Pacific Graphics 2018
H. Fu, A. Ghosh, and J. Kopf
(Guest Editors)

Volume 37 (2018), Number 7

Semantic Reconstruction: Reconstruction of Semantically Segmented 3D Meshes via Volumetric Semantic Fusion

Junho Jeon, Jinwoong Jung, Jungeon Kim, and Seungyong Lee

POSTECH

Abstract

Semantic segmentation partitions a given image or 3D model of a scene into semantically meaning parts and assigns predetermined labels to the parts. With well-established datasets, deep networks have been successfully used for semantic segmentation of RGB and RGB-D images. On the other hand, due to the lack of annotated large-scale 3D datasets, semantic segmentation for 3D scenes has not yet been much addressed with deep learning. In this paper, we present a novel framework for generating semantically segmented triangular meshes of reconstructed 3D indoor scenes using volumetric semantic fusion in the reconstruction process. Our method integrates the results of CNN-based 2D semantic segmentation that is applied to the RGB-D stream used for dense surface reconstruction. To reduce the artifacts from noise and uncertainty of single-view semantic segmentation, we introduce adaptive integration for the volumetric semantic fusion and CRF-based semantic label regularization. With these methods, our framework can easily generate a high-quality triangular mesh of the reconstructed 3D scene with dense (i.e., per-vertex) semantic labels. Extensive experiments demonstrate that our semantic segmentation results of 3D scenes achieves the state-of-the-art performance compared to the previous voxel-based and point cloud-based methods.

CCS Concepts

•***Computing methodologies*** → ***Reconstruction; Scene understanding;***

1. Introduction

Semantic segmentation is one of the challenging problems for high-level scene understanding, which has various applications such as autonomous robots and augmented reality. In recent years, semantic segmentation has been spotlighted and its performance has been drastically improved along with rapid advances of deep learning. While plausible semantic segmentation results could be obtained for 2D images of indoor as well as outdoor scenes, semantic segmentation of 3D scene models still remains a hard problem.

The success of 2D semantic segmentation is largely built upon the availability of huge labeled image datasets and advances in knowledge transfer techniques. For example, Microsoft COCO dataset [LMB*14] contains about 50K semantically annotated 2D images. Although the dataset size may not be enough to train a deep convolutional neural network with millions of parameters from scratch, knowledge transfer learning makes it possible to exploit the features learned from a huge number of images, such as ImageNet dataset [DDS*09], and finetune a network for semantic segmentation.

On the other hand, in the case of 3D semantic segmentation, it becomes more challenging to develop a deep learning based approach. Although several semantically annotated 3D scanned scene datasets have been recently released [DCS*17, HPN*16, SCH*16], their sizes are about hundreds to a thousand, which are not large enough to train complex neural networks. In addition, transfer learning is not easily applicable to finetune a 3D CNN for semantic segmentation as there is no such a huge general 3D dataset similar to ImageNet [DDS*09].

Despite these difficulties, a few approaches based on deep learning have been proposed for 3D semantic segmentation, which can be categorized into two groups: direct and indirect methods. To obtain a semantic map of the entire 3D scene, a direct method performs semantic label estimation directly on the 3D model, usually represented in a voxelized format [DCS*17] or point cloud [QSMG17, QYSG17, HWN18]. In contrast, an indirect method integrates 2D semantic segmentation predictions onto the 3D model, represented in the form of surfels [MHDL17] or a point cloud [HFL14, LDT*17]. Direct methods require a large-scale 3D labeled training dataset, which would be harder to be annotated than 2D data. In addition, semantic segmentation results of an indirect method based on surfels or a point cloud would need further processing to be used for reconstructed 3D meshes.

In this paper, we propose a novel framework that automatically generates semantically segmented triangle meshes from a RGB-D stream for a large-scale indoor scene by combining 2D semantic segmentation with 3D volumetric fusion. Differently from prior methods [DCS*17, HFL14, LDT*17, MHDL17] that use voxels, surfels, or a point cloud, our method utilizes a volumetric data

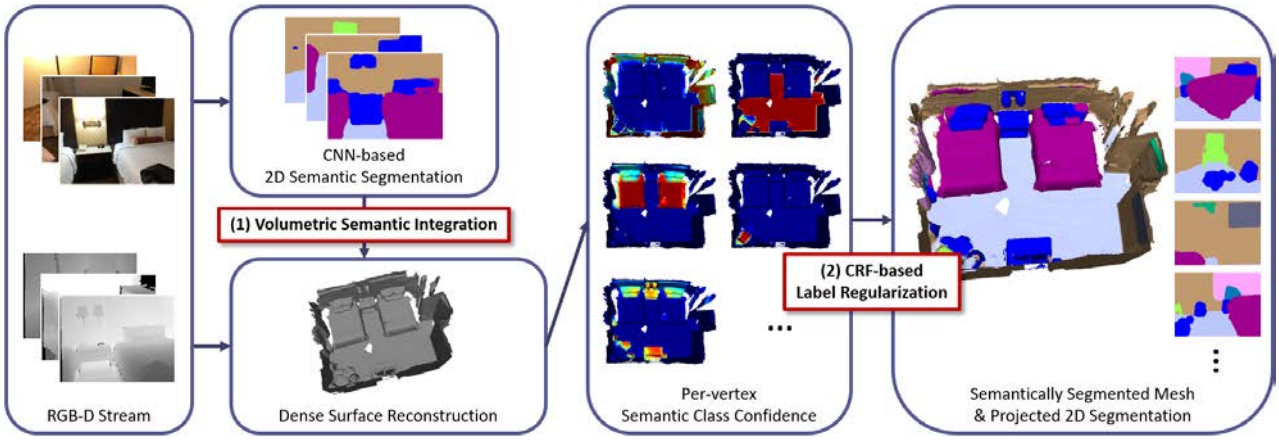


Figure 1: *Our pipeline for semantic segmentation of reconstructed 3D indoor scenes. 2D semantic segmentation results, as well as the geometry information, of the input RGB-D stream are fused into a volumetric representation, and then per-vertex semantic class confidences are extracted when the mesh is reconstructed from the volumetric representation. The per-vertex confidences are refined through CRF-based label regularization, generating the final semantic segmentation of the reconstructed mesh. Projections of the semantically labelled reconstructed mesh can be used for producing semantic segmentations of input frames, which are more accurate than the initial segmentations of the frames which have been fused into the volumetric representation.*

structure and a triangle mesh complementarily to integrate geometry and semantic information. The volumetric data structure enables us to efficiently integrate the high-level 2D semantic information from a state-of-the-art deep learning technique onto the reconstructed geometry, and the connectivity of the reconstructed triangle mesh is utilized for CRF-based semantic label regularization that enhances the segmentation result by restoring miss-labeled parts and removing noisy labels.

The key characteristics of our framework include:

- range-sensitive volumetric semantic fusion method that incrementally integrates CNN-based 2D semantic segmentation results onto the densely reconstructed 3D geometry.
- CRF-based semantic label regularization using the geometric and photometric information of the reconstructed triangle mesh to incorporate the global scene context.

Experimental results on various RGB-D streams of large-scale indoor scenes show that our method can precisely predict the dense (i.e., per-vertex) semantic labels for a reconstructed mesh without directly training a deep neural network on a 3D dataset. As applications of the segmentation results, we present 3D scene completion and manipulation, where semantic information is used for detecting and filling holes inside objects and transforming objects in the scene independently from others.

2. Related Work

RGB and RGB-D image semantic segmentation Deep convolutional neural networks (CNNs) have been successfully used for semantic segmentation of single RGB images [CPK*16, LMSR17, LSD15, NHH15, PLCD16, YK15]. For 2.5D RGB-D images, Long et al. [LSD15] trained and tested their fully convolutional networks (FCNs) on the NYU-Depth V2 dataset [NSF12]. Hazirbas et al. proposed FuseNet [HMDC16] that integrates the intermediate depth and color features using sparse and dense fusion. Park et al. proposed RDFNet [PHL17] that uses multi-modal feature fusion to incorporate the depth information into semantic segmentation and shows the state-of-the-art performance.

To integrate global context to the final semantic segmentation prediction, Conditional Random Fields (CRFs) can be adopted as a post-processing step. Krähenbühl and Koltun [KK11] proposed an efficient approximate algorithm for fully connected pairwise CRF to refine semantic segmentation results. It uses pairwise Gaussian edge potentials considering the distances based on pixel positions and appearances.

Full 3D semantic segmentation Several works have been proposed for semantic segmentation of scanned full 3D geometric data. Huang et al. [HY16] firstly proposed a method that uses a 3D CNN by voxelizing a point cloud into a regular grid. Dai et al. [DCS*17] released 15K semantically annotated triangle meshes, and trained a 3D CNN by voxelizing the meshes. Instead of a voxelized regular grid, recently proposed point cloud based methods [QSMG17, QYSG17, HWN18] directly estimate the class labels of an unordered set of points. RSNet [HWN18] achieved the state-of-the-art results by modeling local geometric dependencies.

To avoid direct training of 3D CNNs, indirect segmentation methods have been proposed. Lawin et al. [LDT*17] projects a 3D point cloud onto a set of synthetic 2D images, which are used to predict semantic labels of the projected points using a 2D CNN. McCormac et al. [MHDL17] proposed a 3D semantic mapping method that predicts semantic maps from RGB-D frames and fuses them onto the surfel data structure.

Although indirect 3D semantic segmentation methods [LDT*17, MHDL17] have similar pipelines that fuse 2D predictions onto a

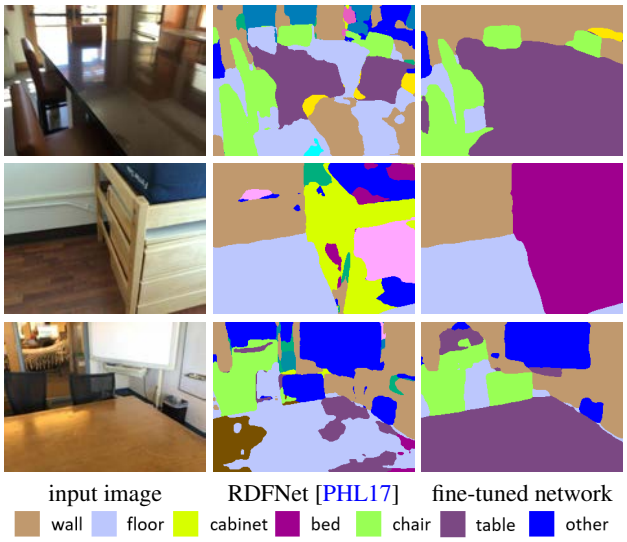


Figure 2: *Results of 2D CNN-based semantic segmentation. The network fine-tuned on ScanNet dataset [DCS*17] shows cleaner and more accurate segmentation results compared to the original RDFNet [PHL17].*

single geometry, our method differs from them in that we use a volumetric representation for intermediate fusion and produce a triangle mesh with per-vertex class labels as the output.

Dense 3D surface reconstruction After KinectFusion [NIH*11] was introduced, volumetric integration [CL96] of a signed truncated distance function (TSDF) becomes a common approach for reconstructing the geometry for RGB-D video based 3D scanning [CZK15, NZIS13, DNZ*17]. VoxelHashing [NZIS13] uses a hash-based volumetric structure to enable the reconstruction of a large-scale indoor scene. BundleFusion [DNZ*17] shows the state-of-the-art performance in real-time 3D reconstruction, which can handle incremental drifts in pose estimation using color features. Our framework uses BundleFusion [DNZ*17] and VoxelHashing [NZIS13] for pose estimation and geometry reconstruction, respectively.

RGB-D image dataset As consumer depth cameras become popular, several RGB-D image datasets have been published for indoor scene reconstruction and understanding. NYUDv2 dataset [NSF12] consists of semantically annotated thousands of RGB-D images, and SUN RGB-D dataset contains 10K images. ScanNet dataset [DCS*17] is now the largest RGB-D image dataset and consists of 2.5 million images. Especially, ScanNet contains a semantically labeled triangle mesh for each input RGB-D stream. In this paper, we use ScanNet dataset for experiments.

3. Our Framework

The overall process of our approach for 3D semantic segmentation is shown in Fig. 1. Our framework uses a RGB-D image stream as the input. During the semantic class confidence integration step, we apply CNN-based 2D semantic segmentation to the input RGB-D steam to obtain a class confidence map for each frame. Then the estimated class confidence maps are fused into a 3D volumetric data structure along with the geometry during the reconstruction process. In the semantic mesh generation step, we extract a 3D triangle mesh from the integrated volume data structure. At the same time, we extract per-vertex class confidences for object classes using a modified marching cube algorithm [LC87]. The per-vertex object class labels are finally determined by fully-connected CRF inference using the unary and pairwise potentials considering local and global context information.

Dense volumetric reconstruction In our work, we use BundleFusion [DNZ*17] to estimate the camera poses of the input RGB-D stream. BundleFusion [DNZ*17] robustly tracks the camera poses of incoming RGB-D frames and minimizes the accumulated drift of the geometry by globally optimizing the tracked camera poses based on geometric and photometric feature matching. We then perform the volumetric integration of geometry and semantic information using VoxelHasing [NZIS13]. It employs a sparse volumetric grid based on spatial hashing as the geometry representation which enables us to handle a large-scale indoor scene.

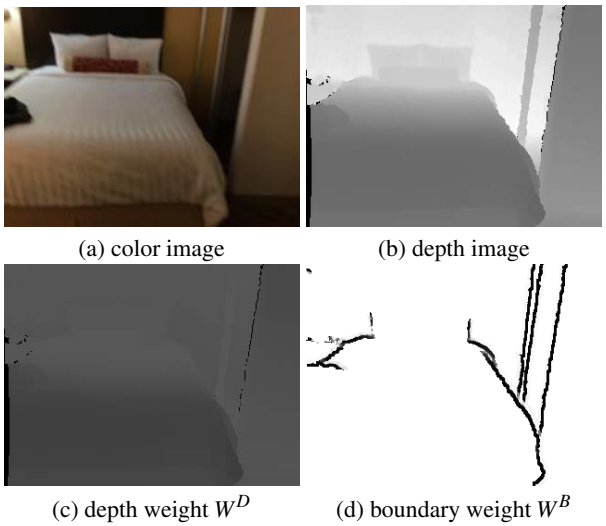
4. CNN-based 2D Semantic Segmentation

In this section, we present the details of our single image semantic segmentation step. Given a RGB-D stream, we apply a CNN-based semantic segmentation method to each frame. The network is fine-tuned on the annotated RGB-D images from ScanNet dataset [DCS*17] to improve the segmentation quality.

Semantic segmentation network Semantic segmentation of input frames is an independent component in our framework, and we could use any 2D image segmentation methods [CPK*16, LSD15] for the step. However, we found that depth features help better discriminate ambiguous semantic classes (e.g. floor and table top surface), and a RGB-D semantic segmentation method would work better. In this paper, we use RDFNet [PHL17] that effectively exploits multi-level RGB-D CNN features and learns the optimal fusion of multi-modal features. RDFNet shows the state-of-the-art accuracy for RGB-D semantic segmentation of indoor scenes tested on NYUDv2 dataset [NSF12].

Network transfer learning Although RDFNet [PHL17] reports the state-of-the-art performance on RGB-D semantic segmentation, it does not always produce satisfactory results depending on the property of the input scene. RDFNet is trained on NYUDv2 dataset [NSF12], many of whose images were captured moderately far from the target objects covering the whole scenes. On the other hand, ScanNet dataset [DCS*17] is constructed for dense surface reconstruction, and the images in the dataset were usually captured close to the target objects, containing only parts of large objects, such as beds or tables (Fig. 2). Furthermore, the RGB images may suffer from motion blurs as they were captured using hand-held RGB-D sensors. Consequently, the original RDFNet shows inferior performance on ScanNet dataset.

To compensate this limitation, we perform transfer learning



(a) color image (b) depth image

(c) depth weight W^D (d) boundary weight W^B

Figure 3: *Reliability weight maps for a RGB-D image. The depth-based accuracy weight W^D is the highest at the moderate distance, and the boundary misalignment weight W^B has low values around depth discontinuities, where a darker color means a lower weight.*

for RDFNet on ScanNet dataset. We extract 16K RGB-D images and their corresponding semantic annotations from 1041 training streams by sampling every 100th frame. Based on the author-provided pre-trained model of RDFNet, we fine-tune the entire network for 20 epochs. We use 20 major object classes in accordance with the original paper [DCS*17] of ScanNet dataset. Following the RDFNet paper [PHL17], random crop and horizontal flip are used for data augmentation to prevent the training from over-fitting. As shown in Fig. 2, the fine-tuned network shows robust semantic segmentation results, even when the input images are captured extremely close to the target objects.

Confidence vs. probability In this paper, we use the term *confidence* instead of *probability* to refer to the output values of a semantic segmentation network. It is known that modern complex CNNs may not be well-calibrated [GPSW17], which means the output values of a neural network cannot be directly interpreted as the correctness likelihoods for ground truths. Therefore, we treat the output of a semantic segmentation network as a *confidence* of each object class.

5. Semantic 3D Reconstruction

In this section, we present the details of our semantic 3D reconstruction framework. Although naive implementation of the framework could be rather straightforward, it would produce noisy and less inaccurate segmentation results, not remedying inherently incomplete results of 2D semantic segmentation. To obtain high-quality semantic segmentation of the reconstructed mesh, we introduce and use the reliability of confidence values for improving semantic class confidence integration and CRF-based label regularization.

5.1. Semantic class confidence integration

Single-view 2D semantic segmentation predicts the segmentation result of each view independently, and it often produces noisy and unstable results, as shown in Fig. 2. In 3D reconstruction techniques [DNZ*17, CZK15, NIH*11, NZIS13], the reconstructed geometry becomes smooth and clean as the multiple noisy depth frames are integrated incrementally. To resolve the noise and uncertainties in the single-view 2D semantic segmentation results, we use incremental semantic fusion, similarly to geometry refinement.

During surface reconstruction, each voxel holds the truncated signed distance function (TSDF) value to represent its surrounding local geometry. To integrate the semantic information of the incoming RGB-D stream, we also store the confidences of object classes at each voxel to represent which object class the voxel may belong to. The integration of class confidences follows a similar way to the original TSDF update of VoxelHashing [NZIS13].

Each pixel p of the input image has its corresponding voxel o according to the estimated camera pose and the pixel coordinates. Then given a predicted class confidence map from 2D semantic segmentation of the t-th frame, we incrementally update the class confidences of the corresponding voxels $C_t(o)$ by taking the weighted running averages [NIH*11], defined as follows:

$$C_t(o) = \frac{W_{t-1}(o)C_{t-1}(o) + W_{F_t}(p)C_{F_t}(p)}{W_{t-1}(o) + W_{F_t}(p)}, \tag{1}$$

$$W_t(o) = W_{t-1}(o) + W_{F_t}(p), \tag{2}$$

where $C_{t-1}(o)$ and $W_{t-1}(o)$ are the *integrated* class confidence and reliability weight of voxel o, respectively. $C_{F_t}(p)$ and $W_{F_t}(p)$ are respectively the class confidence and reliability of pixel p in the t-th frame. Note that a class confidence map from 2D semantic segmentation contains confidence values for *all* labels at each pixel, whose sum is one. So $C_t(o)$ is a d-dimensional vector, where d is the number of different labels in semantic segmentation.

Although we use the state-of-the-art 2D semantic segmentation method, the semantic prediction result still contains noise and inaccurate confidence values. These artifacts vary depending on input frames, and we define the per-pixel class reliability $W_{F_t}(p)$ of the semantic segmentation result and use it for adaptively integrating the semantic predictions. $W_{F_t}(p)$ is a d-dimensional vector determined by local geometry information as:

$$W_{F_t}(p) = W^D_{F_t}(p) W^B_{F_t}(p), \tag{3}$$

where $W^D_{F_t}(p)$ is the depth-based accuracy weight and $W^B_{F_t}(p)$ is the boundary misalignment weight (Fig. 3). $W^D_{F_t}$ and $W^B_{F_t}$ are also d-dimensional vectors, and the product of two d-dimensional vectors is defined as element-wise multiplication in this paper.

Depth-based accuracy weight $W^D_{F_t}$ A CNN has a fixed receptive field size according to its network structure, and the accuracy of semantic prediction varies depending on the object scale in the input image. To reflect this, $W^D_{F_t}(p)$ is defined as a function of the depth value of pixel p. Recall that the object size in an image changes with the distance from the camera. We evaluated RDFNet [PHL17], our choice for the 2D semantic segmentation method used in our

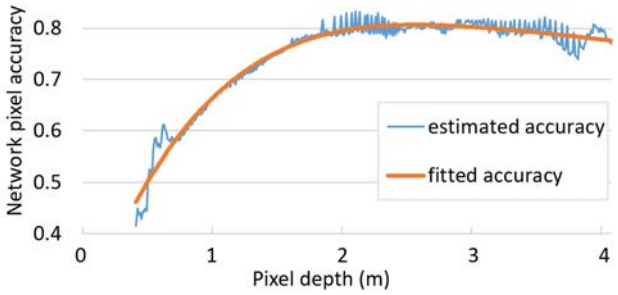


Figure 4: *Semantic prediction accuracy along with the input pixel's depth value (blue), and its 4th degree polygonal fitting (red). Too close capturing of an object reduces the accuracy.*

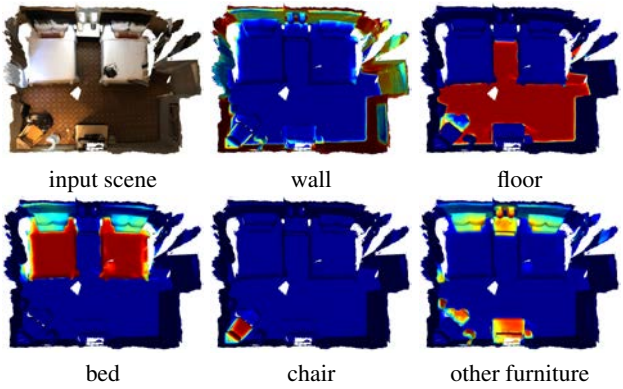


Figure 5: *Class confidence visualization for five major classes (red: high, blue: low).*

framework, on the validation set of ScanNet dataset [DCS*17], and measured the accuracies of semantic predictions for different depth values in *1cm* intervals (Fig. 4). Then we fitted a 4th degree polynomial to approximate the prediction accuracy function with respect to the depth value. The function value ranges from 0 to 1, and $W^D_{F_t}(p)$ consists of d occurrences of the value.

Boundary misalignment weight $W^B_{F_t}$ Semantic segmentation results of a RGB-D stream mainly depend on the color images while depth information is supplementarily used for distinguishing ambiguous labels. However, even with a well-calibrated RGB-D sensor, there still exist misalignments between the color and depth images. Especially, such misalignments may become large along the boundaries between foreground objects and the background room layout (i.e., wall and floor) due to large depth differences. When the semantic predictions are integrated with geometry reconstruction, these misalignments would introduce mis-labeled voxels around object boundaries in the scene.

To address this issue, we introduce the boundary misalignment weight $W^B_{F_t}$, which gives a low reliability to the room layout classes (wall and floor) for the foreground pixels around depth discontinuities. Specifically, we first detect the depth edge pixels p which contain depth differences larger than *30cm* in the neighboring 7×7 windows, and then define $W^B_{F_t}(p)$ as:

$$W^B_{F_t}(p) = \frac{1}{1+\exp(-\alpha(r(p)-\beta))} \qquad r(p) = \frac{d(p)-d_{min}}{d_{max}-d_{min}} \tag{4}$$

where $d(p)$ is the depth of pixel p, and d_{min} and d_{max} are the minimum and maximum depths in the window centered at p, respectively. $W^B_{F_t}$ is computed with Eq. (4) only for the two room layout classes, i.e., wall and floor, and remains 1 for all other classes, preventing foreground pixels from being labeled as layout classes. Note that in Eq. (4), $W^B_{F_t}(p)$ becomes small when the depth of pixel p is small, meaning that p tends to belong to a foreground object. For non-depth edge pixels p, $W^B_{F_t}$ is 1 for all semantic classes. In the experiments, we set $\alpha = 8$, $\beta = 0.5$.

Fig. 3 shows weight maps $W^D_{F_t}$ and $W^B_{F_t}$ that visualize the per-pixel reliability used for adaptively integrating a class confidence map during the reconstruction process. Moreover, at the end of reconstruction, the integrated reliability weight $W_t(o)$ of each voxel o represents the reliability of the integrated class confidence $C_t(o)$ of o, and we use it in the CRF-based label regularization.

Besides the depths of pixels and the boundary misalignments, other factors, e.g., physical object sizes, could be related to the class reliability. In our work, however, we consider the two factors only as they are most intuitive and can be easily estimated.

5.2. CRF-based semantic mesh generation

After integrating geometry and semantic information into a volumetric representation, we generate a semantically labelled triangle mesh from the volume using the marching cube algorithm [LC87]. By modifying the original marching cube algorithm, we assign the object class confidence to each vertex by linearly interpolating the confidences integrated at neighboring voxels. Fig. 5 shows the reconstructed mesh and assigned class confidences for five major object classes, where red and blue vertex colors represent high and low confidences of a vertex for an object class, respectively.

CRF-based label regularization Our semantic reconstruction effectively reduces the noise and uncertainty by fusing multiple semantic predictions. However, 2D semantic segmentation only considers local appearance and geometry in a limited field-of-view of an input frame. Simply taking the maximum value from the integrated class confidences $C(x)$ to determine the class label of a vertex x would produce a noisy segmentation result. To incorporate the global context of the reconstructed scene into semantic segmentation, we use conditional random field (CRF), which is a common approach to refine the output of a CNN in 2D semantic segmentation. Prior 3D semantic segmentation methods [HFL14, MHDL17] also used CRFs on a point cloud or surfels to regularize the semantic map. In contrast, we apply CRF to the reconstructed 3D triangle mesh to determine the final semantic labels of vertices.

We construct a fully connected CRF and use the mean-field approximation algorithm [KK11] to efficiently solve it. In the CRF construction, we treat a 3D vertex as a graph node in the field. In a fully connected CRF, each node is connected to every other node

no matter how far it is, and so we can consider the local and global scene contexts simultaneously for semantic mesh segmentation.

The labeling status of a complete CRF graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be formulated as the Gibbs energy $E(\mathbf{x})$, where $\mathbf{x} = \{x_i\}$ denotes a labeling set of all vertices $\{v_i\} \in \mathcal{V}$. The energy $E(\mathbf{x})$ is defined as combination of the unary potential ψ_u and the pairwise potential ψ_p:

$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j), \tag{5}$$

where i and j are vertex indices.

Reliability-based unary potential ψ_u Basically the unary potential $\psi_u(x_i)$ is defined as a log-probability of a given class label for a vertex v_i which comes from the result of volumetric semantic fusion. To obtain the class probability, we use the integrated object class confidences $C(v_i)$, additionally exploiting the prior knowledge on the input scene. We assume that there exists only one floor in the scene, and ignore the floor class confidence of the vertices above the detected floor. We first estimate the 3D plane of the floor by applying RANSAC [FB87] to the vertices which have high confidences on the floor class. We then reduce the confidence on the floor class to be zero for the vertices away from the floor plane.

We also use the integrated reliability weight of each vertex as the certainty of the object class confidence for the unary potential. For example, the object class confidences should be adjusted to follow the uniform distribution if the reliability weight is *0*, and the confidences should keep the original distribution when the reliability is *1*. In our implementation, we use the following equation to adjust the confidence distribution:

$$C'(v) = \lambda_r W(v) C(v) + (\mathbf{1} - \lambda_r W(v)) p_u, \tag{6}$$

where $C(v)$ is the integrated confidence of vertex v and $W(v)$ is the averaged reliability, i.e., the integrated reliability divided by the number of accumulated frames. $\mathbf{1}$ is a d-dimensional vector, all of whose components equal to one. p_u is the uniform probability distribution. λ_r is a parameter to modulate the effect of the reliability, and we set λ_r as 1.5 in the experiments as the maximum value of $W(v)$ was about 0.6.

Pairwise potential ψ_p The pairwise potential $\psi_p(x_i, x_j)$ consists of three bilateral kernels:

$$\begin{aligned} \psi_p(x_i, x_j) &= \mu(x_i, x_j) \left(w_a k_a(x_i, x_j) + w_n k_n(x_i, x_j) + w_s k_s(x_i, x_j) \right), \\ \mu(x_i, x_j) &= \begin{cases} 1, & \text{if } x_i \neq x_j \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{7}$$

where ψ_p introduces a high penalty to differently labeled but similar neighboring nodes. The similarity between nodes are defined by following three kernels.

The appearance kernel $k_a(x_i, x_j)$ lets geometrically nearby vertices with similar appearances have the high similarity:

$$k_a(x_i, x_j) = \exp\left(-\frac{|p_i - p_j|^2}{2\theta_p^2} - \frac{|c_i - c_j|^2}{2\theta_c^2} \right), \tag{8}$$

where p_i and c_i are 3D vertex position and color of vertex v_i, respectively. Similarly, the surface smoothness kernel $k_n(x_i, x_j)$ enforces locally consistent predictions on smooth surfaces by taking into account the positions and surface normals:

$$k_n(x_i, x_j) = \exp\left(-\frac{|p_i - p_j|^2}{2\theta_p^2} - \frac{|n_i - n_j|^2}{2\theta_n^2} \right), \tag{9}$$

where n_i is the surface normal vector of vertex v_i.

Lastly we formulate the semantic similarity kernel $k_s(x_i, x_j)$ between two vertices utilizing the confusion matrix of 2D semantic segmentation network. A confusion matrix encodes how much each object class can be confused with other classes. For a vertex that has the maximum integrated confidence value on class k, we interpret the k-th row of the confusion matrix as a semantic feature.

$$k_s(x_i, x_j) = \exp\left(-\frac{|p_i - p_j|^2}{2\theta_p^2} - \frac{|s_i - s_j|^2}{2\theta_s^2} \right), \tag{10}$$

$$s_i = \mathcal{M}_{(k,:)}, \quad \text{(the k-th row of matrix } \mathcal{M}\text{)}$$

$$k = \text{argmax}_m C^m(v_i), \tag{11}$$

where $\mathcal{M}$ is the confusion matrix and $C^m(v_i)$ is the confidence of vertex v_i on object class m. By incorporating the confusion matrix to define the similarity of the confidences of two vertices, instead of directly comparing the classes with the maximum confidence values, different but similar classes (e.g., table and desk) can have higher values in the semantic similarity kernel k_s, encouraging the corresponding nodes to have the same label finally by CRF regularization even though their integrated confidences are not accurate. Similar to the depth-based accuracy weight in Section 5.1, we evaluate RDFNet [PHL17] on ScanNet validation set to obtain the confusion matrix $\mathcal{M}$.

The Gibbs energy $E(\mathbf{x})$ can be efficiently minimized using a mean field approximation algorithm proposed by Krähenbühl and Koltun [KK11]. The algorithm inferences the result in linear time in the number of nodes. Therefore, despite the fact that our dense surface reconstruction process produces a mesh with millions of vertices, the CRF-based mesh segmentation only takes a few seconds.

In the following experiments, based on previous literatures [HFL14, KK11, MHDL17], we use Gaussian parameters $\theta_p = 0.1$, $\theta_c = 0.1$, $\theta_n = 0.1$ and $\theta_s = 0.3$. We also empirically set the balance parameters as $w_a = 10$, $w_n = 10$ and $w_s = 3$.

6. Experimental Results

6.1. Experimental setting

We conducted experiments to evaluate our method using ScanNet dataset [DCS*17], which is a RGB-D stream collection of large scenes captured by Structure sensor [Occ16] and semantically annotated by crowd workers. All experiments were performed on a PC with an Intel i7-6700K 4.0GHz CPU, 32GB RAM and NVidia GTX 1080ti GPU. Publicly available implementations of dense surface reconstruction [DNZ*17], 2D semantic segmentation [PHL17], and CRF inference [KK11] were used with proper modifications to build our framework. For RDFNet, we use single-scale prediction rather than multiple-scale ensemble for efficiency.

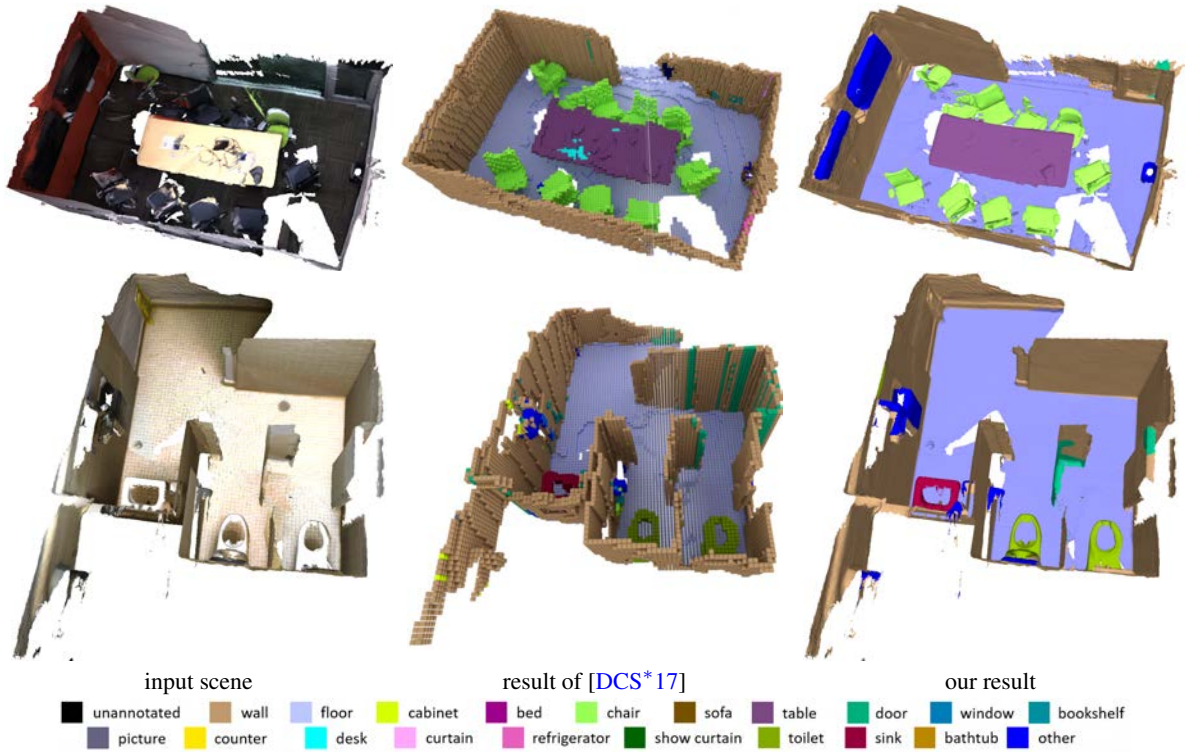


Figure 6: *Comparison with the results of Dai et al. [DCS*17]. We use the same false coloring as [DCS*17].*

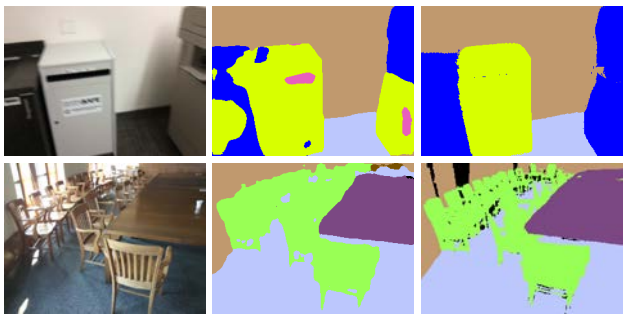
Computation time Dense 3D reconstruction and volumetric fusion of 20 class semantic labels run in near real-time, but 2D segmentation takes about 0.3s per frame, preventing our framework from achieving real-time performance. In our current implementation using a single GPU, both components cannot be run at the same time due to the limited resource of the GPU. Mesh extraction and CRF regularization take about tens of seconds for a scene, which depends on the size of the reconstructed mesh.

6.2. Qualitative evaluation of segmentation results

Figs. 6 and 10 show the final results of our semantic reconstruction framework. For a given RGB-D stream, our framework produces a high-quality dense 3D triangle mesh with semantically labeled vertices. As shown in the figures, the mesh is precisely segmented along the object class boundaries, such as between floor and wall.

Visual comparison with voxel-based method Dai et al. [DCS*17] proposed a 3D CNN-based semantic labeling algorithm using a low-resolution volume as the input and producing voxel labels as the output. Here we visually compare the labeling results of RGB-D streams with [DCS*17]. As shown in Fig. 6, our segmentation results show comparable quality in the perspective of labeling large structural object classes, such as floor, wall, and bed. Moreover, as our framework works on a dense triangle mesh, our results can distinguish the labels of small-scale objects that cannot be adequately handled by low-resolution volumes.

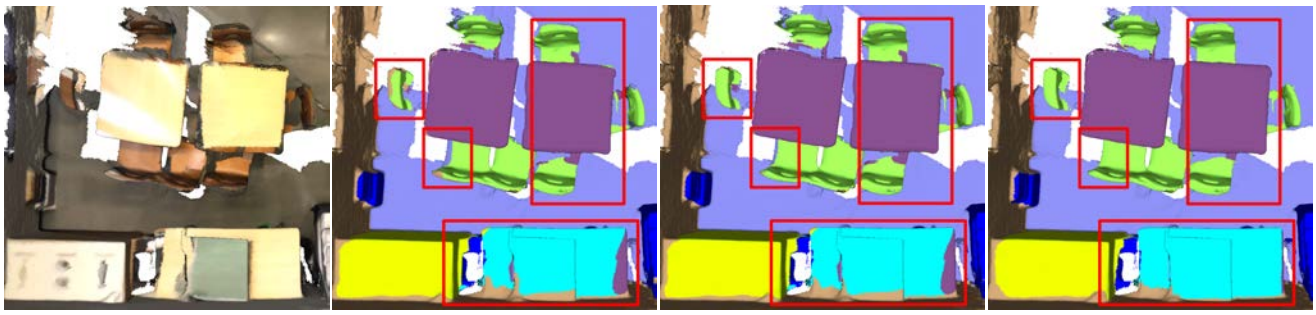
Projected semantic segmentation images Since we have camera parameters of the input frames, we can project the semantically segmented mesh using the parameters to obtain 2D segmentation results. As shown in Fig. 7b, outputs of the 2D semantic segmentation network (fine-tuned RDFNet [PHL17]) contain large mislabeled regions as the network only considers local content of a given frame. In contrast, our projected segmentation results show remarkably enhanced quality and details thanks to the multiple-frame integration and CRF regularization process (Fig. 7c).



(a) color image (b) single-image result (c) projected result

Figure 7: *Comparison between single-image semantic segmentation results and the projected results of our semantically segmented 3D meshes. Projection of missing geometry may introduce unlabeled pixels (black color).*

Component analysis We proposed adaptive integration and CRF regularization to exploit the characteristics of RDFNet and the



(a) input scene (b) naive integration (c) adaptive integration, no CRF (d) final result

Figure 8: *Experiments on the effects of two major components in our framework.*

global context of the scene. To evaluate the effects of these components, we test the framework with and without them. Fig. 8 shows the results of four cases of component analysis where each component is used or not. As expected, adaptive integration reduces mislabeled vertices around the boundaries of objects, such as chair and table, and CRF regularization drastically reduces small noisy labels.

6.3. Quantitative experiments

In addition to visual comparison, we quantitatively evaluate our semantic segmentation results using the semantic annotation of vertices in ScanNet dataset [DCS*17]. For the test, we used ScanNet test dataset that consists of 312 RGB-D streams in total.

Evaluation on ScanNet dataset Since 3D reconstruction produces different meshes according to the parameters, the ground truth mesh in ScanNet dataset and our result mesh may not share the same geometric structure (i.e., vertex and edge connectivity). Consequently, we cannot directly compare the vertex labels of our result mesh to the vertex labels in ScanNet dataset.

To evaluate the segmentation results independently of the mesh structures, we treat a mesh as a point cloud by ignoring the edges. Following the approach described in [HWN18], we voxelize the result and the ground truth point clouds by merging the points in each voxel, where majority voting is used to decide the label of a voxel when several points belong to the voxel. Then we measure the global accuracy of the entire dataset by comparing the result voxel labels with the ground truths for all scenes. If a voxel is empty or not annotated, it is not counted in the evaluation.

Table 1 shows that our framework could achieve global accuracy of 80.5% over the 312 test scenes. It also shows accuracies of different settings of our framework and comparison to the voxel-based semantic labeling of [DCS*17]. Note that [DCS*17] voxelizes the scene into a low-resolution sparse grid and labels each grid cell. In contrast, we densely label each vertex with a semantic class, which is a relatively hard problem setting. In that sense, direct comparison between our results and [DCS*17] with the voxelization would not be fair although still our method shows a higher accuracy. Our adaptive integration and CRF regularization mainly address the mislabeled and noisy vertices around object boundaries,

Configurations	Accuracy
Voxel-based labeling [DCS*17]	73.00%
Naive integration and without CRF	79.02%
Adaptive integration and without CRF	79.28%
Naive integration with CRF	79.79%
Adaptive integration and with CRF	79.86%

Table 1: *Global accuracy of semantic segmentation on ScanNet. For the naive integration, we use a uniform weight for all labels in per-pixel reliability W_{F_i}.*

as shown in Fig. 8. These improvements are clear in terms of visual quality but may not introduce significant changes on the global accuracy.

Comparison with point-based methods RSNet [HWN18] mentioned that the global accuracy is not enough for evaluating segmentation results on highly unbalanced ScanNet dataset. For better evaluation of our results, we measured the mean intersection over union (mIOU) and mean accuracy (mAcc), as in [QSMG17, QYSG17, HWN18]. Table 2 shows our method achieved the state-of-the-art results on ScanNet dataset. Compared to RSNet with RGB information, ours improves mIOU and mAcc by 2.84% and 15.30%, respectively. Table 2 also shows IOU of each category.

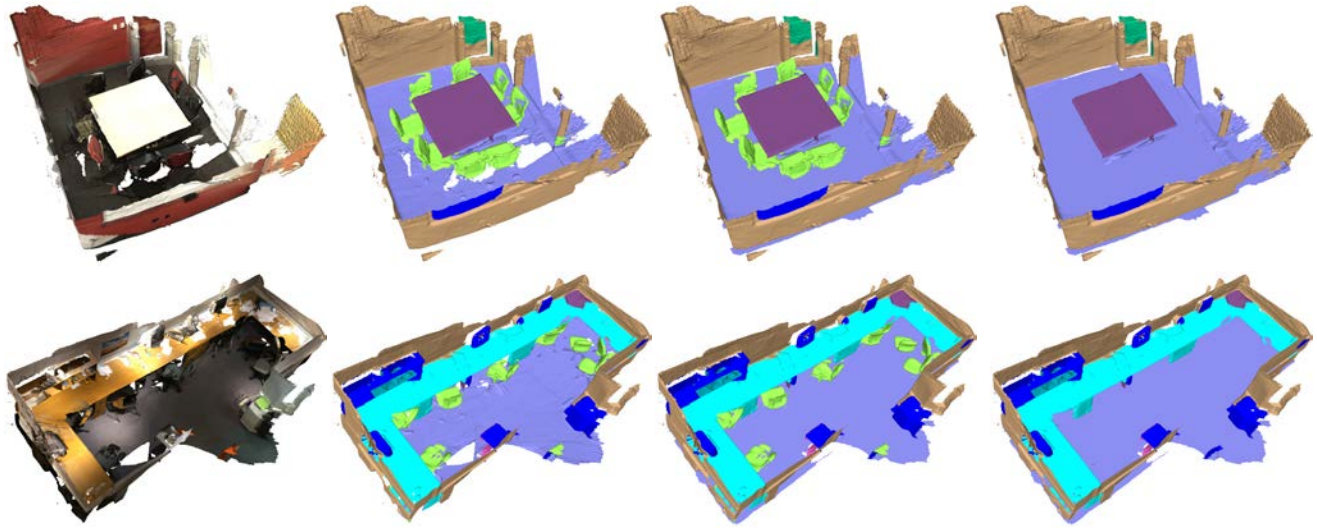
Evaluation on projected 2D segmentation We also evaluate the performance of our semantic integration by comparing the results against the single image 2D semantic segmentation results, which have been used for volumetric fusion. We measure three types of accuracy metrics (pixel accuracy, mean accuracy [LSD15], and mean IOU [EVGW*10]) on 53K semantic segmentation results in total by projecting every 10th frame in the 312 ScanNet test scenes. The results in Table 3 are matched with the visual comparisons in Fig. 7, and our semantic reconstruction improves the segmentation accuracy over the original 2D semantic segmentation.

6.4. 3D Scene completion and manipulation

As applications of our 3D semantic segmentation, we demonstrate 3D scene completion and manipulation. Single surface reconstruction using a RGB-D stream cannot build a complete geometry of

Method	*mIOU*	*mAcc*	wall	floor	cabinet	bed	chair	sofa	table	door	window
PointNet [QSMG17]	14.69	19.90	69.44	88.59	4.99	17.96	35.93	32.79	32.78	0.00	0.00
PointNet++ [QYSG17]	34.26	43.77	77.48	92.50	23.81	51.32	64.55	52.27	46.60	2.02	3.56
RSNet [HWN18]	39.35	48.37	79.23	94.10	31.29	55.95	64.99	55.41	51.04	3.00	8.75
RSNet w/ RGB [HWN18]	41.16	50.34	79.38	94.21	30.06	53.09	63.65	51.06	48.67	15.32	15.67
Ours	**44.00**	**65.64**	66.90	80.67	31.76	52.66	64.01	58.39	51.64	30.93	21.10
Method	bookshelf	picture	counter	desk	curtain	refridg	shower	toilet	sink	bathtub	others
PointNet [QSMG17]	3.18	0.00	5.09	2.63	0.00	0.00	0.00	0.00	0.00	0.17	0.13
PointNet++ [QYSG17]	52.93	0.00	20.04	12.69	32.97	18.51	27.43	31.37	30.23	42.72	2.20
RSNet [HWN18]	53.02	0.95	22.72	34.53	6.78	37.90	29.92	54.16	34.84	49.38	18.98
RSNet w/ RGB [HWN18]	53.67	4.30	20.90	35.27	8.30	39.76	24.36	63.20	41.00	60.37	20.98
Ours	31.21	7.38	24.07	26.18	30.36	56.34	23.63	73.37	46.26	69.74	33.33

Table 2: *Quantitative results on ScanNet dataset. We report mean IOU and mean Accuracy of all labels as well as IOU for each label. Measurements of previous methods are from [HWN18].*



(a) original scene (b) semantic segmentation (c) completed scene (d) completed & manipulated scene

Figure 9: *Geometric hole-filling & scene manipulation (removing the chairs).*

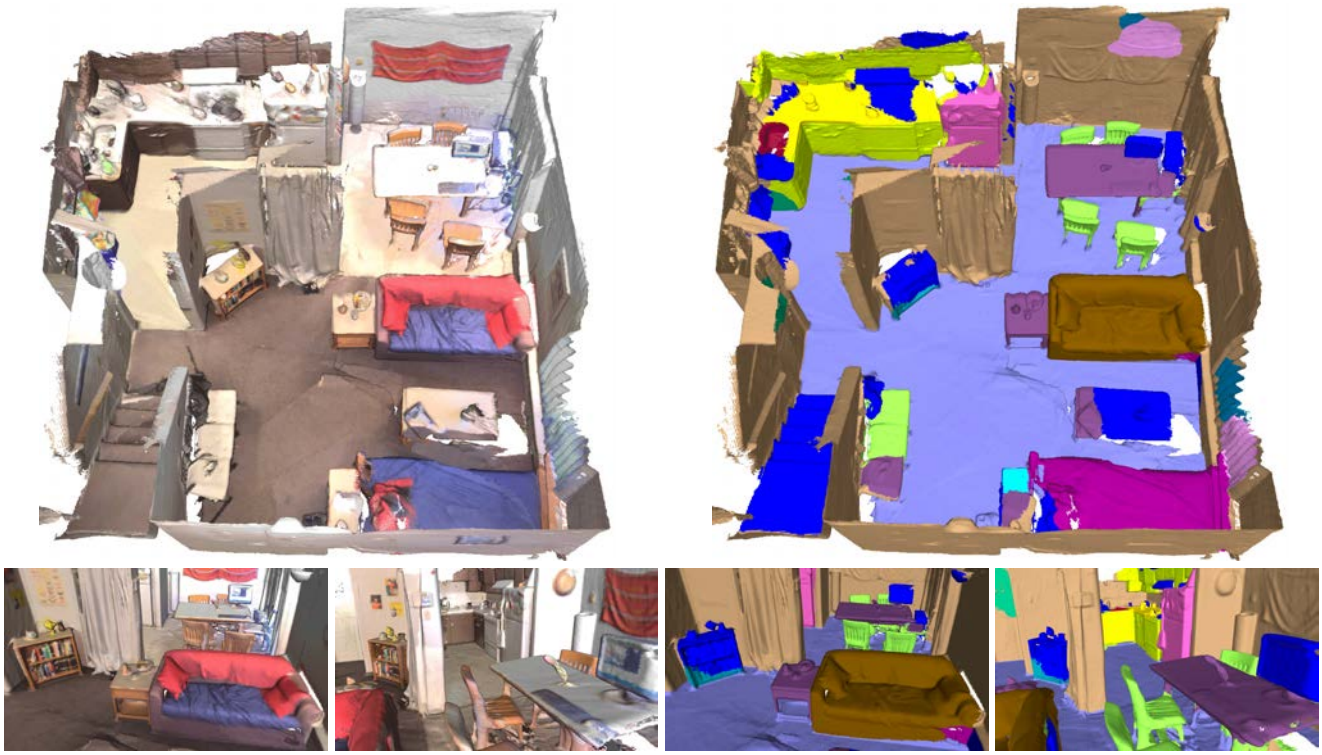
Method	pixel acc.	mAcc.	mIOU
single image segmentation (original) [PHL17]	60.44	47.32	29.34
single image segmentation (fine-tuned on ScanNet)	73.55	59.82	45.60
projected segmentation	77.18	63.20	50.69

Table 3: *Accuracy comparison of 2D segmentation results and our projected segmentations, where the ground truths are obtained using projection images of ScanNet [DCS*17] test scenes.*

the scene due to uncaptured regions. For example, a floor region under the desk or bed cannot be reconstructed as it is not visible from any input frame. This incomplete geometry could limit the usage of the reconstructed scene model in various applications, such as virtual reality and interior redesign. For example, removing or moving objects on the floor would reveal the uncaptured regions, which we call geometric holes.

Semantic-aware scene completion Geometric holes of the reconstructed mesh can be easily filled using semantic information. Dense 3D reconstruction algorithms such as BundleFusion [DNZ*17] and VoxelHashing [NZIS13] generate a single huge connected geometry without any semantic information, and it is not straightforward to detect scene structures, such as wall, floor, and ceiling. In contrast, our method generates a semantically segmented mesh, and we can easily estimate the structures by grouping vertices with the same semantic labels.

For example, we can fill the geometric holes on the floor with a few simple steps. First, we fit a 3D plane to the floor by performing RANSAC [FB87] on the floor vertices. We subdivide the estimated plane into a 2D grid and project all vertices with floor and wall class labels onto the plane, finding unoccupied grid cells,



reconstructed scene our segmentation result

Figure 10: *Semantic segmentation results of large-scale complex scenes reconstructed using thousands of RGB-D frames.*

which correspond to geometry holes. We can then extract the hole boundary using a simple contour tracing algorithm [SA85]. Finally, re-triangulating the grid cells inside the hole boundary gives us a clean, hole-filled floor mesh. As shown in Fig. 9c, our geometric hole-filling algorithm effectively recovers the missing parts on the floor, which has not been captured in the reconstruction process.

3D Scene manipulation As we have determined the semantic labels of all the vertices, we can easily divide the entire reconstructed mesh into separate meshes for different object classes. Then we can manipulate the individual object meshes independently from others, e.g., by applying 3D transforms, as shown in Fig. 9d. Note that in the example, objects on the floor can be freely moved without revealing any holes as we have already restored the complete floor plane in our semantic-aware scene completion step.

7. Conclusions

This paper introduced a novel framework that automatically generates a semantically segmented triangular mesh from a RGB-D video. Our method exploits the recent successes of deep neural networks on semantic segmentation of images by adaptively integrating 2D semantic predictions through volumetric fusion. In addition, our CRF-based semantic label regularization produces a more robust segmentation result by incorporating global scene context using geometric and photometric information.

Our method does not require a labeled 3D training dataset, which would be harder to obtain than a 2D dataset. In addition, advances of 2D semantic segmentation can be readily incorporated for improving 3D segmentation in our framework. For example, 3D instance segmentation could be made possible with our approach as high quality 2D instance segmentation becomes available.

Limitation and future work Our framework contains 3D reconstruction and RGB-D image segmentation, and any failure on each process will lower the quality of the results. Especially, camera tracking failure or drifting during the reconstruction may introduce erroneous results. Noisy and inaccurate results of 2D semantic segmentation are less critical as our method compensates the errors by integrating segmentation results from multiple frames (Fig. 7). Our future work includes real-time semantic reconstruction of 3D scenes that can provide integrated semantic information, as well as geometry and color, of the reconstructed scene during the capturing process, probably based on multi-GPU implementation. It would also be interesting to investigate on an effective way for incorporating semantically segmented meshes into semantic modeling [CLW*14, LGW17].

Acknowledgements We appreciate the constructive comments from the reviewers. This work was supported by the Ministry of Science and ICT, Korea, through Giga Korea grant (GK18P0300), IITP grant (IITP-2015-0-00174), and NRF grant (NRF-2017M3C4A7066317).

References

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 303–312. 3

[CLW*14] CHEN K., LAI Y., WU Y.-X., MARTIN R. R., HU S.-M.: Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information. *ACM Transactions on Graphics 33*, 6 (2014). 10

[CPK*16] CHEN L.-C., PAPANDREOU G., KOKKINOS I., MURPHY K., YUILLE A. L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915* (2016). 2, 3

[CZK15] CHOI S., ZHOU Q.-Y., KOLTUN V.: Robust reconstruction of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), IEEE, pp. 5556–5565. 3, 4

[DCS*17] DAI A., CHANG A. X., SAVVA M., HALBER M., FUNKHOUSER T., NIESSNER M.: Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 1, 2, 3, 4, 5, 6, 7, 8, 9

[DDS*09] DENG J., DONG W., SOCHER R., LI L.-J., LI K., FEI-FEI L.: ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009). 1

[DNZ*17] DAI A., NIESSNER M., ZOLLHÖFER M., IZADI S., THEOBALT C.: Bundlefusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG) 36*, 3 (2017), 24. 3, 4, 6, 9

[EVGW*10] EVERINGHAM M., VAN GOOL L., WILLIAMS C. K., WINN J., ZISSERMAN A.: The pascal visual object classes (voc) challenge. *International journal of computer vision 88*, 2 (2010), 303–338. 8

[FB87] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*. Elsevier, 1987, pp. 726–740. 6, 9

[GPSW17] GUO C., PLEISS G., SUN Y., WEINBERGER K. Q.: On calibration of modern neural networks. In *International Conference on Machine Learning* (International Convention Centre, Sydney, Australia, 06–11 Aug 2017), Precup D., Teh Y. W., (Eds.), vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 1321–1330. 4

[HFL14] HERMANS A., FLOROS G., LEIBE B.: Dense 3D semantic mapping of indoor scenes from rgb-d images. In *IEEE International Conference on Robotics and Automation (ICRA)* (2014), IEEE, pp. 2631–2638. 1, 5, 6

[HMDC16] HAZIRBAS C., MA L., DOMOKOS C., CREMERS D.: Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian Conference on Computer Vision* (2016), Springer, pp. 213–228. 2

[HPN*16] HUA B.-S., PHAM Q.-H., NGUYEN D. T., TRAN M.-K., YU L.-F., YEUNG S.-K.: Scenenn: A scene meshes dataset with annotations. In *International Conference on 3D Vision (3DV)* (2016), IEEE, pp. 92–101. 1

[HWN18] HUANG Q., WANG W., NEUMANN U.: Recurrent slice networks for 3D segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 2626–2635. 1, 2, 8, 9

[HY16] HUANG J., YOU S.: Point cloud labeling using 3D convolutional neural network. In *International Conference on Pattern Recognition (ICPR)* (2016), IEEE, pp. 2670–2675. 2

[KK11] KRÄHENBÜHL P., KOLTUN V.: Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems* (2011), pp. 109–117. 2, 5, 6

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. In *ACM Transactions on Graphics (TOG)* (1987), vol. 21, ACM, pp. 163–169. 3, 5

[LDT*17] LAWIN F. J., DANELLJAN M., TOSTEBERG P., BHAT G., KHAN F. S., FELSBERG M.: Deep projective 3D semantic segmentation. In *International Conference on Computer Analysis of Images and Patterns* (2017), Springer, pp. 95–107. 1, 2

[LGW17] LIU M., GUO Y., WANG J.: Indoor scene modeling from a single image using normal inference and edge features. *The Visual Computer 33*, 10 (2017), 1227–1240. 10

[LMB*14] LIN T.-Y., MAIRE M., BELONGIE S., HAYS J., PERONA P., RAMANAN D., DOLLÁR P., ZITNICK C. L.: Microsoft coco: Common objects in context. In *European Conference on Computer Vision* (2014), Springer, pp. 740–755. 1

[LMSR17] LIN G., MILAN A., SHEN C., REID I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 2

[LSD15] LONG J., SHELHAMER E., DARRELL T.: Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 3431–3440. 2, 3, 8

[MHDL17] MCCORMAC J., HANDA A., DAVISON A., LEUTENEGGER S.: Semanticfusion: Dense 3D semantic mapping with convolutional neural networks. In *IEEE International Conference on Robotics and Automation (ICRA)* (2017), IEEE, pp. 4628–4635. 1, 2, 5, 6

[NHH15] NOH H., HONG S., HAN B.: Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)* (2015). 2

[NIH*11] NEWCOMBE R. A., IZADI S., HILLIGES O., MOLYNEAUX D., KIM D., DAVISON A. J., KOHI P., SHOTTON J., HODGES S., FITZGIBBON A.: Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE international symposium on Mixed and augmented reality (ISMAR)* (2011), IEEE, pp. 127–136. 3, 4

[NSF12] NATHAN SILBERMAN DEREK HOIEM P. K., FERGUS R.: Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision* (2012). 2, 3

[NZIS13] NIESSNER M., ZOLLHÖFER M., IZADI S., STAMMINGER M.: Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG) 32*, 6 (2013), 169. 3, 4, 9

[Occ16] OCCIPITAL: Occipital: The structure sensor, 2016. 6

[PHL17] PARK S.-J., HONG K.-S., LEE S.: Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)* (2017). 2, 3, 4, 6, 7, 9

[PLCD16] PINHEIRO P. O., LIN T.-Y., COLLOBERT R., DOLLÁR P.: Learning to refine object segments. In *European Conference on Computer Vision* (2016), Springer, pp. 75–91. 2

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3D classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE 1*, 2 (2017), 4. 1, 2, 8, 9

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems* (2017), pp. 5099–5108. 1, 2, 8, 9

[SA85] SUZUKI S., ABE K.: Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing 30*, 1 (1985), 32–46. 10

[SCH*16] SAVVA M., CHANG A. X., HANRAHAN P., FISHER M., NIESSNER M.: PiGraphs: Learning Interaction Snapshots from Observations. *ACM Transactions on Graphics (TOG) 35*, 4 (2016). 1

[YK15] YU F., KOLTUN V.: Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122* (2015). 2