exploits the structure of the quaternion form of the first-order optimality conditions for Problem 2, which are a nonlinear eigenvalue problem for stationary points. We also augmented this local solver with a fast global optimality certificate based on duality with the SDP relaxation. In synthetic and real-world experiments, we demonstrated that our solver was significantly faster than other approaches, with and without global optimality checks. The mixed estimation results suggest more work is needed to develop fast and accurate semantic keypoint detectors.

References

[1] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE Int. J. Trans. Safety*, vol. 4, no. 1, 2016.

[2] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Conference on Robot Learning (CoRL)*, Oct. 2018, pp. 306–316.

[3] 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*, vol. 39, no. 5, pp. 4131–4151, 2023,

[4] 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)*, vol. 5, no. 2, pp. 1127–1134, 2020, arXiv preprint:1909.08605 (with supplemental material),

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

[6] 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*, vol. 39, no. 3, pp. 1360–1382, Jan. 2018, ISSN: 0895-4798, 1095-7162. DOI: 10.1137/17M115935X.

[7] C. Gauss, *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Mabientium [Theory of the Motion of the heavenly Bodies Moving about the Sun in Conic Sections]*. Hamburg, Germany: Perthes and Besser, 1809, English translation available at http://name.umdl.umich.edu/AGG8895.0001.001.

[8] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," *Quart. Appl. Math*, vol. 2, no. 2, pp. 164–168, 1944.

[9] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *Journal of Machine Learning Research*, vol. 15, no. 42, pp. 1455–1459, 2014.

[10] C. Hertzberg, "A framework for sparse, non-linear least squares problems on manifolds," *UNIVERSIT AT BREMEN*, 2008.

[11] J. Chen, Y. Yin, T. Birdal, B. Chen, L. J. Guibas, and H. Wang, "Projective manifold gradient layer for deep rotation regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6646–6655.

[12] S. T. Smith, "Optimization techniques on Riemannian manifolds," *Hamiltonian and Gradient Flows, Algorithms and Control, Fields Inst. Commun., Amer. Math. Soc.*, vol. 3, pp. 113–136, 1994.

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

[14] Burer, Samuel and Monteiro, Renato D C, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003.

[15] 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*, vol. 64, no. 1, pp. 57–67, Oct. 2021, ISSN: 1573-7683. DOI: 10.1007/s10851-021-01054-y.

[16] 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,

[17] 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, , ,

[18] K. Arun, T. Huang, and S. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 5, pp. 698–700, Sep. 1987.

[19] M. Shuster, "Maximum likelihood estimation of spacecraft attitude," *J. Astronautical Sci.*, vol. 37, no. 1, pp. 79–88, Jan. 1989.

[20] L. Ke, S. Li, Y. Sun, Y.-W. Tai, and C.-K. Tang, "GSNet: joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision," in *European Conf. on Computer Vision (ECCV)*, Springer, 2020, pp. 515–532.

[21] 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.

[22] X. Liu, G. Wang, Y. Li, and X. Ji, "Catre: Iterative point clouds alignment for category-level object pose refinement," in *European Conference on Computer Vision*, Springer, 2022, pp. 499–516.

[23] Y. Chen et al., "Secondpose: Se (3)-consistent dual-stream feature fusion for category-level pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9959–9969.

[24] 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.

[25] J. Shi, R. Talak, H. Zhang, D. Jin, and L. Carlone, "CRISP: Object pose and shape estimation with test-time adaptation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[26] 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.

[27] J. Liu et al., "Deep learning-based object pose estimation: A comprehensive survey," *arXiv preprint arXiv:2405.07801*, 2024.

[28] S. Altmann, *Rotations, Quaternions, and Double Groups* (Dover Books on Mathematics). Dover Publications, 2013, ISBN: 9780486317731.

[29] 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, 2019.

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

[31] R. V. Mises and H. Pollaczek-Geiringer, "Praktische verfahren der gleichungsauflösung.," *Journal of Applied Mathematics and Mechanics*, vol. 9, no. 2, pp. 152–164, 1929. DOI: https://doi.org/10.1002/zamm.19290090206.

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

[33] 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*, vol. 41, no. 3, pp. 957–983, Jan. 2020, ISSN: 0895-4798, 1095-7162. DOI: 10.1137/19M1270872.

[34] 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].

[35] A. Papalia, A. Fishberg, B. W. O'Neill, J. P. How, D. M. Rosen, and J. J. Leonard, "Certifiably correct range-aided slam," *IEEE Transactions on Robotics*, 2024.

[36] N. Boumal, *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023. DOI: 10.1017/9781009166164.

[37] A. Papalia, Y. Tian, D. M. Rosen, J. P. How, and J. J. Leonard, "An overview of the burer-monteiro method for certifiable robot perception," *arXiv preprint arXiv:2410.00117*, 2024.

[38] X. Song et al., "ApolloCar3D: A large 3d car instance understanding benchmark for autonomous driving," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5452–5462.

[39] J. Shi, H. Yang, and L. Carlone, "ROBIN: a graph-theoretic approach to reject outliers in robust estimation using invariants," *arXiv preprint: 2011.03659*, 2020,

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

[41] G. Jocher, A. Chaurasia, and J. Qiu, *Ultralytics YOLO*, version 8.0.0, Jan. 2023.

[42] T. Hodaň et al., "BOP challenge 2020 on 6D object localization," *European Conference on Computer Vision Workshops (ECCVW)*, 2020.