# A Graph-Based Method for Joint Instance Segmentation of Point Clouds and Image Sequences

Montiel Abello, Joshua G. Mangelson, and Michael Kaess

***Abstract*— We address the problem of class agnostic, joint instance segmentation of scene data. While learning-based semantic instance segmentation methods have achieved impressive progress, their use is limited in robotics applications due to reliance on expensive training data annotations and assumptions of single sensor modality or known object classes. We propose a novel graph-based instance segmentation approach that combines information from a 2D image sequence and a 3D point cloud capturing the scene. Our approach propagates information with a general graph representation to produce a segmentation taking into account both geometric and photometric information. This allows us to leverage information from complementary sensor modalities without requiring training data. Our method shows improved object recall and boundary identification over state-of-the-art RGB-D segmentation methods. We demonstrate generality by evaluating on both RGB-D data and a LiDAR+image sensor data.**

## I. Introduction

Object-level scene representations are important for robotics applications. Typically, a scene is represented with 3D data obtained from exteroceptive sensors such as RGB-D cameras or LiDAR scanners. Object segmentations within the scene are typically performed on 2D images as a pre-processing step or directly on a 3D representation such as a point cloud. 2D data segmentation is challenging as camera images often contain photometric edges resulting from lighting, colour and occlusion that do not correspond to a true object boundary. In 3D data, reconstruction artefacts and poorly sampled object boundaries limit segmentation accuracy. Improved object level consistency can be achieved by combining photometric and geometric information acquired from multiple sensing modalities.

Deep learning methods have been extensively applied in segmentation of single-sensor data such as point clouds or images [1], [2], [3]. Recent works operating on 2.5D [4] and multi-view [3] data have shown a lot of progress on standard benchmarks, but their use is limited in a robotics context due to assumptions such as single sensor modality, reliance on expensive training data annotations and knowledge of a finite set of object classes. Alternatively, graph-based methods [5] are able to incorporate multimodal data. Prior work has applied graph-based methods to segmentation of a 3D LiDAR scan with colour information [6], [7] and semantic labels [8] from a registered 2D image.

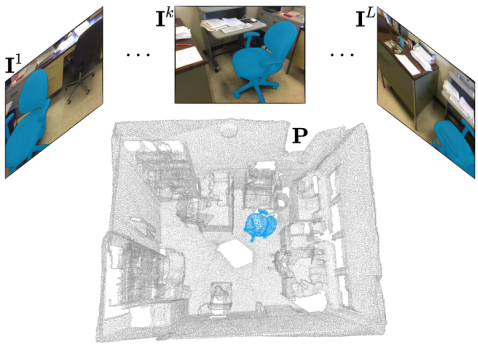


Fig. 1. Our method addresses joint segmentation of a point cloud and sequence of images. An example multimodal segment is shown in blue. Deep-learning methods are popular in segmentation of data from a single sensor as large annotated training datasets are available. This data-driven approach is less readily applied to multimodal data or reconstructed scenes where training data is expensive to obtain.

In this work, we propose a novel graph-based instance segmentation approach that combines information from both 2D images and 3D point clouds with a common graph representation. While many computer vision applications have RGB-D image sequences available, we maintain generality by performing joint segmentation on a single point cloud and sequence of RGB images. By reasoning jointly over 2D and 3D data, we combine geometric and photometric information to perform segmentation without training data.

Our contributions are: (i) A general graph representation that allows data from a variety of sensing modalities to be incorporated in a joint segmentation problem. (ii) A joint scene segmentation approach that combines photometric and geometric information to produce a consistent segmentation of all objects in a scene. (iii) An evaluation of our method on real-world RGB-D data [9] with indoor sequences varying in size and complexity. We compare our performance to state-of-the-art deep learning methods for segmentation of single view [4] and multi-view [3] RGB-D data, showing improved object recall and boundary precision. We also apply our method to a self-collected LiDAR dataset, demonstrating generalizability to different sensors and robotics platforms.

## II. Related Work

Accurate object segmentation improves perceptual understanding of the environment, furthering the capabilities of mobile robots. Fusion++ [10] uses a state-of-the-art instance segmentation network [1] to segment image frames and

This work was partially supported by Amazon Lab 126. We thank Paloma Sodhi, Allie Chang, and Eric Dexheimer for advice and feedback, and Zimo Li for his work in data collection.

M. Abello and M. Kaess are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA {montielabello, jmangels, kaess}@cmu.edu

J. Mangelson is with the Electrical and Computer Engineering Department, Brigham Young University, Provo, UT 84602, USA joshuamangelson@byu.edu

perform volumetric SLAM at the object level for improved robustness and efficiency. In the multi-view 3D reconstruction task [11], [12] incorporate per-pixel semantic labels, showing that both geometry and label accuracy are improved when jointly optimised. Segmentation is also useful when a scene contains challenging geometries. [13] use point cloud segmentation to extract thin structures from a coarse 3D reconstruction for further refinement. In the area of grasping, [14] apply superpixel segmentation to bounding box object detections in a single image, merging them using known object geometry to produce accurate segmentation masks for pose estimation. While deep-learning based instance segmentation methods perform well in terms of object recall, they have difficulty capturing fine details and classical methods using photometric information directly are often preferred when accurate object boundaries are required.

As shown in [13], [14], classical segmentation techniques such as [15], [5], [16], [17] perform well in real-world conditions, but are quite sensitive and require careful tuning for the combination of sensor and deployment environment. Most recent work in segmentation favours deep-learning methods [1], [18], [2], [19], [20] which make use of large labelled datasets [21], [22] for improved robustness, though segmentation is limited to object classes available in training.

The most relevant example of a modern, semantic instance-based segmentation method is 3D-SIS [3], a multi-view RGB-D segmentation network. Learned features from RGB images are backprojected to associated voxels in a 3D grid. 3D geometry and learned features are then used to estimate per-voxel object masks, predicted by combining 3D region proposals and semantic label estimation. SceneCut [4] is another relevant method for single view RGB-D images. It makes use of learned object boundaries rather than semantic instances so it is able to segment objects of classes it has not encountered in training. A hierarchical segmentation tree is computed from detected boundaries and an optimal segmentation combining photometric and geometric information is obtained with dynamic programming.

Deep learning methods relying on training data are less readily applied to robotics applications involving multiple sensors. Graph representations are often used to propagate segmentation information across data representations. For segmentation of a LiDAR scan and image, a common approach [6], [7] is to combine the inputs into a single graph which is segmented with a popular graph segmentation method [5]. LDLS [8] use an iterative label diffusion process to propagate semantic labels from an RGB image to 3D, for improved instance segmentation of a 3D LiDAR scan. Graph-based methods have also been used to combine information across images in multi-view segmentation [23], though this method extracts only a single object of interest which must be observed from many vantage points.

While the previous works combine information from multiple sensors, they only seek to segment a single data type. Some robotics applications may require a consistent scene segmentation of multimodal data. In our formulation, joint segmentation seeks to identify all points from a point cloud and all pixels from a sequence of images belonging to each object in the scene. This variant of joint segmentation is far less studied. Zhang [24] use a graph representation to propagate information and jointly segment a large scale point cloud and a registered set of images, though in this case only semantic segmentation is performed, with pixels or points assigned to one of five classes.

## III. Problem Formulation

We refer to the task of decomposing a point cloud and a set of registered images into a set of objects as joint instance segmentation. The point cloud is modelled as a set of $N$ 3D points $\mathbf{P} = \{\mathbf{p}_1, ..., \mathbf{p}_N\}$, $\mathbf{p}_i \in \mathbb{R}^3$ expressed in the global coordinate frame. The set of $L$ images is denoted $\mathbf{I} = \{\mathbf{I}^1, ...\mathbf{I}^L\}$ where $\mathbf{I}^k \in [0,1]^{W^k \times H^k \times 3}$. The corresponding camera viewpoints in the global coordinate frame are $\mathbf{T}^k \in SE(3)$. Each image can also be considered as a set of $M^k = W^k \times H^k$ pixels: $\mathbf{I}^k = \{\mathbf{q}_1^k, ..., \mathbf{q}_M^k\}$. Each pixel $\mathbf{q}_j^k = (\mathbf{c}_j^k, \mathbf{x}_j^k)$ consists of a normalised colour vector $\mathbf{c}_j^k \in [0,1]^3$ and image coordinates $\mathbf{x}_j^k \in [0, W^k - 1] \times [0, H^k - 1]$.

Performing segmentation on a point cloud or image is equivalent to finding a partition of an associated set. For a set $A = \{a_1, ..., a_N\}$, we denote a segmentation $S^A$ as a partitioning of $A$ into a set of unique, non-overlapping subsets such that $S^A = \{S_1^A, ..., S_K^A\}$, where $\bigcup_i^K S_i^A = A$. A segmentation has the property that each $a_i$ is assigned to a single segment $S^A(a_i) = S_j^A$, $j \in [1, K]$.

The goal of joint instance segmentation is to find a segmentation $S^{\{P,I\}}$ that accurately represents the distribution of objects in the scene captured by $\mathbf{P}$ and $\mathbf{I}$. As shown in Figure 1 each segment is a set that may contain both points $\mathbf{p}_i$ and pixels $\mathbf{q}_j^k$ and represents a distinct object. In this work, we perform class-agnostic segmentation, identifying objects with photometric and geometric information only, without the use of predefined object models or learned classes.

## IV. Approach

We use a general graph-based framework to propagate segmentation information between a 3D pointcloud and a set of registered 2D images to improve the joint instance segmentation of the scene. An efficient graph-based segmentation method [5] (FH) is used as the backbone of our approach. A graph $G = (V, E)$ consists of a set of vertices $\mathbf{v}_i \in V$ and a set of edges $\mathbf{e}_{i,j} \in E$, where $\mathbf{e}_{i,j} = (\mathbf{v}_i, \mathbf{v}_j, w_{i,j})$. Edge weight $w_{i,j}$ quantifies the distance between associated

**Algorithm 1** Joint instance segmentation approach

**for** $\mathbf{I}^k \in \mathbf{I}$ **do**
  $G^{\mathbf{I}^k} \leftarrow$ `imageToGraph`$(\mathbf{I}^k)$
  $S^{\mathbf{I}^k} \leftarrow$ `graphSegmentation`$(G^{\mathbf{I}^k})$
**end for**
$G^{\mathbf{P}} \leftarrow$ `pointcloudToGraph`$(\mathbf{P})$
$G^{\mathbf{P}+} \leftarrow$ `propagate2Dto3D`$(G^{\mathbf{P}}, \{S^{\mathbf{I}^0}, ..., S^{\mathbf{I}^L}\})$
$S^{\mathbf{P}} \leftarrow$ `graphSegmentation`$(G^{\mathbf{P}+})$
**for** $\mathbf{I}^k \in \mathbf{I}$ **do**
  $S^{\mathbf{I}^k+} \leftarrow$ `segmentationRefinement`$(S^{\mathbf{I}^k}, S^{\mathbf{P}})$
**end for**

vertices, smaller $w_{i,j}$ indicating similar vertices more likely to belong to the same segment. The FH algorithm examines edges of the graph and performs segment merges by comparing the current edge weight to those in existing segments. By examining edges by increasing edge weight, regions grow in a conservative manner.

The joint segmentation approach is outlined in Algorithm 1 and depicted in Figure 2. The 3D point cloud and each individual image are represented as separate graphs. Decoupling the problem in this manner reduces computational complexity, and single-modality preprocessing techniques can be applied to the raw data and included as inputs.

After conversion to a graph representation, an initial segmentation of each image is computed. These image segmentations are propagated to the 3D point cloud graph as described in IV-B, and used in the 3D segmentation. As segmentation artefacts caused by occlusion and lighting are typically not persistent across multiple viewpoints, combining multi-view photometric information improves the 3D segmentation. This 3D segmentation information is propagated back to each image graph and used to refine each image segmentation, detailed in IV-C. This reduces the effect of occlusion and lighting to produce object-consistent image segmentations.

### A. Initial Image Segmentation

For simplicity, in this section we drop notation indicating the $k-$th image and refer to image $\mathbf{I}^k$ as $\mathbf{I}$. The initial image segmentation $S^{\mathbf{I}}$ is computed using the FH segmentation algorithm on a graph constructed from the photometric information in each image. Shown in Figure 2 (top-left), boundaries are computed from each image using [25], a boundary detector using a structured learning framework to learn basic edge types from local image patches, trained in a class-agnostic manner with the BSDS500 dataset [16].

To represent an image with a graph, vertices are formed $\mathbf{v}_i^{\mathbf{I}} = (\mathbf{c}_i, \mathbf{x}_i, b_i)$, where pixel values $b_i$ are from boundary image $\mathbf{B}$. Edges are constructed according to an 8-connected

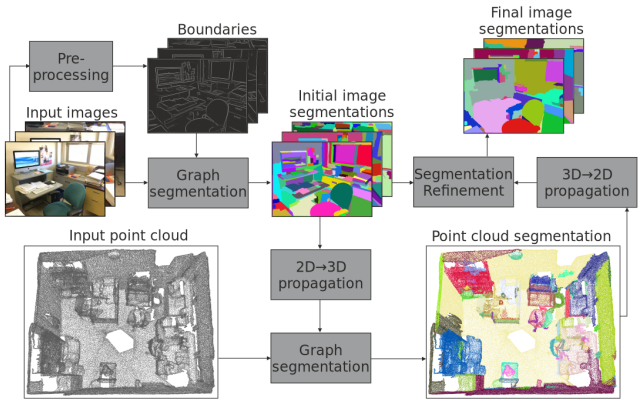


Fig. 2. The proposed joint segmentation method: 2D pixel – 3D point associations are used to propagate initial segmentation information from each image to adjust weights in the 3D point cloud graph. The point cloud is segmented, and this segmentation is used to refine each image segmentation.

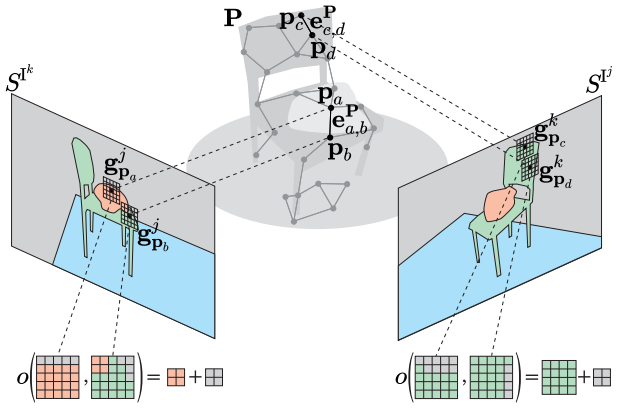


Fig. 3. Propagation of image segmentation information to 3D point cloud graph. For image $k$, 3D point $p_i$ is reprojected into image segmentation $S^{\mathbf{I}^k}$, and a grid $\mathbf{g}^k_{p_i}$ centered at corresponding pixel $\mathbf{x}^k_{p_i}$ is extracted. For each edge in the 3D point cloud graph, the overlap of the two grids associated with each end point is computed according to Eq. 5.

grid structure, connecting vertices according to Eq. 1.

$$\forall (i,j) \text{ s.t. } 1 \leq \|\mathbf{x}_i, \mathbf{x}_j\|_1 \leq 2,\ \exists\, \mathbf{e}^{\mathbf{I}}_{i,j} = \left(\mathbf{v}^{\mathbf{I}}_i, \mathbf{v}^{\mathbf{I}}_j, w^{\mathbf{I}}_{i,j}\right) \tag{1}$$

Edge weights are derived purely from photometric information in the image according to Eq. 2. For neighbouring pixels, the 2-norm of the colour channel and boundary magnitude are combined with hyperparameters $w_c$ and $w_b$. The use of the minimum boundary value is a design decision that reduces the effect of noise in the boundary image and pixel-level error in boundary localization.

$$w^{\mathbf{I}}_{i,j} = w_c \|\mathbf{c}_i - \mathbf{c}_j\|_2 + w_b \min(b_i, b_j) \tag{2}$$

The FH graph segmentation algorithm is then applied to the graph $G^{\mathbf{I}}$ to produce $S^{\mathbf{I}}$.

### B. Point Cloud Segmentation

The point cloud $\mathbf{P}$ is represented with a graph $G^{\mathbf{P}}$. Each 3D point is represented with a vertex $\mathbf{v}^{\mathbf{P}}_i = (\mathbf{p}_i, \mathbf{n}_i)$, where $\mathbf{n}_i$ is the surface normal estimated at $\mathbf{p}_i$ with first-order plane fitting [26]. Edges are constructed by connecting each vertex with their $K$ nearest neighbours in Euclidean space.

Geometric information used in segmentation combines Euclidean distance and $\theta_{i,j}$, the angle between surface normals of neighbouring points,

$$\theta_{i,j} = \text{atan2}(\mathbf{n}_i \times \mathbf{n}_j, \mathbf{n}_i \cdot \mathbf{n}_j) \tag{3}$$

The edge weights in $G^{\mathbf{P}}$ combine this geometric information with 2D segmentation information propagated from each image, using hyperparameters $w_d, w_\theta, w_r$ according to Eq. 4:

$$w^{\mathbf{P}}_{i,j} = w_d \|\mathbf{p}_i - \mathbf{p}_i\|_2 + w_\theta \theta_{i,j} + w_r r\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right) \tag{4}$$

The reprojection term $r\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right)$ quantifies the strength of the connection between points $\mathbf{p}_i, \mathbf{p}_j$ inferred from image segmentations $S^{\mathbf{I}}$. Figure 3 shows the process used to compute $r\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right)$ and explains its component parts. As objects will likely obscure one another, the visibility of

each $\mathbf{p}_i \in \mathbf{P}$ from each image viewpoint $\mathbf{T}^k$ is determined with the hidden point removal method [27]. Visibility count $v\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right)$ is the number of images in which both points $\mathbf{p}_i$ and $\mathbf{p}_i$ are visible. Each visible point reprojects into a pixel $\mathbf{x}_{p_i}^k$ in image segmentation $S^{\mathbf{I}^k}$. A grid $\mathbf{g}_{p_i}^k$ of size $t \times t$ centered at $\mathbf{x}_{p_i}^k$ is extracted from $S^{\mathbf{I}^k}$. Associating $\mathbf{x}_{p_i}^k$ with a grid as opposed to a single pixel allows for some error in image registration. The overlap $o$ shown in Eq. 5 measures the local similarity of image segmentations.

$$o\left(\mathbf{g}_{p_i}^k, \mathbf{g}_{p_j}^k\right) = \frac{\sum_l^{\left\{\mathbf{g}_{p_i}^k\right\} \cup \left\{\mathbf{g}_{p_j}^k\right\}} \min\left(\left|\mathbf{g}_{p_i}^k = l\right|, \left|\mathbf{g}_{p_j}^k = l\right|\right)}{\min\left(\left|\mathbf{g}_{p_i}^k\right|, \left|\mathbf{g}_{p_j}^k\right|\right)} \tag{5}$$

Finally, the hyperparameter $r_o$ acts as a mean value for reprojection term $r\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right)$, which is computed according to Eq. 6.

$$r\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right) = 1 - r_o + \frac{v\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right)}{L}\left(r_o - \sum_{k=1}^{L} \frac{o(\mathbf{g}_{p_i}^k, \mathbf{g}_{p_j}^k)}{v\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right)}\right) \tag{6}$$

The workings of Eq. 6 become clear when the domains of terms are considered: $r\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right) \in [0,1]$, $r_o \in [0,1]$, $\frac{v\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right)}{L} \in [0,1]$ and $\sum_{k=1}^{L} \frac{o(\mathbf{g}_{p_i}^k, \mathbf{g}_{p_j}^k)}{v\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right)} \in [0,1]$. The effect is that overlap greater than $r_o$ corresponds to low weight and overlap below $r_o$ corresponds to a high weight. Edges with low visibility count are less certain, so will have less effect by pulling the value of the reprojection term towards $r_o$. Conversely, edges with high visibility will have a low or high $r\left(S^{\mathbf{I}}, \mathbf{p}_i, \mathbf{p}_j\right)$ and contribute more.

As in the Section IV-A, the FH graph segmentation algorithm is applied to graph $G^{\mathbf{P}}$ to produce $S^{\mathbf{P}}$.

### C. Final Image Segmentation

For simplicity, we again drop notation indicating the $k$-th image. The final image segmentation $S^{\mathbf{I}+}$ refines $S^{\mathbf{I}}$ by propagating information from the 3D point cloud segmentation $S^{\mathbf{P}}$. As $\mathbf{P}$ is sparse, image graph edges cannot be adjusted in a method similar to IV-B. Instead, segments are merged and split to achieve consistency with $S^{\mathbf{P}}$.

The method used to propagate point cloud segmentation information into each image is shown in Figure 4. A vertex in $S^{\mathbf{P}}$ reprojects into image $\mathbf{I}$ according to $\mathbf{x}_{\mathbf{p}_i} = M(\mathbf{K}, \mathbf{T}, \mathbf{p}_i)$ where $M()$ models camera extrinsics and intrinsics. Shown in Figure 4 a), a reprojection graph $G^R$ is constructed with vertices $\mathbf{v}_i^R = (\mathbf{x}_{\mathbf{p}_i}, S^{\mathbf{P}}(\mathbf{p}_i))$. Edges $E^R$ are constructed to connect vertices within a fixed radius $R$ in the image plane.

$$\forall (i,j) \text{ s.t. } ||\mathbf{x}_{\mathbf{p}_i}, \mathbf{x}_{\mathbf{p}_j}||_2 \leq R,\ \exists\, \mathbf{e}_{i,j}^R = (\mathbf{v}_i^R, \mathbf{v}_j^R, w_{i,j}^R) \tag{7}$$

The segment labels of these reprojected 3D points are used to partition $E^R$ into non-overlapping subsets of internal $E_{\text{int}}^R$ and external $E_{\text{ext}}^R$ edges according to Eq. 8, as shown in Figure 4 b). Internal edges are constructed from points from same 3D segment, and external edges from 3D points in different 3D segments.

$$\mathbf{e}_{i,j}^R \in \begin{cases} E_{\text{int}}^R & \text{if } S^{\mathbf{P}}(\mathbf{p}_i) = S^{\mathbf{P}}(\mathbf{p}_j) \\ E_{\text{ext}}^R & \text{if } S^{\mathbf{P}}(\mathbf{p}_i) \neq S^{\mathbf{P}}(\mathbf{p}_j) \end{cases} \tag{8}$$

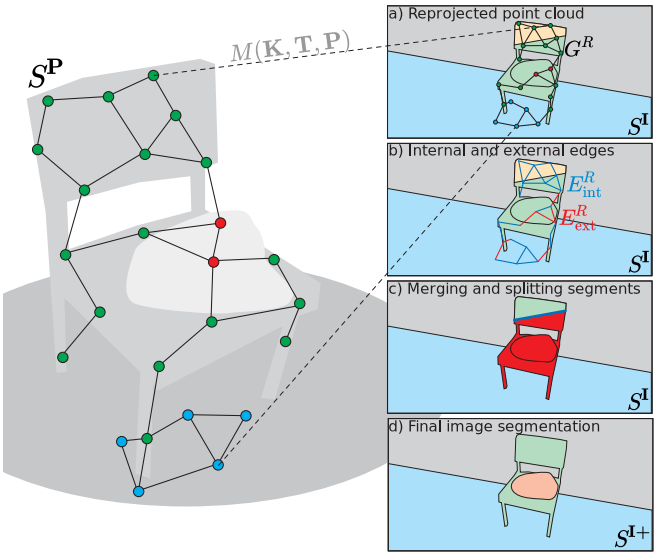


Fig. 4. Propagation of 3D point cloud segmentation into image for refinement of image segmentation. a) For each image, a 2D graph is constructed with set of visible reprojected points. b) Edges of this graph are *internal* (blue)/*external* (red) if the associated 3D points are in the same/different segment. c) Two image segments are merged if they are connected by many internal edges (blue boundary). An image segment (red) is split with the minimum-cut algorithm if it contains many external edges.

Figure 4 c) and d) depict segments in $S^{\mathbf{P}}$ that are merged or split to maintain consistency with the point cloud segmentation which is represented by $E_{\text{int}}^R$ and $E_{\text{ext}}^R$.

Internal edges connecting two image segments suggest that they represent the same object and a photometric edge has caused an undesired separation. We use the notation $\mathbf{e}_{i,j}^R \in \left(S_m^{\mathbf{I}}, S_n^{\mathbf{I}}\right)$ to indicate that $\mathbf{x}_{\mathbf{p}_i} \in S_m^{\mathbf{I}}$ and $\mathbf{x}_{\mathbf{p}_j} \in S_n^{\mathbf{I}}$. The numbers of internal, external and total reprojected edges are denoted:

$$n_{\text{int}} = \left|E_{\text{int}}^R \in \left(S_m^{\mathbf{I}}, S_n^{\mathbf{I}}\right)\right| \tag{9}$$

$$n_{\text{ext}} = \left|E_{\text{ext}}^R \in \left(S_m^{\mathbf{I}}, S_n^{\mathbf{I}}\right)\right| \tag{10}$$

$$n_R = \left|E^R \in \left(S_m^{\mathbf{I}}, S_n^{\mathbf{I}}\right)\right| \tag{11}$$

In Eq. 12 the ratio of internal edges to the total number connecting two segments is compared with hyperparameter $r_m$ to determine if they should be merged.

$$\left\{S_m^{\mathbf{I}+}, S_n^{\mathbf{I}+}\right\} \leftarrow \begin{cases} S_m^{\mathbf{I}} \cup S_n^{\mathbf{I}} & \text{if } \frac{n_{int}}{n_R} \geq r_m \\ \left\{S_m^{\mathbf{I}}, S_n^{\mathbf{I}}\right\} & \text{if } \frac{n_{ext}}{n_R} < r_m \end{cases} \tag{12}$$

External edges fully contained by a 3D segment suggest undersegmentation. The greedy FH graph segmentation algorithm will merge segments connected by a low weight edge regardless of whether connecting edges indicate a strong boundary. In this case a subgraph representing the segment is extracted and we perform splitting the minimum-cut approach based on [17]. While inefficient for an entire image, minimum-cut is well suited for finding an optimum boundary in a single segment. It splits a segment along this boundary according to Eq. 13.

$$\texttt{min-cut}\left(S_l^{\mathbf{I}}\right) = \left\{S_m^{\mathbf{I}}, S_n^{\mathbf{I}}\right\} \tag{13}$$

where $S^{\mathbf{I}}_m \cup S^{\mathbf{I}}_n = S^{\mathbf{I}}_l$ and $S^{\mathbf{I}}_m \cap S^{\mathbf{I}}_n = \emptyset$.

The number of external edges is compared to hyperparameter $n_s$ to determine if the minimum-cut method should be applied, as shown in Eq. 14. This thresholding is performed using the number of external edges rather than the fraction to account for small missed segments that would otherwise be undetected.

$$S^{\mathbf{I}+}_l \leftarrow \begin{cases} \{S^{\mathbf{I}}_m, S^{\mathbf{I}}_n\} & \text{if } \left|E^R_{\text{ext}} \in (S^{\mathbf{I}}_m, S^{\mathbf{I}}_n)\right| \geq n_s \\ S^{\mathbf{I}}_l & \text{if } \left|E^R_{\text{ext}} \in (S^{\mathbf{I}}_m, S^{\mathbf{I}}_n)\right| < n_s \end{cases} \tag{14}$$

Finally, image segments are merged with the mode of reprojecting point cloud segments to produce a joint segmentation of the scene, where each segment represents an object in the point cloud and across all images it is observed in.

## V. Evaluation

### A. ScanNet v2 RGB-D Dataset and Benchmarks

We use the ScanNet v2 [9] RGB-D dataset to evaluate our method. It contains semantic instance labelled RGB-D sequences of a variety of indoor scenes. In addition, a labelled point cloud derived from [28] is available. Our method uses only the position data in the point cloud and the colour channels of each image. We evaluate using sequences 144, 378, 423, 427 and 664 which are augmented for use with 3D-SIS and provided in the Github repository of [3]. These sequences vary in size, lighting conditions and clutter. Each contains 1–2K images and a point cloud consisting of 100–200K points. We use the same parameter values across all sequences.

We perform class agnostic instance segmentation evaluation against two state-of-the-art methods. The point cloud segmentation is compared with 3D-SIS [3]. It uses multi-view RGB-D images which are roughly equivalent with our input in terms of available information. Image segmentation is compared with SceneCut [4], whose off-the-shelf implementation is not fine tuned on ScanNet v2, and which uses single RGB-D images so scores are not strictly comparable to our multi-view approach. Rather, we seek to compare characteristics of the methods that make them suitable for different tasks. The boundary detection submodules of SceneCut and our method are both trained on the BSDS500 dataset [16]. We perform a single joint segmentation on each sequence and compare the point cloud and image sequence segmentations to 3D-SIS and SceneCut respectively, rather than comparing with separately derived solutions.

We use symmetric segmentation covering (SSC) and F-measure for objects and parts (F) described in [29] as evaluation metrics. F is derived from precision (Pr) and recall (Re) scores computed in an approach similar to that used in the standard average-precision (AP) metric. F is more appropriate for class agnostic segmentation of all data in an image or point cloud, is it identifies fragments (undersegmented), objects and parts (oversegmented), rather than simply counting true and false positive object detections. We use thresholds $\gamma_o = 0.75$, $\gamma_p = 0.25$, $\beta = 0.1$, which are described in detail in [29]. We use more relaxed values to better identify objects and parts, as the ScanNet v2 sequences contain a high amount of clutter. For comparison to 3D-SIS which is based on object detection, we use Pr and Re to compute an analogue to standard AP. Where AP varies the IoU value determining a positive detection, we separately compute scores varying both the object and part thresholds $\gamma_o, \gamma_p$ to compute $\text{AP}_o$, $\text{AP}_p$ which quantify object and part segmentation accuracy respectively.

TABLE I

Point cloud segmentation comparison on ScanNet v2

| Method | SSC | Pr | Re | F | $\text{AP}_o$ | $\text{AP}_p$ |
|---|---|---|---|---|---|---|
| 3D-SIS (**P**) | 48.63 | **72.78** | 25.31 | 29.36 | 19.69 | 24.30 |
| Ours (**P**) | 50.80 | **22.33** | 28.49 | 19.81 | 10.22 | 8.581 |
| 3D-SIS ($\mathbf{P}_D$) | **63.66** | 67.66 | 26.20 | 31.73 | 16.28 | 29.50 |
| Ours ($\mathbf{P}_D$) | **55.02** | 24.06 | 33.05 | 24.51 | 13.52 | 29.54 |

### B. ScanNet v2 RGB-D Point Cloud Segmentation

Across the five sequences, approximately 65.38% of the **P** is overlapped by 3D-SIS outputs in the form of object detections and masks. This output is considered in two ways: (i) As a segmentation of the entire scene, with the undetected portion considered a background object. We compare using **P**. (ii) As detections of objects of interest. Here we extract a subset $\mathbf{P}_D$ corresponding to detected ground truth objects only. Table I shows these computed scores. Scores are percentages and higher values desired. Across (i) and (ii) our method achieves higher Re scores. This is seen in Figure 5, where 3D-SIS produces far fewer segments. Conversely, our class agnostic method produces oversegmentation resulting in reduced Pr scores. $\text{AP}_p$ is comparable on $\mathbf{P}_D$, as this score does not penalise oversegmentation of objects. On **P**, 3D-

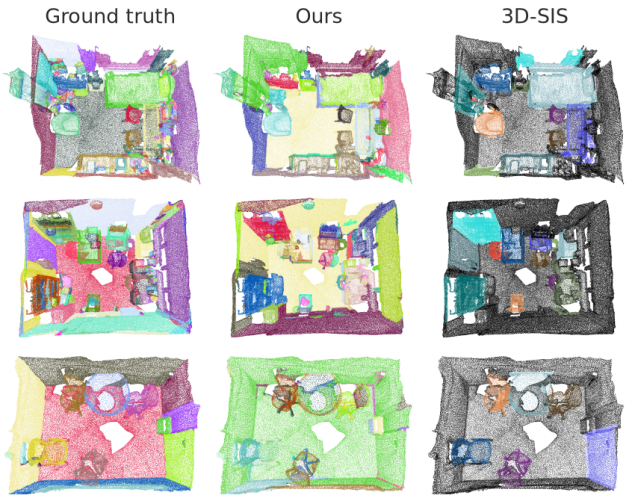


Fig. 5. Segmented point clouds from ScanNet v2 sequences 144 (top row), 378 (middle row), 427 (bottom row). In our result, colour differentiates object segment but does not correspond to any class. Oversegmentation is common in our method, often present in areas where geometric or colour information suggest a boundary. 3D-SIS is trained semantically so it can produce object-consistent segments, but does not segment large regions shown in black, as training is limited to a finite set of classes.

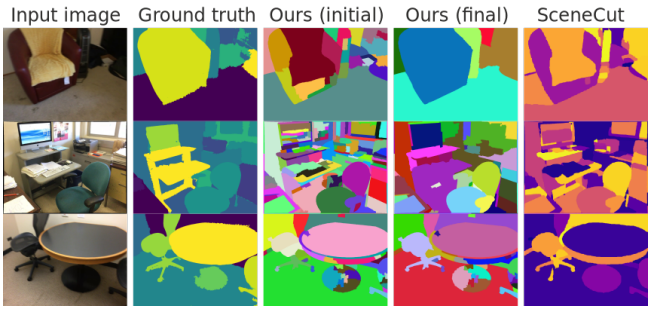


Fig. 6. Example input images and segmentation results from ScanNet sequences 144 (top row), 378 (middle row), 427 (bottom row). Our initial segmentation identifies photometric boundaries well, but is oversegmented at the object level, while the final result achieves a suitable segmentation level. SceneCut has strong recall for objects, but tends to oversegment.

SIS has far higher $AP_p$ as this metric rewards very severe undersegmentation. We achieve comparable SSC on $\mathbf{P}_D$, as boundary shapes have roughly the same level of accuracy. Our method performs comparably to 3D-SIS in inference time. Point cloud normal estimation, graph construction and segmentation was performed on an i7 CPU with an average total time of 4.5s, compared to an average 4.1s for GPU inference on 3D-SIS.

### C. ScanNet v2 RGB-D Image Sequence Segmentation

We compare our initial and final image segmentations with SceneCut, reporting scores in Table II. Our final result achieves almost equal SSC with SceneCut. As seen in Figure 6, our result has slight undersegmentation while SceneCut is oversegmented. Poor Pr in both methods may be caused by error in ground truth annotations resulting from reconstruction limitations in dataset generation [28]. However, our higher Pr is justified when comparing boundary shapes between input images and our results by eye. Fine boundary details are lost with CNN-based methods due to image size reduction, while they can be identified with our graph-based method that uses all photometric information. Conversely, SceneCut has much stronger Re as it uses a boundary detector trained with semantic annotations. Our method incorrectly merges or splits objects due to uneven point cloud density, such as the table base in Figure 6.

The inference time for our method was greater than that of SceneCut, in part because of the processing required to include information from the point cloud segmentation. The total time for point reprojection, initial image segmentation and refinement was 14.6s per image, compared to an average of 5.0s per image for SceneCut inference.

TABLE II
IMAGE SEGMENTATION COMPARISON ON SCANNET V2

| Method | SSC | Pr | Re | F |
|---|---|---|---|---|
| Ours-initial (**I**) | 42.34 | 5.116 | 42.36 | 8.474 |
| Ours-final (**I**) | 46.70 | **10.28** | **38.11** | 13.37 |
| SceneCut (**I**) | 46.47 | **4.701** | **52.54** | 7.928 |

### D. LiDAR+Image Dataset Evaluation

To demonstrate the broad applicability of our method to data from a range of sensor types we perform a quantitative evaluation on a self-collected dataset previously used in [30]. It consists of scans from a Velodyne VLP-16 LiDAR and a FLIR Grasshopper3 camera. We use state estimation from [31] to align scans into a single point cloud. Figure 7 demonstrates the performance of our joint segmentation method. The combination of 2D and 3D information is effective in identifying object boundaries. However, without object-level semantic knowledge, oversegmentation occurs.

## VI. CONCLUSION

We have presented an approach for joint segmentation of 2D and 3D data, a task that is less amenable to deep learning methods. The use of a general graph representation allows the method to be applied to data acquired from a variety of sensors. We compare to state-of-the-art segmentation methods combining 2D and 3D information, showing improved object recall and boundary shape. Strong performance in terms of these qualities is essential in robotics applications such as manipulation and mapping for navigation. The decoupling of image and point cloud segmentation means that external techniques can be applied to the input data and incorporated into our approach. This presents an opportunity to combine information from deep networks well trained for segmentation of either 2D or 3D data into our method. Future work could investigate whether this approach allows data-driven methods to be widely deployed in challenging, real-world robotics applications.

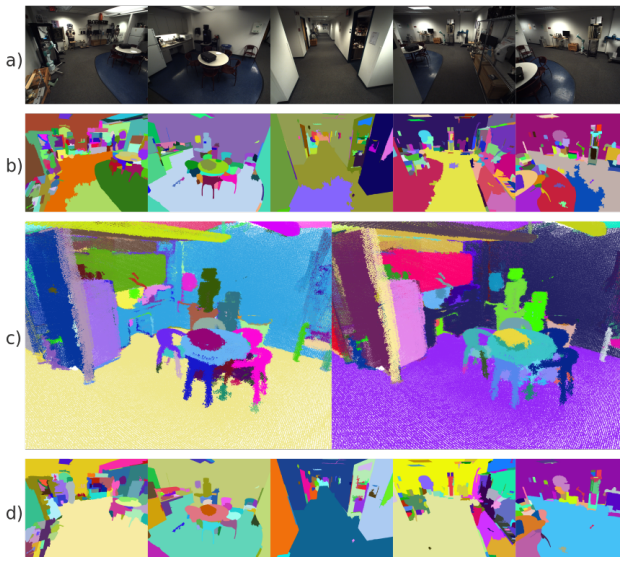


Fig. 7. The performance of our method on a LiDAR+Image Dataset. a) Input images show a challenging environment with many small objects. b) Initial image segmentations tend to oversegment the scene and harsh lighting causes artefacts. c) A comparison of a point cloud segmented with geometric only/combined information (left/right). d) Final image segmentations are improved (incorrect boundaries on the floor merged).

## REFERENCES


[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. Intl. Conf. on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 2961–2969.

[2] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, United States, June 2018, pp. 2569–2578.

[3] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3d semantic instance segmentation of RGB-D scans," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, United States, June 2019, pp. 4421–4430.

[4] T. Pham, T. Do, N. Sünderhauf, and I. Reid, "SceneCut: Joint geometric and object segmentation for indoor scenes," in *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Brisbane, Australia, May 2018, pp. 1–9.

[5] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *Intl. J. of Computer Vision (IJCV)*, vol. 59, pp. 167–181, Sept. 2004.

[6] J. Schoenberg, A. Nathan, and M. Campbell, "Segmentation of dense range information in complex urban scenes," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, Oct. 2010, pp. 2033–2038.

[7] J. Strom, A. Richardson, and E. Olson, "Graph-based segmentation for colored 3D laser point clouds," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, Oct. 2010, pp. 2131–2136.

[8] B. H. Wang, W. L. Chao, Y. Wang, B. Hariharan, K. Q. Weinberger, and M. Campbell, "LDLS: 3-D object segmentation through label diffusion from 2-D images," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, pp. 2902–2909, July 2019.

[9] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, United States, July 2017, pp. 5828–5839.

[10] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *2018 International Conference on 3D Vision (3DV)*, Verona, Italy, Sept. 2018, pp. 32–41.

[11] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, "Joint 3D scene reconstruction and class segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Portland, United States, June 2013, pp. 97–104.

[12] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint semantic segmentation and 3D reconstruction from monocular video," in *Proc. Eur. Conf. on Computer Vision (ECCV)*, Zurich, Switzerland, Sept. 2014, pp. 703–718.

[13] L. Liu, N. Chen, D. Ceylan, C. Theobalt, W. Wang, and N. J. Mitra, "CurveFusion: Reconstructing thin structures from RGBD sequences," *ACM Trans. on Graphics (ToG)*, vol. 37, no. 6, pp. 1–12, 2018.

[14] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis, "Single image 3D object detection and pose estimation for grasping," in *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Hong Kong, June 2014, pp. 3936–3943.

[15] A. J. B. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," *Semantic Perception Mapping and Exploration (SPME)*, May 2013.

[16] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, pp. 898–916, Aug. 2010.

[17] A. Golovinskiy and T. Funkhouser, "Min-cut based segmentation of point clouds," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, Kyoto, Japan, Sept. 2009, pp. 39–46.

[18] X. Chen, R. Girshick, K. He, and P. Dollár, "TensorMask: A foundation for dense object segmentation," in *Proc. Intl. Conf. on Computer Vision (ICCV)*, Seoul, Korea, Oct. 2019, pp. 2061–2069.

[19] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "GSPN: Generative shape proposal network for 3D instance segmentation in point cloud," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, United States, June 2019, pp. 3947–3956.

[20] L. Han, T. Zheng, L. Xu, and L. Fang, "OccuSeg: Occupancy-aware 3D instance segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, virtual, June 2020, pp. 2940–2949.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. on Computer Vision (ECCV)*, Zurich, Switzerland, Sept. 2014, pp. 740–755.

[22] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, United States, June 2016, pp. 3213–3223.

[23] A. Djelouah, J.-S. Franco, E. Boyer, F. L. Clerc, and P. Pérez, "Multi-view object segmentation in space and time," in *Proc. Intl. Conf. on Computer Vision (ICCV)*, Sydney, Australia, Apr. 2013, pp. 2640–2647.

[24] H. Zhang, J. Wang, T. Fang, and L. Quan, "Joint segmentation of images and scanned point cloud in large-scale street scenes with low-annotation cost," *IEEE Trans. on Image Processing*, vol. 23, no. 11, pp. 4763–4772, Aug. 2014.

[25] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 8, pp. 1558–1570, Aug. 2014.

[26] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[27] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," in *ACM SIGGRAPH 2007 papers*, San Diego, United States, Aug. 2007, pp. 24–es.

[28] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," *ACM Trans. on Graphics (ToG)*, vol. 36, no. 4, p. 1, May 2017.

[29] J. Pont-Tuset and F. Marques, "Supervised evaluation of image segmentation and object proposal techniques," *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 7, pp. 1465–1478, July 2015.

[30] Z. Li, P. Gogia, and M. Kaess, "Dense surface reconstruction from monocular vision and LiDAR," in *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019.

[31] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time." in *Proc. Robotics: Science and Systems (RSS)*, Berkeley, United States, July 2014.