OBJECT DETECTION AND CLASSIFICATION FROM CLUTTERED LARGE-SCALE INDOOR SCENE VIA ANCHOR-BASED GRAPH

Fei Su [1], Yifan Liang [1], Zhou Gang [1], Xinkai Zuo [1], Fan Yang [1], Haihong Zhu [1], Lin Li [1,2] *

[1] School of Resource and Environmental Sciences, Wuhan University, 129 Luoyu Road, Wuhan 430079, China; {sftx016, lyf0312, 2014301130059, zuoxinkai2012, yhlx125, hhzhu, lilin}@whu.edu.cn;
[2] Collaborative Innovation Centre of Geospatial Technology, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

Commission II, WG II/3

KEY WORDS: Object detection, Object classification, Graph matching, Geometric Similarity, Point Cloud

ABSTRACT:

Indoor object detection and classification from scanned point clouds has recently attracted considerable research interest. However, detecting and classifying objects with arbitrary upward orientation has emerged as a substantial challenge. This paper presents an anchor-based graph method via geometric and topological similarity among indoor objects. With this method, the misclassification that usually occurs in the objects placed non-vertical with the floor is overcome by extracting anchor in each graph via nodes' geometric attribute and by matching graph via topological relationship between nodes and anchor, rather than the features along the upward orientation. A region growing-based method along the anchor's upward orientation is proposed for classifying the unlabeled over-segmentation parts. Such an anchor-based method ensures both the accuracy of object classification and the geometric integrity of object. A series of experimental tests using three real-world 3D scans of indoor environments show the effectiveness and feasibility of the proposed method.

1. INTRODUCTION

3D indoor object detection and classification have received increasing attention in recent years (Landrieu, 2018; Mattausch et al., 2014). It is a fundamental research area for certain applications, such as autonomous vehicles (Mattausch et al. 2014; Naseer et al. 2018), indoor reconstruction (Wang et al., 2016), robotics (Breuer et al., 2011). Moreover, recent advances in scanning technology greatly accelerate data acquisition and improve the accuracy of the scanned point cloud (Wang et al., 2016; Mattausch et al., 2014). All these combined factors have contributed to the flourishment of research towards 3D indoor object recognition and scene understanding.

Indoor object classification from the scanned point clouds is still remarkably challenging (Naseer et al. 2018), and this procedure is complicated by restrictions in the data and the complexity of indoor environments, which may exhibit high levels of clutter and occlusion. Despite the advances in recent research efforts, a satisfactory solution for indoor object classification is still undeveloped, especially in the cluttered indoor area (Mattausch et al. 2014; Wang et al. 2016). Specialized object classification methods try to reduce the complexity of indoor scene by assuming that all indoor objects should be placed vertical with the floor (Mattausch et al. 2014; Nan et al. 2012), or more restrictively, all objects should have the same upward orientation (Armeni et al. 2016). Although these assumptions may hold true for some indoor objects, many elements in real world deviate from that. More recently, the focus has shifted to address these problems by segmented patch-based method (Czerniawski et al. 2018; Mattausch et al. 2014), which segment raw point cloud into a patch set and cluster these patches through their geometric attributes. These methods consider the object classification problem as a patch segmenting and clustering issue, which shows effectiveness in many cases but conducts its difficulty in classifying objects without capturing the relationship among segmentations. More recent works (Dai 2017; Qi et al. 2016; Qi et al. 2017; Shi et al. 2019) exploit object repetitions to segment the indoor scene and classify the objects by learning-based method. The limitation of these approaches is the need to carefully acquire and learn the 3D geometry of each type of object one wish to detect, which entails large amounts of time. Thus, detecting and classifying the objects with arbitrary poses without enough train sets in a cluttered environment is still a challenge for these methods.

As observation by (Spina, 2015; Laga et al., 2013; Fu et al., 2008), there is a strong correlation on geometric shape and upward orientation between functional parts (referring to anchors) in man-made objects. In the light of these observations, this study proposes an anchor-based graph method for detecting and classifying indoor object, which describe one object as a graph formed by connecting anchors with other parts. The raw point cloud is considered as the input and the labeled points representing object types are the output results.

The remainder of this paper is organized as follows. Important related works are introduced in Section 2. The proposed method is described in Section 3. Experiments on the three datasets are presented in Section 4, followed by a discussion in Section 5. Finally, the conclusions are drawn in Section 6.

2. RELATED WORKS

According to capacity of dealing with orientation of indoor objects, the current methods for object detection and classification can be classified into two groups: upward orientation-based and graph-based.

* Corresponding author

2.1 Upward Orientation-based Methods

The methods in this group assume all indoor objects placed vertical with the ground and extract the geometric features along upright orientation in each cell to classify indoor objects. The cell can be a patch, a voxel or a point.

By defining cell as a patch, those works classify the indoor objects by patch clustering methods based on Manhattan-world Hypothesis, where the point cloud is first segmented into patches. Mattausch et al. (2014) provide a patch similarity measurement and exploit a DBScan clustering in the diffusion embedding, which can automatic segment and classify the whole scene consistently. This method uses diffusion embedding to reduce geometric deviation among similar patches, which can detect both object furniture and the indoor structures, such as walls, doors and windows, especially in working environments. Valero et al. (2016) recognize the indoor furniture by comparing their height and shape templates with pre-constructed models, where shape templates differ with the type of objects, for instance, a tabletop template is a large horizontal rectangle, the chair leg template is disposed at the vertices of regular polygons or a star-like patch after projecting these legs into floor. This approach is capable of both recognizing typical indoor objects and generating semantic 3D models of furnished interiors for TLS datasets without occlusions. Czerniawski et al. (2018) employ a six-dimensional DBScan-based method to obtain better segmentations by adding points' normals into Euclidean space as the last three dimension, which successfully segment indoor structures, while show its difficulty in classifying objects without capturing the relationship among segmented patches.

By geometric features extracted from partitioned cells, some works (Tchapmi et al., 2017; Wang et al., 2017) expand the well-studied structure of 2D convolutional neural network (CNN), which has been widely used for image, into 3D CNN to segment and classify indoor objects, while the performance of these voxel-based methods is limited by the resolution of the voxels (Liang et al., 2019). Some deep learning-based researches (Li et al., 2018; Qi et al., 2017) take raw point clouds as input without extra preprocessing and directly exploit the geometric similarity among points to classify indoor objects. Although those works develop a unified architecture for applications ranging from object classification, part segmentation to scene semantic parsing, they rely heavily on the local geometric information extracted from voxels (or points) but fail to refer local relationship among them, which limits robustness.

The above methods show feasibility in detecting objects by geometric features via various cells and show their achievements in dealing with indoor objects with upward direction with respect to the ground.

2.2 Graph-based Methods

To deal with object arbitrarily oriented, graph-based methods are presented to address point cloud semantic segmentation via adjacent relationship provided by various graphs (Armeni et al., 2016; Armeni et al., 2017; Liang et al., 2019; Shi et al., 2015; Simonovsky and Komodakis, 2017; Spina, 2015; Wang et al., 2016; Wang et al., 2018).

As sliding windows show effective in reducing workload in computation for deep learning-based methods, many works (Armeni et al., 2016; Simonovsky and Komodakis, 2017; Armeni et al., 2017; Wang et al., 2018; Liang et al., 2019) use adjacent graph to enhance relationship among cells for classifying the indoor objects. Armeni et al. (2016) employ adjacent graph among voxels in sliding window for redefining the detected objects, which are extracted by geometric features similarity after partitioning indoor scenes into k-by-k-k voxel grid. Although this work can extract both indoor structures and indoor objects, its object classification still limits by the resolution of the voxels. Graph CNN-based methods (Liang et al., 2019; Simonovsky and Komodakis, 2017; Wang et al., 2018) address point cloud semantic segmentation directly on raw point clouds via extracted local features from point's geometric attributes and contextual information from the KNN graph for the center point in each fixed sliding window. As the contextual information enhances the consistency of the classified objects, use of these approaches still depend on the priori-defined upright orientation.

The cited researches showed that the objects detection and classification has been far from satisfactory, and the prominent deficiency of these method lies in finding a method available for handling non-upward direction in a cluttered indoor environment.

3. METHOD

3.1 Overview

In this section, we introduce our object detection and classification method. The proposed method uses raw point clouds as inputs and the labeled points representing object types as outputs, which consists of three main steps:

- Pre-processing: The input point cloud is first segmented into a collection of nearly-planar patches and filtered the indoor structure patches via their fitting rectangle areas. Then the anchor-based graphs in the scene are constructed by anchor extraction and patches' adjacent relationship.
- Graph clustering: The graphs are roughly clustered by their anchors' geometric similarity, followed by graph matching algorithm via super-graph to find the corresponding nodes among graphs within maximum likelihood. Then each clustered group is labeled as one type by matching graphs in this group with prepared template-graphs in $TGS = \{TG_1, TG_2, \dots TG_m\}$.
- Object refinement: Each detected object is refined by extending its anchor' fitting rectangle along normal to add the unlabeled patches into it.

3.2 Pre-processing

3.2.1 Patch Segmentation: The indoor scene is first partitioned into a set of nearly-planar patches by the region-growing method (Rabbani et al., 2012; Truong-Hong and Laefer, 2015). An initial seeding point s is selected in the area with the smallest curvature and has not been assigned to a patch. The point p will add into the patch if the following conditions are satisfied:

$$\left\| n_p \cdot n_s \right\| > \cos(\theta_{th}) \tag{1}$$

$$(p - s) \cdot n_s > dis_{(p,s)} \tag{2}$$

The point p will add into the list of potential seed points and continue to grow from the points in the list of potential seed points if the following conditions are satisfied:

$$r_p < r_{th} \tag{3}$$

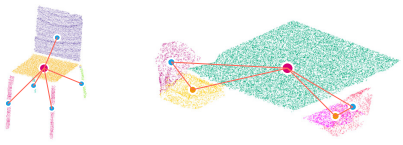
The process is iteratively applied until all the points are segmented and assigned to patches. n_s is the normal of s and n_p is the normal of p. θ_{th} is a smoothness threshold, which should be specified in terms of the angle between the normal of s and p. $dis_{(p,s)}$ is a distance threshold, which should be specified in terms of the Euclidean distance between s and p. r_p is the curvature of point p and r_{th} is a curvature threshold, which could be specified by the percentile of the sorted curvatures.

In each patch, the fitting rectangle is constructed by taking the bounding box for the patch and projecting it onto the plane spanned by its first two dominant axes to describe its geometric shape. The indoor structures that occupy a large spatial area (Ochmann et al., 2019; Wang et al., 2016; Zolanvari et al., 2018), such as the main walls, grounds and ceilings, will be filtered by their areas of fitting rectangles. As a result, the indoor scene is partitioned into a set of patches with their fitting rectangles.

3.2.2 Anchor-based Adjacency Graph Construction: A straightforward strategy to construct the adjacency graph through segmented patches is connecting every adjacent patch successively. Apparently, there must be a large number of combinations and the size of the topological graphs would be huge, which may bring difficulties for graph matching. By observations (Spina, 2015; Laga et al., 2013; Fu et al., 2008), one object can be represented by an anchor-based graph with its anchors connecting other parts in the object. As the indoor scene contains some sub-scenes as the combinations of various objects that are spatially close to each other, shown in Figure 1b, rather than independent objects shown in Figure 1a, we propose the sub-anchor to handle these cases.

Given a patch set, the adjacency graph $G(E,V)$ is constructed by connecting every adjacent patch, which can be considered as one object or one sub-scene with some objects, where V and E denote the node set and edge set in the graph which contain node $v_i \in V$ and edge $e_i \in E$. In each graph, a node with its fitting rectangle area higher than the threshold (0.3 in our experiments) and neighbor size more than 1 is referred to sub-anchor v^{SA}. The v^{SA} with the maximum number of neighbors in each graph is assigned as v^A. If two adjacent nodes are both assigned as v^{SA}, the node with larger angle between its normal and v^A's normal will be removed. Finally, the anchor-based graphs are constructed by connecting v^As and v^{SA}s with all their adjacent nodes.

The adjacency graph of one chair is shown in Figure 1a and the adjacency graph of a sub-scene with one table and two chairs is shown in Figure 1b.



(a) (b)

Figure 1. The anchor-based graphs: (a) the graph of one chair, where red node represents the anchor v^A and yellow line between red node and blue node represents edge; (b) the graph of a sub-scene with one table and two chairs, where yellow node represents the sub-anchor v^{SA}.

3.2.3 Graph Descriptors: The attributes of v_i compose anchor flag, center point and the features of corresponding patch. Anchor flag represents its node type, 1 for v^A, 2 for v^{SA} and 0 for other type. As patch representation is to reduce the complexity of the input data, the segmented patches may suffer over-segmentation problem. Thus, for each node, we compute the geometric features mainly from its fitting rectangle, as shown in Figure 2. The node features used in present work are shown in Table 1.

F	Definition
F_1	Area: w*l
F_2	Ratio of width to length: w/l
F_3	Ratio of areas: The ratio between area of the patch and F_1
F_4	The distance from the centroid of the patch to the fitting rectangle.
F_5	Non-planarity: $F_4/(F_4 + l + w)$

Table 1. The node features.

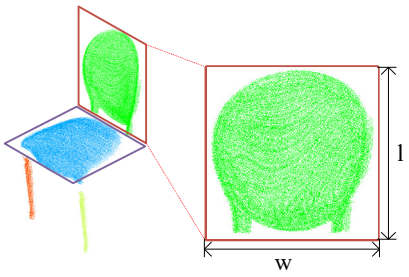


Figure 2. The schematic diagram of the parameters in Table 1. The red box of the patch represents the fitting rectangle, whose length is l and width is w.

The attributes of e_i include edge flag and the features for the relationship between v_i and its v^A (or v^{SA}). Edge flag give its edge's type, 1 for the edge between v^A and v^{SA} and 0 for other edge type. The edge features used in present work are shown in Table 2. The distance is the Euclidean distance between two center points in e_i. The edge orientation is the vector between two center points of nodes in e_i.

C	Definition
C_1	Distance
C_2	Angle between orientation and v^A's normal
C_3	Angle between orientation and v_i's normal
C_4	Angle between v_i's and v^A's normal

Table 2. The edge features.

3.3 Graph Clustering

The indoor scene has become a set of anchor-based graphs $GS = \{G_1, G_2, \dots G_n\}$ in the last section. Thus, the object classification problem can be considered as assigning each node from graph in GS with one type among type-set $TS = \{T_1, T_2, \dots T_m\}$, which contains three parts: rough clustering via anchor similarity, graph clustering via super-graph and clusters labeling.

3.3.1. Rough Clustering: The graphs in GS are first clustered roughly by anchor similarity. This procedure starts by selecting two graphs randomly. Let $G(E,V)$ and $G_t(E_t,V_t)$ be the selected graphs with corresponding anchor v^A and v_t^A, where the selected graph can be one object or one sub-scene. Inspired by Mattausch et al. (2014), an anchor geometric similarity measurement is performed on v^A and v_t^A, which can be expressed as:

$$S_s(v^A, v_t^A) = e^{-2\sum_{u=1}^{5}(1-\rho_u(v^A, v_t^A))} \quad (4)$$

where $\rho_u(v^A, v_t^A)$ can be presented as:

$$\rho_u(v^A, v_t^A) = \left| \min\left(\frac{F_u^{v^A}}{F_u^{v_t^A}}, \frac{F_u^{v_t^A}}{F_u^{v^A}} \right) \right| \quad if\ u = 1,2,3,4,5 \qquad (5)$$

Once the $S_s(v^A, v_t^A)$ is more than the threshold α_1, $G(E,V)$ and $G_t(E_t, V_t)$ have the similar anchor and these two graph will be clustered into a group.

This processing is iteratively performed until all graphs in GS have been grouped and the graphs with similar anchor are clustered as $GS = \{GC_1, GC_2, \dots GC_n\}$.

3.3.2. Graph Clustering: For $GC_i \in GS$, an initial seed graph $G_t(E_t, V_t)$ is selected with the maximum number of anchor's neighbors and highest F_3 of its anchor. The first indicator ensures the edge's integrity of the selected graph while the second indicator exploits the anchor's geometric completeness via F_3. Then the proposed graph matching algorithm will be performed on the super-graph $\bar{G}$ that constructed by G_t and other graph G in GC_i, this graph matching algorithm will be illustrated in Section 3.3.3. The returned sub-graph $s\bar{G}$ after matching represents objects with the same type and will be partitioned into two pieces G_t' and G' along its axis of symmetry. After all graphs in GC_i have been tested with G_t, all G's are clustered as one group C_i, meanwhile, all G_t's are remerged into one graph and added this remerged graph into current group C_i. A new seed graph will be selected from the remaining graphs and continued to match the remaining graphs in GC_i.

The processing above repeats until all graph set GC_i in GS have been clustered and indoor scene have been classified into a cluster set $CS = \{C_1, C_2, \dots C_n\}$, each C_i represents one type of objects.

3.3.3. Graph Matching: Given two graphs $G(E,V)$ and $G_t(E_t, V_t)$, the graph matching can be seen as finding the optimal correspondence nodes among them, which can be represented as a binary indicator matrix $X \in \{0,1\}_{|V|\times|V_t|}$. If $v_i \in V$ matches $v_j^t \in V_t$, the corresponding entry of X is 1, e.g., $X(i,j) = 1$; 0 otherwise. After transferring the matrix into a vector, $x \in \{0,1\}_{|V||V_t|\times 1}$, the graph matching between G and G_t can be formulated to find the optimal correspondences x^* that maximizes the matching similarity between G and G_t, which can be stated as:

$$x^* = argmax_x(S(x|G, G_t)), x \in \{0,1\}_{|V||V_t|\times 1} \qquad (6)$$

where $S(x|G, G_t)$ is a function measuring the matching similarity between G and G_t under corresponding node-pairs indicator x, which conducts that the maximum of $S(x_i|G, G_t)$ can be found by traversing all combination of node-pairs between G and G_t.

As graph matching between G_t and G is formulated as searching all possible corresponding node-pairs from G_t and G for maximizing $S(x_i|G, G_t)$, we present a super-graph $\bar{G}(\bar{E}, \bar{V})$ to represent all corresponding node-pairs between G_t and G, as shown in the black box in Figure 3. A super graph $\bar{G}(\bar{E}, \bar{V})$ is constructed by connecting v_t^A with v^A via an edge and connecting other node $v_i \in V$ with $v_j^t \in V_t$ by a virtual edge to represent the corresponding node-pairs between G and G_t. $\bar{G}(\bar{E}, \bar{V})$ contains all nodes and edges in G and G_t, the reconstructed $\bar{G}(\bar{E}, \bar{V})$ is shown in Figure 3.

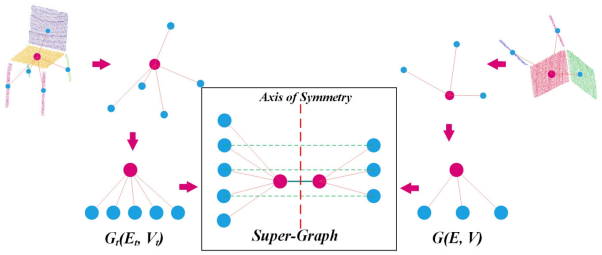


Figure 3. The schematic diagram of the super-graph.

After reconstructing $\bar{G}$, $S(x_i|G, G_t)$ can be rewritten as $L(e_1', e_2', \dots, e_k'|\bar{G})$, where e_m' conducts the corresponding node-pair and k is the size of node-pairs, which is reduced from $|V|$ to β_1 successively, as descripted in Eq.7. Thus, finding the optimal x^* can be stated as finding the sub-graph sG from $\bar{G}$ maximizing $L(e_1', e_2', \dots, e_k'|\bar{G})$. The sG searching details are shown in Algorithm 1.

$$L(e_1', e_2', \dots, e_k'|\bar{G}) = \frac{\sum_{m=1}^{k} L(e_m'|\bar{G})}{k} * \frac{k}{|V|} = \frac{\sum_{m=1}^{k}(w_s S_s(e_m') + w_c S_c(e_m') + w_p S_p(e_m'))}{|V|} \qquad (7)$$

where $S_s(e_m')$ captures geometric similarity between node-pair in e_m', $S_c(e_m')$ measures connective similarity between edge-pair in e_m' and $S_p(e_m')$ is the splitting/merging penalty for nodes in e_m'; e_m' is a virtual edge between $v_{im} \in V$ and $v_{jm}^t \in V_t$; w_s, w_c and w_p are scalar weights that satisfy $w_s + w_c + w_p = 1$. $S_s(e_m')$, $S_c(e_m')$ and $S_p(e_m')$ are defined by Eq. 8, Eq. 9 and Eq. 11, respectively.

$$S_s(e_m') = S_s(v_{im}, v_{jm}^t) \qquad (8)$$

$$S_c(e_m') = e^{-2\sum_{u=1}^{4}(1-\psi_u(e_{im}, e_{jm}^t))} \qquad (9)$$

where $\psi_u(e_{im}, e_{jm}^t)$ can be presented as:

$$\psi_u(e_{im}, e_{jm}^t) = \begin{cases} \min\left(\frac{C_u^{e_{im}}}{C_u^{e_{jm}^t}}, \frac{C_u^{e_{jm}^t}}{C_u^{e_{im}}} \right) & if\ u = 1 \\ \left| \cos(C_u^{e_{im}} - C_u^{e_{jm}^t}) \right| & if\ u = 2,3,4 \end{cases} \qquad (10)$$

A v^{SA} has a high likelihood to be an anchor for its neighbor. Thus, directly assigning v^{SA} as one part of certain object may lead to misclassification for its adjacent objects. Inspired by (Alhashim et al. 2015), the anchor likelihood for v^{SA},ζ_a^v, is proposed to compute the probability of merging v^{SA} into current graph, which can be calculated as the ratio between the volume of the external cuboid after and before splitting v^{SA} and v^{SA}'s neighbors from the graph. The smaller ζ_a^v is, the less the penalty for assigning v^{SA} to current graph becomes. ζ_a^v of node with anchor flag equaling 0 is 0. $S_p(e_m')$ is stated as:

$$S_p(e_m') = \begin{cases} 1 - \zeta_a^{v_{im}}, if\ \zeta_a^{v_{im}} \geq \zeta_a^{v_{jm}^t} \\ 1 - \zeta_a^{v_{jm}^t}, if\ \zeta_a^{v_{im}} < \zeta_a^{v_{jm}^t} \end{cases} \qquad (11)$$

Each of the three similarity measurement might individually favor different influence on the final clustered results. In this paper, we place more value on the S_c rather than S_s which show sensitive to the clutter and missing parts. The weights used in our experiments are $w_s = 0.3, w_c = 0.5, w_p = 0.2$.

Algorithm 1. Node searching algorithm

Input: $\bar{G}$
Output: sub-graph sG
Set β_1, β_2
initial $maxscore = 0$, $n_{ve} = |V|$
repeat
  form sG by selecting n_{ve} virtual edges in $\bar{G}$
  compute $L(e'_1, e'_2, \dots, e'_k|\bar{G})$ for sG by Eq. 7
  if $L > \beta_2$ & $L > maxscore$
    $maxscore = L$
until all combinations have been tested
if $n_{ve} > \beta_1$
  $n_{ve} = n_{ve} - 1$
  go to line 5
return sG with $L == maxscore$

3.3.4 Clustering Labelling: As the scene have been partitioned and grouped into a cluster set $CS = \{C_1, C_2, \dots C_n\}$, in this section, each cluster $C_i \in CS$ will be labeled as one type $T_i \in TS$ by matching graphs in C_i with template-graphs in $TGS = \{TG_1, TG_2, \dots TG_m\}$, which is constructed by the objects in Alhashim et al. (2015).

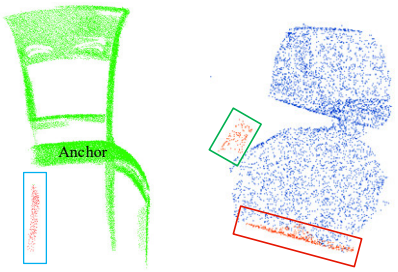
Each graph G in CS is compute the similarity with template-graphs in TGS via graph matching algorithm in Section 3.3.3 with setting $|V|$ as β_1. G is labeled as T_i when G and TG_i obtain the maximum similarity score. C_i is labeled as T_j if most of graphs in C_i are labeled as T_j, and all graphs in C_i will be relabeled as T_j.

As tables are usually surrounded by chairs that cause the structure to be occluded under tabletops, the unlabeled patch with a relatively large area surrounded by more than two chairs is labelled a table.

3.4 Object Refinement

Although most patches have been classified, some extracted patches are still unlabeled for three reasons. First reason is that this patch may contain missing parts, especially for the legs of the chairs or desks, as shown in Figure 4a. Second reason is that the patch may suffer over-segmentation, as shown in red box in Figure 4b. The last reason is that some tiny parts of the objects, such as the chair handrails, are removed during the graph matching processing, as shown in green box in Figure 4b, since the handrails in other chairs may not be scanned, and in turn, no matching on handrails among the chairs has been found.

Under the definition of the anchor, the legs and tiny parts of the objects tend to be covered by the oriented bounding box of anchor along its normal, as shown in Figure 4. Thus, we extend each anchor's fitting rectangle with a distance of β_3 along its normal. Once an extending box attach with an unlabeled patch, this patch will be merged into this object.



(a) (b)

Figure 4. The unclassified cases: (a) missing parts in the chair leg (blue box), (b) the anchor over-segmentation (red box) and chair's tiny parts (green box).

4. EXPERIMENTS

The proposed method was tested on three real datasets of indoor scenes, as shown in Figure 5a.The statistics for these datasets were shown in Table 3. The algorithm was implemented in C++ by Cloud Compare and MATLAB. All the experiments were performed on a 3.60 Hz Intel Core i7-4790 processor with 12 GB of RAM.

Dataset-1 and -2 were taken from the S3DIS dataset, captured by Matterport scanner (Armeni et al., 2016). Dataset-3 was obtained by hand-held active-light scanner (MantisVision Inc.) (Nan et al., 2012). Clutter and occlusion were present in these datasets. Dataset-1 and -2 were obtained by RGBD, the density of point clouds was moderate. Dataset-1 and -2 provided highly detailed objects, while a large amount of data were still missing due to occlusions and restricted accessibility. Dataset-1 was a conference room with various chairs and conference table, while Dataset-2 was a large-scaled cluttered environment containing some office rooms (part-1 and -2) and storage (part-3). Dataset-1 and -2 were tested for common office environment with missing data and Dataset-3 was tested for cluttered and noise environment, which contains objects with various poses.

Quantitative evaluations on the classified results were conducted by using three metrics: completeness, correctness and quality.

$$Completeness = \frac{TP}{TP + FN} \tag{12}$$

$$Correctness = \frac{TP}{TP + FP} \tag{13}$$

$$Quality = \frac{TP}{TP + FN + FP} \tag{14}$$

where TP represents true positives, which refer to the number of objects detected both in classified result and ground truth; FP represents false positives, which refer to the number of classified objects that couldn't be found in the ground truth; and FN represents false negatives, which refer to the number of unclassified ground-truth objects.

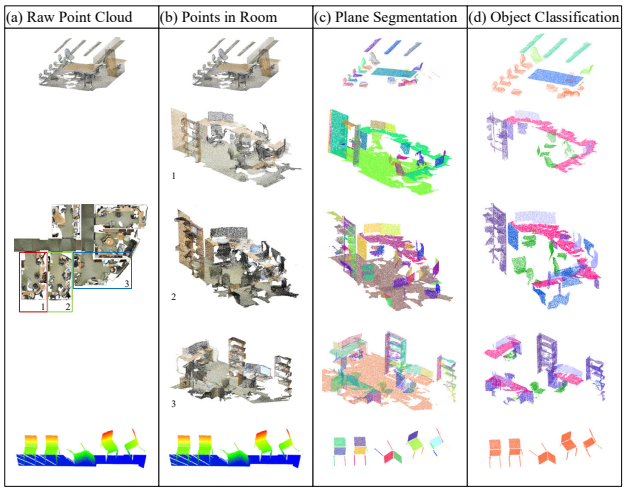


Figure 5. Qualitative results of indoor object classification. (a) Original data (certain datasets have the ceilings removed for clarity). (b) Side view of data in one room. (c) The patch segmentation result of one room. (d) The classification results in one room (One colour represents one type of objects).

5. DISCUSSION

In Figure 5, we showed the experimental results of proposed method. The original point clouds were shown in first column. The points in each room of input data were conducted in the second column. The patch segmentations were shown in third column and the classified results in one room were shown in forth column (One colour represents one type of objects).

For the quantitative analysis on the classified results, the results of three metrics on different type of objects were shown in Table 4. As shown in Table 4, all of chairs in each dataset had a good correctness, which indicated that all classified chairs could be detected in both the raw data and ground truth. The completeness and quality metrics of all datasets on the chairs were higher than 0.8, except for Dataset-2 (part-3), which showed our method was effective with classifying indoor chairs. However, the unclassified chairs occurred, as shown in the red box in Figure 6, in while almost all parts except the back of this chair were missing due to occlusion.

As Table 4 shows, most of tables in the indoor scene could be detected and classified except for Dataset-2 (part-1) since the L-shaped table in Dataset-2 (part-1) was over-segmented and labeled as two long tables for their similar geometric shape. The bookcases were relatively easy-classified objects for its relatively large volume and well-bedded topological graph, which deviated from other objects. The dataset-3 contained various chairs with different poses, and our method was performed well in this dataset.

These results showed that the proposed method was robust for detecting and classifying indoor objects, even with various upward orientation. However, the test on dataset-2 (part-1) indicated that the patch segmentation method encountered difficulties with L-shape table.

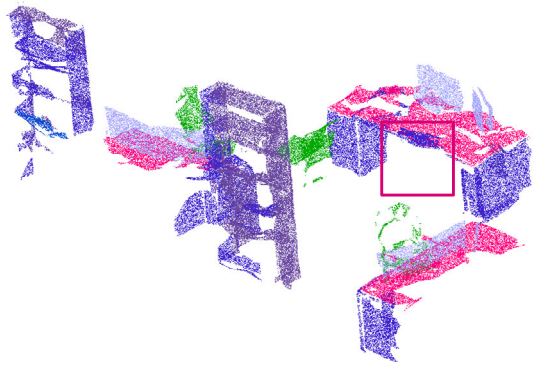


Figure 6. The failure cases in Dataset-2 (part-3).

6. CONCLUSIONS

In this work, an anchor-based graph matching method is proposed for detecting and classifying the indoor objects with freely upward orientation. The graphs are matched by performing graph rough clustering via anchor similarity, super-graph segmentation via graph similarity and object geometric refinement successively.

The proposed method was tested with three real indoor scenes. The experiments showed that the proposed method could achieve indoor objects classification without training dataset. The quantitative result of the experiments showed that the object classification precision with almost all completeness, correctness and quality above 0.8. These findings show that the presented method is appropriate for arbitrary upward oriented object classification. The experiments show the effectiveness and availability of the proposed method.

However, the presented method currently can only segment planar surfaces and shows its weakness with L-shape table classification. Those topic will be our future study.

ACKNOWLEDGEMENTS

This study is funded by the National Natural Science Foundation of China (41871298) and the National Key R&D Program of China (2017YFB0503701).

The authors acknowledge Iro Armeni (Armeni et al. 2017; Armeni et al. 2016) for the acquisition of the 3D point clouds. The authors would like to gratefully acknowledge Nan (Nan et al. 2012) for their help.

Test Sites	Rooms	Clutter	Points	Area(m^2)	Relative Accuracy	From
Dataset-1	1	Moderate	1,136,617	22.8	2-3 cm	Matterport3D
Dataset-2	6	Moderate	6,160,304	161.8	2-3 cm	Matterport3D
Dataset-3	1	Moderate	167,768	-	1 cm	MantisVision Inc.

Table 3. Description of the datasets.

Test Sites	Chair			Table			Bookcase		
	Com.	Cor.	Qua.	Com.	Cor.	Qua.	Com.	Cor.	Qua.
Dataset-1	0.83	1	0.83	1	1	1	-	-	-
Dataset-2(part-1)	0.8	1	0.8	0	0	0	1	1	1
Dataset-2(part-2)	0.83	1	0.83	1	1	1	0.5	1	0.5
Dataset-2(part-3)	0.75	1	0.75	1	1	1	1	1	1
Dataset-2(all)	0.8	1	0.8	0.8	0.8	0.67	0.8	1	0.8
Dataset-3	1	1	1	-	-	-	-	-	-

Table 4. Quantitative results of the datasets.

REFERENCES

Alhashim, I., Li, H., Xu, K., Cao, J., Ma, R., & Zhang, H. (2014). Topology-varying 3D shape creation via structural blending. ACM TRANSACTIONS ON GRAPHICS, 33, 1-10.

Alhashim, I., Xu, K., Zhuang, Y., Cao, J., Simari, P., & Zhang, H. (2015). Deformation-driven topology-varying 3D shape correspondence. ACM TRANSACTIONS ON GRAPHICS, 34, 1-13.

Armeni, I. et al., 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces, Computer Vision and Pattern Recognition, Las Vegas, NV, pp. 1534-1543.

Armeni, I., Sax, S., Zamir, A.R., & Savarese, S. (2017). Joint 2D-3D-Semantic Data for Indoor Scene Understanding. arXiv:1702.01105.

Breuer, T., Macedo, G.R.G., Hartanto, R., Hochgeschwender, N., Holz, D., Hegger, F., Jin, Z., Müller, C., Paulus, J., & Reckhaus, M. (2011). Johnny: An autonomous service robot for domestic environments. JOURNAL OF INTELLIGENT & ROBOTIC SYSTEMS, 66(1), 245-272.

Czerniawski, T., Sankaran, B., Nahangi, M., Haas, C., & Leite, F. (2018). 6D DBSCAN-based segmentation of building point clouds for planar object classification. AUTOMATION IN CONSTRUCTION, 88, 44-58.

Dai, A. (2017). ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. 2017 IEEE Conference on Computer Vision and Pattern Recognition. arXiv:1702.04405

Kasper, A., Jäkel, R. and Dillmann, R., 2011. Using Spatial Relations of Objects in Real World Scenes for Scene Structuring and Scene Understanding, Proceedings of the 15th International Conference on Advanced Robotics, Tallinn, pp. 421-426.

Laga, H., Mortara, M., & Spagnuolo, M. (2013). Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes. ACM TRANSACTIONS ON GRAPHICS, 32(5), 1-16.

Lai, K., Bo, L., Ren, X., & Fox, D. (2011). A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In, IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011. doi: 10.1109/ICRA.2011.5980382

Lai, K., & Fox, D. (2010). Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation. The International Journal of Robotics Research, 29, 1019-1037

Landrieu, L.S.M. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4558-4567.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., & Chen, B. (2018). PointCNN: Convolution On X-Transformed Points. arXiv:1801.07791

Liang, Z., Yang, M., Deng, L., Wang, C. and Wang, B., 2019. Hierarchical Depthwise Graph Convolutional Neural Network for 3D Semantic Segmentation of Point Clouds. International Conference on Robotics and Auomation, Montreal, Canada, 20-24 May 2019.

Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., & Pajarola, R. (2014). Object Detection and Classification from Large-Scale Cluttered Indoor Scans. COMPUTER GRAPHICS FORUM, 33, 11-21.

Nan, L., Xie, K., & Sharf, A. (2012). A Search-Classify Approach for Cluttered Indoor Scene Understanding. ACM TRANSACTIONS ON GRAPHICS, 31, 1-10.

Naseer, M., Khan, S.H. and Porikli, F., 2018. Indoor Scene Understanding in 2.5/3D for Autonomous Agents: A Survey, IEEE Access, 7: 1859-1887.

Qi, C.R., Su, H., Mo, K., & Guibas, L.J. (2016). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. arXiv:1612.00593.

Qi, C.R., Yi, L., Su, H., & Guibas, L.J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. arXiv: 1706.02413.

Shi, Y., Chang, A.X., Wu, Z., Savva, M., & Xu, K. (2019). Hierarchy Denoising Recursive Autoencoders for 3D Scene Layout Prediction. arXiv: 1903.03757.

Simonovsky, M., & Komodakis, N. (2017). Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. arXiv: 1704.02901.

Spina, S. (2015). Graph-based segmentation and scene understanding for context-free point clouds. In: University of Warwick

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. Computer Vision and Pattern Recognition, pp. 945-953). arXiv:1505.00880

Truong-Hong, L. and Laefer, D.F., 2015. Quantitative evaluation strategies for urban 3D model generation from remote sensing data. Computers & Graphics, 49(C): 82-91.

Verdoja, F., Thomas, D. and Sugimoto, A., 2017. Fast 3D point cloud segmentation using supervoxels with geometry and color for 3D scene understanding, IEEE International Conference on Multimedia and Expo, pp. 1285-1290.

Wang, J., Xie, Q., Xu, Y., Zhou, L., & Ye, N. (2016). Cluttered indoor scene modeling via functional part-guided graph matching. COMPUTER AIDED GEOMETRIC DESIGN, 43, 82-94.

Wang, P., Liu, Y., Guo, Y., Sun, C., & Tong, X. (2017). O-cnn: Octree-based convolutional neural networks for 3d shape analysis. ACM Transactions on Graphics (TOG), 36(4), 72.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., & Solomon, J.M. (2018). Dynamic Graph CNN for Learning on Point Clouds. arXiv: 1801.07829.