Supplementary Material for ELLIPSDF: Joint Object Pose and Shape Optimization with a Bi-level Ellipsoid and Signed Distance Function Description

Mo Shan, Qiaojun Feng, You-Yi Jau, Nikolay Atanasov
University of California San Diego
{moshan,qjfeng,yjau,natanasov}@ucsd.edu

1. Trained Object Models

This section provides additional visualizations for the trained object models. Training loss for the chair category is visualize in Fig. 1, which shows the loss is decreasing and stabilizes around 40,000 epochs.

Fig. 2 visualizes the rendering results for some chairs in the training set. It shows that the scale of the primitive-based representation varies proportionally with the high-resolution representation.

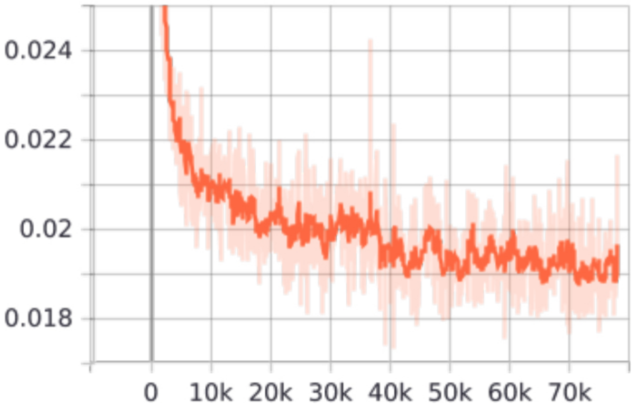

Figure 1. Visualization of the training loss for chairs.

Fig. 3 visualizes the rendering results for sofas in the training set. There is a lack of shape variation since the majority of sofas have similar structure. Nevertheless, the ellispoid for the angle sofa is still different with that of other sofas.

Fig. 4 visualizes the rendering results for tables in the training set. Similar to sofas, the variation is limited due to similar table shapes. Nonetheless, the ellipsoid for the rounded table is different from the rest.

Fig. 5 visualizes the rendering results for trashbins in the training set. It could be observed that the ellipsoid shape varies based on the object shape, for instance, the ellipsoid is enlongated for a tall trashbin.

Fig. 6 visualizes the rendering results for displays in the

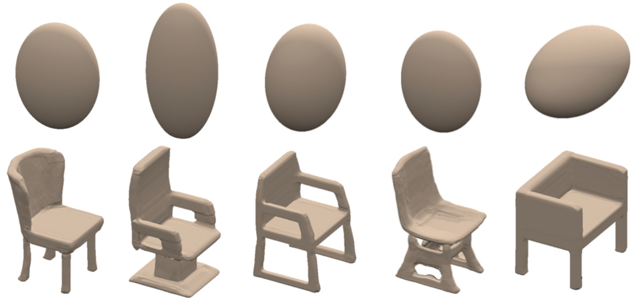

Figure 2. Visualization of the trained object model for chairs. Upper row: coarse ellipsoid shapes regressed from g_{ϕ} and $\mathbf{z}$. Lower row: SDF object model from f_{θ} and $\mathbf{z}$.

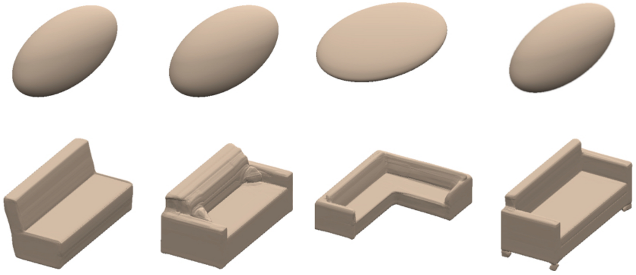

Figure 3. Visualization of the trained object model for sofas. Upper row: coarse ellipsoid shapes regressed from g_{ϕ} and $\mathbf{z}$. Lower row: SDF object model from f_{θ} and $\mathbf{z}$.

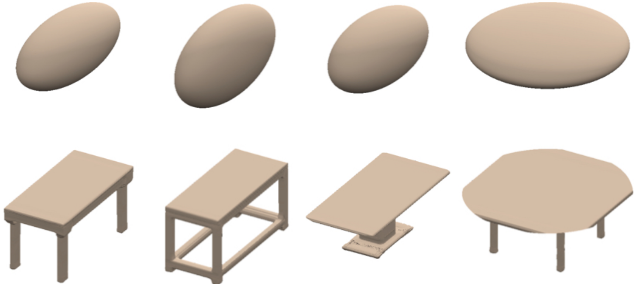

Figure 4. Visualization of the trained object model for tables. Upper row: coarse ellipsoid shapes regressed from g_{ϕ} and $\mathbf{z}$. Lower row: SDF object model from f_{θ} and $\mathbf{z}$.

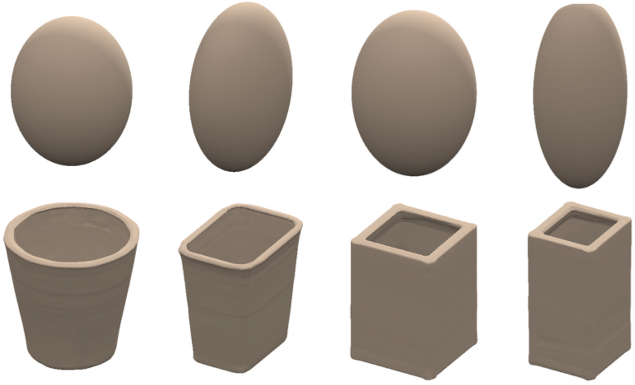

Figure 5. Visualization of the trained object model for trashbins. Upper row: coarse ellipsoid shapes regressed from g_ϕ and $\mathbf{z}$. Lower row: SDF object model from f_θ and $\mathbf{z}$.

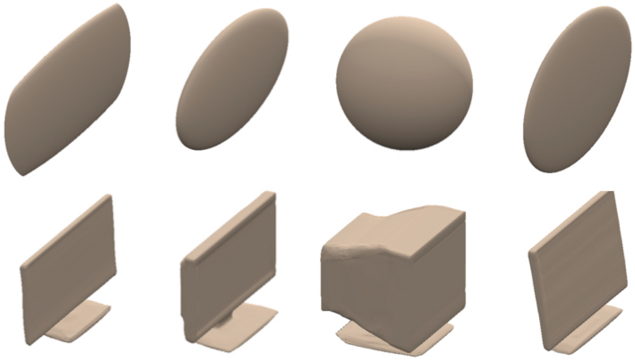

Figure 6. Visualization of the trained object model for displays. Upper row: coarse ellipsoid shapes regressed from g_ϕ and $\mathbf{z}$. Lower row: SDF object model from f_θ and $\mathbf{z}$.

training set. The ellipsoid is rounded for the thicker display and is very thin for the rest.

Fig. 7 visualizes the rendering results for cabinets in the training set. The ellipsoid varies according to the different cabinet shapes.

2. More Qualitative Results on ScanNet

This section presents more qualitative results on ScanNet [2]. Fig. 8 shows a reconstruction with table, trashbins, and cabinet. The cabinet and trashbins are reconstructed well, as can be seen from the resulting meshes which resemble the original object shapes. However, the table is poorly reconstructed, since the shape is quite different and the pose is inaccurate. This is because the available observation in the scene for the table is very limited, as can be seen in the segmented mesh, which is insufficient for optimization.

A ScanNet scene with bookshelves and tables are shown in Fig. 9, to demonstrate the usefulness of the coarse and fine level residuals. The figure illustrates that the initialized object pose and shape are different from the actual scene, since the two bookshelves in the center are not parallel and are too small compared to the observation. In contrast, the bookshelves become larger after applying the fine level residual, which is more consistent with the observations. The reconstructions are further improved with both the coarse and fine level residuals, where the bookshelves become parallel. Moreover, the bottom bookshelf and the top right table also become thinner, which agrees more with the observation. This example clearly shows the effectiveness of the proposed bi-level model for joint object pose and shape optimization.

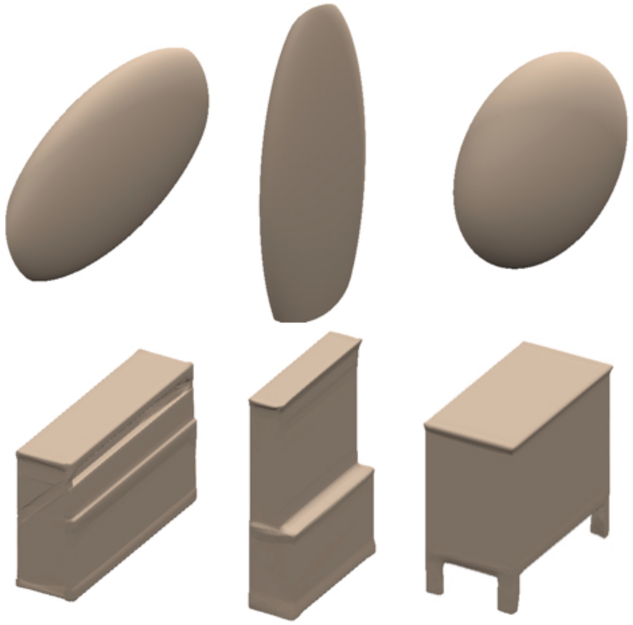

Figure 7. Visualization of the trained object model for cabinets. Upper row: coarse ellipsoid shapes regressed from g_ϕ and $\mathbf{z}$. Lower row: SDF object model from f_θ and $\mathbf{z}$.

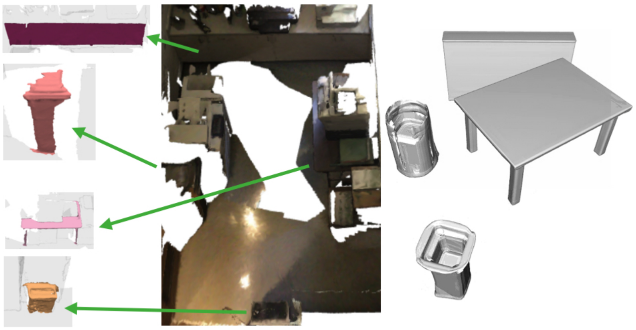

Figure 8. Visualization of the original scene and reconstructed objects for ScanNet scene 0077. The green arrows point to the segmented mesh of the objects.

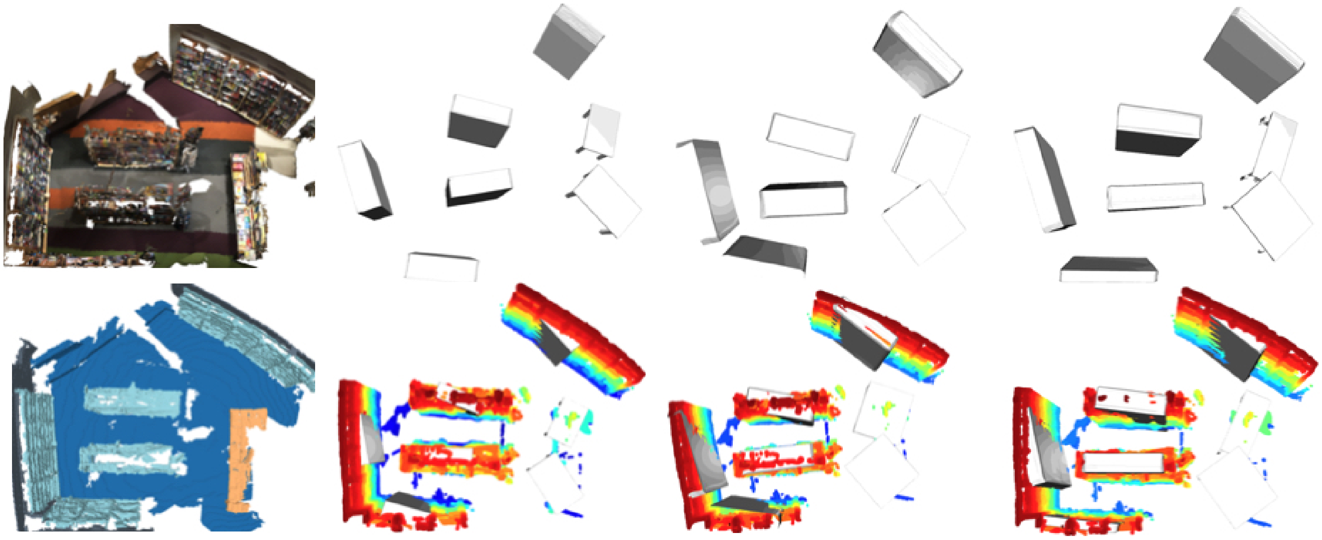

Figure 9. Visualization of the original scene and reconstructed objects for ScanNet scene 0208. First row from left to right: original scene, reconstruction using initialized pose and mean categorical object shape, reconstruction using optimized pose and shape with fine level residual only, reconstruction using optimized pose and shape with both coarse and fine level residuals. Second row from left to right: original scene with bookshelves and tables highlighted in light blue and beige, the rest are reconstructions overlaid with object point clouds and added pseudo points.

3. Pose Estimation Metric

This section presents the metric used to evaluate the object pose, which follows Scan2CAD [1]. We introduce the details on how to decompose a pose $\mathbf{T} \in \mathrm{SIM}(3)$ into rotation $\mathbf{q}$, translation $\mathbf{p}$ and scale $\mathbf{s}$ and the error functions for each element separately. For rotation and scale, $\mathbf{R}_s = \mathbf{P}\mathbf{T}\mathbf{P}^\top$:

$$s_1 = \|\mathbf{R}_s\mathbf{e}_1\|_2 \quad s_2 = \|\mathbf{R}_s\mathbf{e}_2\|_2 \quad s_3 = \|\mathbf{R}_s\mathbf{e}_3\|_2, \\ \mathbf{R}\mathbf{e}_1 = \frac{\mathbf{R}_s\mathbf{e}_1}{s_1} \quad \mathbf{R}\mathbf{e}_2 = \frac{\mathbf{R}_s\mathbf{e}_2}{s_2} \quad \mathbf{R}\mathbf{e}_3 = \frac{\mathbf{R}_s\mathbf{e}_3}{s_3}. \tag{1}$$

Suppose $\boldsymbol{R} = \{m_{ij}\}, i, j \in [1, 2, 3]$, we transform it to quaternion $\mathbf{q}$ by

$$q_0 = \frac{\sqrt{\mathrm{tr}(R)+1}}{2}, q_1 = \frac{m_{23}-m_{32}}{4q_0}, q_2 = \frac{m_{31}-m_{13}}{4q_0}, q_3 = \frac{m_{12}-m_{21}}{4q_0}. \tag{2}$$

Suppose the prediction and groundtruth are $\mathbf{q}_{pred}, \mathbf{q}_{gt}$, we compute the difference by

$$e_{\mathrm{SO}(3)}(\mathbf{q}, \hat{\mathbf{q}}) := 2\arccos(|\mathbf{q}_{gt}^\top \mathbf{q}_{pred}|). \tag{3}$$

Translation is $\mathbf{p} = \mathbf{T}[1:3, 4]$, and we compare the difference between prediction and groundtruth by

$$\|\mathbf{p}_{pred} - \mathbf{p}_{gt}\|_2. \tag{4}$$

For scale percentage error, we compute it by

$$100 \times |\frac{1}{3}\sum_{i=1}^{3} \bar{s}_i - 1|, \tag{5}$$

where $\bar{s}_i = \frac{s_{pred}}{s_{gt}}$ for each of s_1, s_2, s_3 recovered from the SIM(3) matrix.

4. Timing

Table 1. ELLIPSDF timing breakdown (sec)

Init	Latent Code Opt	SIM(3) Opt	SDF Decoding	Meshing
0.04	0.13	0.58	1.38	2.34

Timing for one instance is provided in Table 1. *Init* is the pose initialization in (14) for 100 views. *Latent Code Opt* and *SIM(3) Opt* are a single SGD step with respect to $\delta\mathbf{z}$ and $\mathbf{T}$ respectively using 10000 points as batch size. *SDF Decoding* and *Meshing* are optional steps that generate SDF predictions over 256^3 points and apply Marching Cubes to generate a mesh. Our approach does not currently operate in real-time but it is more efficient than existing work. We will investigate how to accelerate the current slow python SIM(3) optimization.

References

[1] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019. 3

[2] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. 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