2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

3D Object Recognition by Geometric Context and Gaussian-Mixture-Model-Based Plane Classification

Xiangfei Qian and Cang Ye, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a new 3D object recognition method. The method segments a 3D point set into a number of planar patches and extracts the Inter-Plane Relationships (IPRs) for all patches. Based on the IPRs, the method determines the High Level Feature (HLF) for each patch. A Gaussian-Mixture-Model-based plane classifier is then employed to classify each patch into one belonging to a certain model object. Finally, a recursive plane clustering procedure is performed to cluster the classified planes into the model objects. Experimental results demonstrate that the proposed method has high success rates in object recognition with real-world data. Also, the method can be implemented for real-time operation.**

I. Introduction

In order to perform autonomous navigation, a mobile robot needs to perceive its surrounding in 3D and understand the scene through processing the 3D data. 3D LIDARs and stereovision systems have been widely used in the existing research. However, a 3D LIDAR is not suitable for a small-sized robot due to its dimension and weight; while a stereovision system may not produce reliable 3D data for robot navigation due to the need of dense feature points for stereo matching. A commercially available 3D Flash LIDAR Camera (FLC) now has sufficient accuracy for navigating a small robot. An FLC simultaneously produces intensity and range images of the scene at a video frame rate. It has consistent measurement accuracy in its full range (due to the Time-Of-Flight measurement principle) and may return complete depth data for the scene in its field-of-view. The compact size also makes an FLC an ideal perception sensor for a small robot. A more affordable alternative sensor for small robot application is an RGB-D camera, e.g., a Kinect sensor, which has been used in indoor robotic mapping and navigation. Similar to an FLC, an RGB-D camera simultaneously produces depth and color image data. However, the sensor determines depth based on triangulation. As a result its depth measurement accuracy drops quadratically with the true distance, just like a stereovision system. Also, pixel correspondences between the depth and color images of an RGB-D camera depend on each pixel's depth and thus a precise pixel-to-pixel correspondence cannot be achieved.

We are developing a robotic navigation aid, called Smart Cane (SC) [1], for guiding a visually impaired person in indoor environments. To satisfy the portability and accuracy requirements, an FLC (SwissRanger SR4000), instead of a Kinect sensor, is used in the SC for 3D perception. The SC is a computer-vision-enhanced white cane where a forward-looking SR4000 is installed on a conventional white cane for perception. The intensity and the range data of the SR4000 are processed by a computer to achieve two navigational functions—6 Degree-of-Freedom pose estimation and object recognition. For pose estimation, the computer processes the FLC's intensity and range data and determines the device's pose change between each two camera views [2]. This pose change information is then used to compute the device's pose in the world coordinate for way-finding and 3D map-building [3]. For object recognition, the 3D map (point-cloud data) is first segmented into planar surfaces by the NCC-RANSAC method (proposed by the authors earlier in [4]). The extracted planes are then clustered into objects for object recognition/detection. Finally, the detected objects are used for obstacle avoidance [5] or used as waypoints for navigation. The object level information may greatly help a blind traveler move around more effectively in an indoor environment. In this paper, we will consider the recognition of a few typical indoor structures/objects, including doorway, hallway, stairway, parallelepiped, monitor, table, ground and wall.

Object recognition has been a long-lasting research issue in computer vision. Object recognition in 2D image domain has been studied for several decades. The most popular method to identify an object out of an image is to extract the SIFT (Scale Invariant Feature Transform) features [6] from the image and find the match between the SIFT features and that of a model object [7]. Object recognition in 3D data domain [7], [8], [9], [10], [11] is a relatively new and promising field due to the recent advances in 3D depth cameras (FLCs and RGB-D cameras). Ben-Yaacov *et al.* [8] develop a set of 3D rotation invariants based on implicit polynomials for 3D object recognition. However, the method requires high quality segmentation of object, which is not possible in case of noisy range data (such as the SR4000's range data). Johnson and Hebert [9] propose a method based on spin image for shape-based recognition of objects in cluttered scenes. The method first creates 3D models for the target objects and generates the spin image for each model. It then matches a scene spin image to the model spin images. Finally, the spin images matches produce scene-model surface matches and the matched objects in the scene are identified. The method may produce very good results even in cluttered scene. However, it incurs high computational time and is thus unsuitable for our case. Xiong and Huber [10] introduce a Conditional Random Field (CRF) model for classification of the planar patches extracted from a 3D scene. The method defines four types of object models, walls, floors, ceilings and clutter (i.e., unclassified object). Each of them is a single-plane-object. It first extracts from each planar patch

This work was supported in part by the NSF under Award IIS-1017672, the National Institute of Biomedical Imaging and Bioengineering and National Eye Institute of the NIH under Award R01EB018117, and NASA under Award NNX13AD32A. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

X. Qian and C. Ye are with Systems Engineering Department, the University of Arkansas at Little Rock, Little Rock, AR 72204, USA. (e-mail: xxqian@ualr.edu, cxye@ualr.edu).

a number of local features, including the orientation, area and height, and the contextual features (inter-plane relationships), including orthogonal, parallel, adjacent and coplanar. Based on these features, the method constructs a graph *G = (V, E)* by connecting each planar patch with its k-nearest neighbors. Finally, a CRF model taking into account the local features and the contextual features is used to compute, for each object in the scene, the likelihood of being one of the four models. The largest likelihood indicates a match between the object and the corresponding model. Anand *et al.* [11] employ an undirected graph to model a 3D scene. In the graph, a node describes the so-called node features while an edge describes the so-called edge features. The node features include the visual appearance, and local shape and geometry of a segment; and the node features are inter-segment relationships between two segments. They propose a discriminant function to capture the dependencies between the segments of the graph. The discriminant is computed as a weighted sum of the node and edge features given a predicted classification. The objective is to find a classification that maximizes the discriminant value. They use 48 node features and 11 edge features that allow the method to recognize up to 9 different objects in an office environment. The major problem of the method is its high computational cost: in average, it takes ~18 minutes to identify 17 objects from an indoor scene with 50 segments. For this method, the complexity for identifying *m* objects out of a scene with *N* segments is $O(m^N)$, i.e., it involves a NP problem. This is because in order to maximize the discriminant value, the method needs to exhaust all possible classifications, the discriminant computation of which is of polynomial time complexity. Apparently, the method is unsuitable for the SC that requires real-time object recognition capability.

To overcome the NP complexity, we define all Inter-Plane-Relationships (IPRs) that exist in the object models (i.e., target objects) and divide the object recognition problem into three stages. First, we extract the IPRs between each two planar patches and determine if a patch has the IPR(s) with another patch. This procedure classifies the planar patches into 6 categories with exclusive High Level Features (HLF), each of which is a set of IPRs existing in the object models. Second, we classify a patch associated with a particular HLF into a plane of the corresponding object model using the patch's local attributes (area, orientation and height) and the parameter(s) of the IPR(s). This is done by a plane classifier based on Gaussian Mixture Models. Finally, we cluster each type of planar patches into the target objects. In this work, 6 HLFs are defined on top of 9 IPRs for 8 object models. It can be seen that the use of the HLFs discards tremendous number of unwanted combinations of IPRs and thus substantially reduces the computational cost in object recognition.

In this work, an HLF is represented as a vector with a dimension no more than 5. Such a low dimension makes it difficult to attain sufficient classification accuracy if a traditional classification algorithm, such as Support Vector Machines (SVM) [12] is used. 3D object recognition based on SVM generally has feature vector with hundreds of dimensions [13], [14]. Due to the low dimension of feature vector, Gaussian Mixture Model (GMM) [15], [16] is used in this work.

II. Object Recognition by GMMs

The proposed object recognition method is depicted in Fig. 1. It consists of five main procedures: 3D range data acquisition, plane extraction from range data, feature extraction, design and training of GMM Plane Classifier, and plane clustering. Each of them will be described in this section.

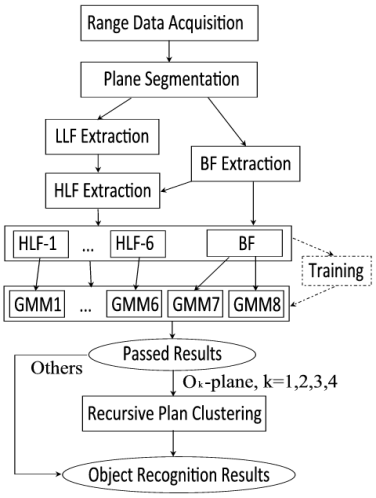

Fig. 1. Diagram of the proposed object recognition method

A. Range Data Acquisition and Plane Segmentation

3D range data are captured from the SR4000 with various camera poses (camera positions and orientations). At each camera pose, the SC's pose estimation system computes the pose of the SR4000. Using this pose information, the 3D range data are registered to form a large view of the scene (in the form of 3D point cloud). The 3D data is then segmented into *N* planar patches, $P_1, P_2, \ldots, P_N$, by the NCC-RANSAC method [4].

B. Features and Feature Vectors

Features are the properties of a planar patch that describe its intrinsic attributes and geometric context (i.e., the geometric arrangement with reference to another planar patch). In the paper, the geometric context refers to Inter-Plane Relationships (IPRs). The features of a patch are described by a feature vector that will be used later to classify the patch into one belonging to a specific type of objects. For each of the *N* patches, $P_1, P_2, \ldots, P_N$, a feature vector construction procedure is performed to assign each of them a feature vector. In this paper, we define three classes of features—Basic Feature (BF), Low Level Feature (LLF), and High Level Feature (HLF).

B.1. Basic Features

BFs are local features that describe a patch's intrinsic attributes. They serve as the identity of the planar patch and play important roles in object recognition. Similar to [10], three BFs—Orientation, Area, and Height (OAH)—are defined for a patch. In this paper, the OAH of the i^{th} patch are computed by the following three equations:

$$O_i = arcos(\mathbf{n_i} \cdot \mathbf{Z}), \quad (1)$$

where $\boldsymbol{n_i}$ is the patch's normal and $Z = (0, 0, 1)$ is the vector of Z-axis;

$$A_i = \sum_{j=1}^{K} \left(\frac{h}{f}\right)^2 d_{ij}^2 = \sum_{j=1}^{K} \gamma d_{ij}^2, \quad (2)$$

where d_{ij} is the distance from the j^{th} point of the patch to the camera, h is the pixel width/pitch of the SR4000, and f is the camera's focal length; and

$$H_i = \max(z_{ij})|_{j=1,\cdots,K}, \quad (3)$$

where z_{ij} is the z coordinate of the j^{th} point and K is the total number of points of the i^{th} patch. Eq. 2 is an approximate method for calculating the i^{th} patch's area. The method uses a pinhole camera model and assumes that an image pixel occupies a square area. While constant γ in Eq. 2 may be determined using the camera's parameter, its actual value does not affect the object recognition result as long as the same value is used in both training and classification phases. For simplicity, we use γ=1.

B.2. Low Level Features

To classify a planar patch into a constituent element of a model object, the patch's Inter-Plane Relationship (IPR) must be considered in addition to its BFs. In this paper, the following 9 IPRs are defined for patch P_i with reference to patch P_j:

plane-distance is a value representing the minimum distance between the points of P_i and the points of P_j.

plane-angle represents the angle between P_i and P_j. It is computed as $\alpha_{ij} = arcos(\mathbf{n_i} \cdot \mathbf{n_j})$.

parallel-distance is the distance between two parallel planes, P_i and P_j. This distance, denoted l_i^j, is computed as the mean of the distance from the centroid of P_i to patch P_j and the distance from the centroid of P_j to patch P_i.

projection-overlap-rate is a value representing to what extent P_i overlaps P_j.

projection-distance is the minimum distance from the points of P_i to patch P_j.

Fig. 2 illustrates how the *projection-overlap-rate* and *projection-distance* for plane P_i with reference to P_j are calculated. For simplicity, we use a 2D illustration. L_j represents the size of patch P_j and L_{ij} stands for the size of the overlap region. The *projection-overlap-rate* of P_i with reference to P_j is L_{ij}/L_j and the *projection-distance* of P_i with reference to P_j is d_{ij}.

is-parallel describes if P_i is parallel to P_j. A value of 1 means that P_i is parallel to P_j; while a value of 0 means they are not parallel. Detection of this relationship within the SR4000's data requires taking into account the camera's noise. For this reason, we let *is-parallel*=1 if $|\alpha_{ij}| < 5^\circ$ and $l_i^j \geq 5$ cm, or *is-parallel*=0 otherwise.

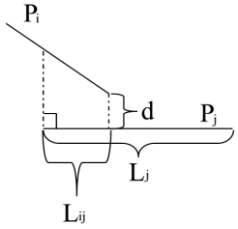

Fig. 2. Definition of *projection-overlap-rate* and *projection-distance*

is-perpendicular describes if P_i is perpendicular to P_j. To account for the sensor's noise, we let *is-perpendicular*=1 if $|\alpha_{ij} - 90^\circ| < 5^\circ$, or *is-perpendicular*=0 *otherwise.*

is-coplanar describes if P_i is co-planar with P_j. *is-coplanar=1* if $|\alpha_{ij}| < 5^\circ$ and $l_i^j < 5$ cm, otherwise *is-coplanar=0.*

is-adjacent describe if P_i is adjacent to P_j. *is-adjacent=1* if the *plane-distance* between P_i and P_j is smaller than 7 cm, otherwise *is-adjacent*=0.

Since there are N planar patches, an $N \times N$ matrix is formed to record each of the 9 IPRs among the N patches. Each matrix is called an LLF, which is denoted by the corresponding IPR in boldface. For instance, ***is-parallel*** stands for the *is-parallel* matrix. In an LLF matrix, an element at (i, j) (i.e., row i and column j) describes the IPR of P_i with reference to P_j. For example, *is-parallel*(i, j)=1 indicates that P_i is parallel to P_j.

B.3. High Level Features and Feature Vectors

The goal of object recognition is to identify one of the eight objects, namely, doorway, hallway, stairway, parallelepiped, monitor, table, ground and wall, from a scene. In this work, each planar patch is classified to a plane belonging to one of the eight objects (models). To this end, we define 6 mutually exclusive High Level Features (HLFs) as shown in Fig. 3. Each HLF represents a set of particular IPRs that exists in a model. The HLF extraction is a process to identify the HLF for each of the N planar patches and assign each patch a HLF vector. The BFs (OAH) extracted earlier from each patch are used in constructing the HFL vector. We first construct a BF vector *[O, A, H]* for each planar patch. Each BF vector is then extended based on the patch's HLF. For a planar patch with HLF-1, HLF-2, HLF-3 or HLF-6, we add parameter d into its BF vector to form a HLF vector [*O, A, H, d*]. For a patch with HLF-5, we add parameter d_1 and d_2 to form a HLF vector [*O, A, H,* d_1, d_2]. For a patch with HLF-4, we simply use the BF vector as the HLF vector. In this work, a plane is treated as an object if it is a wall/ground. Such a plane does not have an IPR. We therefore simply use its BF vector as the HLF vector. A plane with an HLF as depicted in Fig. 3 is called a complex plane, while a wall/ground plane is called an elementary plane. An elementary plane is a standalone one without an IPR.

The HLF detection and HLF vector assignment for the N planar patches are achieved by analyzing the LLF matrices according to the following procedures:

(1) HLF-1 (coplanar with distance *d)*: examine ***is-coplanar*** and find all elements with value "1"; for each element e_{ij}=1, find the corresponding d (Fig. 3a) value from ***plane-distance***; form the HLF-1 vector [*O, A, H, d*] and assign it to the planar patch.

(2) HLF-2 (parallel and face-to-face with distance *d*): examine ***is-parallel*** and find all elements with value "1"; for each element e_{ij}=1, ensure that the two planes are face-to-face by checking their normals; examine ***projection-overlap-rate*** and ensure the elements at (*i, j*) and (*j, i*) are larger than 0.7; let *d=parallel-distance(i, j)*; and form HLF-2 vector [*O, A, H, d*] and assign it to patch P_i.

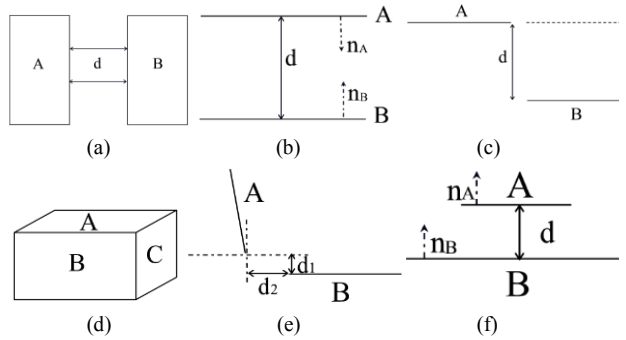

Fig. 3. Definition of HLFs: (a) co-planar with distance *d;* (b) parallel and face-to-face with distance *d;* (c) step-shape with distance *d;* (d) parallelepiped-shape; (e) side-above with distance d_1 and d_2: d_1 is the *A*'s *projection-distance* while d_2 is the minimum distance between *A*'s projection-points on *B* and *B*'s points*;* (f) above-with-distance *d*. For simplicity, we use *A*, *B* and *C* to represent any three patches P_i, P_j and P_k out of the *N* planar patches.

(3) HLF-3 (step-shape with distance *d*): examine ***is-parallel*** and find all elements with value "1"; for each element e_{ij}=1, examine ***projection-overlap-rate*** and ensure the elements at (*i, j*) and (*j, i*) are 0; let *d=parallel-distance(i, j)*; form HLF-3 vector [*O, A, H, d*] and assign it to P_i.

(4) HLF-4 (parallelepiped-shape): examine ***is-perpendicular*** and find all elements with value "1"; for each element e_{ij}=1, ensure *is-adjacent*(*i, j*) =1; perform the same procedure for each of e_{ik} and e_{jk}, and ensure perpendicular and adjacent relations; form HLF-4 vector [*O,A,H*] and assign it to patch P_i.

(5) HLF-5 (side-above with distance d_1 and d_2): examine ***plane-angle*** and find all elements with a value in [*90°, 130°*]; for each of these elements e_{ij}, let d_1*=projection-distance(i, j)* and compute d_2; form HLF-5 vector [*O,A,H,*d_1*,*d_2] and assign it to patch P_i.

(6) HLF-6 (above with distance *d*): examine ***is-parallel*** and find all elements with value "1"; for each element e_{ij}=1, ensure that the two planes face the same direction by checking their normals; ensure *projection-overlap-rate*(*i, j*)=1.0; let *d=parallel-distance(i, j)*; form HLF-6 vector [*O, A, H, d*] and assign it to patch P_i. It is noted that *d=0* means that there is only one plane, B (i.e., A merges with B).

The HLF vector assignment process also generates 6 matrices, $\boldsymbol{Q_1}$ **,…,** $\boldsymbol{Q_6}$, each of which records the *N* planar patches' IPRs for HLF-1, **…**, HLF-6, respectively. The HLF extractor in essence serves as an IPR classifier that identifies if a given planar patch has a geometric context as defined in Fig. 3. After HLF vector assignment, the HLF vectors are sent to the GMM Plane Classifier (GMM-PC) (Fig. 1) for plane classification.

C. GMM Plane Classifier

The GMM-PC consists of 8 GMMs each of which has been trained using data captured from a particular type of objects and is thus able to identify a plane related to that type of object when the relevant HLF vector is present. For instance, a planar patch with a HLF-1 vector is classified as a doorway-plane (i.e., a plane belonging to a doorway) if the parameters of the vector (*O, A, H, d*) are right. This is because the HLF-1 vector causes the doorway-GMM to produce a large response—a likelihood value representing how likely the plane is a doorway-plane.

In this subsection, we present the technical details and training procedure for the GMMs, as well as plane classification by the GMM-PC. For simplicity we call a GMM for detecting object type O_k, for $k = 1, \cdots, 8$, an O_k-GMM. Here, O_1, O_2,…, O_8 represent doorway, hallway, stairway, parallelepiped, monitor, table, ground and wall, respectively. We also call a plane belonging to O_k an O_k-plane and a scene with object O_k an O_k-scene. In the GMM-PC, the O_i-GMM receives all HLF-*i* vectors and identify if the associated patch is an O_i-plane (here, $i = 1, \cdots, 6$). The ground-GMM and the wall-GMM receive the remaining BFs and classify each of the associated planar patches into a ground-plane or a wall-plane.

C.1. Gaussian Mixture Model

A GMM is a probability density function represented as a weighted sum of *M* Gaussian component densities. It is given by:

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} \omega_i g(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), \quad (4)$$

where $\boldsymbol{x}$ is a *D*-dimensional (in our case, *D*=3, 4 or 5) vector with continuous values, ω_i, $i = 1, \ldots, M$, are the mixture weights, and $g(x|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), i = 1, \ldots, M$, are the component Gaussian densities. Each component density is a *D*-variate Gaussian function given by:

$$g(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma_i}|^{1/2}} exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})'\boldsymbol{\Sigma_i^{-1}}(\boldsymbol{x}-\boldsymbol{\mu_i})\right\}, \quad (5)$$

where $\boldsymbol{\mu_i}$ and $\boldsymbol{\Sigma_i}$ are the mean vector and covariance matrix, respectively. The mixture weights satisfy $\sum_{i=1}^{M} \omega_i = 1$. The complete GMM is parameterized by $\boldsymbol{\mu_i}$, $\boldsymbol{\Sigma_i}$ and ω_i for $i = 1, \ldots, M$. We denote these parameters collectively by λ from now on for conciseness.

In this paper, the configuration (*M* and λ) of an O_k-GMM is estimated by training the GMM using a set of HLF vectors obtained from scenes with O_k type of objects. The maximum likelihood estimate of λ is iteratively obtained by the Expectation Maximization (EM) method. The value of *M* is determined by repeating the training process with an increasing *i* and observing the trained GMM's output $p(\boldsymbol{x}|\lambda)$. If the mean of the output difference between an *I*-component GMM and an (*I*+1)-component GMM,

$$\gamma = \frac{\sum_{t=1}^{T}(\sum_{i=1}^{I+1} \omega_i' g(\boldsymbol{x_t}|\boldsymbol{\mu_i'}, \boldsymbol{\Sigma_i'}) - \sum_{i=1}^{I} \omega_i g(\boldsymbol{x_t}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}))}{T}, \quad (11)$$

is below a threshold, we let M=I because having more Gaussian component densities does not change the GMM's probability density. In other words, the GMM with M=I and the λ is sufficient to describe the distribution of the training vectors.

C.2. Training of GMMs

We used a Pan-Tilt Unit (PTU) to acquire point cloud data. The ground truth pose of the PTU allowed us to register the SR4000's range data into a point cloud dataset with a large view of the scene. To train the O_k-GMM we obtained 200 point-cloud datasets from different O_k-scenes in 4 buildings on campus. After plane segmentation and feature extraction, we obtained a set of HLF vectors for each point cloud set. We then train the O_k-GMM using the HLF vectors extracted from all of the point cloud datasets. The training process determined the O_k-GMM's configuration (M and λ). The smallest probability density of the trained O_k-GMM denoted p_{min}^k for the training data is recorded as the threshold for plane classification in a later stage.

C.3. Plane Classification by GMMs

Once trained, each GMM in the GMM-PC is able to classify a planar patch into one of the eight plane types using the patch's HLF vector. To be specific, a planar patch's HLF (or BF if HLF is unavailable) vector is presented to the relevant GMM. The GMM's output, $p_k(\boldsymbol{X}|\lambda)$ for $k = 1, \cdots, 8$ is compared with the early recorded threshold p_{min}^k. If $p_k(\boldsymbol{X}|\lambda) > p_{min}^k$, the planar patch is classified as an O_k-plane. Based on the HLF vectors we define, a planar patch captured from a wall/ground may be classified as both an elementary plane and a complex one. For instance, a path from a ground may be classified as a ground-plane and a stairway-plane. In this case, the complex plane classification overrides the elementary plane classification. All extracted HLF vectors are then presented to the GMM-PC and the corresponding planar patches are classified into the eight types of planes. The GMM-PC results in 8 arrays, G_k for $k = 1, \cdots, 8$, each of which stores the indexes of the planar patches that have been classified as O_k-planes.

D. Recursive Plane Clustering

In this stage, the classified planes, O_k-planes for $k = 1, \cdots, 4$, are recursively clustered into a number of objects, O_k for $k = 1, \cdots, 4$. This means that the doorway-/hallway-/stairway/parallelepiped-planes are grouped into doorway(s) /hallway(s)/stairway(s)/parallelepiped(s) by a Recursive Plane Clustering (RPC) process. A monitor-/table-/ground-/wall-plane is treated as a standalone object and thus no further process is needed. Four independent procedures, RPC-1, …, and RPC-4, process G_1, …, and G_4, respectively and cluster the object planes into the four types of objects. For conciseness, the stairway-plane and parallelepiped-plane clustering (RPC-3 and RPC-4) are briefly described while the doorway-plane and hallway-plane clustering (RPC-1 and RPC-2) are omitted due to their simplicities.

Stairway plane clustering (RPC-3): It is initiated by fetching a random planar-patch-index i from G_3 and pushing it into a First-In-First-Out (FIFO) buffer U. The RPC then: (1) pop an index number from U into j; (2) push j into another FIFO buffer R; (3) for all remaining indexes in G_3, push k into U if $Q_3(j, k)$=1 || (*is-perpendicular(j,k)* =1 && *is-adjacent(j,k)*=1). The RPC-3 procedure calls itself recursively until U is empty. The planar patches indexed by the numbers in buffer R are then clustered as a stairway object. These index numbers are then removed from G_3 and R is cleared for clustering the next stairway.

Parallelepiped plane clustering (RPC-4): RPC-4 is initiated by fetching a random planar-patch-index i from G_4 and pushing it into a FIFO buffer U. The RPC then: (1) pop an index number from U into j; (2) push j into another FIFO buffer R; (3) for all remaining indexes in G_4, push k into U if $Q_4(j, k)$=1. The RPC-4 procedure calls itself recursively until U is empty. The planar patches indexed by the numbers in buffer R are then clustered as a parallelepiped. These index numbers are then removed from G_4 and R is cleared for clustering the next parallelepiped.

III. Experimental Results

We collected 500 datasets from various O_i-scenes ($i = 1, \cdots, 8$) with different camera poses and used these datasets to train the O_i-GMMs ($i = 1, \cdots, 8$). After all eight GMMs were trained, we collected another 60 datasets from each type of scenes, half (type I) from the same scenes used for collecting the training data and the other half (type II) from similar scenes with the same type of objects. We ran the proposed method on the 480 (8×60) datasets and evaluated its performance in term of success rate of object recognition and runtime. The results are tabulated in Table I.

TABLE I. Success Rate in Object Recognition

Target Object	Type I Data	Type II Data	Average
Stairway	0.967	0.900	0.933
parallelepiped	1.000	0.933	0.967
Hallway	1.000	1.000	1.000
Doorway	0.933	0.867	0.900
Table	1.000	0.967	0.983
Monitor	0.900	0.800	0.850
Wall	1.000	1.000	1.000
Ground	1.000	1.000	1.000

It can be observed that the overall success rate (average success rate) in object recognition is over 90% except for monitor. The success rate is much higher than that of the Spin Image method [9] and the CRF method [10]. For each object type, the success rates of type I data are slightly higher than that of type II data because the GMM was trained using the similar type of data. The fact that the success rate of type II data is over 86.7% (except for monitor) indicates that the trained GMMs generalize well. Taking stairway scenario as an example, the method achieves a 96.7% success rate for type I data and a 90% success rate for type II data. In average, its success rate in stairway recognition is 93.3%.

Fig. 4 renders the object recognition results of a stairway-scene, a doorway-scene, a parallelepiped-scene and a monitor-scene in 3D. In all cases, the proposed method identifies the target objects correctly. Taking the stairway-scene as an example, over 50 planar patches are extracted by the NCC-RANSAC method. There exist thousands of IPRs between these patches. However, the proposed method extracts only ~200 HLFs from ~10,000 IPRs. This HLF-extraction step discards enormous unwanted IPRs and thus

saves tremendous amount of computational time. These HLFs are then sent to the stairway-GMM, and the associated planar patches with large GMM responses (above the threshold) are classified as stairway-planes, which are then grouped into a stairway (Fig. 4. A-(b)) by the RPC-3 process.

We implemented the proposed method using Matlab R2012b on a desktop with an Intel i7-2600K CPU, 8 GB memory, and Windows 7 as OS. It took 50 ms – 150 ms to identify each object from the scene, depending on the data size and scene type. The runtime may be drastically reduced if the method is implemented in C code. This means the method may achieve real-time object recognition for the SC application. Based on the reported results, we believe the proposed method outperforms the spin image method [9] and the methods in [10], [11] in term of runtime.

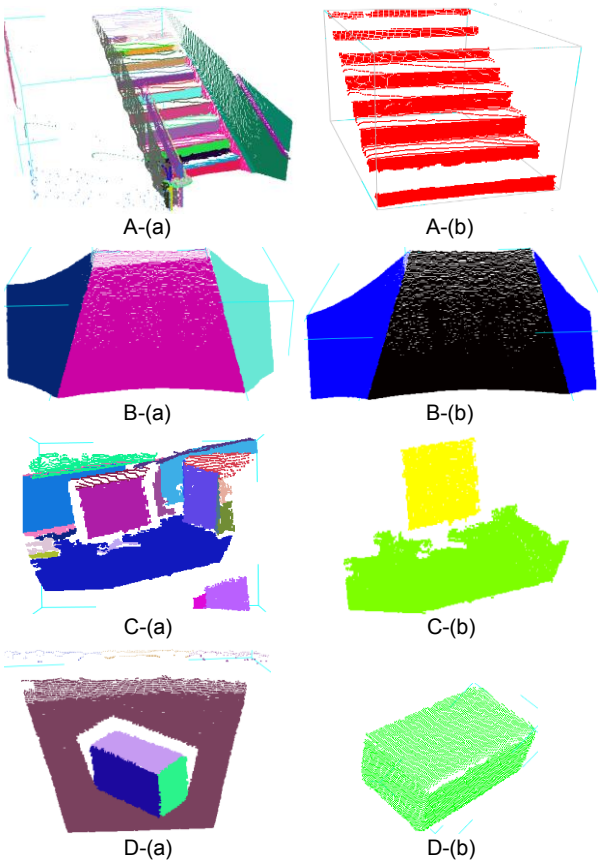

Fig. 4. Experimental results with a stairway-scene (A), a hallway-scene (B), a monitor-scene (C), and a parallelepiped-scene (D): (a) 3D view of the extracted planes; (b) 3D view of the detected objects (stairway: red; ground: black; hallway: blue; monitor: yellow; table: light green; parallelepiped: green)

IV. Conclusion

We have presented a new method for 3D object recognition in indoor environments. The method segments a scene into planar patches and classifies each of them into a plane of a model object. It then clusters the classified planes into the model objects. To achieve real-time plane classification, we define 9 LLFs (9 types of IPRs) and 6 HLFs that exist in the model objects. We then use the HLFs for geometric context classification and the GMM plane classifier for plane classification. This scheme results in a small number of needed IPRs for plane classification and thus substantially save the computational time. Experimental results with real-world data validate the efficacy of the proposed method in term of success rate in object recognition and computational time. The method also has good scalability. Additional HLFs and GMMs may be added to the method to extend its capability without affecting the existing HLF extractors and the GMMs. Although the method is proposed for the SC, it may be employed by a mobile robot for autonomous navigation and object manipulation.

With respect to the future work, we will investigate the possibility of using visual features as additional local attributes for recognition of object whose 3D geometric features are not distinctive enough for identification. For instance, a closed door is difficult to identify if only geometric features are used. In this case, door knob/handle detection using visual feature will help to solve the problem.

References

[1] C. Ye, "Navigating a Portable Robotic Device by a 3D Imaging Sensor," in *Proc. IEEE Sensors Conference*, 2010, pp. 1005-1010.

[2] C. Ye and M Bruch, "A visual odometry method based on the SwissRanger SR-4000," in *Proc. Unmanned Systems Technology XII Conference, SPIE Defense, Security, and Sensing Symposium*, 2010.

[3] A. Tamjidi, C. Ye, and S. Hong, "An Extended Kalman Filter for Pose Estimation of a Portable Navigation Aid for the Visually Impaired," in *Proc. IEEE International Symposium on Robotic and Sensors Environments*, 2013, pp. 178-183.

[4] X. Qian and C. Ye, "NCC-RANSAC: a fast plane extraction method for range data segmentation," in *Proc. IEEE International Conference on Automation Science and Engineering,* 2013, pp. 267-273.

[5] C. Ye and J. Borenstein, "Obstacle Avoidance for the Segway Robotic Mobility Platform," *in Proc. ANS Int. Conf. Robotics and Remote Systems for Hazardous Environments*, 2004, pp. 107-114.

[6] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Computer Vision*, 1999, pp. 1150-1157.

[7] S. Lee, E. Kim and Y. Park, "3D object recognition using multiple features for robotic manipulation," in *Proc. IEEE International Conference on Robotics and Automation*, 2006, pp. 3768-3774.

[8] H. Ben-Yaacov, D. Malah and M. Barzohar, "Recognition of 3D objects based on implicit polynomials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 954-960, 2010.

[9] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 433-449, 1999.

[10] X. Xiong and D. Huber, "Using context to create sematic 3D models of indoor environments," In *Proc. British Machine Vision Conf.*, 2010.

[11] A. Anand, H. S. Koppula, T. Joachims and A. Saxena, "Contextually guided semantic labeling and search for three-dimensional point clouds," *The Int. J. Robotics Research*, vol. 32, no. 1, pp. 19-34, 2012.

[12] C. Cortes and V. Vapnik, "Support-vector network", *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[13] M. Pontil and A. Verri, "Support vector machines for 3D object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 637-646, 1998.

[14] J. Ayoub, B. Granado, O. Romain and Y. Mhanna, "3-D object recognition based on SVM and stereo-vision: application in endoscopic imaging," in *proc. International Conference of Soft Computing and Pattern Recognition*, 2010.

[15] T. Nguyen and Q. Wu. "Gaussian-mixture-model-based spatial neighborhood relationships for pixel labeling problem," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 42, no. 1, pp. 193-202, 2012.

[16] H. Hu, M. Xu and W. Wu. "GMM supervector based SVM with spectral features for speech emotion recognition," In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2007.