Iterative Superquadric Recomposition of 3D Objects from Multiple Views
-
Supplementary Material

In this supplementary material, we provide a derivation on how we initialize superquadrics (Sec. A), additional implementation details (Sec. B), analyses on the effect of the hyperparameter λ (Sec. C) on our loss function (Eq. 6), and show how the camera viewpoints impact the performance of ISCO (Sec. D). We will then report quantitative results using Chamfer distance as metric (Sec. E), a comparison with trained abstraction methods (Sec. F), and additional qualitative results (Sec. G) on ShapeNet.

A. Superquadric initialization

To instantiate new superquadrics around object parts that have not been covered yet, we estimate the origin of rendering errors in the 3D space. We use a dense voxel grid $G \in \mathbb{R}^{N\times N\times N}$ with resolution N around the object to which we propagate rendering errors.

First, we evaluate the superquadrics density at each point $\mathbf{g}$ in the voxel grid

$$V_{\mathbf{g}} = \sigma(\mathbf{g};\theta) \tag{1}$$

where V is the the complete grid of density values while $V_{\mathbf{g}}$ is the density value at $\mathbf{g}$. To simplify notation we omit θ from V and any further definitions. We then render this volume grid through ray marching and calculate the loss $\mathcal{L}_{k-1}^{\lambda=0}$. To do so, the density of each point along the camera ray $\mathbf{r}(t)$ is obtained by applying a sampling kernel locally

$$K_{\mathbf{r}(t)} = \sum_{\mathbf{g}\in G} V_{\mathbf{g}} k(\mathbf{r}(t) - \mathbf{g};\Phi) \tag{2}$$

where Φ are the parameters of a generic sampling kernel $k(\cdot;\Phi)$ and we evaluate all ray points $\mathbf{r}(t)$ from the stratified sampling approach to render the reconstructed image. In practice, we use trilinear interpolation for the kernel k which results in

$$K_{\mathbf{r}(t)} = \sum_{\mathbf{g}\in G} V_{\mathbf{g}} \prod_{i=1}^{3} \max(0, 1 - \frac{|x_i^{(\mathbf{r}(t))} - x_i^{(\mathbf{g})}|}{l}) \tag{3}$$

where $[x_1^{(\mathbf{p})}, x_2^{(\mathbf{p})}, x_3^{(\mathbf{p})}]$ are the three-dimensional coordinates of point $\mathbf{p}$ and l is the uniform distance between voxels in the grid.

Using the resampled density values from $K_{\mathbf{r}(t)}$ to replace the true densities of the superquadrics, we can now render the reconstructed image with

$$D(\mathbf{r}) = \int_{t_n}^{t_f} T(t) K_{\mathbf{r}(t)} dt \tag{4}$$

and calculate the loss $\mathcal{L}_{k-1}^{\lambda=0}$ the same way as described in the paper for direct rendering of the superquadrics.

Finally, to propagate the rendering errors, we calculate the gradient of the loss with respect to the densities $V_{\mathbf{g}}$ of the grid points

$$\frac{\partial \mathcal{L}_{k-1}^{0}}{\partial V_{\mathbf{g}}} = \sum_{\mathbf{r}\in\mathcal{R}} \sum_{\mathbf{r}(t)\in\mathbf{r}} \frac{\partial \mathcal{L}_{k-1}^{0}}{\partial K_{\mathbf{r}(t)}} \prod_{i=1}^{3} \max(0, 1 - \frac{|x_i^{(\mathbf{r}(t))} - x_i^{(\mathbf{g})}|}{l}) \tag{5}$$

where $\frac{\partial \mathcal{L}_{k-1}^{0}}{\partial K_{\mathbf{r}(t)}}$ is calculated as is common in ray marching and the remaining part of the term is a result of the trilinear sampling that distributes the gradient to each point $\mathbf{g}$ (using $V_{\mathbf{g}}$ as a representative value for its density contribution) in the voxel grid G.

Intuitively, each point $\mathbf{g}$ is a candidate location for initializing a new superquadric. Hence, we estimate how accurate the density at each location $\mathbf{g}$ is. We use $\lambda = 0$ in $\mathcal{L}_{k-1}^{\lambda=0}$ to only consider errors where part of the object is not covered by a superquadric yet. By propagating the error to each $\mathbf{g}$, the point with the highest error corresponds to the location where we have the best potential to improve our reconstruction loss if we initialize a new superquadric there.

We choose the voxel grid resolution $N = 64$ such that there are $64^3 \approx 262$k candidate locations for new superquadrics in each ISCO iteration. The voxel grid is placed in the center of the scene (where we expect the object to be) and the distance l between voxels is chosen such that the object is completely enclosed by the voxel grid. Since the exact position and size of the object are unknown, we choose l based on the distance between the cameras and the center of the scene.

B. Implementation details

Learning rate. We found that fitting the superquadrics to the ground-truth 2D views, can be sensitive to the learning rate during optimization. Specifically, when the learning rate is too small, significantly more update steps are required to fit the superquadric parameters well, before introducing the next superquadric. We found that a learning rate of 0.01 helps in optimizing the first superquadrics reliably within our chosen 250 update steps per iteration. However, as smaller regions of the shape are being reconstructed later on, more fine-grained optimization steps are necessary, and thus we gradually reduce the learning rate to 0.001. Hence, for all our experiments, we use the Adam [1] optimizer and a cosine learning rate schedule that starts with a learning rate of 0.01 and is annealed to 0.001 by the end of the optimization. All the hyperparameters have been selected using a small subset of object instances in ShapeNet training set and kept constant across all the experiments.

Ray sampling. Due to computational reasons, it is not common to render the whole image from all camera angles during training, but instead subsample rays, i.e. pixels, for every viewpoint [3]. Since our loss is defined on object silhouettes, a large amount of pixels will fall outside both the target object and the superquadrics rendering, not contributing to the loss. Hence, we employ an importance sampling strategy, by sampling rays that contributed to the loss in previous update steps with higher probability. This improves sampling efficiency and allows us to use 500 rays per view point rather than all and 250 update steps per superquadric.

C. Choice of λ

The hyperparameter λ in our loss $\mathcal{L}(\lambda)$ regulates local vs. global loss terms. For instance, when fitting a single superquadric to the object, $0 < \lambda < 0.5$ encourages the superquadric to fully enclose the object shape and promoting global fitting, while $\lambda > 0.5$ produces tighter boundaries, promoting local fitting. As Table 1 shows, performance rapidly decrease if λ is either too low (*e.g.* $\lambda = 0.4$) or too high (*e.g.* $\lambda = 0.8$) since either the model cover larger regions (even beyond the target object) or it focuses on too fine-grained details, even ignoring other object parts.

Quantitatively, we observed that a larger λ (*i.e.* higher focus on locality) lead to a better locality and accuracy in decomposing objects into meaningful parts, with $\lambda = 0.6$ leading to the best overall performance by $\sim 1\%$ IoU on ShapeNet, cf. Table 1. Based on these observations, we choose $\lambda = 0.6$ throughout the experiments in this work.

D. Dependence on camera viewpoints

In the main paper, camera views are sampled randomly around the object. If, in practice, this is not feasible, information about the object from certain angles might be missing. We investigate how such a constraint could limit the reconstruction accuracy on the ShapeNet dataset where we can control the choice of camera views best. If we constrain the 16 random views to be inside a spherical cap with polar angle 90°(top hemisphere), 45°and 22.5°(where 0°is the top-view), the reconstruction mIoU decreases from 0.656 to 0.634, 0.592 and 0.528, respectively. Despite the limited drop, these results show that ISCO depends on the choice of camera views.

# views	λ 0.4	0.5	0.6	0.7	0.8
4	0.547	0.560	**0.569**	0.551	0.541
8	0.606	0.629	**0.630**	0.615	0.593
16	0.629	0.643	**0.650**	0.631	0.613

Table 1. **Volumetric IoU on ShapeNet** of ISCO for different values of λ and different number of views. We measure the results as mean Intersection over Union (IoU) of the object volume on ShapeNet (higher is better).

Input	Point Cloud		2D Views
Method	EMS [2]	NBP [6]	ISCO
airplane	0.1201	0.0590	**0.0508**
bench	0.2105	0.0725	**0.0611**
cabinet	0.1327	**0.0990**	0.1097
car	0.0743	0.0730	**0.0530**
chair	0.2225	0.1303	**0.1259**
display	0.1353	0.1009	**0.0978**
lamp	0.2012	0.1338	**0.1259**
speaker	0.2098	0.1775	**0.1428**
rifle	0.1050	0.0577	**0.0191**
sofa	0.1684	0.0952	**0.0807**
table	0.2467	0.1649	**0.1418**
phone	0.0634	0.0406	**0.0350**
vessel	0.0987	0.0468	**0.0393**
mean	0.1530	0.0962	**0.0833**

Table 2. **Chamfer-L1 Distance on ShapeNet.** We report Chamfer-L1 distance on ShapeNet (lower is better). EMS and NBP use point cloud as input, ours uses 16 random views.

E. Chamfer-L1 evaluation on ShapeNet

In addition to the IoU reconstruction results in the main paper, here we report the Chamfer-L1 distance between the reconstructed superquadrics shapes of EMS [2], NBP [6] and ISCO to the ground-truth shapes in Table 2. We sample 100k points on the surface of both the ground truth and predicted shape to calculate the Chamfer-L1 distance. Note that in Table 2 ISCO uses multi-view inputs, while EMS and NBP points clouds are extracted from the target 3D object.

In all but one category, the superquadric reconstructions of ISCO are closer to the original shapes than both EMS and NBP. This result follows the same trend as the IoU (Ta-

ble 1 of the main paper), providing further evidence that ISCO can better recompose objects with superquadrics than its competitors, despite relying on cheaper multiple-view inputs rather than ground-truth 3D representations.

F. Comparison with unsupervised abstraction methods using a training set

Shape abstraction has been studied by performing single-view reconstruction with a neural network. These methods typically use a single view as input and dense 3D point clouds at targets. By performing training on a large dataset, these networks learn shape priors such that they can reconstruct an object with simple shapes from an image. Related works include SQ [5] and HSQ [4] which also use superquadrics as primitive shapes. However, since ISCO takes a different instance-based perspective without requiring a training set, it is difficult to make a comparison on the same ground. For instance, SQ and HSQ, both single-view but with a training set, achieve 0.277 and 0.580 mIoU on ShapeNet, respectively, while ours 0.656 mIoU, not trained but with multi-views. We tried to perform a fair assessment, training HSQ with 16 views as input on all ShapeNet classes and also evaluating it with the same 16 views as ISCO. In this case, the mIoU of HSQ improves slightly by 0.029, with ISCO still outperforming it. Note that, unlike our instance-based method, SQ and HSQ suffer from distribution shifts as they cannot generalize beyond the training set.

G. Additional qualitative examples

In Figure 1, we show additional qualitative results of the classes from ShapeNet where we randomly sample instances from the test set. We use the same hyper-parameters described in the main paper, using 10 superquadrics to represent the instances. From the results, we can see some failure cases of our model. For instance, thin parts may not be well decomposed, especially if they are hard to model with 2D views only, such as the body/tube of *lamps* (4th row) or legs of *benches* (2nd row).

At the same time, the figure also shows the ability of ISCO to capture the structure of the underlying object. For instance, objects with distinct parts consistently have these decomposed by ISCO such as the wings and engines of *airplanes* (top row), body of *vessels* (last row), and legs, seats, and back rests of *chairs*. For simpler shapes (e.g. *phones*, 2nd from bottom), superquadrics are used in later stages to fill inaccuracies in corners of the object and smaller details. Note that if higher abstraction is desired, one could reduce the number of superquadrics, or prune small superquadrics post-hoc.

References

[1] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

[2] Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. Robust and accurate superquadric recovery: a probabilistic approach. In *CVPR*, 2022. 2

[3] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[4] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *CVPR*, 2020. 3

[5] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *CVPR*, 2019. 3

[6] Yuwei Wu, Weixiao Liu, Sipu Ruan, and Gregory S Chirikjian. Primitive-based shape abstraction via nonparametric bayesian inference. In *ECCV*, 2022. 2

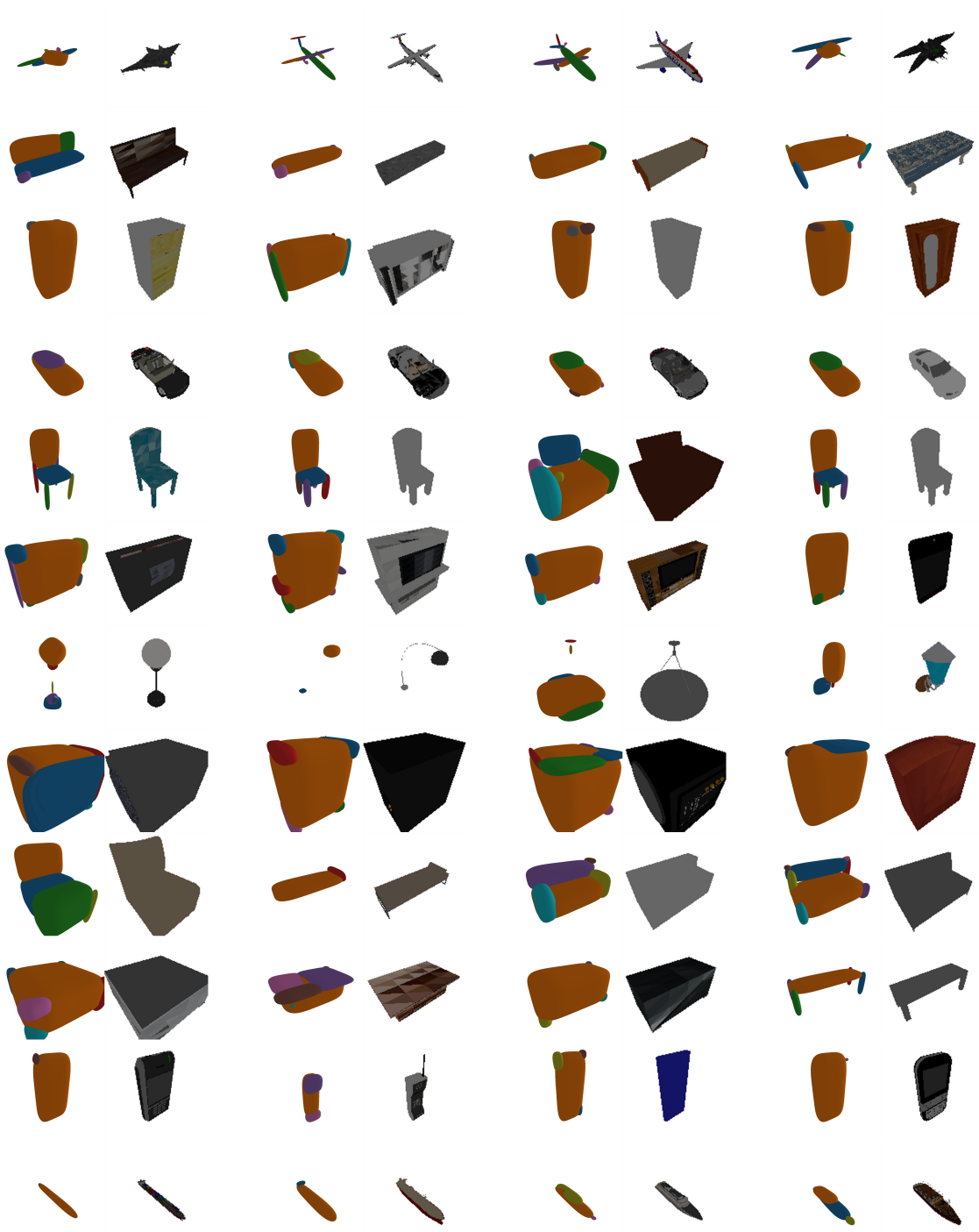

Figure 1. **Additional qualitative results for ShapeNet classes.** We show additional qualitative examples for *random* instances of ShapeNet classes. Each row is a different class, from top to bottom: *airplane*, *bench*, *cabinet* , *car*, *chair*, *display*, *lamp*, *speaker*, *sofa*, *table*, *phone*, *vessel*. On each column, the left parts shows the result of our model while the right part the ground-truth shape.