3D Semantic Parsing of Large-Scale Indoor Spaces

Iro Armeni[1] Ozan Sener[1,2] Amir R. Zamir[1] Helen Jiang[1]
Ioannis Brilakis[3] Martin Fischer[1] Silvio Savarese[1]

[1] Stanford University [2] Cornell University [3]University of Cambridge

http://buildingparser.stanford.edu/

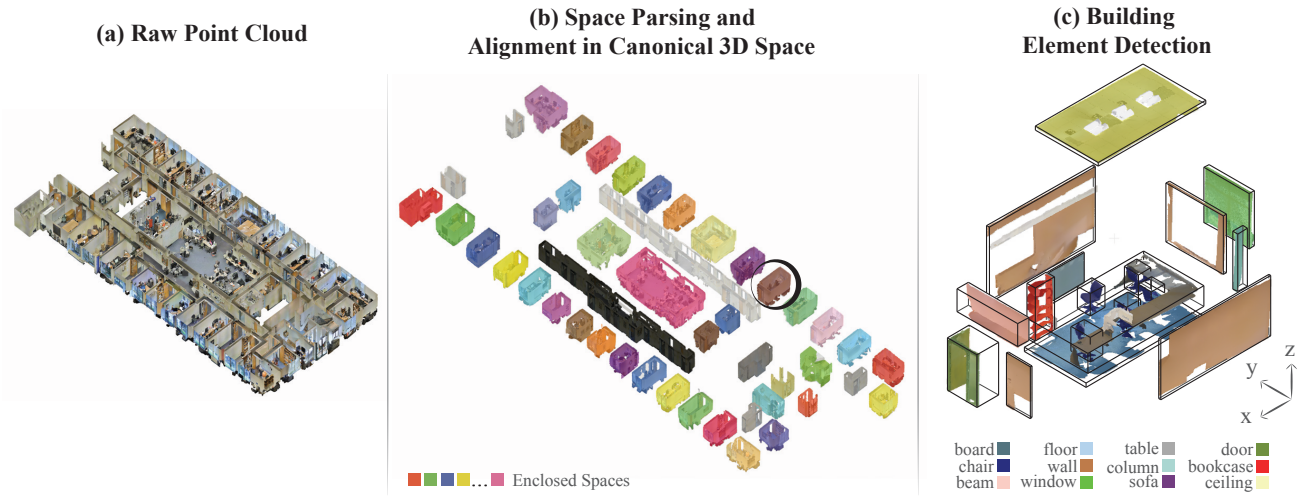

Figure 1: Semantic parsing of a large-scale point cloud. Left: the raw point cloud. **Middle:** the results of parsing the point cloud into disjoint spaces (*i.e.* the floor plan). **Right:** the results of parsing a detected room (marked with the black circle) into semantic elements.

Abstract

*In this paper, we propose a method for semantic parsing the 3D point cloud of an entire building using a hierarchical approach: first, the raw data is parsed into semantically meaningful spaces (*e.g. *rooms, etc) that are aligned into a canonical reference coordinate system. Second, the spaces are parsed into their structural and building elements (*e.g. *walls, columns, etc). Performing these with a strong notation of global 3D space is the backbone of our method. The alignment in the first step injects strong 3D priors from the canonical coordinate system into the second step for discovering elements. This allows diverse challenging scenarios as man-made indoor spaces often show recurrent geometric patterns while the appearance features can change drastically. We also argue that identification of structural elements in indoor spaces is essentially a detection problem, rather than segmentation which is commonly used. We evaluated our method on a new dataset of several buildings with a covered area of over* $6,000m^2$ *and over* 215 *million points, demonstrating robust results readily useful for practical applications.*

1. Introduction

During the past few years, 3D imaging technology experienced a major progress with the production of inexpensive depth sensors (*e.g.* Kinect [2]). This caused a leap in the development of many successful semantic segmentation methods that use both RGB and depth [27, 32, 8]. However, the 3D sensing field has recently undergone a follow-up shift with the availability of mature technology for scanning large-scale spaces, *e.g.* an entire building. Such systems can reliably form the 3D point cloud of thousands of square meters with the number of points often exceeding hundreds of millions (see Fig. 1 left). This demands semantic parsing methods capable of coping with this scale, and ideally, exploiting the unique characteristics of such data.

Large-scale scans of buildings pose new challenges/opportunities in semantic parsing that are different from, or not faced in, small-scale RGB-D segmentation:

Richer Geometric Information: Large-scale point clouds make the entire building available at once. This allows utilizing recurrent geometric regularities common in man-made structures. Such possibilities are beyond what a

single-view depth sensor would provide, as they have part of one room or at most few rooms in their scope.

Complexity: Existing semantic segmentation methods designed for small-scale point clouds or RGB-D images are not immediately applicable to large-scale scans due to complexity issues and the fact that choosing a set of representative views from an unbounded number of feasible single-views is non-trivial.

Introduction of New Semantics: Large-scale point clouds of indoor spaces introduce semantics that did not exist in small-scale point clouds or RGB-D images: disjoint spaces like rooms, hallways, etc. Parsing a raw point cloud into such spaces (essentially a floor plan) is a relatively new and valid problem.

Novel Applications: A number of novel applications becomes feasible in the context of whole building point clouds, such as, generating space statistics, building analysis (*e.g.*, workspace efficiency), or space manipulation (*e.g.*, removing walls between rooms).

Aforementioned points signify the necessity of adopting new approaches to semantic parsing of large-scale point clouds. In this paper, we introduce a method that, given a raw large-scale colored point cloud of an indoor space, first parses it into semantic spaces (*e.g.*, hallways, rooms), and then, further parses those spaces into their structural (*e.g.* floor, walls, etc.) and building (*e.g.* furniture) elements (see Fig. 1). One property of our approach is utilizing in semantic element detection the geometric priors acquired from parsing into disjoint spaces, and then, reincorporating the detected elements in updating the found spaces (Sec. 3.2).

Another key property is reformulating the element parsing task as a detection problem, rather than segmentation. Existing segmentation paradigms start with the assumption that each point must belong to a single segment/class. However, the problem of building element parsing better fits a detection approach. Clutter can occlude parts of important elements, *e.g.* a white board can occlude a wall. To a segmentation technique, this wall would be an irregular entity with a hole on it, while detecting the wall as a whole provides a better structural understanding of it (see Sec. 4).

The contributions of this paper can be summarized as:

I) We claim and experimentally evaluate that space dividers (*i.e.* walls) can be robustly detected using the empty space attributed to them in the point cloud. In other words, instead of detecting points belonging to the boundaries of a room, we detect the empty space bounded by them.

II) We show that structural and building elements can be robustly detected using strong geometric priors induced by space parsing. We demonstrate satisfactory parsing results by heavily exploiting such features.

III) We collected a large-scale dataset composed of colored 3D scans[1]) of indoor areas of large buildings with various architectural styles. A few samples of these spaces can be seen in Fig. 1 and 5. We annotated the semantic spaces and their elements in 3D. We further collected a set of RGB-D images registered on the colored point cloud to enrich the dataset (not used by our method). Annotations are consistent across all modalities (3D point cloud and RGB, and depth images). The dataset, annotations, the code and parsing results of the proposed framework are available to public at buildingparser.stanford.edu.

[1]Collection of points with 3D coordinates and RGB color values.

2. Related Work

We provide an overview of the related literature below, but as a brief summary, the following main points differentiate our approach from existing techniques: 1) processing a large-scale point cloud of an entire building (indoor spaces), rather than one or few RGB-D images, 2) detection of space dividers (walls) based on their void (empty) space rather than planar-surface/linear-boundary assumptions, 3) utilizing a set of geometric priors extracted in a normalized canonical space, 4) adopting a detection-based approach, rather than segmentation, to element parsing.

Semantic RGB-D and 3D segmentation have been investigated in a large number of papers during the past few years. For instance, [31, 25] proposed an RGB-D segmentation method using a set of heuristics for leveraging 3D geometric priors. [22] developed a search-classify based method for segmentation and modeling of indoor spaces. These are different from our method as they address the problem in a small-scale. A few methods attempted using multiple depth views [30, 15], yet they as well remain limited to a small-scale and do not utilize the advantages of a larger scope. [23] performed semantic parsing of buildings but for outdoor spaces. To parse a panoramic RGB-D image, [42] uses the global geometry of the room and cuboid like objects. Though an RGB-D panorama includes more information than a typical RGB-D image, it is not as comprehensive as a 3D point cloud. There also exist many object detection methods developed for RGB-D. These methods either try to extend the RGB methods directly into RGB-D by treating depth as a fourth channel [14, 20, 32, 27, 3] or use external sources like CAD models [33]. These methods use image-specific features and do not extend to point clouds. They are also not designed to handle large structural elements, such as floor and ceiling.

In the context of floor plan estimation, [4] proposed an approach based on trajectory crowd sourcing for estimating a floor plan, while we use an automatically generated 3D point cloud. [39] reconstructed museum type spaces based on Hough transform which is challenged in cluttered scenes (as verified by our experiments), though their goal is not estimation of floor plan. [41] also employs similar planar surface assumption in order to estimate the semantics of a single room using contextual information. [21] reconstructed

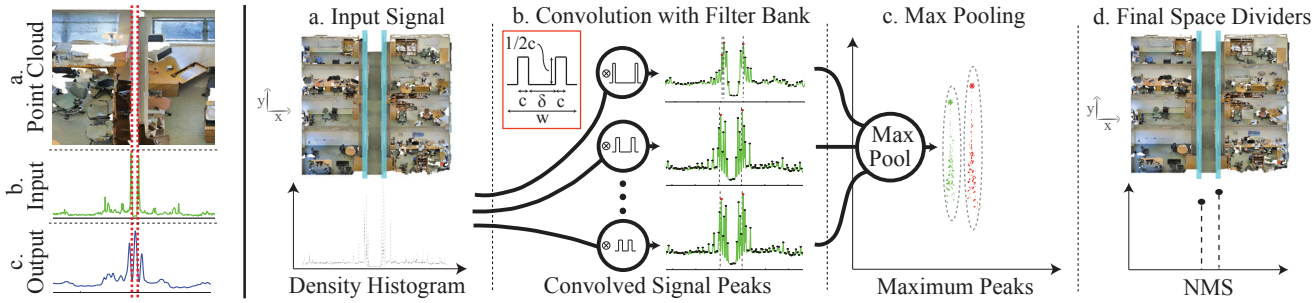

Figure 2. **Left:** Convolution of the devised filter with the histogram signal. The Histogram signal along axis x is the histogram of x coordinates of all points. **Right:** Space divider detection algorithm. We start with the density histogram signal (a), convolve it with the filter bank (b), and perform max-pooling (c) to identify the space dividers (d).

cluttered indoor spaces but their method as well as that of [24] require prior knowledge of scan locations and extraction of planar patches as candidate walls. [37] generated a minimalistic floor plan by first triangulating the 2D floor plan and then merging adjacent segments to obtain the final space partitioning. Their approach does not handle occlusions effectively and requires the scan locations. Liu et al. [19] reconstructed a building in 3D given monocular images and the floor-plan. On the contrary, we find the floor-plan as well as semantic elements therein given a 3D point cloud.

3. Parsing Point Cloud into Disjoint Spaces

Our hierarchical parsing method starts with parsing the whole building into spaces that are semantically meaningful (*e.g.* rooms). This step yields an understanding of the spatial layout and the spaces therein, which will play a central role in the formulation of the second step (Sec. 3.2).

3.1. Detection incorporating void spaces

Each scanned element in a point cloud is represented as a group of points encompassing its inner void (empty) space. The scanned points belong to the exterior surfaces of the element since only this outer shell is visible to a 3D sensor. However, the inner void is a crucial component in defining the element and its overall shape. This perspective suggests that void space could be actively incorporated in detecting and understanding 3D elements.

Space dividers (*e.g.* walls) separating neighboring enclosed spaces are not an exception. Previous efforts towards detecting walls in point clouds overlook this and try to to fit planar surfaces (*e.g.* [41]) or linear boundaries employing algorithms, such as RANSAC or Hough Transform. These are easily challenged in practice since walls are often cluttered with furniture, and sometimes even not visible.

In contrast, we follow the "void-based" approach and detect space dividers based on their signature in point clouds: *a space divider is depicted as a void space bounded by two coarsely parallel margins*. This signature remains robust even if the enclosed space is severely cluttered, since we do not detect surfaces or space boundaries but the void in-between. This is shown in Fig. 2 left (b) which depicts two adjacent rooms. The wall and its void space are indicated with red lines. If we form a 1 dimensional histogram of density of points along the x axis (*i.e.* the signal in Fig. 2 left (b)), the wall appears with the signature of two peaks with an empty space in-between. Attempting to find a wall through detecting planar surfaces would be equivalent to looking for peaks in this signal. As apparent, many strong peaks (*e.g.* due to the bookcase or table side) appear which make detection of walls difficult. Instead, the *peak-gap-peak* structure is significantly more discriminative and robust. This signature is one of the useful consequences of having the point cloud of the entire building at once.

3.1.1 Detecting the *peak-gap-peak* pattern

In order to detect the *peak-gap-peak* pattern, we follow a template matching approach using a bank of *peak-gap-peak* filters and perform the matching operation via convolutions on a density histogram signal (Fig. 2 left (a)). This filter bank is shown in Fig. 2 right (b) and has the characteristic of two peaks separated by void with varying widths. The blue curve in Fig. 2 left (c) is the convolution response which shows the filter has strongly responded to the peak-gap-peak signature and suppressed the other peaks. It should be noted that the employed filter assumes buildings with roughly planar walls, and hence, does not handle circular and oval shaped rooms or other configurations that deviate from the major axes of the building. However, as non-rectangular rooms make up for a considerably small portion of indoor space layouts [34], this assumption is considered reasonable. A remedy to irregular walls/rooms would be to employ a similar approach, but with a 2D filter bank that is also parametrized over curvature. Since though they account for a small portion, the practical importance of this improvement would not be obvious.

In greater detail, given a raw point cloud, we first align the three main axes [2] of $x-y-z$ with the general struc-

[2] We used PCA. However, there are other methods dedicated to this task (Manhattan frame estimation) that could be employed off-the-shelf in more complex cases [13, 35].

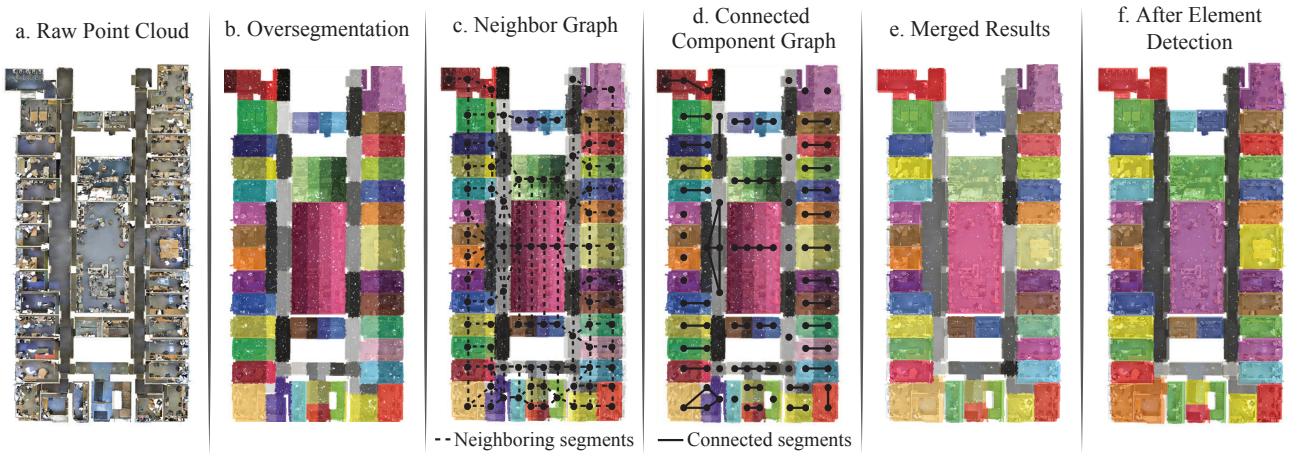

Figure 3. **Merging the over-segments:** We start with a set of over-segments (b) generated from the point cloud (a) and create their neighbor graph (c). Then, we merge nodes (d-e) as explained in Sec. 3.1.2. We update (f) the results given the output of element detection (Sec. 4.2).

ture of the building. We form a 1 dimensional histogram of density of points along one of the three axes, say $H(s)$. Then, we create a bank of filters parametrized by the pass-width (c) and the stop-width (δ) as shown in Fig. 2 right (b). The filters can be represented as $g_{\delta,c}(s) = \frac{1}{2C}\Pi_{\frac{\delta}{2}+C}(s) - \frac{1}{2C}\Pi_{\frac{\delta}{2}}(s)$ where $\Pi_k(s) = \mathbb{1}[|s| \leq k]$ and $\mathbb{1}[A]$ is an indicator function which is 1 when A is true and 0 otherwise.

We compute responses of filters when convolved with $H(s)$ as shown in Fig. 2 right (b). Each convolution results in a 3-dimensional score function over the axis of choice s and the parameters c and δ. We then apply max-pooling across s (*i.e.* pooling parameters are c and δ) to detect a set of wall candidates over the axis of choice. Finally, we apply non-maximum suppression to detect the final wall locations *(see [5] for details)*. We use a bank of filters and pooling since the shape characteristics of space dividers (*e.g.* width) is not known a priori.

The found dividers decompose the point cloud into *slices* along the direction of the detection axis. We then perform the same procedure for the 2^{nd} and 3^{rd} axes *on each slice* to fully partition the point cloud. Since we process each axis independently, any divider is elongated in its original direction resulting in an over-segmented grid (see Fig. 3 (b)). This is due to the fact that we detect the dividers in a 1-dimensional manner (*i.e.* by considering one axis at a time). This reduction to 1 dimension enables us to scale to large point clouds (linearly with respect to covered area), but it cannot count for the fact that a divider may not extend across the entire building, thus leading to an over-segmentation. In order to efficiently recover the correct segmentation, we perform a series of merging operations.

3.1.2 Merging

In order to merge the over-segments, we adopt a bottom-up approach by recursively merging neighbors. We form a graph in which each oversegment is represented by a node, and edges exist between each node and its closest spatial neighbors (see Fig. 3 (c)). We then examine each edge for the existence of a divider between its incident nodes. We check this by detecting the *peak-gap-peak* on the chunk of point cloud formed by the two incident nodes using the same method of Sec. 3.1.1. If a divider is detected, the edge is removed from the graph. When all edges are examined, the surviving ones (shown in Fig. 3 (d)) denote the over-segments that should be merged. Therefore, the final spaces (Fig. 3 (e)) are the Connected Components of the graph with survived edges (each Connected Component is one space). Through transitivity, the merging operation can extend to any shape and size. In other words, any two over-segments with a path between them will be merged (*e.g.*, see the large room in the middle of Fig. 3 (a)).

In summary, by exploiting the void-based principle we developed an unsupervised, parameter-free and efficient algorithm to parse a large point cloud into disjoint spaces.

3.2. Canonical Coordinate System Among Spaces

Decomposing the raw point cloud into disjoint spaces provides geometric priors for detecting semantic elements. This is mostly because spaces have recurrent structure and layout configuration. This structure can be easily exploited by creating a common coordinate system for all spaces. Specifically, we perform the following operations on one semantic space (*e.g.* a room, hallway, etc.) to form an $x-y-z$ Cartesian reference coordinate system.

I) We choose the (z) axis of the reference system as the gravitational axis.

II) We align the x axis along the entrance to the room. Consequently, y axis will be perpendicular to the entrance wall. *(see [5] for details)*.

III) We then scale the space into a unit cube by simply normalizing the coordinates of the aligned points to range in [0,1]. This allows better generalization and information transfer across different spaces and buildings.

An example reference system is shown in Fig. 4. This procedure puts each space in a unit cube aligned across all detected spaces. It results in a geometric representation of it in a single and coherent coordinate system. Such a procedure is not straightforward in the conventional single-view 3D or RGB-D scans since global context is not captured.

4. Parsing Disjoint Spaces into Elements

Given a space in the common reference system, we wish to detect and label the semantic elements therein.

Parsing-by-Detection: Structural building analysis and augmented reality are some of the applications that benefit from parsing a point cloud into semantic elements. An analysis of such applications suggests that assuming every point must belong to one class, as in the conventional segmentation paradigm, is not a concrete assumption since it results in elements of incomplete geometry (*e.g.* hole in wall segment due to clutter). The applications can benefit from a notion of the parsed element and its structural characteristics as a whole regardless of occlusions (Sec. 1). Also, there is always a considerable number of points that either do not belong to any class or are not in the interest of the application. Hence, we argue that a more suitable approach is detecting and localizing each element, rather than segmentation.

Representing Detections: Our detection framework follows a 3D sliding window approach; we slide a set of candidate windows (boxes in 3D) for each class and classify if there is an object of the class of interest in the window. These classifiers, window sizes, and their shapes are all learned.

In order to learn the size and shape of the candidate windows, we first need a representation for 3D windows. Since the semantic spaces are normalized with respect to the common coordinate system, our candidate windows should lie in it as well. In addition to the size, we also need to represent the shape. We create K-by-K-by-K voxel grid by dividing the window into equal sized sub-3D windows and define the occupancy pattern B_i for $i \in [\mathrm{K}{-}by{-}\mathrm{K}{-}by{-}\mathrm{K}]$ as B_i is 1 if the sub-window i is part of the shape and 0 otherwise. To summarize, a candidate window is represented by its position P (location of the bottom-left corner in the common coordinate system), its 3D size S in the unit cube, and its occupancy pattern B.

To classify each candidate window as an object or not we need a set of features which can discriminatively represent the geometry and appearance of the volume specified by the window. Since our points lie in the normalized unit cube, P and S are informative about the global geometry of the window with respect to the space (global features). We also compute a set of features for each occupied sub-window as local geometry and appearance features (local features). We list our features in Table 1 and visualize them in Fig. 4.

Table 1. Features that represent each 3D window. The number in the parenthesis shows the dimensionality of the feature component.

Global Features	
P	**Position:** normalized position of the 3D window (3)
S	**Size:** normalized size of the 3D window (3)
Local Features (per voxel $l \in [K \times K \times K]$)	
B_l	**Occupancy:** 1 if l is occupied, 0 otherwise (1)
d_l	**Ratio:** ratio of the number of points in the l to the total number of points in the window (1)
C_l^r, C_l^g, C_l^r	**Color:** average color of the points in the l (3)
n_l^x, n_l^y, n_l^z	**Normal:** surface normal of the points in the l (3)
κ	**Curvature:** Surface curvature of points in the l (1)

Learning to Detect Elements: Our learning approach consists of learning candidate window shapes and learning object detectors. ***Learning candidate windows:*** In order to learn a dictionary of candidate detection windows, we compute a set of detection windows as the tightest bounding boxes and their occupancy pattern for each element in the training data. We then group this set into clusters using Affinity Propogation [11] with distance metric intersection-over-union and the occupancy pattern. After clustering, we compute a single detection window per cluster with size equal to the average of the cluster members size and occupancy pattern equal to the mode of that of the cluster members. ***Training element detectors:*** In order to learn the element detectors, we use linear classifiers such that given a feature vector Φ of the detection window and the classifier w, $\Phi^\intercal w > \tau$ means the candidate window corresponds to a semantic element. We train a linear classifier per class via LIBLINEAR [10]. Negative examples include both elements of other classes and randomly mined hard negatives.

Semantic Element Proposal Generation: Our learning procedure results in element detectors w_e and a dictionary of shapes per class. Given the learnt parameters, we use a sliding window approach to detect element proposals. At each sliding position, the SVM detectors are evaluated *for each shape atom* in the dictionary. The resulting detections are further eliminated with non-maximum suppression producing a final set of proposals as $\{(D_i, e_i, l_i)\}_{1...N}$ where D_i is the position of the detection, e_i is the label of the semantic element class, and l_i is the detection score .

4.1. Enforcing Contextual Consistency using CRF

The element proposal generation step does not exploit the context of space, as all elements are generated with no explicit consideration of others. However, there is a strong context among semantic elements since the location of one gives a prior over the location of the others. To exploit this property, we employ a graphical model based approach.

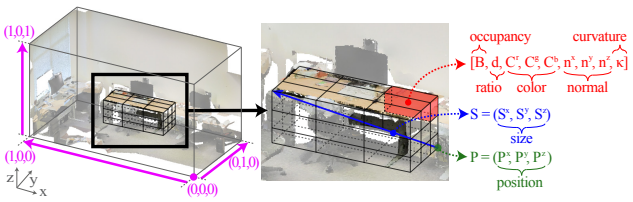

Figure 4. **Detection box in the unit cube reference coordinate system and features for a sample object (table).** Our features are the detection anchor point, size and features of each sub-box. Features of non-occupied voxels are 0 (see Table 1 for all features).

Given a collection of detection proposals, we want to choose a subset of them as the final elements. We define our model based on a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ in which the nodes correspond to the detection proposals and the edges model the relationship among elements. Each node is connected to its k_e nearest proposals from each class e. Hence, we have $(\sum_e k_e)\|\mathcal{V}\|$ edges. For each node, we want to infer if *it should be in the final detection set or not* which results in a binary label space as $y_v \in \{0, 1\}$. The edge features are $\Phi_{e=(v,w)} = [B_v, B_w, S_v, S_w, |P_v - P_w|,]$, where $|\cdot|$ is the absolute value function. The unary feature is the detection score acquired from the SVM classifier.

Following the log-linear model [16], we predict the final elements as a maximization problem of the energy function:

$$\arg\max_{y} \sum_{v \in \mathcal{V}} w_0 l_v y_v + \sum_{(u,v) \in \mathcal{E}} y_v y_u \left(w_{e_u, e_v} \cdot \Phi_{u,v}\right), \quad (1)$$

which can be written as an integer program by introducing auxiliary variables $y_{uv} = y_u y_v \quad \forall u, v \in \mathcal{V}$ as:

$$\begin{aligned} \arg\max_{y} \quad & \sum_{v \in \mathcal{V}} w_0 l_v y_v + \sum_{(u,v) \in \mathcal{E}} y_{vu} \left(w_{e_u, e_v} \cdot \Phi_{u,v}\right) \\ & s.t. y_{uv} \leq y_u \quad \forall u \in \mathcal{V}, \forall v \in \mathcal{N}(u) \\ & s.t. y_u + y_v \leq y_{uv} + 1 \quad \forall u, v \in \mathcal{E}. \end{aligned} \quad (2)$$

This maximization is performed using an off-the-shelf LP/MIP solver and the weight vectors w are learned using Structured SVM [36]. Our implementation follows the existing S-SVM-CRF implementations [17, 18, 29] and the details can be found in the supplementary [5].

4.2. Updating the Disjoint Space Parsing Results

Since 'wall' is one of the classes in the element detection step, we utilize the identified walls to update the space dividers found by the peak-gap-peak method of Sec. 3.1.1. This may recover the walls missed by the peak-gap-peak filters as the element detection step incorporates additional features, such as, color or local geometry. In a similar way to the merging operation discussed in Sec. 3.1.2, we obtain the neighbors graph of the found spaces, and for each pair of neighbors we check if there is a detected wall in the connection area; the only difference is that the walls now come from the element detection step and not the peak-gap-peak filters. We then remove edges from the graph when no wall is found and use a connected components graph to form the final space parsing (see Fig. 3 (f)).

5. Experiments

In this section, we present our experimental results and share the insights we drew from them. We also showcase them in [6].

5.1. Dataset

Our dataset is composed of five large-scale indoor areas from three different buildings, each covering approximately 1900, 450, 1700, 870 and 1100 square meters (total of 6020 square meters). These areas show diverse properties in architectural style and appearance and include mainly office areas, educational and exhibition spaces, and conference rooms, personal offices, restrooms, open spaces, lobbies, stairways, and hallways are commonly found therein. One of the areas includes multiple floors, whereas the rest have one. The entire point clouds are automatically generated without any manual intervention using the Matterport [1] scanner (only 3D point clouds; no images used by our method). Parts of these areas can be seen in Fig. 5.

We detect 12 semantic elements, which are structural elements (*ceiling, floor, wall, beam, column, window and door*) and commonly found items and furniture (*table, chair, sofa, bookcase and board*). Notice that these classes are more fine-grained and challenging than many of the semantic indoor segmentation datasets [32, 40].

5.2. Parsing into Disjoint Spaces

The qualitative results of the proposed space parsing method for several sample areas in the dataset are provided in Fig. 5. Parts (a), (g), and (e) show the raw point cloud, manually generated ground truth, and our results before the update step, respectively. Part (d) illustrates the over-segmented space before merging which shows the effectiveness of the merging step. It is worth mentioning that the hallways are sometimes over-segmented although they belong to one segment in the ground truth. This is attributed to "bottleneck" areas in some hallways which in combination with their narrow width creates the illusion of a space divider in the density histogram. However, after updating the parsed spaces such issues are resolved (Part (f)).

Baselines: We compare our method with a RANSAC-based plane fitting and a Hough transform-based line fitting methods. These approaches were used in two prominent [28, 39] papers in this area. Even though their goal is not space parsing their intermediate results can be adopted. To make the baselines appropriate for space segmentation we post process their detections and well tweaked their parameters. The results shown in Fig. 5 and Table 2 were achieved using these parameters.

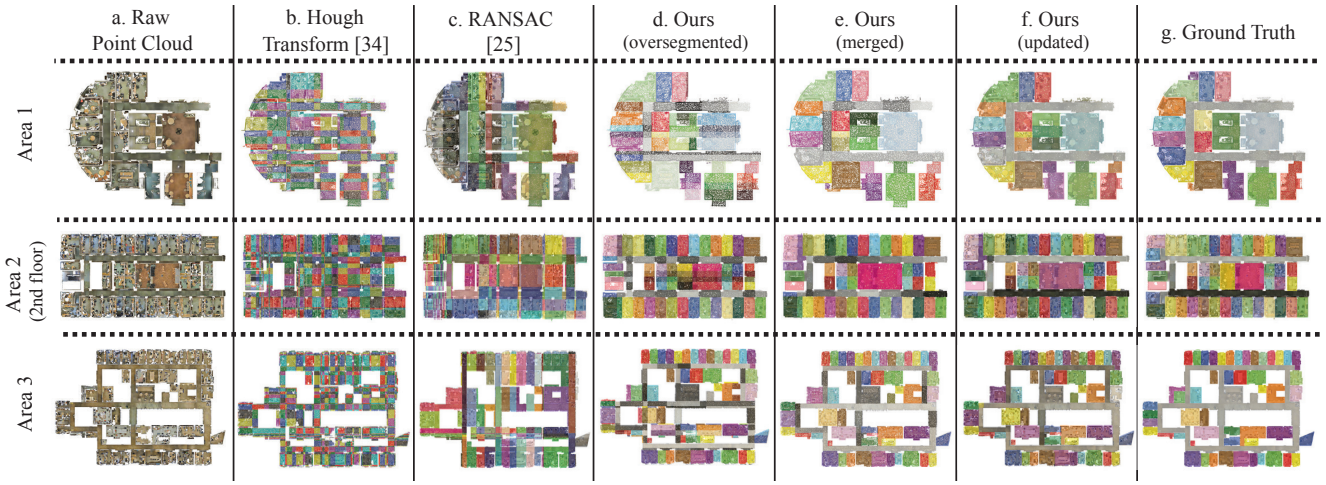

Figure 5. Space Parsing Qualitative Results.

Table 2. Evaluation of space parsing (floor plan generation).

Building	Ours final	Ours over-segm	RANSAC based [28]	2D Hough based [39]
(1)	**0.94**	0.59	0.29	0.27
(2)	**0.82**	0.76	0.30	0.31
(3)	**0.69**	0.44	0.14	0.37
(4)	**0.66**	0.42	0.15	0.3
mean	**0.77**	0.55	0.2	0.31

Quantitative Results: Table 2 provides quantitative results for space parsing. We adopt the standard unsupervised clustering metric Adjusted Rand Index (ARI) [26] as the measure. Given the ground truth and the parsing result, ARI considers all feasible matching between space labels and computes a weighted average of accuracy of each matching. Both the final and oversegmented results of the proposed method outperform the baselines.

5.3. Parsing into Semantic Elements

Baselines: We compare our method against the top performing algorithms from the KITTI object detection [12] dataset, **mBOW [7]** and **Vote3D [38]**. We only compare against the algorithms only using point clouds, not RGB-D.

In order to evaluate the contribution of each feature, we also compare against: ***No Local Geometry:*** We remove the surface normal (n_l^x, n_l^y, n_l^z), point densities (d_l) and the curvature (κ) from the feature set to evaluate the importance of local geometry, ***No Global Geometry:*** We remove the normalized position P_i^x, P_i^y, P_i^z to evaluate the importance of global geometry, ***No Color:*** We remove the RGB color values C_l^r, C_l^g, C_l^b to evaluate the importance of color.

Experimental Setup: We use k-fold strategy such that each building is a single fold. Hence, the models do not see any part of the test building during training.

Qualitative Results: We visualize the semantic elements parsed by our algorithm and the baselines in Fig. 6. Our results are provided in three different granularities: as detection boxes (h), voxelized detection boxes (g) and points (i). Our algorithm outputs the voxelized detection box (Sec. 4), and we find the others by computing the tightest bounding box and point memberships.

Fig. 6 shows that the drop in accuracy due to no color or local geometry modelling is minor, suggesting that global features are the most important ones. Moreover, the local geometry and the color modeling are more useful in fine-localizing objects, while the global geometry is particularly crucial for roughly detecting the object or labeling. This is expected since global features can only provide a very rough location. As shown in Fig. 6, although our results almost always capture the context and structure, the method sometimes fails to localize the element precisely resulting in empty areas in the voxel/point level results. This is mostly due to not including detailed features such as edges or HOG. It is also interesting to note that although the localization accuracy changes drastically when using different features, the number of objects is consistently accurate in diverse cases. We hypothesize that this can be attributed to the strong context learnt by the CRF.

Quantitative Results: For the quantitative analysis, we follow the Pascal VOC [9] detection conventions. We consider a detection box with an overlap greater than 0.5 with the ground truth as a true positive and the rest as false positive. Each detection is assigned to at most one ground truth object, and duplicate ones to the same ground truth object are taken as false positives. After computing the detection results, we draw class-level ROC curves (we defer them to [5]) and compute the mean average precision (mAP).

Table 4 provides the mAP of each algorithm and shows the relative importance of global geometry, which is consistent with our motivation of understanding the semantic geometry of a building by parsing into spaces. The appearance features help the least, which is expected since it is harder to

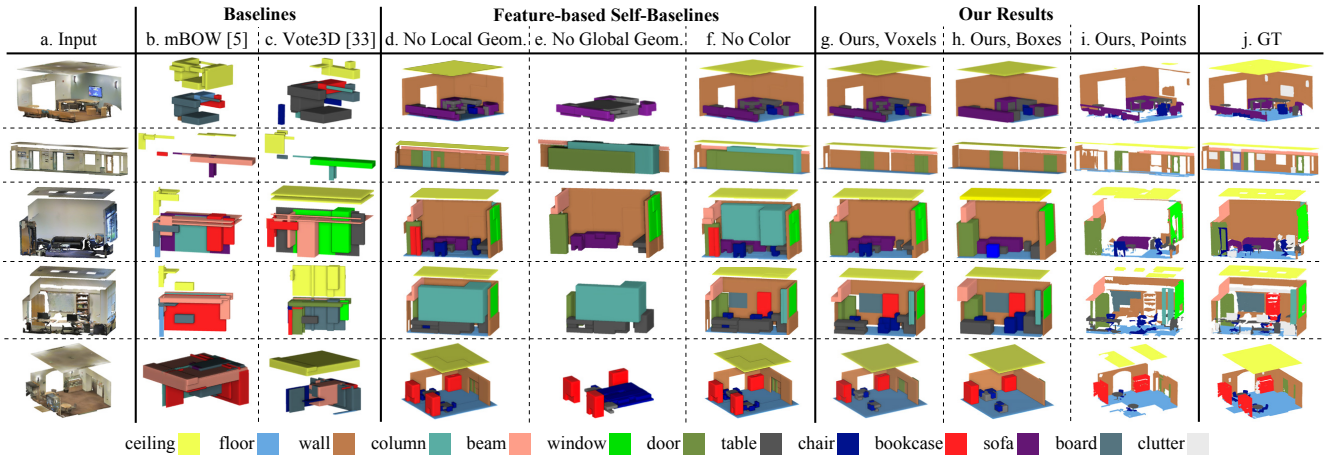

Figure 6. Qualitative results of parsing spaces into their semantic elements. Notice the heavy contribution of our global geometry features.

Table 3. Class specific average precision of our method when using different features.

	Structural Elements								Furniture						overall
	ceiling	floor	wall	beam	column	window	door	mean	table	chair	sofa	bookcase	board	mean	mean
Ours(full)	**71.61**	**88.70**	**72.86**	66.67	**91.77**	**25.92**	**54.11**	**67.38**	46.02	**16.15**	**6.78**	54.71	3.91	**25.51**	**49.93**
Ours(no glob.)	48.93	83.76	65.25	62.86	83.15	22.55	41.08	57.27	37.57	11.80	4.57	45.49	3.06	20.35	41.87
Ours(no loc.)	50.74	80.48	65.59	**68.53**	85.08	21.17	45.39	58.73	39.87	11.43	4.91	**57.76**	3.73	23.78	44.19
Ours(no col.)	48.05	80.95	67.78	68.02	87.41	25.32	44.31	59.73	**50.56**	11.83	6.32	52.33	**4.76**	25.30	45.41

Table 4. Quantitative evaluation of semantic element detection.

	mBOW [7]	Vote3D [38]	Ours			
			no local g.	no global g.	no color	full
mAP	36.11	39.21	44.19	41.87	45.41	**49.93**

generalize due to intra-class variance among different buildings. Similarly, our method's performance on structural elements is high, however on furniture is limited (see Table 3). We attribute this to the generalization of the structural elements among different buildings, something that does not apply to the same extent on furniture. Also, structural elements show a stronger spatial regularity (captured by our global features) compared to furniture.

Emerging Applications: Using the detection results we propose three emerging applications: space statistics, natural illumination modeling and space manipulation. For more details see [5].

5.4. Comparison with Conventional RGB-D

We compare our method against semantic RGB-D segmentation baselines mainly to evaluate the performance of our 3D parsing method against such techniques (results in Table 5). We also aim to answer the question whether it is better to carry out semantic parsing on RGB-D images or to perform it in 3D on the point cloud and transfer the results onto image domain. To this end, we enriched our dataset with 300 RGB-D images registered on the point cloud in a semi-automatic way and used the image-point cloud correspondences to transfer the 3D semantic annotations.

Table 5. Evaluation as RGB-D segmentation: Mean intersection-over-union of our method and [8]

method	RGB-D [8]	Ours
mIOU	20.9	**38.5**

We use the trained models[3] of [8] as the RGB-D baseline and generate the segmentation masks for our images. Similar to transferring annotations, we project the label of each point from our point-level parsing results to the RGB-D images. The results are tabulated in Table 5.

6. Conclusion

We proposed a detection-based semantic parsing method for large-scale building point clouds and argued that such 3D scans pose new challenges and potentials compared to conventional RGB-D images or small point clouds. Our approach can parse a raw point cloud into disjoint spaces and enables extraction of rich, discriminative and low-dimensional features in a common reference system. This helps with parsing spaces into their composing elements. Such a scene understanding can serve as a stepping stone for greater analysis of man-made structures both in breadth and depth and to developing systems, agents, and applications for smart indoor environments.

Acknowledgements: We acknowledge the support of NSF CAREER Grant N. 1054127, Bosch, and European Grant Agreements No. 247586 and 334241.

[3] We considered semantic classes common in both NYU-RGBD [32] and our dataset *see [5] for details*.

References

[1] Matterport 3d models of interior spaces. http://matterport.com/. Accessed: 2015-06-01.

[2] Microsoft kinect. https://www.microsoft.com/en-us/kinectforwindows/. Accessed: 2015-06-01.

[3] *Accurate Localization of 3D Objects from RGB-D Data using Segmentation Hypotheses*, 2013.

[4] M. Alzantot and M. Youssef. Crowdinside: automatic construction of indoor floorplans. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 99–108. ACM, 2012.

[5] I. Armeni, O. Sener, A. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. Supplementary material for: 3d semantic parsing of large-scale indoor spaces. http://buildingparser.stanford.edu/images/supp_mat.pdf. Accessed: 2016-04-09.

[6] I. Armeni, O. Sener, A. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. Supplementary video for: 3d semantic parsing of large-scale indoor spaces. https://youtu.be/WiYmY1S35kQ. Accessed: 2016-04-09.

[7] J. Behley, V. Steinhage, and A. Cremers. Laser-based segment classification using a mixture of bag-of-words. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4195–4200. IEEE, 2013.

[8] L. Bo, K. Lai, X. Ren, and D. Fox. Object recognition with hierarchical kernel descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1729–1736. IEEE, 2011.

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[11] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013.

[13] B. Ghanem, A. Thabet, J. Carlos Niebles, and F. Caba Heilbron. Robust manhattan frame estimation from a single rgb-d image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[14] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV 2014*, pages 345–360. Springer, 2014.

[15] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2631–2638. IEEE, 2014.

[16] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[17] H. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. 2013.

[18] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[19] C. Liu, A. Schwing, K. Kundu, R. Urtasun, and S. Fidler. Rent3d: Floor-plan priors for monocular layout estimation. In *CVPR*, 2015.

[20] T. Malisiewicz, A. Gupta, A. Efros, et al. Ensemble of exemplar-svms for object detection and beyond. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 89–96. IEEE, 2011.

[21] C. Mura, O. Mattausch, A. J. Villanueva, E. Gobbetti, and R. Pajarola. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Computers & Graphics*, 44:20–32, 2014.

[22] L. Nan, K. Xie, and A. Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (TOG)*, 31(6):137, 2012.

[23] A. Nüchter and J. Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.

[24] S. Ochmann, R. Vock, R. Wessel, M. Tamke, and R. Klein. Automatic generation of structural building descriptions from 3d point cloud scans. In *GRAPP 2014 - International Conference on Computer Graphics Theory and Applications*. SCITEPRESS, Jan. 2014.

[25] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2027–2034. IEEE, 2013.

[26] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[27] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2759–2766. IEEE, 2012.

[28] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[29] O. Sener and A. Saxena. rcrf: Recursive belief estimation over crfs in rgb-d activity videos. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[30] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Transactions on Graphics (TOG)*, 31(6):136, 2012.

[31] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 601–608. IEEE, 2011.

[32] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012*, pages 746–760. Springer, 2012.

[33] S. Song and J. Xiao. Sliding shapes for 3d object detection in rgb-d images. In *European Conference on Computer Vision*, volume 2, page 6, 2014.

[34] P. Steadman. Why are most buildings rectangular? *Architectural Research Quarterly*, 10(02):119–130, 2006.

[35] J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher. A mixture of manhattan frames: Beyond the manhattan world. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3770–3777. IEEE, 2014.

[36] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM, 2004.

[37] E. Turner and A. Zakhor. Floor plan generation and room labeling of indoor environments from laser range data, 2014.

[38] D. Z. Wang and I. Posner. Voting for voting in online point cloud object detection. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[39] J. Xiao and Y. Furukawa. Reconstructing the worlds museums. In *Computer Vision–ECCV 2012*, pages 668–681. Springer, 2012.

[40] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1625–1632. IEEE, 2013.

[41] X. Xiong, A. Adan, B. Akinci, and D. Huber. Automatic creation of semantically rich 3d building models from laser scanner data. *Automation in Construction*, 31:325–337, 2013.

[42] Y. Zhang, S. Song, P. Tan, and J. Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *Computer Vision–ECCV 2014*, pages 668–686. Springer, 2014.