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

Kejie Li[1, 2], Daniel DeTone[2], Steven Chen[2], Minh Vo[2], Ian Reid[1], Hamid Rezatofighi[3], Chris Sweeney[2], Julian Straub[2], and Richard Newcombe[2]

[1]The University of Adelaide, [2]Facebook Reality Labs, [3]Monash University

GNN details. The object feature descriptor is mapped to a 256-dimensional embedding using the 3-layer MLP encoder, the output dimensions of which are 64, 256, and 256, before being processed by the GNN. After the encoding, every node in the graph is described by a 256-dimensional feature vector. An attention layer of the GNN takes as input node features of the last layer and outputs the updated node features by aggregating information from other nodes. Specifically, the message passing among nodes is achieved by self-attention or cross-attention depending on the connection type among the nodes (lines 281-285 in the main text).

The update of each node feature in an attention layer is proceeded as follows: (1) For each node in the graph, we employ a 4-head attention mechanism to aggregate information from other nodes; (2) The aggregated feature is then concatenated with the node feature; (3) The concatenated feature is passed to a 3-layer MLP (with dimensions of 512, 512, 256) to update the node feature. In the second part of the GNN for frame-to-model association, we we use the optimal matching layer [1] to obtain the assignment matrix. We train the GNN in a supervised fashion using the ground-truth assignments by minimizing the negative log likelihood the correct assignment:

$$L = -\sum_{(i,j)\in\mathcal{S}} log\hat{\boldsymbol{M}}_{i,j} - \sum_{i\in\mathcal{S}_0} log\hat{\boldsymbol{M}}_{i,n+1} - \sum_{j\in\mathcal{S}_1} log\hat{\boldsymbol{M}}_{m+1,j}\,, \tag{1}$$

where $\hat{\boldsymbol{M}} \in \mathbb{R}^{m+1,n+1}$ (the extra one dimension for the dustbin [1]) is the assignment prediction, $\mathcal{S}$ is the ground-truth matching pairs, and $\mathcal{S}_0$ and $\mathcal{S}_1$ are the objects or detections that are not matchable due to occlusion or out-of-frame, which should be assigned to the dustbin.

The effect of the prior term. Besides reporting the overall performance gain due to the prior term in the optimization (see Table 3 in the main text), we demonstrate the performance difference between optimization w/ prior and wo/ prior in different levels of 2D observation errors in this section. We rank the predicted 3D objects using the mean 2D IoU between the associated 2D bounding boxes and the ground-truth bounding boxes in descending order, and plot the mean 3D detection performance measured by 3D IoU at different levels of 2D observation errors. Fig. 1 shows that as the error in 2D observations increase, the performance of optimization wo/ prior drops significantly whereas the optimization w/ prior is less affected, which further validates that the prior term can increase robustness of the multi-view optimization to error in 2D observations.

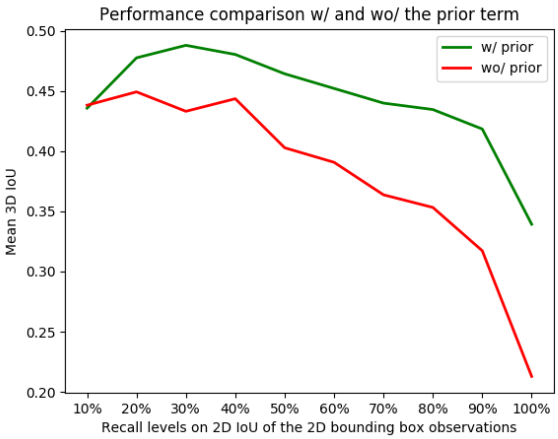

Figure 1. Comparison of 3D IoU between optimization w/ the prior term (in green) and wo/ the prior term (in red) against the 2D observation errors. Optimization with the prior term is less affected by the errors in the 2D observations.

Representation comparison.

Fig. 2 shows some examples demonstrating the limitation of cuboid or quadric representation. Although cuboid seems to be more favorable than ellipsoid as reported in Table 3 in the main text, one should note that most object classes in the Scan2CAD annotations for evaluation are box-like furniture. Ellipsoid would be advantageous for round or cylinder objects, such as cups, fruits, and balls.

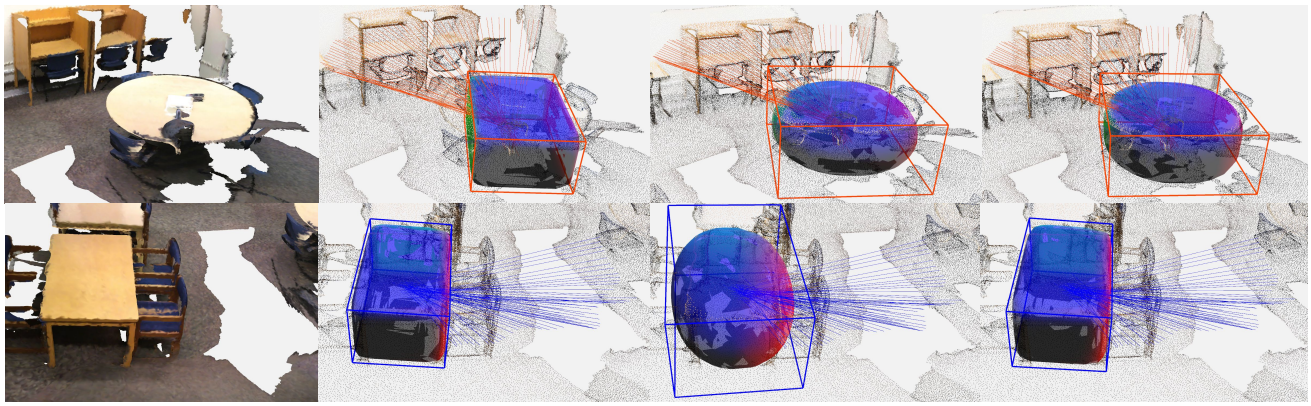

Figure 2. Visualization of cuboid, quadric, and super-quadric representation. The super-quadric representation can adapt to different object shapes while cuboid or quadric can only fit box-like and round objects well respectively.

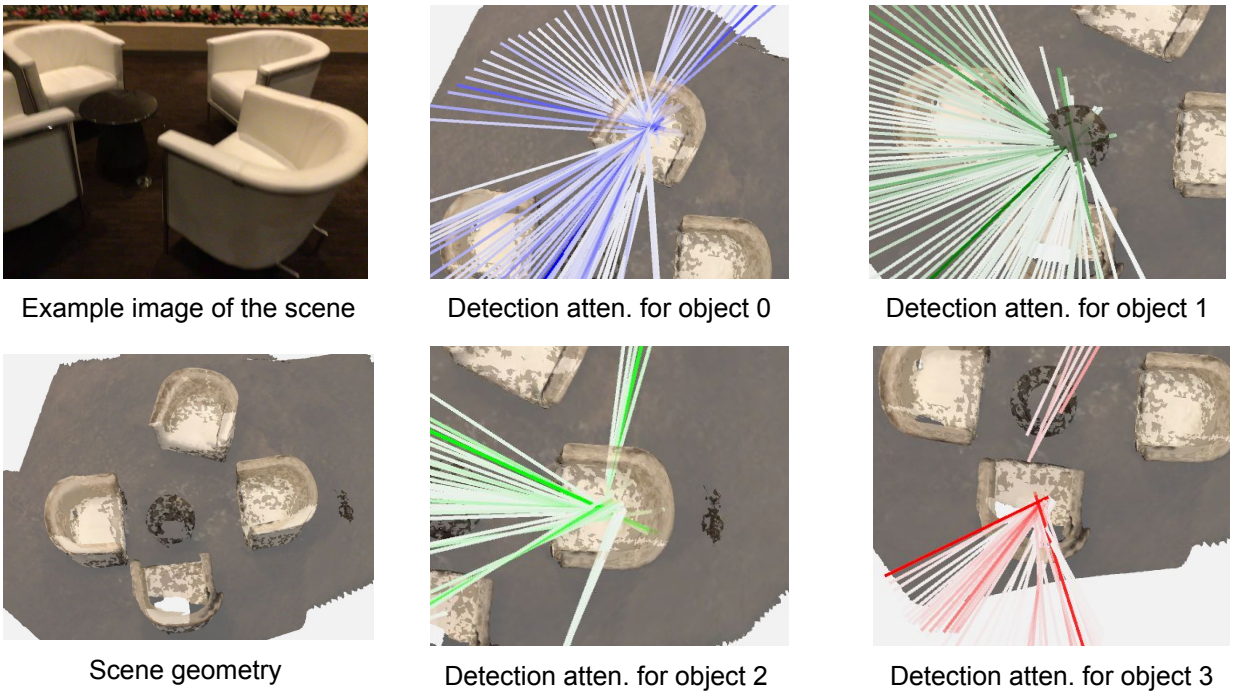

Figure 3. Visualization of object fusion attention. Each line represents a previously associated detection of an object. The attention score of a detection used for fusion is represented by the intensity of the color. The network learns to attend to detections from various viewpoints.

Super-quadric, as a unified representation for shapes including but not limited to cuboids, cylinders, and ellipsoids, is a more flexible representation for generic object shapes, as shown in Fig. 2 and Table 3 in the main text.

Self-attention visualization. Fig. 3 visualizes the attention weights of the object fusion block in the GNN (as described in the main text). Note that the network focuses on a subset of observations with a large viewpoint difference.

More qualitative results and failure case analysis. More qualitative results including failure cases are shown in the supplementary video.

References

[1] 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. 1