Segment and Label Indoor Scene Based on RGB-D for the Visually Impaired

Zhe Wang, Hong Liu, Xiangdong Wang, and Yueliang Qian

Key Laboratory of Intelligent Information Processing &&
Beijing Key Laboratory of Mobile Computing and Pervasive Device
Institute of Computing Technology, Chinese Academy of Sciences
Beijing 100190, China
{wangzhe01,hliu,xdwang,ylqian}@ict.ac.cn

Abstract. The growing study in RGB-D sensor and 3D point cloud have made new progress in obstacle avoidance for the visually impaired. However, it remains a challenging problem due to the difficulty in design a robust and real-time algorithm. In this paper, we focus on scene segmentation and labeling. As man-made indoor scene contains many planar area and structure, plane segmentation and classification is important for further scene analysis. This work propose a multiscale-voxel strategy to reduce the effects of noise and improve plane segmentation. Then the segmentation result is combined with depth data and color data to apply graph-based image segmentation algorithm. After that, a cascaded decision tree is trained to classify different segments into different semantical type. The method is tested on part of the NYU Depth Dataset. Experimental results show that the proposed method combines the advantages of depth data and the geometry characteristics of the scene, and improves scene segmentation and obstacle detection.

Keywords: RGB-D, Plane Segmentation, Scene Segmentation, Obstacle Detection, Blind Navigation.

1 Introduction

In recent years, obstacle avoidance system or Electronic Travel Aids(ETA) [1] have been developed to help the visually impaired to walk safely. ETA takes advantage of modern sensing devices to obtain information of the surrounding environment and the detected information is converted to non-visual signal that can be received and understood by the visually impaired. Among different sensors for ETA, image sensors can provide the most abundant surrounding visual information. Traditionally, stereo vision is used to obtain the depth data of the scene. But it requires large amount of calculation and sensitive to changes in illumination, occlusion, shadow, etc. Coming with the RGB-D sensors, like the Microsoft Kinect, it becomes very easy to obtain both RGB and depth data. Using the Kinect sensors in scene segmentation and obstacle detection becomes a key research topic.

C. Gurrin et al. (Eds.): MMM 2014, Part I, LNCS 8325, pp. 449–460, 2014.

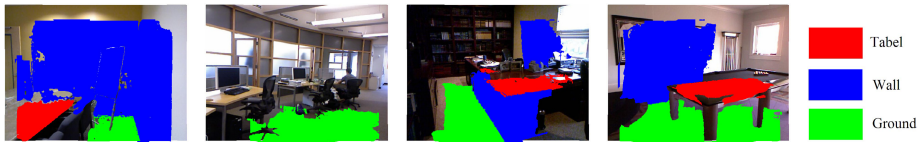

Fig. 1. Output of our system

However, RGB-D image segmentation is not an easy task. The depth data captured by the Kinect sensors is very noisy and the noise is not uniformly distributed. In addition, many image segmentation and object classification algorithms for 2D image processing are not suitable for 3D point cloud. For pointwise classification, it is also very hard to design effective 3D features that is invariant to translation, rotation and scaling. Furthermore, the lack of labeled dataset that is suitable for scene analysis for the visually impaired is another problem. So the classification method should be effective using a small training dataset.

This paper proposes a framework to semantically segment and detect object for indoor scene. Fig. 1 shows the sample output of the system. The original result is generated on point cloud and we project it to 2D image for better presentation. There are two major contributions in our work. Firstly, we propose a multiscale-voxel plane segmentation algorithm which build two different voxel grids on the point cloud. A dense voxel grid is built on the near area of the scene and a sparse voxel grid is built on the far area of the scene. Using multiscale voxel could solve the problem of un-uniformly distributed noise of depth data. Secondly, we extract segment-wise geometry and spatial features to train a cascaded decision tree. The extraction of segment-wise feature is fast and robust in point cloud. And compared with pointwise features, the training on geometry feature and spatial feature requires less data. The experimental results show the proposed algorithms perform well and are applicable in obstacle detection tasks for the visually impaired.

2 Related Work

In recent years, computer vision based ETA have been developed very fast and achieved new progress. Lin et al. [2] develop a wearable stereo vision system composed of an eyeglasses and an embedded processing device to help avoid obstacles in real-time. Zoellner et al. [3] uses the Microsoft Kinect and optical marker tracking to help visually impaired people find their way inside buildings. It provides continuous vibration feedback on the user's waist to give an impression of the environment. And the optical markers can be used to tag points of interest to enable synthesized voice instructions. However, these systems failed to give any semantical description of the scene. So the user could only accept "instruction" and not aware of what the whole scene is like. Tian et al. [4] proposed a computer vision-based wayfinding aid for blind people. Firstly, doors, elevators, and cabinets are detected on their general geometric shape. Then the intra-class

objects are recognized using optical character recognition (OCR) software on extracted text regions. Lee et al. [5] incorporate visual odometry and feature based metric-topological Simultaneous Localization And Mapping (SLAM) into the navigation system. Then a vicinity map based on dense 3D data obtained from RGB-D camera is built to do path planning. These methods only focus on some specific part of the scene. So the user can't receive the information about the whole scene.

Semantical scene analysis could help the visually impaired know better of the surrounding environment. And there have been many development in RGB-D scene analysis method recently. Silberman et al. [6] use depth for bottom-up segmentation and use context features to infer support relationships in the scene. Ren et al. [7] use kernel descriptors on superpixels and use a Markov Random Field (MRF) on superpixel with segmentation tree to model the context of the scene. Choi et al. [8] use 3D geometric phrase model to capture the semantic and geometric relationship between objects which frequently co-occur in the same 3D spatial configuration and then understand the indoor scenes. Gupta et al. [9] propose algorithms for object boundary detection and hierarchical segmentation. Their algorithms visit the segmentation problem afresh from ground-up and develop a gPb like machinery to combine depth information naturally. Wang et al. [10] propose a label propagation method to utilize the existing massive 2D semantic labeled datasets such as ImageNet. Koppula et al. [11] parse the indoor scene with RGB-D data in a mobile robots. A full 3D reconstruction is applied with multiple views of the scene acquired with a Kinect sensor. Then the 3D point cloud is over-segmented and used as underlying structure for a MRF model. These methods focus on the algorithm for general scene segmentation and labeling, while lacking specific analysis for the visually impaired. Wang et al. [12] use hough transform to extract the concurrent parallel lines on the RGB channels and then use depth information to distinguish stairs from pedestrian crosswalks. Then stairs are be recognized as upstairs and downstairs. These methods are mainly focus on the accuracy of scene segmentation while neglect the efficiency of the algorithm which, however, is a key factor in our work. Liu et al. [13] use a graph-based segmentation algorithm which combines the result of plane segmentation and RGB-D data. The method is more focused on the efficiency of the algorithm. However, in order to help the visually impaired to know better of the scene, more semantical analysis, like the type of different structures, should be conducted.

In man-made indoor environment, there exist many planes which contain much structural information. Extracting these planes could be very helpful in scene segmentation. There are many plane segmentation algorithms in literature. One way to extract planes is applying 2D segmentation methods on 3D data. However, this approach performs badly if two planes are very close to each other. In order to take advantage and make full use of 3D data, many new methods have been proposed. Holz et al. [14] compute local surface normal of point clouds using integral images. And then the points are clustered, segmented, and classified in both normal space and spherical coordinates. This method achieves

a frame rate at $30Hz$ at the resolution of 160×120 pixels. Dube et al. [15] use Randomized Hough Transformation to extract planes from depth images. This algorithm could run in real-time on a mobile platform. However it can only detect planes, and cannot segments out the planes. Wang et al. [16] propose a two-step fast plane segmentation algorithm which combines the speed of voxel-wise cluster and the accuracy of pixel-wise process. The algorithm is fast and robust if the 3D data is accurate and don't have much noise. However, due to the limitation of Kinect sensor, the 3D data that is more than 3 meters away from the sensor is inaccurate and very noisy. So, this algorithm could not properly extract planes 3 to 5 meters away from the sensor.

3 Overview of the System

The framework of our whole system is shown in Fig. 2. At first, the input RGB-D data is preprocessed. The position of the RGB camera and the depth camera are not the same, so they should be aligned to the same camera coordinate. The second step is scene segmentation, including plane extraction and graph-based image segmentation. Firstly, the proposed multiscale-voxel algorithm is applied to do plane extraction. Then the result is introduced into the graph-based image segmentation. The third step is segment classification. At first the segment is classified as planes and obstacles. Then planes are classified into different types. Segment classification involves training and testing. In training, the dispersion for each segments is calculated and the multilevel thresholds is determined. And then a cascaded decision tree is trained on geometry and spatial features. In testing, each segment is classified by dispersion into planes or obstacles. Then planes are further classified into different types using the trained cascaded decision tree. Thus, the whole scene is semantically labeled.

4 Scene Segmentation

Scene segmentation is the first step in our system. We propose a fast plane segmentation algorithm based on multiscale voxels. The algorithm is fast and robust to the noise in depth data. After plane segmentation, the results are used as part of the weights in graph-based image segmentation which also combines RGB data and depth data.

4.1 Fast Plane Segmentation with Multiscale Voxels

This paper proposes a novel plane segmentation algorithm to improve the two-step fast plane segmentation algorithm.

The two-step plane segmentation algorithm has two stages [16]. At the first stage, a voxel grid is created on the 3D point cloud data to achieve sparse sampling and noise suppression. For each voxel, the least square method is used to fit a plane and get the plane normal vector. Then region growing algorithm

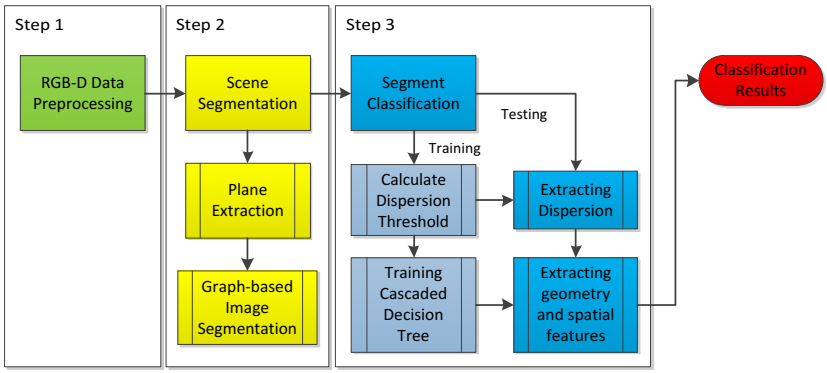

Fig. 2. System architecture

is applied to cluster the voxels according to the normals. At the second stage, accurate judgment is done for each point in un-clustered voxels. After plane segmentation, if the angle between the normal of two different planes is close to zero, these two planes is merge as one.

The size of the voxel grid directly affects the result of plane segmentation. Using an inappropriate side length of the voxel would probably obtain seldom planes. However, as mentioned before, the quality of the 3D point cloud data varies according to the distance. The depth data near the sensor would have less noise and higher resolution than the depth data far away from the sensor. As a result, using one voxel grid with fixed size cannot achieve the best segmentation result. For example, using a set of fine parameters, planes near the sensor can be extracted out well, but further planes may not be well extracted out because they would be considered as non-planar area under this condition.

This paper proposes a multiscale-voxel strategy to solve this problem. Firstly, a dense voxel grid, which has short side, is created on the whole 3D point cloud data. Using this voxel grid, we apply the two-step plane segmentation algorithm and get several planes. Secondly, a sparse voxel grid, which has comparatively long side, is created on the further part of the 3D point cloud. In each voxel grid, the two-step segmentation algorithm would extract different planes. Finally, overlapping areas are merged together.

Formally, a dense voxel grid V_d is created on the whole point cloud C. Then a threshold T is set to separate the point cloud into two parts: the near part C_n and the far part C_f. That is,

$$\forall p_i(x,y,z) \in C, p_i \in \begin{cases} C_n, & p_{iz} < T \\ C_f, & p_{iz} \geq T \end{cases} \tag{1}$$

where $p_i(x, y, z)$ represents the 3D coordinate of a point, p_{i_z} is the depth of point p_i. Then a sparse voxel grid V_s is created on the far part C_f. Apply the two-step plane segmentation algorithm on both V_d and V_s. Then we can get the candidate planes P_d from V_d and P_s from V_s. At last, merge the overlapping or duplicated planes. For each $P_i \in P_d$,$P_j \in P_s$, if P_i and P_j is adjacent and their normal is similar, merge P_i and P_j as P_n and add P_n to the final plane set P. Otherwise, add both P_i and P_j to P.

Applying the proposed multiscale voxel strategy, our algorithm is more robust to depth noise in indoor scenes which range from $0.5m \sim 5m$. As a comparison, the original two-step algorithm could only perform well at the range from $0.5m \sim 3m$. That means the proposed method is more suitable for object detection for the visually impaired because $3m$ is not safe enough if the user of the system moves relatively fast.

4.2 Graph-Based Image Segmentation

After plane segmentation, we apply graph-base image segmentation [17] on the image. The graph-based image segmentation algorithm is a fast unsupervised segmentation method, which is suitable in our task. The approach in Liu [13] is adopt to combine the result of plane segmentation in graph-based image segmentation. The weight for RGB and depth data is defined as

$$\omega(v_i, v_j) = \alpha \cdot \omega_{RGB}(v_i, v_j) + \delta \tag{2}$$

where $\omega_{RGB}(v_i, v_j)$ is the difference of the RGB values between two pixels. And if v_i and v_j belongs to the same plane, α is set a small value in order to reduce the RGB difference in weight. Otherwise, α is set a value that is close to 1 in order to let the RGB difference become a main factor in weight. δ is used to balance the value under different condition so that the weight would not be too small or too big. In this way, the plane segmentation result is semantically combined with RGB and depth data, which can improve the result of image segmentation.

5 Segments Classification

After image segmentation, the scene has been divided into several segments. The next step is to classify these segments into different semantic blocks. In indoor scene, there are many planes which usually come from ground, walls and the surface of tables. These information is very important in understanding the scene. Other non-planar parts can treat as obstacles. So the first step is to divide the segments into planes and obstacles and then planes are classified into different types. In addition, small segments is ignored because they usually are fragments of a large objects and don't have significant affects in scene understanding. So only large segments would be classified and small segments would be merged into adjacent large segments.

5.1 Plane-Obstacle Classification

Although in plane extracting we have already known the extracted segments are planes, after image segmentation these segments would changed and many new segments are generated. So each segments should be classified again. The dispersion of all the points in a segment indicates whether the segment is a plane or not. Planar segments have low degree of dispersion and Non-planar segments have high degree of dispersion. In order to quantify the degree of dispersion, the least square estimation is used to fit a plane for all the points of each segment. It is obvious that if a segment is a plane, the average distance between the points to the plane is small and if a segment is not a plane, the average distance would be large. So, the average distance between the points to the fitting plane is used to represent a segment's dispersion. Formally, we assume that the plane equation of a segment is $Ax + By + Cz + 1 = 0$. Then the least square estimation for A, B, C is

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i z_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i z_i \\ \sum x_i z_i & \sum y_i z_i & \sum z_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum x_i \\ \sum y_i \\ \sum z_i \end{bmatrix} \tag{3}$$

where $(x_i, y_i, z_i), i = 1, 2, 3, ..., N$ represents the coordinates of points in the segment. Secondly, the dispersion of the segment is calculated as

$$D = \frac{\sum_{i=1}^{N} F(p_i, P)}{N} \tag{4}$$

where $F(p_i, P)$ represents the distance of point p_i to plane P.

The dispersion of every segment is calculated. If the dispersion is large, the segment would be an obstacle. Otherwise, the segment is more likely to be a plane. Then a threshold of dispersion is set to divide the segments into plane and obstacle. However, a fixed threshold is not suitable because of noise and imprecise image segmentation. We know that the depth data from Kinect sensors are very noisy and the image segmentation cannot be so precise. Moreover, larger segments tend to have more noise and the larger the segment the more imprecise points it would contain. Considering this, a multilevel threshold strategy is applied. In practice, we conduct statistic analysis on the labeled dataset and determine different thresholds for segments according to their size.

5.2 Plane Classification

In indoor environment, there exists many planes. The ground plane, wall plane, and table are the largest ones and they are the major factors that define the structure of the scene. Therefore, as we have got planes from the former step, they should be classified into different types so that useful information could be provided to the visually impaired. In this paper, we define three types of plane: ground, wall, and table. In addition, the side face of large cabinets are considered

as wall because they impede walking just like walls. And the top face of beds, sofa or small cabinets are considered as table because people would put things on them just like tables.

As the obstacle detection system should run at a high frame rate, the plane classification algorithm should be efficient. In classification, feature extracting usually is the most time consuming step. Therefore, extracting features on pixel scale is not appropriate. On the other hand, ground, wall and table have significant structural differences which reflected in their geometry features and spatial features. It is obvious that the ground plane is horizontal and the wall plane is vertical and the table plane is much higher than the ground. So we extract the angle between the normal and the horizontal plane to represent the geometry feature of that plane.

Before that, we should determine the horizontal plane. We know that in the camera coordinate, the vertical axis may not be vertical in real world because the camera is always moving and rotating. In order to know the real horizontal plane, a two-step horizontal-plane-finding strategy is proposed. Firstly, the angle a_i between the normal $\boldsymbol{n}_i$ of each segment (vector (A, B, C) in Equation 3) and the vertical vector N_v of the camera coordinate (usually one of the axis in the coordinate system) is calculated. And all the normals is divided into approximate vertical and approximate horizontal according to the angle. More specifically,

$$n_i \in \begin{cases} N_v, & a_i < \frac{\pi}{2} \\ N_h, & a_i \geq \frac{\pi}{2} \end{cases} \tag{5}$$

Secondly, the average vertical normal $\boldsymbol{n}_v$ is calculated of the approximate vertical normals set N_v. Then $\boldsymbol{n}_v$ is the normal of real horizontal plane. Next, the angle between the segment normal and the vertical normal $\boldsymbol{n}_v$ is calculated as one of the features for classification. Another feature is the vertical distance between the segment and the lowest point of the scene. It can be easily calculated since we already have the plane equation of each segment and the vertical normal of the scene.

After extracting these features, we train decision trees on the training data. As the geometry feature and spatial feature have semantical meanings, we could use them as priors when training the decision trees. The geometry feature could indicate whether a plane is vertical or horizontal and all vertical planes in the indoor scene are considered as walls as discussed before. So at the first step, a decision tree is trained just on geometry feature to classify vertical plane and horizontal plane. Next step, for all horizontal planes, the spatial feature could indicate whether a plane is ground or table. Therefore, another decision tree is trained on spatial feature to classify ground and table. At last, the two decision trees are cascaded as one classifier. Taking advantage of the semantical meaning of the features, the manually intervened decision tree is more rapid and accurate. And the structure of the decision tree is simple and explainable, which makes the analysis of the training and testing results easily and clearly.

6 Experimental Results

6.1 Dataset

The data used in our experiments come from the NYU Depth Dataset [6]. The NYU Depth Dataset is comprised of video sequences from a variety of indoor scenes recorded by both the RGB and Depth cameras from the Microsoft Kinect. The resolution of RGB images and depth data are both 640×480 pixels. In this paper, 89 pairs of RGB and Depth images are selected. The dataset is manually labeled for ground, wall, table and obstacle.

6.2 Plane Segmentation

We test the proposed multiscale-voxel segmentation algorithm on the whole dataset. In our experiments, a dense voxel grid with side of $20cm$ is created on the area range from $0.5m$ to $3m$ and a sparse voxel grid with side of $30cm$ is created on the area range from $3m$ to $5m$. The segmentation result is compared with the original fixed-size-voxel segmentation algorithm [16], which is also applied on our dataset. Some of the result are shown in Fig. 3. The first row is the original RGB images, the second row is the segmentation result using fixed-size-voxel algorithm and the third row is the segmentation result using the proposed multiscale-voxle algorithm. The difference between the two segmentation algorithm is mark with yellow circles. We can see that in the near area both of the two algorithms perform well in extracting planes. However, in the far area, the fixed-sized-voxel fails to extract the planes. In comparison, the multiscale-voxel algorithm could also extract the planes in far area as well as in the near area. The experimental results indicate that the multiscale strategy is more robust to noise. It is because when the depth of field get larger the depth data get more noisy. This proves the proposed multiscale strategy is suitable for the depth data generated by the Kinect sensors and have better plane extracting results.

6.3 Segments Classification

In order to determine the appropriate thresholds for plane-obstacle classification, we calculate the dispersion for every plane on the training dataset after above segmentation. Then the ground truth type of each segment can be determined by the annotation. For each segment, if it is annotated as a plane, the dispersion is calculated using Equation 4. Then the segments is divided into three different set according to their size (the number of points in a segment). Empirically, we define them as: small segments (size $<$ 5000), medium segments (5000 $\leq$ size $<$ 10000) and big segment (size$\geq$ 10000). For each set, the average $\overline{d}$ of top 10% largest dispersion is calculated. However, due to the imprecision of image segmentation, the average number would be too strict for classification. So we use a compensation coefficient α to adjust it. Then the threshold for a set is $(1+\alpha)\cdot\overline{d}$. In our experiment, α is 0.1. After the thresholds have been determined, we run plane-obstacle classification on the testing dataset. We have 89 labeled images

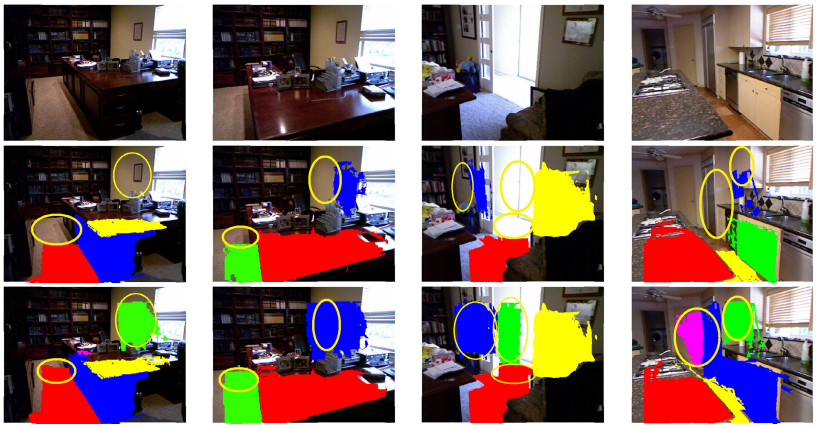

Fig. 3. Plane segmentation results of fixed-size voxels and multiscale voxels. Different planes are randomly colored.

for our test, which consist 262 segments of wall, 112 segments of ground, and 91 segments of table. Half of the segments is used for training, and the others for testing.

After plane-obstacle classification, we test the proposed plane type classification. We conduct three set of experiments. The first is training decision tree only on geometry feature. The second is training decision tree only on spatial feature. And the third is the cascaded decision tree that combines the decision tree trained on geometry feature and the decision tree trained on spatial feature. Fig. 4 shows the confusion matrix for each set of the experiments. From Fig. 4(a) we can see that the recognition rate for wall and ground is very high while all the table have been classified to ground. It's because the normal of wall is horizontal and the normal of ground and table are both vertical. So the geometry feature could not classify ground and table since they have the same geometry feature. Fig. 4(b) shows that most wall and ground are correctly classified but most table have been wrongly classified as wall. It is obvious that tables are much higher above the ground just like the upper part of walls. So table would have the same spatial feature as part of the wall. Therefore it is reasonable that tables are classified to walls. In Fig. 4(c), all the plane types achieve very high recognition rate. This indicates that the cascaded decision tree combines both the advantage of the geometry feature and spatial feature. Theoretically, using plane normal could recognize walls from the other two kind of planes as only walls have horizontal normals. Then tables and ground could be separate by their height above the lowest point of the scene. The experiments proves the combination of the proposed geometry feature and spatial feature are mutually complementary. Then the results of plane type classification and plane-obstacle classification are integrated to analyze the overall classification result. We apply Liu's approach [13] on our dataset and compare the result with our method as Fig. 5 shows. We can see that our method significantly improves the classification result in wall, ground and table. This prove the spatial feature and geometry

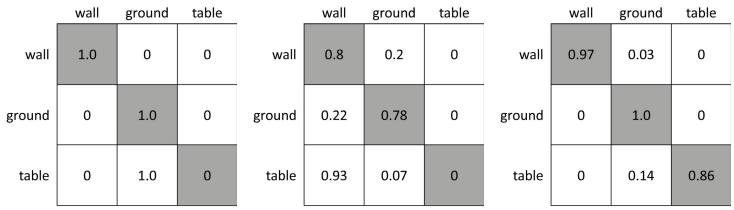

Fig. 4. Confusion Matrix. (a) is the confusion matrix of decision tree trained on geometry feature. (b) is the confusion matrix of decision tree trained on spatial feature (c) is the confusion matrix of the cascaded decision tree.

feature could well describe the character of large planar segments. However, when classifying obstacles, the proposed features performs not well. That's why our method haven't achieve a higher classification rate in obstacles.

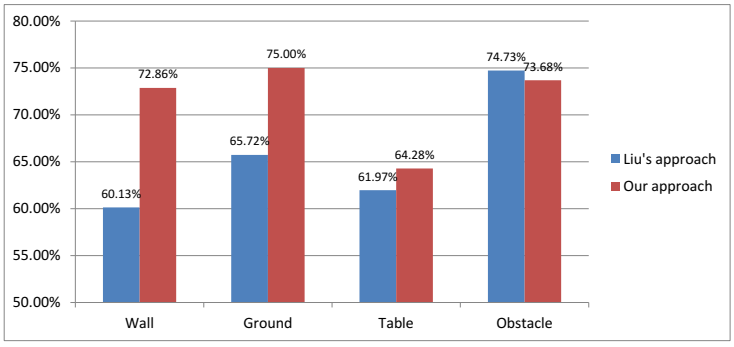

Fig. 5. Comparison between Liu's approach and our approach

7 Conclusions

For indoor scene analysis based on RGB-D for the visually impaired, we propose a multiscale-voxel strategy which improves the accuracy of scene segmentation. Compared with fixed-size-voxel method, our method is more robust to noise and have better performance for the far area of the scene. We also propose a cascaded decision tree based plane classification algorithm to structurally label the indoor scene. Our algorithm extract geometry and spatial feature from the segments after image segmentation. Experimental results show that the proposed method significantly improves the scene structural labeling. The whole system is fast and robust, which meet the requirement of obstacle detection for the visually impaired. In the future, we will apply our method on image sequence or videos and detect more structural area that the visually impaired may care.

Acknowledgments. The research work is supported by the National Nature Science Foundation of China No.60802067 and No.61202209.

References

1. Dakopoulos, D., Bourbakis, N.: Wearable obstacle avoidance electronic travel aids for blind: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews (2010)
2. Lin, K.W., Lau, T.K., Cheuk, C.M., Liu, Y.: A wearable stereo vision system for visually impaired. In: ICMA (2012)
3. Zöllner, M., Huber, S., Jetter, H.-C., Reiterer, H.: NAVI – A proof-of-concept of a mobile navigational aid for visually impaired based on the microsoft kinect. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011, Part IV. LNCS, vol. 6949, pp. 584–587. Springer, Heidelberg (2011)
4. Tian, Y., Yang, X., Yi, C., Arditi, A.: Toward a computer vision-based wayfinding aid for blind persons to access unfamiliar indoor environments. Machine Vision and Applications (2012)
5. Lee, Y.H., Medioni, G.: A rgb-d camera based navigation for the visually impaired. In: RSS 2011 RGB-D: Advanced Reasoning with Depth Camera Workshop (2011)
6. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012)
7. Ren, X., Bo, L., Fox, D.: Rgb-(d) scene labeling: Features and algorithms. In: CVPR (2012)
8. Choi, W., Chao, Y.W., Pantofaru, C., Savarese, S.: Understanding indoor scenes using 3d geometric phrases. In: CVPR (2013)
9. Gupta, S., Arbelez, P., Malik, J.: Perceptual organization and recognition of indoor scenes from rgb-d images. In: CVPR (2013)
10. Wang, Y., Ji, R., Chang, S.F.: Label propagation from imagenet to 3d point clouds. In: CVPR (2013)
11. Koppula, H.S., Anand, A., Joachims, T., Saxena, A.: Semantic labeling of 3d point clouds for indoor scenes. In: Advances in Neural Information Processing Systems, vol. 24 (2011)
12. Wang, S., Tian, Y.: Detecting stairs and pedestrian crosswalks for the blind by rgbd camera. In: BIBM Workshops (2012)
13. Liu, H., Wang, Z., Wang, X., Zhao, G., Qian, Y.: Adaptive scene segmentation and obstacle detection for the blind. Journal of Computer-Aided Design & Computer Graphics (2013)
14. Holz, D., Holzer, S., Rusu, R.B., Behnke, S.: Real-time plane segmentation using RGB-D cameras. In: Röfer, T., Mayer, N.M., Savage, J., Saranlı, U. (eds.) RoboCup 2011. LNCS, vol. 7416, pp. 306–317. Springer, Heidelberg (2012)
15. Dube, D., Zell, A.: Real-time plane extraction from depth images with the randomized hough transform. In: ICCV Workshops (2011)
16. Wang, Z., Liu, H., Qian, Y., Xu, T.: Real-time plane segmentation and obstacle detection of 3D point clouds for indoor scenes. In: Fusiello, A., Murino, V., Cucchiara, R. (eds.) ECCV 2012 Ws/Demos, Part II. LNCS, vol. 7584, pp. 22–31. Springer, Heidelberg (2012)
17. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004)