2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

# Robust Object-based SLAM for High-speed Autonomous Navigation

Kyel Ok[*1], Katherine Liu[*1,2], Kris Frey[2], Jonathan P. How[2], and Nicholas Roy[1,2]

***Abstract*— We present Robust Object-based SLAM for High-speed Autonomous Navigation (ROSHAN), a novel approach to object-level mapping suitable for autonomous navigation. In ROSHAN, we represent objects as ellipsoids and infer their parameters using three sources of information – bounding box detections, image texture, and semantic knowledge – to overcome the observability problem in ellipsoid-based SLAM under common forward-translating vehicle motions. Each bounding box provides four planar constraints on an object surface and we add a fifth planar constraint using the texture on the objects along with a semantic prior on the shape of ellipsoids. We demonstrate ROSHAN in simulation where we outperform the baseline, reducing the median shape error by $83\%$ and the median position error by $72\%$ in a forward-moving camera sequence. We demonstrate similar qualitative result on data collected on a fast-moving autonomous quadrotor.**

## I. Introduction

We are interested in autonomous surveillance missions using a fast-moving micro air vehicle (MAV). We would like to use a camera to build a map of the world that contains both semantic labels for scene understanding and geometric information for navigating around obstacles.

Past work in building vision-based geometric maps of the world, i.e., vision-based simultaneous localization and mapping (vSLAM), focuses on constructing accurate geometric representations of the world, but is often inadequate for real-time path planning. Sparse [1], [2] and semi-sparse [3]–[5] methods employ a point-cloud representation of the world for computational efficiency, but the sparsity of this representation impedes collision-checking. Dense methods [6], [7] address the problem of sparsity by using a volumetric or mesh-based representation, but these methods often have a high computational burden, while the reconstruction quality deteriorates in scenes with low texture.

Assuming high-quality computationally inexpensive monocular dense reconstructions, given additional semantic scene segmentation, labelled dense geometric maps can be built online [8], [9]. However, semantic segmentation, which outlines a tight boundary, can be computationally expensive [10] compared to some object detectors [11], [12] that only infer bounding box approximations of object detections. Utilizing such inexpensive object detectors, some previous work [13]–[15] detects and explicitly models an object of interest as a single entity, relaxing the constraint that an object must have texture on its entire surface for a good reconstruction. However, these object-based methods do not focus on building an online map of an unknown environment for collision avoidance, and assume either the existence of precise models of objects [14]–[16], or use bounding box detections as noisy centroid measurements [13] to construct only a sparse representation of the world. This lack of online *volumetric* reconstruction of objects impedes collision-checking desired on autonomous vehicles.

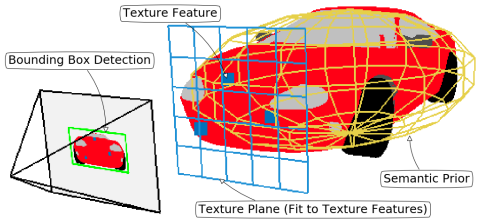



Fig. 1: In ROSHAN, we combine bounding box detections (green), texture planes (blue), and semantic knowledge of the shape of objects (yellow) to achieve an ellipsoid-based object SLAM system robust to undesirable camera motions.

For the purpose of obstacle avoidance and scene understanding in an unknown world, exact models of objects may not be required as long as approximate models can sufficiently support collision-checking. Some semantic mapping approaches [17], [18] build lightweight approximations of objects offline by fitting bounding box measurements to a low-dimensional parametric model of primitive shapes. While an online version of this approach seems suitable for autonomous navigation, there is an observability problem in using only the bounding boxes to constrain all object models, when common types of vehicle motions such as straight line motions do not generate diverse viewpoints of the objects; this is similar to the problem in point-based monocular SLAM [19], where the depths of points triangulated with cameras on a small baseline are difficult to observe.

In order to better constrain an object-based SLAM system that lacks diversity in viewpoints, we show how to use two additional sources of information: texture on objects that can be used to infer the distance to the objects and semantic knowledge of shapes of objects that can mitigate the scale unobservability problem in monocular cameras [20]. While similar to recent work [21] which uses surface normals from RGB-D cameras to further constrain quadrics, we focus on adding only the information available in a monocular camera.

We propose robust object-based SLAM for high-speed autonomous navigation (ROSHAN), where we represent semantically-meaningful objects *volumetrically* as ellipsoids,

[*]The first two authors contributed equally to this paper.

[1]Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA {kyelok,katliu,nickroy}@csail.mit.edu

[2]Department of Aeronautics and Astronautics, MIT, Cambridge, MA 02139, USA {kfrey, jhow}@mit.edu

and infer the parameters of the ellipsoids online using three sources of information: bounding box detections, texture, and semantic shape constraints. We make an improvement to the state-of-the-art bounding box measurement model [18], introduce a differentiable closed-form measurement model for texture, describe a semantic shape prior, propose a single measurement initialization scheme useful on a fast-moving vehicle, and contrary to modern offline methods [17], [18] do not assume known data associations or batch optimization.

Finally, we demonstrate the advantages of ROSHAN in simulation using 50 randomly generated maps of ellipsoids, where we outperform the baseline, reducing the median error on the shape estimates by $83\%$ and the median error on the position estimates by $72\%$ when compared to the baseline in a forward-moving camera sequence. In addition, we present promising results running ROSHAN real-time on simulated and real autonomous high-speed flight sequences.

## II. Problem Overview

In ROSHAN, we represent objects as ellipsoids and infer their parameters using object detections, object texture, and semantic knowledge in a SLAM framework. In this section, we first discuss the strengths of our landmark representation then formulate our object-based SLAM problem.

### A. Ellipsoids as Object Representation

We choose the ellipsoid representation, a specific form of quadric representation [22] as the low-dimensional parametric form of our objects. Similar to [7], [17], we minimally parametrize the ellipsoid with 9 independent parameters that represent the orientation $\boldsymbol{R} \in \mathrm{SO}(3)$, position $\boldsymbol{t} \in \mathbb{R}^3$, and shape $\boldsymbol{d} \in \mathbb{R}^3$ of the ellipsoid. While there are two forms of ellipsoids, $\boldsymbol{Q}$ and the dual-form $\boldsymbol{Q}^* = \mathrm{adjoint}(\boldsymbol{Q})$, we are interested in the dual-form $\boldsymbol{Q}^* \in \mathbb{E}^{4\times4}$, where $\mathbb{E}^{4\times4}$ represents the subset of all $4\times4$ symmetric matrices defined by

$$\boldsymbol{Q}^* = \begin{bmatrix} \boldsymbol{R}\boldsymbol{D}\boldsymbol{R}^T - \boldsymbol{t}\boldsymbol{t}^T & -\boldsymbol{t} \\ -\boldsymbol{t}^T & -1 \end{bmatrix}, \tag{1}$$

where $\boldsymbol{D} \in \mathbb{R}^{3\times3}$ is a positive diagonal shape matrix with the diagonal entries formed with regularized squared shape parameters, i.e., $\boldsymbol{D}_{i,i} = \boldsymbol{d}_i^2 + \gamma$, where $\gamma \in \mathbb{R}$ is a regularization constant enforcing a minimum shape.

While an ellipsoid is only a rough approximation of an object in 3D, a strong advantage of the ellipsoid representation is that its entire parametrization can be constrained using only bounding box measurements from camera images. This property of ellipsoids comes from the dual-form where all homogeneous planes $\boldsymbol{\pi}_k \in \mathbb{R}^4$ tangent to the dual-form of an ellipsoid $\boldsymbol{Q}_j^*$ must obey

$$\boldsymbol{\pi}_k^T \boldsymbol{Q}_j^* \boldsymbol{\pi}_k = 0. \tag{2}$$

This system of equations, when solved as a function of the vehicle pose $\boldsymbol{x}_{t_k} \in \mathrm{SE}(3)$ and the observed ellipsoid $\boldsymbol{Q}_{j_k}^*$ as illustrated in section III-A, forms a closed-form differentiable bounding box measurement model

$$\hat{\boldsymbol{B}}_k = h_{bb}(\boldsymbol{Q}_{j_k}^*, \boldsymbol{x}_{t_k}; \boldsymbol{K}), \tag{3}$$

where $\boldsymbol{K} \in \mathbb{R}^{3\times3}$ in the camera intrinsic matrix and $\hat{\boldsymbol{B}}_k \in \mathbb{R}^4$ is the predicted bounding box measurement. While the family of quadrics all share the same smooth measurement model, we specifically limit our landmarks to ellipsoids to further constrain the landmarks without losing the ability to approximate objects for the purpose of collision-checking.

Coupled with computationally inexpensive object detectors [11], [12], the above closed-form measurement model allows for the use of readily available bounding box detections as the only source of measurements to fully constrain vehicle poses and approximate object volumes. This property makes the ellipsoid representation attractive for graph-based SLAM [23] formulations, and is similar to the property of point-based landmarks in feature-based SLAM [1] that associated feature detections in camera images can be the only source of information to constrain the entire system.

### B. SLAM Formulation

We would like to solve for all ellipsoidal approximations of objects $\boldsymbol{\mathcal{Q}} = \{\boldsymbol{Q}_j^*\}_{j=0}^J$ with $J$ objects of interest, and $T$ poses of the vehicle $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_t\}_{t=0}^T$, where $\boldsymbol{x}_t \in \mathrm{SE}(3)$. We are given $T$ images $\boldsymbol{\mathcal{I}} = \{\boldsymbol{I}_t\}_{t=0}^T$ with $\boldsymbol{I}_t : \Omega \in \mathbb{N}^2 \to \mathbb{R}$, where $\Omega$ is the image pixel domain. Using an object detector, we extract $K$ bounding box measurements of objects $\boldsymbol{\mathcal{B}} = \{\boldsymbol{B}_k \in \Omega^2\}_{k=0}^K$ along with the semantic class labels $\boldsymbol{\mathcal{C}} = \{c_k \in \mathbb{N}\}_{k=0}^K$, where each bounding box is parametrized by two pixel locations representing the opposite corners of the bounding box. We extract high-gradient features [24] from the texture of the objects in images, and fit a homogeneous plane $\boldsymbol{\pi}_d^t \in \mathbb{R}^4$ to the triangulated locations of the features of each object; these $D$ planes $\boldsymbol{\Pi}^t = \{\boldsymbol{\pi}_d^t\}_{d=0}^D$ that we call *texture planes*, e.g. the blue plane in Fig. 1, represent measurements of the distance between the cameras and the camera-facing sides of objects. Assuming a uniform prior on the measurements and independence assumptions between all measurements, we write our object-level SLAM problem as

$$\begin{aligned} P(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Q}} | \boldsymbol{\mathcal{B}}, \boldsymbol{\Pi}^t, \boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{C}}) \propto & \underbrace{\prod_{k=0}^{K} P(\boldsymbol{B}_k | \boldsymbol{Q}_{j_k}^*, \boldsymbol{x}_{t_k})}_{\text{Bounding Box (III-B)}} \\ & \underbrace{\prod_{d=0}^{D} P(\boldsymbol{\pi}_d^t | \boldsymbol{Q}_{j_d}^*, \boldsymbol{x}_{t_d})}_{\text{Texture (III-C)}} \underbrace{\prod_{j=0}^{J} P(\boldsymbol{Q}_j^* | c_j)}_{\text{Semantic Prior (III-D)}} \underbrace{\prod_{t=0}^{T} P(\boldsymbol{x}_t | \boldsymbol{I}_{0:t})}_{\text{Pose Prior}}, \end{aligned} \tag{4}$$

where we assume that the data-association problem has been pre-solved (implementation details discussed in section V-A), i.e., that the associated indices $j_k$ and $j_d$ for objects and $t_k$ and $t_d$ for poses are known for each of the measurements $\boldsymbol{B}_k$ and $\boldsymbol{\pi}_d^t$, and that the class labels $c_j$ for ellipsoids are deduced from labels $c_k$ of bounding boxes.

We can then obtain optimal estimates of vehicle poses $\boldsymbol{\mathcal{X}}^*$ and objects $\boldsymbol{\mathcal{Q}}^*$ by maximizing the posterior probability

$$\boldsymbol{\mathcal{X}}^*, \boldsymbol{\mathcal{Q}}^* = \underset{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Q}}}{\arg\max}\, P(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Q}} | \boldsymbol{\mathcal{B}}, \boldsymbol{\Pi}^t, \boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{C}}). \tag{5}$$

In the following sections, we discuss the details of the bounding box measurement model (III-B), texture plane

measurement model (III-C) and the semantic prior on the ellipsoids (III-D) to demonstrate how multiple sources of information can be combined to constrain ellipsoidal approximations of objects. However, in this work, we assume an external vision-based[1] localization system $f_{pose}$ [19], [25] that produces pose estimates $\boldsymbol{x}_t = f_{pose}(\boldsymbol{I}_{0:t})$ to be loosely-coupled with our system and incorporate the MAP estimates along with a heuristic covariance as priors on our vehicle poses. In the next section, we first describe a limitation in the state-of-the-art bounding box measurement model [18], and suggest an improved bounding box measurement model.

## III. ROSHAN

In ROSHAN, we combine bounding box measurements, texture plane measurements, and semantic shape priors in an online optimization framework to realize an object-level SLAM system that is robust under undesirable vehicle motions. Before introducing the two additional sources of information, texture and semantic knowledge, we first revisit the state-of-the-art bounding box measurement model [18].

### A. Geometric Bounding Box Measurement Model

The projection of a dual-form of a quadric on a camera plane is called a dual-conic $\boldsymbol{G}^* \in \mathbb{R}^{3\times3}$, and has a similar property that all tangent lines must obey

$$\boldsymbol{l}_h^T \boldsymbol{G}^* \boldsymbol{l}_h = 0, \tag{6}$$

where $\boldsymbol{l}_h \in \mathbb{R}^3$ is a homogeneous form of a line. Since a dual-form of a quadric can be projected to a dual-conic [18] by

$$\boldsymbol{G}^* = \boldsymbol{K}[\boldsymbol{R}_t|\boldsymbol{t}_t]\boldsymbol{Q}_j^*[\boldsymbol{R}_t|\boldsymbol{t}_t]^T\boldsymbol{K}^T, \tag{7}$$

where $\boldsymbol{K} \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix, $\boldsymbol{R}_t \in \mathrm{SO}(3)$ is the rotation, and $\boldsymbol{t}_t \in \mathbb{R}^3$ is the translational portion of the pose $\boldsymbol{x}_t \in \mathrm{SE}(3)$, we can solve Eq. 6 in closed-form for the bounding box edges $\boldsymbol{l}_u = [1, 0, -u]$ and $\boldsymbol{l}_v = [0, 1, -v]$, i.e.,

$$\begin{aligned} \hat{u}_{min}, \hat{u}_{max} &= \boldsymbol{G}^*_{1,3} \pm \sqrt{{\boldsymbol{G}^*_{1,3}}^2 - \boldsymbol{G}^*_{1,1}\boldsymbol{G}^*_{3,3}}, \\ \hat{v}_{min}, \hat{v}_{max} &= \boldsymbol{G}^*_{2,3} \pm \sqrt{{\boldsymbol{G}^*_{2,3}}^2 - \boldsymbol{G}^*_{2,2}\boldsymbol{G}^*_{3,3}}, \end{aligned} \tag{8}$$

to form the closed-form measurement model in Eq. 3, where the predicted bounding box is a collection of these edge locations, i.e., $\hat{\boldsymbol{B}}_k = [\hat{u}_{min}, \hat{u}_{max}, \hat{v}_{min}, \hat{v}_{max}]_k$.

### B. ROSHAN Bounding Box Measurement Model

The assumption that each bounding box edge measurement $\boldsymbol{l}_h$ projects to a plane $\boldsymbol{\pi}_h$ tangent to the object of interest is broken in the case of partial, occluded, or truncated detections. A naive approach to using bounding boxes as measurements might simply keep all measurements, and hope that enough additional measurements will be made to mitigate the erroneous measurements. Nicholson et al. [18] present a truncated measurement model that ignores the portion of the measurement error that is outside of the image boundaries. When an object is well estimated, the truncated geometric model does reduce false measurement errors on edges that do not constrain the object by recognizing that the measured bounding box edge is the best observation the object detector can make. However, the truncated measurement model underestimates error in cases where the instantaneous bounding box estimate of the object position in the image plane is poor and a measured bounding box edge *is* in fact a constraining edge. For example, if the true object projects entirely into the image, but the instantaneous estimate of that object in the image plane is an overestimate that extends off the image, the truncated model will underestimate the error.

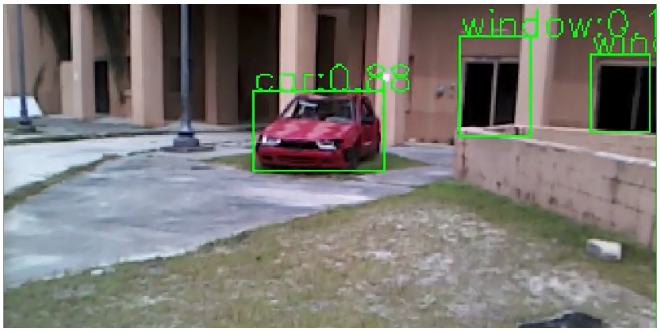



Fig. 2: An example of bounding box detections containing different types of non-constraining edges. The right edge of the rightmost window is a non-constraining edge at an image boundary, and the right edge of the leftmost window is a non-constraining edge formed by an occlusion between two objects: a pillar and a window. The car in the middle is tightly detected, as expected in the nominal case.

In ROSHAN, we first classify a bounding box edge as constraining (tangent to the object) or non-constraining (not tangent to the object) based on the proximity to the closest image boundary, before adding the edge as a constraint on the detected object. As shown in Fig. 2, we observe that a non-constraining edge can be formed both near the image boundaries and the occlusion boundaries between objects. However, as is the case of the truncated measurement model [18], we focus on identifying only the non-constraining edges near the image boundaries and leave potential ways to identify occlusions between objects, such as using relative depth from optical flow [26] or learning [27], as future work. Once a bounding box edge is classified as non-constraining based on the distance to the closest image boundary, instead of applying a truncated measurement model, we simply discard the edge, realizing that it is not an actual constraint.

Note that the closed-form measurement model in Eq. 8 has imaginary solutions when the term under the square root is negative. Geometrically, the imaginary solutions represent a camera being inside or axis-aligned with an observed ellipsoid, which may happen when the estimates of the ellipsoid parameters move during the optimization. In the case of this degeneracy, we set the measurement error to be high to discourage the iterative optimizer from stepping towards the degenerate solution; an alternative way would be to add an explicit cost such as the inverse barrier cost [28].

[1]Note that some vSLAM systems require additional sensors such as an IMU that could be added to our loosely-coupled formulation.

### C. Texture Plane Measurement Model

While bounding box measurements from diverse viewpoints can fully constrain an ellipsoid, given any *single* viewpoint, there are parameters of an ellipsoid that a bounding box measurement simply cannot observe. This is similar to the case in feature-based monocular SLAM where in any single image, a 2D landmark detection can only constrain the bearing of the landmark, but not the depth [19]. Similarly, a bounding box detection, which is a set of 4 orthogonal planar constraints induced by each of the bounding box edges, cannot fully constrain an ellipsoid inside a cuboid, i.e., fully constrain the volume of the ellipsoid, without two additional orthogonal planes for the missing faces of the cuboid.

However, there is a fifth measurable plane that is parallel to the camera image plane and fit to the high-gradient texture on the object. This plane that we refer to as the *texture plane* can be measured using triangulated feature points on the surface of the object, i.e., detected inside the bounding box, with co-observations in two or more cameras. Assuming that the triangulated feature points are all observations of the same tangent plane $\hat{\boldsymbol{\pi}}^t{}_d = [0, 0, 1, -\hat{z}]$, we can utilize the plane exactly the same way that bounding box planes are used to constrain an ellipsoid, i.e., solve the system of equations

$$[0, 0, 1, -\hat{z}]^T \left([\boldsymbol{R}_t|\boldsymbol{t}_t]\boldsymbol{Q}^*_j[\boldsymbol{R}_t|\boldsymbol{t}_t]^T\right)[0, 0, 1, -\hat{z}] = 0, \quad (9)$$

and obtain the predicted pseudo-measurement of the texture plane $\hat{z}$ as a differentiable closed-form solution, i.e.,

$$\hat{\boldsymbol{\pi}}^t{}_d = h_{tp}(\boldsymbol{Q}^*_{j_d}, \boldsymbol{x}_{t_d}; \boldsymbol{K}). \quad (10)$$

This additional texture plane helps better constrain our SLAM system, when the vehicle motion is not orbital and diverse viewpoints of objects cannot be guaranteed.

### D. Semantic Shape Prior

While the texture plane introduces a fifth plane to constrain an ellipsoid, for any single viewpoint there is one more orthogonal plane needed to fully constrain the volume of the ellipsoid. In the absence of this plane or a different view, the scale of the ellipsoid is ambiguous and the ellipsoid may be arbitrarily long on the other side of the texture plane.

To mitigate this problem of scale unobservability, we introduce *semantic* priors on the ellipsoids where we assume a semantically-informed Gaussian priors on the shape $\boldsymbol{d} \in \mathbb{R}^3$ and uniform priors on the position and the orientation of ellipsoids. While the semantic priors could be learned from large data sets as done in [29], we observe that many objects of interest are relatively consistent in size to allow a model-free specification using standard sizes. In this work, we create a function $h_{shape}$ using publicly available data on the metric shape of things to approximate the mean $\mu_{c_j} \in \mathbb{R}^3$ based on the class label $c_j \in \mathbb{N}$, i.e., $\mu_{c_j} = h_{shape}(c_j)$, and specify a diagonal covariance matrix $\Sigma_{c_j} \in \mathbb{R}^{3\times3}$ per object class to reflect the degree of consistency in the shape of objects. In our real-world and simulated flight experiments, we use the dimensions of a Toyota Camry as a reasonable mean of the prior, with the largest covariance on the length of the car to account for longer size cars.

### E. Single Image Initialization

Similar to the inverse depth initialization [30] of point-based landmarks, we can also initialize ellipsoids using a single bounding box measurement without having to do the delayed initialization in [18], allowing ROSHAN to quickly perceive and avoid obstacles during high-speed flight.

To realize a fast initialization scheme for the full 9 parameters of an ellipsoid, we make three reasonable assumptions. First, we assume that the position of the ellipsoid is somewhere along the camera ray that passes through the center of the bounding box [13]; the depth along this ray is estimated to be at an experimentally chosen average scene depth as done in [2]. Second, while there are single-image object orientation estimators [31], we assume the initial orientation to be identity for simplicity. Lastly, we assume the shape of the ellipsoid to be at the mean of the semantic shape prior.

Given these assumptions, we initialize an ellipsoid with the first detection, trading off the accuracy in our initial estimates for a faster perception. In ROSHAN, the inaccuracy in the initial estimates is mitigated by the faster converging bounding box model discussed in the previous sections.

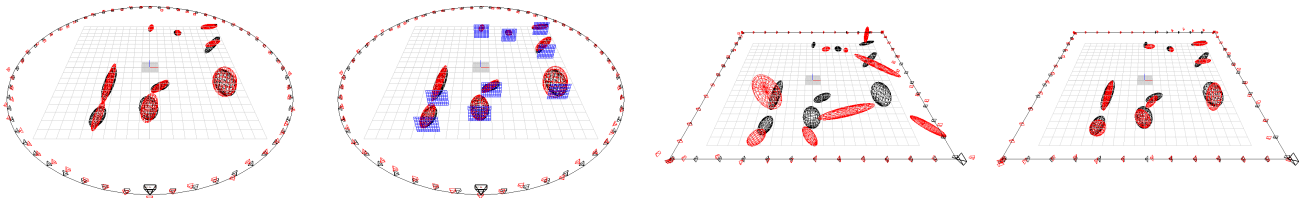
### F. Online Optimization

Assuming Gaussian measurement and process models, we can write Eq. 4 as a nonlinear least-squares problem [23]:

$$\begin{aligned}\boldsymbol{\mathcal{X}}^*, \boldsymbol{\mathcal{Q}}^* &= \underset{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{Q}}}{\arg\min} \; -\log P(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Q}}|\boldsymbol{\mathcal{B}}, \boldsymbol{\Pi}^t, \boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{C}}) \\ &= \underset{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{Q}}}{\arg\min} \Big\{ \sum_{t=0}^{T} \|f_{pose}(\boldsymbol{I}_{0:t}) - \boldsymbol{x}_t\|^2_{\Sigma_{o_t}} + \\ &\sum_{k=0}^{K} \|h_{bb}(\boldsymbol{Q}^*_{j_k}, \boldsymbol{x}_{t_k}; \boldsymbol{K}) - \boldsymbol{B}_k\|^2_{\Sigma_{b_k}} + \\ &\sum_{d=0}^{D} \|h_{tp}(\boldsymbol{Q}^*_{j_d}, \boldsymbol{x}_{t_d}; \boldsymbol{K}) - \boldsymbol{\pi}^t_d\|^2_{\Sigma_{t_d}} + \\ &\sum_{j=0}^{J} \|h_{shape}(c_j) - d(\boldsymbol{Q}^*_j)\|^2_{\Sigma_{c_j}} \Big\},\end{aligned} \quad (11)$$

where $\|\cdot\|^2_\Sigma$ is the Mahalanobis norm that directly scales the measurement error inversely proportional to the square root of the covariance term $\Sigma$. The covariance on the pose prior $\Sigma_{x_t} \in \mathbb{R}^{6\times6}$, which is computationally expensive to obtain from the external source, is set to a heuristically chosen value, the diagonal covariance on the bounding box measurements $\Sigma_{b_k} \in \mathbb{R}^{4\times4}$ is also set to an experimentally chosen noise value, the variance on the texture plane $\Sigma_{t_d} \in \mathbb{R}$ is the empirical variance in the depth of the triangulated points, and the covariance on the prior $\Sigma_{p_j} \in \mathbb{R}^{6\times6}$ is specified as described in section III-D.

We periodically linearize the problem in Eq. 11, and optimize in real-time for the cameras and the objects using Levenberg-Marquardt [32] algorithm. In the next section, we present experimental results on simulated and real flight sequences using an online optimization scheme, which can be more susceptible to poor solutions compared to offline batch methods, to demonstrate the advantages of ROSHAN.



(a) Baseline (Orbit Path) (b) ROSHAN (Orbit Path) (c) Baseline (Forward Path) (d) ROSHAN (Forward Path)

Fig. 3: Final estimates (red) of ellipsoids and cameras inferred using the baseline (bounding boxes only) and ROSHAN in a randomly generated map of ellipsoids. Shown in (a) and (b), estimating the parameters of the ellipsoids using diverse viewpoints of an orbiting vehicle path resulted in a small error in shape and position for both methods. However, when using measurements from a forward-moving vehicle path, where only limiting views were available, ROSHAN outperformed the baseline by a larger margin showing the strength of our approach under undesirable but common vehicle motions.

TABLE I: Median error in estimated ellipsoids for ROSHAN and the baseline in 50 randomly simulated maps of ellipsoids.

| | Orbit Path | | | Forward Path | | |
|---|---|---|---|---|---|---|
| | shape | pos. | orient. | shape | pos. | orient. |
| Baseline | 0.26 | 0.11 | 26.81 | 1.16 | 1.66 | 43.65 |
| ROSHAN | **0.17** | **0.10** | **17.97** | **0.20** | **0.47** | **30.93** |

## IV. Experimental Results in Simulation

We tested ROSHAN in an OpenGL simulation, where all objects are exactly ellipsoids, so that the ground-truth parameters of the ellipsoids can be used to evaluate the estimation accuracy of ROSHAN and a baseline in terms of shape, position, and orientation. We considered the baseline to only use the bounding box measurements as done in [18], but kept our online optimization framework with improvements on shape regularization and the bounding box measurement model to obtain a baseline meaningful for comparison.

We compared ROSHAN against the baseline in 50 randomly generated maps in two sequences with diverse (Orbit) and non-diverse (Forward) paths with Gaussian noises added to the bounding boxes and initial estimates for poses and ellipsoids. Summarized in Table I, we observed that all systems performed similarly well when given diverse viewpoints (Orbit). However, as illustrated in Fig. 3, when given degenerate viewpoints typical of forward-moving vehicle motions (Forward), ROSHAN outperformed the baseline with a 83% reduction in the error in the shape estimates (meters) based on the median error across average error per map, and 72% error reduction in the position estimates (meters); there was a smaller improvement of 29% on the orientation estimates (degrees) but neither method performed particularly well.

To further analyze the effect of degenerate viewpoints on the systems, we randomly sampled 20 ellipsoids, and for each ellipsoid, estimated its parameters using randomly sampled views from a Gaussian clipped to fixed ranges of viewpoints (yaw) around the ellipsoid. Shown in Fig. 4, for the baseline method, more views from a greater viewpoint range was required to reduce the error in both the shape and the position. However, for ROSHAN, the error in the shape estimate was small even with a single view due to the usage of shape information, and the error in position was also relatively small even with less views and viewpoint ranges, indicating a more robust system under challenging motions.

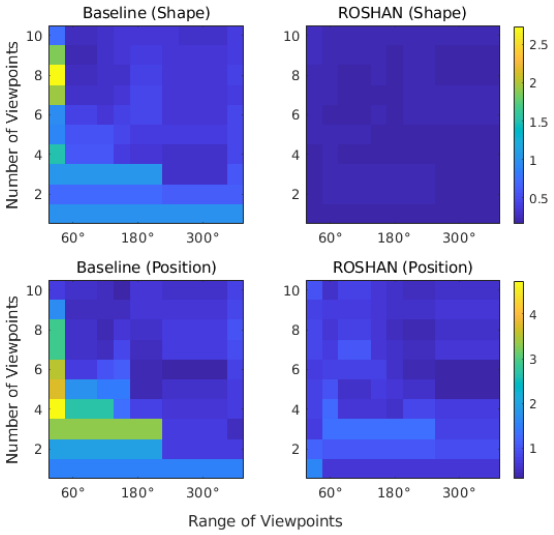



Fig. 4: Median shape error (top row) and position error (bottom row) for baseline (left column) and ROSHAN (right column) computed using different number of viewpoints (y-axis) randomly sampled from varying allowed ranges of yaw (x-axis). For the baseline method, more views from a greater viewpoint range was required to reduce the error (meters) in both shape and position. However, for ROSHAN, both errors were small even with less views from limiting viewpoints due to combining multiple sources of information.

## V. Experimental Results on Flight Sequences

To demonstrate the advantages of ROSHAN, we evaluated the performance of the algorithm both in simulation and on real-world data collected in an urban environment using a flight stack developed by the MIT/Draper team for the DARPA Fast Lightweight Autonomy (FLA) program. The photo-realistic simulation environment is a mock city rendered via the Unity Game Engine; for the simulation experiments we used ground-truth poses and added Gaussian noise to the bounding boxes at run-time. On the real flight data, the pose estimates were provided by an external SAMWISE VIO algorithm [25], which consumed monocular images and measurements from an IMU. For object detection on the real flight, we used the Mobilenet-SSD network [11] running at roughly 8 Hz. In both flight segments, the vehicles observed three cars, and did not explicitly orbit the cars.

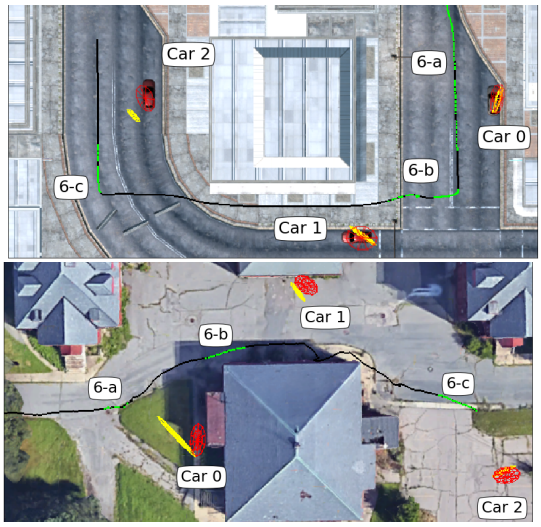


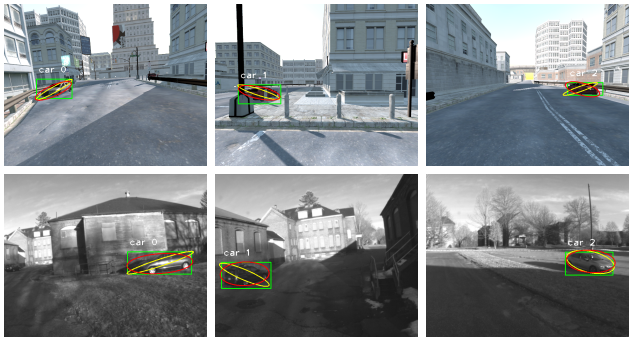

Fig. 5: Ellipsoids estimated by ROSHAN (red) and the baseline (yellow) drawn as orthographic projections along with the raw trajectory (black) and ROSHAN estimated poses with valid object detections (green). In the photo-realistic simulation (top), ROSHAN had a lower average position error of 0.84m, compared to the 1.54m of the baseline. In the real-world experiment (bottom), the origin of the projected estimates were hand-aligned in 2D to a metrically scaled overhead GPS image for qualitative analysis.

### A. Implementation details

Each valid bounding box detection was associated to an existing ellipsoid, or triggered a new landmark creation. Given a new bounding box detection, we filtered out ellipsoids using the distance between the measured centroid and predicted centroid, and the best match was chosen using a correlation score between the image hue and saturation histograms within a detection and those of previous detections; if no match was found, we initialized a new landmark. To exploit texture information, we extracted ORB features [24] from the bounding box patches, and used Lucas-Kanade [33] to track the features. While a more sophisticated sparse SLAM system [19] could be used instead, in this work we used a minimal technique, where a simple triangulation was performed between two detections of the object; the texture plane was then fit to the mean depth of the points. We observed that our assumption that all triangulated points lie on the same tangent plane can be broken here if an object is oblong and sufficiently rotated; we chose which planes to add to the graph using metrics such as the variance of the triangulated points and the length of the baseline. As in the OpenGL simulation experiments, the baseline had improvements in ROSHAN but did not incorporate the texture plane or the semantic shape prior. As the system was run online, measurements were sometimes stochastically dropped; we present here representative results from both methods.

### B. Results on Simulated and Real Flight

For visualization purposes, in each experiment the objects were aligned to an overhead image in Fig. 5, demonstrating the usefulness of our representation for autonomous surveillance missions. In Fig. 6, the conic projections of both ROSHAN and the baseline are plotted onto images from similar points in the two trajectories, providing qualitative evidence that the ROSHAN estimates better fit the cars, and indicating higher accuracy. Without semantic shape information, the baseline often optimized to low-volume ellipsoids that still satisfied the bounding box constraints. By adding the semantic shape information, we were able to avoid solutions of unreasonable volumes.



(a) (b) (c)

Fig. 6: Comparison of the projected ellipsoid estimates of ROSHAN (red) and the baseline (yellow) implementations onto the images at similar points in the raw test trajectory for the photo-realistic simulation (top) and the real-world flight (bottom). The noisy bounding boxes (green) correspond to the baseline run. Projected conics estimated with ROSHAN better approximated the outline of the cars.

## VI. Conclusions

We have presented ROSHAN – an ellipsoid-landmark based object-level SLAM system which improves estimation quality in the case of vehicle trajectories that are characterized by forward-motion, rather than orbiting. These improvements are achieved by the introduction of a texture plane factor, which constrains the depth of the landmark by exploiting texture information, and a prior on object shape that enables fast object initialization, useful for high-speed vehicle motions. We have shown in an OpenGL simulation featuring forward-motion that using these extra sources of information reduced the median errors on shape and position by 83% and 72% respectively, compared to the baseline. Similar improvements were also observed in a photo-realistic Unity simulation environment, and qualitative results were obtained on a real-world dataset, where ROSHAN estimated the shape and the position of cars reasonably well.

## Acknowledgment

This work was supported by NASA under Award No. NNX15AQ50A and DARPA under Fast Lightweight Autonomy (FLA) program, Contract No. HR0011-15-C-0110. We thank Jake Ware, John Carter, W. Nicholas Greene, Draper, and the rest of the FLA team for supporting the autonomous flight experiment and providing constructive feedback.

## References


[1] K. Ok, D. Gamage, T. Drummond, F. Dellaert, and N. Roy, "Monocular image space tracking on a computationally limited MAV," in *Proc. ICRA*, IEEE, 2015.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. ICRA*, IEEE, 2014.

[3] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. ECCV*, Springer, 2014.

[4] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *TPAMI*, vol. 40, pp. 611–625, 2018.

[5] W. N. Greene, K. Ok, P. Lommel, and N. Roy, "Multi-Level Mapping: Real-time dense monocular SLAM," in *Proc. ICRA*, IEEE, 2016.

[6] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. ICCV*, IEEE, 2011.

[7] W Nicholas Greene and N. Roy, "FLaME: Fast lightweight mesh estimation using variational smoothing on delaunay graphs," in *Proc. ICCV*, IEEE, 2017.

[8] M. Rünz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects," in *Proc. ICRA*, IEEE, 2017.

[9] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *Proc. ISMAR*, IEEE, 2018.

[10] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Proc. ECCV*, Springer, 2016.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[12] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. CVPR*, IEEE, 2017.

[13] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *Proc. ICRA*, IEEE, 2017.

[14] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *Proc. CVPR*, IEEE, 2013.

[15] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. Montiel, "Towards semantic SLAM using a monocular camera," in *Proc. IROS*, IEEE, 2011.

[16] S. Y. Bao and S. Savarese, "Semantic structure from motion," in *Proc. CVPR*, IEEE, 2011.

[17] C. Rubino, M. Crocco, and A. Del Bue, "3D object localisation from multi-view image detections," *TPAMI*, vol. 40, no. 6, pp. 1281–1294, 2018.

[18] L. Nicholson, M. Milford, and N. Sünderhauf, "QuadricSLAM: Constrained dual quadrics from object detections as landmarks in semantic SLAM," *arXiv preprint arXiv:1804.04011*, 2018.

[19] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *T-RO*, vol. 31, no. 5, pp. 1147–1163, 2015.

[20] D. A. Forsyth and J. Ponce, "A modern approach," *Computer vision: a modern approach*, 2003.

[21] M. Hosseinzadeh, Y. Latif, T. Pham, N. Suenderhauf, and I. Reid, "Towards semantic slam: Points, planes and objects," *arXiv preprint arXiv:1804.09111*, 2018.

[22] G. Cross and A. Zisserman, "Quadric reconstruction from dual-space geometry," in *Proc. ICCV*, IEEE, 1998.

[23] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *IJRR*, vol. 25, no. 12, 2006.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. ICCV*, IEEE, 2011.

[25] T. J. Steiner, R. D. Truax, and K. Frey, "A vision-aided inertial navigation system for agile high-speed flight in unmapped environments," in *Proc. Aerospace Conference*, IEEE, 2017.

[26] K Prazdny, "Egomotion and relative depth map from optical flow," *Biological Cybernetics*, 1980.

[27] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. CVPR*, IEEE, 2017.

[28] D Den Hertog, C. Roos, and T. Terlaky, "Inverse barrier methods for linear programming," *RAIRO-Operations Research*, vol. 28, no. 2, 1994.

[29] P. Gay, C. Rubino, V. Bansal, and A. Del Bue, "Probabilistic structure from motion with objects (PSfMO)," in *Proc. ICCV*, IEEE, 2017.

[30] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," *T-RO*, vol. 24, no. 5, 2008.

[31] A. Saxena, J. Driemeyer, and A. Y. Ng, "Learning 3-D object orientation from images," in *ICRA*, IEEE, 2009.

[32] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis*, Springer, 1978, pp. 105–116.

[33] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, 1981, pp. 674–679.