

# ODAM: Object Detection, Association, and Mapping using Posed RGB Video

Kejie Li<sup>1,2</sup>, Daniel DeTone<sup>2</sup>, Steven Chen<sup>2</sup>, Minh Vo<sup>2</sup>, Ian Reid<sup>1</sup>, Hamid Rezatofghi<sup>3</sup>, Chris Sweeney<sup>2</sup>, Julian Straub<sup>2</sup>, and Richard Newcombe<sup>2</sup>

<sup>1</sup>The University of Adelaide, <sup>2</sup>Facebook Reality Labs Research, <sup>3</sup>Monash University

## Abstract

Localizing objects and estimating their extent in 3D is an important step towards high-level 3D scene understanding, which has many applications in Augmented Reality and Robotics. We present ODAM, a system for 3D Object Detection, Association, and Mapping using posed RGB videos. The proposed system relies on a deep learning front-end to detect 3D objects from a given RGB frame and associate them to a global object-based map using a graph neural network (GNN). Based on these frame-to-model associations, our back-end optimizes object bounding volumes, represented as super-quadratics, under multi-view geometry constraints and the object scale prior. We validate the proposed system on ScanNet where we show a significant improvement over existing RGB-only methods.

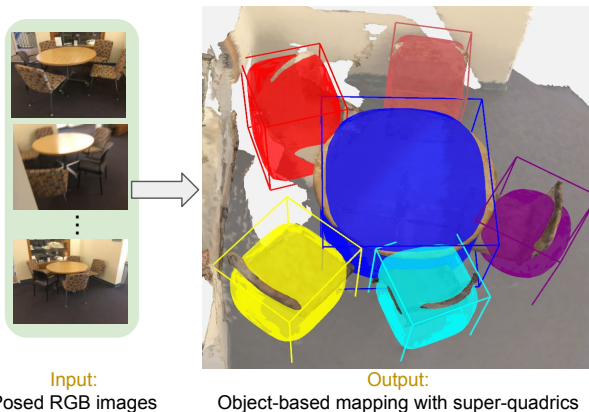


Figure 1: ODAM overview. Given a posed RGB video, ODAM estimates oriented 3D bounding volumes of objects represented by super-quadratics in a scene.

## 1. Introduction

Endowing machine perception with the capability of inferring 3D object-based maps brings AI systems one step closer to semantic understanding of the world. This task requires building a consistent 3D object-based map of a scene. We focus on the space between the category-level semantic reconstructions [29] and object-based maps with renderable dense object models [27, 45] and represent objects by the 3D bounding volumes from posed RGB frames. As an analogy to the use of 2D bounding boxes (BBs) in images, a 3D bounding volume presents a valuable abstraction of location and space, enabling for example, object-level planning for robots [13, 15], learning scene-level priors over objects [55], or anchoring information on object instances. A robust way of inferring bounding volumes and associated views of individual objects in a scene is a stepping stone toward reconstructing, embedding and describing the objects with advanced state-of-the-art methods such as NeRF [32], and GRAF [47], which commonly assume a set of associated frames observing an object or part of a scene that can

be obtained from the proposed reconstruction system.

Nevertheless, this task of localizing objects and estimating their extents in 3D using RGB-only videos presents a number of challenges. First, despite the impressive success of deep learning methods for 2D object detectors [7, 16, 43], recent efforts that formulate 3D object mapping as a single-view 3D detection problem [5, 24, 33] suffer from accuracy due to the depth-scale ambiguity in the perspective projection (as demonstrated empirically in Sec. 4.2). Second, unlike estimation of 3D points from multiple 2D observations that has been studied extensively in SfM and SLAM [6, 14, 22, 34, 53], there has been little work and consensus on how to leverage multi-view constraints for 3D bounding volume location and extent [35, 60]. Specifically, the representation for 3D volume and how to formulate a suitable energy function remain open questions. Third, the crucial problem that needs to be solved prior to multi-view optimization is the associations of detections of individual 3D object instances from different viewpoints, where unlike SfM or SLAM incorrect association noticeably biases

the 3D object localization. However, this problem is under-explored for cluttered indoor environments, where specific problems such as having multiple objects with near identical visual appearance and heavy occlusion (*e.g.*, multiple chairs closely arranged in a room as can be seen in Fig. 6) are commonplace. Depth ambiguity and partial observations complicate the data association problem.

We propose ODAM, a novel framework that incorporates a deep learning front-end and multi-view optimization back-end to address 3D object mapping from posed RGB videos. The advantage of using RGB-only over RGB-D is significantly less power consumption. We assume the poses of the images are known; these are readily available with modern mobile/AR devices. The front-end first detects objects of interest and predicts each object’s 2D attributes (2D BB, object class), as well as its 3D BB parameterized by 6 Degree-of-Freedom (DoF) rigid pose and 3 DoF scale given a single RGB frame as shown in Fig. 2. The primary use of the 3D attributes for each detection is to facilitate data association between a new frame and the current global 3D map. Concretely we develop a graph neural network (GNN) which takes as inputs the 2D and 3D attributes of the current frames detections and matches them to existing object instances in the map. The front-end of can run 6 fps on average on a modern GPU on cluttered scenes such as those in ScanNet [10].

The back-end of ODAM is a multi-view optimization that optimizes each object’s oriented bounding volume represented by a super-quadric surface given multiple associated 2D bounding box (BB) observations. Previous object-level SLAM frameworks have adopted either cuboids [60] or ellipsoids [18, 35] as their object representation, but they are often not a good model for the extent of a generic object as depicted in Fig. 3. Super-quadric – a unified representation for shape primitives including cuboids, ellipsoids, and cylinders – permits blending between cuboids and ellipsoids (and cylinders) and can therefore provide a tight bounding volume for the multi-view optimization. While super-quadric has been used to fit point cloud data [39, 40, 49] or recently parse object shapes from a single image using a deep network [38], we present the first approach to optimize super-quadrics given multiple 2D BB observations to the best of our knowledge. Besides the representation, we realize that the 2D BBs given by the object detector are not error free due to occlusions in cluttered indoor environments. We incorporate category-conditioned priors in the optimization objective to improve the robustness.

**Contribution.** Our contributions are threefold: (1) we present ODAM, a novel online 3D object-based mapping system that integrates a deep-learning front-end running at 6 fps, and a geometry-based back-end. ODAM is the current best performing 3D detection and mapping RGB-only systems for complex indoor scenes in ScanNet [10];

(2) we present a novel method for associating single-view detections to the object-level. Our association employs a novel attention-based GNN taking as inputs the 2D and 3D attributes of the detections; (3) we identify the limitations of common 3D bounding volume representations used in multi-view optimization and introduce a super-quadric-based optimization under object-scale priors which shows clear improvements over previous methods.

## 2. Related Work

**3D object-based mapping.** Approaches to 3D object-based mapping can be broadly classified into two categories: learning-based and geometry-based. The first category mostly extends existing 2D detectors to also output 3D bounding box from single images [23, 26, 28, 33, 54, 58]. If a video sequence is available, the single-view 3D estimations can be fused using a filter or a LSTM to create a consistent mapping of the scene [5, 20, 24]. Yet, the fused 3D detections might not satisfy multi-view geometry constraints. While the front-end of our proposed system is inspired by these learning-based approaches, we notice that single-view 3D inference is inaccurate because the inherent scale and depth ambiguity in 2D images and solve this issue with a back-end multi-view optimization. The second category focuses on estimating the bounding volume of an 3D object given 2D detections from multiple views in a similar way to the reprojection error used in SfM and SLAM. [9, 44] estimate the 3D ellipsoid representing the bounding volume of an object by minimizing the discrepancies between the projected ellipsoid and the detected 2D bounding boxes. QuadricSLAM [35] represents objects as dual quadrics to be optimized with a novel geometric error and extends it to a full SLAM system. CubeSLAM [60] uses 3D cuboids in the optimization and enforces reprojection error on the vertices of the 3D cuboids. Our proposed multi-view optimization uses super-quadric – a representation subsuming both ellipsoid and cuboid – with the energy function formulation using joint 2D BBs and a scale prior constraint.

**Object-based mapping with 3D shape estimation.** Extending beyond 3D oriented bounding boxes, several works focus on estimating dense object shapes via shape embedding [45] or CAD model retrieval [27] given posed RGB video. RfD-Net [36] explores completing full object shapes by first detecting 3D bounding boxes followed by a shape completion network for each detected object from 3D point cloud. Although these methods estimate high-resolution object mapping, they require known 3D shape priors. We do not assume prior knowledge of CAD models and instead focus on instance-agnostic pose estimation.

**Associating detection across video frames.** Associating object detections of the same 3D object instance across multiple frames has been studied in different contexts, and

most prominently in the context of Multiple Object Tracking (MOT). MOT focuses on tracking dynamic objects (e.g. cars and pedestrians) and often follows frame-to-frame paradigm by heavily exploiting the discriminative visual appearance features of objects [20, 52]. Until the recent end-to-end tracking approaches [4, 30, 51], most approaches rely on simple motion continuity priors [11, 31, 57] to link instances. More closely related to ODAM is Weng *et al.* [56] that uses a GNN to learn a matching cost given both point cloud and RGB images. Our proposed framework takes as input RGB-only images, and hence solves a more difficult but ubiquitous problem. Moreover, rather than associating people or cars with discriminative visual appearance, we focus on indoor static object mapping where we associate a sets of highly repetitive objects from drastically different views (e.g. front and back of a chair) in a frame-to-model fashion. Prior methods working for indoor environments resort to handcrafted matching by IoU [24] or SLAM feature point matching [18, 60], whereas we learn the matching using the GNN.

**3D detection from 2.5D and 3D input.** There are several methods for 3D detection using 3D input [42, 48, 61] or single RGBD images [8, 50]. Methods for instance segmentation [17, 19, 59] given 3D input are also able to produce 3D bounding boxes. Since depth information directly resolves the scale ambiguity of an observed object, these methods solve a strictly easier problem.

### 3. Method

The goal of ODAM is to localize objects and estimate their bounding volume accurately in 3D given posed *RGB-only* image sequence. As shown in Fig. 2, given an RGB frame, the front-end first detects objects and predicts their 2D and 3D attributes in the camera coordinate frame (Sec. 3.1). These detections are associated to existing object instances in the map or become a new object instance by solving an assignment problem using a GNN (Sec. 3.2). Given the association from the front-end, our back-end system optimizes a super-quadratic surface presentation of each object from multiple associated 2D BB detections and category-conditioned object-scale priors from all associated views. (Sec. 3.3).

#### 3.1. Single-view 2D and 3D Object Detection

ODAM first detects objects of interest given a new RGB frame. Our detector is a single-view 3D detector that estimates not only 2D attributes – 2D BB and object class – but also 3D attributes – translation  $t_{co}$ , rotation  $R_{co}$ , and 3D BB dimensions  $s$  with respect to the local camera coordinate frame. Specifically, we estimate  $t_{co}$  by predicting its depth and 2D center on the image.  $R_{co}$  is formulated as a classification on three Euler angles.

#### 3.2. Association of Detections to Object-based Map

Detections from the single-view detector are matched to existing object instances in the map using an attention-based GNN as opposed to handcrafted data association algorithms used in prior art [24, 27]. The benefit of using a GNN for data association is twofold. First, different attributes (e.g. 2D BB, 3D BB, object class) can be taken as joint input to the network to extract more discriminative features for matching. Second, instead of only considering pair-wise relationships in handcrafted data association methods, the attention mechanism in GNN aggregates information from other nodes in the graph for more robust matching. Thus, our GNN can infer the association of an object detection from the full set of objects in the scene, as visualized in Fig. 2.

Our association graph is implemented as a GNN where each node is a feature descriptor comprising of 2D and 3D information of an object detection and edges connect (1) among previously associated detections of an object in the object fusion; (2) a new detection to other detections and a fused object feature vector to other object feature vectors for self-attention; (3) a new detection to fused object feature vectors for cross-attention, as shown in Fig. 2. This graph predicts a matching between a set of input detection and existing objects in the map. For every object in the map, we fuse its descriptors from all associated views using self-attention GNN layers. These fused descriptors are matched to the descriptor of the newly detected objects using self- and cross-attention GNN layers.

**Input detection features.** The  $m^{th}$  new detection at frame  $t$  is represented by a feature descriptor  $d_m^t \in \mathbb{R}^{16}$  comprising the frame ID, the detected 2D BB, object class, detection score, 6 DoF object pose, and 3 DoF object scale given by the monocular detector. The  $n^{th}$  object instance is represented by a set of associated detections in previous frames  $\{d_n^{t_0}, d_n^{t_1}, \dots, d_n^{t_l}\}$ , where  $d_n^{t_l}$  is a previously associated detection of the  $n^{th}$  object instance at frame  $t_l$ . To facilitate association of the detections in new RGB frame to the mapped objects, the detected 2D BB and 6 DoF object pose in  $d_n^{t_l}$  are replaced by the projection from the estimated 3D bounding volume to the current frame coordinate.

**Object fusion.** We first fuse all associated detections of a mapped object using a self-attention GNN into a single feature descriptor vector:

$$o_n = f_d(\{d_n^{t_0}, d_n^{t_1}, \dots, d_n^{t_l}\}), \quad (1)$$

where  $f_d(\cdot)$  is the self-attention GNN taking as input a set of previously associated detections of an object instance, and  $o_n \in \mathbb{R}^{256}$  is the fused feature vector for the object instance for data association. This step allows information across observations of the same object from different viewpoints to be aggregated before matching to the new detection at the current frame.

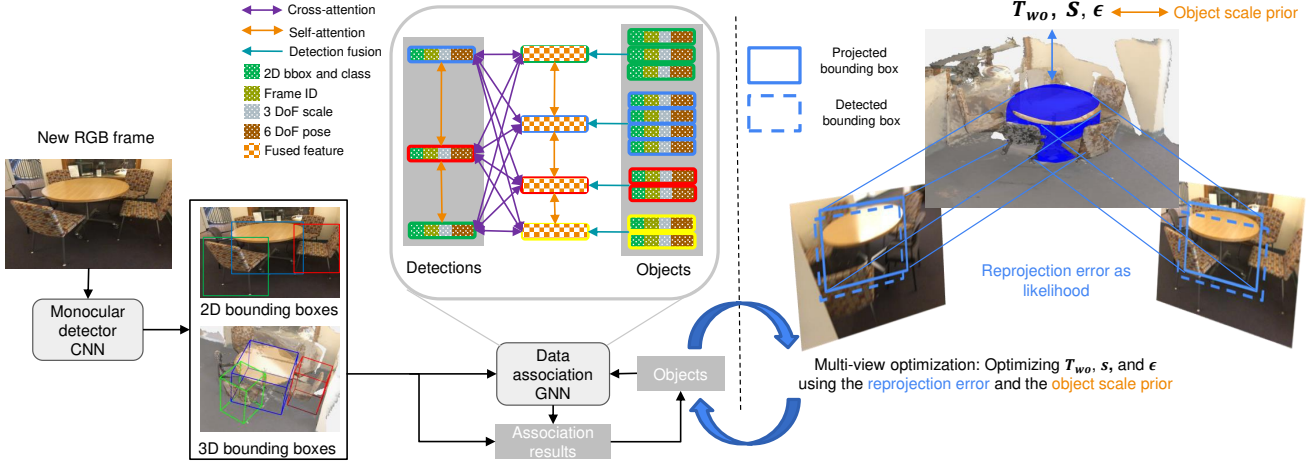


Figure 2: ODAM pipeline. Given a new RGB frame, a single-view detector (Sec. 3.1) detects objects at the current frame. A GNN takes as inputs the new detections and existing objects in the map to predict the assignment matrix (Sec. 3.2). Concurrent with the front-end of the system (*i.e.* detection and association), the location and extent of each object are represented by a super-quadric, which is optimized using the associated 2D BBs and category-conditioned scale prior (Sec. 3.3).

**Frame-to-model association.** After the fusion step, the frame-to-model data association becomes a bipartite matching problem, where the two disjoint subsets are the fused vectors of  $m$  existing objects and  $n$  new detections at the current frame  $t$  respectively. This matching problem is solved by the second part of the GNN, which contains a stack of alternating self-attention layers aggregating information within the subset and cross-attention layers aggregating information from the other subset. The assignment matrix  $M \in \mathbb{R}^{m \times n}$  is computed as:

$$M = f_m(\{\mathbf{o}_0, \mathbf{o}_1, \dots, \mathbf{o}_m\}, \{\mathbf{d}_0^t, \mathbf{d}_1^t, \dots, \mathbf{d}_n^t\}), \quad (2)$$

where  $f_m(\cdot)$  is second part of the GNN taking as input objects' fused vectors  $\mathbf{o}_n$  and new detections  $\mathbf{d}^t$ . Please refer to Sec. 3.4 and the supplementary material for more network and training details.

### 3.3. Multi-view Optimization

Instead of relying on a single-view 3D detector to solve the ill-posed monocular 3D detection problem, we propose a multi-view optimization for accurate 3D object mapping given multiple associated 2D BBs obtained from the previous step (Sec. 3.2). The key to the optimization is representing a bounding volume via a super-quadric with the realization that both ellipsoid and cuboid used in prior art are only suitable to a subset of object shapes. Specifically, given multiple 2D BBs, the estimated 3D bounding volume is a convex set bounded by the intersection region of all frustums. As the number of viewpoints increases, the convex set converges to the convex hull (*i.e.* the tightest convex set of the object shape). However, neither ellipsoid or

cuboid is flexible enough to approximate the convex hull for generic objects. For instance, while an ellipsoid is suitable for round objects, it introduces inherent inconsistency when representing a box-like object as shown in Fig. 3. Super-quadric alleviates this issue by using the best fitted shape primitive in the family. Although dense object shape representations (*e.g.* shape codes [37], or CAD models [27]) do not suffer from the inconsistency in projection, they require knowledge of instance-level object shape.

**Super-quadric formulation.** We represent an object's bounding volume in 3D by a super-quadric. The canonical implicit function of a super-quadric surface has the following form [3]:

$$f(\mathbf{x}) = \left( \left( \frac{x}{\alpha_1} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{y}{\alpha_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_1}{\epsilon_2}} + \left( \frac{z}{\alpha_3} \right)^{\frac{2}{\epsilon_1}}, \quad (3)$$

where  $\mathbf{x} = [x, y, z]$  is a 3D point,  $\alpha = [\alpha_1, \alpha_2, \alpha_3]$  controls the scale on three axes (*i.e.* the object's 3D dimensions), and  $\epsilon_1$ , and  $\epsilon_2$  decide the global shape curvature. The shape transition from an ellipsoid to a cube controlled by  $\epsilon_1$ , and  $\epsilon_2$  is visualized at Fig. 4.

A point  $\mathbf{x}$  on the surface of a super-quadric can be transformed from the canonical coordinate to the world coordinate by a 6 DoF rigid body transformation matrix  $T_{w0} \in \text{SE}(3)$ . Thus, a super-quadric in the world coordinate is parameterized by  $\theta \in \mathbb{R}^{11}$ , comprising of  $T_{w0}$  (6 DoF to represent the rigid body transformation), and the 5 parameters of the super-quadric  $\alpha$  and  $\epsilon_1, \epsilon_2$ .

**Optimization objective.** The detected 2D BBs are inevitably inaccurate and noisy. While existing methods using multi-view constraints for 3D object mapping [18, 27,




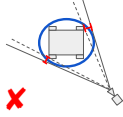
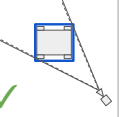
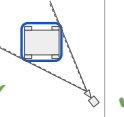
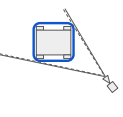

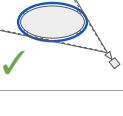
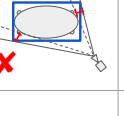
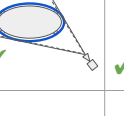
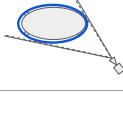
View	Side	Top	Top	Top	Top
Representation	-	Ellipsoid	Cuboid	Super-quadric (ours)	Dense object shape (Shape code/CAD models)
Boxy Object (e.g. chair)					
Round Object (e.g. round table)					
Requires Instance-level Models	-	No	No	No	Yes

Figure 3: Limitations of object representations in multi-view optimization from 2D BB observations. Ellipsoid and cuboid are only suitable for a subset of objects. Dense representations require instance-level models.

35, 60] only consider the reprojection error, we observe that prior knowledge on the object’s 3D scale can improve the robustness of the estimation. To incorporate the prior knowledge about object scale, we formulate object-based mapping as Maximum a Posterior (MAP) estimation of each object’s super-quadric parameters  $\theta$ . With the reprojection likelihood  $P(\mathbf{b}_i|\theta)$  and the category-conditioned scale prior  $P(\theta)$  the MAP problem is:

$$\arg \max_{\theta} P(\theta|\mathbf{B}) = \arg \max_{\theta} P(\theta) \prod_i P(\mathbf{b}_i|\theta), \quad (4)$$

where  $\mathbf{B} = \{\mathbf{b}_0, \dots, \mathbf{b}_N\}$  is a set of  $N$  associated 2D detected BBs.  $\mathbf{b}_i$  is the detected 2D BB at frame  $i$  described by its four corner points  $[x_{\min}, x_{\max}, y_{\min}, y_{\max}]$ . Assuming zero-mean Gaussian noise on the 2D BB detection corners, the reprojection likelihood is

$$P(\mathbf{b}_i|\theta) = \mathcal{N}(\mathbf{b}_i|\hat{\mathbf{b}}_i, \sigma^2), \quad (5)$$

where  $\sigma$  is the assumed image noise and  $\hat{\mathbf{b}}_i$  is the super-quadric’s projection. It is computed as:

$$\hat{\mathbf{b}}_i = \text{Box}(\pi(\mathbf{T}_{cw}\mathbf{T}_{wo}\mathbf{X}_o)) \quad (6)$$

$$\text{Box}(\mathbf{X}) = [\min_x \mathbf{X}, \max_x \mathbf{X}, \min_y \mathbf{X}, \max_y \mathbf{X}]. \quad (7)$$

The transformation  $\mathbf{T}_{cw}\mathbf{T}_{wo}$  brings sampled surface points of the super-quadric in canonical coordinates,  $\mathbf{X}_o = S(\alpha, \epsilon_1, \epsilon_2)$ , into camera coordinates before projecting them into the image using the perspective projection function  $\pi$ . We obtain  $\mathbf{X}_o$  using the equal-distance sampling techniques of super-quadrics [41]. We model the prior object scale distribution of each object category as  $P(\theta) = \mathcal{N}(\alpha|\mu_0, \Sigma_0)$  using a multi-variate Gaussian distribution. Ideally this prior would capture the uncertainty of the averaged detector-predicted 3D BB  $\mu_0$  for proper Bayesian

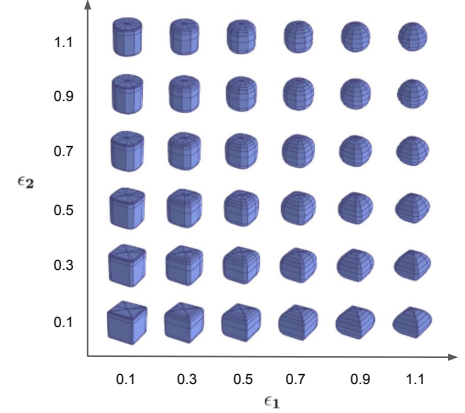


Figure 4: Super-quadric visualization. Different  $\epsilon_{1,2}$  values include cuboids, ellipsoids, and cylinders (figure credit [1]).

MAP estimation. While there are ways to train CNNs to produce uncertainty estimates [21], we found that we can simply use the variance  $\Sigma_0$  of the scale distribution of each object category in Scan2CAD [2] as a proxy. Intuitively, since the detector is trained on this distribution,  $\Sigma_0$  is a conservative upper-bound on the variance—a well trained 3D BB detector should do significantly better.

### 3.4. Implementation

**Detector training.** Our detector is built upon DETR [7], a state-of-the-art 2D object detector that predicts objects as a set without post-processing. We add three additional heads to DETR, each of which comprises three 512-dimension fully-connected layers, for object depth, 3D dimensions, 3D BB orientation respectively. We fine-tune our detector from the pre-trained network weights on MSCOCO dataset [25] for 10 epochs using ScanNet images and Scan2CAD annotations. Although we use DETR in this work, other detectors such as MaskRCNN [16] can also be adopted.

**Graph neural network details.** A 3-layer MLP encoder is used to map the input to a 256D feature vector. The detection fusion block contains four self-attention layers producing 256D fused features. The matching network for the fused features and frame detections is similar to SuperGlue [46] except we use six alternating cross- and self-attention layers.

**Optimization details.** All parameters of the super-quadrics except  $\epsilon_{1,2}$  which are initialized to 1, are initialized using the average of associated single-view 3D prediction. We sample 1000 points on the super-quadric surface for the optimization and found an assumed image variance of the 2D BB detector of  $\sigma^2 = 20$  worked well. We use the Adam optimizer in Pytorch to optimize the logarithm of the posterior in Eq. (4) for 20 iterations for every 50 associated 2D obser-

Prec./Rec./F1 IoU > 0.25	bathub	bookshelf	cabinet	chair	display	sofa	table	trashbin	avg.
Vid2CAD [27]	45.5/30.0/36.1	18.0/12.7/14.9	46.3/34.6/39.6	70.1/78.6/74.1	44.1/42.8/43.5	40.8/45.1/42.8	46.6/50.2/48.3	60.2/37.9/46.5	56.1/54.5/55.2
MOLTR [24]	67.5/41.6/51.5	42.8/21.3/28.4	62.7/22.8/33.5	58.7/77.4/68.6	17.7/34.5/23.4	69.4/52.2/59.5	63.5/57.4/60.3	49.0/42.6/45.6	54.2/55.8/55.0
ODAM (ours)	58.6/34.2/43.2	52.0/25.1/33.7	63.0/26.4/37.2	68.3/78.7/73.1	37.5/37.5/37.5	75.9/53.1/62.5	65.5/58.9/62.0	67.8/60.8/64.1	64.7/58.6/61.5
IoU > 0.5									
Vid2CAD [27]	2.5/1.6/2.0	0.0/0.0/0.0	7.7/5.7/6.6	29.2/32.8/30.9	0.0/0.0/0.0	0.8/0.8/0.8	6.7/7.2/6.9	23.2/14.6/17.9	16.8/16.3/16.5
MOLTR [24]	10.3/6.6/8.1	8.6/4.7/6.1	19.6/8.1/11.5	20.0/28.4/23.5	1.8/4.1/2.5	20.0/15.9/17.7	12.1/11.7/11.9	13.0/12.9/12.9	15.2/17.1/16.0
ODAM (ours)	14.3/8.3/10.5	11.5/5.7/7.6	25.9/10.9/15.3	39.0/44.8/41.7	7.7/7.7/7.7	39.2/27.4/32.3	26.0/23.3/24.6	31.6/28.0/29.5	31.2/28.3/29.7

Table 1: Quantitative ScanNet evaluation. ODA M outperforms MOLTR [24] and Vid2CAD [27] in four classes at IoU > 0.25 and all classes at IoU > 0.5 respectively.

variations followed by a final optimization for 200 iterations at the end of the sequence.

## 4. Experiments

We evaluate the performance of our object-based mapping using the precision and recall metrics on ScanNet [10] and Scan2CAD [2]. Because the original annotations do not provide amodal 3D BBs, following prior art [24, 27], we use the amodal 3D BB annotations from Scan2CAD as ground-truth. The precision is defined as the percentage of estimated super-quadratics being close enough to an annotated ground-truth 3D BB. The recall is the percentage of ground-truth 3D BBs that are covered by an estimated super-quadratic. Specifically, a super-quadratic is considered a true positive if the Intersection-over-Union (IoU) between its minimum enclosing 3D oriented 3D BB and a ground-truth BB in the same object class is above a pre-defined threshold. We use 0.25 and 0.5 in our experiments. A ground-truth BB can only be matched once to penalize repeated objects. Note that we do not use mean Average Precision (mAP) which is typically used for the object detection because the proposed system outputs an unordered set of 3D bounding volumes without confidence scores.

### 4.1. Comparing with RGB-only Methods

We compare ODA M against two previous posed RGB videos methods, Vid2CAD [27] and MOLTR [24], on ScanNet. These methods use are handcrafted data association (3D GIoU in MOLTR and Vid2CAD uses a combination of 2D IoU and visual appearance). MOLTR does not use multi-view geometry but fuses monocular 3D predictions via a filter in 3D, and the multi-view optimization in Vid2CAD lacks the scale prior in ours to alleviate the effect of inaccurate 2D observations. In contrast, we use attention-based GNN for association, followed by multi-view optimization. Tab. 1 shows precision, recall, and F1 score comparison per class at IoU thresholds of 0.25 and 0.5. Overall, ODA M outperforms Vid2CAD and MOLTR by about 6% at IoU > 0.25, and about 14% at IoU > 0.5. As shown in Fig. 6, we can see duplicated objects in MOLTR and Vid2CAD due to failure in data association. Notably, our multi-view optimization estimates accurate oriented bound-

Methods	Components			matching accuracy
	GNN	monocular 3D	F2M association	
Baselines	✓	✓	✗	0.86
	✓	✗	✓	0.84
	✗	✓	✓	0.85
ODAM (ours)	✓	✓	✓	<b>0.88</b>

Table 2: Ablation study on the learned data-association component of ODA M. The full model using GNN, monocular 3D detection, and frame-to-model (F2M) association achieves the best result.

ing volumes of large objects (*e.g.* tables), whereas MOLTR and Vid2CAD often produce misaligned results.

### 4.2. Ablation study

We validate the design choices in all key parts of ODA M using three ablation studies.

**Data association.** The key aspects we consider in this ablation study are: (1) GNN vs. handcrafted pairwise cost, (2) the effect of single-view 3D attribute estimation vs. 2D only attribute in data association, (3) frame-to-model association vs. frame-to-frame association. When the GNN is not used, we use 3D BB IoU as the matching cost following Li *et al.* [24]. To validate the importance of the detection fusion block in the GNN (*i.e.* frame-to-model association), we compare it against a baseline GNN which only takes as input the latest observation of existing object instances, which can be considered as a frame-to-frame association. Besides the final 3D mapping result, we also use the matching accuracy as a direct measurement for the data association algorithms. Tab. 2 shows all three key components contribute to the performance gain. Fig. 5 visualizes how the attention scores change across different layers in the GNN.

**Shape representation.** Tab. 3 shows that optimizing with the super-quadratic representation performs better than cuboid and ellipsoid by 2.5% and 9%, respectively. Cuboid outperforms ellipsoid because a considerable amount of objects are cuboid-like in the evaluated object classes. Further qualitative comparisons can be found in the supplement.

**Optimization.** Tab. 3 shows the effect of the back-end multi-view optimization and the scale prior terms in the objective function. The “no optimization” results are obtained by taking the average of associated monocular 3D predictions without any multi-view optimization and have

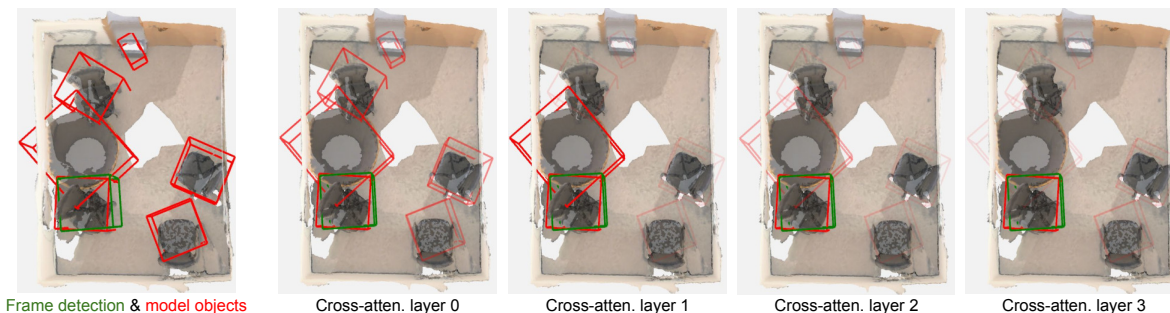


Figure 5: Visualization of GNN attention. The cross-attention scores of the 3D detection from the current frame detection (green) and model objects (red) are shown across various layers. Higher attention scores correspond to more opaque red BBs. The spread of the cross-attention shrinks and focuses on the correct assignment in deeper layers of the GNN.

Switched component		Result (Prec./Rec./F1)
Shape representation	ellipsoid	21.9/19.6/20.7
	3D cuboid	28.5/26.1/27.2
Optimization	no optimization	25.2/22.8/23.9
	wo/ scale prior	22.9/21.3/22.1
ODAM (ours)		<b>31.2/28.3/29.7</b>

Table 3: Ablation study on different shape representations and the multi-view optimization. The combination of super-quadratic representation and scale-prior in the multi-view optimization leads to the best performance.

the worst in the group. This indicates that single-view 3D detector alone is not sufficient for object-based mapping. Using only 2D bounding box observations for the multi-view optimization is also suboptimal, giving a minor 1.8% deterioration. Our full approach (using 2D bbox and prior jointly) outperforms the “no optimization” baseline by 5.8%. To better demonstrate how the errors in 2D BBs affect the optimization, we show how the performance gap between optimization w/ and wo/ the prior term changes as the errors in the 2D BBs increase in the supplement.

### 4.3. Comparing with RGB-D methods

This comparison is to identify the current gap between RGB and RGB-D methods. We compare to VoteNet [42], a state-of-the-art 3D object detection network using colored point clouds. Compared to RGB-only, the additional depth information, which is fused into a point cloud before 3D object detection, significantly simplifies the task. The 3D structure is explicitly represented and becomes an input to the 3D object-detection system and does not have to be inferred by the system. Yet the RGB-only methods are valuable because depth sensors consume additional power and most consumer-grade devices have limited range.

We train our detector and the GNN using the original ScanNet annotations to be consistent with VoteNet. We select the score threshold in VoteNet that leads to the best F1 score. As shown in Tab. 4, we achieve comparable or even

superior performance to VoteNet in some object classes (e.g. bed, table, desk, fridge, toilet, and bath). This is because these objects are normally arranged distantly to other object instances in the same class, making the data association easier. On the other hand, our method struggles with thin objects, such as door, window, picture, and curtain, because a small localization error results in a significant drop in 3D IoU leading to worse F1 score.

### 4.4. Run-time analysis

All experiments are run on a Nvidia GeForce GTX 1070 GPU. The monocular detector can run at about 10 fps. Although the inference time of the GNN grows linearly with the number of objects in the map, the GNN runs at 15 fps on average in all ScanNet validation sequences. Overall, the front-end of ODAM can achieve around 6 fps. A naive back-end optimization using the Pytorch Adam optimizer takes 0.2 seconds for 20 iterations. This back-end optimization is not time critical and can be run in a parallel thread. It could also be accelerated significantly using second order methods such as implemented in GTSAM [12].

## 5. Conclusion

We presented ODAM, a system to localize and infer 3D oriented bounding volumes of objects given posed RGB-only videos. Key to ODAM is (1) an attention-based GNN for robust detection-to-map data association, and (2) a super-quadratic-based multi-view optimization for accurate object bounding volume estimation from the associated 2D BB and class observations. ODAM is the best performing RGB-only method for object-based mapping. The fact that the proposed RGB-only methods can close the accuracy gap to RGB-D methods in a subset of object categories is encouraging and points to a future where depth cameras are unnecessary for 3D scene understanding.

**Acknowledgment** KL and IR gratefully acknowledge the support of the ARC through the Centre of Excellence for



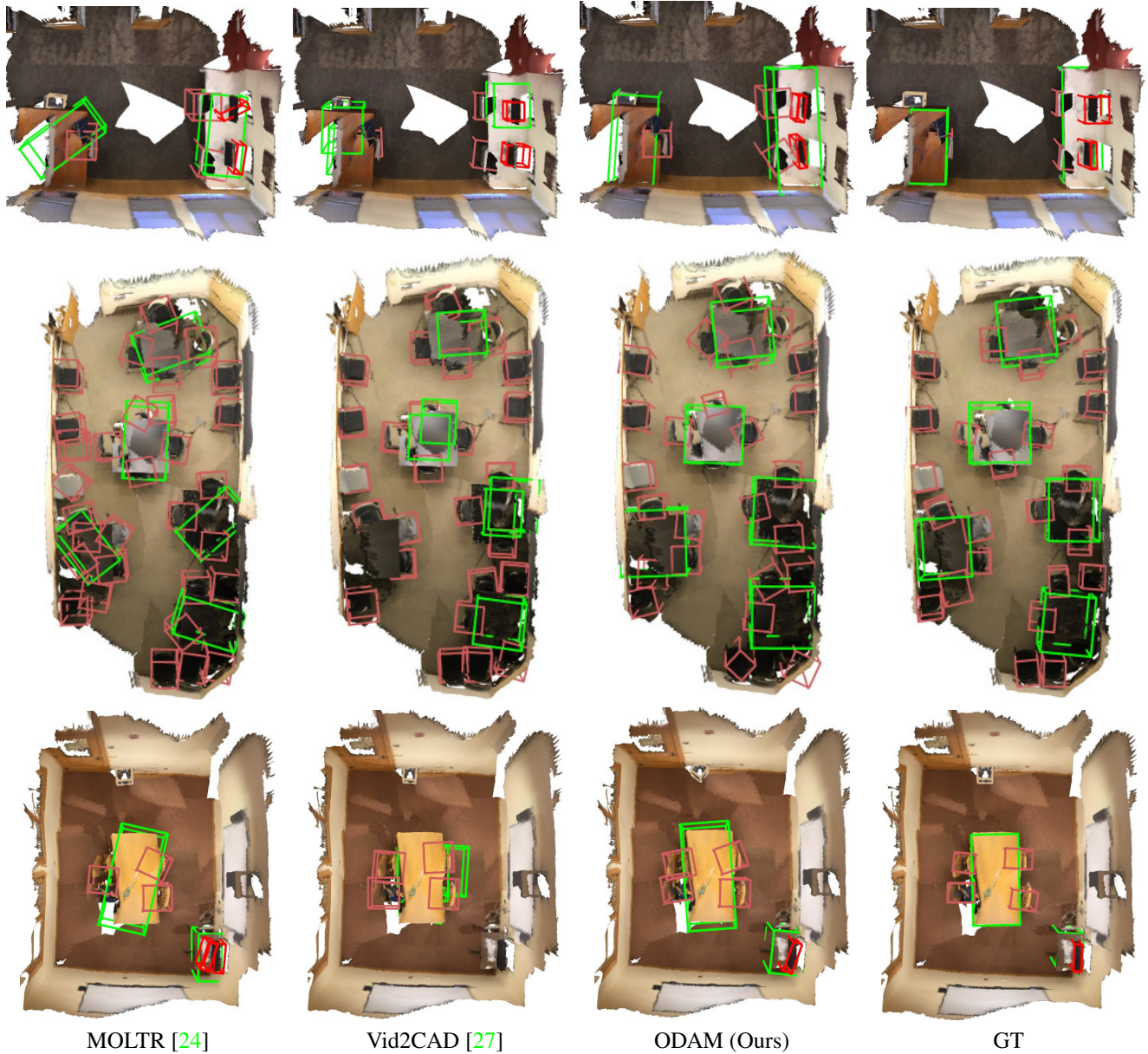


Figure 6: Qualitative comparison on ScanNet sequences. The colors of 3D BBs denote different categories. Both Vid2CAD and MOLTR suffer from replicated objects due to data association failure. 3D bounding boxes from our method are closer to ground-truth boxes thanks to our robust multi-view optimization.

F1	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	desk	curtain	fridge	shower	toilet	bath	others
VoteNet [42]	<b>40.9</b>	<b>88.1</b>	<b>85.2</b>	<b>79.8</b>	60.0	<b>53.0</b>	<b>40.7</b>	<b>50.0</b>	<b>12.5</b>	<b>52.3</b>	62.2	<b>50.0</b>	47.4	<b>50.5</b>	90.7	53.9	<b>91.5</b>
ODAM (ours)	22.1	87.7	74.9	61.8	<b>65.6</b>	12.5	12.8	40.5	7.4	9.3	<b>65.0</b>	13.1	<b>48.7</b>	41.2	<b>93.1</b>	<b>64.8</b>	83.3

Table 4: Comparison to VoteNet [42]. VoteNet relies on colored 3D point cloud which greatly simplifies 3D object localization. ODAM performs similarly in most categories but struggles with thin objects such as door, window and curtain.



- [2] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019. 5, 6
- [3] Alan H Barr. Superquadrics and angle-preserving transformations. *IEEE Computer graphics and Applications*, 1(1):11–23, 1981. 4
- [4] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020. 3
- [5] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *European Conference on Computer Vision*, pages 135–152. Springer, 2020. 1, 2
- [6] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016. 1
- [7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1, 5
- [8] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 3
- [9] Marco Crocco, Cosimo Rubino, and Alessio Del Bue. Structure from motion with objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4141–4149, 2016. 2
- [10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 2, 6
- [11] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4091–4099, 2015. 3
- [12] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012. 7
- [13] Staffan Ekvall and Danica Kragic. Robot learning from demonstration: a task-level planning approach. *International Journal of Advanced Robotic Systems*, 5(3):33, 2008. 1
- [14] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 1
- [15] Suke Harada, Tokuo Tsuji, Kazuyuki Nagata, Natsuki Yamanobe, and Hiromu Onda. Validating an object placement planner for robotic pick-and-place tasks. *Robotics and Autonomous Systems*, 62(10):1463–1477, 2014. 1
- [16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 5
- [17] Tong He, Chunhua Shen, and Anton van den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. *arXiv preprint arXiv:2011.13328*, 2020. 3
- [18] Mehdi Hosseinzadeh, Kejie Li, Yasir Latif, and Ian Reid. Real-time monocular object-model aware sparse slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7123–7129. IEEE, 2019. 2, 3, 5
- [19] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019. 3
- [20] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krahenbuhl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5390–5399, 2019. 2, 3
- [21] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017. 5
- [22] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. 1
- [23] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018. 2
- [24] Kejie Li, Hamid Rezaatofighi, and Ian Reid. Mo-ltr: Multiple object localization, tracking, and reconstruction from monocular rgb videos. *arXiv preprint arXiv:2012.05360*, 2020. 1, 2, 3, 6, 8
- [25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5
- [26] Siddharth Mahendran, Haider Ali, and Rene Vidal. A mixed classification-regression framework for 3d pose estimation from 2d images. *arXiv preprint arXiv:1805.03225*, 2018. 2
- [27] Kevis-Kokitsi Maninis, Stefan Popov, Matthias Nießner, and Vittorio Ferrari. Vid2cad: Cad model alignment using multi-view constraints from videos. *arXiv*, 2020. 1, 2, 3, 4, 5, 6, 8
- [28] Francisco Massa, Renaud Marlet, and Mathieu Aubry. Crafting a multi-task cnn for viewpoint estimation. *arXiv preprint arXiv:1609.03894*, 2016. 2
- [29] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pages 4628–4635. IEEE, 2017. 1
- [30] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object track-

- ing with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 3
- [31] Anton Milan, Stefan Roth, and Konrad Schindler. Continuous energy minimization for multitarget tracking. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):58–72, 2013. 3
- [32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 1
- [33] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. 1, 2
- [34] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1
- [35] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. Quadriclam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 4(1):1–8, 2018. 1, 2, 5
- [36] Yinyu Nie, Ji Hou, Xiaoguang Han, and Matthias Nießner. Rfd-net: Point scene understanding by semantic instance reconstruction. *arXiv preprint arXiv:2011.14744*, 2020. 2
- [37] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 4
- [38] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [39] Alex Pentland. Parts: Structured descriptions of shape. In *AAAI*, pages 695–701, 1986. 2
- [40] Alex Pentland and Stan Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Computer Architecture Letters*, 13(07):715–729, 1991. 2
- [41] Maurizio Pilu and Robert B Fisher. Equal-distance sampling of superellipse models. 1995. 5
- [42] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4404–4413, 2020. 3, 7, 8
- [43] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1
- [44] Cosimo Rubino, Marco Crocco, and Alessio Del Bue. 3d object localisation from multi-view image detections. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1281–1294, 2017. 2
- [45] Martin Runz, Kejie Li, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, et al. Frodo: From detections to 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14720–14729, 2020. 1, 2
- [46] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 5
- [47] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *arXiv preprint arXiv:2007.02442*, 2020. 1
- [48] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 3
- [49] Franc Solina and Ruzena Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE transactions on pattern analysis and machine intelligence*, 12(2):131–147, 1990. 2
- [50] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 808–816, 2016. 3
- [51] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 3
- [52] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017. 3
- [53] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9(2):137–154, 1992. 1
- [54] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015. 2
- [55] Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 1
- [56] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6499–6508, 2020. 3
- [57] Zheng Wu, Ashwin Thangali, Stan Sclaroff, and Margrit Betke. Coupling detection and data association for multiple object tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1948–1955. IEEE, 2012. 3

- [58] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from shape: Deep pose estimation for arbitrary 3d objects. *arXiv preprint arXiv:1906.05105*, 2019. [2](#)
- [59] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *arXiv preprint arXiv:1906.01140*, 2019. [3](#)
- [60] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 35(4):925–938, 2019. [1](#), [2](#), [3](#), [5](#)
- [61] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1951–1960, 2019. [3](#)