PrimitiveNet: Primitive Instance Segmentation with Local Primitive Embedding under Adversarial Metric

Jingwei Huang[1], Yanfeng Zhang[1], and Mingwei Sun[1,2]

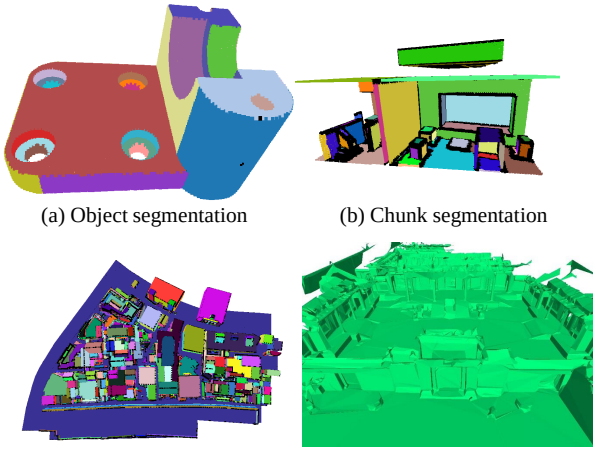
[1]Riemann Lab, Huawei Technologies. [2]Wuhan University

Abstract

We present PrimitiveNet, a novel approach for high-resolution primitive instance segmentation from point clouds on a large scale. Our key idea is to transform the global segmentation problem into easier local tasks. We train a high-resolution primitive embedding network to predict explicit geometry features and implicit latent features for each point. The embedding is jointly trained with an adversarial network as a primitive discriminator to decide whether points are from the same primitive instance in local neighborhoods. Such local supervision encourages the learned embedding and discriminator to describe local surface properties and robustly distinguish different instances. At inference time, network predictions are followed by a region growing method to finalize the segmentation. Experiments show that our method outperforms existing state-of-the-arts based on mean average precision by a significant margin (46.3%) on ABC dataset [31]. We can process extremely large real scenes covering more than 0.1km^2. Ablation studies highlight the contribution of our core designs. Finally, our method can improve geometry processing algorithms to abstract scans as lightweight models. Code and data will be available based on Pytorch[1] and Mindspore[2].

1. Introduction

3D scanning techniques have made rapid advances in recent years with 3D sensors. State-of-the-art algorithms [47, 28, 65, 12] or commercial softwares [1] ease the reconstruction and digitization of the real-world environments. However the quality and the complexity of the output model are below the required standards for target applications including gaming and virtual/augmented reality (AR/VR). For example, a typical indoor scan contains several million faces filled with noises, which is not affordable for a cell phone. Main directions to address these issues include local mesh decimation [18, 37, 52] and primitive instance assembly [5, 32, 23, 2]. Mesh decimation collapses edges iteratively but fails to preserve important structures. Primitive assembly requires segmenting points into instances of primitives and thus is limited by the segmentation quality. We aim to significantly improve the segmentation quality and the final production of lightweight models from scans.



(a) Object segmentation (b) Chunk segmentation

(c) Scene segmentation (d) Scene abstraction

Figure 1. We propose PrimitiveNet to robustly segment primitive instances at the level of (a) objects or (b) chunks of scenes. (c) We can handle extremely large scenes covering 0.1km^2. (d) We improve scene abstraction and deliver lightweight models.

Primitive instance segmentation has a long history in geometry processing with two standard solutions using Ransac [53] or region growing [41, 51]. The main challenge is to find appropriate parameters to robustly recover shapes from noises and robustly preserve boundaries of similar primitives. Recently, this problem is partially addressed using deep learning techniques [35, 57, 39] at the object level. However, they require to extract global shape properties and have limited capacity for correctly predicting small

[1]https://github.com/hjwdzh/PrimitiveNet
[2]https://gitee.com/mindspore/mindspore/tree/master/model_zoo/research/3d/PrimitiveNet

instances or processing point clouds on a large scale.

To address these limitations, we transform the global primitive fitting problem into local tasks that are easier to learn and generalize, which robustly distinguish different instances and derive high-quality segmentation at different scales (Figure 1 (a-c)). We train a primitive embedding network that focuses on learning per-point local surface properties including both explicit geometry features and implicit latent features. One popular choice of explicit feature is the object center, which proves to be effective for accurate semantic instance segmentation[29, 15, 22] with post clustering. However, it is not suitable for primitive segmentation: While object scale is relatively local, sizes of primitives like floor planes can be large enough to cover the whole scene. Further, centers can be shared among different primitives and thus not a discriminative feature for clustering. Instead, we design explicit features for a point as its local tangent plane supervised by the location with a normal direction on the ground truth shape nearest to the point. We use latent features to distinguish primitive instances. We supervise features with primitive types if available during training. Since it is insufficient to distinguish instances with the same type, we additionally train an adversarial metric as a primitive discriminator to decide whether two latent features indicate different instances. To encourage latent features to capture local properties, we constrain the primitive discriminator to evaluate features of closed points. We find such local constraint highlights feature differences at boundaries and is robust for supervision.

Our design of primitive embedding network combines PointNet [49] and sparse convolution [20, 7] to extract high-resolution point features locally and regular volume features within a larger receptive field. They are concatenated and passed through multi-layer perceptrons to derive per-point explicit and implicit surface features, where implicit features are translated as primitive type scores via a linear layer. Explicit features and primitive scores are supervised by ground truth data. To enforce primitive discriminator and implicit features to be local, we sample implicit features from only pairs of closed points below a certain distance threshold for discriminator input.

Experiments show that we significantly outperform existing methods at ABC dataset [31] and self-collected scene dataset under several metrics. Notably, we outperform state-of-the-art methods under mean average precision by 46.3% on the ABC dataset. We handle extremely large scenes by processing chunks and merging them seamlessly. Ablation studies show that local properties are critical to the performance, and our high-resolution backbone further improves the prediction. Our explicit property supervision helps to increase robustness for different levels of noise. Finally, we integrate our approach into a robust pipeline to abstract scanned point clouds as light-weight models.

In sum, our core research contributions are:

- We design an adversarial primitive embedding network that learns discriminative local surface properties.
- We propose a high-resolution backbone combining point and voxel features.
- Our design significantly outperforms the state-of-the-arts and can handle extremely large-scale environment.
- We integrate our algorithm to a pipeline that produces lightweight models from real scans.

2. Related Works

Primitive fitting Primitive fitting of point clouds is the process of clustering input points and fitting them with explicit parametric models. Two common directions are through Ransac [9, 17, 30, 8, 59] and region growing [41, 51]. Ransac iteratively estimates parametric models to fit inliers, where a robust Ransac framework [53] is available. Region growing methods fit local primitives and propagate the hypothesis until fitting error is unaccepted. [36, 48, 44, 25] further refine or regularize the primitive geometry based on additional assumptions. These methods usually suffer from low prediction accuracy caused by noises or complex surface structures.

To address these issues, supervised [68] and unsupervised [60, 56, 16] learning-based primitive fitting algorithm have been proposed. However, fitting accuracy is still unsatisfactory and [68, 60, 16] only support cuboid fitting. [35] classifies points into a finite number of instances. [57] uses triplet loss with post clustering steps to segment instances. These methods extract global features for objects and are hard to extend to the scene level. [39] learns the boundaries where final segmentation is sensible to wrong predictions. Our network design focuses on local properties that recover segmentation and supports large-scale and accurate primitive segmentation.

Primitive fitting is an important step for holistic 3D reconstruction from scans required by shape assembly [5, 32, 23], space slicing [45, 4, 2] or outer-hull abstraction [26, 43]. While these methods extract the primitive instances according to geometry heuristics, our approach can replace these parts by providing learned instances. We demonstrate that our result benefits [43] to generate clean and light-weight models.

Semantic instance segmentation Treating primitive types as semantic classes, our task is similar to semantic instance segmentation where scene-level solutions are available [67, 24, 33, 46, 66]. While object center is one effective feature to learn [29, 15, 22], primitive centers can be shared by different instances and thus not a discriminative feature. Our network learns discriminative and latent features reflecting local surface properties guided by an adversarial metric.

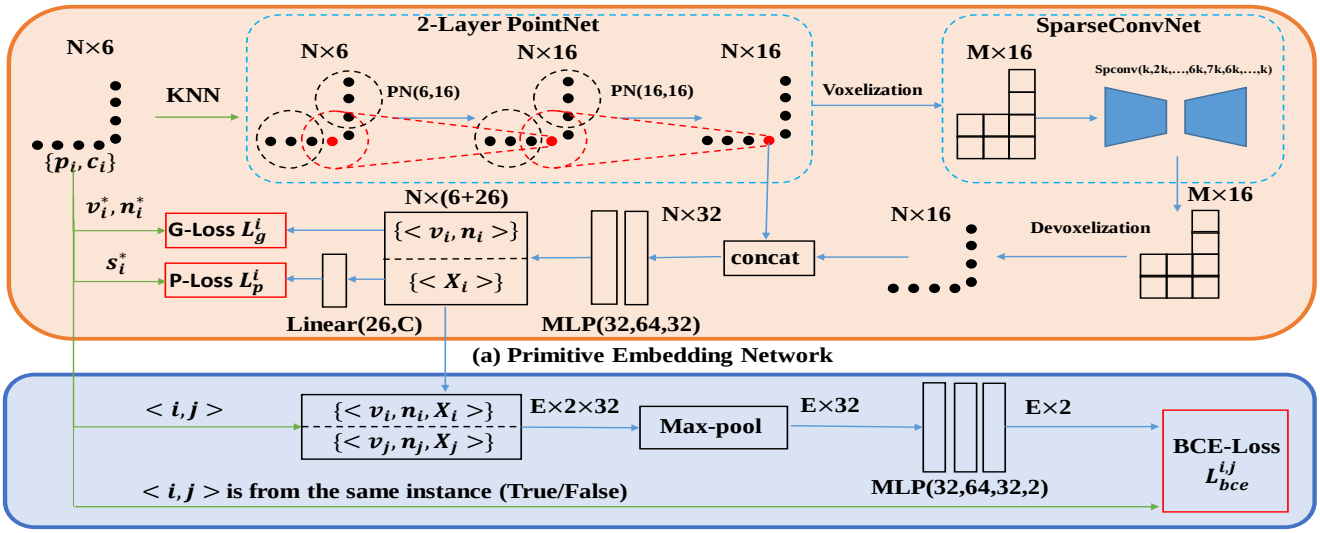


Figure 2. PrimitiveNet architecture. We jointly train (a) a primitive embedding network and (b) a discriminator network. We combine PointNet [49] and SparseConvNet [20] to extract high-resolution features, where explicit and implicit features are supervised by ground truth tangent space (G-Loss) and primitive types (P-Loss), and jointly trained with the discriminator using binary cross-entropy (BCE).

3D backbone 3D feature extraction backbone has been well-studied for point-based networks [49, 50, 64] and voxel-based networks [42, 20, 7]. While sparse convolution [20, 7] is one of the most effective solution for semantic segmentation at limited resolution, point networks accurately describe features at point locations. [38, 58] address the resolution issue by concatenating point coordinate and voxel features. We further improve it by integrating PointNet [49] features into sparse convolutions [20] to provide high-resolution point features.

Metric Learning Contrastive loss [21] and triplet loss [54] are two widely used loss functions for deep metric learning. Accordingly, a Mahalanobis distance can be learned to measure similarities of samples [13, 19, 55, 62]. Deep learning approaches have been proposed to learn non-linear mappings [10, 40, 61, 63]. Recent approaches [14, 6, 27] further explores adversarial network as learned metrics. We introduce such an adversarial metric into the primitive segmentation task.

3. Approach

3.1. Overview

The input to our problem is $< \mathcal{P}, \mathcal{C}, \mathcal{E} >$ as a set of points $\mathcal{P} = \{\mathbf{p}_i\}$ optionally with point features $\mathcal{C} = \{\mathbf{c}_i\}$ including point colors or normals, and an edge set $\mathcal{E} = \{< i, j >\}$ denoting $\mathbf{p}_i$ and $\mathbf{p}_j$ are neighbors. Our goal is to produce $< \mathcal{V}, \mathcal{N}, \mathbf{L} >$ as a noise-free point cloud $\mathcal{V} = \{\mathbf{v}_i\}$ with per-point surface normal $\mathcal{N} = \{\mathbf{n}_i\}$ and instance label $\mathbf{L} = \{l_i\}$, where points with the same label belong to the same primitive instance. For specific applications requiring concrete primitive parameters, we produce primitive type classification scores $\mathbf{S} = \{\mathbf{s}_i\}$ for each instance where parameters can be directly computed through primitive fitting given point locations and normals. Instead of directly computing instance labels as a function problem, we convert it as a decision problem to decide whether adjacent points belong to the same instance. As a result, the subset of $\mathcal{E}$ within the same instances forms a graph where each connected component represents a primitive instance. Accordingly, we propose our solution as a high-resolution primitive embedding network (Figure 2(a)) and a primitive discriminator network (Figure 2(b)). The embedding network takes input as a point cloud with per-point locations and features and output per-point features reflecting local surface properties. The discriminator takes inputs as pairs of features and produces a 2-dimensional vector for each pair indicating whether they are from the same instance.

We discuss the embedding network in Section 3.2 and primitive discriminator in Section 3.3. Section 3.4 describes the details for training and generation of final segmentation during inference time.

3.2. High-resolution Primitive Embedding

The role of the embedding network is to extract point features reflecting local surface properties.

$$\mathcal{Y} = f_e(\mathcal{P}, \mathcal{C}; \theta_e) \tag{1}$$

The network is a function f_e with parameters θ_e (Equation 1) that maps points $\mathcal{P}$ and their features $\mathcal{C}$ to higher-dimensional features $\mathcal{Y}$ where $\mathbf{y}_i$ reflects the local surface

property for point $\mathbf{p}_i$. Specifically, $\mathbf{y}_i$ is composed of explicit geometry feature ($\mathbf{v}_i$ and $\mathbf{n}_i$) and implicit latent feature ($\mathbf{X}_i$). We set the explicit geometry feature as the local tangent space around $\mathbf{p}_i$. Specifically, we define the local tangent space of p_i as its nearest location $\mathbf{v}_i$ at the ground truth shape with its surface normal $\mathbf{n}_i$. The design purpose of such explicit features is to regularize the network to predict local properties under noises. Accordingly, the explicit feature is supervised by ground truth data $\mathbf{v}^*$ and $\mathbf{n}^*$ using geometry loss (G-Loss in Figure 2(a)) in Equation 2, where ϵ represents the scale of noise related to the data.

$$\mathcal{L}_g^i = \frac{1}{\epsilon^2}||\mathbf{v}_i - \mathbf{v}_i^*||_2^2 + ||\mathbf{n}_i - \mathbf{n}_i^*||_2^2 \tag{2}$$

Implicit feature aims to encode rich and more complex local surface properties as a high-dimensional vector. If primitive types are provided in the training data, we optionally pass implicit feature $\mathbf{X}_i$ to a linear layer f_s (Equation 3)

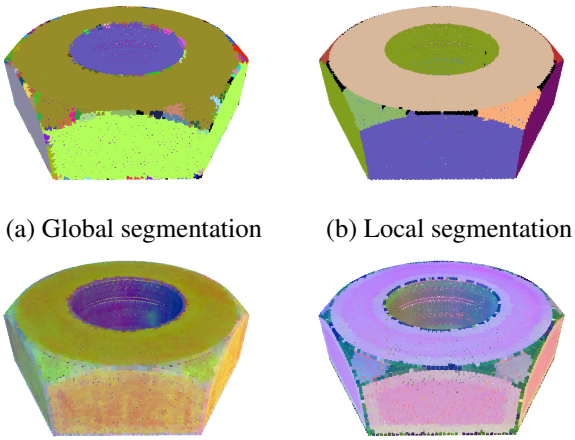
$$\mathbf{s}_i = f_s(\mathbf{X}_i; \theta_s) \tag{3}$$

to produce scores $\mathbf{s}_i$ for each of k primitive types. We treat ground truth label $\mathbf{s}_i^*$ as a one hot vector to supervise $\mathbf{X}_i$ with a primitive type loss $\mathcal{L}_s^i$ (P-Loss in Figure 2(b)) using cross entropy. It can be further used by the primitive discriminators (Section 3.3) to distinguish different instances.

One contribution in our network architecture design is the high-resolution embedding network f_e, which combines PointNet [49] and sparse convolution [20] to provide high-resolution point features. Similar ideas have been proposed by [38, 58] where point features are acquired from trilinear interpolation of voxel features followed by MLP layers. However, we find such interpolation is insufficient to describe high-resolution features in our task. We pass points with their local neighbors into a 2-Layer PointNet [49] (Figure 2(a)) to describe features at point resolution. Inside each PointNet block, point features are passed through an MLP layer and max-pooled with features of $k = 16$ nearest neighbor points. The features are averaged into belonging voxels and passed through UNet structure [34] using sparse convolutions [20]. Voxel features are copied back to points and concatenated with PointNet features. Finally, we pass it through an MLP layer to describe $\mathbf{y}_i =< \mathbf{v}_i, \mathbf{n}_i, \mathbf{X}_i >$.

3.3. Primitive Discriminator

The role of primitive discriminator $\mathcal{D}(\mathbf{y}_i, \mathbf{y}_j; \theta_b)$ is to decide whether two features $\mathbf{y}_i$ and $\mathbf{y}_j$ belong to the same primitive instance. The network architecture is shown in Figure 2(b). Pairs of features $\mathbf{y}_i$ and $\mathbf{y}_j$ are selected via edge set $\mathcal{E}$ and aggregated by max-pooling. The fused features are passed through MLP layers to generate two-dimensional scores $\mathbf{b}_{i,j}$, denoting whether points i and j belong to different instances. This is supervised by the ground truth instance labels $\{l_i^*\}$ using binary cross-entropy as shown in Equation 4.



(a) Global segmentation (b) Local segmentation

(c) Global feature (d) Local feature

Figure 3. (a) Segmentation with globally sampled pairs for supervision. (b) Segmentation with adjacent vertices in the mesh for supervision. (c) Feature visualization of three most salient dimensions for (a). (d) Feature visualization of three most salient dimensions for (b).

$$\mathcal{L}_{bce}^{(i,j)} = \text{CrossEntropy}(\mathbf{b}_{i,j}, l_i^* \neq l_j^*) \tag{4}$$

$\mathbf{b}_{i,j}^*$ is a binary value depending on whether l_i equals l_j.

During training, the discriminator evaluated each pair of points inside an edge set $\mathcal{E}$. The choice for edges is non-trivial since it leads to different feature interpretations. For example, we could randomly pick pairs of points to form $\mathcal{E}$, where discriminator and implicit feature are trained at a global scale. An alternative choice is to define $\mathcal{E}$ as the edges in the triangle mesh, where only closed points in local neighborhoods are evaluated by the discriminator. Figure 3(a,b) shows the segmentation where the discriminator is supervised with global or local pairs. As a result, local supervision is clearly better. Figure 3(c,d) visualize three most salient dimensions of implicit features. While global supervision yields even colors inside the same primitive with smooth change across different primitives, local supervision encourages features to highlight local boundaries and are more robust for segmentation. Therefore, we construct $\mathcal{E}$ to sample from local neighborhoods, which is helpful to train a primitive discriminator together with embedded features to capture local surface properties. If the input is given as a triangle mesh, we directly extract the edge set from triangle edges. Otherwise we generate $k = 16$ nearest neighbors for each point to form $\mathcal{E}$.

3.4. Implementation

During training, we accept the dataset with or without primitive type labeling. If primitive type information is given, we optimize the network parameters (θ_e,θ_s,θ_b) as

shown in Equation 5.

$$\mathcal{L} = \sum_{i=1}^{N}(\mathcal{L}_g^i + \mathcal{L}_s^i) + \sum_{<i,j>\in\mathcal{E}} \mathcal{L}_{bce}^{(i,j)} \tag{5}$$

If primitive type is not labeled, the linear layer f_s and loss $\mathcal{L}_s^i$ are omitted. For object datasets where points are normalized and bounded in $[-1, 1]$, we set $\epsilon = 0.1$. For a real scene dataset with units in meters, we set $\epsilon = 0.05$.

During inference, we obtain predictions of the cleaned point cloud $\mathcal{V}$ and $\mathcal{N}$ and optionally a per-point primitive type for each point. We rely on the primitive discriminator network to decide whether each pair of points inside $\mathcal{E}$ belongs to the same instance. Ideally, we can collect edges belonging to the same instances in $\mathcal{E}$ to form a graph, where each connected component is a primitive instance. However, any wrong prediction of $\mathbf{b}_{i,j}$ across different instance would lead to incorrect merge of l_i and l_j. Therefore, we use a region-growing method to conservatively grow the region for each instance. When growing the region, we ensure that newly-added points are not conflicting with any points in the current region. Details of the implementation are shown in Algorithm 1. If primitive types are trained, we simply set the primitive type of an instance as the one with maximum overlap with per-point prediction in this instance.

Algorithm 1: Region growing-based primitive instance segmentation.

Input: $\mathcal{E} = \{< x, y >_i\}$, $\mathcal{H} = \{h_{i,j}\}$ indicating whether $< i, j >$ is instance boundary.
Output: Per-point instance label $\{l_i\}$.

```
l_i ← 0  ∀i <= N.
id ← 1.
for i ← 1 to N do
    if l_i ≤ 0 then
        Q.push(i)
        l_i ← id
        while Q is not empty do
            v ←Q.front()
            for j ← Neighbors of v do
                if h_{v,j} = 0 then
                    l_j ← -id
                    continue
                end
                if l_j ≤ 0 and l_j ≠-id then
                    l_j ← id
                    Q.push(j)
                end
            end
            Q.pop()
        end
    end
end
```

4. Experiments

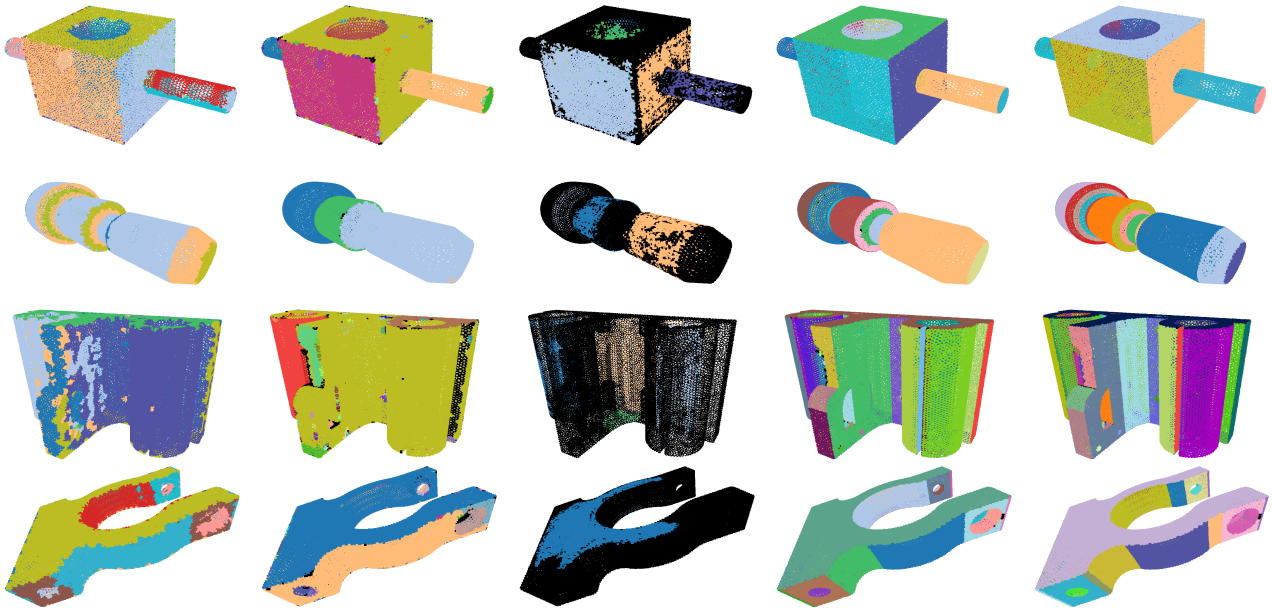
We make comparisons with existing methods in Section 4.1, where we show that our method handles extremely large scenes. Ablation studies (Section 4.2) highlight contributions of our novel designs. Finally, we demonstrate improvements of an important application that abstracts scans as lightweight models using our method in Section 4.3.

4.1. Comparison

We select several state-of-the-arts methods that could potentially be used for primitive instance segmentation. While [35] firstly proposed a primitive fitting network, ParseNet [57] is a recent method that outperforms it and is the current state-of-the-art for primitive instance segmentation and fitting. BoundaryNet [39] explicitly learns the boundaries and uses them to segment part instances. PointGroup [29] is one of the current state-of-the-art for semantic instance segmentation, which can be directly applied to our task treating primitive types as semantic classes. We adopt several evaluation metrics used in existing methods. Following ParseNet [57], segmentation mean IOU ("seg mIOU") computes averaged mean IOU of optimally matched segments, which measures the overall similarity of predicted segments with ground truth segments. Labeling IOU ("label mIOU") measures the overall primitive type prediction accuracy of optimally matched segments. For instance segmentation, another popularly used metric is mean average precision treating segments as inliers above certain mean IOU, including AP_{25}, AP_{50} and AP [11, 29].

ABC dataset ABC dataset [31] is a big CAD model dataset with annotations of per-point labeling of instance IDs and primitive types. We randomly segment the dataset into the training and test sets with a ratio of 3:1. We compare our method with PointGroup [29], BoundaryNet [39] and ParseNet [57]. During test time, we evaluate all methods with the full resolution. Since BoundaryNet and ParseNet accept a limited number of points as input, we randomly select 10 thousand points that the networks can afford, and map the predictions to the full resolution according to nearest neighbors. Table 1 shows the scores of different methods under different metrics in the test set.

We outperform existing methods under the overall metrics including seg mIOU and label mIOU. Notably, we significantly outperform existing methods under the AP-related metrics, indicating we are better at handling small instances. The contribution mainly comes from our high-resolution network with local surface property learning. Specifically, our joint learning of primitive embedding net-



(a) ParseNet (b) BoundaryNet (c) PointGroup (d) Ours (e) Ground Truth

Figure 4. Visualization of primitive instance segmentation on ABC dataset [31]. Our method correctly recover small instances with various primitive types and outperform existing state-of-the-arts.

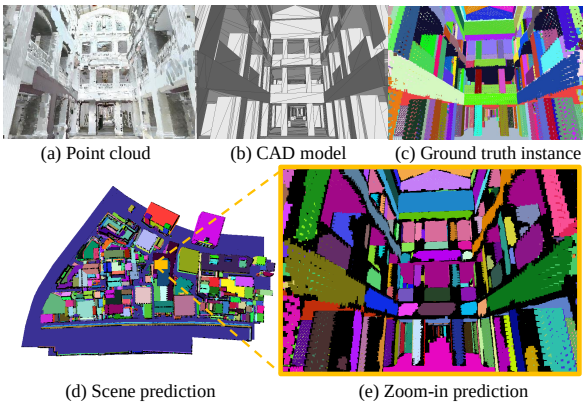
	seg mIOU	label mIOU	AP_{25}	AP_{50}	AP
ParseNet	71.1%	89.9%	25.7%	15.3%	11.4%
BoundaryNet	63.5%	90.6%	21.5%	13.6%	10.4%
PointGroup	61.4%	90.8%	19.9%	12.4%	10.2%
Ours	**85.7%**	**91.3%**	**74.3%**	**63.0%**	**57.7%**

Table 1. Evaluation for primitive instance segmentation on ABC dataset. Our method outperform existing state-of-the-arts especially according to AP-related metrics.

work and discriminator shows higher robustness comparing to methods with boundary predictions [39], instance centers as explicit supervision [29], or global descriptive features using triplet loss [57]. Figure 4 visualizes several examples where we accurately recover small primitive instances with various primitive types.

Real scene dataset Our method can be directly applied to real data in large scenes. Since there are no available real-world scene-level scans with primitive-level annotations, we prepare a self-collected dataset and use it for comparison. We collect 154 scene point clouds (Figure 5(a)) mixed with indoor and outdoor scans captured with Navvis[3] or reconstructed from multi-view stereo. For each scan, we ask an experienced artist to draw a CAD model (Figure 5(b)) to highlight the structure that is aligned with the scanned point clouds, where the median distance between points and the CAD model is below 2cm. For each CAD model, we consider adjacent triangles as from the same instance if their

[3] https://www.navvis.com/m6



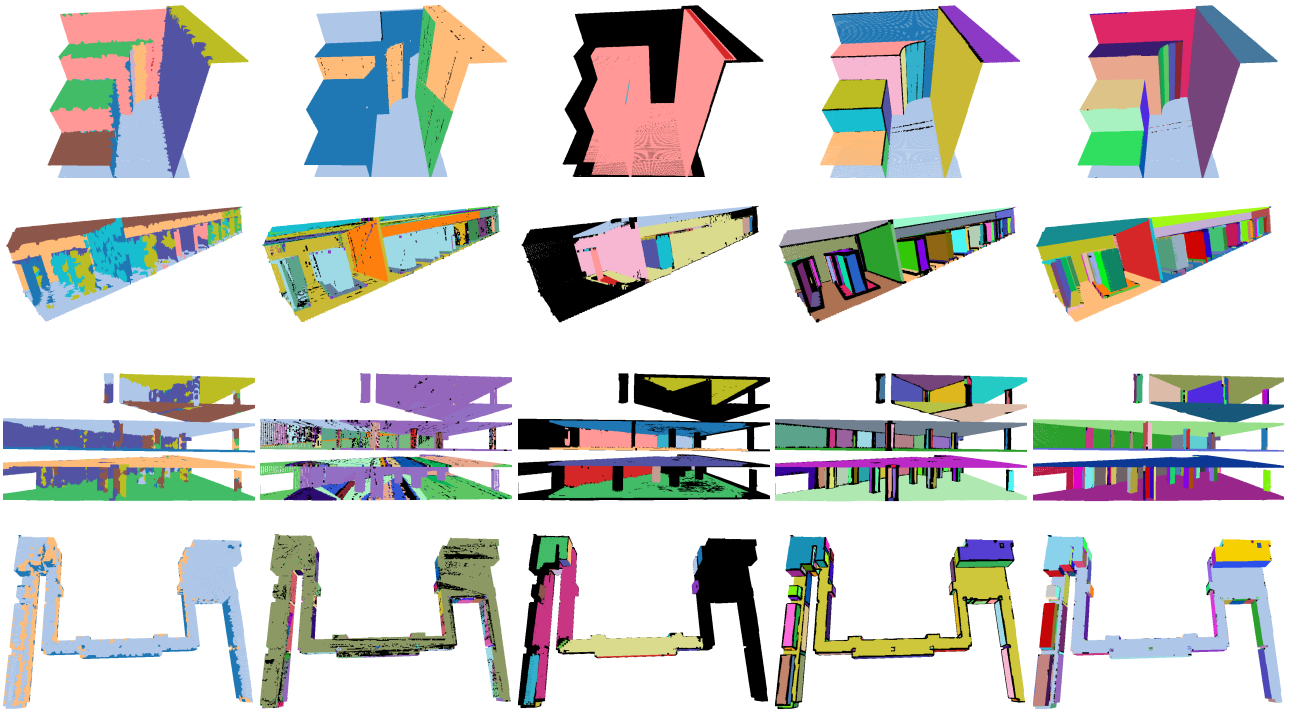
(a) Point cloud (b) CAD model (c) Ground truth instance

(d) Scene prediction (e) Zoom-in prediction

Figure 5. Real scene dataset. We capture (a) real scans and draw (b) CAD models with accurate alignment. (c) Instance labels are propagated from CAD models to scans for training. (d,e) We produce instance segmentation for scenes with extremely large scale.

normal angles are below 15°. We map instance labels to the original point clouds via nearest neighbors (Figure 5(c)). In this experiment, we assume that no primitive-type information is given. Points are labeled as "primitive" or "non-primitive" depending on whether their nearest neighbor distances to CAD models are smaller than 0.1m, where "non-primitive" indicates that the point should not be considered as primitive and can be removed.

Since collected scenes are huge, we further split them into chunks of $25m^3$. We split the dataset into training and test sets with a ratio of 3:1. We train different approaches and make comparisons with existing state-of-the-arts in the



(a) ParseNet (b) BoundaryNet (c) PointGroup (d) Ours (e) Ground Truth

Figure 6. Primitive instance segmentation for real scenes. We produce the most reasonable segmentation compared with other methods.

	seg mIOU	AP_{25}	AP_{50}	AP
ParseNet	45.4%	6.4%	2.9%	1.5%
BoundaryNet	53.3%	47.8%	32.4%	24.5%
PointGroup	60.3%	12.9%	6.8%	5.0%
Ours	**66.9%**	**60.0%**	**46.9%**	**39.1%**

Table 2. Evaluation for primitive instance segmentation on a self-collected real scene dataset.

	seg mIOU	label mIOU	AP_{25}	AP_{50}	AP
PointNet++	71.8%	87.9%	28.4%	16.5%	12.7%
DGCNN	72.0%	89.1%	30.1%	17.4%	13.0%
SpConv	85.3%	**91.8%**	73.6%	59.1%	53.1%
SPVCNN	85.1%	91.1%	72.9%	56.7%	50.5%
Ours	**85.7%**	91.3%	**74.3%**	**63.0%**	**57.7%**

Table 3. Primitive instance segmentation evaluation using our method with different backbones on ABC dataset.

test set. Table 2 shows the scores under different metrics, where label mIOU is not reported since primitive types are not available. Similar to the results on the ABC dataset, we outperform existing methods under the overall metrics and show significant improvements under the AP-related metrics. According to Figure 6, we produce the most reasonable segmentation at the scene level.

Large scale Our method can seamlessly merge chunks by aggregating discriminator predictions and run Algorithm 1 at the entire scene. Figure 5(d,e) visualizes our primitive segmentation of a large scene covering 0.1km^2 with more than 500M points.

4.2. Ablation Study

High resolution We experiment with different choices of networks as our backbone for the primitive embedding network on the ABC dataset. Alternative choices for the backbone include PointNet++ [50], DGCNN [64], SpConv [20] or SPVCNN [58]. We replace our high-resolution backbone with these methods and train the networks on the ABC dataset. We evaluate different backbones in Table 3. Results show that backbones with sparse convolutions (SpConv [20] or SPVCNN [58]) outperform other two networks. Our high-resolution backbone achieves the best performance combining PointNet [49] and SpConv [20].

Local surface property One of our insights is to learn local surface property by encouraging the embedding network and discriminator to focus on discriminative features in local neighborhoods. According to Section 3.3, the key difference is the choice of edge set $\mathcal{E}$ for the discriminator to train. Our method selects only closed point pairs according to $\mathcal{E}$. We experiment with two alternative methods, where the global one randomly picks pairs of points in the whole point cloud and the semi-local one randomly picks point pairs whose distance is smaller than 3ϵ. Table 4 lists the

ABC Data	seg mIOU	label mIOU	AP_{25}	AP_{50}	AP
Global	73.6%	87.9%	52.6%	35.3%	25.9%
Semi-local	76.1%	91.1%	59.8%	44.9%	38.7%
Local (Ours)	**85.7%**	**91.3%**	**74.3%**	**63.0%**	**57.7%**

Real Data	seg mIOU	AP_{25}	AP_{50}	AP
Global	37.0%	34.0%	17.1%	19.7%
Semi-local	49.3%	48.1%	32.5%	33.9%
Local (Ours)	**68.3%**	**61.4%**	**50.7%**	**40.8%**

Table 4. Evaluation on ABC and real dataset supervised point pairs at different scales of distances.

	Euclidean	Mahanalobis	Ours
Difference IOU	71.1%	69.8%	**88.8%**

Table 5. Evaluation on ABC dataset where primitive embedding network is trained with different metrics.

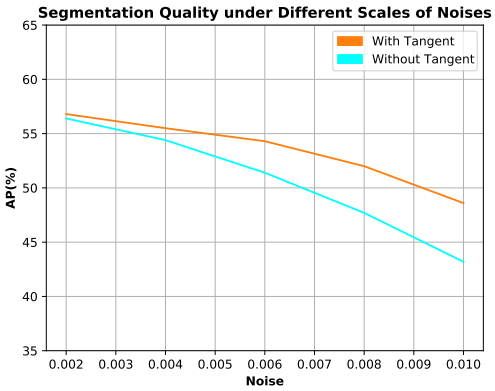
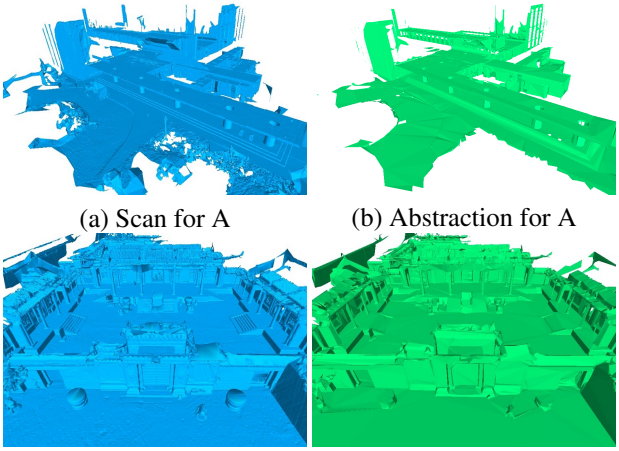


Figure 7. With tangent space prediction, our method is more robust to different levels of noises.

scores on both ABC and real datasets by selecting edges at different scales. As a result, local supervision is the best choice for primitive instance segmentation.

Choice of metrics Alternatives to our adversarial metric include euclidean or Mahanalobis distance (E-dis or M-dis) trained with triplet loss, where we set the distance margin as one to decide whether two features belong to the same instance. We report the IOU of edges connecting different instances according to network prediction and ground truth. As shown in Table 5, our adversarial metric achieves the best score comparing with these alternatives, probably because that the adversarial metric has more capacity to distinguish implicit local surface properties.

Noise levels To understand the behavior of our method under different noise levels, we simulate noises on the ABC dataset by perturbing each point by a random Gaussian noise at different scales. Figure 7 shows the prediction quality under different noise scales. Our tangent space prediction improves our method and makes it more robust to segment primitive instances at different noise scales.



(a) Scan for A (b) Abstraction for A

(c) Scan for B (d) Abstraction for B

Figure 8. Our approach can be integrated into [43] to abstract scans (a,c) as lightweight models (b,d).

Scenes	Input	[43] + [51]	[43] + Ours	CAD
A	18M	163k	52k	7k
B	12M	94k	39K	12k

Table 6. Number of triangles in the scans, results from different abstraction solutions, and human-created CAD models.

4.3. Application

Results produced by our approach can be further processed by geometry processing algorithms [5, 32, 23, 4, 2, 43] to abstract point clouds as light-weight models. We first triangulate the scanned pointcloud using [3]. We predict instance segmentation and implement [43] to derive the abstracted shape. Figure 8 shows two scenes (A and B) where scans are converted to lightweight models, where geometry structures are well-preserved. In Table 6, we additionally report numbers of triangles from original scans, [51] combined with traditional primitive segmentation [51], our results and human-created CAD models. While the abstracted results preserve geometry, it is not as clean or lightweight as human-created CAD models. This is the limitation of [43], where properly assembling primitives is another open research problem. However, we find our method yields fewer triangles than [51] combined with [43], indicating that our method is better than traditional primitive detection for such a pipeline. We provide more results and comparisons in the supplemental material.

5. Conclusion

We present a novel primitive instance segmentation approach by jointly training an embedding network and a discriminator to learn local surface properties. It significantly outperforms existing works and handles large scenes. Our approach benefits traditional algorithms for abstracting point clouds as lightweight 3D models.

References

[1] Agisoft metashape. https://www.agisoft.com/. 1

[2] Jean-Philippe Bauchet and Florent Lafarge. Kinetic shape reconstruction. *ACM Transactions on Graphics (TOG)*, 39(5):1–14, 2020. 1, 2, 8

[3] Dobrina Boltcheva and Bruno Lévy. *Simple and scalable surface reconstruction*. PhD thesis, LORIA-Université de Lorraine; INRIA Nancy, 2016. 8

[4] Alexandre Boulch, Martin de La Gorce, and Renaud Marlet. Piecewise-planar 3d reconstruction with edge and corner regularization. In *Computer Graphics Forum*, volume 33, pages 55–64. Wiley Online Library, 2014. 2, 8

[5] Jie Chen and Baoquan Chen. Architectural modeling from sparsely scanned range data. *International Journal of Computer Vision*, 78(2-3):223–236, 2008. 1, 2, 8

[6] Shuo Chen, Chen Gong, Jian Yang, Xiang Li, Yang Wei, and Jun Li. Adversarial metric learning. *arXiv preprint arXiv:1802.03170*, 2018. 3

[7] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2, 3

[8] Ondrej Chum and Jırı Matas. Randomized ransac with td, d test. In *Proc. British Machine Vision Conference*, volume 2, pages 448–457, 2002. 2

[9] Ondrej Chum and Jiri Matas. Matching with prosac-progressive sample consensus. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 220–226. IEEE, 2005. 2

[10] Yin Cui, Feng Zhou, Yuanqing Lin, and Serge Belongie. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1153–1162, 2016. 3

[11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 5

[12] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(3):24, 2017. 1

[13] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216, 2007. 3

[14] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2780–2789, 2018. 3

[15] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9031–9040, 2020. 2

[16] Hao Fang, Florent Lafarge, and Mathieu Desbrun. Planar shape detection at structural scales. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2973, 2018. 2

[17] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[18] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997. 1

[19] Amir Globerson and Sam Roweis. Metric learning by collapsing classes. *Advances in neural information processing systems*, 18:451–458, 2005. 3

[20] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 2, 3, 4, 7

[21] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 3

[22] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2940–2949, 2020. 2

[23] Thomas Holzmann, Michael Maurer, Friedrich Fraundorfer, and Horst Bischof. Semantically aware urban 3d reconstruction with plane-based regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 468–483, 2018. 1, 2, 8

[24] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019. 2

[25] Jingwei Huang, Angela Dai, Leonidas J Guibas, and Matthias Nießner. 3dlite: towards commodity 3d scanning for content creation. *ACM Trans. Graph.*, 36(6):203–1, 2017. 2

[26] Jingwei Huang, Hao Su, and Leonidas Guibas. Robust watertight manifold surface generation method for shapenet models. *arXiv preprint arXiv:1802.01698*, 2018. 2

[27] Jingwei Huang, Justus Thies, Angela Dai, Abhijit Kundu, Chiyu Jiang, Leonidas J Guibas, Matthias Nießner, Thomas Funkhouser, et al. Adversarial texture optimization from rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1559–1568, 2020. 3

[28] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew 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. ACM, 2011. 1

[29] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4867–4876, 2020. 2, 5, 6

[30] Zhizhong Kang and Zhen Li. Primitive fitting based on the efficient multibaysac algorithm. *PloS one*, 10(3):e0117341, 2015. 2

[31] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9611, 2019. 1, 2, 5, 6

[32] Florent Lafarge and Pierre Alliez. Surface reconstruction through point set structuring. In *Computer Graphics Forum*, volume 32, pages 225–234. Wiley Online Library, 2013. 1, 2, 8

[33] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9256–9266, 2019. 2

[34] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019. 4

[35] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J Guibas. Supervised fitting of geometric primitives to 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2652–2660, 2019. 1, 2, 5

[36] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. Globfit: Consistently fitting primitives by discovering global relations. In *ACM SIGGRAPH 2011 papers*, pages 1–12. 2011. 2

[37] Peter Lindstrom. Out-of-core simplification of large polygonal models. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 259–262, 2000. 1

[38] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *arXiv preprint arXiv:1907.03739*, 2019. 3, 4

[39] Marios Loizou, Melinos Averkiou, and Evangelos Kalogerakis. Learning part boundaries from 3d point clouds. In *Computer Graphics Forum*, volume 39, pages 183–195. Wiley Online Library, 2020. 1, 2, 5, 6

[40] Jiwen Lu, Junlin Hu, and Yap-Peng Tan. Discriminative deep metric learning for face and kinship verification. *IEEE Transactions on Image Processing*, 26(9):4269–4282, 2017. 3

[41] David Marshall, Gabor Lukacs, and Ralph Martin. Robust segmentation of primitives from range data in the presence of geometric degeneracy. *IEEE Transactions on pattern analysis and machine intelligence*, 23(3):304–314, 2001. 1, 2

[42] Daniel Maturana and Sebastian 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. IEEE, 2015. 3

[43] Ravish Mehra, Qingnan Zhou, Jeremy Long, Alla Sheffer, Amy Gooch, and Niloy J Mitra. Abstraction of man-made shapes. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–10. 2009. 2, 8

[44] Aron Monszpart, Nicolas Mellado, Gabriel J Brostow, and Niloy J Mitra. Rapter: rebuilding man-made scenes with regular arrangements of planes. *ACM Trans. Graph.*, 34(4):103–1, 2015. 2

[45] TM Murali and Thomas A Funkhouser. Consistent solid and boundary representations from arbitrary polygonal data. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 155–ff, 1997. 2

[46] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. *arXiv preprint arXiv:1903.01177*, 2019. 2

[47] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011. 1

[48] Sven Oesau, Florent Lafarge, and Pierre Alliez. Planar shape detection and regularization in tandem. In *Computer Graphics Forum*, volume 35, pages 203–215. Wiley Online Library, 2016. 2

[49] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 3, 4, 7

[50] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 3, 7

[51] Tahir Rabbani, Frank Van Den Heuvel, and George Vosselmann. Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences*, 36(5):248–253, 2006. 1, 2, 8

[52] David Salinas, Florent Lafarge, and Pierre Alliez. Structure-aware mesh decimation. In *Computer Graphics Forum*, volume 34, pages 211–227. Wiley Online Library, 2015. 1

[53] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007. 1, 2

[54] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 3

[55] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. *Advances in neural information processing systems*, 16:41–48, 2004. 3

[56] Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. Csgnet: Neural shape

parser for constructive solid geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5523, 2018. 2

[57] Gopal Sharma, Difan Liu, Subhransu Maji, Evangelos Kalogerakis, Siddhartha Chaudhuri, and Radomír Měch. Parsenet: A parametric surface fitting network for 3d point clouds. In *European Conference on Computer Vision*, pages 261–276. Springer, 2020. 1, 2, 5, 6

[58] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision*, pages 685–702. Springer, 2020. 3, 4, 7

[59] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1):138–156, 2000. 2

[60] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017. 2

[61] Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. *arXiv preprint arXiv:1611.00822*, 2016. 3

[62] Mikaela Angelina Uy, Jingwei Huang, Minhyuk Sung, Tolga Birdal, and Leonidas Guibas. Deformation-aware 3d model embedding and retrieval. In *European Conference on Computer Vision*, pages 397–413. Springer, 2020. 3

[63] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2593–2601, 2017. 3

[64] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 3, 7

[65] Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. Elasticfusion: Dense slam without a pose graph. *Proc. Robotics: Science and Systems, Rome, Italy*, 2015. 1

[66] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *arXiv preprint arXiv:1906.01140*, 2019. 2

[67] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2019. 2

[68] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 900–909, 2017. 2