Supplementary Material: Primitive-based Shape Abstraction via Nonparametric Bayesian Inference

Yuwei Wu[1], Weixiao Liu[1,2], Sipu Ruan[1], and Gregory S. Chirikjian[1*]

[1] National University of Singapore
[2] Johns Hopkins University
{yw.wu, mpewxl, ruansp, mpegre}@nus.edu.sg

Abstract. In this supplementary material, we first discuss more details about the evaluation metrics we select in the experiments. Furthermore, we discuss details of the merging and splitting process. Finally, we provide additional qualitative results on the ShapeNet dataset.

1 Metrics

In this section, we provide more details about the two metrics that we use to evaluate the experiments.

1.1 Chamfer L_1-distance

Recall that Chamfer L_1-distance is defined as follows:

$$D_{\text{chamfer}}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{N} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}} \min_{\boldsymbol{y}_j \in \boldsymbol{Y}} \left\| \boldsymbol{x}_i - \boldsymbol{y}_j \right\|_1 + \frac{1}{M} \sum_{\boldsymbol{y}_j \in \boldsymbol{Y}} \min_{\boldsymbol{x}_i \in \boldsymbol{X}} \left\| \boldsymbol{y}_j - \boldsymbol{x}_i \right\|_1, \quad (1)$$

where $\boldsymbol{X} = \{\boldsymbol{x_i}\}$ denotes the points sampled from the original model, $\boldsymbol{Y} = \{\boldsymbol{y_j}\}$ denotes the points sampled from the predicted model, and N and M is the number of points of the sets $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. For $\boldsymbol{X}$, since the ShapeNet dataset does not provide point cloud representations for the objects, we need to sample points from the original mesh files. Specifically, for each original model, we sample points densely on each face of the mesh. Then, we downsample the point cloud to be around $50 - 60K$ points. D-FAUST provides the point cloud for each data. We directly downsample the original point cloud to be around $50-60K$ points. For $\boldsymbol{Y}$, we apply the equal-distance sampling strategy [3] on each superquadric surface of the predicted model to get a collection of sets of points. By taking union of all the point sets, we obtain a point cloud representation for the predicted model. Then we downsample the point cloud to be around $50 - 60K$ points, as well. The first term of Eq. 1 computes how far on average the closest point of the predicted model is to the original model, and the second term calculates how far on average the closest point of the original model is to the predicted model. Thus, a lower value of Chamfer distance implies a better abstraction accuracy in terms of surface fitness.

1.2 Intersection over Union (IoU)

Recall the definition of IoU as follows:

$$\text{IoU} = \frac{V\left(S_{\text{pred}} \cap S_{\text{target}}\right)}{V\left(S_{\text{pred}} \cup S_{\text{target}}\right)}, \tag{2}$$

where S_{pred} is the predicted model, S_{target} is the original model, and $V(.)$ computes the volume. It is difficult, if not impossible, to obtain the volume of the intersection or union of two models. Therefore, we approximate the volume with the Monte Carlo method. We sample 50K points uniformly inside the axis-aligned bounding box of the original model. For each point $\mathbf{x}$, we check its position relative to the original mesh model and the predicted multi-superquadric model, respectively. We approximate the IoU with the ratio of the points belonging to the set of the intersection and the union. If two models match perfectly, the IoU will be 1 and if two models disjoint from each other, the IoU is 0. In other words, a higher IoU indicates a better abstraction accuracy in terms of volumetric occupation.

2 Merging & Splitting Process

2.1 Merging Process

We design a merging process to reduce the number of primitives while maintaining geometric accuracy, leading to a more interpretable and concise abstraction. Given an abstraction result with $\tilde{\mathcal{X}} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_K\}$ and $\tilde{\boldsymbol{\theta}} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K\}$, where $\mathcal{X}_j$ is a cluster of points and $\boldsymbol{\theta}_j$ is the corresponding superquadric representation, we propose a method to determine if any two clusters can be merged. Specifically, for $\mathcal{X}_i$ and $\mathcal{X}_j$ $(i \neq j)$, we let

$$\begin{aligned} M_i &= \frac{1}{n_i} \sum_l d^2(\boldsymbol{\theta}_i, \boldsymbol{u}_l) \\ M_j &= \frac{1}{n_j} \sum_l d^2(\boldsymbol{\theta}_j, \boldsymbol{v}_l), \end{aligned} \tag{3}$$

where $\boldsymbol{u}_l \in \mathcal{X}_i$, $\boldsymbol{v}_l \in \mathcal{X}_j$, $d(.)$ computes the radial distance between a point and a superquadric [2], and n_i and n_j is the number of points in $\mathcal{X}_i$ and $\mathcal{X}_j$, respectively. Eq. 3 measures how good the superquadric representation for a cluster is. Subsequently, we fit a new superquadric $\boldsymbol{\theta}_{new}$ for the cluster $\mathcal{X}_{new} = \mathcal{X}_i \cup \mathcal{X}_j$ [3], and let

$$M_{new} = \frac{1}{n_{new}} \sum_l d^2(\boldsymbol{\theta}_{new}, \boldsymbol{w}_l), \tag{4}$$

where $\boldsymbol{w}_l \in \mathcal{X}_{new}$ and n_{new} is the number of $\mathcal{X}_{new}$. If $M_{new} \leq max(M_i, M_j)$ or $M_{new} \leq C_{thre}$ (C_{thre} is a constant), we call these two clusters is a good merge. Furthermore, as mentioned in [1], we do not want to see that the volume

of $\boldsymbol{\theta}_{new}$ is much larger than the volume of $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ as this usually means we wrongly merge two separate parts of an object into one and represent it with a large superquadric. The detailed process of the merging is summarized in Algorithm 1. We incorporate this merging process at the end of the iteration of the optimization-based Gibbs sampling when $t = \frac{T}{2}$ and $t = T$.

Algorithm 1 Merging Process

Input: $\tilde{\mathcal{X}} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_K\}$, $\tilde{\boldsymbol{\theta}} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K\}, \tilde{\sigma}^2 = \{\sigma_1^2, \sigma_2^2, ..., \sigma_K^2\}, C_{thre}$
Output: $\tilde{\mathcal{X}}_{new} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_S\}$, $\tilde{\boldsymbol{\theta}}_{new} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_S\}, \tilde{\sigma}^2_{new} = \{\sigma_1^2, \sigma_2^2, ..., \sigma_S^2\}$
$merged \leftarrow \{\}$
for $i = 1, 2, ..., K-1$ **do**
  for $j = i+1, i+2, ..., K$ **do**
    if $j \in merged$ **then**
      ▷ If jth cluster has been merged, we will skip this cluster
      Continue
    end if
    if $V(OBB(\mathcal{X}_i)) + V(OBB(\mathcal{X}_j)) \leq 0.7 \times V(OBB(\mathcal{X}_{new}))$ **then**
      ▷ OBB is the oriented bounding box of the point set, $V(.)$ computes the volume
      Continue
    end if
    if $goodMerge(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j, C_{thre})$ & $V(\boldsymbol{\theta}_i) + V(\boldsymbol{\theta}_j) > 0.7 \times V(\boldsymbol{\theta}_{new})$ **then**
      $\mathcal{X}_i \leftarrow \mathcal{X}_{new}$
      $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_{new}$
      $\sigma_i^2 \leftarrow \sigma_{new}^2, \sigma_{new}^2 = \frac{1}{\gamma}, \gamma \sim \boldsymbol{\Gamma}\left(\frac{n_{new}-1}{2}, \frac{2}{M_{new} \cdot n_{new}}\right)$
      ▷ We draw a new sample σ_{new}^2 for the merged cluster
      $merged.Append(j)$
    end if
  end for
end for
$\tilde{\mathcal{X}}_{new} \leftarrow \tilde{\mathcal{X}}.Remove(\mathcal{X}_l)$
$\tilde{\boldsymbol{\theta}}_{new} \leftarrow \tilde{\boldsymbol{\theta}}.Remove(\boldsymbol{\theta}_l)$
$\tilde{\sigma}^2_{new} \leftarrow \tilde{\sigma}^2.Remove(\sigma_l^2), l \in merged$
▷ We remove all the clusters that have been merged into other clusters

2.2 Splitting Process

Recall the way we sample membership for each point $\boldsymbol{x}_i$:

$$p\left(z_i = j \mid \boldsymbol{Z}_{-i}, \boldsymbol{\theta}_j, \sigma_j^2, \boldsymbol{X}, \alpha\right) \propto \frac{n_{-i,j}}{N-1+\alpha} \frac{1}{2\sqrt{2\pi}\sigma_j} \exp\left(-\frac{d^2(\boldsymbol{\theta}_j, \boldsymbol{x}_i)}{2\sigma_j^2}\right), \quad (5)$$

and

$$p\left(z_i = K+1 \mid \boldsymbol{Z}_{-i}, \boldsymbol{\theta}_j, \sigma_j^2, \boldsymbol{X}, \alpha\right) \propto \frac{\alpha}{N-1+\alpha} p_0, \quad (6)$$

where α is the concentration factor of DP, $\boldsymbol{Z}_{-i}$ denotes $\boldsymbol{Z}$ excluding z_i, and $n_{-i,j}$ is the number of points belonging to cluster j, excluding $\boldsymbol{x}_i$. Eq. 5 shows that

the larger an existing cluster is (larger $n_{-i,j}$), the more attractive it will be for a sample point $\boldsymbol{x}_i$. This might sometimes incur unintended results. For example, some points are assigned wrongly to cluster $\mathcal{X}_k$ by accident and thus increase the size of $\mathcal{X}_k$. Then, at the next iteration the $\mathcal{X}_k$ becomes more attractive to sample points and might take in more unwanted points, leading to a vicious cycle. Thus, at each iteration before sampling $\boldsymbol{Z}$, we calculate how good each superquadric representation $\boldsymbol{\theta}_i$ for each cluster $\mathcal{X}_i$ is as in Eq. 3. If M_i is too high, namely $\boldsymbol{\theta}_i$ is a bad representation, we will segment cluster $\mathcal{X}_i$ into subclusters $\{\mathcal{Y}_{i1}, \mathcal{Y}_{i2}, ..., \mathcal{Y}_{ig}\}$ based on Euclidean distance [4]. We denote the largest subcluster, which contains most points, as $\mathcal{Y}_{iL}$ and fit a new superquadric $\boldsymbol{\theta}_{iL}$ to it. Afterwards, we replace $\mathcal{X}_i$ with $\mathcal{Y}_{iL}$ and replace $\boldsymbol{\theta}_i$ with $\boldsymbol{\theta}_{iL}$. By doing so, we prevent the cluster from growing in an unintended way.

3 Additional Results

In this section, we provide additional abstraction results on the ShapeNet dataset. As illustrated in Fig. 1, 2, 3, and 4, our method is able to abstract the semantic parts from objects of different kinds accurately. Taking the table as an example, from the superquadric-based abstractions, we can identify which parts are legs and which part is the desktop. In addition, we can also learn about the relative sizes and poses between different parts. We believe such high-level abstractions will allow the machine to understand the objects better and interact with them with higher intelligence.

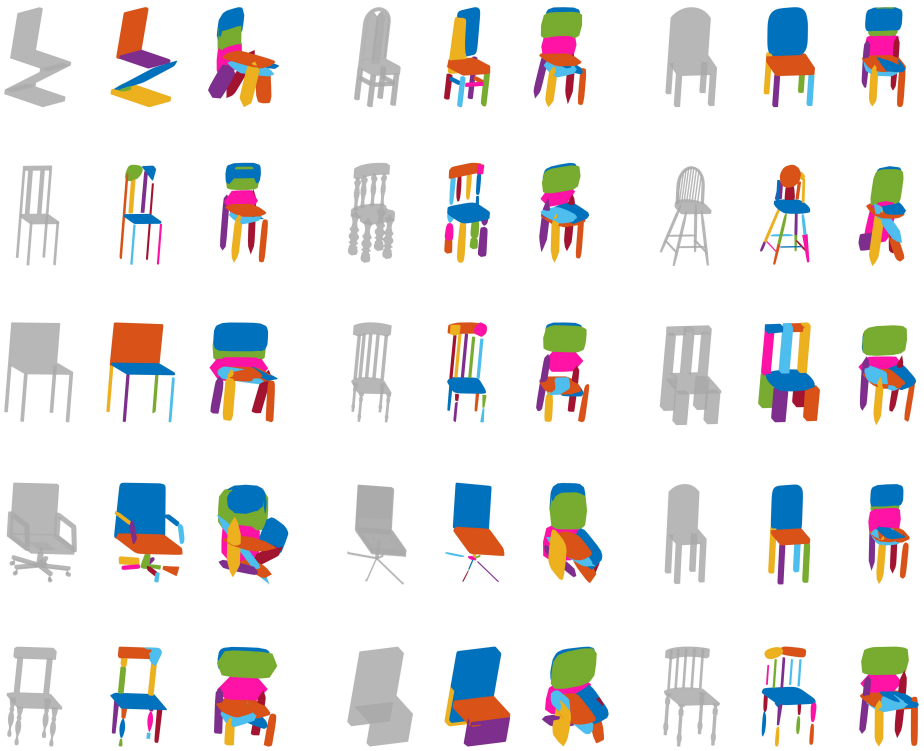

Fig. 1. Qualitative results of 3D abstraction on chairs. The left ones are the original meshes, the middle ones are our inferred results, and the right ones are inferred from SQs [5].

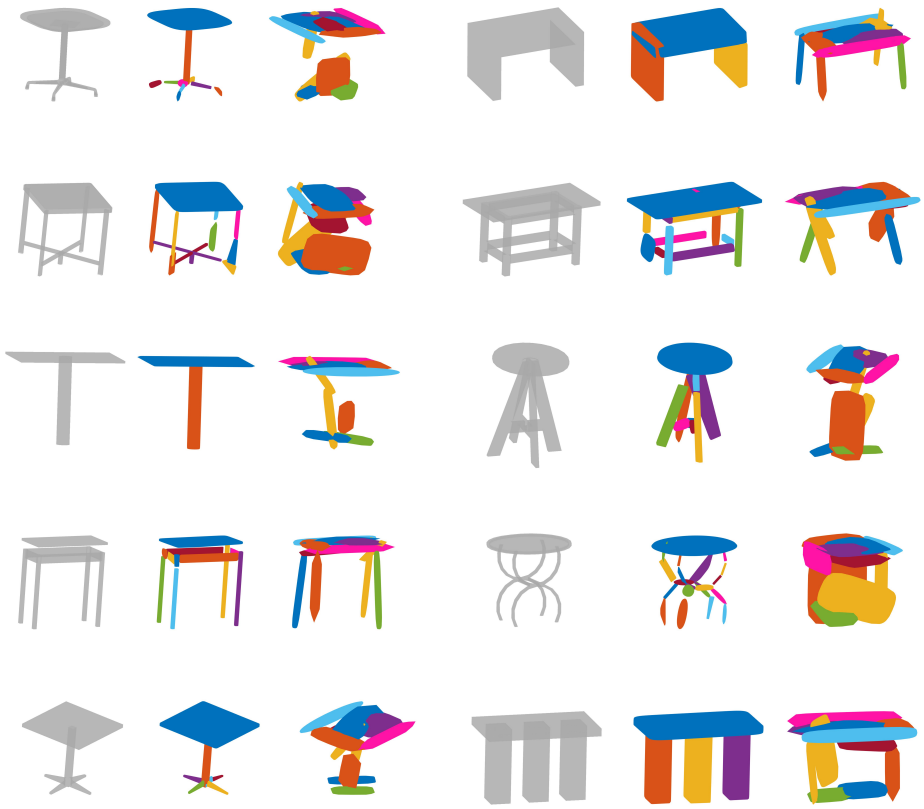

Fig. 2. Qualitative results of 3D abstraction on tables. The left ones are the original meshes, the middle ones are our inferred results, and the right ones are inferred from SQs [5].

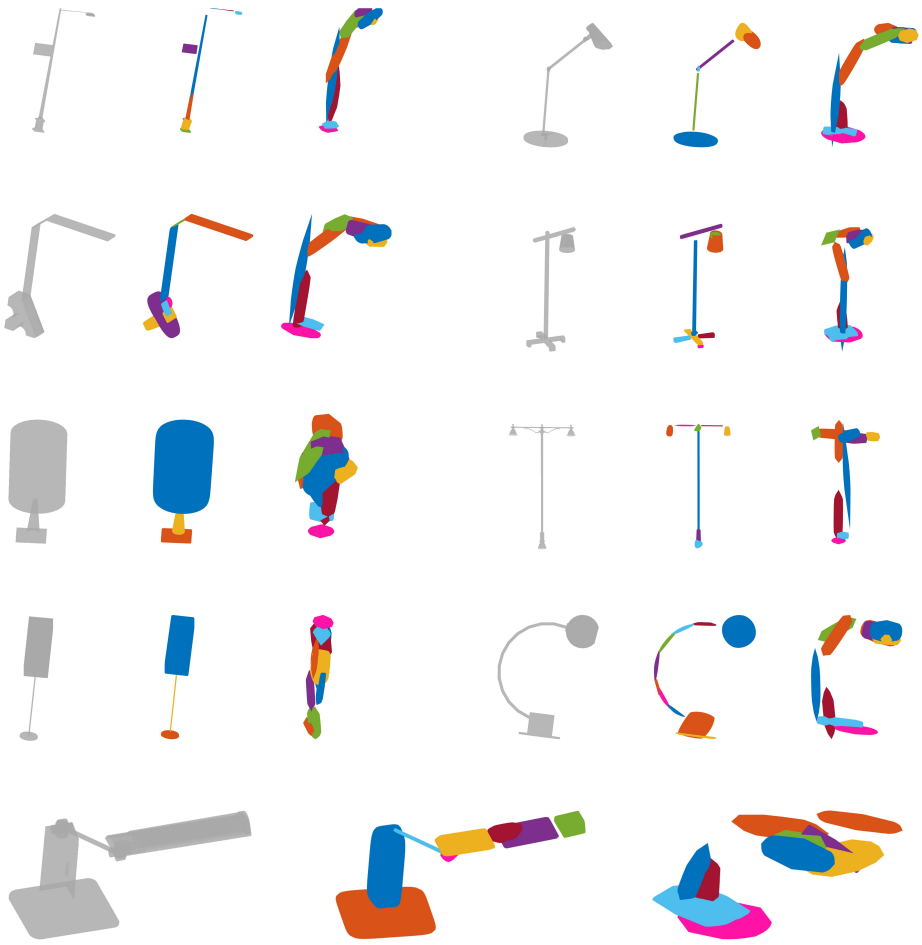

Fig. 3. Qualitative results of 3D abstraction on lamps. The left ones are the original meshes, the middle ones are our inferred results, and the right ones are inferred from SQs [5].

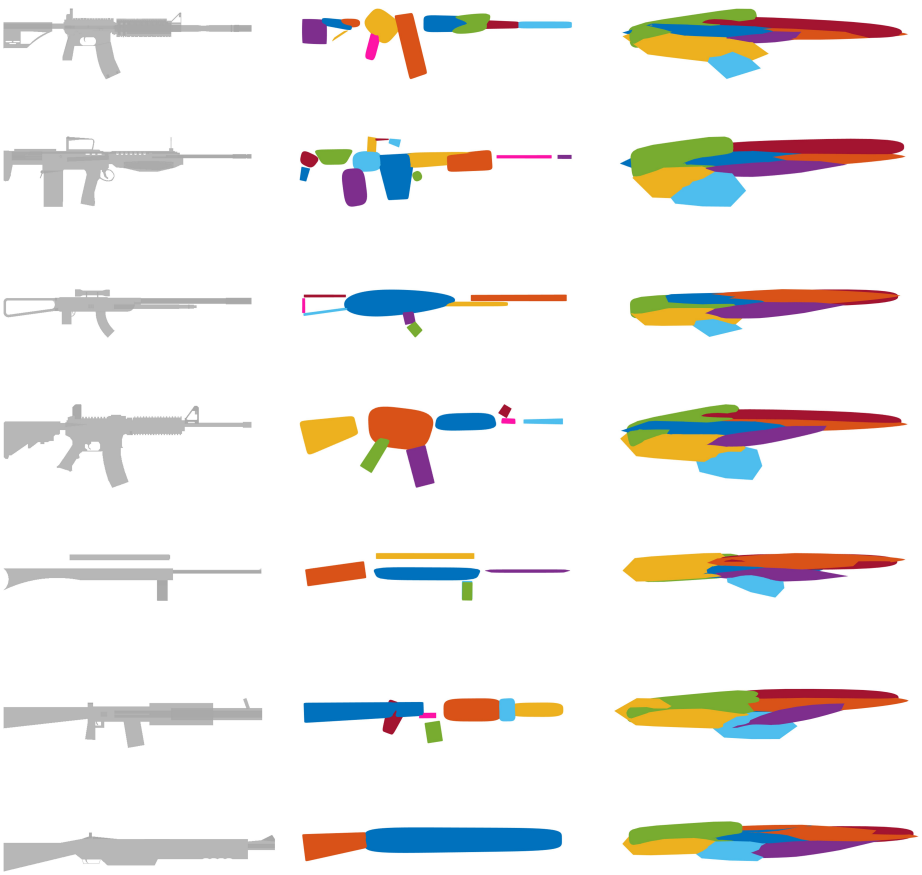

Fig. 4. Qualitative results of 3D abstraction on rifles. The left ones are the original meshes, the middle ones are our inferred results, and the right ones are inferred from SQs [5].

References

1. Chevalier, L., Jaillet, F., Baskurt, A.: Segmentation and superquadric modeling of 3d objects. In: WSCG (2003)
2. Gross, A.D., Boult, T.E.: Error of fit measures for recovering parametric solids. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (1988)
3. Liu, W., Wu, Y., Ruan, S., Chirikjian, G.S.: Robust and accurate superquadric recovery: a probabilistic approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2676–2685 (2022)
4. Matlab optimization toolbox (2021a), the MathWorks, Natick, MA, USA
5. Paschalidou, D., Ulusoy, A.O., Geiger, A.: Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10344–10353 (2019)