# VoluMon: Weakly-Supervised Volumetric Monocular Estimation with Ellipsoid Representations

Katherine Liu, Kyel Ok, and Nicholas Roy

***Abstract*— Deep learning approaches to estimating 3D object pose and geometry present an attractive alternative to online estimation techniques, which can suffer from significant estimation latency. However, a practical hurdle to training state-of-the-art deep 3D bounding box estimators is collecting a sufficiently large dataset of 3D bounding box labels. In this work, we present a novel framework for weakly supervised volumetric monocular estimation (VoluMon) that requires annotations in the image space only, i.e., associated object bounding box detections and instance segmentation. By approximating object geometry as ellipsoids, we can exploit the dual form of the ellipsoid to optimize with respect to bounding box annotations and the primal form of the ellipsoid to optimize with respect to a segmented pointcloud. For a simulated dataset with access to ground-truth, we show monocular object estimation performance similar to a naive online depth based estimation approach and after online refinement when depth images are available, we also approach the performance of a learned deep 6D pose estimator, which is supervised with projected 3D bounding box keypoints and assumes known model dimensions. Finally, we show promising qualitative results generated from a real-world dataset collected using a stereo pair.**

## I. Introduction

We would like to enable low-latency object-level estimation from RGB sensors. As robots venture further into the real-world, semantic scene understanding is increasingly important. Object-level estimation, i.e., inferring the pose and geometry of objects, is relevant for a diverse set of applications from autonomous navigation to manipulation. Performing both pose estimation of objects and geometric reconstruction concurrently without *a priori* geometric models or from limited measurements can suffer from perspective projection ambiguity challenges as well as ambiguities arising from object self-occlusion.

Online inference methods present flexible solutions for 3D object estimation, but require iterative optimization and may also assume that several measurements will be aggregated over time, both of which increase the latency from sensor measurement to estimate. For example, vision-based simultaneous localization and mapping (vSLAM) approaches have demonstrated success in fusing multiple object-level measurements to build object-level maps using a diverse set of representations including points [1], ellipsoids [2]–[4], and learned representations [5]. While fusing a number of measurements can benefit the estimation accuracy and aid with observability issues, the inference process introduces latency between image registration and estimation. Previous approaches to single-view estimation used pointcloud measurements to fit representations such as meshes [6] and superquadrics [7]. Such approaches may reduce the latency from measurement to estimate, but still require an online optimization process.

Deep learning approaches have gained popularity as a method to exploit offline datasets to reduce latency and enable single-shot 3D object estimation, but generally require annotations that are difficult to obtain. Fully supervised approaches to deep monocular volume estimation [8], [9] directly regress object parameters, and thus rely on datasets of images labelled with object positions, orientations, and sizes, which are difficult to obtain in practice. In contrast to 2D labelling problems such as bounding boxes or pixel-wise segmentation, which are annotated purely on the image plane, annotating the 3D object parameters from 2D image

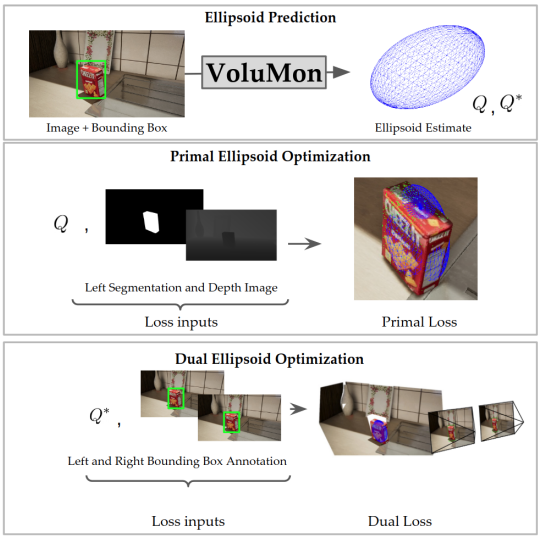



Fig. 1: (Top row) VoluMon predicts the 3D object pose and shape from a monocular image and bounding box detections (green box) by approximating objects as ellipsoids (blue mesh). We weakly supervise learning by exploiting two different forms of 3D ellipsoid representations. (Middle row) The primal form of the ellipsoid provides a differentiable algebraic metric to fit to 3D points (green points) extracted from an instance segmentation annotation and depth image. (Bottom row) The dual form of the ellipsoid provides a differentiable geometric metric for VoluMon to compare annotated bounding boxes from a stereo pair with the projected bounding box from the ellipsoid estimate. Without requiring an *a priori* model, VoluMon can also further refine the estimated ellipsoid parameters online given a pointcloud.

All authors are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology in Cambridge, USA. {katliu,kyelok,nickroy}@mit.edu. This research was supported by the Army Research Laboratory under Cooperative Agreement No. W911NF-17-2-0181, by NASA under Award No. NNX15AQ50A, and the NeuroAutonomy MURI under ONR Award No. N00014-19-1-2571. Their support is gratefully acknowledged.

data often requires interfacing with additional information beyond the image itself, such as dense 3D pointclouds [10], detailed geometric models [11], or using pre-trained 3D object detection neural networks [12]. Although modern object detectors such as [13], [14] have benefited immensely from large open-sourced datasets with 2D annotations over diverse classes of objects such as [15], such datasets are much more difficult to obtain for 3D object estimation tasks. The relative lack of diverse 3D datasets and the cost of obtaining 3D annotations pose practical limitations to extending supervised approaches to arbitrary classes. Using detailed *a priori* models, some modern approaches exploit the ability to generate annotations from photo-realistic simulation data [16] or render the object [17]. When assuming known mesh models, [18] showed that further online optimization given depth information can greatly increase accuracy. However, obtaining geometrically detailed *a priori* models remains a non-trivial task for arbitrary objects.

In this work, we introduce a novel framework for learned volumetric monocular estimation (VoluMon) capable of estimating the position, orientation, and size of objects, while requiring only image-space annotations on 2D images at training time instead of detailed geometric models or 3D labels. The key insight in this work is that by approximating object geometry with ellipsoids, we can exploit differentiable geometric and algebraic relationships between ellipsoids and 2D annotations to enable a weakly supervised learning process and significantly lower the annotation burden for deeply learned methods. VoluMon trains a deep neural network to predict the 3D size and 6D pose of objects using annotated bounding boxes, instance segmentations paired with depth images, or both. The core of our approach requires only a bounding box detection and single RGB image at inference time; obtaining an estimate of the pose and size of an object is simply a feed-forward pass through the network, rather than an iterative optimization process. However, given segmented depth information at run-time, the ellipsoid approximation allows for object pose estimates from VoluMon to be further refined without *a priori* geometric models.

We demonstrate the advantages of our approach by exploring several variants of VoluMon depending on the available sensors and annotations at training time and run time on subset of the Falling Things Dataset [19]. We show that using only a monocular sensor at inference time, VoluMon performs similarly to a naive point-cloud based online ellipsoid estimation approach while requiring less than 1% of the average computation time. Given segmented depth information to further refine pose estimates at inference time, for some metrics we approach or surpass the performance of a deeply learned 6D object pose algorithm that assumes groundtruth size information. Finally, we present promising qualitative results on several real-world datasets.

## II. Preliminaries

VoluMon approximates 3D objects with ellipsoidal volumes parameterized by the orientation $\boldsymbol{R} \in \mathrm{SO}(3)$ and position $\boldsymbol{t} \in \mathbb{R}^3$ with respect to the sensor viewing the object, as well as the size $\boldsymbol{d} \in \mathbb{R}^3$ of the major axes of the ellipsoid similar to [2]. In particular, we consider two mathematically convenient forms of the ellipsoidal representation[1]: the dual and the primal. The primal ellipsoid in $\mathbb{R}^3$ can be expressed as the set of all 3D points $\boldsymbol{x}$ in homogeneous form satisfying the implicit algebraic relationship $x\boldsymbol{Q}x^T = 0$, where

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{A} & -(\boldsymbol{A})\boldsymbol{t} \\ -\boldsymbol{t}^T(\boldsymbol{A}) & -1 + \boldsymbol{t}^T(\boldsymbol{A})\boldsymbol{t} \end{bmatrix}, \tag{1}$$

$\boldsymbol{A} = \boldsymbol{R}\boldsymbol{D}^{-1}\boldsymbol{R}^T$, and $\boldsymbol{D} = (\mathrm{diag}(\frac{\boldsymbol{d}}{2}))^2$. In contrast, the dual form of the ellipsoid, defines the infinite set of planes $\boldsymbol{\pi}$ tangent to the surface of the ellipsoid, i.e., $\boldsymbol{\pi}\boldsymbol{Q}^*\boldsymbol{\pi}^T = 0$,

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

For the remainder of this document, we refer to the primal and dual ellipsoids by their governing real symmetric 4x4 matrices, $\boldsymbol{Q}$ and $\boldsymbol{Q}^*$ respectively.

A strong advantage of approximating objects as ellipsoids, as opposed to other geometric models such as 3D bounding boxes, is that both the dual and primal forms provide differentiable relationships between the object and quantities obtained from image-space annotations paired with sensor measurements, suggesting their suitability use in deep learning frameworks that are optimized via back-progagation. The dual ellipsoid form allows for the closed-form calculation of a projected image axis-aligned 2D bounding box induced by an ellipsoid estimate, therefore providing a differentiable geometric metric with which to compare bounding box detections of an object. The primal form of the ellipsoid provides a differentiable algebraic metric to measure how well an observed surface point of an object, which can be obtained from an instance segmentation and depth image, agrees with an ellipsoid estimate. VoluMon trains a deep neural network (described in Sec. III) to predict the parameters $q_i = (\boldsymbol{d}_i, \boldsymbol{R}_i, \boldsymbol{t}_i)$ of an ellipsoid approximation using differentiable measurement functions derived from the ellipsoid representation (described in Sec. IV).

## III. Model Overview

VoluMon trains a model (shown in Fig. 2) to predict ellipsoid parameters from images $\boldsymbol{I}_t : \Omega \in \mathbb{N}^2 \rightarrow \mathbb{R}$, where $\Omega$ is the image pixel domain, and bounding boxes $\boldsymbol{B}$, which are characterized by the pixel locations of the four corners. Let $\Phi$ be a neural network parameterized by $\boldsymbol{\beta}$ that takes as input a set of images $\boldsymbol{\mathcal{I}} = \{\boldsymbol{I}_i\}_{i=0}^K$ and bounding boxes $\boldsymbol{\mathcal{B}} = \{\boldsymbol{B}_i \in \Omega^2\}_{i=0}^K$ around the K objects of interest. The network outputs free parameters $\boldsymbol{\phi}_i = [\boldsymbol{\phi}_{i,D}, \boldsymbol{\phi}_{i,R}, \boldsymbol{\phi}_{i,UV}, \boldsymbol{\phi}_{i,Z}] \in \mathbb{R}^{10}$ per object estimate, where $\boldsymbol{\phi}_{i,D} \in \mathbb{R}^3$, $\boldsymbol{\phi}_{i,R} \in \mathbb{R}^4$, $\boldsymbol{\phi}_{i,UV} \in \mathbb{R}^2$, and $\boldsymbol{\phi}_{i,Z} \in \mathbb{R}^1$ are used to reconstruct respectively the size, rotation, centroid projection, and depth of the object in the camera frame. To ensure the prediction of valid ellipsoids, we formulate an additional function $\boldsymbol{f}$ that maps the outputs of the model $\Phi$ to reasonable $\boldsymbol{q}$, yielding the relationship

$$\boldsymbol{q} = \boldsymbol{f}(\Phi(\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{B}}; \boldsymbol{\beta}), \boldsymbol{\mathcal{B}}), \tag{3}$$

[1]We derive the primal by transforming a scaled ellipsoid via relationships described in [20]. The definition of the dual can be found in [3].

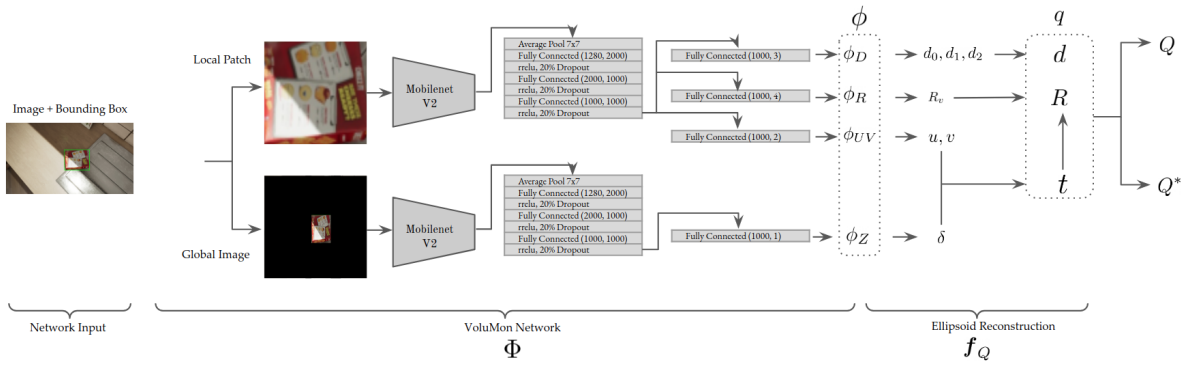



Fig. 2: VoluMon system diagram. Given an image and bounding box, VoluMon passes the resized contents of the bounding box to the local patch sub-network, and the resized full image with contents outside of the bounding box set to zero in all channels to the global image sub-network. Each image and bounding box pair yields one ellipsoid estimate. Both sub-networks utilize MobilenetV2 as a convolutional feature extractor that feeds into a series of fully connected layers. To improve generalization performance, VoluMon predicts the image coordinates of the projected object centroid, the allocentric rotation, and shape parameters from the contents in the region of interest only, while depth of the centroid of the object is predicted using features from the global image. The raw output of the network $\boldsymbol{\phi}$ are reconstructed to $q$ to constrain predictions to reasonable ellipsoids. The subscript notation $i$ is dropped for readability.

where $\boldsymbol{q} = \{q_i\}_{i=0}^{K}$, and $\boldsymbol{f}$ takes as input $\boldsymbol{\mathcal{B}}$ to constrain the projection of the centroid estimate to lie within the detected bounding box in the image frame.

Rather than predict the rotation, shape, and translation of an object from a single set of shared features, which could be difficult to generalize to arbitrary object locations, VoluMon splits the prediction of object properties between two decoupled sub-networks. The local patch sub-network predicts object properties that are independent of where in the image the bounding box is located and outputs $\boldsymbol{\phi}_{i,D}, \boldsymbol{\phi}_{i,R}, \boldsymbol{\phi}_{i,UV}$ per object, while the global image sub-network helps to estimate global pose properties and outputs $\boldsymbol{\phi}_{i,Z}$ per object. Finally, the mapping from $\boldsymbol{\phi}$ to $q$ developed in the following sections defines the function $\boldsymbol{f}$ required by Equation 3.

*1) Global Image Branch:* The global image network receives the RGB image resized to 224x224 pixels, with all channels outside the observed bounding box set to zero with some padding, and predicts the "disparity" of the centroid, $\delta_i$. The possible centroid depth is obtained by constraining the raw outputs such that $\delta_i = \alpha_\delta \text{sigmoid}(\boldsymbol{\phi}_{i,Z})$, and letting $t_{z,i} = bf/\delta_i$, where $f, b$ are set to roughly the focal length and baseline of the stereo camera, and $\alpha_\delta$ is the maximum allowed disparity. This parameterization seeks to abstract prediction of the centroid depth from camera parameters.

*2) Local Patch Branch:* The local prediction network receives the image in the bounding box, resized to $224 \times 224$ pixels. To ensure reasonable $\boldsymbol{t}$ and $\boldsymbol{d}$, the raw outputs of the network are constrained such that $[d_{i,0}, d_{i,1}, d_{i,2}] = \exp(\boldsymbol{\phi}_{i,D})$, and $[u_i, v_i] = \text{sigmoid}(\boldsymbol{\phi}_{i,UV})$, where exp and sigmoid are applied element-wise. We then formulate the per-object translation and shape estimates as

$$\begin{aligned} \boldsymbol{t}_i &= [((u_i w_i + u_{min,i}) - \bar{u}_i) t_{z,i}/f, \\ &\quad ((v_i h_i + v_{min,i}) - \bar{v}_i) t_{z,i}/f, t_{z,i}] \\ \boldsymbol{d}_i &= \alpha_y [d_{i,0}, d_{i,0} + d_{i,1}, d_{i,0} + d_{i,1} + d_{i,2}] + \epsilon_s, \end{aligned} \tag{4}$$

where $u_i, v_i$ are the projected centroid coordinates with respect to the image bounding box edges. Additionally, $w_i, h_i, u_{min,i}, v_{min,i}, \bar{u}_i, \bar{v}_i$ are the bounding box width, height, the two coordinates of the lower left corner of the bounding box, and the image center, respectively. $\alpha_y$ is a size scaling parameter and $\epsilon_s$ enforces a minimum shape. Rather than allowing for arbitrary object location, Equation 4 constrains the projected centroid to lie within the 2D bounding box. In an effort to reduce the optimization space due to the potentially ambiguous relationship between rotation and shape, Equation 4 also constrains the representation of $\boldsymbol{d}$ to learn a shape where each axis is of increasing size.

From the local patch, we also predict the rotation $\boldsymbol{R}_v$ with respect to the object, i.e., the allocentric rotation. We constrain the raw output of the network such that $[r_{i,0}, r_{i,1}, r_{i,2}, r_{i,3}] = \frac{\boldsymbol{\phi}_{i,R}}{||\boldsymbol{\phi}_{i,R}||_2}$, where the left hand vector is interpreted as $\boldsymbol{R}_v$ in quaternion form. Unlike the egocentric rotation of the object, previous works in deeply learned 3D bounding box detection [21], [22] have shown that objects with similar allocentric rotations have similar visual appearances in the local patch. To recover the egocentric rotation $\boldsymbol{R}_i$ we use a similar transform as described in [17] using $\boldsymbol{t}$ and $\boldsymbol{R}_v$.

## IV. Weakly Supervised Ellipsoid Prediction

In this work, rather than assuming a dataset of 3D size and 6D pose annotations, we rely instead on 2D image-space annotations on RGB and depth images, which we posit are easier in practice to obtain. We assume in this work images are obtained from a calibrated stereo pair and that $\boldsymbol{P}_L, \boldsymbol{P}_R \in \mathbb{R}^{3x4}$ are known and constant projection matrices from world coordinates to the image plane (i.e., includes both the intrinsics and extrinsics) for the left and right cameras. We futher assume a dataset of measurements of $K$ labelled objects by $\mathcal{G} = \{\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{B}}_L, \boldsymbol{\mathcal{B}}_R, \boldsymbol{\mathcal{S}}\}$. $\boldsymbol{\mathcal{B}}_L$ and $\boldsymbol{\mathcal{B}}_R$ are the set of bounding box observations from the left images $\boldsymbol{\mathcal{I}}$ and right image, respectively. The set of pixel-wise segmentations $\boldsymbol{\mathcal{S}} = \{\boldsymbol{S}_k\}_{k=0}^{K}$ is composed of individual segmentations $\boldsymbol{S}_k : \Omega \in \mathbb{N}^2 \rightarrow \{0, 1\}$ that denote whether a pixel in the left image is associated with the object in question.

3D points expected to lie on the surface of the object can also be extracted from $\mathcal{G}$, provided accurate depth images.

Assuming a stereo dataset, we assume a pre-processing step that calculates a depth image from a left and right image and uses $\boldsymbol{P}_L$ and $\boldsymbol{S}$ to return the set of $J_k$ 3D points $\boldsymbol{X}_k = \{\boldsymbol{x}_{k,i}\}_{j=0}^{J_k}$ corresponding to the pixels annotated to be upon the object $k$. The respective sets of points for the objects are then aggregated in $\boldsymbol{\mathcal{X}} = \{\boldsymbol{X}_k\}_{k=0}^{K}$. We observe that $\boldsymbol{\mathcal{X}}$ could also in practice come from an arbitrary depth sensor aligned with the RGB images.

To optimize the parameters of Equation 3, VoluMon leverages two loss functions given different types of image-space annotation: a loss $\mathcal{L}_D$ based on the dual form given two bounding box measurements of the object from a stereo pair (described in Sec. IV-A), and a loss $\mathcal{L}_P$ based on the primal form given observed points from the surface of the object from a segmented depth image (described in Sec. IV-B). Intuitively, VoluMon attempts to learn to predict ellipsoid parameters that are consistent with observed measurements. The overall loss function is

$$\begin{aligned}\boldsymbol{\beta}^* = &\arg\min_{\boldsymbol{\beta}} \alpha_P \mathcal{L}_P(\boldsymbol{f}(\Phi(\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{B}}_L; \boldsymbol{\beta}), \boldsymbol{\mathcal{B}}_L), \boldsymbol{\mathcal{X}}) \\ &+ \alpha_D \mathcal{L}_D(\boldsymbol{f}(\Phi(\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{B}}_L; \boldsymbol{\beta}), \boldsymbol{\mathcal{B}}_L), \boldsymbol{\mathcal{B}}_L, \boldsymbol{\mathcal{B}}_R) \\ &+ \alpha_S \mathcal{L}_S(\boldsymbol{f}(\Phi(\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{B}}_L; \boldsymbol{\beta}), \boldsymbol{\mathcal{B}}_L))\end{aligned} \tag{5}$$

where $\alpha_P, \alpha_D, \alpha_S \geq 0$ are all hand-tuned weighting terms. Equation 5 also includes a regularization term $\mathcal{L}_S$ that encourages shape predictions of an object to be similar (described in Sec. IV-C).

### A. Bounding Boxes and Dual Ellipsoid Optimization

To develop a loss function to weakly supervise ellipsoid prediction from bounding boxes, we work with the dual ellipsoid representation. 2D bounding box detections from many state-of-the-art object detection pipelines can be interpreted as measurements of axis-aligned bounding planes for objects approximated as 3D ellipsoids, where each edge of a bounding box detection projects into a plane in 3D space which constrains the ellipsoid. As in [3], [4], to solve for the expected bounding box measurements given some $\boldsymbol{Q}^*$ and camera projection matrix $\boldsymbol{P}$, we first project the dual ellipsoid to a dual-conic $\boldsymbol{C}$ on the image plane:

$$\boldsymbol{C}^* = \boldsymbol{P}\boldsymbol{Q}^*\boldsymbol{P}^T, \tag{6}$$

Solving for axis aligned bounding boxes that satisfy the implicit dual conic function in Equation 6 yields

$$\begin{aligned}B_{umin}, B_{umax} &= \frac{1}{\boldsymbol{C}^*_{3,3}}[\boldsymbol{C}^*_{1,3} \pm \sqrt{{\boldsymbol{C}^*_{1,3}}^2 - \boldsymbol{C}^*_{1,1}\boldsymbol{C}^*_{3,3}}], \\ B_{vmin}, B_{vmax} &= \frac{1}{\boldsymbol{C}^*_{3,3}}[\boldsymbol{C}^*_{2,3}] \pm \sqrt{{\boldsymbol{C}^*_{2,3}}^2 - \boldsymbol{C}^*_{2,2}\boldsymbol{C}^*_{3,3}}].\end{aligned} \tag{7}$$

Using Equations 6 and 7, we can form a closed-form, differentiable measurement model $\hat{\boldsymbol{B}} = \boldsymbol{h}(q, \boldsymbol{P})$ that maps quadric parameters to an expected 2D bounding box measurement. For any given ground-truth bounding box $\boldsymbol{B}$ and predicted bounding box $\hat{\boldsymbol{B}}$, we define the projection error $e_b(\boldsymbol{B}, \hat{\boldsymbol{B}})$ as the sum of squared differences between the bounding box centroids and dimensions.

A single bounding box measurement is insufficient to fully constrain all parameters of the ellipsoid representation. One approach to partially resolving the measurement ambiguity is to triangulate the quantity of interest from multiple views. In this work, we propose using stereo data at training time to impose projective consistency, as visualized in the bottom panel of Figure 1. Let $\boldsymbol{Q}^*$ be defined with respect to the left camera. The final loss function for using bounding boxes from a stereo pair to estimate object parameters is then

$$\begin{aligned}&\mathcal{L}_D(\boldsymbol{q}, \boldsymbol{\mathcal{B}}_L, \boldsymbol{\mathcal{B}}_R) = \\ &\frac{1}{K}\sum_{k=0}^{K}[e_b(\boldsymbol{B}_{L,k}, \boldsymbol{h}(q_k, \boldsymbol{P}_L)) + e_b(\boldsymbol{B}_{R,k}, \boldsymbol{h}(q_k, \boldsymbol{P}_R))],\end{aligned} \tag{8}$$

where we have used the known and constant stereo projection matrices $\boldsymbol{P}_L, \boldsymbol{P}_R$. Although the bounding box label in the right image is used at train time to calculate the loss for back-propagation, it is not required at inference time.

### B. 3D Points and Primal Ellipsoid Optimization

To specify a loss function for weakly supervised ellipsoid prediction from segmented depth images, we turn to the primal ellipsoid representation. An algebraic error metric on an observed surface point $\boldsymbol{x}$ can be obtained from implicit algebraic definition of a primal ellipsoid. In particular, $\boldsymbol{x}\boldsymbol{Q}\boldsymbol{x}^T$ evaluates to strictly less than zero if $\boldsymbol{x}$ is inside the ellipsoid, strictly greater than zero if $\boldsymbol{x}$ is outside the ellipsoid, and to zero if and only if the point $\boldsymbol{x}$ lies on the surface of the ellipsoid. An algebraic error metric follows directly:

$$e_s(\boldsymbol{x}, q) = e_s(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{R}, \boldsymbol{t}) = \sqrt{\boldsymbol{d}_0\boldsymbol{d}_1\boldsymbol{d}_2}(x\boldsymbol{Q}x^T)^2. \tag{9}$$

Similar to previous work [23], [24] an additional term involving the product of the axes lengths is added to mitigate the bias of fitting to larger primitive sizes.

Taking the average of Equation 9 per object over the entire dataset is then:

$$\mathcal{L}_P(\boldsymbol{q}, \boldsymbol{\mathcal{X}}) = \sum_{k=0}^{K}\sum_{j=0}^{J_k}\frac{e_s(\boldsymbol{x}_{k,j}, q_k)}{J_k K}. \tag{10}$$

The middle panel of Figure 1 visualizes an example of an ellipsoid estimate and points extracted from a pixel-wise segmentation paired with a depth image. Segmentations and depth images are required during only training, not for the feed-forward pass of the network.

### C. Intra-Class Size Consistency Loss

While stereo triangulation provides up to eight bounding edges, the quality of the triangulation can vary with the stereo baseline and size of the bounding box detection. Additionally, although $\mathcal{L}_P$ relies on depths extracted from a stereo pair, severely self-occluded views (such as seeing only the front surface of a box) can introduce significant shape ambiguity. Therefore, to impose additional structure to the optimization, we introduce an intra-class size consistency loss by penalizing the shape variance with constant offset $\epsilon_v$:

$$\begin{aligned}\mathcal{L}_S(\boldsymbol{q}) = &\operatorname{var}(\{\boldsymbol{d}_{i,0}\}_{i=0}^{k}) + \operatorname{var}(\{\boldsymbol{d}_{i,1}\}_{i=0}^{k}) \\ &+ \operatorname{var}(\{\boldsymbol{d}_{i,2}\}_{i=0}^{k}) + \epsilon_v,\end{aligned} \tag{11}$$

where the sets of $\boldsymbol{d}$ can be obtained from $q$. The size consistency loss can be useful for object classes that are expected to have similar dimensional characteristics, such as mass-produced household objects.

### D. Network Training and Post Inference Refinement

VoluMon optimizes network parameters to minimize Equation 5 via backpropagation using Adam. In practice, we do not calculate losses over all $K$ objects in the dataset, but optimize over minibatches. At runtime, we assume an off-the-shelf object detector such as [14], [25] provides an initial detection, and prediction is simply a feedfoward pass through the network.

While VoluMon implicitly learns a regression of object parameters from measurements, ellipsoid parameters may also be regressed directly. Given a segmented depth image providing a measured pointcloud $\boldsymbol{X}_k$ and bounding box $\boldsymbol{B}_k$ at run-time, the ellipsoid estimate provided by the network can also be used as an initial estimate for further online optimization. Let $\boldsymbol{\phi}_k$ again be free parameters for a given object. We apply Equation 10 to directly update $\boldsymbol{\phi}_k$ via gradient descent methods, i.e.,

$$\boldsymbol{\phi}_k^* = \arg\min_{\boldsymbol{\phi}_k} \mathcal{L}_P(\boldsymbol{f}(\boldsymbol{\phi}_k, \boldsymbol{B}_k), \boldsymbol{X}_k). \tag{12}$$

Although initial estimates for $\boldsymbol{\phi}_k$ may be obtained from $\boldsymbol{X}_k$, we will show in Sec. V that direct online regression can be both slow and inaccurate compared to a learned model. Using VoluMon's learned model to provide an initial estimate for the regression (similar in spirit to [18]) can enable faster and more accurate estimates on some metrics. In this work, we update only the pose components when using an initial estimate from VoluMon, keeping $\boldsymbol{\phi}_{k,D}$ fixed. The ellipsoid parameters can be reconstructed as $q_k^* = \boldsymbol{f}(\boldsymbol{\phi}_k^*, \boldsymbol{B}_k)$.

## V. Simulation Experiments

We evaluate several variants of VoluMon compared to two baseline methods. In addition to training on both the bounding box and segmentation annotations (where $\alpha_D = 1, \alpha_P = 1$, denoted *VoluMon Both*), we evaluate the performance of the network using bounding box annotations only (where $\alpha_D = 1, \alpha_P = 0$, denoted *VoluMon Dual Only*), and the network trained using segmented depth images only (where $\alpha_D = 0, \alpha_P = 1$, denoted *VoluMon Primal Only*). For training and evaluation, we use ground-truth bounding box and segmentation annotations as an input to our approach. We also test a variant of *VoluMon Both* denoted *VoluMon Both Noisy* where the test time input data is obtained from a FasterRCNN[14] architecture trained on the same train test spilt. To condition the network for noisy bounding boxes at run-time, we add random noise to ground-truth bounding box input to the network during training, and keep the original bounding box annotation for accurate loss calculation. To ensure only a single detection per image, we hand-tune for the minimum probability to accept a detection from the object detector on the test set and keep the highest probability detection. We leave further study of the interaction between object detector and VoluMon for future work. Finally, we test a variant of VoluMon (denoted *VoluMon PostOpt*) where the pose estimate of the *VoluMon Primal Only* network is further refined for 50 optimization steps as described in Sec. IV-D.

Additionally, we consider a regression only technique (denoted *Optimization from scratch*), which optimizes Equation 12 without learning for both shape and pose parameters. We use the average depth of 100 points randomly sampled from the observed object points as the initial depth, and assume the projected centroid of the object is in the center of the bounding box detection. The extents of the segmented pointcloud with respect to the frame of reference of the camera set the initial shape parameters, and the initial rotation estimate is set to identity. If a valid initial size estimate cannot be obtained, an initial size estimate is set to approximately the minimum size by setting $\boldsymbol{\phi}_D = [-10.0, -10.0, -10.0]$. We consider two variants of the regression only technique with different numbers of optimization iterations, i.e., (*Optimize from scratch (500 iterations)*) and (*Optimize from scratch (50)*). Both *VoluMon PostOpt* and *Optimize from scratch* approaches sample 300 points once at run-time for the primal loss calculation and use a learning rate of 0.05. Finally, we compare our approach to a deep 6D object pose learning method [16] (*DOPE*) that requires the training data to be annotated with object dimensions and projected 3D bounding box vertices. *DOPE* does not require depth images at training time and the results are not further refined online.

All variants of VoluMon as well as the pure online optimization approach are implemented in Python using PyTorch. For VoluMon, we set $\alpha_S = 100, \alpha_y = 10.0, \epsilon_s = 3.0, \alpha_\delta = 150, \epsilon_v = 0.1$. We train with a batch size of 50 and a learning rate of 0.00001 for 14000 epochs, resetting the optimizer state halfway through training. We use open-source code released by the authors of [16] to train (*DOPE*) for 240 epochs, reducing the batch size at train time to 10 to optimize on lower cost GPUs.

We report translational prediction performance by comparing the true centroid of the object to the predicted centroid. Additionally, given true mesh models, we compare the distance between the mesh vertices $\{m_0 ... m_M\}$ and points on the surface of the prediction $\{p_0 ... p_N\}$ [26], i.e.,

$$ADI = \frac{1}{N} \sum_{j=0}^{N} \min_{i \in M} ||m_j - p_i||, \tag{13}$$

where $p_i$ are interpolated points from the surface of the prediction for the ellipsoid-based methods, and true mesh model transformed using the estimated parameters for *DOPE*. Although this metric is generally used to compare two known mesh models [18], [26], we apply it to measure surface fitting performance. Because *DOPE* is both a detection and estimation algorithm, we report pass rates where missed detections are removed entirely from the total number of samples considered, and pass rates where the missed detections are penalized (denoted NF). We also report the 3D IOU with groundtruth 3D bounding boxes.

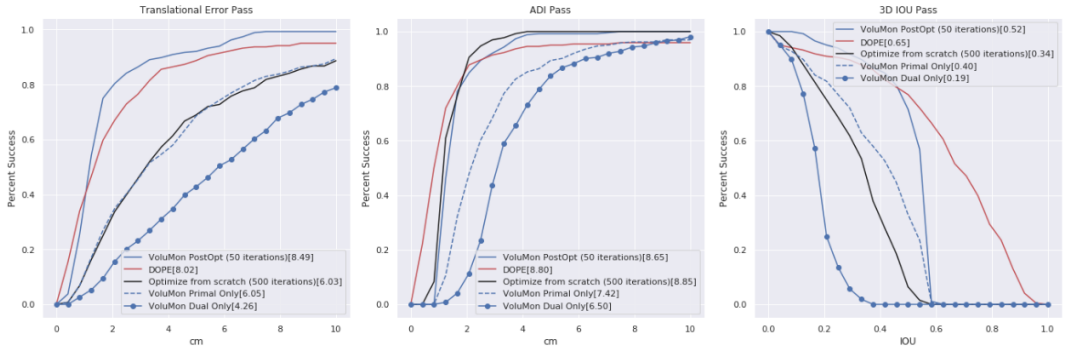
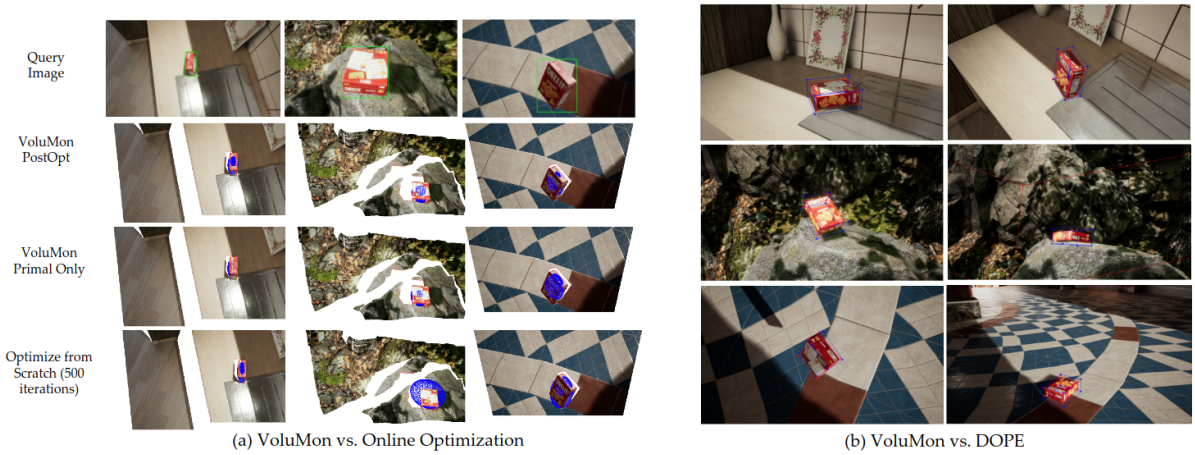



Fig. 3: The performance of various variants of VoluMon as well as the two baselines are shown for translation error (lower better), ADI (lower better), and 3D bounding box IOU (higher better). Trained on only 2D annotations, *VoluMon Primal Only* (blue dashed) performs similarly to an online optimization approach (black) with respect to translational error without requiring online optimization. After online optimization (solid blue) VoluMon approaches or surpasses the performance of *DOPE* (red), a deep 6D pose estimation approach, with respect to translational error and ADI. All ellipsoid-based methods underperform *DOPE* in the 3D IOU metric, which is not unexpected, given that *DOPE* is a 6D optimization assuming known object dimensions, while the other methods estimate both pose and dimension.




(a) VoluMon vs. Online Optimization (b) VoluMon vs. DOPE

Fig. 4: Qualitative results for selected methods. (a) Ellipsoid estimates visualized as blue meshes. Although *VoluMon PostOpt* requires less steps of online optimization compared to *Optimize from scratch (500 iterations)*, the initial pose estimates provided by *VoluMon Primal Only* enable *VoluMon PostOpt* to generate more accurate estimates of the ellipsoids. Potentially due to shape ambiguities, it can be difficult to estimate the extents of the object; by keeping the size estimate from *VoluMon Primal Only*, *VoluMon PostOpt* benefits from observations over a dataset to estimate size at run-time. (b) Ellipsoid estimates from *VoluMon Primal Only* visualized as blue bounding boxes, and the estimates from DOPE visualized as red bounding boxes using the ground-truth object size. VoluMon constrains the projected centroid of the object to fall within the 2D object bounding box, helping to avoid some failure cases of DOPE, e.g., top row, right column. Additionally, while VoluMon tends to slightly overestimate the size of the object, many of the estimates appear qualitatively reasonable.

### A. *Datasets*

We test on a subset of the Falling Things Dataset [19], which provides groundtruth bounding boxes, object pose, and geometry. We focus on the cracker box object data, which can be reasonably approximated by an ellipsoid, and is shown in Figure 4. Our constructed dataset features three different environment categories (kitchen, kite, and temple) with four variants in each yielding a total of twelve different environments with one hundred data points per camera. Each training sample includes the left RGB image, ground-truth depth image, groundtruth 2D bounding boxes for the left and right images, and segmentation images for all images. Each image contains a single object instance. For VoluMon, we filter out data points with bounding boxes extending past the image boundaries, and withhold one environment from each category `kitchen_0`, `kite_0`, `temple_0` for the test set, resulting in 1066 datapoints in *Cracker Train* and 264 datapoints in *Cracker Test*. For DOPE, we allow the network to train on all available data and test on *Cracker Test*. While training VoluMon, we use built-in PyTorch functions for random data augmentation, including adding random color jitter to the hue and saturation and random erasing. In practice, we randomly sample 4 points from the extracted surface points every epoch to calculate the primal loss.

### B. *Simulation Experimental Results*

We show the pass rate metrics for selected variants of each algorithm in Figure 3, and report area under the curve (AUC) and timing results in Table I. Qualitative samples are depicted in Figure 4. The AUC metric is calculated using a naive rectangular integral approximation. On simulated data, *VoluMon Primal Only* and *VoluMon Both* outperform *VoluMon Dual Only* with respect to AUC for translation and ADI. This matches our intuition that while using only two bounding box measurements may struggle to overcome shape and rotation ambiguity for certain objects, as reflected in the relatively poor ADI performance. However, it is important

| | Estimated Parameters | Train Time Requirements | Inference Time Requirements | Timing Mean/STD (ms) | Translation AUC | ADI AUC | 3D IOU | Translation AUC (NF) | ADI AUC (NF) | 3D IOU (NF) |
|---|---|---|---|---|---|---|---|---|---|---|
| ***VoluMon PostOpt (50 Iterations)*** | $d, R, t$ | Left bounding box<br>Left RGB image<br>Left segmentation<br>Depth image | Left bounding box<br>Left segmentation<br>Left RGB image<br>Depth image | 15/9 for network<br>341/3 for optimization<br>+ detection time (variable) | **8.49** | 8.65 | <u>0.52</u> | | | |
| ***VoluMon Dual Only*** | | Left bounding box<br>Right bounding box<br>Left RGB image | Left bounding box<br>Left RGB Image | 15/9 for network<br>+ detection time (variable) | 4.26 | 6.50 | 0.19 | | | |
| ***VoluMon Primal Only*** | | Left bounding box<br>Left RGB image<br>Left segmentation<br>Depth image | | | 6.05 | 7.42 | 0.40 | | | |
| ***VoluMon Both*** | | Left bounding box<br>Right bounding box<br>Left RGB image<br>Left segmentation<br>Depth image | | | 5.80 | 7.60 | 0.31 | | | |
| ***VoluMon Both Noisy*** | | Left bounding box<br>Right bounding box<br>Left RGB image<br>Left segmentation<br>Depth image | | | 4.96 | 7.05 | 0.28 | <u>4.94</u> | <u>7.02</u> | <u>0.27</u> |
| ***DOPE*** | $R, t$ | Projected cuboid corners<br>Bounding box size<br>RGB image | RGB image | 220/12 | <u>8.02</u> | <u>8.80</u> | **0.65** | **6.71** | **7.37** | **0.55** |
| ***Optimize from scratch (500 iterations)*** | $d, R, t$ | None | Left bounding box<br>Left segmentation<br>Left RGB image<br>Depth image | 3346/17<br>+ detection time (variable) | 6.03 | **8.85** | 0.34 | | | |
| ***Optimize from scratch (50 iterations)*** | | | | 341/3<br>+ detection time (variable) | 6.55 | 8.03 | 0.29 | | | |

TABLE I: Performance metrics over the various methods, with best performance bolded and the second best underlined. (NF) indicates when aggregate metrics consider instances when objects are not detected as failure to pass, where for VoluMon we impose a minimum detection probability from the auxiliary object detector. Timing results are collected using a GTX 1070Ti. We expect all variants of VoluMon to have similar run time for a single feed-forward pass of the network, and for the same number of optimization steps to have similar run times, reporting accordingly. *VoluMon* and *Optimize from scratch* timing results do not include detection and segmentation; the object detector used to generate test time detections for *VoluMon Both Noisy* requires roughly 200 ms per image, but the speed of the object detector can vary depending on the architecture. Timing results for *DOPE* include detection and PnP solving.

to note that *VoluMon Dual Only* requires only bounding box annotations from a stereo pair for training and a single camera at inference time, and yet achieves about 73% the ADI AUC of the highest performing method in the table. We additionally observe that training on both the primal and the dual objectives does not yield noticeable performance benefits when using groundtruth depth images, motivating using *Primal Only* in the post optimization experiments.

Our results also show that a single forward pass through the network of *VoluMon Primal Only* achieves a higher 3D IOU AUC score as compared to *Optimize from scratch (500 iterations)*, despite taking less than 1% of the time, not including detection time. Although all network-only VoluMon variants underperform the pure optimization methods in terms of ADI, the ADI metrics can fail to capture certain types of shape estimation errors as suggested by the 3D IOU results. As seen in Fig. 3, the translation performance of *VoluMon Primal Only* and *Optimize from scratch (500)* are also relatively similar. After applying the same number of further post optimization, *VoluMon PostOpt* outperforms *Optimize from scratch (500 iterations)* on translation AUC by over 40% and on 3D IOU AUC by over 50%, indicating that the estimates from VoluMon can be useful initial estimates for downstream algorithms.

Assuming that pointcloud data is available, *VoluMon PostOpt* outperforms *DOPE* with respect to translation AUC, and approaches the ADI performance with an ADI AUC that is 98% of *DOPE*'s. Although one of the contributions of [16] is to use large quantities of simulation data, the performance on our much smaller datasets still enables an approximately 60% pass rate for both the 2 centimeter threshold for translation and ADI. By leveraging the ellipsoid representation and object consistency properties, VoluMon does not require 3D annotations or *a priori* geometric models, and can still be further refined online if additional computation time and segmented depth information is available. We observe that *VoluMon PostOpt* requires more computation time than *DOPE*; future work includes direct integration into an object detection framework to reduce computation.

## VI. Real-World Experiments

To test the performance of VoluMon on real-world objects, we collected additional real-world datasets for four objects (a mug, a toy bus, an orange, and a bowl) using a ZED Mini stereo sensor. An initial set of annotations was generated using Mask RCNN pre-trained on COCO, with inaccurate annotations removed in a manual post-processing step. The physical objects are the same between the training and test set. One network per object was trained using bounding box annotations only on approximately 500 images per object, with a batch size of 25, and a learning rate of 0.0001 for the mug, orange, and bus, and 0.00001 for the bowl. We set a maximum centroid disparity of 300, a minimum size of 3 cm and a size scaling factor of 10 cm. As seen in Figure 5, without requiring detailed 3D models or 3D annotations, for some objects VoluMon produces qualitatively reasonable pose and size approximations on withheld evaluation data. We observed greater optimization instability on real data, and report results from the stable regimes of training. Notably, we also found that in practice, using the bounding box annotations only on real data generally outperformed other VoluMon variants, and enabled the most reasonable performance for the majority of objects investigated. This may be due to the noisier nature of the stereo pointcloud, seen in Figure 6c.

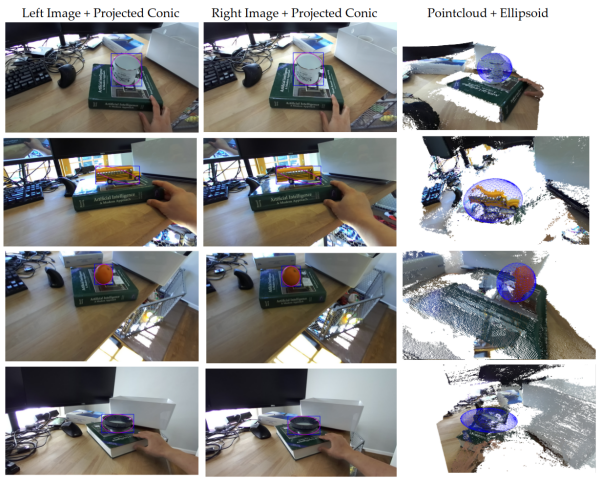



Fig. 5: Qualitative results from the MIT Desk Dataset. The first and second columns show projected conics (pink) and bounding boxes (blue) for the left and right images respectively; the second column visualizes the primal ellipsoid (blue mesh). While in this case bounding boxes from both the left and right sensor are used at train time, VoluMon predicts object geometry and pose using only the left image and bounding box input at run-time.

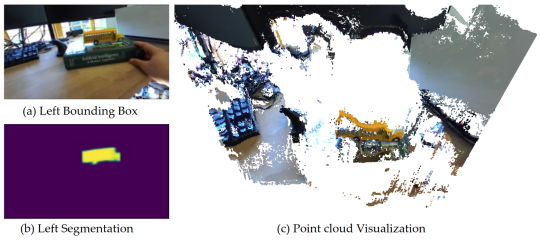



Fig. 6: To collect image space annotations on the real-world datasets, we leverage a pretrained network to annotate bounding boxes (a) and object segmentations (b) to use with the estimated pointcloud (c), reducing by-hand annotation burden.

## VII. Conclusions

We have presented VoluMon, a novel method for weakly supervised monocular object estimation. VoluMon leverages the primal and dual forms of the ellipsoid representation in addition to intra-class size properties to train a neural network to predict the parameters of a bounding ellipsoidal volume for an object. In addition to reducing computation time, potential future areas of investigation further include studies of performance on more diverse objects.

## References

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

[2] C. Rubino, M. Crocco, and A. Del Bue, "3D object localisation from multi-view image detections," *TPAMI 2018*,

[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. P. How, and N. Roy, "Robust object-based SLAM for high-speed autonomous navigation," in *Proc. ICRA 2019*.

[5] E. Sucar, K. Wada, and A. Davison, "NodeSLAM: Neural object descriptors for multi-view shape reconstruction," *Proc. 3DV 2020*,

[6] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3D modelling of novel objects from a single view," in *Proc. IROS 2010*.

[7] G. Vezzani, U. Pattacini, G. Pasquale, and L. Natale, "Improving superquadric modeling and grasping with prior on object shapes," in *Proc. ICRA 2018*.

[8] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals for accurate object class detection," in *Proc. NeurIPS 2015*.

[9] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. CVPR 2016*.

[10] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. CVPR 2012*.

[11] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3D object detection in the wild," in *Proc. WACV 2014*.

[12] J. Lee, S. Walsh, A. Harakeh, and S. L. Waslander, "Leveraging pre-trained 3D object detection models for fast ground truth generation," in *Proc. ITSC 2018*.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR 2016*.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[15] 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*.

[16] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

[17] A. Kundu, Y. Li, and J. M. Rehg, "3D-RCNN: Instance-level 3d object reconstruction via render-and-compare," in *Proc. CVPR 2018*.

[18] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," *Proc. RSS 2018*,

[19] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3D object detection and pose estimation," in *CVPR 2018 (Workshop)*.

[20] R. H. A. Zisserman, "Multiple view geometry in computer vision," 2004.

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

[22] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, "Deep fitting degree scoring network for monocular 3D object detection," in *Proc. CVPR 2019*.

[23] F. Solina and R. Bajcsy, "Recovery of parametric models from range images: The case for superquadrics with global deformations," *TPAMI*, vol. 12, no. 2,

[24] 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.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. ECCV 2016*.

[26] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Proc. ACCV 2012*.