# 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.** 3D shape abstraction has drawn great interest over the years. Apart from low-level representations such as meshes and voxels, researchers also seek to semantically abstract complex objects with basic geometric primitives. Recent deep learning methods rely heavily on datasets, with limited generality to unseen categories. Furthermore, abstracting an object accurately yet with a small number of primitives still remains a challenge. In this paper, we propose a novel non-parametric Bayesian statistical method to infer an abstraction, consisting of an unknown number of geometric primitives, from a point cloud. We model the generation of points as observations sampled from an infinite mixture of Gaussian Superquadric Taper Models (GSTM). Our approach formulates the abstraction as a clustering problem, in which: 1) each point is assigned to a cluster via the Chinese Restaurant Process (CRP); 2) a primitive representation is optimized for each cluster, and 3) a merging post-process is incorporated to provide a concise representation. We conduct extensive experiments on two datasets. The results indicate that our method outperforms the state-of-the-art in terms of accuracy and is generalizable to various types of objects.

**Keywords:** Superquadrics, Nonparametric Bayesian, Shape abstraction

## 1 Introduction

Over the years, 3D shape abstraction has received considerable attention. Low-level representations such as meshes [12, 19], voxels [2, 10], point clouds [1, 13] and implicit surfaces [16, 27] have succeeded in representing 3D shapes with accuracy and rich features. However, they cannot reveal the part-level geometric features of an object. Humans, on the other hand, are inclined to perceive the environment by parts [34]. Studies have shown that the human visual system makes tremendous use of part-level description to guide the perception of the environment [43]. As a result, the part-based abstraction of an object appears to be a promising way to allow a machine to perceive the environment more intelligently and hence perform higher-level tasks like decision-making and planning. Inspired by those advantages, researchers seek to abstract objects with

* Corresponding author

volumetric primitives, such as polyhedral shapes [37], spheres [21] and cuboids [28, 42, 47, 48]. Those primitives, however, are very limited in shape expressivity and suffer from accuracy issues. Superquadrics, on the other hand, are a family of geometric surfaces that include common shapes such as spheres, cuboids, cylinders, octahedra, and shapes in between, but are only encoded by five parameters. By further applying global deformations, they can express shapes such as square frustums, cones, and teardrop shapes. Due to their rich shape vocabulary, superquadrics have been widely applied in robotics, *e.g.* grasping [35, 44, 45], collision detection [39], and motion planning [38].

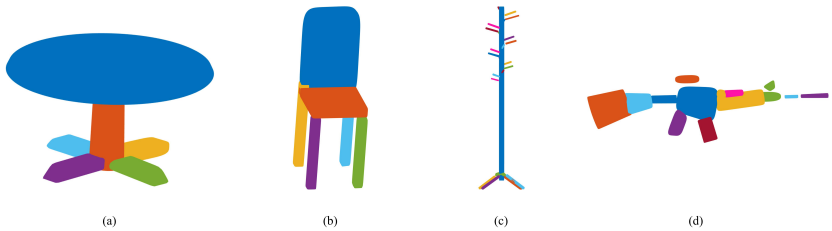



**Fig. 1.** (a)-(d) Examples of multi-tapered-superquadric-based structures of a table, chair, cloth rack, and rifle, inferred by our proposed method.

The authors of [8, 25] pioneered abstracting superquadric-based representations from complex objects. Recently, the authors in [26] developed a hierarchical process to abstract superquadric-based structures. But, their method necessitates that an object has a hierarchical geometric structure. In [31, 33], the authors utilize deep learning techniques to infer superquadric representations from voxels or images. However, the data-driven approaches show limitations in abstraction accuracy and generality beyond the training dataset.

Our work focuses on accurately abstracting a multi-tapered-superquadric-based representation of a point cloud using a small number of primitives. By assuming that an object is composed of superquadric-like components, we can regard the problem as a clustering task, which provides a means for us to reason about which portion of the point set can be properly fitted by a single tapered superquadric and thus belongs to the same cluster. The collection of tapered superquadrics fitted to each cluster constitutes the multi-tapered-superquadric-based model. Inspired by the work [26] in which the authors construct a Gaussian model around a superquadric, we build a probabilistic model by mixing Gaussian components to account for numerous components of an object. Since the number of components of an object is unknown in advance and varies case by case, we adapt our model to a nonparametric perspective to assure generality. Gibbs sampling is applied to infer the posterior distribution, in which we incorporate both an optimization method [26] for recovering superquadrics accurately from the point set and a merging process for minimizing the number of primitives, leading to a more exact, compact, and interpretable representation. Evaluations

on Shapenet [7] and D-FAUST [6] corroborate the superior performance of our method in the abstraction of 3D objects.

## 2 Related Work

In this section, we cover the mathematical definition of superquadrics and discuss relevant work on 3D representations.

### 2.1 Superquadrics

Superquadrics [4] are a family of geometric surfaces that include common shapes, such as spheres, cuboids, cylinders, and octahedra, but only encoded by five parameters. A superquadric surface can be parameterized by $\omega \in (-\pi, \pi]$ and $\eta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$:

$$\mathbf{p}(\eta, \omega) = \begin{bmatrix} C_\eta^{\varepsilon_1} \\ a_z S_\eta^{\varepsilon_1} \end{bmatrix} \otimes \begin{bmatrix} a_x C_\omega^{\varepsilon_2} \\ a_y S_\omega^{\varepsilon_2} \end{bmatrix} = \begin{bmatrix} a_x C_\eta^{\varepsilon_1} C_\omega^{\varepsilon_2} \\ a_y C_\eta^{\varepsilon_1} S_\omega^{\varepsilon_2} \\ a_z S_\eta^{\varepsilon_1} \end{bmatrix} \tag{1}$$
$$C_\alpha^\varepsilon \triangleq \text{sgn}(\cos(\alpha))|\cos(\alpha)|^\varepsilon,\ S_\alpha^\varepsilon \triangleq \text{sgn}(\sin(\alpha))|\sin(\alpha)|^\varepsilon,$$

where $\otimes$ denotes the spherical product [4], $\varepsilon_1$ and $\varepsilon_2$ define the sharpness of the shape, and $a_x$, $a_y$, and $a_z$ control the size and aspect ratio. Eq. 1 is defined within the superquadric canonical frame. The expressiveness of a superquadric can be further extended with global deformations such as bending, tapering, and twisting [5]. In our work, we apply a linear tapering transformation along z-axis defined as follows:

$$x' = \left(\frac{k_x}{a_3} z + 1\right) x,\ y' = \left(\frac{k_y}{a_3} z + 1\right) y,\ z' = z, \tag{2}$$

where $-1 \leq k_x, k_y \leq 1$ are tapering factors, $(x, y, z)$ and $(x', y', z')$ are untapered and tapered coordinates, respectively. To have a superquadric with a general pose, we apply a Euclidean transformation $g = [\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3] \in SE(3)$ to it. Thus, a tapered superquadric $\boldsymbol{\mathcal{S}}_{\boldsymbol{\theta}}$ is fully parameterized by $\boldsymbol{\theta} = [\varepsilon_1, \varepsilon_2, a_x, a_y, a_z, g, k_x, k_y]$.

### 2.2 3D representations

Based on how a 3D shape is represented, we can categorize it as a low-level or semantic representation.

**Low-level Representations** Standard 3D representations such as voxels, point clouds, and meshes have been extensively studied. In the work of [2, 10, 20, 36, 40, 41], the authors try to recover voxel models from images, which represents the 3D shapes as a regular grid. A high-resolution voxel model requires a large amount of


**Fig. 2.** (a) Examples of tapering: a cylinder can be tapered to a cone and a cuboid can be tapered to a square frustum. (b) Part-based models inferred by SQs [33]. The left one is the original mesh; the middle one is the superquadrics representation inferred from the network trained on the chair category; the right one is inferred from the network trained on the table category, indicating a limited generality of the DL approach.

memory, which limits its applications. Point clouds are a more memory-efficient way to represent 3D shapes that are utilized in [1, 13], but they fail to reveal surface connectivity. Hence, researchers also turn to exploiting meshes [12, 19, 23, 24, 29, 46] to show connections between points. Additionally, using implicit surface functions to represent 3D shapes has gained a lot of popularity [3, 9, 15, 16, 27, 30]. Although those representations can capture detailed 3D shapes, they lack interpretability as they cannot identify the semantic structures of objects.

**Part-based Semantic Representations** To abstract the semantic structures of objects, researchers have attempted to exploit various kinds of volumetric primitives such as polyhedral shapes [37], spheres [21] and cuboids [28, 42, 48]. However, their results are limited due to the shape-expressiveness of the primitives. Superquadrics, on the other hand, are more expressive. The authors of [8, 25] are pioneers in abstracting part-based structures from complex objects using superquadrics. They first segmented a complex object into parts and then fitted a single superquadric for each part. However, their work suffers from limited accuracy. Recently, the authors in [26] proposed a fast, robust, and accurate method to recover a single superquadric from point clouds. They exploited the symmetry of the superquadrics to avoid local optima and constructed a probabilistic model to reject outliers. Based on the single superquadric recovery, they developed a hierarchical way to represent a complex object with multiple superquadrics. The method is effective but requires that an object possess an inherent hierarchical structure. Another line of primitive-based abstraction is by deep learning [31–33]. Their networks demonstrate the ability to capture fine details of complex objects. However, the data-driven DL approaches are less generalizable to unseen categories. Besides, they are of a semantic-level approximation, which is lack accuracy. Instead, our method builds a probabilistic model to reason about primitive-based structures case by case, ensuring generality. Moreover, optimizations are incorporated to yield a more accurate representation for each semantic part.

# 3 Method

## 3.1 Nonparametric Clustering Formulation

In this section, we will show how to cast the problem of superquadric-based abstraction into the nonparametric clustering framework. To begin with, we model how a random point (observation) $\boldsymbol{x}$ is sampled from a superquadric primitive. First, for a superquadric parameterized by $\boldsymbol{\theta} = [\varepsilon_1, \varepsilon_2, a_x, a_y, a_z, g, k_x, k_y]$, a point $\boldsymbol{\mu} \in \mathcal{S}_{\bar{\boldsymbol{\theta}}}$, where $\bar{\boldsymbol{\theta}} = [\varepsilon_1, \varepsilon_2, a_x, a_y, a_z, (I_3, \mathbf{0}), 0, 0]$, is randomly selected across the whole surface; a noise factor $\tau$ is sampled from an univariate Gaussian distribution $\tau \sim \mathcal{N}(0, \sigma^2)$. Then, an point $\bar{\boldsymbol{x}}$ is generated as

$$\bar{\boldsymbol{x}} = (1 + \frac{\tau}{|\boldsymbol{\mu}|})\boldsymbol{\mu}, \tag{3}$$

where $\tau$ denotes the noise level that shows how far a point deflects the surface. After that, we obtain the point $\boldsymbol{x}$ by applying tapering and rigid transformation to $\bar{\boldsymbol{x}}$, *i.e.* $\boldsymbol{x} = g \circ Taper(\bar{\boldsymbol{x}})$. We call the above generative process the *Gaussian Superquadric Taper Model* (GSTM), denoted by:

$$\boldsymbol{x} \sim GSTM(\boldsymbol{\theta}, \sigma^2). \tag{4}$$

Subsequently, given a point cloud of an object $\boldsymbol{X} = \{\boldsymbol{x}_i \in \mathbb{R}^3 | i = 1, 2, ..., N\}$, we assume that each element $\boldsymbol{x}_i$ is generated from some GSTM parameterized by $(\boldsymbol{\theta}_j, \sigma_j^2)$. As a result, we consider the point cloud $\boldsymbol{X}$ as sample points generated by a mixture model as follows:

$$\boldsymbol{X} = \{\boldsymbol{x}_i | \boldsymbol{x}_i \sim \sum_{j=1}^{K} \omega_j GSTM(\boldsymbol{\theta}_j, \sigma_j^2)\}, \tag{5}$$

where $\sum_{j=1}^{K} \omega_j = 1$, and each $\omega_j$ denotes the probability that an observation is drawn from $(\boldsymbol{\theta}_j, \sigma_j^2)$. Given the observation, we can estimate a set of $\boldsymbol{\theta}$ from the mixture model. Subsequently, we assume each $\boldsymbol{\theta}_j$ is a shape representation for one semantic part of the object. And thus, we attain a set of tapered superquadrics representing the semantic structures for the object.

The EM algorithm [11] is a classical inference to solve a mixture model problem. However, EM implementation requires knowledge of $K$ – the number of components, which in our case is hard to determine beforehand from a raw point cloud. Therefore, we handle this difficulty by adapting our model to a nonparametric clustering framework, where we consider $K$ to be infinite.

To deal with the mixture model with infinitely many components, we introduce the *Dirichlet Process* (DP) into our formulation. A DP, parameterized by a base distribution $G_0$ and a concentration factor $\alpha$, is a distribution over distributions:

$$G \sim DP(G_0, \alpha), \tag{6}$$

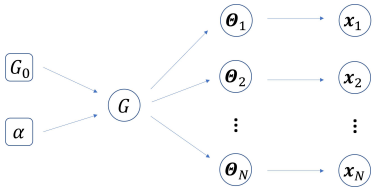



**Fig. 3.** Generative process of a point cloud. Notice that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ where $i \neq j$ could be sampled from the same $\boldsymbol{\Theta}$. Since we only draw finite samples, the number of cluster is in fact finite.

which is equivalent to

$$G = \sum_{j=1}^{\infty} \omega_j \delta_{\boldsymbol{\Theta}_j},\ \boldsymbol{\Theta}_j \sim G_0,\ \pi \sim GEM(\alpha), \tag{7}$$

where $\boldsymbol{\Theta}_j = (\boldsymbol{\theta}_j, \sigma_j^2)$, sampled from $G_0$, is the parameter of the $j$th GSTM, $\delta$ is the indicator function which evaluates to zero everywhere, except for $\delta_{\boldsymbol{\Theta}_j}(\boldsymbol{\Theta}_j) = 1$, $\pi = (\omega_1, \omega_2, ...)$, $\sum_{i=1}^{\infty} \omega_i = 1$, and GEM is the Griffiths, Engen and McCloskey distribution [17]. Therefore, an observation $\mathbf{x}_i$ is regarded as sampled from:

$$z_i \sim \pi, \boldsymbol{x}_i \sim GSTM(\boldsymbol{\Theta}_{z_i}), \tag{8}$$

where $z_i$ is a latent variable sampled from a categorical distribution parameterized by $\pi$, indicating the membership of $\boldsymbol{x}_i$. Fig. 3 illustrates the process.

Even though we have an infinite mixture model, in practice we only draw finite samples, which means the number of clusters is actually finite. One advantage of our formulation is that we do not need to impose any constraints on $K$, which is inferred from the observation $\boldsymbol{X}$. On the other hand, unlike learning-based approaches that require a large amount of training data, our method reasons about primitive-based structures case by case, relying entirely on the geometric shapes of the object. These two facts contribute to increasing the generality of being able to cope with objects of varying shapes and component counts.

### 3.2 Optimization-based Gibbs Sampling

We apply Bayesian inference to solve the mixture model problem, where the goal is to infer the posterior distribution of the parameters $\tilde{\boldsymbol{\theta}} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K\}, \tilde{\sigma}^2 = \{\sigma_1^2, \sigma_2^2, ..., \sigma_K^2\}$ and the latent variables $\boldsymbol{Z} = \{z_1, z_2, ..., z_N\}$ given the observation $\mathbf{X}$:

$$p(\tilde{\boldsymbol{\theta}}, \tilde{\sigma}^2, \boldsymbol{Z} \mid \boldsymbol{X}). \tag{9}$$

However, in practice, the Eq. 9 is intractable to obtain in a closed-form. Thus, we apply Gibbs sampling [14], an approach to estimating the desired probability

**Algorithm 1** Optimization-based Gibbs sampling

**Input:** $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$
**Output:** $\{\tilde{\boldsymbol{\theta}}^t, \tilde{\sigma}^{2^t}, \boldsymbol{Z}^t\}_{t=1}^T$
**initialize** $\{\tilde{\boldsymbol{\theta}}^t, \tilde{\sigma}^{2^t}, \boldsymbol{Z}^t\}$ for $t = 0$ by K-means clustering
**for** $t = 1, 2, ..., T$ **do**
1. draw a sample $\boldsymbol{Z}'$ for $\boldsymbol{Z}$, where $\boldsymbol{Z}' \sim p(\boldsymbol{Z} \mid \boldsymbol{X}, \tilde{\boldsymbol{\theta}}^t, \tilde{\sigma}^{2^t})$
2. optimize each element $\boldsymbol{\theta}_j$ of $\tilde{\boldsymbol{\theta}}$ conditioned on $\{\boldsymbol{Z}', \boldsymbol{X}, \tilde{\sigma}^{2^t}\}$, and let $\tilde{\boldsymbol{\theta}}'$ be the optimized $\tilde{\boldsymbol{\theta}}$
3. draw a sample $\tilde{\sigma}^{2'}$ for $\tilde{\sigma}^2$, where $\tilde{\sigma}^{2'} \sim p(\tilde{\sigma}^2 \mid \boldsymbol{X}, \boldsymbol{Z}', \tilde{\boldsymbol{\theta}}')$
4. let $\{\tilde{\boldsymbol{\theta}}^{t+1}, \tilde{\sigma}^{2^{t+1}}, \boldsymbol{Z}^{t+1}\} = \{\tilde{\boldsymbol{\theta}}', \tilde{\sigma}^{2'}, \boldsymbol{Z}'\}$
**end for**

distribution via sampling. Apart from sampling, we also incorporate an optimization process, which is used to obtain an accurate superquadric representation for each cluster. The following algorithm 1 shows how optimization-based Gibbs sampling works in our case. In the following sections, we will derive and demonstrate explicitly how to obtain each parameter.

**Sample $\boldsymbol{Z}$** To begin with, as defined in Eq. 3, we have the sampling distribution of $\boldsymbol{x}$

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}, \sigma^2) = \frac{1}{2\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{\mu}_1(\boldsymbol{\theta}, \boldsymbol{x})\|_2^2}{2\sigma^2}\right) + \frac{1}{2\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{\mu}_2(\boldsymbol{\theta}, \boldsymbol{x})\|_2^2}{2\sigma^2}\right), \tag{10}$$

where $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are two intersection points between the superquadric surface and the line joining the superqadric's origin and $\boldsymbol{x}$, as Fig. 4 shows.

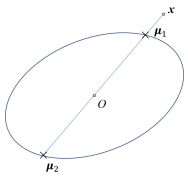



**Fig. 4.** Demonstration for computing sampling density of $\boldsymbol{x}$. According to GSTM, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the only two points accounting for the generation of $\boldsymbol{x}$.

We denote $\boldsymbol{\mu}_1$ as the closer intersection point to $\boldsymbol{x}$. In the general case, Eq. 10 is dominated by the former part, and hence we let

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}, \sigma^2) \approx \frac{1}{2\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{\mu}_1(\boldsymbol{\theta}, \boldsymbol{x})\|_2^2}{2\sigma^2}\right). \tag{11}$$

By further examination, we discover that the term $\|\boldsymbol{x} - \boldsymbol{\mu}_1(\boldsymbol{\theta}, \boldsymbol{x})\|_2$ is the radial distance between a point and a superquadric as defined in [18]. We denote it by $d(\boldsymbol{\theta}, \boldsymbol{x})$. Integrating out the $\boldsymbol{\theta}$ and $\sigma^2$ gives:

$$p(\boldsymbol{x}) = \int_{\boldsymbol{\theta},\sigma^2} p(\boldsymbol{x} \mid \boldsymbol{\theta}, \sigma^2) p(\boldsymbol{\theta}, \sigma^2)\, d\boldsymbol{\theta} d\sigma^2 \approx \int_{\boldsymbol{\theta},\sigma^2} \frac{p(\boldsymbol{\theta}, \sigma^2)}{2\sqrt{2\pi}\sigma} \exp\left(-\frac{d^2(\boldsymbol{\theta}, \boldsymbol{x})}{2\sigma^2}\right) d\boldsymbol{\theta} d\sigma^2. \tag{12}$$

Eq. 12 denotes the prior predictive density of $\boldsymbol{x}$ and is intractable to compute in closed form. It can be approximated by Monte Carlo sampling or approximated by a constant [22]. In our work, we treat $p(\boldsymbol{x})$ as a tunable hyper-parameter and denote it as $p_0$. To sample membership for each point $\boldsymbol{x}_i$, we have

$$\begin{aligned} p\left(z_i = j \mid \boldsymbol{Z}_{-i}, \boldsymbol{\theta}_j, \sigma_j^2, \boldsymbol{X}, \alpha\right) &\propto p\left(z_i = j \mid \boldsymbol{Z}_{-i}, \alpha\right) p\left(\boldsymbol{x}_i \mid \boldsymbol{\theta}_j, \sigma_j^2, \boldsymbol{Z}_{-i}\right) \\ \propto \frac{n_{-i,j}}{N - 1 + \alpha} p(\boldsymbol{x}_i \mid \boldsymbol{\theta}_j, \sigma_j^2) &= \frac{n_{-i,j}}{N - 1 + \alpha} \frac{1}{2\sqrt{2\pi}\sigma_j} \exp\left(-\frac{d^2(\boldsymbol{\theta}_j, \boldsymbol{x}_i)}{2\sigma_j^2}\right), \end{aligned} \tag{13}$$

and

$$\begin{aligned} p\left(z_i = K + 1 \mid \boldsymbol{Z}_{-i}, \boldsymbol{\theta}_j, \sigma_j^2, \boldsymbol{X}, \alpha\right) &\propto p\left(z_i = K + 1 \mid \alpha\right) p\left(\boldsymbol{x}_i \mid \boldsymbol{\theta}_j, \sigma_j^2, \boldsymbol{Z}_{-i}\right) \\ &\propto \frac{\alpha}{N - 1 + \alpha} p(\boldsymbol{x}_i) = \frac{\alpha}{N - 1 + \alpha} p_0, \end{aligned} \tag{14}$$

where $\alpha$ 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. 13 computes the probability that $\boldsymbol{x}_i$ belongs to some existing cluster, whereas Eq. 14 determines the probability of generating a new cluster. The term $p\left(z_i = j \mid \boldsymbol{Z}_{-i}, \alpha\right)$ of Eq. 13 and $p\left(z_i = K + 1 \mid \alpha\right)$ of Eq. 14 come from the *Chinese Restaurant Process* (CRP), where a point tends to be attracted by a larger population and has a fixed probability to generate a new group. The term $p\left(\boldsymbol{x}_i \mid \boldsymbol{\theta}_j, \sigma_j^2, \boldsymbol{Z}_{-i}\right)$ reasons about what the likelihood is that $\boldsymbol{x}_i$ belongs to some existing cluster or a new one, based on the current $\tilde{\boldsymbol{\theta}} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K\}$ and $\tilde{\sigma}^2 = \{\sigma_1^2, \sigma_2^2, ..., \sigma_K^2\}$. After the assignment of all points, some existing clusters may be assigned with none of the points and we remove those empty clusters. Thus, the $K$ keeps changing during each iteration. To increase the performance, we incorporate a splitting process before sampling $\boldsymbol{Z}$. Details are presented in the supplementary.

**Optimize $\tilde{\boldsymbol{\theta}}$** By independence between individual $\boldsymbol{\theta}$, the density function of each $\boldsymbol{\theta}_j$ is conditioned only on $\boldsymbol{X}^j$ and $\sigma_j^2$ as follows:

$$p(\boldsymbol{\theta}_j \mid \boldsymbol{X}^j, \sigma_j^2), \tag{15}$$

where $\boldsymbol{X}^j = \{\boldsymbol{x}_l \mid \boldsymbol{x}_l \in \boldsymbol{X}, z_l = j\}$, *i.e.* the set of points belonging to cluster $j$. By assuming that the prior for $\boldsymbol{\theta}$ is an uniform distribution, we have

$$p(\boldsymbol{\theta}_j \mid \boldsymbol{X}^j, \sigma_j^2) \propto p(\boldsymbol{\theta}_j) p(\boldsymbol{X}^j \mid \boldsymbol{\theta}_j, \sigma_j^2) \propto p(\boldsymbol{X}^j \mid \boldsymbol{\theta}_j, \sigma_j^2), \tag{16}$$

where

$$p(\boldsymbol{X}^j \mid \boldsymbol{\theta}_j, \sigma_j^2) = \prod_l \frac{1}{2\sqrt{2\pi}\sigma_j} \exp\left(-\frac{d^2(\boldsymbol{\theta}_j, \boldsymbol{x}_l)}{2\sigma_j^2}\right). \tag{17}$$

Combining Eq. 16 and Eq. 17 gives

$$p(\boldsymbol{\theta}_j \mid \boldsymbol{X}^j, \sigma_j^2) \propto \prod_l \exp\left(-\frac{d^2(\boldsymbol{\theta}_j, \boldsymbol{x}_l)}{2\sigma_j^2}\right) = \exp\left(-\sum_l \frac{d^2(\boldsymbol{\theta}_j, \boldsymbol{x}_l)}{2\sigma_j^2}\right). \tag{18}$$

Gibbs sampling requires sampling $\boldsymbol{\theta}_j$ from Eq. 18. However, directly sampling from Eq. 18 is difficult due to its complexity. Instead, optimization is used as a substitute for the sampling process, which, we believe, is a reasonable replacement. By inspecting Eq. 18 more closely, we discover that the $\boldsymbol{\theta}_j$ minimizing $\sum d^2(\boldsymbol{\theta}_j, \boldsymbol{x}_l)$ maximizes the density function. Therefore, an optimized $\boldsymbol{\theta}_j$ has relatively higher likelihood to be close to the actual sample of $\boldsymbol{\theta}_j$ drawn from $p(\boldsymbol{\theta}_j \mid \boldsymbol{X}^j, \sigma_j^2)$. And closeness implies similar shapes. Additionally, we recognize that optimizing Eq. 18 can be regarded as a single superquadric recovery problem, which requires the abstraction of an optimal superquadric primitive from the cluster points $\boldsymbol{X}^j$. In other words, we fit an optimal superquadric to each cluster, and those superquadrics will affect the membership of each point in the subsequent iteration, which is a process similar to EM. As a result, we use the robust and accurate recovery algorithm [26], which yields an optimal superquadric with high fidelity, to acquire each $\boldsymbol{\theta}_j$ in replacement of the sampling.

**Sample $\tilde{\boldsymbol{\sigma}}^{\boldsymbol{2}}$** Similarly, by independence, we have

$$\begin{aligned} \sigma_j^{2'} &\sim p(\sigma_j^2 \mid \boldsymbol{X}^j, \boldsymbol{\theta}_j) \\ \tilde{\sigma}^{2'} &= \{\sigma_1^{2'}, \sigma_2^{2'}, ..., \sigma_K^{2'}\}. \end{aligned} \tag{19}$$

We also assume the non-informative prior for $\sigma^2$ is the uniform distribution, which gives

$$p(\sigma_j^2 \mid \boldsymbol{X}^j, \boldsymbol{\theta}_j) \propto p(\sigma_j^2) p(\boldsymbol{X}^j \mid \boldsymbol{\theta}_j, \sigma_j^2) \propto p(\boldsymbol{X}^j \mid \boldsymbol{\theta}_j, \sigma_j^2). \tag{20}$$

Combining Eq. 17 and Eq. 20 gives

$$\begin{aligned} p(\sigma_j^2 \mid \boldsymbol{X}^j, \boldsymbol{\theta}_j) &\propto \prod_l \frac{1}{2\sqrt{2\pi}\sigma_j} \exp\left(-\frac{d^2(\boldsymbol{\theta}_j, \boldsymbol{x}_l)}{2\sigma_j^2}\right) \\ &\propto \left(\frac{1}{\sigma_j}\right)^{n_j} \exp\left(-\sum_l \frac{d^2(\boldsymbol{\theta}_j, \boldsymbol{x}_l)}{2\sigma_j^2}\right), \end{aligned} \tag{21}$$

where $n_j$ is the number of $\boldsymbol{X^j}$. Let $D = \sum_l d^2(\boldsymbol{\theta}_j, \boldsymbol{x}_l)$ and $\gamma_j = \frac{1}{\sigma_j^2}$. By change of variable, we have

$$\gamma_j' \sim p(\gamma_j \mid \boldsymbol{X}^j, \boldsymbol{\theta}_j) \propto \gamma_j^{\frac{n_j - 3}{2}} \exp(-\frac{D}{2}\gamma_j). \tag{22}$$

Hence, $\gamma_j$ follows a gamma distribution with shape parameter $\frac{n_j-1}{2}$ and scale parameter $\frac{2}{D}$. In other words,

$$\begin{aligned} \sigma_j^{2'} &= \frac{1}{\gamma_j'} \\ \gamma_j' &\sim \boldsymbol{\Gamma}\left(\frac{n_j-1}{2}, \frac{2}{D}\right). \end{aligned} \tag{23}$$

$D$ reflects how good the optimized superquadric fits the cluster points. With lower value of $D$, $\gamma_j'$ will have a better chance to be higher and hence $\sigma_j^{2'}$ will be smaller. In other words, the better the fitting is, the smaller the noise level will be.

### 3.3 Merging process

We observe that our method yields structures consisting of excessive components, resulting in less interpretability. Therefore, we design a merging post-process minimizing component numbers while maintaining accuracy. Specifically, for any two clusters represented by two superquadrics, we make a union of the two sets of points into one set, from which we recover a superquadric. If the newly recovered superquadric turns out to be a good fit for the new point set, we will merge these two clusters into one, and replace the two original superquadrics with the newly fitted one. Detailed formulations and procedures are presented in the supplementary.

## 4 Experiment

In this section, we demonstrate our approach to abstracting part-level structures exhibits high accuracy, compared with state-of-the-art part-based abstraction method [33]. We do not compare with the work of [31, 32] since their work focuses mainly on abstracting 3D shapes from 2D images. We also include a simple clustering method as a baseline, where the point cloud is parsed into $K$ clusters via K-means and each cluster is then represented by an optimized superquadric [26]. We conduct experiments on the ShapeNet dataset [7] and the D-FAUST dataset [6]. The ShapeNet is a collection of CAD models of various common objects such as tables, chairs, bottles, etc. On the other hand, the D-FAUST dataset contains point clouds of 129 sequences of 10 humans performing various movements, *e.g.*, punching, shaking arms, and running. Following [33], we evaluate the results with two metrics, Chamfer $L_1$-distance and Intersection over Union (IoU). Detailed computations of the two metrics are discussed in the supplementary.

**Initialization:** we parse the point cloud into $K$ components based on the K-means clustering algorithm. Although we specify the value of $K$ initially, the final value of $K$ will be inferred by our nonparametric model and vary from case to case. The latent variable $z_i$ of each point $\boldsymbol{x}_i$ is assigned accordingly.

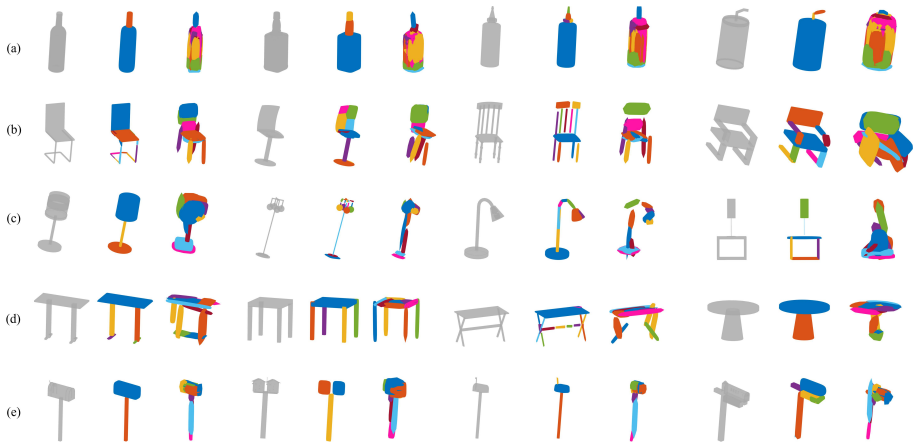
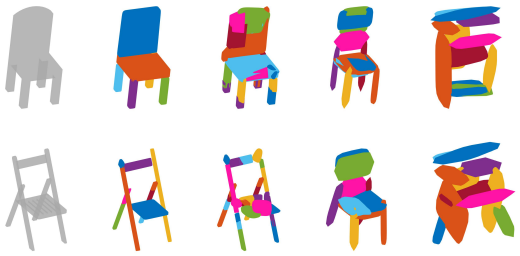



**Fig. 5.** Qualitative results of 3D abstraction on various objects. The left ones are the original meshes, the middle ones are our inferred results, and the right ones are inferred from SQs [33]. (a) Bottle, (b) chair, (c) lamp, (d) table, and (e) mailbox.

Subsequently, each cluster is represented by an ellipsoid $\boldsymbol{\theta}_j^0$ whose moment-of-inertial (MoI) is four times smaller than the MoI of the cluster points; each $\sigma_j^{2^0}$ is randomly sampled within $(0, 1]$. We set the number of the sampling iteration to be $T = 30$, concentration factor $\alpha = 0.5$, and $p_0 = 0.1$.

### 4.1 Evaluation on ShapeNet

We choose seven different types of objects among all of the categories. For deep learning training, we randomly divide the data of each object into two sets – a training set (80%) and a testing set (20%), and we compare the results on the testing set. Since ShapeNet only provides meshes, we first densely sample points on the mesh surfaces and then downsample the point clouds to be around 3500 points. For all categories, we set the $K = 30$. The result is summarized in Table 1, where w/om denotes our method excluding the merging process. Our method outperforms the state-of-the-art [33] and the K-means baseline significantly on all object types. Excluding the merging process improves accuracy but increases the number of primitives, making the abstracted models less interpretable. Therefore, we believe merging is beneficial and important because it reduces the primitive numbers while maintaining excellent accuracy, which improves interpretability. Unlike the learning-based method, which is a semantic-level approximation, our method infers the part-based representation in an optimization-based manner. As a result, our method yields a more geometrically accurate primitive-based structure, yet with a compact number of primitives. A qualitative comparison between our method and SQs [33] is depicted in Fig. 5.



(a)

| | Chamfer-$L_1$ | IoU |
|---|---|---|
| SQs | 0.054 | 0.269 |
| SQs (table/chair) | 0.116 | 0.127 |
| ours | **0.024** | **0.557** |

(b)

**Fig. 6.** (a) Comparison between different results inferred by different models. From left to right: the original meshes, results inferred by our method, results inferred by our method excluding merging, results inferred by the baseline trained on the chair category, and results inferred by the baseline trained on the table category, (b) quantitative results showing that the baseline method has limited generality. The (table/chair) means that a network trained on the table category is used to predict the chair category.

**Table 1.** Quantitative results on ShapeNet

| | Chamfer-$L_1$ | | | | IoU | | | |
|---|---|---|---|---|---|---|---|---|
| Category | K-means | SQs [33] | w/om | Ours | K-means | SQs | w/om | Ours |
| bottle | 0.064 | 0.037 | 0.026 | **0.019** | 0.552 | 0.596 | 0.618 | **0.656** |
| can | 0.086 | 0.032 | 0.028 | **0.014** | 0.690 | 0.736 | **0.803** | 0.802 |
| chair | 0.065 | 0.054 | **0.018** | 0.024 | 0.433 | 0.269 | **0.577** | 0.557 |
| lamp | 0.066 | 0.065 | **0.020** | 0.024 | 0.354 | 0.190 | **0.425** | 0.414 |
| mailbox | 0.054 | 0.059 | 0.019 | **0.019** | 0.529 | 0.400 | **0.687** | 0.686 |
| rifle | 0.018 | 0.022 | **0.009** | 0.013 | 0.517 | 0.291 | **0.594** | 0.536 |
| table | 0.060 | 0.057 | **0.018** | 0.021 | 0.374 | 0.194 | **0.536** | 0.512 |
| mean | 0.057 | 0.053 | **0.017** | 0.021 | 0.410 | 0.242 | **0.547** | 0.526 |

| | #primitives | | | |
|---|---|---|---|---|
| Category | K-means | SQs | w/om | Ours |
| bottle | 30 | 8 | 7.4 | 6.8 |
| can | 30 | 7 | 13.6 | 1.1 |
| chair | 30 | 10 | 26.9 | 13.6 |
| lamp | 30 | 10 | 24.8 | 9.5 |
| mailbox | 30 | 10 | 18.8 | 3.1 |
| rifle | 30 | 7 | 20.0 | 7.6 |
| table | 30 | 11 | 25.1 | 9.5 |

Furthermore, generality is noteworthy. To attain the reported accuracy, the baseline method needs to be trained on a dataset of a specified item category, respectively. A network trained on one item is difficult to generalize to another as Fig. 6 shows. In contrast, our probabilistic formulation reasons about the part-based representation case by case, and the nonparametric formulation makes it possible to adapt to various shapes with varying component numbers.

### 4.2 Evaluation on D-FAUST

We follow the same split strategy in [31] and divide the dataset into training (91), testing (29), and validation (9). Likewise, we compare results on the testing set. For our method, we downsample the point clouds to be around 5500 points and set $K$ to be 30, as well. The results are shown in table 2. Fig. 7 illustrates examples of inferred representations. We can observe that our model can accurately capture the major parts of humans, *i.e.* heads, chests, arms, forearms, hips, thighs, legs, and feet, even when they are performing different movements.

**Table 2.** Quantitative results on D-FAUST

| | Chamfer-$L_1$ | IoU |
|---|---|---|
| SQs[33] | 0.0473 | 0.7138 |
| ours | **0.0335** | **0.7709** |

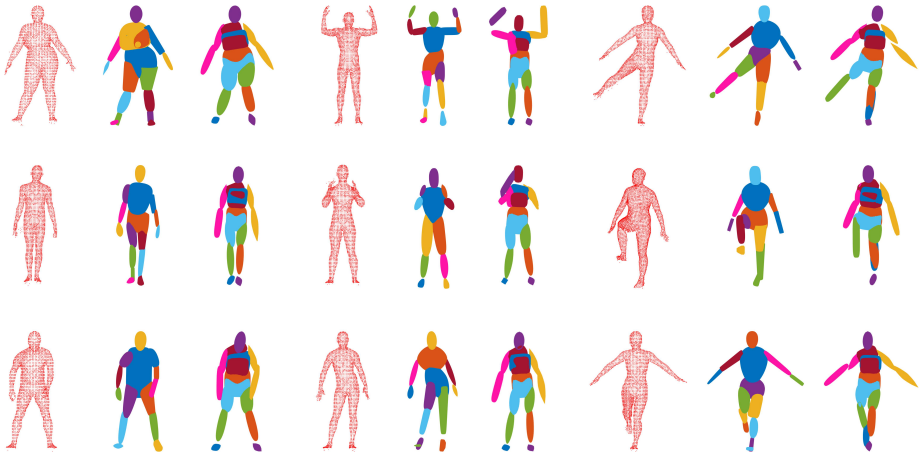


**Fig. 7.** Abstraction results on D-FAUST dataset. The left ones are the original point clouds, the middle ones are inferred by our method, and the right ones are from SQs [33].

### 4.3 Extension: Point Cloud Segmentation

Due to the fact that our method yields a geometrically accurate structure, we can achieve a geometry-driven point clouds segmentation task naturally. All points in the point clouds have been well clustered after we obtain the abstraction of an object and we segment the point clouds accordingly. Fig. 8 illustrates some examples of point clouds segmentation on different objects, inferred by our method.

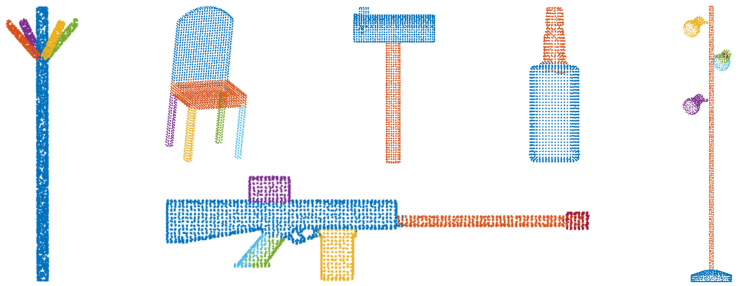


**Fig. 8.** Examples of point clouds segmentation inferred by our method.

## 5 Conclusions & Limitations

In this paper, we present a novel method to abstract the semantic structures of an object. We cast the problem into a nonparametric clustering framework and solve it by the proposed optimization-based Gibbs sampling. Additionally, since our method yields a semantically meaningful structure, we can achieve a geometry-driven point clouds segmentation task naturally. However, there are some limitations to our method. Firstly, compared with deep learning methods, our implementation is less efficient and cannot be applied in real-time at this moment. In addition, for some certain categories of objects, such as watercraft and airplanes, which barely consist of superquadric-like parts, the performance of our algorithm is expected to drop. Furthermore, learning-based methods can produce results with better semantic consistency than ours.

Future work will focus on extending the expressiveness of superquadrics by applying more deformations beyond tapering, such as bending and sheering. Additionally, our formulation of how a random point is sampled from a tapered superquadric primitive can be extended to a more general surface beyond superquadrics. Moreover, trying different priors for both $\boldsymbol{\theta}$ and $\sigma^2$ other than uniform distributions is also an auspicious way to improve performance.

**Acknowledgments** This research is supported by the National Research Foundation, Singapore, under its Medium Sized Centre Programme - Centre for Advanced Robotics Technology Innovation (CARTIN) R-261-521-002-592.

## References


1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International conference on machine learning. pp. 40–49. PMLR (2018)
2. Anwar, Z., Ferrie, F.: Towards robust voxel-coloring: Handling camera calibration errors and partial emptiness of surface voxels. In: 18th International Conference on Pattern Recognition (ICPR). vol. 1, pp. 98–102. IEEE (2006)
3. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2565–2574 (2020)
4. Barr, A.H.: Superquadrics and angle-preserving transformations. IEEE Computer graphics and Applications **1**(1), 11–23 (1981)
5. Barr, A.H.: Global and local deformations of solid primitives. In: Readings in Computer Vision, pp. 661–670. Elsevier (1987)
6. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic FAUST: Registering human bodies in motion. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2017)
7. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
8. Chevalier, L., Jaillet, F., Baskurt, A.: Segmentation and superquadric modeling of 3d objects. In: WSCG (2003)
9. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6970–6981 (2020)
10. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European Conference on Computer Vision (ECCV). pp. 628–644. Springer (2016)
11. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological) **39**(1), 1–22 (1977)
12. Deng, B., Genova, K., Yazdani, S., Bouaziz, S., Hinton, G., Tagliasacchi, A.: Cvxnet: Learnable convex decomposition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 31–44 (2020)
13. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 605–613 (2017)
14. Gelfand, A.E.: Gibbs sampling. Journal of the American statistical Association **95**(452), 1300–1304 (2000)
15. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local deep implicit functions for 3d shape. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4857–4866 (2020)
16. Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7154–7164 (2019)
17. Gnedin, A., Kerov, S.: A characterization of gem distributions. Combinatorics, Probability and Computing **10**(3), 213–217 (2001)

18. 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)
19. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 216–224 (2018)
20. Häne, C., Tulsiani, S., Malik, J.: Hierarchical surface prediction for 3d object reconstruction. In: 2017 International Conference on 3D Vision (3DV). pp. 412–420. IEEE (2017)
21. Hao, Z., Averbuch-Elor, H., Snavely, N., Belongie, S.: Dualsdf: Semantic shape manipulation using a two-level representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7631–7641 (2020)
22. Hayden, D.S., Pacheco, J., Fisher, J.W.: Nonparametric object and parts modeling with lie group dynamics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7426–7435 (2020)
23. Jimenez Rezende, D., Eslami, S., Mohamed, S., Battaglia, P., Jaderberg, M., Heess, N.: Unsupervised learning of 3d structure from images. Advances in neural information processing systems **29** (2016)
24. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 371–386 (2018)
25. Leonardis, A., Jaklic, A., Solina, F.: Superquadrics for segmenting and modeling range data. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(11), 1289–1295 (1997)
26. 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)
27. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4460–4470 (2019)
28. Niu, C., Li, J., Xu, K.: Im2struct: Recovering 3d shape structure from a single rgb image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4521–4529 (2018)
29. Pan, J., Han, X., Chen, W., Tang, J., Jia, K.: Deep mesh reconstruction from single rgb images via topology modification networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9964–9973 (2019)
30. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 165–174 (2019)
31. Paschalidou, D., Gool, L.V., Geiger, A.: Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1060–1070 (2020)
32. Paschalidou, D., Katharopoulos, A., Geiger, A., Fidler, S.: Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3204–3215 (2021)

33. 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)
34. Pentland, A.P.: Perceptual organization and the representation of natural form. In: Readings in Computer Vision, pp. 680–699. Elsevier (1987)
35. Quispe, A.H., Milville, B., Gutiérrez, M.A., Erdogan, C., Stilman, M., Christensen, H., Amor, H.B.: Exploiting symmetries and extrusions for grasping household objects. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 3702–3708. IEEE (2015)
36. Riegler, G., Osman Ulusoy, A., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3577–3586 (2017)
37. Roberts, L.G.: Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology (1963)
38. Ruan, S., Chirikjian, G.S.: Closed-form minkowski sums of convex bodies with smooth positively curved boundaries. Computer-Aided Design **143**, 103133 (2022)
39. Ruan, S., Poblete, K.L., Li, Y., Lin, Q., Ma, Q., Chirikjian, G.S.: Efficient exact collision detection between ellipsoids and superquadrics via closed-form minkowski sums. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 1765–1771. IEEE (2019)
40. Slabaugh, G.G., Culbertson, W.B., Malzbender, T., Stevens, M.R., Schafer, R.W.: Methods for volumetric reconstruction of visual scenes. International Journal of Computer Vision **57**(3), 179–199 (2004)
41. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 2088–2096 (2017)
42. Tulsiani, S., Su, H., Guibas, L.J., Efros, A.A., Malik, J.: Learning shape abstractions by assembling volumetric primitives. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2635–2643 (2017)
43. Tversky, B., Hemenway, K.: Objects, parts, and categories. Journal of experimental psychology: General **113**(2), 169 (1984)
44. Vezzani, G., Pattacini, U., Natale, L.: A grasping approach based on superquadric models. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 1579–1586. IEEE (2017)
45. Vezzani, G., Pattacini, U., Pasquale, G., Natale, L.: Improving superquadric modeling and grasping with prior on object shapes. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 6875–6882. IEEE (2018)
46. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proceedings of the European conference on computer vision (ECCV). pp. 52–67 (2018)
47. Yang, K., Chen, X.: Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. ACM Transactions on Graphics (TOG) **40**(4), 1–11 (2021)
48. Zou, C., Yumer, E., Yang, J., Ceylan, D., Hoiem, D.: 3d-prnn: Generating shape primitives with recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 900–909 (2017)