Hierarchical Object Map Estimation for Efficient and Robust Navigation

Kyel Ok, Katherine Liu, and Nicholas Roy

***Abstract*— We propose a hierarchical representation of objects, where the representation of each object is allowed to change based on the quality of accumulated measurements. We initially estimate each object as a 2D bounding box or a 3D point, encoding only the geometric properties that can be well-constrained using limited viewpoints. With additional measurements, we allow each object to become a higher dimensional 3D volumetric model for improved reconstruction accuracy and collision-testing. Our Hierarchical Object Map Estimation (HOME) is robust to deficiencies in viewpoints and allows planning safe and efficient trajectories around object obstacles using a monocular camera. We demonstrate the advantages of our approach on a real-world TUM dataset and during visual-inertial navigation of a quad-rotor in simulation.**

I. Introduction

We are interested in building compact maps of objects that can support visual-inertial navigation for vehicles with size, weight, and power (SWaP) constraints. Using the computation and the sensors available onboard the vehicles, we aim to fuse partial observations of the world, accumulated over time and distance, into a map that is suitable for efficient volume estimation and collision avoidance.

One way to construct a map that enables efficient collision avoidance is to represent each object as a single 3D bounding volume [1]–[4] over the entire object. Often represented as a 3D cuboid [1], [2] or an ellipsoid [3], [4], a bounding volume can be optimized [4] in real-time using 2D bounding box detections [5] from multiple viewpoints. Consisting of a single part, a 3D bounding volume can be efficiently collision-tested [6] in comparison to multiple-parts-based representations [7]–[13], such as point-clouds or voxels, that must involve testing individual parts of an object.

However, reconstructing single-part models of objects can be difficult on autonomous vehicles that do not explicitly orbit objects or maximize information gain [14] to resolve ambiguities in regions obstructed from vehicles' viewpoints. Properties of an entire object, such as the orientation and the size of an object, can be poorly estimated using partial observations from limited viewpoints and in turn lead to inaccurate collision-testing and unsafe planning behavior. Assuming symmetry in objects [15], leveraging line segments [1], [16], and fusing in texture information [4] can improve object estimation, but strong assumptions made in these approaches limit the applicable set of objects. On the other hand, foregoing the estimation of some properties and constructing more abstract object representations such as a single point-mass [17] can lead to overly conservative and inefficient navigation based on less information.

[1]Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA {kyelok,katliu,nickroy}@csail.mit.edu. This research was supported by NASA under Award No. NNX15AQ50A and by the Army Research Laboratory under Cooperative Agreement No. W911NF-17-2-0181. Their support is gratefully acknowledged.

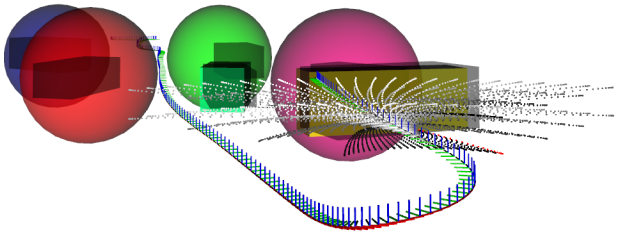


Fig. 1: Dynamically feasible vehicle trajectories scored based on the distance to objects in our hierarchical map (higher costs shown darker). We enable robust monocular mapping and collision avoidance by allowing each object model to change as a function of measurements, e.g., tight cuboids on well-observed objects and inflated spheres around others.

The key insight in this work is that the appropriate object representation to consider for collision-testing varies over the course of navigation, and should be informed by how well the properties encoded in each model can be estimated. We therefore propose building a hierarchical representation of object models, where each model encodes a different set of properties, and sequentially build a less abstract model in the hierarchy based on an empirical degeneracy metric computed from available measurement viewpoints. Specifically, we represent each object as a 2D bounding box, a point in 3D, or an upright bounding ellipsoid, in order of increasing dimensions where each model encodes the bearing, the position, or the volume in 3D space. We sequentially solve for higher fidelity models using estimates from lower fidelity models to provide good initialization.

Our Hierarchical Object Map Estimation (HOME), which allows each object model to become more sophisticated over time, improves efficiency and robustness of visual navigation. By estimating only the properties that can be constrained using available viewpoints, ill-conditioned inference problems are avoided and conservative estimates based on priors on object class [18] are substituted in place. As a result, we plan collision-free trajectories independent of available viewpoints as illustrated in Fig. 1, and improve the ability to plan efficient trajectories as more measurements are collected from new viewpoints. On the TUM [19] dataset we evaluate the reconstruction quality of HOME, and in a Unity simulation filled with object obstacles, we show that our system can use monocular images to enable efficient and robust collision

avoidance during visual-inertial navigation of a quad-rotor. In the rest of the paper, we describe online optimization methods for the object models in the hierarchy, model switching criteria for changing the degree of abstraction, a data association scheme that combines appearance-based cues with geometric information available in each model, and an efficient collision-testing method for utilizing our map in a reactive vehicle trajectory planner.

II. Overview

For the purpose of collision avoidance, we are interested in estimating geometric properties of all objects $\boldsymbol{\mathcal{O}} = \{\boldsymbol{O}_n\}_{n=0}^{N}$ using camera images $\boldsymbol{I}_t$, where we represent camera images as scalar functions defined over the pixel domain $\Omega \in \mathbb{N}^2$, such that $\boldsymbol{I}_t : \Omega \to \mathbb{N}$. We would like to find the maximum likelihood estimate of all objects $\hat{\boldsymbol{\mathcal{O}}}$ conditioned on all 2D bounding box detections $\boldsymbol{\mathcal{B}} = \{\boldsymbol{B}_b \in \mathbb{N}^4\}_{b=0}^{B}$, and camera poses $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_t \in \mathrm{SE}(3)\}_{t=0}^{T}$, given discrete object class [18] labels $\boldsymbol{\mathcal{C}} = \{c_b \in \mathbb{N}\}_{b=0}^{B}$. Assuming camera poses are separately estimated and provided in the form of pseudo-measurements, every object becomes independent of one another and the objective function can be written as

$$\hat{\boldsymbol{O}}_n = \arg\max_{\boldsymbol{O}_n} P(\boldsymbol{O}_n | \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{C}}), \tag{1}$$

where the data association of each measurement to an object is solved in a pre-process described in Section III-C.

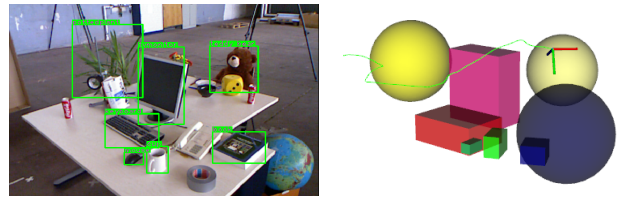
Traditional approaches to formulating Equation 1 choose a single representation for all objects in $\boldsymbol{\mathcal{O}}$. Popular abstractions include 2D bounding boxes $\boldsymbol{B} \in \mathbb{N}^4$, 3D points $\boldsymbol{L} \in \mathbb{R}^3$ and dual-ellipsoids $\boldsymbol{E}^* \in \mathbb{E}^{4\times4}$, where the dual-form $\boldsymbol{E}^* = \mathrm{adjoint}(\boldsymbol{E})$ of an ellipsoid $\boldsymbol{E}$ belongs to a subset of 4 by 4 symmetric matrices $\mathbb{E}^{4\times4}$ defined by

$$\boldsymbol{E}^* = \begin{bmatrix} \boldsymbol{R}\boldsymbol{D}\boldsymbol{R}^T - \boldsymbol{t}\boldsymbol{t}^T & -\boldsymbol{t} \\ -\boldsymbol{t}^T & -1 \end{bmatrix}. \tag{2}$$

A dual-ellipsoid is parametrized by an orientation $\boldsymbol{R} \in \mathrm{SO}(3)$, a position $\boldsymbol{t} \in \mathbb{R}^3$, and a diagonal size matrix $\boldsymbol{D} \in \mathbb{R}^{3\times3}$ formed with a size vector $\boldsymbol{d} \in \mathbb{R}^3$, and provides a minimal representation of 3D bounding volume that can be conveniently collision-tested. A 2D bounding box [20] and a 3D point [17], [21] are popular lightweight representations for tracking an object, and we discuss conservative collision avoidance for these representations in Section IV.

To enable efficient and robust navigation, we allow the level of abstraction of each individual object $\boldsymbol{O}_n$ to vary over the course of navigation as shown in Fig. 2. We introduce three indicator variables per object: $\phi_n^b, \phi_n^l, \phi_n^e \in \{0, 1\}$, which together obey the constraint $\phi_n^b + \phi_n^l + \phi_n^e = 1, \forall n \in [0, N]$. Depending on the value of the indicator variable, each object is represented as a 2D bounding box ($\phi_n^b = 1$), where no optimization over the parameters of the 2D box is performed, or represented and optimized as a point in 3D ($\phi_n^l = 1$), or as a dual-ellipsoid ($\phi_n^e = 1$). This formulation allows us to expand the posterior in Equation 1 as

$$P(\boldsymbol{O}_n | \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{C}}) \propto \phi_n^l P(\boldsymbol{L}_n | \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{C}}) \phi_n^e P(\boldsymbol{E}_n^* | \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{C}}). \tag{3}$$



(a) Object Detection (b) Our Representation

Fig. 2: Our hierarchical object representation allows each object to change its representation based on the available viewpoints. In this figure, the keyboard, monitor, mouse, and cup that were detected using YOLO [5] in (a) and observed from multiple viewpoints are constrained as dual-ellipsoids and visualized as 3D bounding cuboids in (b) while previously occluded objects, such as the book, plant, and teddy bear, are estimated as inflated point objects encoding only the position of the objects, i.e., the spheres in (b).

The summation constraint over each set of indicator variables per object ensures that no object is estimated as multiple representations in Equation 3. Crucially, we do not assume the values of the indicator values to be static, but instead objects move up the hierarchy of abstractions based on the quality of measurements available for estimation.

III. Hierarchical Object Map Estimation

Selecting the appropriate level of abstraction for each object enables HOME to estimate only the properties that can be well-constrained by a given set of viewpoints. In this section, we describe online optimization methods for the object models in the hierarchy, the model switching criteria for changing the degree of abstraction, a data association scheme that combines appearance-based cues with geometric information available in each model, and an efficient collision-testing method for the representations in the hierarchy.

A. Online Optimization for Object Models

Given that every object is independent of one another, we can solve a separate inference problem for each object based on the indicator variable. For example, given $\phi_n^e = 1$, the objective function in Eq. 1 for object $\boldsymbol{O}_n$ can be written as

$$\hat{\boldsymbol{E}}_n^* = \arg\max_{\boldsymbol{E}_n^*} P(\boldsymbol{E}_n^* | \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}; \boldsymbol{\mathcal{C}}). \tag{4}$$

1) 2D Bounding Boxes: Prior to accumulating sufficient baseline to infer the 3D geometry of an object, we directly represent each object as a 2D bounding box on the image plane, encoding only the heading of the object. We do not employ a probabilistic inference process such as filtering over bounding box measurements [20] but instead directly update the parameters of the object with the last associated bounding box measurement. However, we keep the full history of measurements for use after changing the representation to a less abstract model that can leverage the measurements to infer additional geometric properties. We note that learning-based methods [22], [23] can also track bounding boxes on 2D image using a pre-trained deep neural network, but we avoid leveraging the GPU noting that the

object detector [5] alone cannot run in frame-rate on the compute available on an MAV.

2) Points: Next, we consider point representations, which are estimated using bounding boxes and poses. We formulate the conditional probability to be maximized as

$$P(\boldsymbol{L}_n|\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{C}}) \propto \prod_{j=0}^{J} P(\boldsymbol{B}_j|\boldsymbol{L}_n,\boldsymbol{x}_{t_j}) \tag{5}$$

where all bounding boxes $\boldsymbol{B}_j, \forall j \in [0,J]$ have been associated to $\boldsymbol{L}_n$, and $\boldsymbol{x}_{t_j}$ is the pose of the camera when $\boldsymbol{B}_j$ was observed. To solve for the optimal model $\boldsymbol{L}_n^*$ of a point representation, we further abstract each bounding box measurement $\boldsymbol{B}_k$ into a ray measurement $\boldsymbol{p}_k \in \mathbb{N}^2$ through the center of the bounding box. The ray measurement model

$$h_{ray}(\boldsymbol{L}_n,\boldsymbol{x}_{t_j};\boldsymbol{K}) = \boldsymbol{K}(\boldsymbol{x}_{t_j})^{-1}\boldsymbol{L}_n \tag{6}$$

is used to project the point estimate $\boldsymbol{L}_n$ using the intrinsic camera matrix $\boldsymbol{K} \in \mathbb{R}^{3\times 3}$, where dehomogenization is assumed, and we minimize the ray measurement error by solving the nonlinear least-squares problem induced by Equation 5 assuming Gaussian distributions, i.e.,

$$\begin{aligned}\boldsymbol{L}_n^* &= \arg\max_{\boldsymbol{L}_n} P(\boldsymbol{L}_n|\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{C}}) \\ &= \arg\min_{\boldsymbol{L}_n} \sum_{j=0}^{J} \|h_{ray}(\boldsymbol{L}_n,\boldsymbol{x}_{t_j};\boldsymbol{K}) - \boldsymbol{p}_j\|^2_{\Sigma_r}.\end{aligned} \tag{7}$$

As is standard, we re-linearize and solve Eq. 7 using Levenberg-Marquardt with every new measurement.

3) Dual-Ellipsoids: For the dual-ellipsoid representation, we maximize the conditional probability

$$P(\boldsymbol{E}_n^*|\boldsymbol{\mathcal{B}},\boldsymbol{\mathcal{X}};\boldsymbol{\mathcal{C}}) \propto P(\boldsymbol{E}_n^*;c_n,\tilde{\boldsymbol{L}}_n)\prod_{k=0}^{K} P(\boldsymbol{B}_k|\boldsymbol{E}_n^*,\boldsymbol{x}_{t_k}), \tag{8}$$

where all bounding boxes $\boldsymbol{B}_k, \forall k \in [0,K]$ have been associated to $\boldsymbol{E}_n^*$, and $\boldsymbol{x}_{t_k}$ is the pose of the camera when $\boldsymbol{B}_k$ was observed. $P(\boldsymbol{E}_n^*;c_n,\tilde{\boldsymbol{L}}_n)$ is a prior on shape [4] based on the object class and on position based on the best estimate of the point model $\tilde{\boldsymbol{L}}_n$ before the object was promoted to a dual-ellipsoid. Utilizing the estimate of the point model here improves the inference for the dual-ellipsoid. We solve for optimal parameters $\hat{\boldsymbol{E}_n^*}$ by solving the nonlinear least-squares

$$\begin{aligned}\hat{\boldsymbol{E}_n^*} = \arg\min_{\boldsymbol{E}_n^*} \Big\{ &\sum_{k=0}^{K} \|h_{bb}(\boldsymbol{E}_n^*,\boldsymbol{x}_{t_k};\boldsymbol{K}) - \boldsymbol{B}_k\|^2_{\Sigma_b} + \\ &\|f_{sp}(c_n) - f_{size}(\boldsymbol{E}_n^*)\|^2_{\Sigma_s} + \|\tilde{\boldsymbol{L}}_n - f_{pos}(\boldsymbol{E}_n^*)\|^2_{\Sigma_p}\Big\},\end{aligned} \tag{9}$$

where $f_{sp} : \mathbb{N} \to \mathbb{R}^3$ is a function that maps each object class to an approximate size prior $\boldsymbol{d}_n$, $f_{size} : \mathbb{E} \to \mathbb{R}^3$ and $f_{pos} : \mathbb{E} \to \mathbb{R}^3$ are projection functions that extract size or position parameters from a dual-ellipsoid. h_{bb} is the bounding box measurement model for dual-ellipsoids. As done in [4], leveraging the dual-ellipse $\boldsymbol{C}^*$, which is the projection of a dual-ellipsoid $\boldsymbol{E}_n^*$ onto the camera plane, i.e.,

$$\boldsymbol{C}_n^* = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{E}_n^*[\boldsymbol{R}|\boldsymbol{t}]^T\boldsymbol{K}^T, \tag{10}$$

and the property that the dual-ellipse and all homogeneous tangent lines $\boldsymbol{l}_h \in \mathbb{R}^3$ to its surface must obey

$$\boldsymbol{l}_h^T\boldsymbol{C}_n^*\boldsymbol{l}_h = 0, \tag{11}$$

we compute the bounding box measurement

$$\boldsymbol{B}_k = h_{bb}(\boldsymbol{E}_n^*,\boldsymbol{x}_{t_k};\boldsymbol{K}) = [\hat{u}_{min},\hat{u}_{max},\hat{v}_{min},\hat{v}_{max}]_k^T, \tag{12}$$

$$\begin{aligned}\hat{u}_{min},\hat{u}_{max} &= \frac{1}{\boldsymbol{C}_{3,3}^*}[\boldsymbol{C}_{1,3}^* \pm \sqrt{\boldsymbol{C}_{1,3}^{*\,2} - \boldsymbol{C}_{1,1}^*\boldsymbol{C}_{3,3}^*}], \\ \hat{v}_{min},\hat{v}_{max} &= \frac{1}{\boldsymbol{C}_{3,3}^*}[\boldsymbol{C}_{2,3}^* \pm \sqrt{\boldsymbol{C}_{2,3}^{*\,2} - \boldsymbol{C}_{2,2}^*\boldsymbol{C}_{3,3}^*}].\end{aligned} \tag{13}$$

Using the bounding box measurement model, size prior from the object class, and a position prior from the best estimate of the point model, we compute the measurement error and apply the Mahalanobis norm to scale the error inversely proportionally to the square root of the covariance terms Σ_b, Σ_s, and Σ_p. When a new measurement is associated to an object, we re-linearize Equation 9 and solve for the optimal values $\hat{\boldsymbol{E}_n^*}$ using Levenberg-Marquardt [24] algorithm.

B. Switching Object Models

As described in [25], we can linearize Equations 9 and 7 into the form

$$\hat{\boldsymbol{o}}_n = \arg\min_{\boldsymbol{o}_n} \|\boldsymbol{A}\boldsymbol{o}_n - \boldsymbol{b}\|^2_{\Sigma}, \tag{14}$$

where $\boldsymbol{A}$ is the measurement Jacobian matrix, $\boldsymbol{o}_n$ is the vectorized form of object parameters, and $\boldsymbol{b}$ is the vector of measurements. The condition number [26] of the linear system indicates ill-conditioning if large, and can be determined by the ratio of the minimum and the maximum eigenvalues of the squared information matrix $\boldsymbol{A}^T\boldsymbol{A}$. This number can approximate the quality of the structure of measurement viewpoints as encoded in the Jacobian matrix $\boldsymbol{A}$. We set a minimum threshold on the number, allowing an object to switch its representation to the next less abstract model only if the linearized optimization problem for the next model is well-conditioned. In Section V, we show how to experimentally choose this threshold value and compare against another degeneracy metric introduced by Zhang et al. [27]. Given the monotonically growing number of viewpoints, we do not allow an object to switch back to a more abstract representation; however, switching back may have benefits in temporarily reducing computation or adding robustness against noisy object detections.

Due to the potentially high cost of computing the condition number, we add preconditions to computing it. For the point model, we require that there is a minimum translation in the vehicle trajectory to ensure that there is sufficient baseline to triangulate the position of the object. For the dual-ellipsoid model, we maintain egocentric spherical bins[1] of quantized viewpoints and check that a minimum number of bins are filled before computing the condition number. Here, the requirement on the number of bins serves as a heuristic for checking the diversity in viewpoints.

[1]The spherical bins are also used to sparsify measurements based on viewpoints to improve the computational efficiency of model optimization.

C. Object-level Data Association

Prior to optimization, we associate each bounding box $\boldsymbol{B}_b$ with an object $\boldsymbol{O}_n$, i.e., we solve for the association indices for 2D box, point, and dual-ellipsoid models first, before optimizing the models themselves. While it is possible to jointly optimize the models and the associations [17], for computational efficiency, we separately solve for the associations and assume them correct during model optimization.

Inspired by Wojke et al. [20], we combine costs based on the appearance, object geometry, and semantic class in a hand-tuned weighted sum to compute the matching cost between all new detections and the set of known objects. Unlike existing object-level data association methods that require an additional dense point-cloud representation [1], [16] for each object to encode object geometry, learned image-based descriptors [23] or tracking methods [22] that poorly deal with occlusions, we leverage estimated geometric properties available at each hierarchical level and appearance information we can extract from RGB images.

Specifically, we combine a RGB color histogram descriptor [4], an image-based gradient-based ORB [28] descriptor of the 2D object detection, a cost for mismatch between object semantic classes, and a geometric distance between the projection of a model and the detection. The geometric distance varies for each object representation, where for the dual-ellipsoid model, we are able to project the entire 3D volume into a 2D box to penalize object detections of smaller sizes. For the point model, we compute the distance between the projection of the point mass and the centroid of the bounding box on the image plane and for the bounding box model, we leverage the distance between the centroids. Leveraging the estimated position and volume in the dual-ellipsoid and point models allow tracking through occlusions, improving on the more abstract bounding box model.

Given the cost between the detections and objects, we use the Hungarian algorithm to compute an assignment between the detections in an image and the known objects at each hierarchical level. Similar to Wojke et al. [20], we compute the association in a hierarchical order, where we solve for an optimal assignment between the least abstract dual-ellipsoids and available object detections first. Matches with a cost above a hand-tuned threshold are considered incorrect and detections with no matching objects are attempted again in the next most abstract point level, then the bounding box level. If no match is found for an object detection at the bounding box level, we create a new object as a 2D bounding box model based on the measurement.

IV. Collision-testing Hierarchical Map

We propose a reactive motion planner based on [29], where we sample a set of dynamically feasible minimum-jerk trajectories with different heading angles and final velocities. We assign a cost to each trajectory based on the distance to the surface of estimated objects, as well as the remaining distance to a local goal. To compute the distance between a trajectory and an object, we approximate each object as an ellipsoid, and compute the distance between sampled points on the trajectory to the surface of the ellipsoid similar to [6]. While several distance metrics between an ellipsoid and a point exist [30], we choose the algebraic distance

$$e_{algebraic} = \boldsymbol{l}_h^T \boldsymbol{E} \boldsymbol{l}_h, \tag{15}$$

where $e_{algebraic} \in \mathbb{R}$ is based on the error in the algebraic relation [30] between a homogeneous tangent point $\boldsymbol{l}_h \in \mathbb{R}^4$ and the surface of an ellipsoid. We select and follow the trajectory with the smallest cost and re-plan with every map update until we reach all sequential goals. In this section, we discuss approximating each of our models in the hierarchy as an ellipsoid, and introduce a numerically-stable block matrix inversion for converting a dual-ellipsoid into an ellipsoid.

A. Dual-Ellipsoid Costs via Block Matrix Inversion

For objects represented as a dual-ellipsoid, we can compute an exact ellipsoid by taking an adjoint of the dual-ellipsoid [15]. However, numerical matrix inversion can suffer from efficiency and numerical stability issues [31]. Here, we introduce a block matrix inversion for the dual-ellipsoid defined in Equation 2 as

$$(\boldsymbol{E}^*)^{-1} = \begin{bmatrix} (A+BC)^{-1} & (A+BC)^{-1}B \\ C(A+BC)^{-1} & -1 + C(A+BC)^{-1}B \end{bmatrix}, \tag{16}$$

where we chose the blocks as $A = \boldsymbol{R}\boldsymbol{D}\boldsymbol{R}^T - \boldsymbol{t}\boldsymbol{t}^T$, $B = -\boldsymbol{t}$, $C = -\boldsymbol{t}^T$ and $D = -1$. Leveraging the orthogonality of the rotation matrix, i.e., $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$, and cancelling out position terms, i.e., $(A+BC)^{-1} = \boldsymbol{R}\boldsymbol{D}^{-1}\boldsymbol{R}^T$, we obtain a numerically stable primal-form of the ellipsoid

$$\boldsymbol{E} = \frac{1}{|\boldsymbol{E}^*|} \begin{bmatrix} \boldsymbol{R}\boldsymbol{D}^{-1}\boldsymbol{R}^T & -(\boldsymbol{R}\boldsymbol{D}^{-1}\boldsymbol{R}^T)\boldsymbol{t} \\ -\boldsymbol{t}^T(\boldsymbol{R}\boldsymbol{D}^{-1}\boldsymbol{R}^T) & -1+\boldsymbol{t}^T(\boldsymbol{R}\boldsymbol{D}^{-1}\boldsymbol{R}^T)\boldsymbol{t} \end{bmatrix}, \tag{17}$$

where the diagonal size matrix $\boldsymbol{D}$ can be trivially inverted. After every map update, we convert all dual-ellipsoid models into ellipsoids and compute the distance cost in Equation 15.

B. 3D Points and 2D Bounding Boxes

For point models $\boldsymbol{L}$, we impose a large safety bound on each object by inflating around the position estimate using a conservative maximum size $d_{max} \in \mathbb{R}$. We do this by leveraging the prior on the object size based on the object class, which is also used to constrain dual-ellipsoids in Equation 9, and taking the longest dimension as the base inflation radius since the orientation of the object is unknown. Given a potentially large variance in the object size within the same object class, we further inflate the maximum radius and construct a conservative sphere with $\boldsymbol{R} = I_{3\times3}$, $\boldsymbol{D}_{i,i} = d_{max}$, and $\boldsymbol{t} = \boldsymbol{L}$, where $I_{3\times3}$ is an identity matrix. This sphere can be used in Equation 15, to serve as a conservative distance cost, before estimating an accurate 3D model. While we do not attempt to collision-test objects represented as a 2D bounding box, learning-based costs [32], [33] exists. Additionally, leveraging the object size based on the object class, an approximate depth of the bounding box could be inferred from the size of the 2D bounding box for a conservative strategy of avoiding moving towards the heading encoded in nearby bounding boxes.

V. Experiments

We evaluated HOME on a real-world TUM RGB-D [19] dataset for data association and object reconstruction and in a Unity simulation for object reconstruction and navigation efficiency and safety. For the TUM dataset, we detected YOLO [5] bounding boxes in RGB images and used provided ground-truth camera poses as pose measurements, and for the Unity environment, we used ground-truth bounding boxes and pose measurements from the simulation.

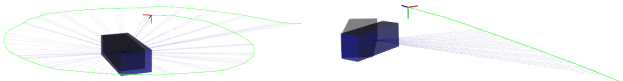
A. Degeneracy Analysis

In the Unity simulation, we first analyzed the reconstruction accuracy of a single object in relation to the condition number [26] and Zhang's inverse degeneracy [27] to empirically determine the model switching criteria described in Section III-B. We used Intersection-over-Union (IoU) and Intersection-with-Ground-Truth (IGT) as reconstruction metrics, where IoU indicated accurate reconstruction of the estimated volume, and IGT indicated accurate *coverage* of the ground-truth volume. IGT was computed as the ratio between the intersecting volume and the ground-truth volume, instead of the union in IoU, to prevent penalizing conservatively estimated volumes that are much larger than the ground-truth volume but still accurately contain it.

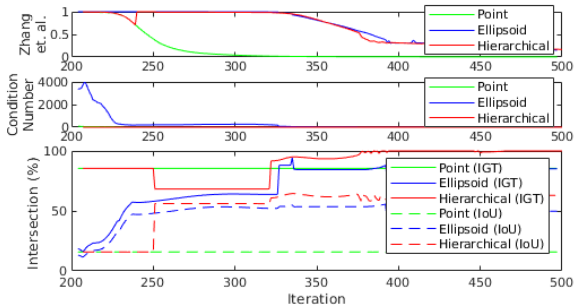
Shown in Fig. 3, during an orbital motion around a car object, we observed that both the condition number and the inverse degeneracy metric declined as more measurements from new viewpoints were collected. The point representation had both metrics decline much more quickly than the ellipsoid representation, indicating that the more abstract representation was better conditioned with less measurements. The point estimate did not significantly change over the course, where the coverage was consistently high at 85.1% IGT due to the conservative inflation, but the accuracy of reconstruction was much lower at 15.8% IoU. On the other hand, for the dual-ellipsoid, coverage started low at 18.6% IGT but increased to 85.4%, while the IoU had a similar pattern and reached 49.8%. The pattern in which the reconstruction quality improved in two large steps closely resembled the steps in the condition number, and based on the similarity, we chose the condition number over the inverse degeneracy metric for the switching criteria and empirically set our threshold at the value after the two steps.

Our hierarchical approach evaluated on the same orbital trajectory achieved high coverage throughout the orbit much similar to the point representation, while the IoU continued to improve similar to the the dual-ellipsoid. At the end of the orbit, we reached a higher reconstruction accuracy of 62.9% in IoU and 100.0% in IGT, further improving on the dual-ellipsoid representation by leveraging better initialization and position priors from the point representation.

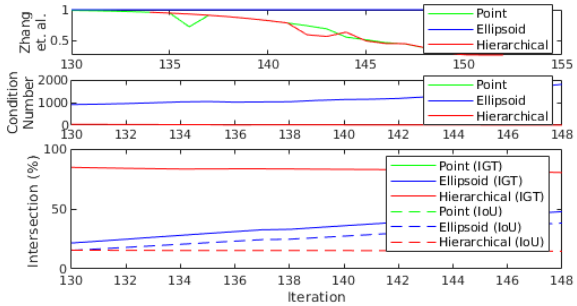
We repeated the analysis for a straight fly-by trajectory shown in Fig. 3b, and observed that the condition number for the dual-ellipsoid was significantly higher than during the orbital motion; reflecting the worse conditioning, the IGT and IoU metrics were worse at 49.1% and 39.4%, respectively. However, the more abstract point representation



(a) Orbital Motion (b) Straight Motion



(c) Degeneracy and Reconstruction Quality (Orbital Motion)



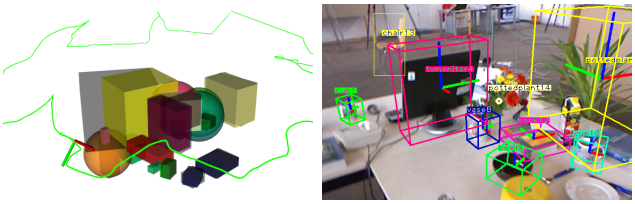
(d) Degeneracy and Reconstruction Quality (Straight Motion)

Fig. 3: Comparison of our hierarchical approach against baseline approaches of using homogeneous point or dual-ellipsoid representation. We observed that the reconstruction quality of the homogeneous ellipsoid representation, shown in (b), was poor compared to an orbital motion, shown in (a). During straight motion, HOME represented the object as a point model (thus identical reconstruction results for point and HOME) and achieved a higher IGT compared to the dual-ellipsoid, indicating that the object volume was better covered with lacking viewpoints. However, during orbital motion, HOME changed its representation to a less abstract dual-ellipsoid model, achieving a higher IoU than either methods, while consistently maintaining a high IGT indicating consistently sufficient obstacle coverage.

was less affected by lacking viewpoints, where the IGT was 79.8% and IoU 14.6%. Leveraging conservative inflation under lacking viewpoints, HOME stayed as a point model for the entire duration, performing identically to the point and achieving a consistently high coverage of the obstacle at the cost of temporarily foregoing improving accuracy.

B. Reconstruction under Orbital Motion

We tested HOME on the real-world TUM RGB-D dataset, shown in Fig. 4, and in a simulated outdoor parking lot of cars, shown in Fig. 5a, to evaluate the object reconstruction quality in comparison to the baseline approaches of homogeneous dual-ellipsoid and point models. Given that no ground-truth labels of object volume exist on the real-world TUM dataset, we manually associated YOLO [5] bounding boxes



(a) Reconstructed Map (b) Projection in Image

Fig. 4: Online reconstruction of HOME, shown in (a), on a TUM sequence and a projection of the map in a representative image (b) for qualitative analysis of the reconstruction. Pseudo-ground-truth shown in grey.

	TUM		Orbital (Sim)		Autonomous (Sim)	
	IGT (%)	IoU (%)	IGT (%)	IoU (%)	IGT (%)	IoU (%)
Point	91.5	6.1	99.9	6.1	96.1	5.1
Ellipsoid	60.4	47.6	34.5	34.5	37.8	28.9
Hierarchical	77.6	49.2	80.0	54.7	83.1	33.7

TABLE I: Reconstruction accuracy for TUM, an orbital trajectory in simulation and autonomous flight in simulation.

to observed objects and used the ground-truth associations to batch optimize pseudo-ground-truth object volumes. Summarized in Table I, HOME achieved consistently high coverage of ground-truth objects in all situations, while improving the reconstruction accuracy when more diverse viewpoints were available. These results closely resembled the reconstruction results on orbiting the single car object.

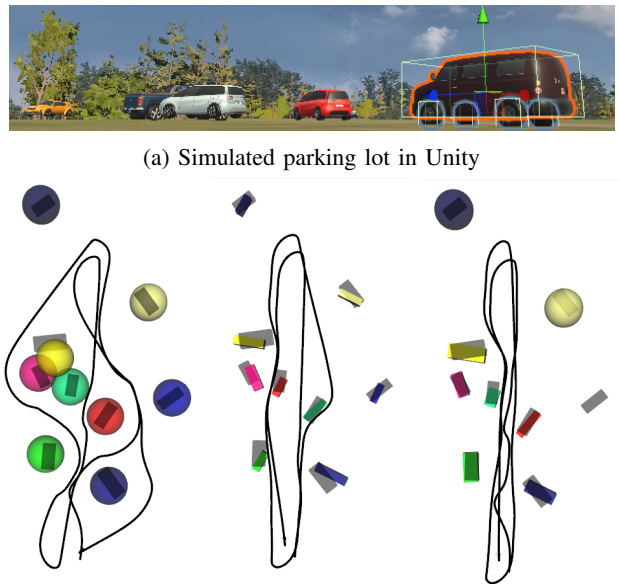
C. Visual Navigation using Hierarchical Map

To show the advantages of HOME during visual navigation of a quad-rotor around object-based obstacles, we navigated to two goal-points on each side of the simulated parking lot in Fig. 5a in two round-trips. We simulated the quad-rotor described in [29] along with RGB images of an Intel Realsense camera, and tested our reactive trajectory planner while varying the object maps used in collision-testing. Shown in Fig. 5, planning with our map resulted in taking a shorter path of $397.3m$ compared to $432.8m$ of using a homogeneous point map, as planning in the point map resulted in unnecessarily long routes around conservatively inflated volumes. Compared to the similar length $397.7m$ trajectory taken on the homogeneous ellipsoid map, HOME planned safer[2] trajectories of 0.0% near-collision compared to 3.4% of dual-ellipsoid model, due to our consistently high coverage of the object obstacles and the better reconstruction accuracy of our approach. While the point model also had 0.0% near-collision, HOME achieved higher efficiency by improving object models over the course of navigation.

VI. Related Works

Scene Graphs: 3D Scene graphs [34]–[36] are examples of constructing hierarchical map representations that can potentially improve vision-based navigation. Rosinol et al. [34] represents the map as a layered directed graph where nodes represent spatial concepts such as objects, people, and rooms, and edges represent their spatio-temporal relations. Leveraging the 3D geometry and semantics of a scene at different levels of abstraction could potentially support complex planning but unlike our approach entities do not change their representation to improve map estimation.

[2]We measured safety as the percent of vehicle poses that had a high collision cost in Eq. 15 to the ground-truth object volumes.



(a) Simulated parking lot in Unity

(b) Point (c) Ellipsoid (d) Hierarchical

Fig. 5: Trajectories of autonomous flights based on different map representations. Qualitatively, our approach (d) builds maps with more accurate object estimates with regards to IoU, while also plans trajectories that are not overly conservative as in (b) or dangerously close to objects in (c).

Learned Object Representations: State-of-the-art object mapping approaches [37]–[40] learn the shape encoding of objects to predict accurate models of objects. In particular, FroDO [40] infers learned shape codes and per-frame poses in a similar hierarchical coarse-to-fine order. While our hierarchical framework can be extended to include higher fidelity learned models, for the purpose of collision avoidance, simple geometric models are better suited.

Learned Bounding Volumes: In addition to the optimization-based [1]–[4], [15], [16] approaches discussed in Sec I, there are learning-based approaches [41]–[43] that predict 3D bounding volumes. However, similar to learned representations, these approaches are limited to the classes of objects found in games and photo-realistic simulations where ground-truth labels of 3D volumes are easier to obtain.

VII. Conclusions

We have presented HOME, a hierarchical object-based mapping system that allows each object model to become more sophisticated over time, and demonstrated the advantages of changing abstraction during visual navigation. We showed that our system plans safe trajectories unaffected by viewpoints and improve to plan efficient trajectories when we better observe the obstacles in the environment.

References

[1] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D object SLAM," *T-RO*, 2019.

[2] F. Manhardt, W. Kehl, and A. Gaidon, "Roi-10D: Monocular lifting of 2D detection to 6D pose and metric shape," in *Proc. CVPR*, 2019.

[3] 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.

[4] K. Ok, K. Liu, K. Frey, J. How, and N. Roy, "Robust object-based slam for high-speed autonomous navigation," in *Proc. ICRA*, IEEE, 2019.

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

[6] A. Dhawale, X. Yang, and N. Michael, "Reactive collision avoidance using real-time local gaussian mixture model maps," in *Proc. IROS*, IEEE, 2018.

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

[8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *T-RO*, 2015.

[9] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *IJRR*, 2016.

[10] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *Proc. IROS*, 2017.

[11] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level SLAM," in *Proc. 3DV*, 2018.

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

[13] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," *arXiv preprint arXiv:1910.02490*, 2019.

[14] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *IJRR*, vol. 34, no. 7,

[15] Z. Liao, W. Wang, X. Qi, X. Zhang, L. Xue, J. Jiao, and R. Wei, "Object-oriented SLAM using quadrics and symmetry properties for indoor environments," *arXiv preprint arXiv:2004.05303*, 2020.

[16] Y. Wu, Y. Zhang, D. Zhu, Y. Feng, S. Coleman, and D. Kerr, "EAO-SLAM: Monocular semi-dense object slam based on ensemble data association," *arXiv preprint arXiv:2004.12730*, 2020.

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

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014.

[19] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IROS*, 2012.

[20] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. ICIP*, 2017.

[21] K. Doherty, D. Fourie, and J. Leonard, "Multimodal semantic SLAM with probabilistic data association," in *Proc. ICRA*, 2019.

[22] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. ECCV*, 2016.

[23] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, T. Vojir, G. Hager, A. Lukezic, A. Eldesokey, *et al.*, "The visual object tracking vot2017 challenge results," in *Proc. ICCV Workshop*, 2017.

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

[25] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *IJRR*, 2006.

[26] E. W. Cheney and D. R. Kincaid, *Numerical mathematics and computing*. Cengage Learning, 2012.

[27] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *Proc. ICRA*, 2016.

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

[29] M. Ryll, J. Ware, J. Carter, and N. Roy, "Efficient trajectory planning for high speed flight in unknown environments," in *Proc. ICRA*, IEEE, 2019.

[30] A. Y. Uteshev and M. V. Goncharova, "Point-to-ellipse and point-to-ellipsoid distance equation analysis," *Journal of computational and applied mathematics*, 2018.

[31] E. Isaacson and B. Keller, "Analysis of numerical methods," *The Mathematical Gazette*, 1969.

[32] F. Sadeghi and S. Levine, "CAD2RL: Real single-image flight without a single real image," 2017.

[33] G. J. Stein and N. Roy, "GeneSIS-RT: Generating synthetic images for training secondary real-world tasks," in *Proc. ICRA*, 2018.

[34] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3*d* dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *arXiv preprint arXiv:2002.06289*, 2020.

[35] U.-H. Kim, J.-M. Park, T.-J. Song, and J.-H. Kim, "3-D scene graph: A sparse and semantic representation of physical environments for intelligent agents," *Transactions on Cybernetics*, 2019.

[36] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3D scene graph: A structure for unified semantics, 3d space, and camera," in *Proc. ICCV*, 2019.

[37] S. Muralikrishnan, V. G. Kim, M. Fisher, and S. Chaudhuri, "Shape unicode: A unified shape representation," in *Proc. CVPR*, 2019.

[38] Y. Liao, K. Schwarz, L. Mescheder, and A. Geiger, "Towards unsupervised learning of generative models for 3D controllable image synthesis," in *Proc. CVPR*, 2020.

[39] E. Sucar, K. Wada, and A. Davison, "Neural object descriptors for multi-view shape reconstruction," *arXiv preprint arXiv:2004.04485*, 2020.

[40] M. Runz, K. Li, M. Tang, L. Ma, C. Kong, T. Schmidt, I. Reid, L. Agapito, J. Straub, S. Lovegrove, *et al.*, "FroDO: From detections to 3D objects," in *Proc. CVPR*, 2020.

[41] F. Manhardt, W. Kehl, and A. Gaidon, "ROI-10D: Monocular lifting of 2D detection to 6D pose and metric shape," in *Proc. CVPR*, 2019.

[42] A. Simonelli, S. R. Bulò, L. Porzi, M. Lopez-Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection," in *Proc. ICCV*, 2019.

[43] C. Sahin, G. Garcia-Hernando, J. Sock, and T.-K. Kim, "A review on object pose recovery: From 3D bounding box detectors to full 6D pose estimators," *Image and Vision Computing*, 2020.