2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

3D Object Recognition from large-scale point clouds with global descriptor and sliding window

Naoyuki Gunji, Hitoshi Niigaki, Ken Tsutsuguchi, Takayuki Kurozumi, and Tetsuya Kinebuchi
NTT Media Intelligence Laboratories, NTT Corporation
1-1 Hikari-no-oka, Yokosuka, Kanagawa, Japan
Email: {gunji.naoyuki, niigaki.hitoshi, tsutsuguchi.ken, kurozumi.takayuki, kinebuchi.t}@lab.ntt.co.jp

***Abstract*—We propose a novel method for the recognition of objects that match a given 3D model in large-scale scene point clouds captured in indoor environments with a laser range finder. Since large-scale indoor point clouds are greatly damaged by noise such as clutter, occlusion, hole, and measurement errors, it is difficult to exactly identify local correspondences between points in a target model point cloud and points in a scene point cloud, based on similarities between local descriptors computed at keypoints on both point clouds. To avoid such a problem, we suggest to utilize sliding window in order to match the input model and pieces of scene point clouds, both of which are represented with Bag-of-Features(BoF). A BoF representation of a window is efficiently calculated by using the integral image, which stores accumulated BoF vectors. Though BoF is robust to partial noises, it does not preserve any spatial information. Then, we propose a method to make a global descriptor of a window which is almost invariant to horizontal rotations of an object inside the divided window and roughly preserves spatial information by dividing sliding window into several parts. Experiments on real world data show that our approach offers better performance than a baseline method in terms of precision and recall.**

I. Introduction

Recent advances in laser range finders makes it easier to capture 3D large-scale point clouds in extensive indoor environments. Annotations to large-scale scene point clouds captured in buildings about where and what kind of objects there are would help robots move by themselves and conduct tasks that involve search and interaction with objects. In this work, in order to make such indoor environmental maps, we propose a method that detects input models in large-scale scene point clouds.

The task of 3D object recognition from unorganized point clouds has been studied widely for a long time. Previous works can be roughly divided into two types as follows. The first estimates 6Degree-of-Freedom poses of given specific models in environmental scenes([1], [4], [8], [9], [14], [16], [26], [27]). The other segments potential objects from scene and classifies them into target categories([6], [7], [18], [19]).

In works of the first type, models are usually not contaminated by noises so that it is easy to describe and exactly match their local shapes around detected keypoints with local descriptors. In this case, correspondence between models and scenes is calculated based on similarities of local descriptors. Then, translations and rotations of the input models are estimated from point-to-point matchings by methods such as RANSAC [16], Geometric Consistency [3] or Hough voting [27]. However, large-scale indoor point clouds, which we consider in this work, contain much noise such as clutter, occlusions, holes, and measurement error. Since this noise obscures the details of object shape, it is hard to estimate accurate normals at all points in the cloud. Furthermore, erroneous normals degrade local descriptors and local reference frames, which utilize normals to describe shape information and make it difficult to obtain the exact point-to-point matches essential for pose estimation. Therefore, recognition methods based on local shape matching tends to fail in large-scale point clouds.

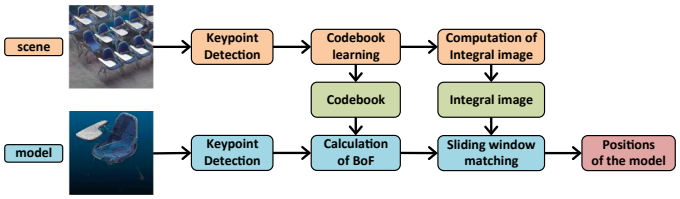


Fig. 1. Outline of the proposed method. Our method takes a model point cloud and a scene point cloud, and outputs positions of the input model in the form of bounding boxes.

Methods of the second type cut out individual objects from a scene point cloud at first and classify them with classifiers obtained by supervised training with manually labeled data. In order to segment objects from a background, a clustering method like SuperVoxels [17] or plane removal by RANSAC [22] is utilized. If the scene is simple like a table-top scene, it is easy to segment those pieces of the point cloud that represent objects from the scene. However, in general, 3D object segmentation is itself a hard task in complex and noisy scenes. Furthermore, unlike the case of 2D image data, it is not easy to construct 3D training data even if it is small in size. It is necessary to rotate the point cloud many times, carefully divide the point cloud into individual objects, and annotate these objects.In addition, if the input scene changes, it is necessary to make training data and train new classifiers for a new scene in order to avoid degradation in recognition performance.

Then, we propose to represent the model and pieces of the scene point cloud with BoF and calculate their similarities by the sliding window technique. Since BoF is robust to partial noises or lack of information and does not require exact match of local descriptors, it is a suitable feature representation for noisy point clouds. Though Spatial Pyramid Matching(SPM) [10] is a popular method to incorporate spatial information into BoF representation, inappropriate partitioning of the sliding window deprive the global description of rotational invariance to object inside the window. Then, we introduce a novel partitioning method, which keeps the global description com-

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

putationally efficient and almost invariant to rotation of objects included in the window. BoFs of pieces of the scene point cloud can be computed efficiently with the integral image that stores precomputed sum of BoF vectors. In order to avoid the tedious task of making training data, our approach is requires only a target model point cloud, which is cropped from a scene point cloud, and recognize the target objects in the scene based on the similarity of BoF. Although this strategy requires target model segmentation from the scene, it is not necessary to gather training data, to train classifiers, and to segment individual objects from a scene point cloud.

II. RELATED WORKS

Many 3D object recognition approaches based on local shape similarities have been proposed. These approaches establish correspondence between interesting points on the model and counterparts in the scene via similarities of local descriptors. Geometric Consistency [3] makes clusters of geometrically consistent point-to-point matches and estimates the translation and rotation of the model based on the clustered correspondences. 3D Hough voting [27] estimates positions of a reference point defined in the model using relative position between the reference point and interesting points on the model. It has been shown that these methods work well on small-scale and clean datasets [14]. However, large-scale indoor point clouds differ from these datasets in that the former are generally damaged by a lot of noise. Since 3D local descriptors are sensitive to subtle deformation of point clouds, it is difficult to estimate correct point-to-point correspondences between the model and the scene, making pose estimation based on local shape matching fail. In this work, in order to avoid such difficulties and recognize objects precisely, we propose to utilize local shape information not in the form of raw descriptor, but in the form of a global shape feature such as BoF, which is known to be robust to loss of partial information due to noise such as holes and occlusion. Although we use BoF in this work, it is possible to use other coding methods such as Locality-constrained Linear Coding (LLC) [29]. In our experiments, we show that the proposed global feature based method performs significantly better than local similarity based methods.

BoF-based representations have been used to express separated 3D models in 3D object retrieval. Toldo et al. [25] proposed a retrieval method that divides 3D models into several parts by clustering and expresses each part with BoF. Bronstein et al. [2] introduced (SS-BoF) to represent 3D models, which incorporates relative positions of local descriptors to sophisticate BoF and to improve retrieval performance. In 3D object retrieval, 3D models are separated from a scene point cloud so that BoF-based approaches can be applied straightforwardly. However, in our case, it is not appropriate to express the whole scene point cloud with a single BoF vector, which contains many 3D objects. Therefore, we compute BoF from local descriptors of a part of the scene and compare it with the BoF of the model. In order to calculate BoF representations and similarities, we combine the sliding window approach and integral image. Integral image is utilized in face detection [28] from images and enables the efficient calculation of feature values. Utilizing integral image which stores accumulated BoF vectors, this work efficiently computes BoF representations and similarities of BoF vectors between the input model and a part of the large scene.

Pang and Neumann [15] proposed a method with sliding window for object recognition in a large-scale point cloud. They discretize the point cloud with regular grids and transform the cloud into 3D images. After preprocessing, they train weak classifiers via Adaboost training, which selects characteristic shapes of target objects. With trained weak classifiers and integral image calculated for the voxelized scene, they succeeded in recognizing various instruments in a large-scale point cloud captured in an industrial plant without segmentation of them from a background point cloud. An important difference between our method and their method is that no training procedure other than codebook learning for BoF is needed. While they utilize Haar-like features learned from training data, we use existing local descriptors and BoF to represent 3D objects. Accordingly, in our approach, it is possible to choose and incorporate local descriptors off the shelf into our approach according to characteristics of the model and the scene.

In the RGBD image domain, Sliding Shape [24] has been proposed for object detection. Song et al. showed how to train Exemplar-SVMs [13] for each object and achieved performance superior to other methods. In order to cope with pose variances of target objects, their method requires training of many Exemplar-SVMs, each of which accounts for the same object in a different pose. However, our method represents objects inside sliding window with rotationally invariant global descriptor, it is sufficient to scan the scene with a single window adjusted to suit the size of the model.

Many kinds of local descriptors for the description of 3D shapes of point clouds have been proposed. For example, Spin Image [8], SHOT [26], and Fast Point Feature Histograms (FPFH) [20] are widely used. FPFH and SHOT use the distribution of normals around a point of interest to describe 3D shape information. These are implemented in Point Cloud Library (PCL) [21]. In this work, we use FPFH of PCL as a local descriptor of BoF.

III. PROPOSED METHOD

In this section, we detail the proposed approach. Our method is mainly composed of two parts. One is preprocessing of the scene point cloud, which includes unsupervised training of a codebook and computation of a 3D integral image. We compute 3D interal image over the 3D space discritized as voxels for fast computation of global description of the sliding window. The other is recognition of a target model by using the sliding window approach to match the model to parts of the scene point cloud. Precomputed 3D integral image enables efficient comparison of the model and pieces of the scene point cloud.

A. 3D Integral Image with BoF

Fig.1 shows an overview of our recognition pipeline. Our approach takes a target model point cloud and a scene point cloud, which contains the target model. We assume that these point clouds are upright. At first, we construct a codebook in order to represent the model with BoF. Local descriptors are computed at interesting points sampled from the scene

(a) (b) (c) (d)

Fig. 2. Screen shots of point clouds of the lecture hall(a,b) and the entrance hall(c,d).

point cloud. The codebook consists of centroids of clusters of local descriptors. We utilize the k-means algorithm to make clusters from local descriptors. After training the codebook, we assign each local descriptor to the nearest codeword and quantize them. Similarly, we detect keypoints on the model, compute local descriptors associated with them, and get BoF representation of the model by quantizing and aggregating these descriptors.

Next, as a preprocessing step for scanning the scene with a sliding window, we divide the scene point cloud into voxels defined by regular grids, which are cubes with side length L. For each voxel we compute a BoF vector from the quantized local descriptors inside it. We define N_x as follows,

$$N_x = ceil\left(\frac{x_{max} - x_{min}}{L}\right), \tag{1}$$

where $ceil()$ is the ceiling function, and x_{max} and x_{min} are the maximum and minimum x coordinates of points in the scene. N_y and N_z are defined for y and z coordinates similarly. Then, we make integral image I on the voxels. Let $V(i,j,k)$ be a set of coded local descriptors included in a voxel corresponding to a cube defined as,

$$\begin{aligned}\{(x,y,z) \in \mathcal{R}^3 \quad | \quad & (i-1)L \le x - x_{min} < iL, \\ & (j-1)L \le y - y_{min} < jL, \\ & (k-1)L \le z - z_{min} < kL\}.\end{aligned}$$

Then, I is calculated as follows,

$$I(i,j,k) = \sum_{i' \le i,\ j' \le j,\ k' \le k} V(i',j',k') \tag{2}$$

Integral image I makes it possible to efficiently compute BoF vector $F(S)$, which represents a feature vector for that piece of the point cloud contained in rectangular S defined by two points (u_1, v_1, w_1) and (u_2, v_2, w_2) as follows,

$$\begin{aligned}F(u_1,v_1,w_1,u_2,v_2,w_2) \quad = \quad & I(u_2,v_2,w_2) - I(u_2,v_2,w_1) \\ & - I(u_2,v_1,w_2) - I(u_1,v_2,w_2) \\ & + I(u_1,v_1,w_2) + I(u_1,v_2,w_1) \\ & + I(u_2,v_1,w_1) - I(u_1,v_1,w_1).\end{aligned}$$

B. Matching Model and Scene with Sliding Window

The size of sliding window is decided according to the extent of the model. Given each voxel has side length of L, we define M_x, M_y and M_z for the model in the same way as N_x, N_y and N_z for the scene. In addition, let M be the larger one of M_x and M_y. Then, we scan integral image I with sliding window of size of $M \times M \times M_z$ and compute the similarity between the model and the piece of the scene point cloud contained in the sliding window. Since we assume both the model and the scene are upright, rotations of each object are assumed to be horizontal ones. Therefore, it is sufficient for description of the sliding window to be invariant to horizontal rotations.

Then, inspired by success of SPM, we propose to incorporate rough spatial information about distribution of local descriptors within sliding window by partitioning the window into several subwindows. If we divide the window in an inappropriate way, a global descriptor constructed from BoFs computed from subwindows loses invariance to horizontal rotations of an object inside the window. For example, if we divide the window depicted in Fig.3c and an object centered in the window rotates, parts of the object contained in each subwindow change. As a result, both BoFs computed from subwindows and a global descriptor of the window change. Therefore, it is important to partition the window in the way which keeps a global description of the window invariant to rotations. Then, we proposed to divide the window by horizontal planes orthogonal to z-axis and upright rectangular tubes placed at the center of the window(see, Fig.3b). We call division by plane horizontal division(HD) and division by rectangular tubes vertical division(VD) below. By partitioning the window in such a way, even if an object centered in the window rotates, parts of the object contained in each subwindow almost unchange. As a result, the global description of the window also almost unchanges. In addition, BoFs of all subwindows can be computed efficiently with the precomputed integral image because volume of each subwindow can be calculated via subtraction of rectangles(Fig.4). Although this is a simple method, we show that it improves recognition performance significantly in our experiments.

We normalize each BoF vector computed from subwindows by their L2 norms and use cosine similarity as a similarity function between global descriptors. Since we expand the sliding window, outer tube-like regions created by VD tend to contain local descriptors irrelevant to the centered object. Therefore, we halve similarity scores calculated from vectors of outer regions. Let $\mathbf{f}_{i1}$ and $\mathbf{f}_{i2}$ be BoF vectors computed from two inner boxes of the window, and $\mathbf{f}_{o1}$ and $\mathbf{f}_{o2}$ be BoF vectors from two outer tube-like regions of the window. Similarly, let $\mathbf{m}_{i1}$,$\mathbf{m}_{i2}$,$\mathbf{m}_{o1}$,and $\mathbf{m}_{o2}$ be BoFs from subwindows defined over the input target model. Then, a similarity score sim

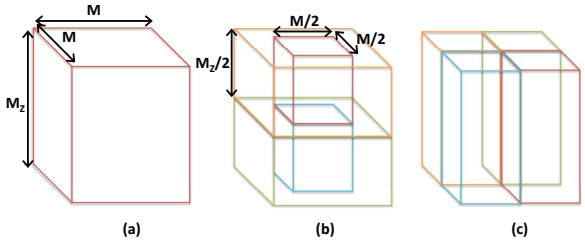


Fig. 3. (a)A sliding window expanded according to the size of the target model. (b)An expanded sliding window with horizontal division(HD) and vertical division(VD). We propose to compute four BoF representations, two from inner boxes and two from outer tube-like regions. (c)An example of partitioning of a sliding window that is not invariant to rotation of an object inside the window.

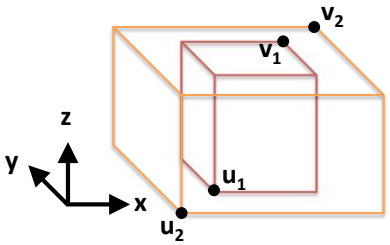


Fig. 4. An example of computation of BoFs for each subwindow. BoF of the inner box is calculated as $F(\mathbf{u}_1, \mathbf{v}_1)$, which is computed with the integral image straightforward. BoF of the outer tube-like region is computed as $F(\mathbf{u}_2, \mathbf{v}_2) - F(\mathbf{u}_1, \mathbf{v}_1)$.

between the model and the sliding window is defined as follows,

$$sim(\mathbf{f}, \mathbf{m}) = \mathbf{f}_{i1} \cdot \mathbf{m}_{i1} + \mathbf{f}_{i2} \cdot \mathbf{m}_{i2} + \frac{1}{2}(\mathbf{f}_{o1} \cdot \mathbf{m}_{o1} + \mathbf{f}_{o2} \cdot \mathbf{m}_{o2}). \quad (3)$$

After applying the sliding window, we takes as object candidates the windows whose similarity score with the model exceeds a threshold and its score is a local maximum relative to scores of neighboring windows. Since it is often the case that candidate detections overlap each other, we eliminate these overlapping boxes by Non Maximal Suppression (NMS) [5]. If the overlap ratio of bounding boxes B1 and B2 exceeds r and similarity score S1 of B1 is greater than S2 of B2, B2 is discarded. We set the value of r to 0.5 in our experiments. Our method regards the remaining object candidates as the final results and outputs them in the form of bounding boxes.

IV. EXPERIMENTS

In order to evaluate our proposed approach, we conduct experiments on real data captured in two indoor environments. In this section, we describe the data and experimental settings at first. Then, we show the results of object recognition.

A. Datasets

In our evaluation experiments, we utilized two point clouds: One is obtained in an entrance hall, the other in a lecture hall. We call the former entrance and the latter lecture hall below. Fig.2 shows screenshots of these point clouds. Since these point clouds are upright, the z axes in those point clouds are orthogonal to the ground. We use these two point clouds as input scenes in our experiments.

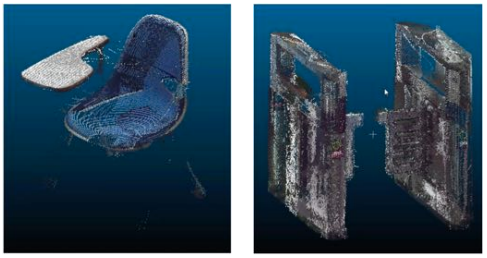


Fig. 5. Input models used in our experiments(left: chair, right: gate). It can be seen that both of the models are damaged by heavy noises.

Both the entrance hall and the lecture hall are made of stitched point clouds captured from different positions with a laser range finder. After composition from several shots, both point clouds are downsampled for postprocessing. We extract the chair and the gate shown in Fig.5 and use them as input models in our experiment.

The size of the lecture hall is 24.1 m $\times$ 16.7 m $\times$ 1.6 m and the number of points is about 71.6 million. This point cloud includes 270 chairs such as shown in the left of Fig.5. We selected one chair from this cloud as the input model in experiments with this cloud. The lecture contains only one room so that its structure is simpler than that of the entrance.

The size of the entrance is 52.4 m $\times$ 43.8 m $\times$ 3.5 m and the number of points is about 50.5 million. This point cloud includes 7 gates as shown in the right of Fig.5. We selected one gate from this cloud as the input model in experiments with this cloud. Although there are fewer points than the lecture hall, the entrance contains several spaces and heavy noise on surfaces so object recognition is harder in the entrance.

B. Experimental Setup

We conducted our experiments in two situations. One takes the chair in Fig.2 as the model and the entrance as the scene. The other takes the gate in Fig.2 as the model and the lecture hall as the scene.

We estimate normal vectors of all points of each input cloud with Moving Least Squares (MLS) [11] and adopt FPFH for local shape description. FPFH expresses a local shape as a 33-dimensional vector based on a distribution of directions of normal vector around a keypoint. To compute FPFH, we use normals within a sphere with a radius of 2 cm centered at a keypoint. We densely sample interesting points at regular intervals of 2 cm on the input cloud instead of using keypoint detectors. We set the regular grid spacing for voxelization at 10 cm for the entrance and 5 cm for the lecture hall.

To investigate the effect of codebook size on recognition performance, we set the number of codewords to 32, 64, 128, and 256. We use implementations of MLS, FPFH, nearest neighbor search via kd-tree in PCL.

We measure performance by precision and recall. We consider bounding boxes that overlap the target object by more 50% as correct bounding boxes. As a baseline method, we used 3D Hough voting with SHOT descriptor implemented in PCL. SHOT descriptors are computed at densely sampled keypoints at the same interval as that in the proposed method.

The number of keypoints extracted from the chair is about two thousand, and from the gate about 30 thousand. We used the method proposed by Lowe [12] in order to get local correspondences, which associates keypoints based on the distance ratio of the two local descriptors nearest to the query.

C. Results

At first, we examined the effect of codebook size when partitioning of sliding window was fixed. Fig.7a and Fig.7b show the precision-recall curves of the detection results for each point cloud with the sliding window with HD and VD. These results indicate that codebook size used for coding pieces of point clouds into BoF does not have significant influence on recognition performance. For object recognition in images, it is known that codebook size and quantization error of local descriptors significantly influences recognition performance [10]. However, in 3D data, most local shapes are planar patches or slightly curved surfaces so the variation in local patches in 3D point clouds is less than that in 2D images. Then, in 3D domain, it is not necessary to use large codebooks.

Next, we examined effectiveness of division of sliding window. In this experiment, we set the size of codebook 128 according to the result of the first experiment and compare performances of a ordinary sliding window without any partitioning(Fig.3a), window only with HD, window only with VD, and window with HD and VD(Fig.3b). Fig.8a and Fig.8b show the experimental results. From these figures, our proposed partitioning method improves recognition performance in both scene point clouds. Especially from the result in the lecture hall, it can be seen that recognition performance is boosted by combining HD and VD.

Table I and II compare our method with the baseline method based on the best F1-measures. This comparison shows that the proposed method works better than the baseline. In addition, the difference between the methods is especially significant in the lecture hall, in which the number of target objects is large and these objects touch each other. Although our method takes into account only horitonzal rotations unlike 3D Hough voting, it does not seem to be problematic in out experiments. In addition, since our method utilizes BoF, which does not depend on exact matches required in Hough voting, we succeeded to recognize objects in large-scale noisy point clouds.

Fig.6 shows examples of true positive detections and false positive detections. From Fig.6b, it can be seen that our method failed detection in the case where target objects spatially concentrated. It is because our global descriptor does not incorporate spatial information of relative positions of object parts in a subwindow. The false positive detection in Fig.6d is considered to be due to planar parts of point clouds. Since most objects contain planar pieces as their parts, planar pieces exist everywhere in scene point clouds. Then if a part of scene point clouds include many planar patches, it yields higher similarity scores rather than the other parts because the chair and the gate also have many planar patches as their parts. To avoid such a case of false detections, it seems to be effective to utilize finer spatial context information such as relative positions of local patches or cooccurence of visual words.

TABLE I. PERFORMANCE INDICES IN LECTURE HALL

	Lecture Hall		
	Precision	Recall	F1-measure
Hough [27]	0.232	0.048	0.080
Proposed	0.806	0.844	0.825

TABLE II. PERFORMANCE INDICES IN ENTRANCE

	Entrance		
	Precision	Recall	F1-measure
Hough [27]	0.182	0.285	0.222
Proposed	1.0	0.857	0.923

V. CONCLUSION

In this work, we proposed a novel approach for 3D object recognition from large-scale indoor point clouds. Our approach utilizes a sliding window with BoF representation of the model, which consists of aggregated local information and is robust to partial noises. We tackled two problems: one is the repetitive appearance of unhelpful primitive shapes (planar patches and curved surfaces), the other is the loss of detailed shape information due to noise such as clutter, occlusions, holes, and measurement error.

Experiments on large-scale indoor point clouds obtained in a lecture hall and an entrance hall showed that object recognition with BoF representation and the sliding window approach performed better than Hough voting method. In addition, it has been shown that our sliding window partitioning method of vertical division and horizon division improves recognition performance significantly. On the other hand, false detections indicate that it is necessary to incorporate finer spatial information such as connectivity of patches or relative positions of parts of objects.

Although we used the same partitioning method in all experiment, optimal division of window may differ in many kinds of object. In future work, we will refine the division of sliding window.

REFERENCES

[1] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze. A Global Hypotheses Verification Method for 3D Object Recognition. In *Proc. of ECCV*, 2012.

[2] A.M. Bronstein, M.M. Bronstein, L. Guibas, and M. Ovsjanikov. Shape Google: Geometric Words and Expressions for Invariant Shape Retrieval. *ACM Transactions on Graphics*, vol. 30, no. 1, pp. 1-20, 2011.

[3] H. Chen and B. Bhanu. 3D free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1252-1262, 2007.

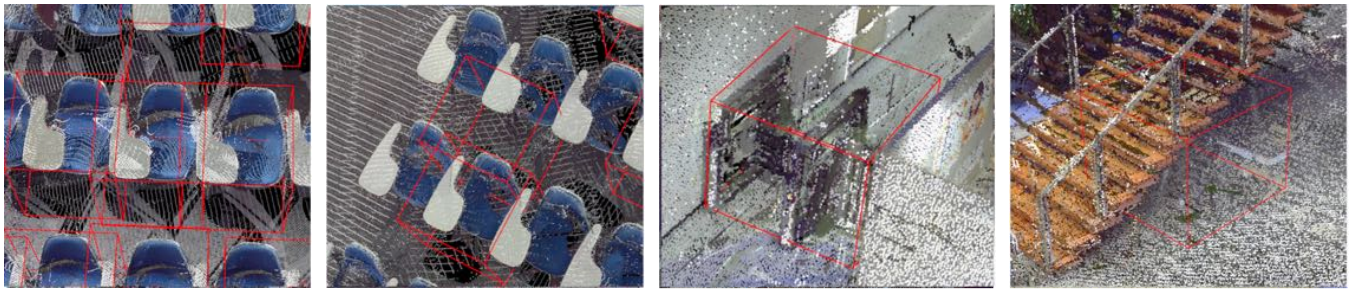
[4] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model Globally, Match Locally: Efficient and Robust 3D Object Recognition. In *Proc. of CVPR*, 2010.

[5] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, 2010.

[6] A. Golovinskiy, V.G. Kim, and T. Funkhouser. Shape-based recognition of 3d point clouds in urban environments. In *Proc. of ICCV*, 2009.

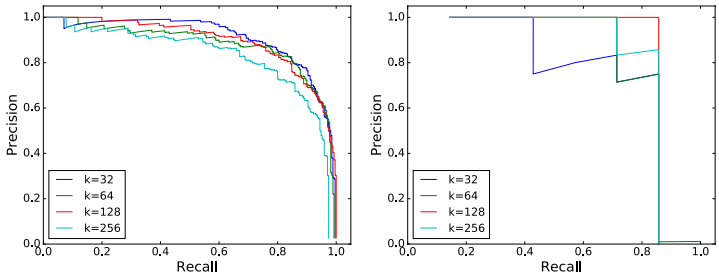
[7] J. Huang and S. You. Detecting Objects in Scene Point Cloud: A Combinatorial Approach. In *Proc. of 3DV*, 2013.

[8] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433-449, 1999.



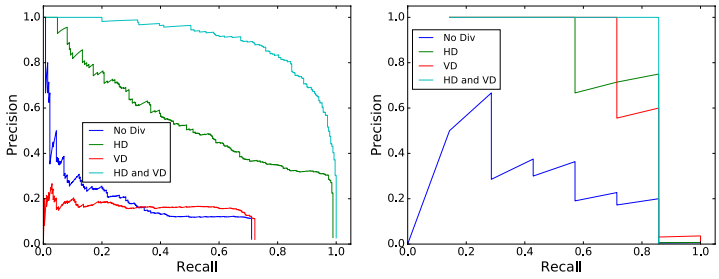
(a) (b) (c) (d)

Fig. 6. True positive detection and false positive detection in the lecture hall(a,b) and the entrance hall(c,d).



(a) Lecture hall (b) Entrance

Fig. 7. Precision-Recall curves when partitioning of sliding window is fixed.



(a) Lecture hall (b) Entrance

Fig. 8. Precision-Recall curves when the codebook size is fixed. "No Div" means the ordinary sliding window without any partitioning.

[9] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L.V. Gool. Hough Transforms and 3D SURF for robust three dimensional classification. In *Proc. of ECCV*, 2010.

[10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognition Natural Scene Categories. In *Proc. of CVPR*, 2006.

[11] D. Levin. Mesh-Independent Surface Interpolation. *Geometric Modeling for Scientific Visualization*, pp. 37-49, 2004.

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[13] T. Malisiewicz, A. Gupta, and A. Efros. Ensemble of Exemplar-SVMs for Object Detection and Beyond. In *Proc. of ICCV*, 2011.

[14] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, vol. 89, no. 2 pp. 348-361, 2010.

[15] G. Pang and U. Neumann. Training-based Object Recognition in Cluttered 3D Point Clouds. In *Proc. of 3DV*, 2013.

[16] C. Papazov and D. Burschka. An Efficient RANSAC for 3D Object Recognition in Noisy and Occluded Scenes. In *Proc. of ACCV*, 2010.

[17] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Proc. of CVPR*, 2013.

[18] A. Patterson, P. Mordohai, and K. Daniilidis. Object Detection from Large-Scale 3D Datasets Using Bottom-Up and Top-Down Descriptors. In *Proc. of ECCV*, 2008.

[19] R.B. Rusu, N. Blodow, Z.-C. Marton, A. Soos, and M. Beetz. Towards 3D Object Maps for Autonomous Household Robots. In *Proc. of IROS*, 2007.

[20] R.B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *Proc of ICRA*, 2009.

[21] R.B. Rusu and S. Cousins. 3D is here: Point Cloud Library(PCL). In *Proc. of ICRA*, 2011.

[22] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, vol. 26, no. 2, pp. 214-226, 2007.

[23] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. of ICCV*, 2003.

[24] S. Song and J. Xiao. Sliding Shapes for 3D Object Detection in Depth Images. In *Proc. of ECCV*, 2014.

[25] R. Toldo, U. Castellani, and A. Fusiello. A bag of words approach for 3d object categorization. In *Proc. of MIRAGE*, 2009.

[26] F. Tombari, S. Salti, and L. Di Stefano. Unique Signatures of Histograms for Local Surface Description. In *Proc. of ECCV*, 2010.

[27] F. Tombari and L. Di Stefano. Object recognition in 3d scenes with occlusions and clutter by hough voting. In *Proc. of PSIVT*, 2010.

[28] P. Viola and M. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, 2004.

[29] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained Linear Coding for Image Classification. In *Proc. of CVPR*, 2010.