3D Instance Segmentation via Multi-task Metric Learning

Jean Lahoud
KAUST

Bernard Ghanem
KAUST

Marc Pollefeys
ETH Zurich

Martin R. Oswald
ETH Zurich

Abstract

We propose a novel method for instance label segmentation of dense 3D voxel grids. We target volumetric scene representations which have been acquired with depth sensors or multi-view stereo methods and which have been processed with semantic 3D reconstruction or scene completion methods. The main task is to learn shape information about individual object instances in order to accurately separate them, including connected and incompletely scanned objects. We solve the 3D instance-labeling problem with a multi-task learning strategy. The first goal is to learn an abstract feature embedding which groups voxels with the same instance label close to each other while separating clusters with different instance labels from each other. The second goal is to learn instance information by estimating directional information of the instances' centers of mass densely for each voxel. This is particularly useful to find instance boundaries in the clustering post-processing step, as well as for scoring the quality of segmentations for the first goal. Both synthetic and real-world experiments demonstrate the viability of our approach. Our method achieves state-of-the-art performance on the ScanNet 3D instance segmentation benchmark [4].

1. Introduction

A central goal of computer vision research is high-level scene understanding. Recent progress now allows for reliable results for a variety of computer vision problems including image classification [23, 43, 47], image segmentation [1, 32, 41], object detection [13, 29, 38–40] or instance segmentation on 2D images [2, 7–9, 17, 20, 26, 27, 31, 36]. Further, it is now possible to recover highly-detailed 3D geometry with low-cost depth sensors [19, 35, 46, 54] or with image-based 3D reconstruction algorithms [11,21,42]. Likewise, many algorithms have been proposed for 3D scene and object classification [33,44,50], 3D object detection [25, 51] as well as method for joint 3D reconstruction and semantic labeling [3, 5, 6, 24, 48].

Advances in 2D instance segmentation were mainly fueled by the large number of datasets and challenges available in the 2D setting. When compared to the plethora of powerful methods for instance segmentation of 2D images, the problem in a 3D setting has been less explored in the literature. In addition to the lack of datasets, the majority of the 2D methods are not applicable to the 3D setting or their extension is by no means straightforward.

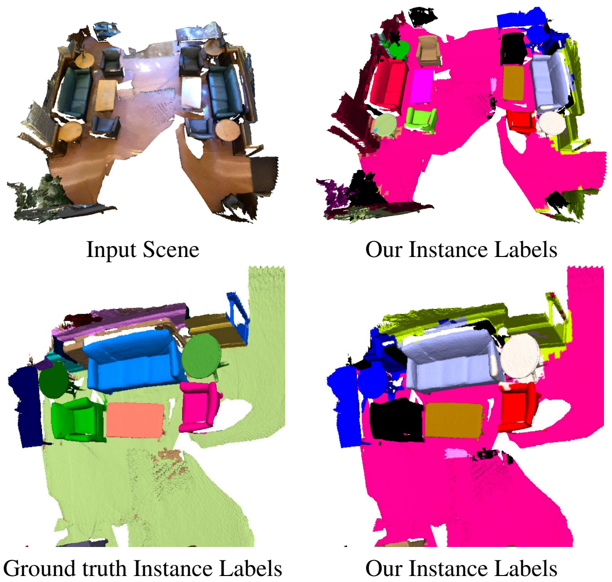


Figure 1. **Example results of our method.** Our method takes as input a 3D point cloud, and outputs instance labels unique to each object within the scene.

With the emergence of labeled datasets and benchmarks for 3D instance segmentation such as ScanNet [4], numerous works have surfaced to tackle 3D object instance segmentation. In many cases, the work in 3D benefits from the pioneering works in 2D, with modifications that allow processing of information in 3D. The 3D processing in many cases is similar to other 3D understanding techniques, mainly semantic segmentation.

In this paper, we address the problem of 3D instance segmentation: Given the 3D geometry of a scene, we want to label all the geometry that belongs to the same object with a unique label. Unlike previous methods that entangle instance labeling with semantic labeling, we propose a technique that would mainly focus on the instance labeling through grouping/clustering of information pertaining to a single object. Our method still benefits from semantic information as a local cue, but adds to it information relating to real 3D dimensions and 3D connectivity, whose usefulness is unique to the 3D setting.

In particular, we propose a learning algorithm that would process a 3D grid and learn two main characteristics: (1) a feature descriptor unique to every instance, and (2) a direction that would point towards the instance center. Our method aims at providing a grouping force that is independent of the size of the scene and the number of instances within. In summary, we make the following **contributions:**

- We propose a multi-task neural network architecture for 3D instance segmentation of voxel-based scene representations. Besides a metric learning task we let the network predict directional information to the object's center. We demonstrate that the multi-task learning improves the results for both tasks. Our approach is robust and scalable, therefore suitable for processing the large amounts of data in a 3D scene.
- Our experiments demonstrate state-of-the-art performance for 3D instance segmentation. By the time of submission, our method ranks first in terms of average AP50 score on the ScanNet 3D instance segmentation benchmark [4].

2. Related Work

This section gives a brief overview about related 2D and 3D approaches. Especially for 2D deep learning-based semantic segmentation and instance label segmentation a large amount of works exist, recent surveys can be found in [12, 15].

2D Instance Segmentation via Object Proposals or Detection. Girshick [13] proposed a network architecture that creates region proposals as candidate object segments. In a series of works, this idea has been extended be faster [40] and to additionally output pixel-accurate masks for instance segmentation [17]. The authors of YOLO [38] and its follow-up work [39] apply a grid-based approach in which each grid cell generates an object proposal. DeepMask [36] learns to jointly estimate an object proposal and an object score. Lin *et al.* [29] propose a multi-resolution approach for object detection which they call feature pyramid networks. In [16], the region proposals are refined with network that predicts the distance to the boundary which is then transformed into a binary object mask. Khoreva *et al.* [20] jointly perform instance and semantic segmentation. A similar path follows [26], which combines fully convolutional networks for semantic segmentation with instance mask proposals. Dai *et al.* [8] use fully convolutional networks (FCNs) and split the problem into bounding box estimation, mask estimation and object categorization and propose a multi-task cascaded network architecture. In a follow-up work [7], they combine FCNs with windowed instance-sensitive score maps.

While all these approaches have been very successful in the 2D domain, many of them are already require large amounts of resources and their extension to the 3D domain is non-trivial and challenging.

2D Instance Segmentation via Metric Learning. In [27], Liang *et al.* [27] propose a method without object proposals as they directly estimate bounding box coordinates and confidences in combination with clustering as a post-processing step. Fathi *et al.* [9] compute likelihoods for pixels belonging to the same object and then by grouping similar pixel together within an embedding space. Kong and Fowlkes [22] train a network that assigns all pixels to a spherical embedding in which points of the same object instance within are within a close vicinity and non-instance related points are placed apart from each other. The instances are then extracted via a variant of mean-shift clustering [10] that is implemented as a recurrent network. The approach by DeBrabandere *et al.* [2] follows the same idea, but the authors do not impose constraints on the shape of the embedding space. Likewise, they compute the final segmentation via mean-shift clustering in the feature space.

None of these approaches has been applied to a 3D setting. Our approach builds upon the work of DeBrabandere *et al.* [2]. We extend this method with a multi-task approach for 3D instance segmentation on dense voxel grids.

3D Instance Segmentation. Wang *et al.* [49] propose SGPN, an instance segmentation for 3D point clouds. In first step they extract features with PointNet [37] and subsequently build a similarity matrix in which each element classifies weather two points belong to the same object instance. The approach is not very scalable and limited to small point cloud sizes as they construct a similarity matrix with the size of the number of points squared.

Further, there are a number of recent concurrent or unpublished works that address 3D instance segmentation. GSPN [53], a generative shape proposal network relies on object proposals to identify instances in 3D point clouds. 3D-SIS [18] jointly evaluating 3D features and 2D features which are aggretated from multiple RGB-D input views. MASC [30] relies on the high performance of the SparseConvNet [14] architecture and combine it with an affinity score for instances that is estimated across multiple scales. PanopticFusion [34] predicts pixel-wise labels for RGB frames and carries them over into a 3D grid where a fully connected CRF is used for final inference.

Apart from these recent concurrent works there has been generally very little research performed on 3D instance segmentation so far.

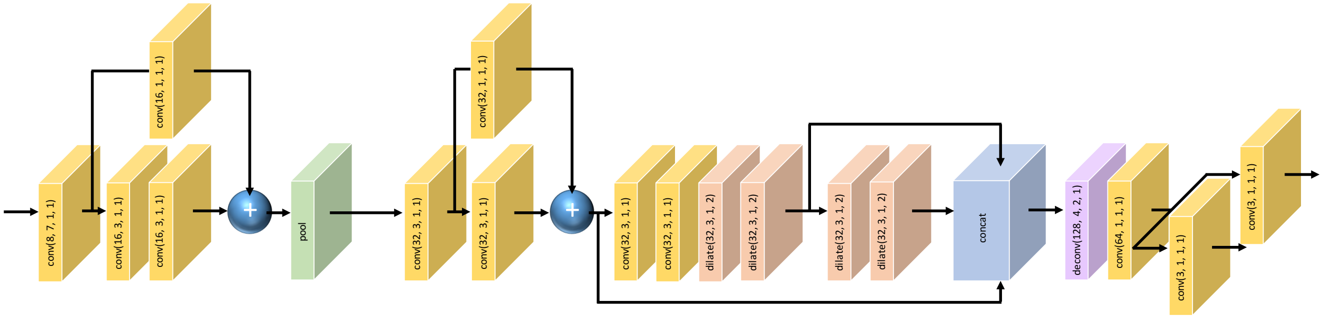


Figure 2. **Overview of our network architecture.** We cast the 3D instance segmentation task as a multi-task learning problem. The input to our method is a voxel grid and the output are two latent spaces: 1) a feature vector embedding that groups voxels with similar instance label close in the latent space; 2) a 3D latent space that encodes directional predictions for each voxel.

3. Method Overview

In this work, we aim at segmenting 3D instances of a given 3D scene. To fully locate a 3D instance, one would require both a semantic label and an instance label. Rather than solving the complex task of scene completion, semantic labeling and instance segmentation at once, we model our 3D instance segmentation process as a post-processing step of the semantic segmentation labeling. We focus on the grouping and splitting of semantic labels, relying on inter-instance and intra-instance relations. We benefit from the real distances in 3D scenes, where sizes and distances between objects are key to the final instance segmentation.

We split our task into a label-segmentation then instance-segmentation problem as we believe that features learned in each step possess task-specific information. Semantic segmentation can rely on local information to predict the class label. Learning to semantically label a volumetric representation inherently encodes features from neighboring volumes but does not require knowledge of the whole environment. On the other hand, instance segmentation requires a holistic understanding of the scene in order to join and/or separate semantic labeled volumes.

Problem Setting. Our method takes in a voxelized 3D space with each voxel encoding either a semantic label or a local feature vector learned through semantic labeling. In particular, we use the semantic labeling network in [14]. We fix our voxel size to preserve 3D distances among all voxels within a scene. In problem settings where point cloud or meshes are available, one could generate a 3D voxelization by grouping information from points within every voxel. Our method then processes the voxelized 3D space and outputs instance label masks, each corresponding to a single object in the scene, along with its semantic label. The output mask can also be reprojected back into a point cloud by assigning the voxel label to all points within.

3.1. Network Architecture

In order to process our input, we utilize a 3D convolution network which is based on the SSCNet architecture [45]. We apply some changes to the original SSCNet network to better suit our task. As shown in Figure 2, the network input and output are equally sized. Since the pooling layer scales down the scene size, we use a transpose of convolution (also referred as deconvolutions [55]) to upsample back into the original size. We also use larger dilations for diluted 3D convolution layers to increase the receptive field. We make the receptive field large enough to access all the voxels of usual indoor rooms. With a voxel size of 10cm, our receptive field is as large as 14.2m. With larger scenes, our 3D convolution network would still be applicable to the whole scene, while preserving the filter and voxel sizes, and thus preserving the real distances. Objects lying at distances larger than the receptive field are separated by default.

3.2. Multi-task Loss Function

In order to group voxels of the same instance, we aim at learning two sets of feature embeddings. The first set maps every voxel into a feature embedding in a feature space, where voxels of the same instance have closer feature embeddings. This is similar to the work of DeBrabandere *et al.* [2], but applied in a 3D setting. The second set of feature embeddings assigns a 3D vector to every voxel, where the vector would point towards the physical center of the object. This allows to learn shape containment and removes ambiguities among similar shapes.

In order to learn our feature embeddings, we introduce a multi-task loss function that we minimize in our training. The first loss encourages discrimination in the feature space among multiple instances. The second loss penalizes angular deviations of vectors from the desired direction.

Feature Embedding Loss. We follow the work of DeBrabandere *et al.* [2] and define the feature embedding loss as a weighted sum of three terms: **(1)** An intra-cluster variance term $\mathcal{L}_{\text{var}}$ that pulls features that should belong to the

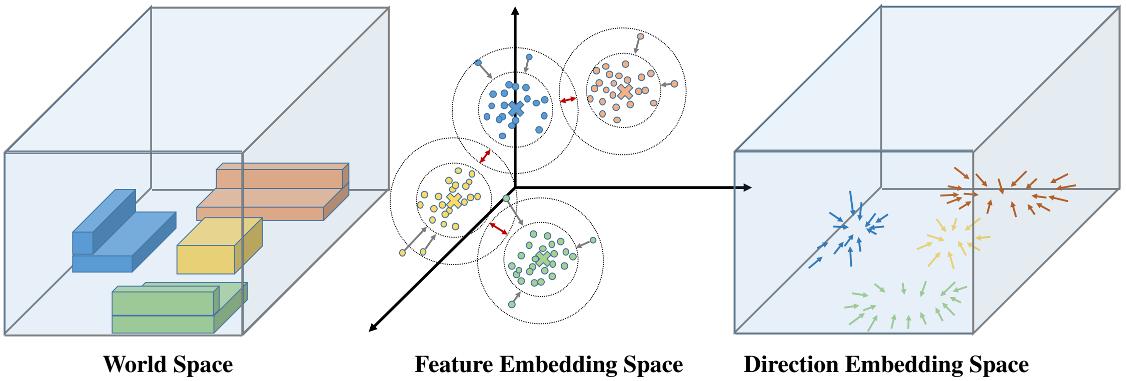


Figure 3. **Embedding space visualization.** Voxels with similar labels are mapped to similar locations in the feature embedding space. The red arrows depict inter-class push forces, while the grey arrows indicate intra-class pull forces.

same instance towards the mean feature, **(2)** An inter-cluster distance term $\mathcal{L}_{\text{dist}}$ which ensures that clusters with different instance labels are pushed apart, **(3)** A regularization term $\mathcal{L}_{\text{reg}}$ that pulls all features towards the origin in order to bound the activations.

$$\mathcal{L}_{\text{FE}} = \gamma_{\text{var}}\mathcal{L}_{\text{var}} + \gamma_{\text{dist}}\mathcal{L}_{\text{dist}} + \gamma_{\text{reg}}\mathcal{L}_{\text{reg}} \tag{1}$$

The individual loss functions are weighted by $\gamma_{\text{var}} = \gamma_{\text{dist}} = 1$, $\gamma_{\text{reg}} = 0.001$ and are defined similar to [2] as follows:

$$\mathcal{L}_{\text{var}} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}[\|\mu_c - x_i\| - \delta_{\text{var}}]_+^2 \tag{2}$$

$$\mathcal{L}_{\text{dist}} = \frac{1}{C(C-1)}\sum_{c_A=1}^{C}\sum_{\substack{c_B=1\\ c_B\neq c_A}}^{C}[2\delta_{\text{dist}} - \|\mu_{c_A} - \mu_{c_B}\|]_+^2 \tag{3}$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{C}\sum_{c=1}^{C}\|\mu_c\| \tag{4}$$

Here C is the number of ground truth clusters, N_c denotes the number of elements in cluster c, μ_c is the cluster center, i.e. the mean of the elements in cluster c, and x_i is a feature vector. Further, the norm $\|\cdot\|$ denotes the ℓ_2-norm and $[x]_+ = max(0, x)$ the hinge. The parameter δ_{var} describes the maximum distance which feature vector x_i is allowed to have from the cluster center μ_c in order to belong to cluster c. Likewise, $2\delta_{\text{dist}}$ is the minimum distance that different cluster centers should have in order to avoid overlap. A visualization of the forces and the embedding spaces can be found in Figure 3. Feature embeddings of different clusters exert forces on each other, which means that each feature embedding is affected by the number and location of other cluster centers. This connection might be disadvantageous in some cases, especially when a large number of instances occur in a single scene. Therefore, we propose next an additional loss that provides local information essential for instance separation without being affected by other instances.

Directional Loss. We here aim to generate a vector feature that would locally describe the inter-cluster relationship without being affected by other clusters. We choose the vector to be the one pointing towards the real center of the obect. In order to learn the vector feature, we define the directional loss as follows

$$\mathcal{L}_{\text{dir}} = -\frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}v_i{\cdot}v_i^{GT} \quad \text{with} \quad v_i^{GT} = \frac{X_i - X_c}{\|X_i - X_c\|} \tag{5}$$

Here v_i denotes the normalized vector feature, v_i^{GT} is the desired direction which points towards the object center, X_i is the voxel center location, and X_c is the object center location.

Joint Loss. We jointly minimize both the feature embedding loss and the directional loss during training. Our final joint loss read as

$$\mathcal{L}_{\text{joint}} = \alpha_{\text{FE}}\mathcal{L}_{\text{FE}} + \alpha_{\text{dir}}\mathcal{L}_{\text{dir}} \tag{6}$$

Post-processing. We apply mean-shift clustering [10] on the feature embedding. Similar to object detection algorithms, instance segmentation does not restrict the labeling to one coherent set, and thus allows overlap between multiple objects. We use the mean-shift clustering output with multiple thresholds as proposals that are scored according to their direction feature consistency. We also use connected components for suggested splitting that would further be scored by the coherency of its feature embeddings. The coherency of the feature embedding is described by the number of feature embedding that lie within a given threshold from the feature cluster center. The directional feature coherency score is simple - $\mathcal{L}_{\text{dir}}$, which is the average of cosine the angle between the normalized vector pointing from the voxel to the center of the object and the normalized direction loss. We then sort all object proposals and perform non-maximum suppression (NMS) to remove objects that overlap by more than a threshold. The final score is obtained by appending the both feature embedding scores with a score that motivates objects of regular sizes over extremely large or small obects. As for the semantic label, it is chosen to be the most probable label for all points within the clustered voxels.

3.3. Network Training

Training Data. During training, we append flips of voxelized scenes as well as multiple orientations around the vertical axis to our training data. We pretrain our network using groundtruth segmentation labels as input, with labels one-hot encoded to maintain the same sized input as training from segmentation output.

4. Results and Evaluation

Setup. Our network has been implemented in Tensorflow and run under Linux with Nvidia GTX1080Ti GPU. For the network training We use ADAM optimizer and a learning rate $5e{-}4$ and batch size 2. The training converged after about 100 epochs and took about 2 days. The inference time for our network is about 1s for scene sizes of 1.6M voxels.

Datasets. For experimental evaluation, we trained an tested our method on the following datasets which include real and synthetic data.

- **Synthetic Toy Dataset:** In order to validate our approach, we create a synthetic dataset with objects of different sizes and aspect ratios placed on a planar surface. We introduce 5 object shapes, where each shape is analogous to an object class in the real data. The shapes of the objects considered are shown in Figure 5. We then randomly orient and position objects on the surface plane, and randomly choose whether an object is in contact with another object. We generate 1000 scene, and split our dataset into 900 training scenes, and 100 testing scenes.
- **ScanNet [4]:** We conduct experiments on the ScanNet v2 dataset, which contains 1513 scans with 3D instance annotations. We train our network on the train split, which contains 1201 scans, and validate on the remaining 312 scans. Additional unlabeled 100 scans form an evaluation test set.

Evaluation metrics. Following the evaluation procedure adopted in most instance segmentation methods as well as ScanNet evaluation benchmark, we use the average precision metric (AP) score to evaluate our proposed algorithm. We use the AP25 and AP50 metrics, which denote AP score with a minimum intersection-over-union (IoU) threshold of 25% and 50% respectively. The AP score averages scores obtained with IoU thresholds ranging from 50% to 95% with step of 5.

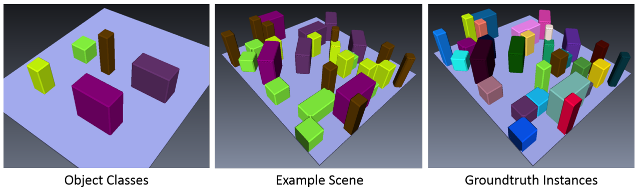


Figure 4. **Overview of the synthetic toy dataset. Left:** We consider 5 different object classes represented by cubes with various edge lengths. **Middle:** Example scene with object colors showing the class labels. **Right:** Corresponding ground thruth instance labeling (random color per instance).

Baselines. For assessing the performance of our method we considered the following baseline methods:

- **Input Segmentation:** In this case, we assume that the segmentation label, which is input to our method, to be the desired instance segmentation label. If every scene contained a single instance of every semantic label, this baseline would be ideal. In reality, these scenes barely occur, but such metric would still serve as an inception to whether splitting and/or grouping voxels is logical.
- **Connected Component:** Given the ground truth segmentation labels, a connected components algorithm already correctly labels all instances as long as they are not touching. Since this happens seldomly in a 3D setting, this is usually a high-scoring and challenging baseline.
- Via the ScanNet [4] benchmark we further compare against recent submissions: **Mask R-CNN proj** [17], **SGPN** [37], **GSPN** [53], **3D-SIS** [18], **MASC** [30], **PanopticFusion** [34], **Occipital-SCS**, and **3D-BoNet** [52].

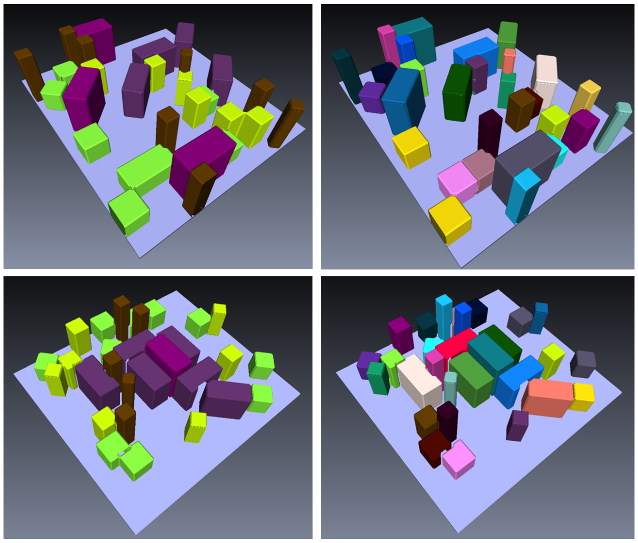
4.1. Evaluation on Synthetic 3D Data

We evaluate our method on the simple toy dataset, and report AP50 score for all objects in Table 1. In this part, we allow only one coherent labeling. Note that the directional loss alone is not discriminative enough for subsequent clustering and is therefore not considered in the ablation study. Generating object proposals from directional information only is tedious since the directional information is noisy and the clustering problem much more difficult and less efficient. Therefore, we do not evaluate the directional prediction alone in the ablation study and we resort to using object proposals from mean shift clustering and using the directional information for scoring.

The goal of the simple toy problem in Figure 5 was to study whether the network can abstract and differentiate various object size although their shape is rather similar. Further, it is interesting to see how the method performs when object instances are spatially touching, especially when they belong to the same semantic class. Al-

Method	Obj1	Obj2	Obj3	Obj4	Obj5
Connected comp.	92.5	85.1	86.9	93.5	79.9
Ours (FE only)	97.3	92.7	95.0	96.4	95.2
Ours (Multi-task)	**98.0**	**93.5**	**96.1**	**96.6**	**95.3**

Table 1. **AP50 results on synthetic toy dataset.** On this dataset with 5 objects our approach with the multi-task learning as well as the basline with only feature embedding (FE) outperform the connected components baseline which are computed on the ground truth semantic labels. The difference between *FE only* and *Multi-task* is small in a noise-free setting.



Input Scenes with Semantic Labels

Output Scenes with Instance Labels

Figure 5. **Experiment on synthetic toy dataset.** Two examples of random scenes for which our network generated instance labels.

though the input features are very similar due to the same object class and the spatial proximity, our network needs to place the corresponding feature vectors in different locations in the feature space.

4.2. Evaluation on Real 3D Data

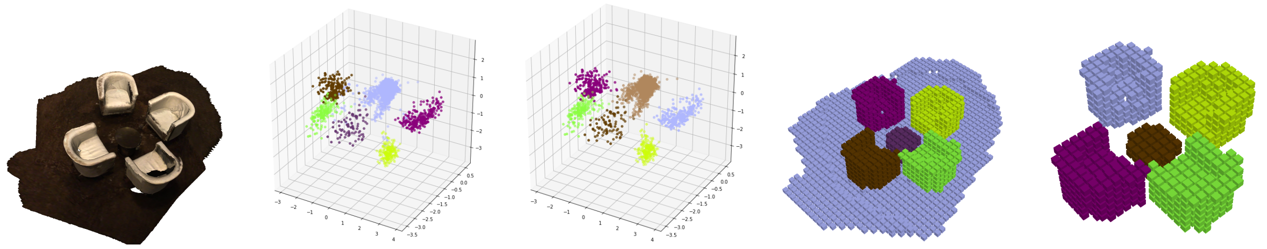
Feature Space Study. The feature loss Eq. (1) mainly performs two intuitive tasks: pulling points belong to the same instance together and pushing clusters of different instances apart. Since real data contains noise, outliers and missing data, the mapping of individual points in the feature space might be less discriminative and clusters might be overlapping. In Figure 6 we visualize the 3D feature space in order to study these effects and observe that feature points are indeed spreading towards neighboring clusters, but for this example the clustering results is not influenced and still achieve high accuracy. Note that we exclude ground and wall labels since their instance segmentation and splitting is less meaningful and also ignored in the benchmark.

Evaluation on ScanNet Output. In Figure 7 we present qualitative results on the ScanNet dataset [4]. The results of our methods on the voxel grid are simply projected onto the mesh which is then used for evaluation on the benchmark. As can be seen in the rightmost column, our method sometimes splits objects like the desk or the labels of furniture bleed into neighboring geometry. Due to our mostly geometric approach, our method needs sufficient structural changes to recognize object boundaries and to potentially relabel a new instance. Nonetheless, our proposed method was in most cases able to group single object instances together.

In Table 2 we provide a comparison against baselines such as the instance labeling performance of our input segmentation labeling SparseConvNet [14] for which we test the output of the SparseConvNet labeling method [14]. Further, we evaluate a simple connected component labeling method on the segmentation labeling, because in the 3D setting in general, including the considered datasets, only very few object instances are touching each other. Hence, this connected component baseline is already a challenging one especially for a rather noise free geometry and labeling. With increasing amounts of noise the connected component labeling rapidly performs worse. In the table it is apparent that the connected component method usually substantially improves the instance labeling results. In rare cases, the results get worse, which is due to the fact that the scenes are not completely scanned and a single object instance might be disconnected due to missing scene parts.

Ablation Study: Single-task vs. Multi-task. Moreover, we focus on comparing our network with single-task learning against multi-task learning. The six rightmost columns in Table 2 show the result of single-task learning and multi-task learning. With very few exceptions, the network trained with a multi-task loss consistently outperforms the single-task one. This is in line with the results on the synthetic dataset and supports our hypothesis that the directional loss adds more discriminative features which are helpful to group the features according to object instances in the feature space. For objects that rarely have multiple instances within a scene, such as the 'counter' class, the segmentation as instance outperforms our method. Still, this occurrence is uncommon as can be noticed with the overall average evaluation.

Table 3 provides an overview of our benchmark results on the ScanNet test dataset (hidden groundtruth). One can see that our method outperforms the other methods in terms of AP50 scores. Other methods include ones that process all RGB-D images that were used to reconstruct the scenes of ScanNet. Instance labels of single RGB-D frames in these methods are propagated throughout the whole scene and concatenated based on the location estimation. On the other hand, our method directly operates in the 3D setting, without the need to use the 2D information. This leads to



Input (RGB) Feature Label GT Feature Label Ours GT Label Clustering Label

Figure 6. **Visualization of the feature embedding and labelling.** This figure shows the input 3D colored scene plots the generated 3D feature embeddings, along with the GT label and our instance labeling result after the the mean-shift clustering (colors of the instances are random and do not correspond to GT).

	Segment. [14] as Instance			Connect. Comp. on [14]			Ours (FE only)			Ours (Multi-task)		
Class	AP	AP50	AP25	AP	AP50	AP25	AP	AP50	AP25	AP	AP50	AP25
cabinet	0.002	0.008	0.039	0.024	0.081	0.153	0.036	0.118	**0.396**	**0.042**	**0.145**	0.346
bed	0.105	0.197	0.540	**0.200**	0.467	0.651	0.154	0.446	0.696	0.197	**0.540**	**0.806**
chair	0.000	0.001	0.027	0.138	0.239	0.434	0.475	0.689	0.814	**0.567**	**0.792**	**0.877**
sofa	0.066	0.240	0.462	0.157	0.398	0.533	0.172	0.369	0.684	**0.226**	**0.488**	**0.803**
table	0.027	0.061	0.160	0.154	0.324	0.428	0.207	0.361	0.593	**0.242**	**0.427**	**0.674**
door	0.019	0.037	0.070	0.041	0.073	0.108	0.142	0.304	0.429	**0.152**	**0.324**	**0.458**
window	0.015	0.023	0.023	0.020	0.031	0.037	0.113	0.258	0.423	**0.152**	**0.327**	**0.472**
bookshelf	0.013	0.024	0.187	0.077	0.198	**0.453**	0.075	0.175	0.423	**0.080**	**0.219**	**0.453**
picture	0.001	0.005	0.005	0.001	0.005	0.008	0.028	0.067	0.169	**0.044**	**0.109**	**0.198**
counter	0.007	0.032	0.216	**0.008**	**0.034**	**0.266**	0.001	0.004	0.094	0.001	0.008	0.097
desk	0.012	0.057	0.211	0.022	0.109	0.364	0.011	0.053	0.327	**0.031**	**0.142**	**0.499**
curtain	0.034	0.085	0.185	0.081	0.173	0.225	0.114	0.285	0.450	**0.174**	**0.399**	**0.542**
refrigerator	0.059	0.112	0.211	0.105	0.162	0.225	0.124	0.302	0.317	**0.185**	**0.421**	**0.441**
shower curtain	0.119	0.231	0.231	0.128	0.227	0.284	0.392	0.593	0.710	**0.402**	**0.643**	**0.749**
toilet	0.326	0.676	0.701	0.575	0.801	0.801	**0.636**	0.962	0.977	0.625	**0.965**	**0.980**
sink	0.048	0.130	0.328	0.054	0.135	0.307	0.094	0.294	0.397	**0.120**	**0.364**	**0.445**
bathtub	0.357	0.677	0.677	**0.319**	0.631	0.700	0.235	0.553	0.674	0.311	**0.708**	**0.794**
otherfurniture	0.004	0.010	0.039	0.021	0.052	0.107	0.061	0.154	0.283	**0.097**	**0.215**	**0.335**
average	0.068	0.145	0.239	0.118	0.230	0.338	0.171	0.333	0.492	**0.203**	**0.402**	**0.554**

Table 2. **Results on the ScanNet dataset [4].** Results on the validation set when using labels from SparseConvNet [14]. We show the instance labeling performance of the segmentation method in [14], connected component labeling on the [14] segmentation, our method with feature embedding (FE) only and our method with multi-task learning.

Method	Avg AP50	bathtub	bed	bookshelf	cabinet	chair	counter	curtain	desk	door	otherfurniture	picture	refrigerator	shower curtain	sink	sofa	table	toilet	window
MTML (Ours)	**0.549**	**1.00**	**0.81**	0.59	0.33	**0.65**	0.00	**0.82**	0.18	**0.42**	0.36	0.18	0.45	**1.00**	0.44	0.69	0.57	**1.00**	0.40
Occipital-SCS	0.512	**1.00**	0.72	0.51	**0.51**	0.61	0.09	0.60	0.18	0.35	0.38	0.17	0.44	0.85	0.39	0.62	0.54	0.89	0.39
3D-BoNet	0.488	**1.00**	0.67	0.59	0.30	0.48	0.10	0.62	**0.31**	0.34	0.26	0.13	0.43	0.80	0.40	0.50	0.51	0.91	**0.44**
PanopticFusion [34]	0.478	0.67	0.71	**0.60**	0.26	0.55	0.00	0.61	0.18	0.25	**0.43**	**0.44**	0.41	0.86	**0.49**	0.59	0.27	0.94	0.36
ResNet-backbone [28]	0.459	**1.00**	0.74	0.16	0.26	0.59	**0.14**	0.48	0.22	0.42	0.41	0.13	0.32	0.71	0.41	0.54	**0.59**	0.87	0.30
MASC [30]	0.447	0.53	0.56	0.38	0.38	0.63	0.00	0.51	0.26	0.36	0.43	0.33	**0.45**	0.57	0.37	0.64	0.39	0.98	0.28
3D-SIS [18]	0.382	**1.00**	0.43	0.25	0.19	0.58	0.01	0.26	0.03	0.32	0.24	0.08	0.42	0.86	0.12	**0.70**	0.27	0.88	0.24
Unet-backbone [28]	0.32	0.67	0.72	0.23	0.19	0.48	0.01	0.22	0.07	0.20	0.17	0.11	0.12	0.44	0.2	0.62	0.36	0.92	0.09
R-PointNet [53]	0.306	0.50	0.41	0.31	0.35	0.59	0.05	0.07	0.13	0.28	0.29	0.03	0.22	0.21	0.33	0.40	0.28	0.82	0.25
3D-BEVIS	0.248	0.67	0.57	0.08	0.04	0.39	0.03	0.04	0.10	0.10	0.03	0.03	0.10	0.38	0.13	0.60	0.18	0.85	0.17
Seg-Cluster	0.215	0.37	0.34	0.29	0.11	0.33	0.03	0.28	0.09	0.11	0.11	0.01	0.08	0.32	0.11	0.31	0.30	0.59	0.12
SGPN [49]	0.143	0.21	0.39	0.17	0.07	0.28	0.03	0.07	0.00	0.09	0.04	0.02	0.03	0.00	0.11	0.35	0.17	0.44	0.14
MaskRCNN proj	0.058	0.33	0.00	0.00	0.05	0.00	0.00	0.02	0.00	0.05	0.02	0.24	0.07	0.00	0.01	0.11	0.02	0.11	0.01

Table 3. **Results on the ScanNet 3D instance segmentation benchmark [4].** The table shows the AP50 score on individual semantic categories and the average score (sorted by avg AP50 score in descending order). Our method achieves the best average score.

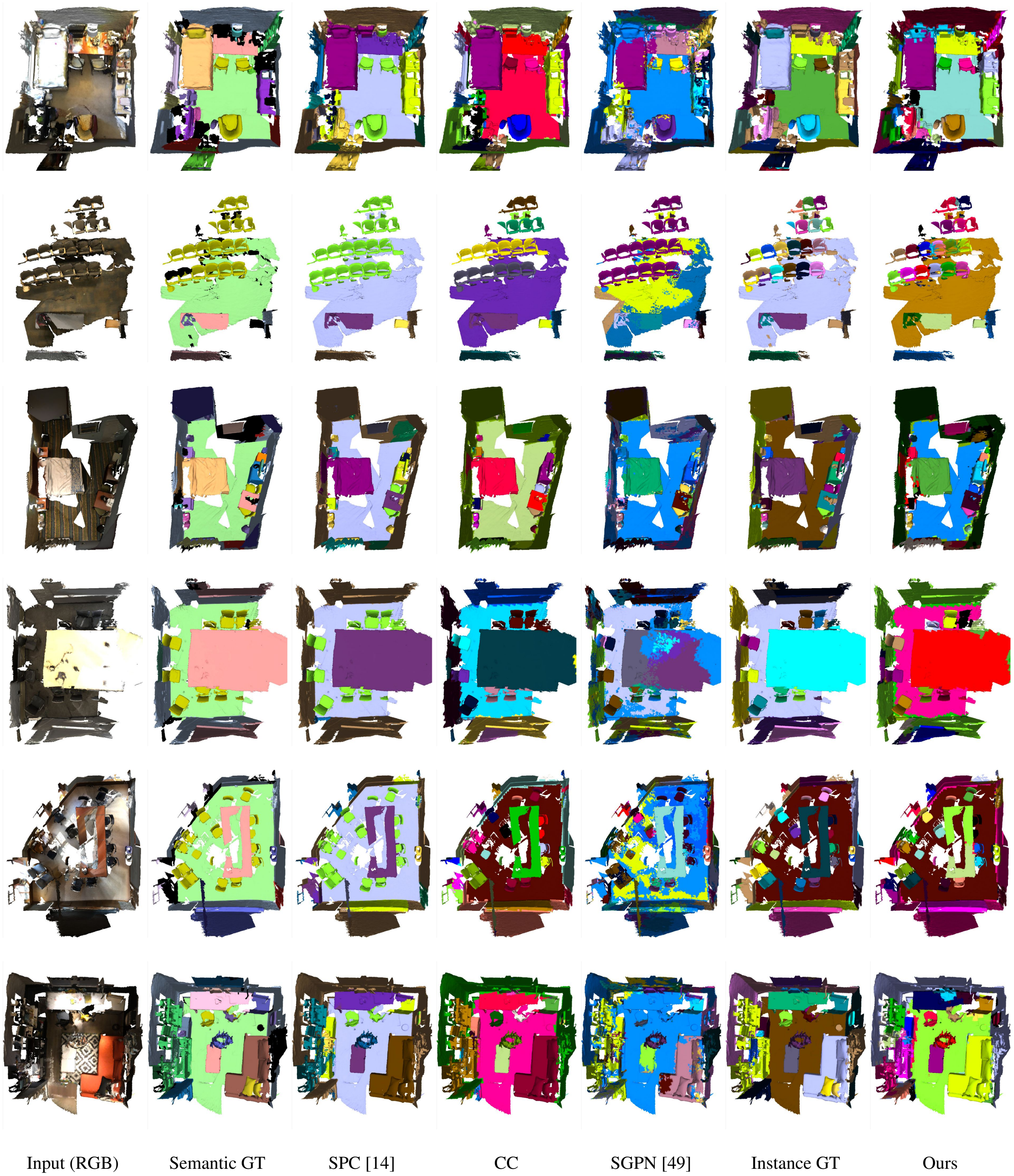


Figure 7. **Qualitative results of our method on the ScanNet validation dataset [4].** This figure shows the original input scene as a textured mesh, the semantic labeling results of SparseConvNet (SPC) [14] which we use as input and our instance labeling results as well as the semantic groundtruth (GT). We further show multiple 3D instance segmentation baselines: connected component (CC) labeling on the SPC semantic labeling, SPGN [49], and the groundtruth instance labels next to our labeling results.

much faster operation on the 3D scenes, and requires suubstantially less information (only 3D point cloud) to extract the 3D object instance segmentations.

5. Conclusion

We proposed a method for 3D instance segmentation of voxel-based scenes. Our approach is based on metric learning and the first part assigns all voxels belonging to the same object instance similar feature vectors which are in close vicinity. Conversely, voxels belonging to different object instances are assigned with features that are further apart from each other in the feature space. The second part estimates directional information of object centers which is used to score the segmentations results of the first part.

6. Acknowledgments

We are grateful for the support by competitive funding from King Abdullah University of Science and Technology (KAUST). This research was also supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00280. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017.

[2] B. D. Brabandere, D. Neven, and L. V. Gool. Semantic instance segmentation with a discriminative loss function. *CoRR*, abs/1708.02551, 2017.

[3] I. Cherabier, J. L. Schönberger, M. R. Oswald, M. Pollefeys, and A. Geiger. Learning priors for semantic 3d reconstruction. In *European Conference on Computer Vision (ECCV)*, September 2018.

[4] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] A. Dai and M. Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, pages 458–474, 2018.

[6] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Niener. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Proc. European Conference on Computer Vision (ECCV)*, pages 534–549, Cham, 2016. Springer International Publishing.

[8] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3150–3158, 2016.

[9] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy. Semantic instance segmentation via deep metric learning. *CoRR*, abs/1703.10277, 2017.

[10] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, January 1975.

[11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376, 2010.

[12] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *ArXiv e-prints*, April 2017.

[13] R. B. Girshick. Fast R-CNN. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[14] B. Graham, M. Engelcke, and L. van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *CVPR*, 2018.

[15] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, Nov 2017.

[16] Z. Hayder, X. He, and M. Salzmann. Boundary-aware instance segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[17] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *Proc. International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[18] J. Hou, A. Dai, and M. Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. *arXiv preprint arXiv:1812.07003*, 2018.

[19] S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. J. Davison, and A. W. Fitzgibbon. Kinectfusion: real-time dynamic 3d surface reconstruction and interaction. In *International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2011, Vancouver, BC, Canada, August 7-11, 2011, Talks Proceedings*, page 23, 2011.

[20] A. Khoreva, R. Benenson, J. H. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1665–1674, 2017.

[21] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009.

[22] S. Kong and C. C. Fowlkes. Recurrent pixel embedding for instance grouping. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9018–9028, 2018.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012.

[24] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *Proc. European Conference on Computer Vision (ECCV)*, pages 703–718. Springer, 2014.

[25] J. Lahoud and B. Ghanem. 2d-driven 3d object detection in rgb-d images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4622–4630, 2017.

[26] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4438–4446, 2017.

[27] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, and S. Yan. Proposal-free network for instance-level semantic object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.

[28] Z. Liang, M. Yang, and C. Wang. 3d graph embedding learning with a structure-aware loss function for point cloud semantic instance segmentation. *arXiv preprint arXiv:1902.05247*, 2019.

[29] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[30] C. Liu and Y. Furukawa. Masc: Multi-scale affinity with sparse convolution for 3d instance segmentation. *arXiv preprint arXiv:1902.04478*, 2019.

[31] S. Liu, J. Jia, S. Fidler, and R. Urtasun. SGN: sequential grouping networks for instance segmentation. In *Proc. International Conference on Computer Vision (ICCV)*, pages 3516–3524, 2017.

[32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[33] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, September 2015.

[34] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. *arXiv preprint arXiv:1903.01177*, 2019.

[35] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph.*, 32(6):169:1–169:11, 2013.

[36] P. H. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1990–1998, 2015.

[37] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.

[38] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788, 2016.

[39] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.

[40] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.

[41] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, pages 234–241, 2015.

[42] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. European Conference on Computer Vision (ECCV)*, 2016.

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[44] R. Socher, B. Huval, B. P. Bath, C. D. Manning, and A. Y. Ng. Convolutional-recursive deep learning for 3d object classification. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 665–673, 2012.

[45] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. A. Funkhouser. Semantic scene completion from a single depth image. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[46] F. Steinbrücker, C. Kerl, and D. Cremers. Large-scale multi-resolution surface reconstruction from RGB-D sequences. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 3264–3271, 2013.

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[48] K. Tateno, F. Tombari, I. Laina, and N. Navab. CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6565–6574, 2017.

[49] W. Wang, R. Yu, Q. Huang, and U. Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[50] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.

[51] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[52] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *arXiv preprint arXiv:1906.01140*, 2019.

[53] L. Yi, W. Zhao, H. Wang, M. Sung, and L. Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. *arXiv preprint arXiv:1812.03320*, 2018.

[54] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust tv-l1 range image integration. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[55] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. 2010.