3DIAS: 3D Shape Reconstruction with Implicit Algebraic Surfaces

Mohsen Yavartanoo* Jaeyoung Chung* Reyhaneh Neshatavar Kyoung Mu Lee
ASRI, Department of ECE, Seoul National University, Seoul, Korea
{myavartanoo,robot0321,reyhanehneshat,kyoungmu}@snu.ac.kr

Abstract

3D Shape representation has substantial effects on 3D shape reconstruction. Primitive-based representations approximate a 3D shape mainly by a set of simple implicit primitives, but the low geometrical complexity of the primitives limits the shape resolution. Moreover, setting a sufficient number of primitives for an arbitrary shape is challenging. To overcome these issues, we propose a constrained implicit algebraic surface as the primitive with few learnable coefficients and higher geometrical complexities and a deep neural network to produce these primitives. Our experiments demonstrate the superiorities of our method in terms of representation power compared to the state-of-the-art methods in single RGB image 3D shape reconstruction. Furthermore, we show that our method can semantically learn segments of 3D shapes in an unsupervised manner. The code is publicly available from this link.

1. Introduction

Single image 3D reconstruction is a procedure of capturing the structure and the surface of 3D shapes from single RGB images, which has various applications in computer vision, computer graphics, computer animation, and augmented reality. Recent advanced methods have substantially improved 3D shape reconstruction with the advent of deep neural networks (DNNs). These methods can be mainly categorized based on the representation of 3D shapes into explicit-based [4, 1, 28] and implicit-based [26, 10, 29, 12, 30] methods. Voxel-grid, as the most straightforward explicit representation, is useful in many applications. However, voxel-based methods generally suffer from large memory usage and quantization artifacts [37]. Polygon mesh [21, 41] has been introduced as alternative representation. However, since many polygon mesh-based methods start from a template mesh and deform it to reconstruct the target 3D shapes [41, 16], they can not produce 3D shapes with arbitrary topologies.

*equal contribution

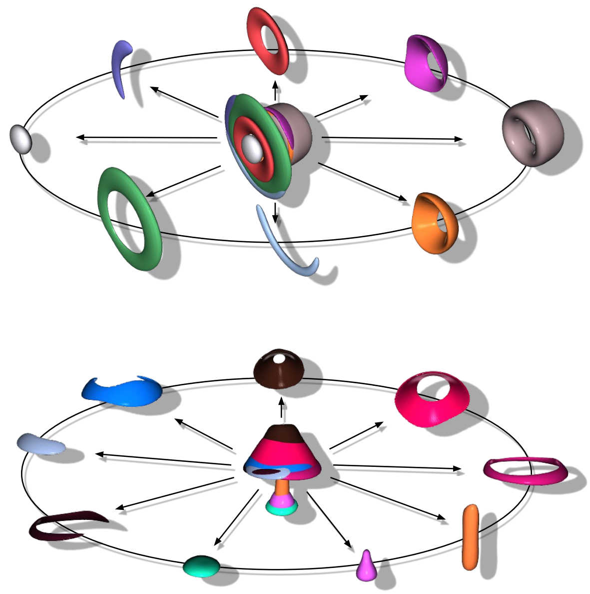
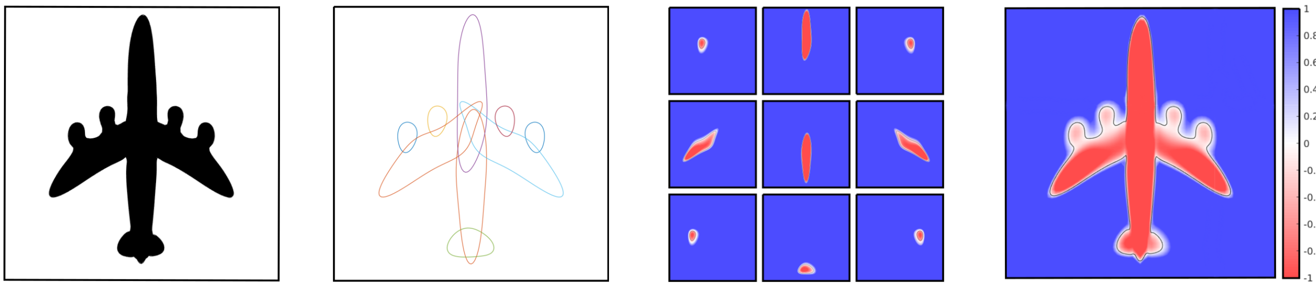


Figure 1: The exploded view of 3DIAS representation. The 3D shapes consist of a union over the proposed constrained implicit algebraic primitives with proper attributes.

On the other hand, implicit representations can approximate surfaces of 3D shapes as zero-sets of continuous functions in the Euclidean space. Recent implicit-based methods have shown some promises to reconstruct arbitrary shapes without any template. [29, 26, 40, 12, 30, 10] These methods can be categorized into two mainstreams; isosurface-based and primitive-based methods. Isosurface-based methods generally generate a surface by employing a neural network [29, 26] as an implicit function that assigns negative and positive values or different probabilities to the points lying inside and outside the shape. However, for each time visualization, these methods require all the neural network parameters to extract the zero-sets by determining the sign of many sample points in 3D space. Furthermore, these representations are unsuitable for computer graphics and virtual reality applications because they require additional postprocessing like marching cubes to generate the final 3D



(a) Target (b) Quartic polynomials (c) Collection of primitives (d) After union

Figure 2: Composition of implicit algebraic surfaces. Our network approximates the target shape as a union over a set of constrained implicit algebraic surfaces. The network estimates the coefficients and the center of polynomials. Note that since the level sets for each primitive are quite different, the final surface has non-uniform level sets as shown in (d).

shapes. Contrastingly, primitive-based methods approximate 3D shapes by a group of primitives such as cubes [40], ellipsoids [12], superquadrics [30], and convexes [10]. Despite their advantages in visualization and direct usage for various applications, the resolutions of reconstructed shapes are limited due to the simple topology (i.e., genus-zero) of the primitives. Consequently, approximating a 3D shape requires many of these simple primitives. Moreover, since the geometrical complexity varies from shape to shape, determining a sufficient number of primitives is challenging for an arbitrary shape.

In this paper, we propose a novel primitive-based 3D shape representation based on the learnable implicit algebraic surfaces named 3DIAS as shown in Figure 1. Since implicit algebraic surfaces have high degrees of freedom, they can describe complex shapes better than simple primitives [2]. Besides, identifying an implicit algebraic primitive is straightforward and depends on only a few parameters. We apply various constraints on these primitives to facilitate learning and achieve detailed appearances. We limit our primitives to the class of algebraically solvable implicit algebraic surfaces to assist fast 2D rendering and 3D visualization, which can be useful in many computer graphics applications. Furthermore, we develop an upper bound constraint with an efficient parameterization to guarantee that the primitives have closed surfaces and controlled sizes. Finally, we guide the primitives to cover different segments of a target shape by restricting the locations of their centers. To generate these primitives, we design a DNN-based encoder-decoder that captures the information of an observation (e.g., single image) and provides the parameters of the primitives. In our experiments, we show that our method outperforms state-of-the-art methods with most of the metrics. Moreover, we experimentally demonstrate that 3DIAS can semantically learn the components of 3D shapes without any supervision and adjust the number of primitives by excluding the primitives with empty volumes.

We summarize, our main **contributions** as follows:

- We propose a novel primitive-based 3D shape representation with the learnable implicit algebraic surfaces, which can produce more complex topologies with few parameters hence appropriate for describing geometrically complex shapes.
- We develop various constraints to produce solvable and closed primitives with proper scales in desired locations to ease learning and generate appealing results.
- We experimentally demonstrate that 3DIAS outperforms state-of-the-art methods. Furthermore, we show that it can semantically learn the components of 3D shapes and adjust the number of used primitives.

2. Related Work

In this section, we review some related DNN-based 3D shape reconstruction methods with various representations.
Explicit representations. A set of voxel is commonly used for discriminative [25, 32] and generative [9, 13] tasks since it is the most simple way in 3D representation. However, the results represented with voxel have a limitation in resolution due to memory issues. Although [17, 38] proposed to reconstruct 3D objects in a multi-scale fashion, they are still limited to comparably small 256^3 voxel grids and require multiple forward passes to generate final 3D voxels. 3D point clouds give an alternative representation of 3D shapes. Qi *et al*. [31, 33] pioneered point clouds as a discriminative representation for 3D tasks using deep learning. Fan *et al*. [11] introduced point clouds in 3D reconstruction task as an output representation. However, since point clouds have no information for connections between the points, it needs additional post-processing procedures [3, 5] to build the 3D mesh as an output. Mesh is another commonly used representation for 3D shape [14, 20]. However, most of

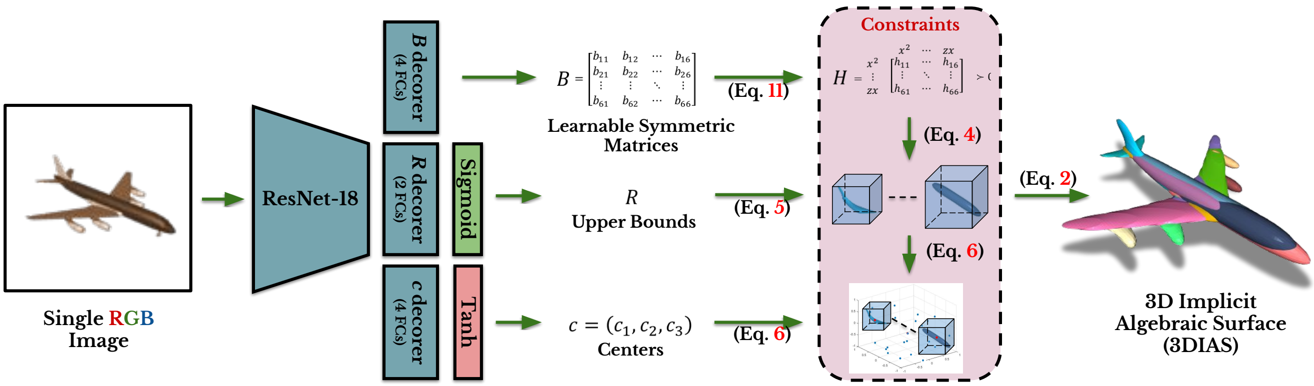


Figure 3: The overview of Single RGB 3D surface reconstruction using 3DIAS. We use an encoder (ResNet-18 [18]) to learn the local and global information from the given single RGB image. Then three sets of fully connected layers decode the latent features to provide the coefficients, the scales, and the location of centers for M primitives. Note that we apply $\min$ operator to take the union over the M primitives.

methods in 3D shape reconstruction using meshes generate meshes with simple topology [41] or utilize a template as a reference. They can only manage the objects from the same class [20, 24] and can not guarantee to produce closed surfaces [14].

Implicit Representations. Implicit representation is a good alternative to avoid the problems above. In contrast to the mesh-based approaches, implicit-based methods do not require a template from the same object class as input. There are mainly two approaches; isosurface-based and primitive-based methods. Chen *et al*. [8] propose a neural network that takes the 3D points and latent code of the shape, then outputs the values for each point, indicating whether the point is outside or inside the shape as an occupancy function. Park *et al*. [29] utilize the signed distance function to obtain the zero-set surface of the shape. Tatarchenko *et al*. [39] proposed the occupancy function that implicitly describes the 3D surfaces as the continuous decision boundary of a DNN-based classifier. Compared to voxel representation, this method can estimate an occupancy function continuous in 3D with a learnable neural network that can generate at any arbitrary resolution. This approach significantly decreases memory usage during training. The surface can be extracted as a mesh representation from the learned model at test time by a multi-resolution isosurface extraction method. The finite resolution of the voxel or octree cells limits the accuracy of the reconstructed shape by these methods and their ability to capture fine details of 3D shapes. Deng *et al*. [10] represented a shape with a convex combination of half-planes. These methods use a single global latent vector to represent entire surfaces of a 3D shape. The latent vector is decoded into continuous surfaces with the corresponding implicit networks. While this technique successfully models geometry, it often requires many primitives to obtain a desirable appearance, and it is unclear how many primitives are required.

For the 3D reconstruction task, we compare our implicit-based approach against several state-of-the-art implicit-based methods such as Structured Implicit Function (SIF) [12], OccNet [26], and CvxNet [10]. Moreover, we select Pixel2Mesh [41] and AtlasNet [15] , which use explicit surface generation in contrast to the previous methods.

3. 3DIAS

In this section, we first introduce our 3D shape representation based on implicit algebraic surfaces. Then we explain the additive constraints for effective learning. Next, we describe the proposed network and learning procedure to reconstruct the surface of 3D shapes with our representation.

3.1. 3D shape representation

3.1.1 Implicit algebraic primitives

We build a complex target 3D shape with a combination of primitives $p(x, y, z)$ that are the building blocks of 3D (i.e., basic geometric forms) as shown in Figure 1. To select a primitive with a large degree of freedom (i.e., complex geometry and topology) and few parameters, we employ the implicit algebraic surface that is a zero-level set of a multivariate polynomial function of x, y, and z as Eq. 1:

$$\begin{aligned} p(x,y,z) &= \sum_{0\leq i+j+k\leq d} a_{ijk} x^i y^j z^k \\ &= \boldsymbol{v} A \boldsymbol{v}^T = 0, \end{aligned} \tag{1}$$

where $\boldsymbol{v} = [1, x, y, z, x^2, y^2, z^2, \dots]$, and d, a_{ijk}, and A are the degree, the coefficients, and coefficient matrix of the polynomial function, respectively. Like many other implicit surfaces, the implicit algebraic surface divides the space and maps points in 3D space into negative and positive values. Therefore, to represent detailed surfaces $\mathcal{S}(x,y,z)$ of 3D shapes, we can combine these primitives by utilizing constructive solid geometry [19] and apply boolean operations to them, which can be formulated as Eq. 2:

$$\begin{aligned}\mathcal{S}(x,y,z) &= \bigcup_{m=1}^{M} p_m(x,y,z) \\ &= \min(p_1(x,y,z),\ldots,p_M(x,y,z)),\end{aligned} \tag{2}$$

where M is the number of primitives in the union.

3.1.2 Constraints on primitives

We apply a set of constraints on the defined implicit algebraic primitive $p(x,y,z)$ to better approximate the surface $\mathcal{S}(x,y,z)$ of a target 3D shape.

Solvability of primitives. Easy visualization and rendering attributes for a 3D shape representation can be useful in many computer graphics and virtual reality applications. The class of implicit algebraic primitives with algebraic solutions are appropriate representations for ray-tracing hence achieving these properties. Accordingly, we use multivariate quartic ($d=4$) polynomial functions as the primitives $p(x,y,z)$ because they have the highest degree of freedom among all implicit algebraic primitives with closed-form algebraic solutions [34].

Closedness and scales of primitives. Beyond the aforementioned constraint, we need to guarantee that the reconstructed shape and hence all primitives have closed surfaces as Figure 2. We can ensure that a quartic primitive has a closed surface by enforcing its fourth-degree terms to always be positive [22] as Eq. 3:

$$\begin{aligned}p^4(x,y,z) &= \sum_{i+j+k=4} a_{ijk} x^i y^j z^k \\ &= \boldsymbol{u} A_{[5:10]} \boldsymbol{u}^T > 0,\end{aligned} \tag{3}$$

where $\boldsymbol{u} = [x^2, y^2, z^2, xy, yz, zx]$ and $A_{[5:10]}$ is the 6×6 sub-matrix of A, including the coefficients of fourth-degree terms. This implies that $A_{[5:10]} \succ 0$ is a positive definite (PD) matrix. Note that, with the PD matrix $A_{[5:10]} \succ 0$, the algebraic surface exists if and only if $p^{3\downarrow}(x,y,z) = \sum_{0\le i+j+k\le 3} a_{ijk} x^i y^j z^k$ is negative and $|p^{3\downarrow}(x,y,z)| > |p^4(x,y,z)|$ for some points in $\mathbb{R}^3$. Otherwise, the primitive has zero volume because it has no real-valued solution.

Moreover, since each primitive reconstructs a different segment of a target 3D shape, we need to ensure that its volume is smaller than the target shape. To prevent generating large primitives and control their scales, we develop an upper bound for each primitive. To reconstruct a primitive $p(x,y,z)$ included in the upper bound $q(x,y,z)$, it is sufficient to satisfies the inequality $q(x,y,z) < p(x,y,z)$ in $\mathbb{R}^3$ which is also equivalent to Eq.4:

$$p(x,y,z) = h(x,y,z) + q(x,y,z), \tag{4}$$

where $h(x,y,z)$ is a positive-valued function. The function $h(x,y,z)$ is always positive if and only if its matrix of coefficients H be positive definite. As a result, the coefficient matrix A of primitive $p(x,y,z)$ is the summation of a positive definite matrix $H_{10\times 10}$ and the coefficient matrix $Q_{10\times 10}$ of the upper bound $q(x,y,z)$. For simplicity we consider the upper bound $q(x,y,z)$ as Eq.5:

$$q(x,y,z) = x^4 + y^4 + z^4 - R, \tag{5}$$

where R is a positive value that controls the size of the upper bound. Note that the developed upper bound is a more general constraint that also holds the criteria for the closedness constraint. For proof, refer to the supplementary materials.

Locations of primitives. We also encourage the primitives to better cover different components of 3D shape by applying a constraint on their locations. Accordingly, we restrict the locations of their centers $c=(c_1,c_2,c_3)$ into the areas nearby the shape and reformulate the primitives as Eq. 6:

$$p(x,y,z) = \sum_{0\le i+j+k\le 4} a_{ijk}(x-c_1)^i (y-c_2)^j (z-c_3)^k = 0. \tag{6}$$

Therefore, within these constraints, we can reconstruct primitives with controlled scales and locations, which facilitates the reconstruction and provides more details.

3.2. 3D shape reconstruction

To reconstruct a 3D shape with the proposed representation of an input observation $o \in \mathcal{X}$ (e.g., single image), we design a DNN architecture that receives the input and outputs the corresponding matrix H, center c, and parameter R for each primitive as shown in Figure 3.

3.2.1 Training losses

We apply various losses to reconstruct 3D shapes.

Loss sign. Since the target surface in 3D space divides the inside and outside, we define a sign function $\mathrm{sign}(x,y,z): \mathbb{R}^3 \longrightarrow \{0,-1,1\}$ on sample points $P \subset \mathbb{R}^3$ where the values 0, -1, and 1 correspond to the points on the target surface, its inside, and its outside, respectively. Likewise, we can classify the points on/inside/outside the reconstructed implicit surfaces $\mathcal{S}(x,y,z)$ and reduce a loss between their predicted and ground truth signs. We use mean square error (MSE) as the loss function as Eq. 7:

$$\mathcal{L}_{\mathcal{B}}^{sign} = \sum_{i\in\{on,in,out\}} \lambda_i \cdot \mathbb{E}_{\mathbf{p}_i\sim P} \|\tanh(\mathcal{S}(\mathbf{p}_i)) - \mathrm{sign}(\mathbf{p}_i)\|^2, \tag{7}$$

where $\mathcal{B} \subset \mathcal{X}$ and λ_i are a training batch and the weights corresponding to each sign, respectively. Note that $\mathcal{L}_{\mathcal{B}}^{sign}$ enforce the network to reconstruct the desired surface and refuse to generate the redundant surfaces simultaneously. Moreover, the MSE loss forces further attention on distinguishing the inside and outside points near the reconstructed surface because their $\tanh \mathcal{S}(\mathbf{p}_i)$ are near zero while their ground truths are -1 and $+1$, respectively.

Loss normal. To improve the reconstruction, we use the normal vectors as second-order information. Therefore, we define a MSE loss between the ground-truth normal vectors of the sample points $\mathbf{p}_{on}$ on the surface of a target mesh model $\mathbf{n}_\text{g}$ and their normal vectors $\mathbf{n}_\text{r}$ obtained by Eq. 8:

$$\mathcal{L}_{\mathcal{B}}^{n} = \mathbb{E}_{\mathbf{p}_{on} \sim P} \|\mathbf{n}_r(\mathbf{p}_{on}) - \mathbf{n}_g(\mathbf{p}_{on})\|^2, \tag{8}$$

where the normal vectors on the union surface can be directly determined for any point on the surface as Eq. 9:

$$\begin{aligned} \mathbf{n}_r =& \frac{\nabla \mathcal{S}(x,y,z)}{||\nabla \mathcal{S}(x,y,z)||} = \frac{(\frac{\partial \mathcal{S}}{\partial x}, \frac{\partial \mathcal{S}}{\partial x}, \frac{\partial \mathcal{S}}{\partial z})}{\sqrt{(\frac{\partial \mathcal{S}}{\partial x})^2 + (\frac{\partial \mathcal{S}}{\partial y})^2 + (\frac{\partial \mathcal{S}}{\partial z})^2}}, \\ \text{subject to:} \quad & \nabla \mathcal{S}(x,y,z) = \nabla p_{m^*}(x,y,z), \\ & m^* = \arg\min_m (p_m(x,y,z)). \end{aligned} \tag{9}$$

Note that m^* is the index of the closest primitive (i.e., the primitive with the smallest value $p_{m^*}(x,y,z)$) to the point.

Finally, the total loss $\mathcal{L}_{\mathcal{B}}^{\text{total}}$ is the weighting average of all defined losses with the corresponding weights as Eq. 10:

$$\mathcal{L}_{\mathcal{B}}^{\text{total}} = \mathcal{L}_{\mathcal{B}}^{\text{sign}} + \lambda_n \mathcal{L}_{\mathcal{B}}^{n}. \tag{10}$$

3.2.2 Implementation details

We consider the bounding box $\mathcal{C} = [-1,1]^3$ and fit the given input 3D shapes into it by keeping its aspect ratio. Then we extract $1M$ points from $[-1.1, 1.1]^3 \in \mathbb{R}^3$ surrounding the 3D shapes, and at each iteration, we randomly select 1% of them as jointly inside points $\mathbf{p}_{in}$ and outside points $\mathbf{p}_{out}$ for all shapes in the batch $\mathcal{B}$. Moreover, we pick $10k$ points $\mathbf{p}_{on}$ on the surface of each 3D shape and randomly select 20% of them at each iteration.

To capture the information of the input observation $o \in \mathcal{X}$ we employ the pretrained ResNet-18 [18] as the encoder. Then three sets of independent fully connected (FC) layers $(4096, 4096, 4096, 55 \times M), (1024, 512, 256, 3 \times M), (256, M)$ decodes the encoded features to obtain the parameters of each symmetric matrix B, scalar R and center c for $M = 100$ primitives in Eq. 2, respectively. All FC layers except the last layers are empowered by the $ReLU$ non-linear activation function. We also apply three batch normalization layers after the first three FC layers of B decoder to accelerate training and boost the performance.

To ensure H is a PD matrix, we parameterize it as Eq. 11:

$$H = BB^T + \alpha I \succ 0 \tag{11}$$

where $\alpha = 0.0001$ is a small scalar factor, and B and I are 10×10 symmetric and identity matrices, respectively. Moreover, we apply the $sigmoid$ function on the output of the R decoder to generate a value in $(0,1) \subset \mathbb{R}^3$ as the parameter R of Eq. 5 to control the size of all primitives and guarantee that they are not larger than the size of the bounding box $\mathcal{C}$. Furthermore, we apply $\tanh$ on the output of the c decoder to generate the centers inside the bounding box $\mathcal{C}$. Therefore, each primitive is parameterized with 59 parameters in total including three parameter for the center $c = (c_1, c_2, c_3)$, one parameter for the R, and 55 parameters for the matrix B.

We set the parameters λ_{on}, λ_{in}, λ_{out}, and λ_n as 2, 1, 10, and 1, respectively. We train our encoder-decoder architecture with Adam optimizer with the initial learning rate 1e-4, weight decay 1e-7, and batch size of 64. We implement our model in Python3.7 using PyTorch via CUDA instruction.

4. Experiments

In this section, we provide information about the evaluation setups and show qualitative and quantitative results of our method compared to state-of-the-art methods on single RGB image 3D shape reconstruction. We also perform various ablation studies to analyze our method better. More experiments are available in the supplementary material.

4.1. Dataset and Metrics

We evaluate our approach on the subset of the ShapeNet dataset [6] with the same image renderings and training/testing split provided by Choy *et al.* [9]. We also employ mesh-fusion [36] to generate watertight meshes from the 3D CAD models. We then use Houdini [35] to extract the inside/outside/on points and the normal vectors. For evaluation, we use the volumetric IoU, Chamfer [11], and F-Score [23] metrics. Volumetric IoU is used to measure the overlapped volume between the ground truth meshes and the reconstructed surfaces. Chamfer is the mean of the accuracy and the completeness score. The mean distance of points on the reconstructed surface to their nearest neighbors on the ground truth mesh is defined as the accuracy metric. The completeness metric is defined in the opposite direction of the accuracy metric. F-score is the harmonic mean of precision which shows the percentage of correctly reconstructed surfaces. To compute IoU, we sample $100k$ points from the bounding box. To evaluate Chamfer and F-score, we first transfer the reconstructed surfaces to meshes, then similar to CvxNet[10] we sample $100k$ points on the reconstructed and the ground-truth meshes.

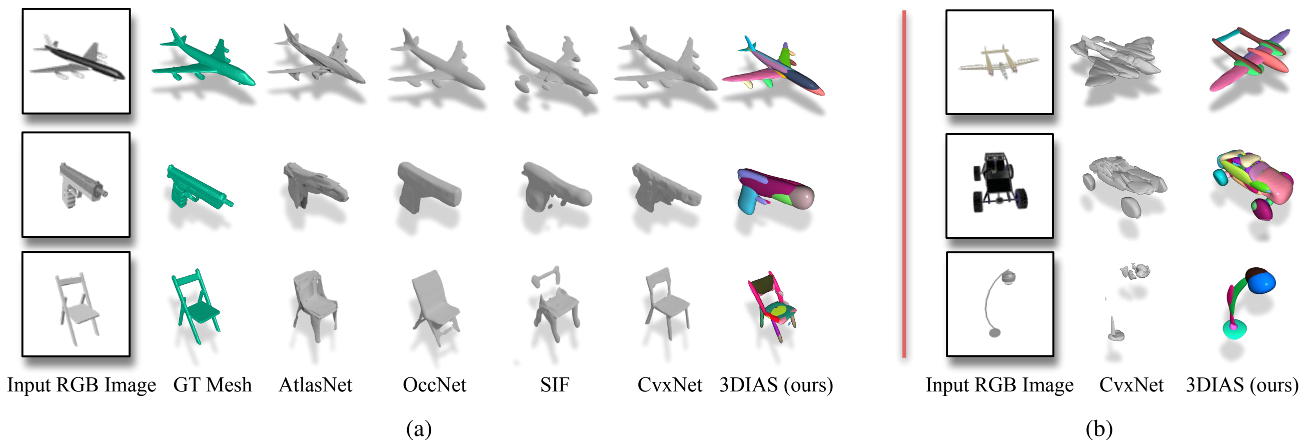


Figure 4: Qualitative comparison on single RGB image 3D shape reconstruction. SIF [12], AtlasNet [15], OccNet[26], CvxNet[10], and our 3DIAS output reconstructed 3D shape from the given RGB image. (a) Comparison with other methods for the samples shown in CvxNet [10]. (b) More qualitative comparisons with CvxNet [10].

Category	IoU						Chamfer						F-Score				
	P2M	AtlasNet	OccNet	SIF	CvxNet	3DIAS	P2M	AtlasNet	OccNet	SIF	CvxNet	3DIAS	AtlasNet	OccNet	SIF	CvxNet	3DIAS
airplane	0.420	-	0.571	0.530	**0.598**	0.549	0.187	0.104	0.147	0.167	0.093	**0.087**	67.24	62.87	52.81	**68.16**	59.48
bench	0.323	-	**0.485**	0.333	0.461	**0.485**	0.201	0.138	0.155	0.261	0.133	**0.106**	54.50	56.91	37.31	54.64	**60.17**
cabinet	0.664	-	**0.733**	0.648	0.709	0.730	0.196	0.175	0.167	0.233	0.160	**0.123**	46.43	61.79	31.68	46.09	**61.81**
car	0.552	-	**0.737**	0.657	0.675	**0.737**	0.180	0.141	0.159	0.161	0.103	**0.091**	51.51	56.91	37.66	47.33	**58.07**
chair	0.396	-	0.501	0.389	0.491	**0.509**	0.265	0.209	0.228	0.380	0.337	**0.186**	38.89	42.41	26.90	38.49	**43.14**
display	0.490	-	0.471	0.491	**0.576**	0.538	0.239	**0.198**	0.278	0.401	0.223	0.211	**42.79**	38.96	27.22	40.69	42.40
lamp	0.323	-	0.371	0.260	0.311	**0.381**	0.308	**0.305**	0.479	1.096	0.795	0.607	33.04	**38.35**	20.59	31.41	37.52
speaker	0.599	-	**0.647**	0.577	0.620	0.638	0.285	**0.245**	0.300	0.554	0.462	0.351	35.75	**42.48**	22.42	29.45	39.16
rifle	0.402	-	0.474	0.463	**0.515**	0.423	0.164	0.115	0.141	0.193	**0.106**	0.116	**64.22**	56.52	53.20	63.74	47.44
sofa	0.613	-	0.680	0.606	0.677	**0.685**	0.212	0.177	0.194	0.272	0.164	**0.158**	43.46	48.62	30.94	42.11	**49.73**
table	0.395	-	0.506	0.372	0.473	**0.509**	0.218	0.190	**0.189**	0.454	0.358	0.245	44.93	**58.49**	30.78	48.10	57.63
phone	0.661	-	0.720	0.658	0.719	**0.751**	0.149	0.128	0.140	0.159	0.083	**0.080**	58.85	66.09	45.61	59.64	**71.35**
vessel	0.397	-	0.530	0.502	**0.552**	0.538	0.212	**0.151**	0.218	0.208	0.173	0.206	**49.87**	42.37	36.04	45.88	40.70
mean	0.480	-	0.571	0.499	0.567	**0.575**	0.216	**0.175**	0.215	0.349	0.245	0.197	48.57	51.75	34.86	47.36	**52.22**

Table 1: Evaluation of single image 3D shape reconstruction. We evaluate and compare our method (3DIAS) to the state-of-the-art methods including P2M [41], AtlasNet [15], OccNet[26], SIF [12], and CvxNet[10] on a part of ShapeNet dataset [6] in terms of IoU, Chamfer, and F-score.

4.2. Reconstruction

We experimentally evaluate our method 3DIAS trained on multi-class and compare it with state-of-the-art methods on single RGB image 3D shape reconstruction and summarize the results in Table 1. The experiments demonstrate the superiority of 3DIAS compared to the explicit-based methods P2M [41] and AtlasNet [15], the isosurface-based method OccNet [26], and the recent primitive-based methods SIF [12] and CvxNet [10] in terms of volumetric IoU and F-score. We also achieve the second-best performance with the Chamfer metric. We show more quantitative results of 3DIAS for the trained network on single-class in the supplementary material.

Moreover, we qualitatively evaluate 3DIAS trained on single-class and compare it with the previous methods in Figure 4. The results illustrate that 3DIAS achieves smooth surfaces with desirable geometrical details. 3DIAS, unlike the previous methods, is successful in reconstructing the 3D shape with more complex topologies (e.g., chair) as shown in Figure 4a. Moreover, compared to CvxNet [10], 3DIAS can better reconstruct thin shapes (e.g., lamp) and when similar shapes are rare in the training dataset (e.g., airplane and car), see Figure 4b.

4.3. Ablation study

We perform several ablation studies to analyze our proposed representation and reconstruction procedure. First, we show the ability of our method to generate more complex primitives compared to other primitive-based methods. Then we compare the required number of parameters to represent 3D shapes with our representation and other methods. Finally, we demonstrate the power of our reconstruction

Representation	SIF	OccNet	CvxNet	3DIAS
Num of params	700	11M	7700	480

Table 2: Number of parameters. The average number of parameters for representing 3D shapes by different methods.

scheme to learn the semantic structures in an unsupervised manner. Moreover, we evaluate the effects of the designed constraints and the defined loss functions in reconstructing 3D shapes with high detailed appearances.

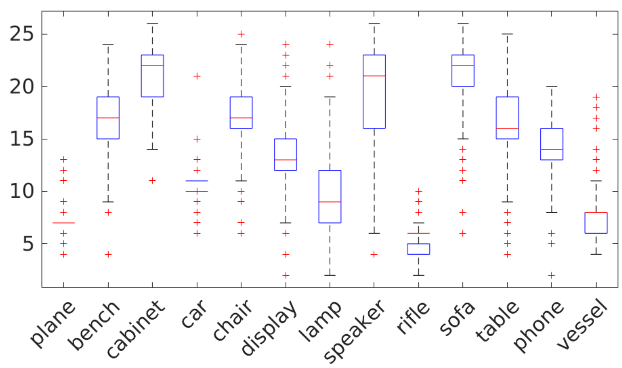


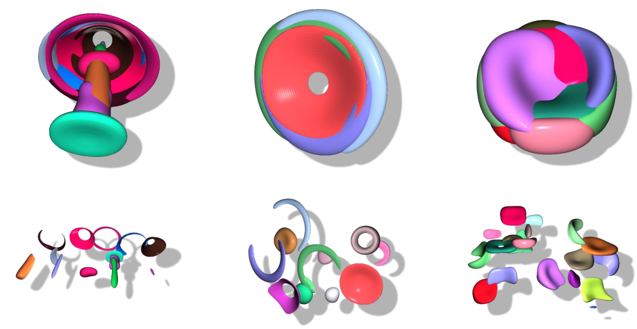
Figure 5: Statistics of the number of primitives. We compute the average number of primitives selected among all $M = 100$ primitives by the network for each category.

4.3.1 Complexity of primitives

We illustrate that our proposed constrained primitive is able to form more geometrical (e.g., curved) and topological (e.g., genus-one) complex shapes as shown in Figure 6. While the previous primitive-based methods such as cubes [40], ellipsoids [12], superquadrics [30], and convexes [10] cannot form such complex shapes.

4.3.2 The number of parameters

In section 3.1 we argue that a primitive may have no real solution when $|p^{3\downarrow}(x,y,z)| < |p^{4}(x,y,z)|$ or $|p^{3\downarrow}(x,y,z)|$ is non-negative for all points $(x,y,z) \in \mathbb{R}^3$ (i.e., no valid surface). Accordingly, our method can ignore some of the primitives among all $M = 100$ primitives by assigning positive definite coefficient matrices A to them and maintain a sufficient number of primitives. Therefore, these not-solvable primitives do not participate in reconstructing the surface $\mathcal{S}$. Note that, our method selects sets of primitives that are mainly different for inter-category shapes and have a large overlap for intra-category shapes. During the test phase, we can efficiently check the eigenvalues



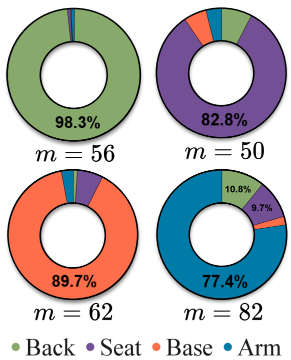
(a) Lamp (b) Speaker (c) Chair

Figure 6: The complexity of our primitives. The first and the second rows show the reconstructed shapes and their corresponding primitives, respectively. The proposed primitive can effectively present curved and torus shapes.

for the coefficient matrix of each primitive and eliminate the primitives with non-negative eigenvalues. Our experiments demonstrate that our network selects few primitives to reconstruct 3D shapes, as shown in Figure 5. In addition, since each quartic primitive can finally be identified with only 35 coefficients a_{ijk}, the surface $\mathcal{S}$ of 3D shapes with 3DIAS representation can be represented with only $35 \times 13.71 \simeq 480$ number of parameters on average. 3DIAS requires 68.571%, 0.004%, and 6.234% of the parameters used in SIF [12], OccNet [26], and CvxNet [10] on average to represent 3D shapes, respectively, see Table 2.

4.3.3 Unsupervised semantics segmentation

We also illustrate that our network learns a semantic structure without any part-level supervision such that one primitive usually covers the same part of reconstructed 3D shapes in the same class with 3DIAS representation. We evaluate the semantic structures on the PartNet[27] dataset having the labels of hierarchical parts of the shapeNet. The quantitative experiments in Figure 7 show that our method achieves better and comparable average accuracy compared to CvxNet [10] and BAE [7], respectively. In addition, 3DIAS achieves better accuracy than both methods for thin parts (e.g., arm). Moreover, our qualitative experiments illustrate that one primitive tends to cover the same semantic part as shown in Figure 8. This tendency is more pronounced for the dominant primitive that covers more points. For instance, the dominant primitives mainly cover the seat of chairs because most of the chairs have seat parts. Please see the supplementary material for more examples.



Part	Accuracy		
	BAE	CvxNet	3DIAS
Back	86.36	**91.50**	88.87
Seat	73.66	**90.63**	70.29
Base	**88.46**	71.95	78.51
Arm	65.75	38.94	**74.86**
mean	**78.56**	73.25	78.13

Figure 7: Evaluation of semantic segmentation. (left) The distribution of PartNet labels within 4 primitives in chair class. (right) The classification accuracy for each part. We follow the evaluation method introduced in cvxnet [10].

Constraints	IoU	Chamfer	F-Score
-center	0.549	0.387	46.72
-scale	0.559	0.261	48.45
-scale, -closedness	0.546	0.280	44.44
All	**0.575**	**0.197**	**52.22**

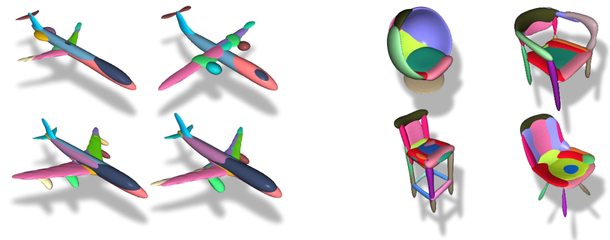
Table 3: Ablation study on constraints. We compare the effects of the center, the scale, and the closedness constraints in terms of IoU, Chamfer, and F-score. Note that in each configuration we ignore one or two constraints.

4.3.4 Effects of constraints

We study the effect of our designed constraints on reconstructing 3D shapes. In each experiment, we evaluate or baseline by ignoring one or more constraints. The quantitative results based on volumetric IoU, Chamfer, and F-Score show the importance of each constraint, see Table 3. We believe these constraints encourage the network to reconstruct a detailed 3D shape, especially the center constraint.

4.3.5 Effects of losses

While $\mathcal{L}^{sign}$ for the inside/outside points tries to distinguish inside and outside of 3D shapes, it is not enough to achieve a detailed surface due to the lack of sample points near the surface. Therefore, points on the surface and their normal vectors can facilitate the reconstruction. Note that normal vectors carry important information on 3D geometry, such as the local orientation of surfaces. Accordingly, we use $\mathcal{L}^{sign}$ loss and $\mathcal{L}^{n}$ loss for the points on the surface to better approximate the surfaces. We evaluate the effects of each $\mathcal{L}^{sign}$, $\mathcal{L}^{n}$, and their combination by excluding them for the points on the surface and summarize the results in Table 4. Note that for all the experiments in Table 4 we do not exclude $\mathcal{L}^{sign}$ for the inside/outside points. The results indicate the importance of points on the surface to achieve more detailed 3D shapes.



(a) Airplane (b) Chair

Figure 8: Qualitative results on unsupervised semantic segmentation. We visualize the results of 3DIAS for some samples in the categories of (a) airplane and (b) chair.

Losses	IoU	Chamfer	F-Score
-$\mathcal{L}^{n}$	0.568	0.210	49.17
-$\mathcal{L}^{sign}$	0.548	0.219	43.77
-$\mathcal{L}^{sign}$, -$\mathcal{L}^{n}$	0.542	0.232	42.37
All	**0.575**	**0.197**	**52.22**

Table 4: Ablation study on losses. We compare the effects of $\mathcal{L}^{sign}$ and $\mathcal{L}^{n}$ losses for the points on the surface in terms of IoU, Chamfer, and F-score. Note that in each configuration we ignore one or two loss functions. Moreover, we do not exclude the $\mathcal{L}^{sign}$ for the inside/outside points. Please see the supplementary material for more examples.

5. Conclusion

In this paper, we propose a primitive-based representation and a learning scheme in which the primitives are learnable implicit algebraic surfaces that can jointly approximate 3D shapes. We design various constraints and loss functions to achieve high-quality and detailed 3D shapes. We experimentally demonstrate that our method outperforms state-of-the-art methods in most of the metrics. Moreover, we illustrate that our method can learn semantic meanings without part-level supervision by automatically selecting sets of primitives parametrized by only a few parameters. In the future, we will utilize the solvability of the designed primitives to develop a soft renderer which leads to reconstruct the 3D shapes with self-supervised learning.

Acknowledgement

This work was supported in part by an IITP grant funded by the Korean government [No. 2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)].

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Representation learning and adversarial generation of 3d point clouds. *CoRR*, 2017. 1

[2] C. Bajaj. The emergence of algebraic curves and surfaces in geometric design. 1992. 2

[3] Fausto Bernardini, J. Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *Visualization and Computer Graphics, IEEE Transactions on*, 5:349 – 359, 11 1999. 2

[4] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016. 1

[5] F. Calakli and Gabriel Taubin. Ssd: Smooth signed distance surface reconstruction. *Computer Graphics Forum*, 30:1993 – 2002, 11 2011. 2

[6] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, 2015. 5, 6

[7] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. Bae-net: Branched autoencoder for shape co-segmentation. *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 7

[8] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 3

[9] Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. *CoRR*, 2016. 2, 5

[10] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31–44, 2020. 1, 2, 3, 5, 6, 7, 8

[11] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *CoRR*, 2016. 2, 5

[12] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas A. Funkhouser. Learning shape templates with structured implicit functions. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3, 6, 7

[13] Rohit Girdhar, David F. Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. *CoRR*, 2016. 2

[14] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[15] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation. *CoRR*, 2018. 3, 6

[16] X. Han, Hamid Laga, and M. Bennamoun. Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1

[17] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. *CoRR*, 2017. 2

[18] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 5

[19] John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, and Kurt Akeley. *Computer Graphics - Principles and Practice, 3rd Edition*. Addison-Wesley, 2014. 4

[20] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. *CoRR*, 2017. 2, 3

[21] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. *CoRR*, 2018. 1

[22] D. Keren, D. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):38–53, 1994. 4

[23] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4), July 2017. 5

[24] C. Kong, C. Lin, and S. Lucey. Using locally corresponding cad models for dense 3d reconstructions from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5603–5611, 2017. 3

[25] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 2

[26] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. *CoRR*, 2018. 1, 3, 6, 7

[27] Kaichun Mo, Shilin Zhu, Angel X. Chang, L. Yi, Subarna Tripathi, L. Guibas, and H. Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7

[28] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. *CoRR*, 2016. 1

[29] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 3

[30] Despoina Paschalidou, Ali O. Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10336–10345, 2019. 1, 2, 7

[31] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, 2016. 2

[32] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. *CoRR*, 2016. 2

[33] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, 2017. 2

[34] Michael I. Rosen. Niels hendrik abel and equations of the fifth degree. *The American Mathematical Monthly*, 102(6):495–505, 1995. 4

[35] SideFX. Houdini, 2020. 5

[36] David Stutz and Andreas Geiger. Learning 3d shape completion under weak supervision. *CoRR*, 2018. 5

[37] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[38] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. pages 2107–2115, 10 2017. 2

[39] Maxim Tatarchenko, Stephan R. Richter, Rene Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[40] Shubham Tulsiani, H. Su, L. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 7

[41] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single RGB images. *CoRR*, 2018. 1, 3, 6