Optimal Pose and Shape Estimation for Category-level 3D Object Perception

This paper has been accepted for publication at the 2021 Robotics: Science and Systems Conference.
Please cite the paper as: J. Shi, H. Yang, and L. Carlone,
"Optimal Pose and Shape Estimation for Category-level 3D Object Perception," *Robotics: Science and Systems (RSS)*, 2021.

Optimal Pose and Shape Estimation for Category-level 3D Object Perception

Jingnan Shi, Heng Yang, Luca Carlone
Laboratory for Information & Decision Systems (LIDS)
Massachusetts Institute of Technology
{jnshi, hankyang, lcarlone}@mit.edu

***Abstract*—We consider a *category-level perception* problem, where one is given 3D sensor data picturing an object of a given category (*e.g.,* a car), and has to reconstruct the pose and shape of the object despite intra-class variability (*i.e.,* different car models have different shapes). We consider an *active shape model*, where —for an object category— we are given a library of potential CAD models describing objects in that category, and we adopt a standard formulation where pose and shape estimation are formulated as a non-convex optimization. Our first contribution is to provide the first *certifiably optimal* solver for pose and shape estimation. In particular, we show that rotation estimation can be decoupled from the estimation of the object translation and shape, and we demonstrate that (i) the optimal object rotation can be computed via a tight (small-size) semidefinite relaxation, and (ii) the translation and shape parameters can be computed in closed-form given the rotation. Our second contribution is to add an outlier rejection layer to our solver, hence making it robust to a large number of misdetections. Towards this goal, we wrap our optimal solver in a robust estimation scheme based on *graduated non-convexity*. To further enhance robustness to outliers, we also develop the first graph-theoretic formulation to prune outliers in category-level perception, which removes outliers via convex hull and maximum clique computations; the resulting approach is robust to $70-90\%$ outliers. Our third contribution is an extensive experimental evaluation. Besides providing an ablation study on a simulated dataset and on the PASCAL3D+ dataset, we combine our solver with a deep-learned keypoint detector, and show that the resulting approach improves over the state of the art in vehicle pose estimation in the ApolloScape datasets.**

I. Introduction

Robotics applications, from self-driving cars to domestic robotics, demand robots to be able to identify and estimate the pose and shape of objects in the environment. In self-driving applications, for instance, the perception system needs to estimate the poses of other vehicles in the surroundings, identify traffic lights and traffic signs, and detect pedestrians [27, 35]. Similarly, domestic applications require estimating the location and shape of objects to support more effective interaction and manipulation [23, 48, 56]. Object pose estimation is made harder by the large intra-class shape variability of common objects: for instance, the shape of a car largely varies depending on the model (*e.g.,* take a station wagon versus a Smart).

Despite the fast-paced progress, reliable 3D object pose estimation remains a challenge, as witnessed by recent self-driving car accidents caused by misdetections [51, 70]. Deep learning has been making great strides in enabling robots to detect objects; popular tools such as YOLO [59] and Mask-RCNN [29] have made object detection possible on commodity hardware and with reasonable performance for in-distribution test data. However, detections are typically at the level of categories (*e.g.,* car vs. pedestrian) rather than at the level of instances (*e.g.,* a specific car model) and —with the current methods— enabling instance-level detections would require an unreasonably large amount of labeled data and computation (*e.g.,* to scale to million of potential instances). In turn, category-level perception renders the use of standard tools for pose estimation (from point cloud registration [30, 55, 87] to 2D-3D pose estimation [34, 63, 88]) ineffective, since they rely on the knowledge of the shape of the object.

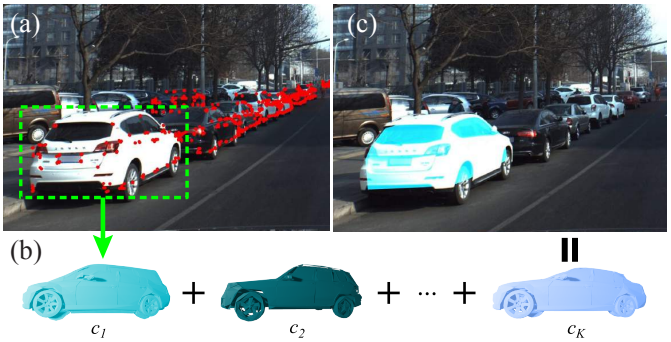


Fig. 1: We propose the first certifiably optimal approach to estimate the 3D pose and shape of objects from 3D keypoint detections (red points in (a)). Our approach estimates pose and shape using an overcomplete library of CAD models (b) and is robust to up to $70-90\%$ outliers in the detections. (c) The approach is more accurate than the state of the art on the ApolloScape dataset [80].

These limitations have triggered robotics and computer vision research on category-level 3D object pose estimation. Traditional methods include the popular *active shape model* [16, 84, 89], where one attempts to estimate the pose and shape of an object given a large database of 3D CAD models. Despite its popularity (*e.g.,* the model is also used in human shape estimation and face detection [89]), pose estimation with active shape models leads to a non-convex optimization problem and local solvers get stuck in poor solutions, and are sensitive to outliers [84, 89]. More recently, research effort has been devoted to end-to-end learning-based 3D pose estimation with encouraging results in human pose estimation [35] and vehicle pose estimation [12, 33, 37, 44, 69]; these approaches still require a large amount of 3D labeled data, which is hard to obtain in the wild.

Contribution. We address the shortcomings of existing approaches for pose and shape estimation based on the active shape model and propose the first approach that can compute optimal pose and shape estimates and is resilient to a large number of outliers. We consider a category-level perception problem, where one is given 3D keypoint detections of an object belonging to a given category (*e.g.,* detections of the wheels, rear-view mirrors, and other interest points of a car), and has to reconstruct the pose and shape of the object despite intra-class variability. We assume the availability of a library of CAD models of objects in that category; such a library is typically available, since CAD models are extensively used in the design, manufacturing, and simulation of 3D objects.

Our first contribution is PACE*, the first *certifiably optimal* solver for 3D-3D pose and shape estimation. In particular, we show that —despite the non-convexity of the problem— rotation estimation can be decoupled from the estimation of object translation and shape, and we demonstrate that (i) the optimal object rotation can be computed via a tight (small-size) semidefinite relaxation, and (ii) the translation and shape parameters can be computed in closed form given the rotation.

Our second contribution is to equip PACE* with an outlier rejection scheme. Towards this goal, we extend existing tools for outlier rejection to category-level perception. In particular, we build on [87] (which assumes the shape to be known) and (i) show how to extend the graph-theoretic outlier pruning in [87] to the case in which the shape is unknown, and (ii) apply a *graduated non-convexity* [86] scheme for robust estimation. The resulting approach is named PACE#.

Our third contribution is an extensive experimental evaluation. We provide an ablation study on a simulated dataset and on the PASCAL3D+ dataset, and show that PACE* is more accurate than iterative solvers, while PACE# dominates other robust solvers and is robust to $70-90\%$ outliers. Finally, we integrate our solver in a realistic system —including a deep-learned keypoint detector— and show that the resulting approach improves over the state of the art in vehicle pose estimation in the ApolloScape [80] driving datasets (Fig. 1).

II. Related Work

Early approaches for category-level perception focus on 2D problems, where one has to locate objects —from human faces [54] to resistors [16]— in images. Classical approaches include *active contour models* [13, 32] and *active shape models* [5, 15, 16]. These works use techniques like PCA to build a library of 2D landmarks from training data, and then use iterative optimization algorithms to estimate the 2D object locations in the images, rather than estimating 3D poses.

The landscape of category-level perception has been recently reshaped by the rapid adoption of **convolutional networks** [36, 40, 66]. Pipelines using deep learning have seen great successes in areas such as human pose estimation [29, 50, 52, 75, 76], and pose estimation of household objects [23, 48, 56]. With the growing interest in self-driving vehicles, research has also focused on jointly estimating vehicle shape and pose [12, 33, 37, 44, 69]. Many open-source driving datasets have also been released for benchmarking [11, 68, 80].

For methods that aim to recover both the 3D shapes and poses of the objects of interests, a common paradigm is to use **end-to-end methods**. Usually, an encoder-decoder network is used to first convert input images to some latent representations, and then use a decoder to map the latent representation back to 3D space [25, 60, 71]. Alternatively, recent work [10] trains CNNs with generative representations of 3D objects to predict probabilistic distribution of object poses. An additional alignment loss can also be incorporated into the network to regress for pose directly [3, 46, 47]. One drawback of such approaches is that it is difficult for neural networks to learn the necessary 3D structure of the object on a per-pixel basis. As shown in [72], such networks can be outperformed by methods trained on model recognition and retrieval only.

Multi-stage methods form another major paradigm for category-level perception. Such approaches first recover the position of semantic keypoints [56] in the images with neural networks, and then recover the 3D pose of the object by solving a geometric optimization problem [31, 53, 56, 57, 64]. In some works, a canonical coordinate space is predicted by a network instead of relying on geometric reasoning [14, 22, 41, 78]. Lim *et al.* [42] establish 2D-3D correspondences between images and textureless CAD models by using HOG descriptors, and render edgemaps of the CAD models. Chabot *et al.* [12] use a two-staged approach to first regress a set of 2D part coordinates, and then choose the best corresponding 3D template and use PnP to solve for the 3D pose. Pavlakos *et al.* [56] use a stacked hourglass neural network [52] for 2D semantic keypoint detection, and then employ block coordinate descent to resolve the object pose. Zhou *et al.* [89, 90] propose a convex relaxation for jointly optimizing 3D shape parameters and object pose from 2D keypoints under a weak perspective camera model. Yang and Carlone [84] apply the moment/sums-of-squares hierarchy [7, 39, 79] to develop tighter relaxations than [89] but lead to semidefinite programs whose size grows with the number of CAD models. Probabilistic guarantees are studied in [81].

Our work belongs to the class of multi-stage methods, but we assume to have access to depth information for the semantic keypoints (*i.e.,* we consider a 3D-3D estimation setup [65]). Depth information is readily available in many robotics problems via direct sensing (*e.g.,* RGB-D or stereo) or algorithms (*e.g.,* mono depth techniques [19, 38]). As we will show in Section IV, the use of depth information allows us to mathematically *decouple* the estimation of rotation from object translation and shape parameters, which leads to the first certifiably optimal solver that runs in a fraction of a second even in the presence of thousands of CAD models.

III. Problem Statement: 3D-3D Category-Level Perception

Active Shape Model. We consider the problem of estimating the 3D pose $(\boldsymbol{R}, \boldsymbol{t})$ and shape of an object, where

$\boldsymbol{R} \in \mathrm{SO}(3)$ and $\boldsymbol{t} \in \mathbb{R}^3$ are the unknown 3D rotation and translation of the object, respectively. We assume the object shape to be partially specified: we are given a library of 3D CAD models $\mathcal{B}_k$, $k = 1, \dots, K$, and assume that the unknown object shape $\mathcal{S}$ (modeled as a collection of 3D points) can be written as a combination of the given CAD models. More formally, each point $\boldsymbol{s}(i)$ of the shape $\mathcal{S}$ can be written as:

$$\boldsymbol{s}(i) = \textstyle\sum_{k=1}^{K} c_k \boldsymbol{b}_k(i) \tag{1}$$

where $\boldsymbol{b}_k(i)$ is a given point belonging to the CAD model $\mathcal{B}_k$; the *shape parameters* $\boldsymbol{c} \triangleq [c_1 \dots c_K]^\mathsf{T}$ are unknown, and the entries of $\boldsymbol{c}$ are assumed to be non-negative and sum up to 1 ($\boldsymbol{c} \geq 0, \sum_{k=1}^{K} c_k = 1$). For instance, if —upon estimation— the vector $\boldsymbol{c}$ has the l-th entry equal to 1 and the remaining entries equal to zero in (1), then the estimated shape of the object matches the l-th CAD model in the library; therefore, the estimation of the shape parameters $\boldsymbol{c}$ can be understood as a fine-grained classification of the object among the instances in the library. However, the model is even more expressive, since it allows the object shape to be a convex combination of CAD models, which enables the active shape model (1) to interpolate between different shapes in the library.

Measurements. Towards the goal of estimating the object pose and shape, we are given a set of N 3D keypoint detections. These are noisy measurements of 3D points belonging to the object and are commonly obtained using learning-based semantic keypoint detectors applied to RGB-D or RGB+Lidar data (*e.g.,* [56]). Each measurement $\boldsymbol{y}(i)$ ($i = 1, \dots, N$) is described by the following generative model:

$$\boldsymbol{y}(i) = \boldsymbol{R} \textstyle\sum_{k=1}^{K} c_k \boldsymbol{b}_k(i) + \boldsymbol{t} + \boldsymbol{\epsilon}(i) \qquad i = 1, \dots, N \tag{2}$$

where the measurement $\boldsymbol{y}(i)$ pictures a 3D point on the object (written as a linear combination $\sum_{k=1}^{K} c_k \boldsymbol{b}_k(i)$ of the shapes in the library as in (1)), after these are rotated and translated according to the 3D pose $(\boldsymbol{R}, \boldsymbol{t})$ of the object, and where $\boldsymbol{\epsilon}(i)$ represents measurement noise. Intuitively, each measurement corresponds to a noisy measurement of a semantic feature of the object (*e.g.,* wheel center or rear-view mirrors of a car) and each $\boldsymbol{b}_k(i)$ corresponds to the feature location for a specific CAD model. We are now ready to state the 3D-3D category-level perception problem.

Problem 1 (3D-3D Category-Level Perception). *Compute the 3D pose $(\boldsymbol{R}, \boldsymbol{t})$ and shape $(\boldsymbol{c})$ of an object given N 3D keypoint measurements in the form (2), possibly corrupted by outliers,* i.e., *measurements with large error $\boldsymbol{\epsilon}(i)$.*

IV. Certifiably Optimal Solver for 3D-3D Category-Level Perception

This section shows how to solve Problem 1 in the outlier-free case, where the noise $\boldsymbol{\epsilon}(i)$ is assumed to follow a zero-mean Gaussian distribution (we generalize the formulation to deal with outliers in Section V). A standard formulation for the pose and shape estimation problem leads to the following *regularized non-linear least squares* problem:

$$\begin{aligned} \min_{\substack{\boldsymbol{R}\in\mathrm{SO}(3),\\ \boldsymbol{t}\in\mathbb{R}^3, \boldsymbol{c}\in\mathbb{R}^K}} \; & \sum_{i=1}^{N} w_i \left\| \boldsymbol{y}(i) - \boldsymbol{R} \sum_{k=1}^{K} c_k \boldsymbol{b}_k(i) - \boldsymbol{t} \right\|^2 + \lambda \left\| \boldsymbol{c} \right\|^2 \\ \text{s.t.} \; & \mathbf{1}^\mathsf{T} \boldsymbol{c} = 1 \end{aligned} \tag{3}$$

where the first summand in the objective minimizes the residual error w.r.t. the generative model (2) ($w_i \geq 0, i = 1, \dots, N$ are given weights), and the second term provides an ℓ_2 regularization (*a.k.a. Tikhonov regularization* [74]) of the shape coefficients $\boldsymbol{c}$ (controlled by the user-specified parameter $\lambda \geq 0$); the constraint $\mathbf{1}^\mathsf{T} \boldsymbol{c} = 1$ (where $\mathbf{1}$ is a vector with all entries equal to 1) forces the shape coefficients to sum-up to 1; in this section, we drop the constraint that $\boldsymbol{c}$ has to be nonnegative for mathematical convenience. Numerically, the regularization term ensures the problem is well-posed regardless of the number of shapes in the library (otherwise, the problem would be under-constrained when K is large). From the probabilistic standpoint, problem (3) is a *maximum a posteriori* estimator assuming that the keypoints measurement noise follows a zero-mean Gaussian with covariance $\frac{1}{w_i}\mathbf{I}_3$ (where $\mathbf{I}_3$ is the 3-by-3 identity matrix) and we have a zero-mean Gaussian prior with covariance $\frac{1}{\lambda}$ over the shape parameters $\boldsymbol{c}$ (proof in Appendix A).

Problem (3) is non-convex due to the product between rotation $\boldsymbol{R}$ and shape parameters $\boldsymbol{c}$ in the objective, and due to the nonconvexity of the constraint set $\mathrm{SO}(3)$ the rotation $\boldsymbol{R}$ is required to belong to, see *e.g.,* [28, 62]. Therefore, existing approaches based on local search [26, 43, 58] are prone to converge to local minima corresponding to incorrect estimates.

Results Overview. The rest of this section provides the first certifiably optimal algorithm to solve Problem (3). Towards this goal we show that (i) the translation $\boldsymbol{t}$ in (3) can be solved in closed form given the rotation and shape parameters (Section IV-A), (ii) the shape parameters $\boldsymbol{c}$ can be solved in closed form given the rotation (Section IV-B), and (iii) the rotation can be estimated (independently on shape and translation) using a tight semidefinite relaxation (Section IV-C). This sequence of results leads to an optimal solver for pose and shape summarized in Section IV-D.

A. Closed-form Translation Estimation

From simple inspection of (3), we observe that the vector $\boldsymbol{t}$ is unconstrained and appears quadratically in the cost function, *i.e.,* from the standpoint of $\boldsymbol{t}$, eq. (3) is a *linear* least squares problem. Therefore, for any choice of $\boldsymbol{R}$ and $\boldsymbol{c}$, the optimal translation can be computed in closed-form as:

$$\boldsymbol{t}^\star(\boldsymbol{R}, \boldsymbol{c}) = \boldsymbol{y}_w - \boldsymbol{R} \textstyle\sum_{k=1}^{K} c_k \boldsymbol{b}_{k,w} \tag{4}$$

where

$$\boldsymbol{y}_w \triangleq \frac{1}{(\sum_{i=1}^{N} w_i)} \sum_{i=1}^{N} w_i \boldsymbol{y}(i), \quad \boldsymbol{b}_{k,w} \triangleq \frac{1}{(\sum_{i=1}^{N} w_i)} \sum_{i=1}^{N} w_i \boldsymbol{b}_k(i), \tag{5}$$

are the weighted centroids of $\boldsymbol{y}(i)$ and $\boldsymbol{b}_k(i)$'s. This manipulation is common in related work, *e.g.,* [84, 89].

B. Closed-form Shape Estimation

Substituting the optimal translation (4) (as a function of $\boldsymbol{R}$ and $\boldsymbol{c}$) back into the cost function (3), we obtain an optimization problem that only depends on $\boldsymbol{R}$ and $\boldsymbol{c}$:

$$\begin{aligned} \min_{\boldsymbol{R}\in\mathrm{SO}(3),\boldsymbol{c}\in\mathbb{R}^K} \; & \textstyle\sum_{i=1}^{N} \left\| \bar{\boldsymbol{y}}(i) - \boldsymbol{R} \sum_{k=1}^{K} c_k \bar{\boldsymbol{b}}_k(i) \right\|^2 + \lambda \left\|\boldsymbol{c}\right\|^2 \\ \text{s.t.} \; & \mathbf{1}^\mathsf{T}\boldsymbol{c} - 1 = 0 \end{aligned} \tag{6}$$

where

$$\bar{\boldsymbol{y}}(i) \triangleq \sqrt{w_i}(\boldsymbol{y}(i) - \boldsymbol{y}_w), \quad \bar{\boldsymbol{b}}_k(i) \triangleq \sqrt{w_i}(\boldsymbol{b}_k(i) - \boldsymbol{b}_{k,w}), \tag{7}$$

are the (weighted) relative positions of $\boldsymbol{y}(i)$ and $\boldsymbol{b}_k(i)$ w.r.t. their corresponding weighted centroids. Using the fact that the ℓ_2 norm is invariant to rotation, problem (6) is equivalent to:

$$\begin{aligned} \min_{\boldsymbol{R}\in\mathrm{SO}(3),\boldsymbol{c}\in\mathbb{R}^K} \; & \textstyle\sum_{i=1}^{N} \left\| \boldsymbol{R}^\mathsf{T}\bar{\boldsymbol{y}}(i) - \sum_{k=1}^{K} c_k \bar{\boldsymbol{b}}_k(i) \right\|^2 + \lambda \left\|\boldsymbol{c}\right\|^2 \\ \text{s.t.} \; & \mathbf{1}^\mathsf{T}\boldsymbol{c} - 1 = 0 \end{aligned} \tag{8}$$

We can further simplify the expression by adopting the following matrix notations:

$$\bar{\boldsymbol{y}} = \left(\bar{\boldsymbol{y}}(1)^\mathsf{T}, \ldots, \bar{\boldsymbol{y}}(N)^\mathsf{T} \right)^\mathsf{T} \in \mathbb{R}^{3N} \tag{9}$$

$$\bar{\boldsymbol{B}} = \begin{bmatrix} \bar{\boldsymbol{b}}_1(1) & \cdots & \bar{\boldsymbol{b}}_K(1) \\ \vdots & \ddots & \vdots \\ \bar{\boldsymbol{b}}_1(N) & \cdots & \bar{\boldsymbol{b}}_K(N) \end{bmatrix} \in \mathbb{R}^{3N\times K} \tag{10}$$

which allows rewriting (8) in the following compact form:

$$\begin{aligned} \min_{\boldsymbol{R}\in\mathrm{SO}(3),\boldsymbol{c}\in\mathbb{R}^K} \; & \left\| \bar{\boldsymbol{B}}\boldsymbol{c} - (\mathbf{I}_N \otimes \boldsymbol{R}^\mathsf{T})\bar{\boldsymbol{y}} \right\|^2 + \lambda \left\|\boldsymbol{c}\right\|^2 \\ \text{s.t.} \; & \mathbf{1}^\mathsf{T}\boldsymbol{c} - 1 = 0 \end{aligned} \tag{11}$$

Now the reader can again recognize that —for any choice of $\boldsymbol{R}$— problem (11) is a linearly-constrained linear least squares problem in $\boldsymbol{c}$, which admits a closed-form solution.

Proposition 1 (Optimal Shape). *For any choice of rotation $\boldsymbol{R}$, the optimal shape parameters that solve (11) can be computed in closed-form as:*

$$\boldsymbol{c}^\star(\boldsymbol{R}) = 2\boldsymbol{G}\bar{\boldsymbol{B}}^\mathsf{T}(\mathbf{I}_N \otimes \boldsymbol{R}^\mathsf{T})\bar{\boldsymbol{y}} + \boldsymbol{g} \tag{12}$$

where we defined the following constant matrices and vectors:

$$\bar{\boldsymbol{H}} \triangleq 2(\bar{\boldsymbol{B}}^\mathsf{T}\bar{\boldsymbol{B}} + \lambda\mathbf{I}_K) \tag{13}$$

$$\boldsymbol{G} \triangleq \bar{\boldsymbol{H}}^{-1} - \frac{\bar{\boldsymbol{H}}^{-1}\mathbf{1}\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}}{\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}\mathbf{1}}, \quad \boldsymbol{g} \triangleq \frac{\bar{\boldsymbol{H}}^{-1}\mathbf{1}}{\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}\mathbf{1}} \tag{14}$$

C. Certifiably Optimal Rotation Estimation

Substituting the optimal shape parameters (12) (as a function of $\boldsymbol{R}$) back into the cost function (11), we obtain an optimization problem that only depends on $\boldsymbol{R}$:

$$\min_{\boldsymbol{R}\in\mathrm{SO}(3)} \left\| \boldsymbol{M}(\mathbf{I}_N \otimes \boldsymbol{R}^\mathsf{T})\bar{\boldsymbol{y}} + \boldsymbol{h} \right\|^2 \tag{15}$$

where the matrix $\boldsymbol{M} \in \mathbb{R}^{(3N+K)\times 3N}$ and vector $\boldsymbol{h} \in \mathbb{R}^{3N+K}$ are defined as:

$$\boldsymbol{M} \triangleq \begin{bmatrix} 2\bar{\boldsymbol{B}}\boldsymbol{G}\bar{\boldsymbol{B}}^\mathsf{T} - \mathbf{I}_{3N} \\ 2\sqrt{\lambda}\boldsymbol{G}\bar{\boldsymbol{B}}^\mathsf{T} \end{bmatrix}, \qquad \boldsymbol{h} \triangleq \begin{bmatrix} \bar{\boldsymbol{B}}\boldsymbol{g} \\ \boldsymbol{g} \end{bmatrix}. \tag{16}$$

Problem (15) is a quadratic optimization over the non-convex set $\mathrm{SO}(3)$. It is known that the set $\boldsymbol{R} \in \mathrm{SO}(3)$ can be described as a set of quadratic equality constraints, see *e.g.,* [62, 77] or [84, Lemma 5]. Therefore, we can succinctly rewrite (15) as a *quadratically constrained quadratic program* (QCQP).

Proposition 2 (Optimal Rotation). *The category-level rotation estimation problem (15) can be equivalently written as a* quadratically constrained quadratic program *(QCQP):*

$$\begin{aligned} \min_{\tilde{\boldsymbol{r}}\in\mathbb{R}^{10}} \; & \tilde{\boldsymbol{r}}^\mathsf{T}\boldsymbol{Q}\tilde{\boldsymbol{r}} \\ \textit{s.t.} \; & \tilde{\boldsymbol{r}}^\mathsf{T}\boldsymbol{A}_i\tilde{\boldsymbol{r}} = 0, \forall i = 1,\ldots,15 \end{aligned} \tag{17}$$

where $\tilde{\boldsymbol{r}} \triangleq [1, \mathrm{vec}\,(\boldsymbol{R})^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{10}$ is a vector stacking all the entries of the unknown rotation $\boldsymbol{R}$ in (15) (with an additional unit element), $\boldsymbol{Q} \in \mathcal{S}^{10}$ is a symmetric constant matrix (expression given in Appendix D), and $\boldsymbol{A}_i \in \mathcal{S}^{10}, i = 1,\ldots,15$ are the constant matrices that define the quadratic constraints describing the set $\mathrm{SO}(3)$ *[84, Lemma 5].*

While a QCQP is still a non-convex problem, it admits a standard semidefinite relaxation, described below.

Corollary 3 (Shor's Semidefinite Relaxation). *The following semidefinite program (SDP) is a convex relaxation of (17).*

$$\begin{aligned} \min_{\boldsymbol{X}\in\mathcal{S}^{10}} \; & \mathrm{tr}\,(\boldsymbol{Q}\boldsymbol{X}) \\ \textit{s.t.} \; & \mathrm{tr}\,(\boldsymbol{A}_0\boldsymbol{X}) = 1, \\ & \mathrm{tr}\,(\boldsymbol{A}_i\boldsymbol{X}) = 0, \forall i = 1,\ldots,15 \\ & \boldsymbol{X} \succeq 0 \end{aligned} \tag{18}$$

Moreover, when the optimal solution $\boldsymbol{X}^\star$ of (18) has rank 1, it can be factored as $\boldsymbol{X}^\star = \begin{bmatrix} 1 \\ \mathrm{vec}\,(\boldsymbol{R}^\star) \end{bmatrix} [1 \; \mathrm{vec}\,(\boldsymbol{R}^\star)]$ where $\boldsymbol{R}^\star$ is the optimal rotation minimizing (15).

The rationale behind using the relaxation (18) is threefold: (i) similar to related quadratic problems over $\mathrm{SO}(3)$ [9, 21, 62, 83, 87], the relaxation (18) empirically produces rank-1 —and hence *optimal*— solutions in common problems; (ii) even when the relaxation is not tight, the problem allows computing how suboptimal the resulting estimate is; (iii) the relaxation entails solving a small semidefinite program (10×10 matrix size, and 16 linear equality constraints), hence it can be solved in milliseconds using standard interior-point methods (*e.g.,* MOSEK [2] interfaced via CVX [24] or CVXPY [18]). The proposed solution falls in the class of *certifiable algorithms* (see [4] and Appendix A in [87]), since it allows solving a hard (non-convex) problem efficiently and with provable a posteriori guarantees.

D. Summary

The results in this section suggest a simple algorithm to compute a certifiably optimal solution to the original pose and shape estimation problem (3): (i) we fist compute the optimal rotation $\boldsymbol{R}^\star$ using the semidefinite relaxation (18) (which is independent from the translation and shape parameters); (ii) we retrieve the optimal shape $\boldsymbol{c}^\star(\boldsymbol{R}^\star)$ given the optimal

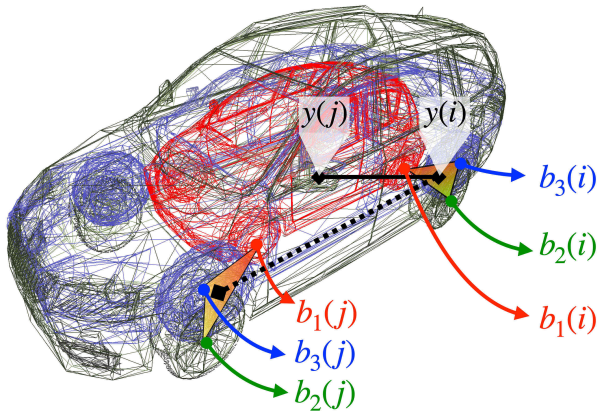


Fig. 2: Example of compatibility test with 3 CAD models of cars (red, dark green, blue, indexed from 1 to 3). (Noiseless) inliers (*e.g.,* the detection of the back wheel $\boldsymbol{y}(i)$ in the figure) must fall in the convex hull of the corresponding points on the CAD models (*e.g.,* triangle $\boldsymbol{b}_1(i) - \boldsymbol{b}_2(i) - \boldsymbol{b}_3(i)$ encompassing the back wheel positions across CAD models). This restricts the relative distance between two inliers and allows filtering out outliers. For instance, the dashed black line shows a distance that is compatible with the location of the convex hulls, while the solid black line is too short compared to the relative position of the wheels (for any car model) and allows pointing out that there is an outlier (*i.e.,* $\boldsymbol{y}(j)$ in the figure).

rotation using (12). Finally, we retrieve the optimal translation $\boldsymbol{t}^\star(\boldsymbol{R}^\star, \boldsymbol{c}^\star)$ using (4). We call the resulting algorithm PACE* (*shaPe and pose estimAtion for Category-level pErception*).

V. Increasing Robustness via Outlier Pruning and Graduated Non-Convexity

This section extends the optimal solver presented in the previous section to deal with the case where some of the measurements are outliers, *i.e.,* some measurements in (2) have unexpectedly large noise. In such as case, problem (3) (even when solved to optimality) does not return an accurate estimate since the quadratic cost in (3) implicitly assumes the measurement noise to be a zero-mean Gaussian. This section first presents a pre-processing that filters out gross outliers from the measurements using a graph-theoretic pruning (Section V-A). Then, we show that the optimal solver (PACE*) can be easily re-used in a robust estimation framework based on graduated non-convexity [86] (Section V-B).

A. Outlier Pruning for Category-Level Perception

We use a graph-theoretic approach to prune outliers, similar to [20, 45, 65, 87]. The key idea is to check if pairs of 3D keypoints can be mutually compatible (*i.e.,* can possibly be simultaneously inliers) and model pair-wise compatibility as edges in a graph where the nodes are the 3D keypoints. Then, since the inliers are all mutually compatible, they must form a large clique in the graph and can be retrieved by computing the maximum clique. Our main novelty is to develop an efficient mutual compatibility test for category-level perception, while related work has focused on known shapes [20, 65, 87].

Mutually Compatible Measurements. The goal here is to design a boolean condition that allows asserting if two measurements can be both inliers for any choice of pose and shape parameters. The challenge is that such a condition should not depend on the pose and shape parameters, which are unknown. Therefore, we show how to manipulate the model (2) to obtain a condition that do not depend on $\boldsymbol{R}$, $\boldsymbol{t}$, and $\boldsymbol{c}$. Towards this goal, let us call ε the maximum error for a measurement to be called an inlier. In other words, a measurement in (2) is an inlier if $\|\boldsymbol{\epsilon}(i)\| \leq \varepsilon$.

A pair of inliers i and j in eq. (2) must satisfy $\|\boldsymbol{\epsilon}(i)\| \leq \varepsilon$ and $\|\boldsymbol{\epsilon}(j)\| \leq \varepsilon$. Taking the difference between measurement i and j in (2) leads to:

$$\boldsymbol{y}(j) - \boldsymbol{y}(i) = \boldsymbol{R} \sum_{k=1}^{K} c_k (\boldsymbol{b}_k(j) - \boldsymbol{b}_k(i)) + (\boldsymbol{\epsilon}(j) - \boldsymbol{\epsilon}(i))$$

where the translation cancels out in the subtraction. Now taking the ℓ_2 norm of both members:

$$\|\boldsymbol{y}(j) - \boldsymbol{y}(i)\| = \|\boldsymbol{R} \sum_{k=1}^{K} c_k (\boldsymbol{b}_k(j) - \boldsymbol{b}_k(i)) + (\boldsymbol{\epsilon}(j) - \boldsymbol{\epsilon}(i))\|$$

Using the triangle inequality and observing that $\|\boldsymbol{\epsilon}(i)\| \leq \varepsilon$ and $\|\boldsymbol{\epsilon}(j)\| \leq \varepsilon$ imply $\|\boldsymbol{\epsilon}(j) - \boldsymbol{\epsilon}(i)\| \leq 2\varepsilon$:

$$-2\varepsilon \leq \|\boldsymbol{y}(j) - \boldsymbol{y}(i)\| - \|\boldsymbol{R} \sum_{k=1}^{K} c_k (\boldsymbol{b}_k(j) - \boldsymbol{b}_k(i))\| \leq 2\varepsilon \quad (19)$$

Now observing that the ℓ_2 norm is invariant to rotation and rearranging the terms:

$$\begin{aligned} \|\textstyle\sum_{k=1}^{K} c_k (\boldsymbol{b}_k(j) - \boldsymbol{b}_k(i))\| - 2\varepsilon \leq \; & \|\boldsymbol{y}(j) - \boldsymbol{y}(i)\| \; \leq \\ & \|\textstyle\sum_{k=1}^{K} c_k (\boldsymbol{b}_k(j) - \boldsymbol{b}_k(i))\| + 2\varepsilon \end{aligned} \quad (20)$$

Considering the extreme cases over the set of possible shape coefficients:

$$\overbrace{\min_{\boldsymbol{c} \geq 0, \mathbf{1}^\mathsf{T}\boldsymbol{c}=1} \|\sum_{k=1}^{K} c_k (\boldsymbol{b}_k(j) - \boldsymbol{b}_k(i))\|}^{b_{ij}^{\min}} - 2\varepsilon \leq \|\boldsymbol{y}(j) - \boldsymbol{y}(i)\| \leq \underbrace{\max_{\boldsymbol{c} \geq 0, \mathbf{1}^\mathsf{T}\boldsymbol{c}=1} \|\sum_{k=1}^{K} c_k (\boldsymbol{b}_k(j) - \boldsymbol{b}_k(i))\|}_{b_{ij}^{\max}} + 2\varepsilon \quad (21)$$

Since $\sum_{k=1}^{K} c_k \boldsymbol{b}_k(j)$ is a convex combinations of the points $\boldsymbol{b}_k(j)$ $(k = 1 \ldots, K)$ and hence lies in the convex hull of such points, the term $\|\sum_{k=1}^{K} c_k (\boldsymbol{b}_k(j) - \boldsymbol{b}_k(i))\|$ represents the distance between two (unknown) points in the two convex hulls defined by the set of points $\boldsymbol{b}_k(j)$ and $\boldsymbol{b}_k(i)$ $(k = 1 \ldots, K)$ (Fig. 2). The minimum $b_{ij}^{\min}$ and the maximum $b_{ij}^{\max}$ over the convex hulls can be easily computed, either in closed form or via small convex programs (details in Appendix F). Therefore, a pair of inliers must satisfy:

$$b_{ij}^{\min} - 2\varepsilon \leq \|\boldsymbol{y}(j) - \boldsymbol{y}(i)\| \leq b_{ij}^{\max} + 2\varepsilon \quad (22)$$

Note that $b_{ij}^{\min}$ and $b_{ij}^{\max}$ only depend on the given library and can be pre-computed. Any pair of measurements that do not satisfy (22) cannot be simultaneously inliers for problem (2).

Largest Set of Compatible Measurements. The compatibility test (22) checks if a pair of measurements can be together in the inlier set. Therefore, after testing compatibility between every pair of keypoints, we can find inliers by searching for the largest set of mutually compatible measurements. References [20, 45, 65, 87] have already established that the largest set of mutually compatible measurements can be found by computing the maximum clique of a graph where nodes correspond to the 3D keypoints and an edge connects nodes i and j is the corresponding measurements satisfy the compatibility test (22). While we refer the reader to those papers for details, here we observe that such graph-theoretic approach has been shown to remove a large amount of gross outliers [87] (while preserving all inliers). We will handle the remaining outliers using graduated non-convexity as discussed in the next section. In our experiments, we show that while graduated non-convexity can be robust to up to $50-60\%$ outliers, the addition of this graph-theoretic outlier pruning boosts robustness to $70-90\%$ outliers.

B. Graduated Non-Convexity for Category-Level Perception

While the graph-theoretic outlier pruning in the previous section is able to filter out a large fraction of gross outliers (without even computing an estimate), in this section we show how to use the remaining measurements (potentially still contaminated by a few outliers) to compute an accurate pose and shape estimate. Towards this goal, we use a standard robust estimation framework, and we optimize the resulting optimization using graduated non-convexity (GNC) [86].

As prescribed by standard robust estimation, we re-gain robustness to outliers by replacing the squared ℓ_2 norm in (3) with a robust loss function ρ:

$$\begin{aligned}\min_{\substack{\boldsymbol{R}\in\mathrm{SO}(3),\\ \boldsymbol{t}\in\mathbb{R}^3,\boldsymbol{c}\in\mathbb{R}^K}} & \sum_{i=1}^{N}\rho\left(\left\|\boldsymbol{y}(i)-\boldsymbol{R}\sum_{k=1}^{K}c_k\boldsymbol{b}_k(i)-\boldsymbol{t}\right\|\right)+\lambda\left\|\boldsymbol{c}\right\|^2\\ \text{s.t.}\quad & \mathbf{1}^{\mathsf{T}}\boldsymbol{c}=1\end{aligned}\tag{23}$$

While GNC can be applied to a broad class of loss functions [86], here we consider a truncated least square loss $\rho(r)=\min(r^2,\varepsilon^2)$ which minimizes the squared residuals whenever they are below ε^2 (note: the constant ε is the same inlier threshold of the previous section) or becomes constant otherwise. Such cost function can be written by using auxiliary slack variables $\rho(r)=\min(r^2,\varepsilon^2)=\min_{\omega\in\{0,1\}}\omega r^2+(1-\omega)\varepsilon^2$ [86], hence allowing to rewrite (23) as:

$$\begin{aligned}\min_{\substack{\boldsymbol{R}\in\mathrm{SO}(3),\\ \boldsymbol{t}\in\mathbb{R}^3,\boldsymbol{c}\in\mathbb{R}^K\\ \omega_i\in\{0,1\}\forall i}} & \sum_{i=1}^{N}\omega_i\left\|\boldsymbol{y}(i)-\boldsymbol{R}\sum_{k=1}^{K}c_k\boldsymbol{b}_k(i)-\boldsymbol{t}\right\|^2+(1-\omega_i)\varepsilon^2+\lambda\left\|\boldsymbol{c}\right\|^2\\ \text{s.t.}\quad & \mathbf{1}^{\mathsf{T}}\boldsymbol{c}=1\end{aligned}\tag{24}$$

In (24), when $\omega_i=1$, the i-th measurement is considered an inlier and the cost minimizes the corresponding squared residual; when $\omega_i=0$, the cost becomes independent of $\boldsymbol{y}(i)$ hence the corresponding measurement is rejected as an outlier. Therefore, problem (24) simultaneously estimates pose and shape variables $(\boldsymbol{R},\boldsymbol{t},\boldsymbol{c})$ while classifying inliers/outliers via the binary weights ω_i $(i=1,\dots,N)$.

Now the advantage is that we can minimize (24) with an alternation scheme where we iteratively optimize (i) over $(\boldsymbol{R},\boldsymbol{t},\boldsymbol{c})$ with fixed weights ω_i and (ii) over the weights ω_i with fixed $(\boldsymbol{R},\boldsymbol{t},\boldsymbol{c})$. This approach is convenient since the optimization over $(\boldsymbol{R},\boldsymbol{t},\boldsymbol{c})$ can be solved to optimality using PACE*, while the optimization of the weights can be solved in closed form [86]. To improve convergence of this alternation scheme, we adopt graduated non-convexity [6, 86, 86], which starts with a convex approximation of the loss function in (24) and then gradually increases the non-convexity until the original robust loss ρ in (24) is retrieved.

We call PACE# *the approach applying graph-theoretic outlier pruning and then using GNC to retrieve a robust estimate.*

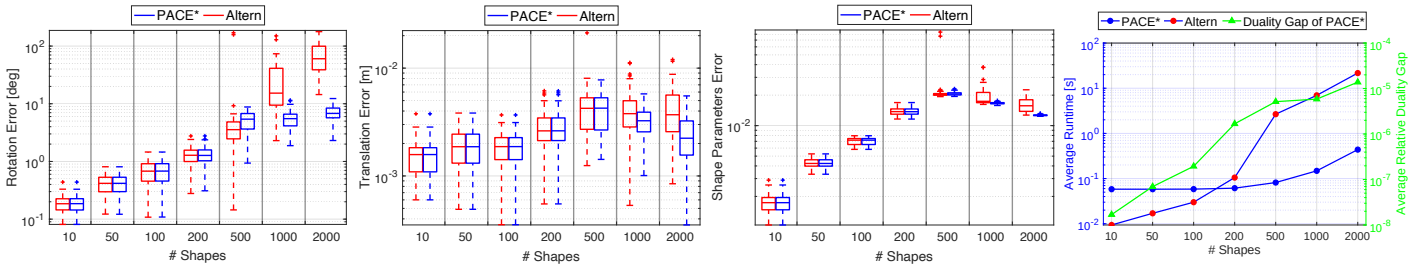
VI. Experiments

In this section, we first demonstrate the optimality of PACE* and the robustness of PACE# in simulated data and in the PASCAL3D+ dataset [82] (Section VI-A). Then we show that PACE# can be integrated in a realistic perception system and achieve state-of-the-art performance on vehicle pose estimation in the ApolloScape dataset [80] (Section VI-B). In both cases, our solvers outperform baseline approaches in terms of accuracy.

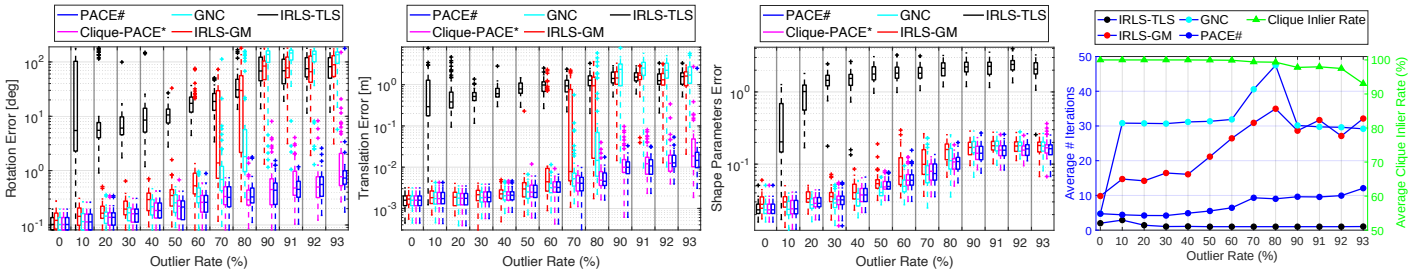
A. Ablation: Optimality and Robustness

Optimality of PACE***.** To evaluate the performance of PACE* in solving the outlier-free problem (3), we randomly simulate K shape models $\mathcal{B}_k$ whose points $\boldsymbol{b}_k(i)$'s are drawn from an i.i.d. Gaussian distribution $\mathcal{N}(\mathbf{0},\mathbf{I}_3)$. We sample shape parameters $\boldsymbol{c}$ uniformly at random in $[0,1]^K$, and normalize $\boldsymbol{c}$ such that $\mathbf{1}^{\mathsf{T}}\boldsymbol{c}=1$. Then we draw random poses $(\boldsymbol{R},\boldsymbol{t})$ as in [87] and generate the measurements $\boldsymbol{y}(i)$ according to the model (2), where the noise $\boldsymbol{\epsilon}(i)$ follows $\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I}_3)$ with standard deviation $\sigma=0.01$. We fix $N=100$, increase K from 10 up to 2000 and set the regularization factor $\lambda=\sqrt{K/N}$ so that larger regularization is imposed when K increases and the problem becomes more ill-posed. We compare PACE* with a baseline approach based on alternating minimization [26, 43, 58] (details given in Appendix G) that offers no optimality guarantees (label: Altern).

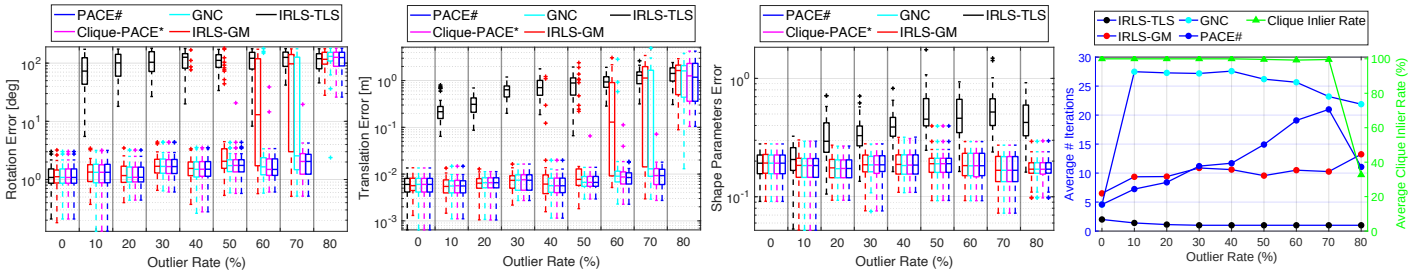
Fig. 3(a) plots the statistics of rotation error (angular distance between estimated and ground-truth rotations), translation error, shape parameters error (ℓ_2 distance between estimated and ground-truth translation/shape parameters), as well as average runtime and relative duality gap (details in Appendix E). We make the following observations: (i) PACE* returns accurate pose and shape estimates up to $K=2000$, while Altern starts failing at $K=500$. (ii) Although Altern is faster than PACE* for small K (*e.g.*, $K<200$), PACE* is orders of magnitude faster than Altern for large K. In fact, the runtime of PACE* only slightly increases because PACE* solves a fixed-size SDP regardless of the increase in K (the increase in runtime is due to inversion of a dense matrix in (13)). (iii)



(a) Performance of the certifiably optimal solver PACE⋆ on outlier-free random simulated data: $N = 100$.



(b) Robustness of PACE# against increasing outliers on random simulated data: $N = 100$, $K = 10$, $r = 0.1$.



(c) Robustness of PACE# against increasing outliers on the *car* category in the PASCAL3D+ dataset [82]: $N = 12$, $K = 9$.

Fig. 3: Performance of PACE⋆ and PACE# compared with baselines in simulated experiments. (a) PACE⋆ compared with alternating minimization (Altern) on random simulated outlier-free data with $N = 100$ and K increasing from 10 to 2000; (b) PACE# along with two individual components of itself (Clique-PACE⋆ and GNC), compared with two variants of iterative reweighted least squares (IRLS-GM and IRLS-TLS) [73] on random simulated outlier-contaminated data with $N = 100$, $K = 10$ and outlier rates up to 93%; (c) same as (b) but using the *car* category CAD models from the PASCAL3D+ dataset [82], with $N = 12$, $K = 9$ and outlier rates up to 80%. Each boxplot and lineplot summarizes 50 Monte Carlo random runs.

The relaxation (18) is empirically tight (duality gap $< 10^{-4}$), certifying global optimality of the solution returned by PACE⋆.

Robustness of PACE#. To test the robustness of PACE# on outlier-contaminated data, we follow the same data generation protocol as before, except that (i) when generating the CAD models, we follow a more realistic active shape model [16] where we first generate a mean shape $\mathcal{B}$ whose points $\boldsymbol{b}(i)$'s are i.i.d. Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I}_3)$, and then each CAD model is generated from the mean shape by: $\boldsymbol{b}_k(i) = \boldsymbol{b}(i) + \boldsymbol{v}(i)$, where $\boldsymbol{v}(i)$ follows $\mathcal{N}(\mathbf{0}, r^2\mathbf{I}_3)$ and represents the *intra-class variation* of semantic keypoints with variation radius r. (ii) we replace a fraction of the measurements $\boldsymbol{y}(i)$ with arbitrary 3D points sampled according to $\mathcal{N}(\mathbf{0}, \mathbf{I}_3)$ and violating the generative model (2). We compare PACE# with two variants: Clique-PACE⋆ (*i.e.,* after pruning outliers using maximum clique, PACE⋆ is applied *without* GNC) and GNC (*i.e.,* GNC is applied to problem data *without* any outlier pruning), as well as two variants of the popular *iterative reweighted least squares* heuristics: IRLS-TLS and IRLS-GM, where TLS and GM denote the truncated least squares cost function and the Geman-McClure cost function [73]. For fair comparison, we use PACE⋆ inside PACE#, GNC, IRLS-TLS, and IRLS-GM when updating $(\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{c})$ given fixed weights. We set $\varepsilon = 0.05$ for outlier pruning and GNC. Fig. 3(b) plots the results under increasing outlier rates up to 93% when $N = 100$, $K = 10$ and $r = 0.1$. We make the following observations: (i) IRLS-TLS quickly fails (at 10% outlier rate) due to the binary nature of the TLS cost, while IRLS-GM is robust to 40% outliers. (ii) GNC alone already outperforms IRLS-TLS and IRLS-GM and is robust to 60% outliers. (iii) With our maximum-clique outlier pruning, the robustness of PACE# is boosted to 92%, a level that is comparable to cases when the shapes are known (*e.g.,* [87]). In addition, outlier pruning speeds up the convergence of GNC (*cf.* number of iterations plot for GNC and PACE# in Fig. 3(b)). (iv) Even without GNC, the outlier pruning is so effective that PACE⋆ alone is able to succeed with up to 90% outliers, despite that the estimates are typically less accurate than PACE#. In fact, looking at the clique inlier rate plot (green lineplot in Fig. 3(b)), the reader sees that the set of measurements after maximum clique pruning is almost free of outliers, explaining the surprising performance of Clique-PACE⋆. In Appendix H, we show extra results for $r = 0.2$ and $K = 50$, which further

confirm PACE#'s robustness to 90% outliers.

Robustness on PASCAL3D+. For a simulation setup that is closer to realistic scenarios, we use the CAD models from the *car* category in the PASCAL3D+ dataset [82], which contains $K = 9$ CAD models of $N = 12$ semantic keypoints. We randomly sample $(\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{c})$ and add noise and outliers as before, and compare the performance of PACE# with other baselines, as shown in Fig. 3(c). The dominance of PACE# over other baselines, and the effectiveness of outlier pruning is clearly seen across the plots. PACE# is robust to 70% outliers, while other baselines break at a much lower outlier rate. Note that at 80% outlier rate, there are only two inlier semantic keypoints, making it pathological to estimate shape and pose.

B. Vehicle Pose Estimation on ApolloScape

Setup and Baselines. We evaluate PACE# on the ApolloScape dataset [67, 80]. The ApolloScape self-driving dataset is a large collection of multi-modal data collected in four different cities in China under varying lighting and road conditions [80]. Within the dataset, annotations are provided for different perception tasks, ranging from pixel-level semantic segmentation to dense semantic 3D point clouds for the environments. For our experiments, we specifically use the subset of ApolloScape named ApolloCar3D. ApolloCar3D consists of high-resolution (3384×2710) images taken from the main ApolloScape dataset, with additional 2D annotations of semantic keypoints, ground truth poses, and 3D CAD models of car instances in each frame. The dataset contains a total of 5277 images, with an average of 11.7 cars per image, and a total of 79 ground-truth CAD models [67]. For each car, a total of 66 semantic keypoints were labeled on 2D images.

We compare PACE# against DeepMANTA [12], 3D-RCNN [37], and GSNet [33], three recent state of the art methods for 3D vehicle pose estimation. For our experiments, we use the official splits of the ApolloCar3D dataset. Namely, we use the validation split (200 images) for all the quantitative experiments shown below, consistent with the evaluation setups reported in other baseline methods.

We use the 2D semantic keypoints extracted by GSNet [33] as measurements for PACE#; in particular we use the pretrained weights from [33] and reject keypoints with confidence less than 0.05. For each 2D semantic keypoint, we retrieve the corresponding depth from the depth images provided by ApolloScape; the resulting technique is labeled PACE#-ApolloDepths. We also provide an ablation study to assess the impact of depth and keypoint quality on PACE#. Towards this goal, we test two variants: PACE#-GTDepths uses ground-truth depths obtained by ray-tracing the GSNet keypoints using ground-truth 3D car models, while PACE#-GTKeypoints uses ground-truth 2D semantic keypoints with ground-truth depths. While the 2D semantic keypoint annotations are provided by ApolloCar3D, the dataset does not provide the corresponding 3D keypoint annotations on the CAD models. To obtain the necessary 2D-3D correspondences, we manually label the 66 3D semantic keypoints on the 79 CAD models. We then provide this set of labeled 3D points as the shape library to PACE#. We use $\lambda = 0.5$ and $\varepsilon = 0.15$ in PACE#.

Results. Table I shows the performance of PACE# against various baselines. We use two metrics called A3DP-Rel and A3DP-Abs (for both, the higher the better) following the same definitions used in [67]. They are measures of precision with thresholds jointly considering translation, rotation, and 3D shape similarity between estimated cars and ground truth. A3DP-Abs uses absolute translation thresholds, whereas A3DP-Rel uses relative translation thresholds. The *mean* column represents the average A3DP-Abs/Rel over 10 different thresholds. *c-l* represents a loose criterion (2.8 m for translation error, $\pi/6$ rad for rotation error, and 0.5 for shape similarity), and *c-s* represents a strict criterion (1.4 m for translation error, $\pi/12$ rad for rotation error, and 0.75 for shape similarity). PACE# outperforms the baselines in terms of the *mean* and *c-s* criteria; this is partially expected since we use depth information, which is not available to the other methods. In terms of the strict criterion *c-s*, PACE# outperforms competitors by a large amount, confirming that it can retrieve highly accurate estimates. PACE#-GTDepths outperforms baselines across all criteria, suggesting that if accurate depth measurements are available, PACE# can roughly double the performance of state-of-the-art methods in terms of *mean* and *c-s* criteria. PACE#-GTKeypoints shows the results produced by PACE# when using ground-truth keypoint detections and depths: this is the best potential accuracy PACE# could achieve if provided with perfect keypoint detections. In our tests, the average number of inliers produced by GSNet is 21.8%,[1] showing that there is still a large margin of improvement for state-of-the-art methods in semantic keypoint detection.

	A3DP-Rel ↑			A3DP-Abs ↑		
	mean	c-l	c-s	mean	c-l	c-s
DeepMANTA [12]	16.0	23.8	19.8	20.1	30.7	23.8
3D-RCNN [37]	10.8	17.8	11.9	16.4	29.7	19.8
GSNet [33]	20.2	**40.5**	19.9	18.9	**37.4**	18.4
PACE#-ApolloDepths	**25.9**	35.7	**33.7**	**22.4**	34.7	**31.6**
PACE#-GTDepths	36.0	45.4	43.6	35.3	44.2	43.2
PACE#-GTKeypoints	64.5	88.1	86.0	64.3	88.1	86.1

TABLE I: Evaluation of PACE# on ApolloScape. Results for DeepMANTA,3D-RCNN, and GSNet are taken from [33]. The best result for each column is highlighted in boldface.

Table II shows the timing breakdown for PACE#. We also report the timing for the GSNet keypoint detection from [33] for completeness. In our current implementation of PACE#, the max-clique pruning is in C++ and its runtime is negligible, while GNC is implemented in Python. All tests are run on a Linux computer with an Intel i9-9920X CPU at 3.5 GHz.

GSNet keypoint detection	PACE#	
	Max-clique	GNC
0.45 s	2 ms	0.45 s

TABLE II: Timing breakdown for PACE#.

[1] We define true inliers as 2D keypoint detections such that there exists a ground-truth annotated keypoint with the same ID within a radius of 5 pixels.

VII. Conclusion

We proposed PACE⋆, the first certifiably optimal solver for the estimation of the pose and shape of 3D objects from 3D keypoint detections. While existing iterative methods get stuck in local minima corresponding to poor estimates, PACE⋆ leverages a tight and fixed-size SDP relaxation to compute certifiably optimal estimates. We also design a second algorithm, PACE#, that adds an outlier rejection layer to PACE⋆ and is able to estimate accurate pose and shape parameters in the face of large amounts of outliers (*e.g.,* $70-90\%$ of the measurements are incorrect). The proposed methods dominate the state of the art in terms of accuracy and robustness on both Monte Carlo simulations and on the PASCAL3D+ dataset. Moreover, we show that PACE# can be successfully combined with deep-learned keypoint detectors, and leads to highly accurate vehicle pose estimates in the ApolloScape driving datasets.

Acknowledgments

This work was partially funded by ARL DCIST CRA W911NF-17-2-0181, ONR RAIDER N00014-18-1-2828, NSF CAREER award "Certifiable Perception for Autonomous Cyber-Physical Systems", and Lincoln Laboratory's Resilient Perception in Degraded Environments program. We would like to also thank Charleen Tan for helping with labeling 3D keypoints.

References

[1] P. Antonante, V. Tzoumas, H. Yang, and L. Carlone. Outlier-robust estimation: Hardness, minimally-tuned algorithms, and applications. *arXiv preprint arXiv: 2007.15109*, 2020. (pdf). 13

[2] M. ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 8.1.*, 2017. URL http://docs.mosek.com/8.1/toolbox/index.html. 4

[3] A. Avetisyan, A. Dai, and M. Nießner. End-to-End CAD Model Retrieval and 9DoF Alignment in 3D Scans. In *Intl. Conf. on Computer Vision (ICCV)*, 2019. 2

[4] A. Bandeira. A note on probably certifiably correct algorithms. *arXiv:1509.00824*, 2015. 4

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24 (4):509–522, 2002. 2

[6] A. Blake and A. Zisserman. *Visual reconstruction*. MIT Press, 1987. 6

[7] G. Blekherman, P. A. Parrilo, and R. R. Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012. 2

[8] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. 13

[9] J. Briales and J. Gonzalez-Jimenez. Fast global optimality verification in 3D SLAM. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4630–4636, Oct 2016. doi: 10.1109/IROS.2016.7759681. 4

[10] B. Burchfiel and G. Konidaris. Probabilistic category-level pose estimation via segmentation and predicted-shape priors. *arXiv preprint arXiv:1905.12079*, 2019. 2

[11] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2

[12] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2040–2049, 2017. 1, 2, 8

[13] T. Chan and L. Vese. An active contour model without edges. In *International Conference on Scale-Space Theories in Computer Vision*, pages 141–151. Springer, 1999. 2

[14] D. Chen, J. Li, Z. Wang, and K. Xu. Learning canonical shape space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11973–11982, 2020. 2

[15] 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):355–365, 1994. 2

[16] 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):38–59, January 1995. 1, 2, 7

[17] E. De Klerk. *Aspects of semidefinite programming: interior point algorithms and selected applications*, volume 65. Springer Science & Business Media, 2006. 14

[18] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016. 4

[19] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 2

[20] O. Enqvist, K. Josephson, and F. Kahl. Optimal correspondences from pairwise constraints. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1295–1302, 2009. 5, 6

[21] 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. 4

[22] Q. Feng and N. Atanasov. Fully convolutional geometric features for category-level object alignment. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8492–8498, 2020. 2

[23] W. Gao and R. Tedrake. kpam 2.0: Feedback control for category-level robotic manipulation. *IEEE Robotics and Automation Letter (RA-L)*, 2020. 1, 2

[24] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming. URL http://cvxr.com/cvx. 4

[25] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. A papier-mâché approach to learning 3d surface generation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 216–224, 2018. 2

[26] L. Gu and T. Kanade. 3D alignment of face in a single image. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1305–1312, 2006. 3, 6, 15

[27] A. Gupta and L. Carlone. Online monitoring for neural network based monocular pedestrian pose estimation. In *Intl. Conf. on Intelligent Transportation Systems*, 2020. (pdf). 1

[28] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *IJCV*, 103(3):267–305, 2013. 3

[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2980–2988, 2017. 1, 2

[30] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer.*, 4(4): 629–642, Apr 1987. 1

[31] T. Hou, A. Ahmadyan, L. Zhang, J. Wei, and M. Grundmann. Mobilepose: Real-time pose estimation for unseen objects with weak shape supervision. *arXiv preprint arXiv:2003.03522*, 2020. 2

[32] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *Intl. J. of Computer Vision*, 1(4):321–331, 1987. 2

[33] 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 Conference on Computer Vision*, pages 515–532. Springer, 2020. 1, 2, 8

[34] L. Kneip, H. Li, and Y. Seo. UPnP: An optimal o(n) solution to the absolute pose problem with universal applicability. In *European Conf. on Computer Vision (ECCV)*, pages 127–142. Springer, 2014. 1

[35] N. Kolotouros, G. Pavlakos, and K. Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. 2

[37] A. Kundu, Y. Li, and J. M. Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3559–3568, 2018. 1, 2, 8

[38] K. Lasinger, R. Ranftl, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019. 2

[39] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3): 796–817, 2001. 2

[40] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[41] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3706–3715, 2020. 2

[42] J. Lim, H. Pirsiavash, and A. Torralba. Parsing IKEA objects: Fine pose estimation. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2992–2999, 2013. 2

[43] Y.-L. Lin, V. I. Morariu, W. H. Hsu, and L. S. Davis. Jointly optimizing 3D model fitting and fine-grained classification. In *European Conf. on Computer Vision (ECCV)*, 2014. 3, 6, 15

[44] J. G. López, A. Agudo, and F. Moreno-Noguer. Vehicle pose estimation via regression of semantic points of interest. In *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 209–214. IEEE, 2019. 1, 2

[45] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan. Pairwise consistent measurement set maximization for robust multi-robot map merging. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2916–2923, 2018. 5, 6

[46] F. Manhardt, W. Kehl, and A. Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2069–2078, 2019. 2

[47] F. Manhardt, M. Nickel, S. Meier, L. Minciullo, and N. Navab. Cps++: Class-level 6d pose and shape estimation from monocular images. *arXiv preprint arXiv:2003.05848*, 2020. 2

[48] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2019. 1, 2

[49] F. L. Markley. Attitude determination using vector observations and the singular value decomposition. *The Journal of the Astronautical Sciences*, 36(3):245–258, 1988. 15

[50] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017. 2

[51] P. McCausland. Self-driving Uber car that hit and killed woman did not recognize that pedestrians jaywalk. *NBC News*, Nov 2019. 1

[52] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer,

2016. 2

[53] M. Oberweger, M. Rad, and V. Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 2

[54] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on pattern analysis and machine intelligence*, 22 (12):1424–1445, 2000. 2

[55] Á. Parra Bustos and T. J. Chin. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE Trans. Pattern Anal. Machine Intell.*, 40(12):2868–2882, 2018. 1

[56] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. 6-dof object pose from semantic keypoints. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017. 1, 2, 3

[57] 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)*, pages 4561–4570, 2019. 2

[58] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3D human pose from 2D image landmarks. In *European Conf. on Computer Vision (ECCV)*, 2012. 3, 6, 15

[59] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[60] S. R. Richter and S. Roth. Matryoshka networks: Predicting 3D geometry via nested shape layers. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1936–1944, 2018. 2

[61] R. T. Rockafellar. *Convex analysis*, volume 36. Princeton university press, 1970. 15

[62] 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. accepted, arxiv preprint: 1611.00128, (pdf). 3, 4

[63] G. Schweighofer and A. Pinz. Globally optimal O(n) solution to the PnP problem for general camera models. In *British Machine Vision Conf. (BMVC)*, pages 1–10, 2008. 1

[64] M. Shan, Q. Feng, and N. Atanasov. Object residual constrained visual-inertial odometry. In *technical report, https://moshanatucsd.github.io/orcvio_githubpage/*, 2019. 2

[65] J. Shi, H. Yang, and L. Carlone. ROBIN: a graph-theoretic approach to reject outliers in robust estimation using invariants. *arXiv preprint arXiv: 2011.03659*, 2020. (pdf). 2, 5, 6

[66] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 2

[67] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, and R. Yang. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5452–5462, 2019. 8

[68] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 2

[69] 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. 1, 2

[70] Taiwan English News. Tesla on autopilot crashes into overturned truck, 2020. URL https://taiwanenglishnews.com/tesla-on-autopilot-crashes-into-overturned-truck/. 1

[71] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2088–2096, 2017. 2

[72] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox. What Do Single-view 3D Reconstruction Networks Learn? In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3405–3414, 2019. 2

[73] K. M. Tavish and T. D. Barfoot. At all costs: A comparison of robust cost functions for camera correspondence outliers. In *Computer and Robot Vision (CRV), 2015 12th Conference on*, pages 62–69. IEEE, 2015. 7

[74] A. N. Tikhonov, A. Goncharsky, V. Stepanov, and A. G. Yagola. *Numerical methods for the solution of ill-posed problems*, volume 328. Springer Science & Business Media, 2013. 3

[75] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *arXiv preprint arXiv:1406.2984*, 2014. 2

[76] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 2

[77] 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"*, 2015. (pdf). 4

[78] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 2

[79] J. Wang, V. Magron, and J.-B. Lasserre. TSSOS: A Moment-SOS hierarchy that exploits term sparsity. *SIAM*

Journal on Optimization, 31(1):30–58, 2021. 2

[80] P. Wang, X. Huang, X. Cheng, D. Zhou, Q. Geng, and R. Yang. The ApolloScape open dataset for autonomous driving and its application. *IEEE Trans. Pattern Anal. Machine Intell.*, 2019. 1, 2, 6, 8

[81] J. Wangni, D. Lin, J. Liu, K. Daniilidis, and J. Shi. Towards statistically provable geometric 3d human pose recovery. *SIAM Journal on Imaging Sciences*, 14(1):246–270, 2021. 2

[82] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond PASCAL: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82. IEEE, 2014. 6, 7, 8

[83] H. Yang and L. Carlone. A quaternion-based certifiably optimal solution to the Wahba problem with outliers. In *Intl. Conf. on Computer Vision (ICCV)*, 2019. (Oral Presentation, accept rate: 4%), Arxiv version: 1905.12536, (pdf). 4, 15

[84] H. Yang and L. Carlone. In perfect shape: Certifiably optimal 3D shape reconstruction from 2D landmarks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. Arxiv version: 1911.11924, (pdf). 1, 2, 3, 4, 13

[85] H. Yang and L. Carlone. One ring to rule them all: Certifiably robust geometric perception with outliers. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. (pdf). 14

[86] 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): 1127–1134, 2020. arXiv preprint arXiv:1909.08605 (with supplemental material), (pdf) . 2, 5, 6

[87] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Trans. Robotics*, 2020. extended arXiv version 2001.07715 (pdf). 1, 2, 4, 5, 6, 7

[88] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi. Revisiting the PnP problem: A fast, general and optimal solution. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2344–2351, 2013. 1

[89] 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. 1, 2, 3

[90] X. Zhou, M. Zhu, S. Leonardos, and K. Daniilidis. Sparse representation for 3D shape estimation: A convex relaxation approach. *IEEE Trans. Pattern Anal. Machine Intell.*, 39(8):1648–1661, 2017. 2

APPENDIX A
PROBLEM (3) IS A MAP ESTIMATOR WHEN THE MEASUREMENT NOISE IS GAUSSIAN

Here we prove that the optimization in eq. (3) is a *maximum a posteriori* (MAP) estimator when the measurement noise $\boldsymbol{\epsilon}(i)$ in (2) follows a zero-mean Gaussian with covariance $\frac{1}{w_i}\mathbf{I}_3$ (where $\mathbf{I}_3$ is the 3-by-3 identity matrix) and we have a zero-mean Gaussian prior with covariance $\frac{1}{\lambda}\mathbf{I}_K$ over the shape parameters $\boldsymbol{c}$. Mathematically:

$$\mathbb{P}\left(\boldsymbol{\epsilon}(i)\right) = \kappa_\epsilon \exp\left(-\frac{w_i}{2}\|\boldsymbol{\epsilon}(i)\|^2\right), \tag{A1}$$

$$\mathbb{P}\left(\boldsymbol{c}\right) = \kappa_c \exp\left(-\frac{\lambda}{2}\|\boldsymbol{c}\|^2\right), \tag{A2}$$

where κ_ϵ and κ_c are suitable normalization constants that are irrelevant for the following derivation.

A MAP estimator for the unknown parameters $\boldsymbol{x} \triangleq \{\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{c}\}$ (belonging to a suitable domain $\mathbb{X}$) given measurements $\boldsymbol{y}(i)$ $(i = 1, \ldots, N)$ is defined as the maximum of the posterior distribution $\mathbb{P}\left(\boldsymbol{x}|\boldsymbol{y}(1) \ \ldots \ \boldsymbol{y}(N)\right)$:

$$\arg\max_{\boldsymbol{x}\in\mathbb{X}} \mathbb{P}\left(\boldsymbol{x}|\boldsymbol{y}(1) \ \ldots \ \boldsymbol{y}(N)\right) = \arg\max_{\boldsymbol{x}\in\mathbb{X}} \prod_{i=1}^{N} \mathbb{P}\left(\boldsymbol{y}(i)|\boldsymbol{x}\right)\mathbb{P}\left(\boldsymbol{x}\right) \tag{A3}$$

where on the right we applied Bayes rule and used the standard assumption of independent measurements. Using (A1) and (2) we obtain:

$$\mathbb{P}\left(\boldsymbol{y}(i)|\boldsymbol{x}\right) = \kappa_\epsilon \exp\left(-\frac{w_i}{2}\left\|\boldsymbol{y}(i) - \boldsymbol{R}\sum_{k=1}^{K} c_k \boldsymbol{b}_k(i) - \boldsymbol{t}\right\|^2\right). \tag{A4}$$

Moreover, assuming we only have a prior on $\boldsymbol{c}$:

$$\mathbb{P}\left(\boldsymbol{x}\right) = \mathbb{P}\left(\boldsymbol{c}\right) = \kappa_c \exp\left(-\frac{\lambda}{2}\|\boldsymbol{c}\|^2\right). \tag{A5}$$

Substituting (A4) and (A5) back into (A3) and observing that the maximum of the posterior is the same as the minimum of the negative logarithm of the posterior:

$$\arg\max_{\boldsymbol{x}\in\mathbb{X}} \prod_{i=1}^{N} \mathbb{P}\left(\boldsymbol{y}(i)|\boldsymbol{x}\right)\mathbb{P}\left(\boldsymbol{x}\right) = \tag{A6}$$

$$\arg\min_{\boldsymbol{x}\in\mathbb{X}} \sum_{i=1}^{N} -\log\mathbb{P}\left(\boldsymbol{y}(i)|\boldsymbol{x}\right) - \log\mathbb{P}\left(\boldsymbol{x}\right) = \tag{A7}$$

$$\arg\min_{\substack{\boldsymbol{R}\in\mathrm{SO}(3),\\ \boldsymbol{t}\in\mathbb{R}^3, \boldsymbol{c}\in\mathbb{R}^K,\\ \mathbf{1}^\mathsf{T}\boldsymbol{c}=1}} \sum_{i=1}^{N} \frac{w_i}{2}\left\|\boldsymbol{y}(i) - \boldsymbol{R}\sum_{k=1}^{K} c_k \boldsymbol{b}_k(i) - \boldsymbol{t}\right\|^2 \tag{A8}$$

$$+\frac{\lambda}{2}\|\boldsymbol{c}\|^2 + \text{constants} \tag{A9}$$

which, after dropping constant multiplicative and additive factors, can be seen to match eq. (3), proving the claim.

APPENDIX B
PROBLEM (23) IS A MAP ESTIMATOR WHEN THE MEASUREMENT NOISE IS HEAVY-TAILED

Here we prove that the optimization in eq. (23) with a truncated least square loss $\rho(r) = \min(r^2, \varepsilon^2)$ is a *maximum a posteriori* (MAP) estimator when the measurement noise $\boldsymbol{\epsilon}(i)$ in (2) follows a max-mixture distribution, where we replace

the tails of a Gaussian with a uniform distribution —a model we borrow from [1]. Mathematically:

$$\mathbb{P}\left(\boldsymbol{\epsilon}(i)\right) = \begin{cases} \kappa_\epsilon \exp\left(-\frac{1}{2}\|\boldsymbol{\epsilon}(i)\|^2\right), & \|\boldsymbol{\epsilon}(i)\| < \varepsilon, \\ \kappa_\epsilon \exp\left(-\frac{1}{2}\varepsilon^2\right), & \|\boldsymbol{\epsilon}(i)\| \in [\varepsilon, \alpha], \\ 0, & \text{otherwise}, \end{cases} \tag{A10}$$

where ε is the maximum noise for an inlier, κ_ϵ is a normalization constant, and α defines the support of the uniform distribution (both κ_ϵ and α are irrelevant for the derivation); in (A10) —without loss of generality— we assumed unit covariance for the Gaussian. Intuitively, eq. (A10) describes a Gaussian distribution for errors below ε, but for errors larger than ε the Gaussian tails have been substituted by a uniform distribution (observe that $\kappa_\epsilon \exp\left(-\frac{1}{2}\varepsilon^2\right)$ is a constant). Then the proof trivially follows from [1, Proposition 5] (the expression of the shape priors remains the same as Appendix A).

Appendix C
Closed-form Shape Estimation: Proof of Proposition 1

Fixing $\boldsymbol{R}$, the Lagrangian of the linearly constrained linear least squares problem (11) is:

$$\mathcal{L} = \left\|\bar{\boldsymbol{B}}\boldsymbol{c} - (\mathbf{I}_N \otimes \boldsymbol{R}^\mathsf{T})\bar{\boldsymbol{y}}\right\|^2 + \lambda\|\boldsymbol{c}\|^2 + \gamma(\mathbf{1}^\mathsf{T}\boldsymbol{c} - 1) \tag{A11}$$

where $\gamma \in \mathbb{R}$ is the multiplier associated with the constraint $\mathbf{1}^\mathsf{T}\boldsymbol{c} = 1$ [8]. Observe that problem (11) has a single equality constraint and trivially satisfies the linear independence constraint qualification (LICQ), therefore, any optimal solution must satisfy the following KKT conditions:

$$\nabla_{\boldsymbol{c}}\mathcal{L} = 2(\bar{\boldsymbol{B}}^\mathsf{T}\bar{\boldsymbol{B}} + \lambda\mathbf{I}_K)\boldsymbol{c} + \gamma\mathbf{1} - 2\bar{\boldsymbol{B}}^\mathsf{T}(\mathbf{I}_N \otimes \boldsymbol{R}^\mathsf{T})\bar{\boldsymbol{y}} = \mathbf{0} \tag{A12}$$

$$\nabla_\gamma\mathcal{L} = \mathbf{1}^\mathsf{T}\boldsymbol{c} - 1 = 0 \tag{A13}$$

which can be written compactly as the following linear system of equations:

$$\begin{bmatrix} 2(\bar{\boldsymbol{B}}^\mathsf{T}\bar{\boldsymbol{B}} + \lambda\mathbf{I}_K) & \mathbf{1} \\ \mathbf{1}^\mathsf{T} & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{c} \\ \gamma \end{bmatrix} = \begin{bmatrix} 2\bar{\boldsymbol{B}}^\mathsf{T}(\mathbf{I}_N \otimes \boldsymbol{R}^\mathsf{T})\bar{\boldsymbol{y}} \\ 1 \end{bmatrix}. \tag{A14}$$

Now let

$$\bar{\boldsymbol{H}} \triangleq 2(\bar{\boldsymbol{B}}^\mathsf{T}\bar{\boldsymbol{B}} + \lambda\mathbf{I}_K) \in \mathcal{S}^K_{++}, \tag{A15}$$

$$\boldsymbol{H} \triangleq \begin{bmatrix} \bar{\boldsymbol{H}} & \mathbf{1} \\ \mathbf{1}^\mathsf{T} & 0 \end{bmatrix} \in \mathcal{S}^{K+1}, \tag{A16}$$

where $\mathcal{S}^K_{++}$ denotes the set of positive definite matrices of size K. Note that the inverse of $\boldsymbol{H}$ exists because $\bar{\boldsymbol{H}}$ is positive definite and invertible ($\bar{\boldsymbol{H}}^{-1}$ is also positive definite):

$$\boldsymbol{H}^{-1} = \begin{bmatrix} \bar{\boldsymbol{H}}^{-1} - \frac{\bar{\boldsymbol{H}}^{-1}\boldsymbol{e}\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}}{\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}\mathbf{1}} & \frac{\bar{\boldsymbol{H}}^{-1}\mathbf{1}}{\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}\mathbf{1}} \\ \frac{\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}}{\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}\mathbf{1}} & -\frac{1}{\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}\mathbf{1}} \end{bmatrix}. \tag{A17}$$

Therefore the optimal $\boldsymbol{c}$ can be obtained from (A14) as:

$$\boldsymbol{c}^\star(\boldsymbol{R}) = 2\boldsymbol{G}\bar{\boldsymbol{B}}^\mathsf{T}(\mathbf{I}_N \otimes \boldsymbol{R}^\mathsf{T})\bar{\boldsymbol{y}} + \boldsymbol{g}, \tag{A18}$$

where

$$\boldsymbol{G} \triangleq \bar{\boldsymbol{H}}^{-1} - \frac{\bar{\boldsymbol{H}}^{-1}\mathbf{1}\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}}{\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}\mathbf{1}}, \quad \boldsymbol{g} \triangleq \frac{\bar{\boldsymbol{H}}^{-1}\mathbf{1}}{\mathbf{1}^\mathsf{T}\bar{\boldsymbol{H}}^{-1}\mathbf{1}} \tag{A19}$$

proving Proposition 1.

Appendix D
Certifiably Optimal Rotation Estimation: Proof of Proposition 2 and Corollary 3

Let us first develop the cost function of problem (15) as a quadratic function of $\boldsymbol{r} \triangleq \mathrm{vec}\left(\boldsymbol{R}\right)$:

$$\left\|\boldsymbol{M}(\mathbf{I}_N \otimes \boldsymbol{R}^\mathsf{T})\bar{\boldsymbol{y}} + \boldsymbol{h}\right\|^2 \tag{A20}$$

$$= \left\|\boldsymbol{M}\mathrm{vec}\left(\boldsymbol{R}^\mathsf{T}\boldsymbol{Y}\right) + \boldsymbol{h}\right\|^2 \tag{A21}$$

$$= \left\|\boldsymbol{M}(\boldsymbol{Y}^\mathsf{T} \otimes \mathbf{I}_3)\mathrm{vec}\left(\boldsymbol{R}^\mathsf{T}\right) + \boldsymbol{h}\right\|^2 \tag{A22}$$

$$= \left\|\boldsymbol{M}(\boldsymbol{Y}^\mathsf{T} \otimes \mathbf{I}_3)\boldsymbol{P}\boldsymbol{r} + \boldsymbol{h}\right\|^2 \tag{A23}$$

$$= \tilde{\boldsymbol{r}}^\mathsf{T}\boldsymbol{Q}\tilde{\boldsymbol{r}} \tag{A24}$$

where $\boldsymbol{P} \in \mathbb{R}^{9\times 9}$ is the following permutation matrix

$$(1,1,1),(2,4,1),(3,7,1), \tag{A25}$$

$$(4,2,1),(5,5,1),(6,8,1), \tag{A26}$$

$$(7,3,1),(8,6,1),(9,9,1), \tag{A27}$$

with the triplet (i,j,v) defining the nonzero entries of $\boldsymbol{P}$ (*i.e.,* $\boldsymbol{P}_{ij} = v$), such that:

$$\mathrm{vec}\left(\boldsymbol{R}^\mathsf{T}\right) \equiv \boldsymbol{P}\mathrm{vec}\left(\boldsymbol{R}\right) \tag{A28}$$

always holds, $\boldsymbol{Y}$ and $\tilde{\boldsymbol{r}}$ are defined as:

$$\boldsymbol{Y} \triangleq \begin{bmatrix} \bar{\boldsymbol{y}}(1) & \cdots & \bar{\boldsymbol{y}}(N) \end{bmatrix} \in \mathbb{R}^{3\times N}, \tag{A29}$$

$$\tilde{\boldsymbol{r}} \triangleq \begin{bmatrix} 1 & \boldsymbol{r}^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{10}, \tag{A30}$$

and $\boldsymbol{Q} \in \mathcal{S}^{10}$ can be assembled as follows:

$$\boldsymbol{Q} \triangleq \begin{bmatrix} \boldsymbol{h}^\mathsf{T}\boldsymbol{h} & \boldsymbol{h}^\mathsf{T}\boldsymbol{M}(\boldsymbol{Y}^\mathsf{T} \otimes \mathbf{I}_3)\boldsymbol{P} \\ \star & \boldsymbol{P}^\mathsf{T}(\boldsymbol{Y} \otimes \mathbf{I}_3)\boldsymbol{M}^\mathsf{T}\boldsymbol{M}(\boldsymbol{Y}^\mathsf{T} \otimes \mathbf{I}_3)\boldsymbol{P} \end{bmatrix}. \tag{A31}$$

Now that the objective function of (15) is quadratic in $\boldsymbol{r}$ ($\boldsymbol{R}$), we can write problem (15) equivalently as the *quadratically constrained quadratic program* (QCQP) in (17), where $\boldsymbol{A}_i \in \mathcal{S}^{10}, i = 1,\ldots,15$, are the constant matrices that define the quadratic constraints associated with $\boldsymbol{R} \in \mathrm{SO}(3)$ [84, Lemma

5]. For completeness, we give the expressions for $\boldsymbol{A}_i$'s:

$$\begin{aligned}
\boldsymbol{A}_0 &: (1,1,1)\\
\boldsymbol{A}_1 - \boldsymbol{A}_3 &: \text{ columns have unit norm}\\
\boldsymbol{A}_1 &: (1,1,1),(2,2,-1),(3,3,-1),(4,4,-1)\\
\boldsymbol{A}_2 &: (1,1,1),(5,5,-1),(6,6,-1),(7,7,-1)\\
\boldsymbol{A}_3 &: (1,1,1),(8,8,-1),(9,9,-1),(10,10,-1)\\
\boldsymbol{A}_4 - \boldsymbol{A}_6 &: \text{ columns are mutually orthogonal}\\
\boldsymbol{A}_4 &: (2,5,1),(3,6,1),(4,7,1)\\
\boldsymbol{A}_5 &: (2,8,1),(3,9,1),(4,10,1)\\
\boldsymbol{A}_6 &: (5,8,1),(6,9,1),(7,10,1)\\
\boldsymbol{A}_7 - \boldsymbol{A}_{15} &: \text{ columns form right-handed frame}\\
\boldsymbol{A}_7 &: (3,7,1),(4,6,-1),(1,8,-1)\\
\boldsymbol{A}_8 &: (4,5,1),(2,7,-1),(1,9,-1)\\
\boldsymbol{A}_9 &: (2,6,1),(1,10,-1),(3,5,-1)\\
\boldsymbol{A}_{10} &: (6,10,1),(1,2,-1),(7,9,-1)\\
\boldsymbol{A}_{11} &: (7,8,1),(5,10,-1),(1,3,-1)\\
\boldsymbol{A}_{12} &: (5,9,1),(1,4,-1),(6,8,-1)\\
\boldsymbol{A}_{13} &: (4,9,1),(3,10,-1),(1,5,-1)\\
\boldsymbol{A}_{14} &: (2,10,1),(1,6,-1),(4,8,-1)\\
\boldsymbol{A}_{15} &: (3,8,1),(2,9,-1),(1,7,-1)
\end{aligned}$$

where the triplets (i,j,v) define the *diagonal and upper triangular* nonzero entries of a symmetric matrix (*i.e.,* $\boldsymbol{A}_{ij} = \boldsymbol{A}_{ji} = v$ with $i \leq j$).

Appendix E
Shor's Semidefinite Relaxation and Relative Duality Gap

To see why problem (18) is a convex relaxation for problem (17), let us first create a matrix variable

$$\boldsymbol{X} = \tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}^{\mathsf{T}} \in \mathcal{S}^{10}, \tag{A32}$$

and notice that $\boldsymbol{X}$ satisfies

$$\boldsymbol{X} \succeq 0, \quad \operatorname{rank}(\boldsymbol{X}) = 1. \tag{A33}$$

Moreover, if $\boldsymbol{X} \succeq 0, \operatorname{rank}(\boldsymbol{X}) = 1$ then $\boldsymbol{X}$ must have a factorization of the form (A32). Therefore, the non-convex QCQP (17) is equivalent to the following rank-constrained matrix optimization problem:

$$\begin{aligned}
\min_{\boldsymbol{X}\in\mathcal{S}^{10}} \quad & \operatorname{tr}(\boldsymbol{Q}\boldsymbol{X}) && \text{(A34)}\\
\text{s.t.} \quad & \operatorname{tr}(\boldsymbol{A}_0\boldsymbol{X}) = 1, && \text{(A35)}\\
& \operatorname{tr}(\boldsymbol{A}_i\boldsymbol{X}) = 0, \forall i = 1,\ldots,15, && \text{(A36)}\\
& \boldsymbol{X} \succeq 0, && \text{(A37)}\\
& \operatorname{rank}(\boldsymbol{X}) = 1, && \text{(A38)}
\end{aligned}$$

where $\boldsymbol{A}_0 \in \mathcal{S}^{10}$ is an all-zero matrix except the top-left entry being 1 (to enforce that the first entry of $\tilde{\boldsymbol{r}}$ is 1), and we have used the fact that

$$\tilde{\boldsymbol{r}}^{\mathsf{T}}\boldsymbol{A}\tilde{\boldsymbol{r}} = \operatorname{tr}\left(\tilde{\boldsymbol{r}}^{\mathsf{T}}\boldsymbol{A}\tilde{\boldsymbol{r}}\right) = \operatorname{tr}\left(\boldsymbol{A}\tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}^{\mathsf{T}}\right) = \operatorname{tr}(\boldsymbol{A}\boldsymbol{X}). \tag{A39}$$

Now observe that the only nonconvex constraint in problem (A34) is the rank constraint (A38), and the SDP relaxation (18) is obtained by simply removing the rank constraint.

In practice, we solve the convex problem (18) and obtain an optimal solution $\boldsymbol{X}^\star$, if $\operatorname{rank}(\boldsymbol{X}^\star) = 1$, then the optimal solution of problem (18) is unique (the rationale behind this is that interior-point methods converge to a maximum rank solution [17]) and it actually satisfies the rank constraint that has been dropped. Therefore, in this situation, we say the convex relaxation is tight and the global optimal solution to the nonconvex problem (17) can be obtained from the rank-one factorization of $\boldsymbol{X}^\star$.

Relative duality gap. Checking if the solution is rank one can sometimes be sensitive to numerical thresholds, therefore, an alternative way to check the quality of the relaxation is to compute the relative duality gap. Let $\boldsymbol{X}^\star$ be a solution of the SDP relaxation (18) and let $f_{\text{SDP}} \triangleq \operatorname{tr}(\boldsymbol{Q}\boldsymbol{X}^\star)$ be the optimal cost. Let $\hat{\boldsymbol{r}} \in \mathrm{SO}(3)$ be a rounded solution from $\boldsymbol{X}^\star$ (the rounding can be done by closed-form projection to $\mathrm{SO}(3)$ [85]), and let $f_{\text{est}} \triangleq [1,\hat{\boldsymbol{r}}^{\mathsf{T}}]\boldsymbol{Q}[1,\hat{\boldsymbol{r}}^{\mathsf{T}}]^{\mathsf{T}}$ be the cost of the non-convex problem (17) evaluated at the rounded solution $\hat{\boldsymbol{r}}$, then we have:

$$f_{\text{SDP}} \leq f^\star \leq f_{\text{est}}, \tag{A40}$$

where $f^\star$ is the true global optimum of the nonconvex problem (17), the first inequality follows from the fact that problem (18) is a convex relaxation and the second inequality follows from the fact that $f^\star$ is the global minimum. We then compute the relative duality gap

$$\eta \triangleq \frac{f_{\text{est}} - f_{\text{SDP}}}{f_{\text{est}}}, \tag{A41}$$

which is informative of the suboptimality of the rounded solution. In particular, if $\eta \approx 0$, then $\hat{\boldsymbol{r}}$ is certified to be the globally optimal solution.

Appendix F
Minimum and Maximum Distances between Convex Hulls

Recall from eq. (21) the definitions of $b_{ij}^{\min}$ and $b_{ij}^{\max}$:

$$b_{ij}^{\min} = \min_{\boldsymbol{c}\geq 0, \mathbf{1}^{\mathsf{T}}\boldsymbol{c}=1} \left\| \sum_{k=1}^{K} c_k(\boldsymbol{b}_k(j) - \boldsymbol{b}_k(i)) \right\|, \tag{A42}$$

$$b_{ij}^{\max} = \max_{\boldsymbol{c}\geq 0, \mathbf{1}^{\mathsf{T}}\boldsymbol{c}=1} \left\| \sum_{k=1}^{K} c_k(\boldsymbol{b}_k(j) - \boldsymbol{b}_k(i)) \right\|, \tag{A43}$$

and let us use the following shorthand:

$$\boldsymbol{b}_{k,ij} \triangleq \boldsymbol{b}_k(j) - \boldsymbol{b}_k(i), \tag{A44}$$

$$\boldsymbol{B}_{ij} \triangleq \begin{bmatrix} \boldsymbol{b}_{1,ij} & \cdots & \boldsymbol{b}_{K,ij} \end{bmatrix} \in \mathbb{R}^{3\times K}, \tag{A45}$$

to write problems (A42) and (A43) compactly as:

$$b_{ij}^{\min} = \min_{\boldsymbol{c}\geq 0, \mathbf{1}^{\mathsf{T}}\boldsymbol{c}=1} \|\boldsymbol{B}_{ij}\boldsymbol{c}\|, \quad b_{ij}^{\max} = \max_{\boldsymbol{c}\geq 0, \mathbf{1}^{\mathsf{T}}\boldsymbol{c}=1} \|\boldsymbol{B}_{ij}\boldsymbol{c}\|. \tag{A46}$$

Compute $b_{ij}^{\max}$. Because $\|\boldsymbol{B}_{ij}\boldsymbol{c}\|$ is a convex function of $\boldsymbol{c}$, and the maximum of a convex function over a polyhedral set (in our case, the standard simplex $\Delta_K \triangleq \{\boldsymbol{c} \in \mathbb{R}^K : \boldsymbol{c} \geq 0, \mathbf{1}^\mathsf{T}\boldsymbol{c} = 1\}$) is always obtained at one of the vertices of the polyhedron [61, Corollary 32.3.4], we have:

$$b_{ij}^{\max} = \max_k \|\boldsymbol{b}_{k,ij}\|, \tag{A47}$$

since the vertices of Δ_K are the vectors $\boldsymbol{e}_k, k = 1, \ldots, K$, where $\boldsymbol{e}_k$ is one at its k-th entry and zero anywhere else.

Compute $b_{ij}^{\min}$. Observe that computing the minimum of $\|\boldsymbol{B}_{ij}\boldsymbol{c}\|$ is equivalent to computing the minimum of $\|\boldsymbol{B}_{ij}\boldsymbol{c}\|^2 = \boldsymbol{c}^\mathsf{T}(\boldsymbol{B}_{ij}^\mathsf{T}\boldsymbol{B}_{ij})\boldsymbol{c}$ because the quadratic function $f(x) = x^2$ is monotonically increasing in the interval $[0, \infty]$, and hence we first solve the following convex quadratic program (QP):

$$\min_{\boldsymbol{c}\in\mathbb{R}^K} \quad \boldsymbol{c}^\mathsf{T}(\boldsymbol{B}_{ij}^\mathsf{T}\boldsymbol{B}_{ij})\boldsymbol{c} \tag{A48}$$

$$\text{s.t.} \quad \boldsymbol{c} \geq 0, \quad \mathbf{1}^\mathsf{T}\boldsymbol{c} = 1 \tag{A49}$$

and then compute $b_{ij}^{\min} = \|\boldsymbol{B}_{ij}\boldsymbol{c}^\star\|$ from the solution $\boldsymbol{c}^\star$ of the QP. Note that the QP (A48) can be solved in milliseconds for large K, so pre-computing $b_{ij}^{\min}$ for all $1 \leq i < j \leq N$ is still tractable even when N is large.

Appendix G
Alternation Approach

In Sections IV-B-IV-C of the main paper, we presented a certifiably optimal solver to solve the joint shape and rotation $(\boldsymbol{c}, \boldsymbol{R})$ problem (8) (after eliminating the translation $\boldsymbol{t}$). Here we describe a baseline method that solves problem (8) using *alternating minimization* (Altern), a heuristic that is popular in related works on 3D shape reconstruction from 2D landmarks [26, 43, 58], but offers no optimality guarantees. Towards this goal, let us denote the cost function of (8) as $f(\boldsymbol{R}, \boldsymbol{c})$; the Altern method starts with an initial guess $(\boldsymbol{R}^{(0)}, \boldsymbol{c}^{(0)})$ (default $\boldsymbol{R}^{(0)} = \mathbf{I}_3, \boldsymbol{c}^{(0)} = \mathbf{0}$), and performs the following two steps at each iteration τ:

1) Optimize $\boldsymbol{c}$:

$$\boldsymbol{c}^{(\tau)} = \underset{\boldsymbol{c}\in\mathbb{R}^K, \mathbf{1}^\mathsf{T}\boldsymbol{c}=1}{\arg\min} \; f(\boldsymbol{R}^{(\tau-1)}, \boldsymbol{c}), \tag{A50}$$

   which is a linearly constrained linear least squares problem and can be solved by the closed-form solution (12).

2) Optimize $\boldsymbol{R}$:

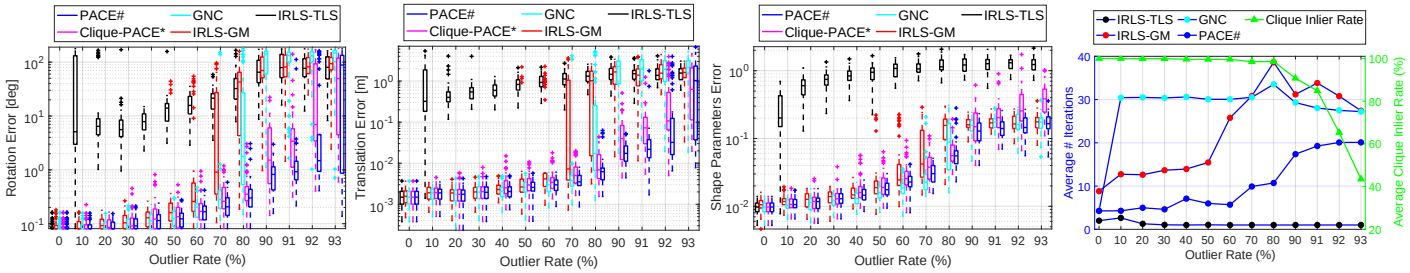
$$\boldsymbol{R}^{(\tau)} = \underset{\boldsymbol{R}\in\mathrm{SO}(3)}{\arg\min} \; f(\boldsymbol{R}, \boldsymbol{c}^{(\tau)}), \tag{A51}$$

   which can be cast as an instance of Wahba's problem [83] and can be solved in closed form using singular value decomposition [49].

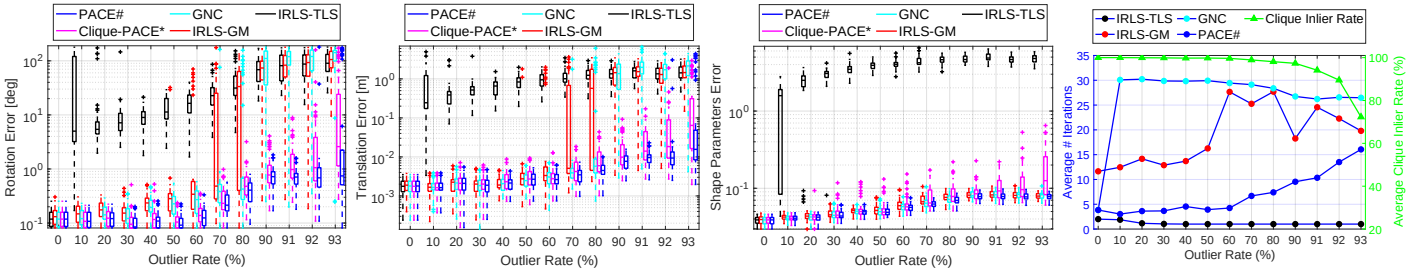
The Altern method stops when the cost function converges, *i.e.,* $|f(\boldsymbol{R}^{(\tau)}, \boldsymbol{c}^{(\tau)}) - f(\boldsymbol{R}^{(\tau-1)}, \boldsymbol{c}^{(\tau-1)})| < \epsilon$ for some small threshold $\epsilon > 0$, or when τ exceeds the maximum number of iterations (*e.g.,* 1000).

Appendix H
Extra Experimental Results

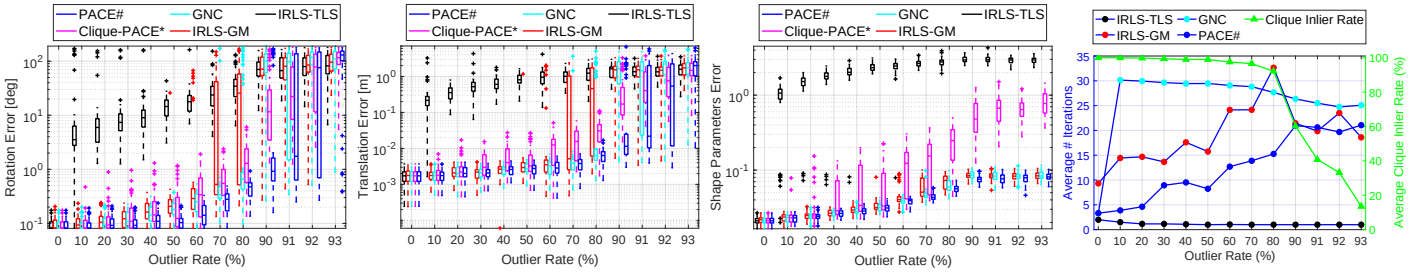
In Section VI-A, we demonstrated the robustness of PACE# to 92% outlier rates when $N = 100$, $K = 10$ and $r = 0.1$. Here we show extra results when K and r are increased. Fig. A1(a) shows the results for $N = 100$, $K = 10$ and $r = 0.2$. One can see that as the intra-class variation radius r is increased, the compatibility check becomes less effective, leading to a slight decrease in the robustness of PACE# against outliers — PACE# is still robust up to 90% outlier rate while has two failures at 91% outlier rate. However, PACE# still outperforms IRLS-TLS and IRLS-GM by a large margin. Fig. A1(b) shows the results for $N = 100$, $K = 50$ and $r = 0.1$. We see that PACE# is robust up to 91% outlier rate while encounters two failures at 92% outlier rate. Finally, when $K = 50$ and $r = 0.2$ (Fig. A1(c)), PACE# is robust to 80% outlier rate.



(a) Robustness of PACE# against increasing outliers on random simulated data: $N = 100$, $K = 10$, $r = 0.2$.



(b) Robustness of PACE# against increasing outliers on random simulated data: $N = 100$, $K = 50$, $r = 0.1$.



(c) Robustness of PACE# against increasing outliers on random simulated data: $N = 100$, $K = 50$, $r = 0.2$.

Fig. A1: Performance of PACE# compared to baselines in simulated experiments with different number of CAD models K and variation radius r. (a) The intra-class variation radius is increased to $r = 0.2$. (b) The number of CAD models is increased to $K = 50$. (c) $K = 50$ and $r = 0.2$. Each boxplot (and lineplot) reports statistics computed over 50 Monte Carlo runs.

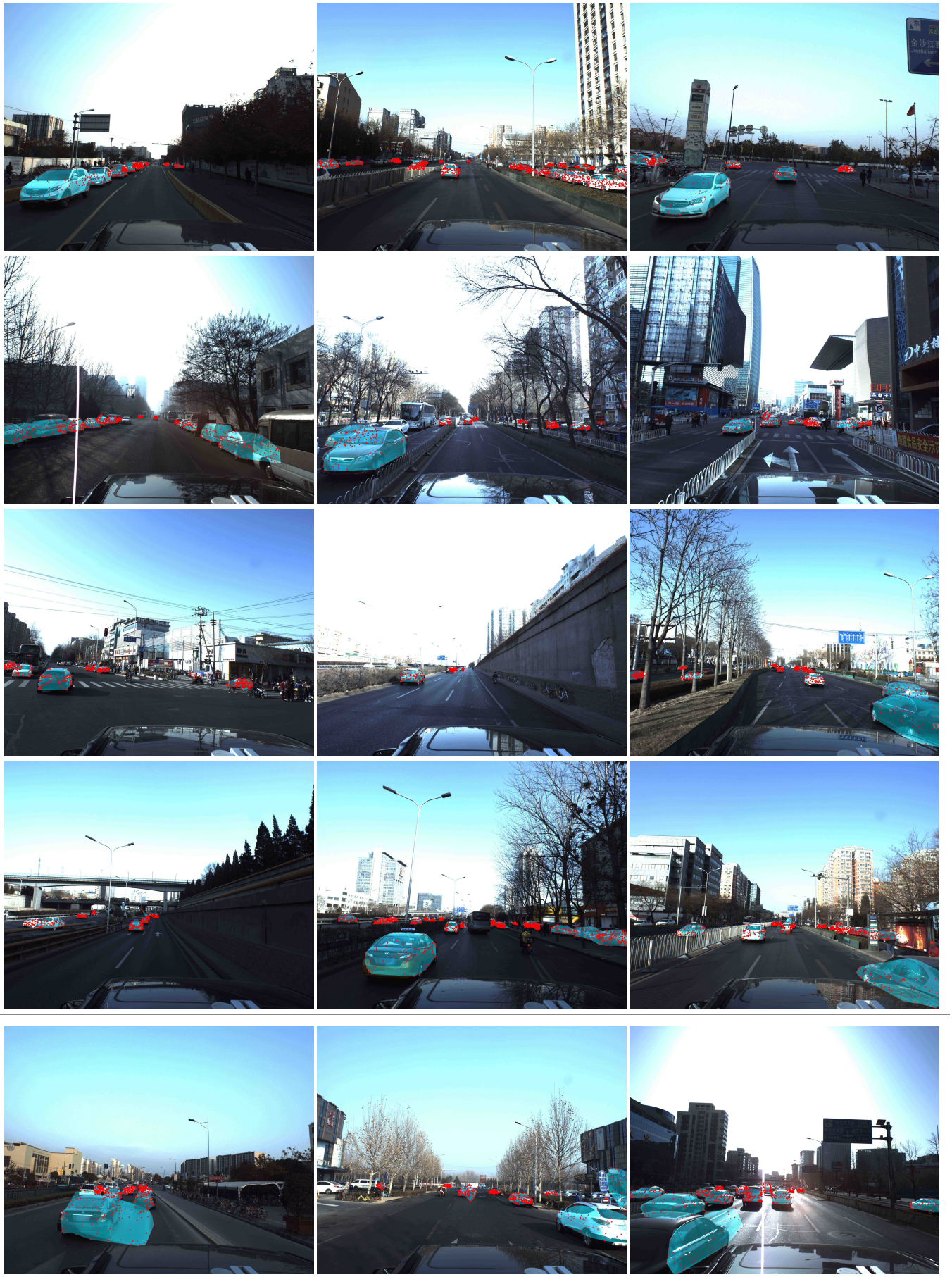


Fig. A1: Qualitative results: overlay of estimated vehicle pose and shape on the images from the ApolloScape dataset. The images are manually selected out of the 5277 images in the dataset to showcase successful vehicle localization (top 4 rows) as well as failure cases (last row).