Marching-Primitives: Shape Abstraction from Signed Distance Function

Weixiao Liu[1,2] Yuwei Wu[1] Sipu Ruan[1] Gregory S. Chirikjian[1]*
[1]National University of Singapore [2]Johns Hopkins University
{mpewxl, yw.wu, ruansp, mpegre}@nus.edu.sg

Abstract

Representing complex objects with basic geometric primitives has long been a topic in computer vision. Primitive-based representations have the merits of compactness and computational efficiency in higher-level tasks such as physics simulation, collision checking, and robotic manipulation. Unlike previous works which extract polygonal meshes from a signed distance function (SDF), in this paper, we present a novel method, named Marching-Primitives, to obtain a primitive-based abstraction directly from an SDF. Our method grows geometric primitives (such as superquadrics) iteratively by analyzing the connectivity of voxels while marching at different levels of signed distance. For each valid connected volume of interest, we march on the scope of voxels from which a primitive is able to be extracted in a probabilistic sense and simultaneously solve for the parameters of the primitive to capture the underlying local geometry. We evaluate the performance of our method on both synthetic and real-world datasets. The results show that the proposed method outperforms the state-of-the-art in terms of accuracy, and is directly generalizable among different categories and scales. The code is open-sourced at https://github.com/ChirikjianLab/Marching-Primitives.git.

1. Introduction

Recent years have witnessed great progress in the areas of 3D shape representation and environmental perception. Low-level representations such as surface meshes, point clouds, and occupancy grids are widely used as inputs to high-level computer vision algorithms and artificial intelligence tasks. They have the advantage of being able to represent and visualize objects with high accuracy and rich local geometric features. However, the low-level representations are ineffective in delivering a general and intuitive sense of structural geometry as well as part-level scene understanding. Studies [3, 20] show that human vision, unlike

Figure 1. Primitive-base representation versus mesh. For each pair of objects, the left one is the superquadric abstraction obtained by our algorithm, and the right one is the original mesh. The mesh of the chair is 6MB in size, while our representation only needs 4KB. An SDF representation discretized on a 128^3 voxel grid occupies 19MB. Our abstraction is equivalent to an implicit continuous SDF, which is an approximation to the discrete SDF.

computer vision, tends to perceive and understand scenes as combinations of simple primitive shapes. Human beings perform well and robustly in complex tasks, providing a basic geometric description of the scene is available [32]. Therefore, researchers turn to exploring the possibility of interpreting complex objects and scenes with basic geometric primitives. Taking advantage of the primitive-based representation, many higher-level tasks, such as segmentation [14, 16, 21, 30], scene understanding [29, 31, 41, 47], grasping [33, 44, 45] and motion planning [35, 36], are able to be solved efficiently.

However, it still remains challenging to extract primitive-based abstractions from low-level representations. Starting from the 1990s, Solina *et al*. [1, 17, 39] aim to extract a single superquadric representation from a simple object by minimizing the least-square error between the primitive and the measured points. Later in [7, 22], their method is extended to represent more complex objects with multiple primitives. More recently, the authors of [24, 47] reformulate the task as a probabilistic inference problem with enhanced accuracy and robustness to noise and outliers. At the same time, with the surge of data-driven techniques, researchers attempt to train neural networks to infer cuboids [27,38,41,48,50] and superquadrics [29,31] representations in an end-to-end fashion. However, both the computational and learning-based approaches have their own limitations. The computational methods are vulnerable to the inherent ambiguity of the point-to-surface relationship. For exam-

*Corresponding author

ple, the algorithms tend to fill empty spaces of a non-convex object with primitives by mistake, due to the inside/outside ambiguity of a surface depicted by a set of points [24, 48]. The main drawback of the learning approaches lies in the lack of generalizability beyond the object category on which the model is trained [24,31,47,48]. Also, the shape abstraction accuracy is inferior to the computational methods.

The signed distance function (SDF) has been a successful 3D volumetric representation in varieties of computer vision and graphics tasks. It is the basic framework for many classic 3D reconstruction algorithms such as TSDF volume reconstruction [10, 15], KinectFusion [19], and DynamicFusion [26]. Recently, the SDF representation is adapted to the deep learning frameworks, and exhibits boosted potentials in shape encoding [8, 18, 28, 43], surface reconstruction [23, 46], and shape completion [11, 12, 34]. Usually, triangular mesh surfaces are extracted from the SDF representation with the marching cubes algorithm [25]. Point cloud and occupancy grid representations are also obtained by keeping the vertices of the meshes and the sign of each voxel point, respectively. The SDF is among the most informative 3D representations since it encodes not only the surface geometry but also the distance and side of a point relative to the shape. Meanwhile, it is easily achievable via range images from 3D sensors [10], or learnable from other input modalities [8, 18, 28]. Since we are able to extract meshes from an SDF, it is natural to think about the possibility of extracting primitives as well. Furthermore, the primitive-based abstraction is a continuous interpretation of the complete geometric information encoded in the original discrete SDF, but requires much less storage size (Fig.1).

Motivated by the aforementioned facts and the bottleneck of the current shape abstraction algorithms, we proposed a general shape abstraction method by reasoning directly on the informative SDF representation. The goal of our method is to find a combination of geometric primitives whose underlying SDF values match the target values evaluated on the evenly spaced discrete grid points (Sec. 3.1 and Sec. 3.2). To solve this problem, we propose a two-step iterative algorithm called the Marching-Primitives. Our algorithm 'marches' on two domains: the signed distance domain and the voxelized space domain, alternately. Firstly, the connectivity of volumes are analyzed by generating isosurfaces on a sequence of decreasing levels of negative signed distances (Sec.3.3). By doing so, volumes of interest (VOIs) where primitives are likely to be encoded can be identified sequentially. In the second step, for each of the VOIs, our algorithm marches on the neighbouring voxels to infer their probabilistic correspondences to the primitive and simultaneously optimizes the shape and pose of the primitive (Sec.3.4). After the primitive representation of a VOI is achieved, the fitted volumes are deactivated from the voxel grid. Our algorithm continues marching on the signed distance domain until it approaches zero, *i.e.*, all the interior volumes of the SDF have been captured by the recovered primitives. We compare our algorithm with the state-of-the-art of both the computational and learning-based approaches on the ShapeNet object dataset [6] and D-FAUST human shape dataset [4] (Sec. 4.1). We also study the performance of our algorithm on different conditions(Sec. 4.2). Finally, we demonstrate the scene abstraction result of the Stanford Reading Room [49], which contains several pieces of furniture of various categories(Sec. 4.3).

2. Related Work

SDF Representation: The SDF can be stored in two different ways: discrete or continuous. A majority of computer vision and graphics algorithms are built on the SDF discretized on a 3D grid of voxel points. The signed distances are stored on each of the corresponding voxel points. The authors of [10, 19, 26] pioneer in fusing several noisy range images into a single discrete SDF. Their work is widely applied in 3D reconstruction and plays an important role in robotics tasks such as simultaneous localization and mapping. The discrete SDF is also a promising input/output representation for 3D deep learning [11, 12, 34]. Recently, it becomes popular to encode shapes as a continuous SDF with neural networks [28]. Vasu *et al*. [43] further improve the shape encoding quality by enforcing local regularities with geometric primitives. In [8], the authors adopt a two-stage meta-learning approach to further extend the generalization capabilities of neural SDF. With the deep neural network, it becomes possible to infer SDF representations from partially observed 3D inputs or even images. Both the discrete and the continuous SDF are implicit representations of geometric surfaces. To extract the explicit surface from the SDF, continuous SDFs need to be discretized on a voxel grid first and then conduct the marching cubes [25]. This method allows high-quality rendering of the objects, however, surface meshes are non-sparse and contain no structural level information. Our method provides an alternative approach to describe the underlying object in the SDF. Instead of meshes, we directly extract a collection of sparsely parameterized primitives from the SDF. Other than that, our primitive-based representation itself is also a concise yet continuous SDF approximation to the original discrete SDF.

Computational Shape Abstraction: The most well-studied primitive for computational shape abstraction is the superquadric, due to its extensive shape vocabulary including cuboids, ellipsoids, cylinders, octohedra, and many shapes in between (*e.g*., cuboids with rounded edges). It is first proposed as a versatile modeling element for complex objects in computer graphics [2, 32]. Later, Solina *et al*. propose a method to conduct abstraction of simple objects from range images with a single superquadric [17, 39]. Leonardis *et al*. [22] and Chevalier *et al*. [7] further extend the previous work to recover complex objects with multiple superquadrics with a *Split-and-Merge* strategy. A numerical instability problem is addressed and revisited in [42]. The authors introduce an auxiliary function in the unstable region and receive a better abstraction accuracy. More recently, Liu *et al*. [24] formulate the problem in a probabilistic fashion and propose a geometric strategy to avoid local optimum, bringing a significant improvement in robustness to outlier and fitting accuracy. Wu *et al*. [47] extend and recast the work as a nonparametric Bayesian inference problem so as to improve the applicability on complex shapes. To the best of the authors' knowledge, the existing computational methods are all based on range images or point clouds, which suffer from geometric ambiguities [48]. In contrast, our method takes advantage of the abundant geometric information encoded in the SDF and is easily compatible with other computer vision algorithms based on the SDF representation.

Learning-based Shape Abstraction: The learning-based method is first seen in [41]. Tulsiani *et al*. propose a 3D convolutional neural network (CNN) to learn shape abstractions with cuboids from the occupancy grid. Sun *et al*. [40] design an adaptive hierarchical cuboid representation and introduce an unsupervised approach to learn to extract the parameters for the representation. Yang *et al*. [48] train a variational auto-encoder network to transform point clouds into parametric cuboids. Other than cuboids, researchers also seek to extract spheres and ellipsoids representations from objects. Hao *et al*. [18] combine the neural SDF with the spherical representation by sharing a same latent layer. In [37], ellipsoids or cuboids are extracted to help segment the input point cloud. However, a single type of primitive has very limited expressiveness. Therefore, Paschalidou *et al*. [29, 31] turn to training neural networks to conduct abstractions with the superquadrics as the atomic elements. Learning-based approaches are versatile in dealing with different input sources. They are able to make shape abstractions from point clouds, voxel grids, or even RGB images which are so ill-conditioned that the computational approaches have little chance of working. However, learning-based approaches rely heavily on the training dataset and thus are less generalizable to unseen categories. Instead, our approach reasons about the primitive abstraction from a case-by-case geometric perspective, which provides an inherent advantage in generalizability and accuracy.

3. Method

3.1. Preliminary

The discrete SDF is a volumetric surface representation built on a voxel grid $\mathbf{V} = \{\mathbf{x}_i \in \mathbb{R}^3, i = 1, 2, ..., N\}$. A scalar $d(\mathbf{x}_i)$ is assigned to each grid point, which indicates the signed distance of $\mathbf{x}_i$ to the nearest surface. The point $\mathbf{x}_i$ lies inside the surface if $d(\mathbf{x}_i) < 0$ and outside otherwise. Typically, the surface mesh of an object is extracted from the SDF by the marching cubes algorithm [25]. In this paper, we call the input SDF $d(\mathbf{x}_i)$ the target SDF.

For the primitive representation, we select the superquadrics [2], a family of geometric primitives defined by the implicit equation

$$f(\mathbf{x}) = \left(\left(\frac{x}{a_x}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_y}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_z}\right)^{\frac{2}{\epsilon_1}} = 1 \quad (1)$$

The superquadric family is only encoded by 5 parameters (shape parameters $\epsilon_1, \epsilon_2 \in [0, 2] \subset \mathbb{R}$, and scale parameters $a_x, a_y, a_z \in \mathbb{R}_{>0}$), but has an extensive shape vocabulary. Note that the shape parameters can exceed 2, resulting in nonconvex shapes. However, practically in most studies [24, 31] and also in our paper, we limit them within the convex region as defined above. The points $\mathbf{x} = [x, y, z] \in \mathbb{R}^3$ satisfying Eq. (1) form the surface of the superquadric. In this paper, we also include the Euclidean transformation $g \in SE(3)$, *i.e.* 3 Euler angles for rotation $\mathbf{R} \in SO(3)$ and 3-dimensional translation $\mathbf{t} \in \mathbb{R}^3$, to parameterize a superquadric with a general pose. In total, we denote a superquadric with a vector $\boldsymbol{\theta}$ of 11 elements. According to Eq. (1), we can approximate the signed distance of a grid point $\mathbf{x}_i$ to a general posed superquadric $\boldsymbol{\theta}$ explicitly by

$$d_{\boldsymbol{\theta}}(\mathbf{x}_i) = \left(1 - f^{-\frac{\epsilon_1}{2}}(g^{-1} \circ \mathbf{x}_i)\right) \|g^{-1} \circ \mathbf{x}_i\|_2 \quad (2)$$

We are not able to use the exact SDF of superquadrics, because it has no analytical solution and is only achievable by numerical optimization [5]. Eq.(2) is the radial distance [17, 24] of $\mathbf{x}_i$ to the superquadric surface, which converges to the exact signed distance as $d_{\boldsymbol{\theta}}(\mathbf{x}_i) \to 0$. Therefore, we truncate both the input target SDF $d(\cdot)$ and the primitive SDF $d_{\boldsymbol{\theta}}(\cdot)$ within the vicinity of zero to ensure the approximation accuracy.

3.2. Problem Formulation

To obtain a primitive-based abstraction from the target SDF, we seek a combination of primitives $\boldsymbol{\Theta} \doteq \{\boldsymbol{\theta}_k, k = 1, 2, ..., K\}$ whose underlying signed distances measured on the voxel points (we call it the source SDF in contrast to the target SDF) match the target SDF, that is

$$\boldsymbol{\Theta} = \arg\min_{\boldsymbol{\Theta}} \sum_{\mathbf{x}_i \in \mathbf{V}} \min_{\boldsymbol{\theta}_k \in \boldsymbol{\Theta}} \left\| d_{\boldsymbol{\theta}_k}(\mathbf{x}_i) - d(\mathbf{x}_i) \right\|_2^2 \quad (3)$$

Our problem formulation is simple and intuitive, however, it is intractable to solve directly. First, the number of primitives K needed is unknown. Also, the correspondences between the voxel points $\mathbf{x}_i$ and the primitives $\boldsymbol{\theta}_k$ are unknown a priori. Therefore, it makes Eq.(3) a chicken-and-egg problem: on the one hand, if we know the set of voxel points contributing to a same primitive, we are able to solve for the parameters of the primitive by optimization; on the other hand, if we know the configurations of the primitives, we are able to tell the belongings of each voxel. Other than those factors, the computation complexity itself is prohibitively high, because we need to evaluate hundreds of thousands or even millions of voxel points. This makes it infeasible to operate directly on the complete dense voxel grid. To tackle these difficulties, we propose an iterative algorithm, the Marching-Primitives, which simultaneously solves the correspondences and primitive parameters efficiently. It is worth noting that our method is generalizable beyond the superquadrics. Any volumetric primitives with easily accessible SDF representations can be adapted to our framework.

3.3. Iterative Connectivity Marching

Instead of setting a predefined number of primitives, our method starts from an empty set of primitives and grows as needed. This is realized by analyzing the connectivity (in this paper, we use the 26-connectivity) of the voxels at different levels of signed distance. Given the target SDF, our algorithm checks the connectivity of voxels whose signed distance is less than a sequence of thresholds

$$T^c \doteq \{t_1^c, t_2^c, ...\}, \quad t_1^c = \min_{\mathbf{x}_i \in \mathbf{V}} d(\mathbf{x}_i), \quad t_{m+1}^c = \alpha t_m^c \quad (4)$$

where $\alpha \in (0,1) \subset \mathbb{R}$ is the common ratio, resulting in a geometric sequence exponentially decaying to zero. Each threshold t_m^c defines a set of disjoint isosurfaces S_m, encompassing the connected voxels whose signed distances are less than the threshold.

$$S_m = \{\mathcal{S}_k, k = 1, 2, ..., |S_m|\} \quad (5)$$

$|S_m|$ denotes the number of the disjoint isosurfaces. We construct a subset $\bar{S}_m$ from S_m

$$\bar{S}_m = \{\mathcal{S}_k \in S_m, |\mathcal{S}_k| \geq N_c\} \subseteq S_m \quad (6)$$

where $|\mathcal{S}_k|$ is the number of the connected voxels within the isosurface $\mathcal{S}_k$. $\bar{S}_m$ is the set of the isosurfaces encompassing no less than N_c connected voxels. The connectivity marching starts from the innermost threshold t_1^c, where the resulting $\bar{S}_1$ might be empty. The marching continues on the sequence (increasing the threshold) until $\bar{S}_m$ is non-empty. Each of the connected volumes in $\bar{S}_m$ (we call it a VOI) is an ideal starting point for growing a primitive, because: (1) those volumes are among the most interior of the target SDF, corresponding to most prominent part of the geometry; (2) disconnected and weakly connected volumes can be separated apart; (3) the SDF of a primitive can also be interpreted as layers of isosurfaces, and thus sharing similar geometric structure; and furthermore (4) the size of the

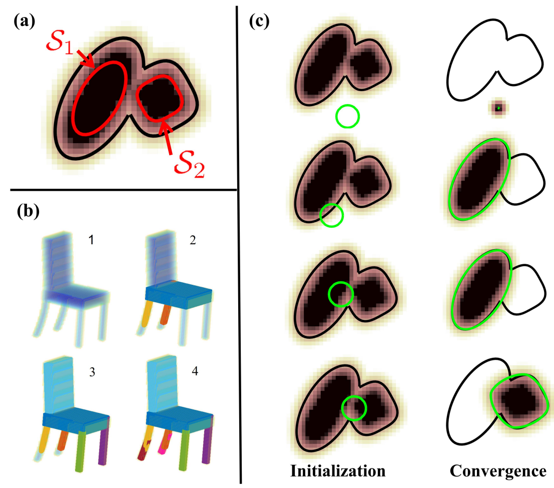

Figure 2. Visualizations of concepts. (a) A 2D illustration of two marched isosurfaces separating the shape into two VOIs. (b) A simple example of the Marching-Primitives algorithm. The first figure is the input target SDF (voxels with signed distances larger than the truncation threshold are left blank for visualization). Figures b2-b4 illustrate the primitives marched on the sequentially detected VOIs. (c) Auto-degeneration and probabilistic marching. The green circles on the left column indicate different initialization configurations. The ones on the right are the marched results.

volumes is large enough to provide sufficient geometric information for the primitive recovery. We illustrate the idea of isosurfaces in Fig.2(a). For each VOI, we initialize a primitive as an ellipsoid, with scales proportional to the size of its smallest bounding-box. The details for the primitive initialization are demonstrated in the Supplementary Material. The procedure of how a primitive marches from the initial guess to capture the local geometry is detailed in the next Sec. 3.4. After obtaining the primitive representations for all the VOIs in $\bar{S}_m$, we subtract the voxels

$$\{\mathbf{x}_i \in \mathbf{V}, d(\mathbf{x}_i) \leq 0 \wedge d_{\boldsymbol{\theta}}(\mathbf{x}_i) \leq 0\} \quad (7)$$

from the set of voxel grid $\mathbf{V}$. In other words, we deactivate the voxels which are interior of the target SDF and well fitted by a recovered primitive, while the updated SDF preserves the exterior and unfitted interior volumes. The connectivity marching repeats with the updated target SDF, until the threshold marches higher than a preset limit close to zero, indicating that no prominent interior volume is left unrepresented.

3.4. Probabilistic Primitive Marching

Firstly, we explain what we mean by primitive marching. In the marching cubes algorithm [25], marching means the progressive extraction of polygonal meshes from the neighboring 8 voxel points in the discrete SDF. In each step, the algorithm focuses within the single cube formulated by the 8 voxel points, calculates triangle vertices using linear interpolation, and then 'marches' to the next one. On the contrary, our method marches on the scope of voxels. That is, we gradually figure out a proper collection of voxel points from which a geometric primitive is extractable. Inspired by [24], we formulate this process as a maximum likelihood estimation problem with latent variables.

For each isosurface $\mathcal{S}_k \in \bar{S}_m$ obtained during the connectivity marching (Eq.(6)), it is assumed that the 'true' underlying primitive is encoded by $\boldsymbol{\theta}_k$, and that z_{ik} is the correspondence between the primitive $\boldsymbol{\theta}_k$ and a voxel point $\mathbf{x}_i \in \mathbf{V}$. z_{ik} is a binary variable that equals 1 when the target signed distance evaluated at the ith voxel point results from the kth primitive, and 0 otherwise. Then the target signed distance $d(\mathbf{x}_i)$ can be regarded as an observation generated from the probabilistic distribution

$$p(d_i|\boldsymbol{\theta}_k, z_{ik}) = \Big(\frac{\mathbb{1}_{d_i\in[-t,0)}}{t}\Big)^{1-z_{ik}} \mathcal{N}\big(d_i|d_{\boldsymbol{\theta}_k}(\mathbf{x}_i), \sigma^2\big)^{z_{ik}} \tag{8}$$

where d_i is short for $d(\mathbf{x}_i)$; t is the truncation value of the SDF; $\mathbb{1}_{d_i\in[-t,0)}$ is an indicator function which equals one when $d_i \in [-t, 0)$ and zero otherwise; $\mathcal{N}\big(\cdot\,|d_{\boldsymbol{\theta}_k}(\mathbf{x}_i), \sigma^2\big)$ is a Gaussian distribution with mean $d_{\boldsymbol{\theta}_k}(\mathbf{x}_i)$ and variance σ^2. Consequently, our goal is to find out the optimal $\boldsymbol{\theta}_k$ and the voxel-primitive correspondences z_{ik} simultaneously, which maximize the likelihood of the target SDF.

Correspondence Marching: We assume that z_{ik} follows a Bernoulli prior distribution $B(p_0)$, independent of $\boldsymbol{\theta}_k$. Then, given the target SDF d_i and the current primitive estimation $\boldsymbol{\theta}_k$, we are able to infer the posterior correspondence z_{ik} by the Bayes' rule

$$p(z_{ik}|\boldsymbol{\theta}_k, d_i) = \frac{p(d_i|\boldsymbol{\theta}_k, z_{ik})p(z_{ik})}{\sum_{z_{ik}\in\{0,1\}} p(d_i|\boldsymbol{\theta}_k, z_{ik})p(z_{ik})} \tag{9}$$

By analyzing the posterior correspondence, we can observe that our design of the probabilistic model (Eq.(8)) possesses two desirable properties: (1) voxels labeled as inside ($d_i < 0$) may not contribute to the current estimation of the primitive $\boldsymbol{\theta}_k$, *i.e.*, $p(z_{ik} = 1|\boldsymbol{\theta}_k, d_i < 0) \in (0, 1) \subset \mathbb{R}$; (2) all non-negative valued voxels always contribute to any primitives, *i.e.*, $p(z_{ik} = 1|\boldsymbol{\theta}_k, d_i \geq 0) = 1$. In other words, the primitive tries to absorb the interior voxels sharing similar local geometry, while avoids occupying any voxels labeled as the exterior. Meanwhile, the probabilistic correspondence establishes a soft relationship between the primitive and the interior voxels, allowing part of the primitive to occupy some interior volumes without complying with the target signed distance values. In this way, the algorithm is able to achieve an overall better fitting quality.

Primitive Update: Given the current estimation of the correspondences between the primitive and voxels, the parameter of the primitive is updated by

$$\boldsymbol{\theta}_k = \underset{\boldsymbol{\theta}_k}{\arg\min} \sum_{\mathbf{x}_i\in\mathbf{V}_a} P_{ik}\big\|d_{\boldsymbol{\theta}_k}(\mathbf{x_i}) - d_i\big\|_2^2 \tag{10}$$

where

$$\begin{aligned} &P_{ik} = p(z_{ik} = 1|\boldsymbol{\theta}_k^{prev}, d_i) \\ &\mathbf{V}_a = \big\{\mathbf{x_i} \in \mathbf{V}|d_{\boldsymbol{\theta}_k^{prev}}(\mathbf{x}_i) \in [-a, a] \subset \mathbb{R}\big\} \end{aligned} \tag{11}$$

$\boldsymbol{\theta}_k^{prev}$ is the previous estimation of the primitive parameters. $\mathbf{V}_a$ is an adaptive subset of the complete voxels $\mathbf{V}$, which includes the voxels close to the surface of the previous estimated primitive. a defines the distance threshold of the activated region. Only voxels in $\mathbf{V}_a$ are activated during the optimization. The reason why we use an adaptive subset instead of the complete voxel space is twofold. As discussed in Sec. 3.1, both the source and the target SDF are truncated within a vicinity of zero. Therefore, small variations around $\boldsymbol{\theta}_k^{prev}$ (*i.e.*, small changes on the shape of the primitive surface) have minor if not zero effects on the source SDF values evaluated at voxels distant to the primitive surface. Moreover, the size of $\mathbf{V}_a$ is much smaller than the complete set, and thus providing a significant boost in performance.

The correspondence marching and the primitive updating alternate until convergence. This process is akin to the EM algorithm [13]. The scope of voxels from which a primitive can be extracted is 'marched' with the progressive variation of the primitive shape, and simultaneously the parameters of the primitive are optimized based on the scope of voxels. We apply the optimization method proposed in [24] to avoid local minima. More derivation and implementation details can be found in the Supplementary Material.

3.5. Fail-safe Auto-degeneration

The primitives are roughly initialized in the connectivity marching step and further grow to capture local geometric shapes. Since the probabilistic marching is based on optimization, an important question to ask is if the recovered primitives can always converge to a proper shape, regardless of poor initialization. We demonstrate that our probabilistic marching strategy is robust to initialization. Firstly, our method possesses a feature called auto-degeneration-*i.e.* the primitive will autonomously degenerate towards a point when accidentally initialized far from the target volumes. This is because, under this circumstance, only variations which shrink the volume of the primitive can decrease the difference between the truncated target and source SDF. Secondly, when the primitive is initialized near or inside the target volumes, the marching process encourages the primitive to converge to one nearest local target shape by analyzing the posterior correspondences. Fig.2(c) illustrates the examples of the above properties. After the primitive marching, our algorithm removes the degenerated primitives. As a fail-safe measure, primitives which significantly contradict the target SDF is also removed. Detailed implementations can be found in the Supplementary Material.

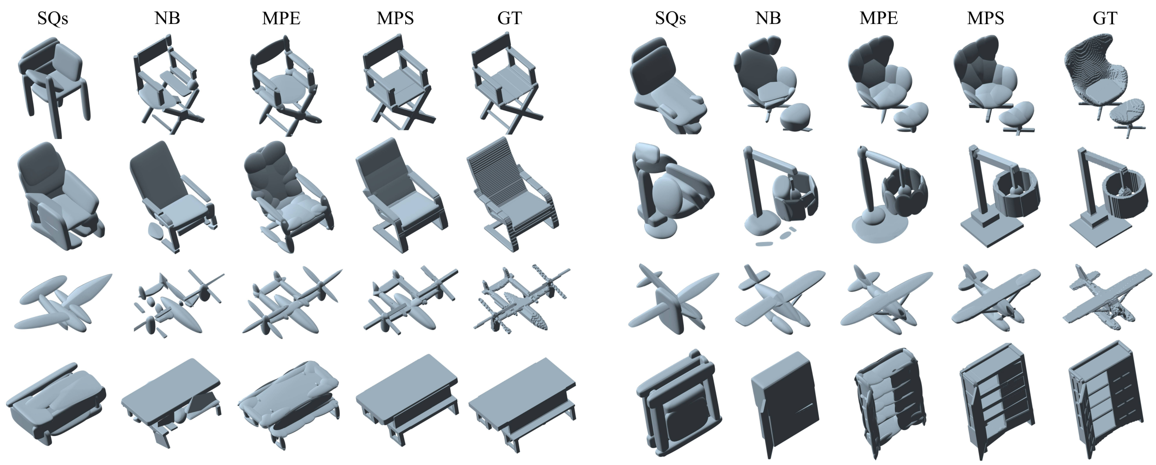

Figure 3. Shape Abstraction results on the ShapeNet dataset. From left to right: SQs [31], Non-parametric Bayesian (NB) [47], Marching-Primitives with ellipsoids (MPE), Marching-Primitives with superquadrics (MPS), and the ground truth (pre-processed watertight mesh).

4. Experiments

In this section, we conduct several experiments to demonstrate the high accuracy and generalizability of our proposed method. First, we show the shape abstraction results on various datasets, ranging from daily objects to human body models. Furthermore, we study the impact of the grid resolution and the truncation threshold on our algorithm. In the end, we apply our algorithm on a large-scale complex indoor scene, which contains a mixed combination of furniture such as sofas, tables, books, lamps, etc. Implementation details can be found in the Supplementary.

4.1. Evaluation on Datasets

Baselines: We compare our method with both the state-of-the-art learning-based method [31] and the computational method [47], which infer superquadric abstractions of the input objects. For convenience, we refer to [31] as SQs, and [47] as NB. We use the official codes and follow the implementation details as stated in these papers, respectively. We do not compare with the cuboid-based algorithms [40, 48, 50], since they focus more on the semantic level abstraction and are limited in expressiveness and accuracy. A detailed performance comparison between the cuboid and superquadric abstractions can be found in [31].

Datasets: We evaluate on two commonly used datasets, the ShapeNet [6] and the DFAUST [4]. In both of the datasets, we are only provided with triangular meshes. Therefore, we need to transform the meshes into discrete SDF representations. It is straightforward to calculate the signed distance value of a point to a watertight mesh. However, the ShapeNet is a human-made synthetic dataset containing many non-watertight meshes formed by non-volumetric 2D surfaces, self-intersecting or overlapped triangles. Therefore, we pre-process the original ShapeNet models into watertight meshes and then generate the SDF representation with the fast marching algorithm, following the same procedure in [8]. In consideration of fairness and consistency, we use the pre-processed watertight meshes as the common ground truth and the inputs to SQs and NB. For the DFAUST dataset, we generate the SDF directly since the provided meshes are already watertight.

Metrics: Following [31, 47] we use the Chamfer-L1 distance and the volumetric intersection over union (IoU) as the quantitative evaluation metrics. The computation of the metrics is detailed in the Supplementary Material.

Results on ShapeNet: We experiment on 14 categories from the ShapeNet dataset. We split the dataset randomly into the training (80%) and testing (20%) sets [9], where we train one SQs model per category on the training sets and evaluate all the methods on the testing sets. For our method, we use the SDF representation discretized on a voxel grid of size 100^3 and range $[-0.5, 0.5]^3$. Other than superquadrics, we also test our method with ellipsoids as the base primitive, which is a special case of the superquadric representation when the shape parameters ϵ_1, ϵ_2 are fixed to 1. The quantitative results are summarized in Table. 1. We also demonstrate the qualitative comparison among different shape abstraction methods in Fig.3. Our method outperforms all the baselines, even implemented with less expressive ellipsoids. The computational method generally performs better than the learning-based method. This is because the parameters of the primitive are so sparse and geometrically interrelated that they are difficult to get mapped from a high dimensional input by a neural network. Compared with NB, our method has richer details, clearer edges, and more importantly does not occupy the exterior space. Two factors contribute to the advantages. The first one is the extra geometric information embedded in the SDF representation, which eliminates the inherent interior/exterior ambiguity of point clouds. The other one is our special de-

Category	Chamfer-L_1 SQs [31]	NB [47]	MPE(Ours)	MPS(Ours)	IoU SQs [31]	NB [47]	MPE(Ours)	MPS(Ours)
airplane	0.037	0.023	0.021	**0.019**	0.441	0.671	0.731	**0.768**
bench	0.056	0.028	0.020	**0.020**	0.238	0.579	0.730	**0.819**
bottle	0.047	0.033	0.026	**0.017**	0.686	0.665	0.886	**0.924**
cabinet	0.059	0.036	0.037	**0.028**	0.394	0.666	0.840	**0.948**
can	0.066	0.036	0.036	**0.022**	0.706	0.553	0.908	**0.950**
chair	0.068	0.027	0.023	**0.020**	0.300	0.685	0.785	**0.871**
lamp	0.072	0.029	0.022	**0.021**	0.234	0.589	0.750	**0.802**
speaker	0.064	0.041	0.037	**0.033**	0.346	0.656	0.858	**0.920**
mailbox	0.095	0.026	0.026	**0.024**	0.333	0.694	0.802	**0.905**
rifle	0.038	0.020	0.019	**0.019**	0.446	0.732	0.744	**0.811**
sofa	0.054	0.037	0.029	**0.023**	0.497	0.726	0.857	**0.940**
table	0.070	0.024	0.024	**0.022**	0.247	0.745	0.818	**0.932**
phone	0.040	0.021	0.023	**0.021**	0.681	0.872	0.891	**0.947**
watercraft	0.048	0.032	0.022	**0.022**	0.465	0.618	0.793	**0.836**
mean	0.057	0.028	0.024	**0.022**	0.368	0.674	0.793	**0.870**

Table 1. Quantitative results on Shapenet. MPE and MPS are short for Marching-Primitives with ellipsoids and superquadrics, respectively

sign of the probabilistic generative model, as discussed in Sec.3.4. It is interesting to observe that our algorithm can extract abstractions from objects not intuitively depictable by the primitives.

Results on DFAUST: We also evaluate on the DFAUST dataset of human body models. We follow the split settings in [29] to train the learning-based method and evaluate all the methods on the testing set. The SDF is discretized on a grid of size 64^3 and side length 1. Similar to the ShapeNet experiment, we test our method with both superquadrics and ellipsoids. The results are shown in Fig.4. Our method can accurately capture various postures, while the baselines fail to distinguish different body parts in some cases. The abstraction quality of SQs is much better on this dataset compared with the ShapeNet, since human bodies share a common articulated structure that can be captured by the neural network. We observe that the ellipsoid abstraction also achieves satisfying accuracy. This is because the human body mostly consists of rounded shapes compared with man-made objects in the ShapeNet.

	SQs	NB	MPE (Ours)	MPS (Ours)
Chamfer-L_1	0.0435	0.0303	0.0301	**0.0291**
IoU	0.7417	0.8107	0.8568	**0.8598**

Figure 4. Shape abstraction results on the DFAUST dataset. From left to right: SQs [31], NB [47], MPE, MPS, and the ground truth.

4.2. Performance Study

This section investigates how the Marching-Primitives algorithm behaves with different voxel grid sizes and distance truncation thresholds. We experiment on the Armadillo from the Stanford 3D scanning repository [10]. First, we discretize the model on the voxel grids of different resolutions. The qualitative abstraction results are visualized in Fig. 5, and the quantitative results are summarized in Fig. 6(a). When the input grid resolution is low, the recovered primitive representation is relatively abstract. With the increase of the grid resolution, our method is able to extract detailed abstraction more faithful to the target shape. This is because the higher the resolution, the more geometric information our method can utilize to guide the primitive marching process. As discussed in Sec.3.1, we use a truncated version of the target and source SDF. Therefore, we also evaluate the performance on different truncation thresholds, starting from 0.1 to 4 times the interval of the voxel grids. The results are shown in Fig.6(b). The abstraction accuracy increases as the threshold decreases at first. The reason lies in that the approximated superquadric

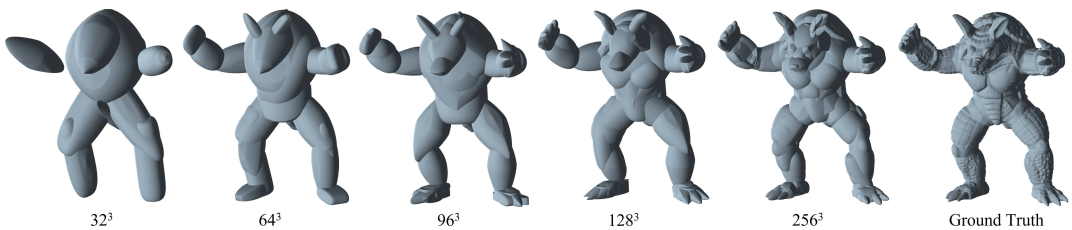

Figure 5. Visualization of the shape abstractions on different grid resolutions. Abstraction accuracy increases with the input grid resolution.

(source) SDF converges to the true value when truncated close to the surface. If we further decrease the threshold, however, we observe an acute decrease in accuracy. This is because an overly small truncation threshold corrupts the original geometric information embedded in the target SDF, as well as the smoothness of the cost function (Eq. (10)).

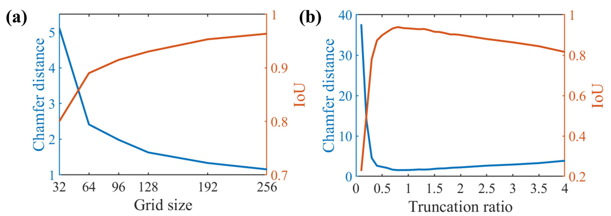

Figure 6. Quantitative results on shape abstraction accuracy. (a) Abstraction accuracy versus input grid resolution. (b) Abstraction accuracy versus truncation ratio, evaluated at grid resolution 128^3. We evaluate on two metrics: Chamfer-L_1 and IoU.

4.3. Scene Abstraction

Other than single objects, we also test the capability of our method in representing complex scenes with geometric primitives. We experiment on a real-world indoor scene called *Reading Room* [49]. The scene is captured by an Asus Xtion Pro Live camera. The RGB-D scans are fused utilizing [19] (an SDF-based 3D reconstruction algorithm), and further fine-tuned with the method proposed in [49]. However, the SDF representation of the scene is not available publicly. Therefore, we pre-process the scene mesh by removing the floor plane, filling the holes, and transforming it into the SDF representation with grid size 400^3. The scene abstraction task is much more difficult compared with the object-level task because various items with great differences in size and shape are present in the very same space. We compare our abstraction result with the mesh extracted by the marching cubes algorithm on the same discrete SDF representation, as shown in Fig.7. Our representation is not only visually satisfying but also contains abundant geometric information the mesh lacks, since it is a continuous SDF approximation to the input discrete SDF. Moreover, our highly sparse representation is only 8.2KB in size, while the discrete SDF occupies 203MB.

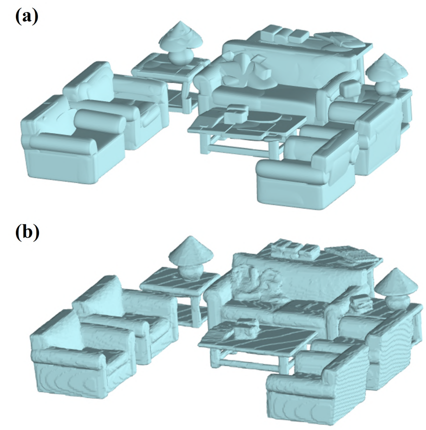

Figure 7. Qualitative result of the scene abstraction. (a) Superquadric abstraction by the Marching-Primitives. (b) Mesh generated by the marching cubes [25] from the same SDF.

5. Conclusion

We propose the Marching-Primitives, the first algorithm to extract primitive-based abstractions from the volumetric SDF representation. Our method outperforms the state-of-the-art in terms of accuracy and generalizability on both synthetic and real-world datasets. Our primitive-based representation is sparse, accurate, generalizable, and can be expressed analytically without training, which we believe will facilitate and inspire further explorations in scene understanding, 3D reconstruction, and robot motion planning. However, our algorithm has the limitation of not being able to properly extract object parts thinner than the interval of the voxel grid. This problem can be solved by either increasing the grid resolution or detecting and thickening the invalid parts. Future work includes the parallelization of the marching process and inferring semantic-level interpretations from the current primitive-based geometric features.

Acknowledgements This research is supported by NUS Startup grants A-0009059-02-00 and A-0009059-03-00, National Research Foundation, Singapore, under its Medium Sized Centre Programme - Centre for Advanced Robotics Technology Innovation (CARTIN), subaward A-0009428-08-00, and AME Programmatic Fund Project MARIO A-0008449-01-00, and JHU internal funds.

References

[1] R. Bajcsy and F. Solina. Three dimensional object representation revisited. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 231–240, 1987. 1

[2] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer graphics and Applications*, 1(1):11–23, 1981. 2, 3

[3] I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 1

[4] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6

[5] D. E. Breen, S. Mauch, and R. T. Whitaker. 3d scan conversion of csg models into distance volumes. In *Proceedings of the 1998 IEEE Symposium on Volume Visualization*, pages 7–14, 1998. 3

[6] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 6

[7] L. Chevalier, J. Jaillet, and A. Baskurt. Segmentation and superquadric modeling of 3D objects. In *The 11-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2003, WSCG 2003*, 2003. 1, 2

[8] G. Chou, I. Chugunov, and F. Heide. Gensdf: Two-stage learning of generalizable signed distance functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 6

[9] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 628–644. Springer International Publishing, 2016. 6

[10] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 303–312, 1996. 2, 7

[11] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2018. 2

[12] A. Dai, C. Ruizhongtai Qi, and M. Niessner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. 5

[14] B. Deng, K. Genova, S. Yazdani, S. Bouaziz, G. Hinton, and A. Tagliasacchi. Cvxnet: Learnable convex decomposition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 31–41, 2020. 1

[15] W. Dong, Q. Wang, X. Wang, and H. Zha. Psdf fusion: Probabilistic signed distance function for on-the-fly 3d data fusion and scene reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 701–717, 2018. 2

[16] K. Genova, F. Cole, D. Vlasic, A. Sarna, W. Freeman, and T. Funkhouser. Learning shape templates with structured implicit functions. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7153–7163, 2019. 1

[17] A. D. Gross and T. E. Boult. Error of fit measures for recovering parametric solids. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 690–694, 1988. 1, 2, 3

[18] Z. Hao, H. Averbuch-Elor, N. Snavely, and S. Belongie. Dualsdf: Semantic shape manipulation using a two-level representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7631–7641, 2020. 2, 3

[19] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 2, 8

[20] D. K. Katherine and S. S. Elizabeth. Core systems in human cognition. In *From Action to Cognition*, volume 164 of *Progress in Brain Research*, pages 257–264. Elsevier, 2007. 1

[21] Y. Kawana, Y. Mukuta, and T. Harada. Neural star domain as primitive representation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7875–7886. Curran Associates, Inc., 2020. 1

[22] A. Leonardis, A. Jaklic, and F. Solina. Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1289–1295, 1997. 1, 2

[23] Y. Liao, S. Donne, and A. Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2916–2925, 2018. 2

[24] W. Liu, Y. Wu, S. Ruan, and G. S. Chirikjian. Robust and accurate superquadric recovery: A probabilistic approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2676–2685, June 2022. 1, 2, 3, 5

[25] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 2, 3, 4, 8

[26] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352, 2015. 2

[27] C. Niu, J. Li, and K. Xu. Im2Struct: Recovering 3D shape structure from a single RGB image. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. 1

[28] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 2

[29] D. Paschalidou, L. V. Gool, and A. Geiger. 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)*, June 2020. 1, 3, 7

[30] D. Paschalidou, A. Katharopoulos, A. Geiger, and S. Fidler. 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)*, pages 3204–3215, June 2021. 1

[31] D. Paschalidou, A. O. Ulusoy, and A. Geiger. Superquadrics revisited: Learning 3D shape parsing beyond cuboids. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 3, 6, 7

[32] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331, 1986. 1, 2

[33] A. H. Quispe, B. Milville, M. A. Gutiérrez, C. Erdogan, M. Stilman, H. Christensen, and H. B. Amor. Exploiting symmetries and extrusions for grasping household objects. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3702–3708, 2015. 1

[34] Y. Rao, Y. Nie, and A. Dai. Patchcomplete: Learning multi-resolution patch priors for 3d shape completion on unseen categories. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[35] S. Ruan, K. L. Poblete, H. Wu, Q. Ma, and G. S. Chirikjian. Efficient path planning in narrow passages for robots with ellipsoidal components. *IEEE Transactions on Robotics (TRO)*, pages 1–18, 2022. 1

[36] S. Ruan, X. Wang, and G. S. Chirikjian. Collision detection for unions of convex bodies with smooth boundaries using closed-form contact space parameterization. *IEEE Robotics and Automation Letters (RAL)*, 7(4):9485–9492, 2022. 1

[37] G. Sharma, B. Dash, A. RoyChowdhury, M. Gadelha, M. Loizou, L. Cao, R. Wang, E. Learned-Miller, S. Maji, and E. Kalogerakis. Prifit: Learning to fit primitives improves few shot point cloud segmentation. In *Computer Graphics Forum*, volume 41, pages 39–50. Wiley Online Library, 2022. 3

[38] D. Smirnov, M. Fisher, V. G. Kim, R. Zhang, and J. Solomon. Deep parametric shape predictions using distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 561–570, 2020. 1

[39] F. Solina and R. Bajcsy. Recovery of parametric models from range images: the case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):131–147, 1990. 1, 2

[40] C. Sun, Q. Zou, X. Tong, and Y. Liu. Learning adaptive hierarchical cuboid abstractions of 3d shape collections. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019. 3, 6

[41] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 3

[42] N. Vaskevicius and A. Birk. Revisiting superquadric fitting: A numerically stable formulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):220–233, 2019. 3

[43] S. Vasu, N. Talabot, A. Lukoianov, P. Baqué, J. Donier, and P. Fua. Hybridsdf: Combining deep implicit shapes and geometric primitives for 3d shape representation and manipulation. In *International Conference on 3D Vision (3DV)*, 2022. 2

[44] G. Vezzani, U. Pattacini, and L. Natale. A grasping approach based on superquadric models. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1579–1586, 2017. 1

[45] G. Vezzani, U. Pattacini, G. Pasquale, and L. Natale. Improving superquadric modeling and grasping with prior on object shapes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6875–6882, 2018. 1

[46] S. Weder, J. L. Schönberger, M. Pollefeys, and M. R. Oswald. Routedfusion: Learning real-time depth map fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[47] Y. Wu, W. Liu, S. Ruan, and G. S. Chirikjian. Primitive-based shape abstraction via nonparametric bayesian inference. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 479–495. Springer Nature Switzerland, 2022. 1, 2, 3, 6, 7

[48] K. Yang and X. Chen. Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021. 1, 2, 3, 6

[49] Q. Zhou, S. Miller, and V. Koltun. Elastic fragments for dense scene reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 473–480, 2013. 2, 8

[50] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem. 3D-PRNN: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 6