A new approach of point cloud processing and scene segmentation for guiding the visually impaired

Kailun Yang, Kaiwei Wang*, Ruiqi Cheng*, Xunmin Zhu**

**State Key Labortory of Modern Optical Instrumentation, Zhejiang University, Hangzhou 310027, China*
E-mail: wangkaiwei@zju.edu.cn

Keywords: point cloud, scene segmentation, Kinect, visually impaired

Abstract

Point clouds of 3D scenes are widely applied in guiding the visually impaired by precious research. Many auxiliary systems for the visually impaired are integrated with RGB-D sensors such as Kinect and binocular cameras, which are able to acquire depth pictures and 3D point clouds. Real-time location of objects is adjusted to the world coordinate system through utilization of attitude angle transducers. This paper proposed a novel approach of scene segmentation based on the estimation of normal vectors of a point cloud. Multiplying a point cloud's normal vectors in two directions helps to eliminate correlation in different directions. It is used to split a stereo scene into several surfaces such as ground, walls and slopes. The method is faster and can obtain more separated results than RANSAC algorithm. Besides, three ways to evaluate surface smoothness are compared, including inconsistent degree of normal vectors, variance of depths and difference between normal vectors of two sizes of adjacent regions. Experimental results attained from indoor and outdoor circumstances are presented to validate the approach. It is demonstrated that the proposed method can be efficiently applied into scene segmentation and guiding the visually impaired.

1 Introduction

Vision is the most important means of one's accessing to outside information and knowledge. The visually impaired people's life is limited on many aspects due to loss of normal visual ability. According to World Health Organization, 285 million people are estimated to be visually impaired worldwide and 39 million are blind in 2014 [20].

Traditional auxiliary tool for the blind is simple and crude with an ordinary tactile stick, which is not intelligent and convenient. Modern solutions for aiding blind people are making progressive development with the improvement of ranging technology accompanied by the increased popularity of wearable and small mobile computer devices [8]. Ranging technique with RGB-D sensors such as Microsoft Kinect [10] and binocular cameras is a ubiquitous choice for blind guiding system since it can fit various technical and economic requirements. RGB-D sensors obtain a depth picture and a colour picture which can generate a 3D point cloud. These RGB-D sensors represent a versatile, reliable and cost-effective solution that is rapidly gaining interest within the blind aiding community.

M. Zöllner and S. Huber presented a proof-of-concept of a mobile navigational aid [22]. They shot a validation video which adopted the Microsoft Kinect and optical marker tracking to find ways inside buildings. Kinect is a sensor for the game controller Xbox 360 [18]. Point clouds and depth pictures acquired by RGB-D sensors are widely used in identifying scenes and guiding blind. L. Johnson and C. Higgins adopted a camera belt with two webcams to acquire depth pictures. Depth pictures are then processed to determine how far a connected object is from the cameras [9]. H. Mori and S. Kotani designed Robotic Travel Aid (RoTA) which acts as an intelligent cart for the visually impaired and the disabled. Their guiding cart is integrated with a stereo camera and an image processing system which are able to get path data and obstacle data [14].

S. Cockrell presented an algorithm which uses point cloud data from the Kinect. His algorithm analyses Kinect data to classify areas according to their drivability then it can identify features of the navigation surface in front of a smart wheelchair robot to assist disabled people [6]. N. Molton and S. Se described a portable vision-based obstacle detection system for blind people. Stereo vision with two Sony cameras is adopted in their system. Their algorithm uses depth pictures acquired by binocular camera to estimate a ground region. In this way, obstacle is detected through comparison of seen depth picture with expected standard ground plane. To determine the coefficients of the ground plane every frame, least square fitting or RANSAC algorithm needs to be employed which requires considerable calculation amount [13]. A. Aladrén and G. lópez-Nicolás combined depth information and colour images for floor segmentation. RGB-D data is captured by the Asus Xtion Pro live camera [21], from which they fused range information and colour information to detect obstacle-free paths. Their system is tested to be 99% of precision in real scenarios but the overall system runs at approximately 0.3 frames/s [1]. Performance in processing speed is not ideal because RANSAC algorithm is used for plane parameters estimation. Masahiro Tanaka put forward an algorithm for robust parameter estimation of road condition by Kinect sensor, taking into account of the pitch angle and roll angle of the sensor [19]. Random sample consensus (RANSAC) is an algorithm for fitting a model to experimental data such as fitting a 3D plane to a point cloud [7]. Such algorithms need a certain number of computation solutions to come to a precise result.

Rather than detecting obstacle or estimating ground plane, M. Bellone and A. Messina put forward an approach to differentiate traversable and non-traversable regions of the environment that could enhance safety. A terrain is represented by accurate and dense 3D point clouds generated from Microsoft Kinect and Asus Xtion in their system. They defined an Unevenness Point Descriptor to assess walkability of the surrounding environment [2, 3, 4]. X. Lu and R. Manduchi also used 3D data from commercial stereo systems to detect and precisely localize curbs and stairways for autonomous navigation. Their system runs at about 4Hz on a 1GHz laptop including stereo computation [12].

In place of attaching importance to accurate obstacle detection and ground plane estimation as with previous studies, our approach focuses on scene segmentation. Despite the fact that blind people care about nearby situation most, they also want to know the whole environment situation. This is a key issue not only to detect near obstacles but also to allow the circumstance of surroundings distinguished. An integrated guiding system should have function of notifying the users such as where the walls are, whether there is a slope and the directions of stairs from the position he stands.

Our approach attempts to solve the problem based on the estimation of normal vectors of a point cloud. Though RANSAC algorithm can fulfil the task of scene segmenting, it is unfit for real-time processing. Multiplying a point cloud's normal vectors in two different directions helps to eliminate correlation and segment. For this reason, it is used to split a stereo scene into several surfaces such as ground, walls and slopes. Other than the new approach to segment scenes, three ways to evaluate surface smoothness are compared.

The paper is organized as follows. In Section 2, an overview of the guiding system is presented including the 3D data acquiring method and the device for human computer interaction. In Section 3, the algorithm of scene segmentation is detailed elaborated and the results are shown. In Section 4, three methods to assess surface smoothness are introduced. The processing speed and efficiency of these three methods are compared. Finally, conclusions are drawn and future work is expected in Section 5.

2 System overview

The visually-impaired aiding system structure is shown in Figure1. For each input from the Kinect, process and segmentation include five main steps. The Kinect and JY-901B which contains MPU6050 module are connected to a laptop to input data. A vibrating belt and a voice module are adopted as feedback device.

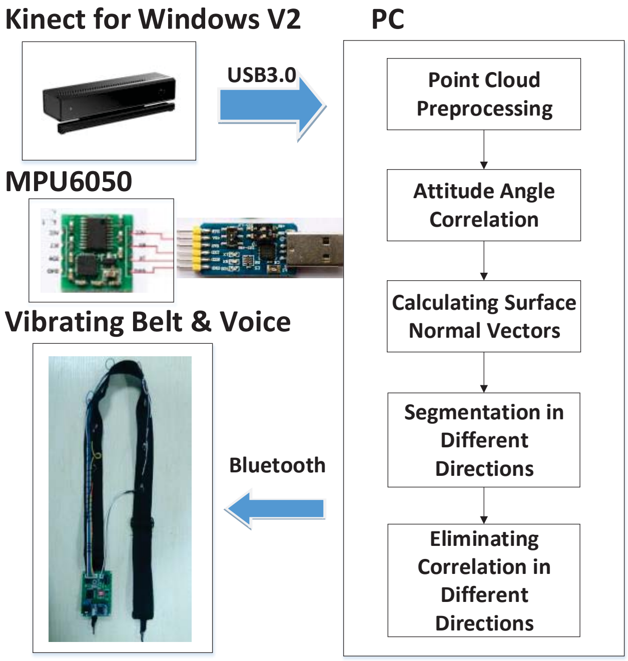

Kinect for Windows V2
PC
USB3.0
MPU6050
Vibrating Belt & Voice
Bluetooth
Point Cloud Preprocessing
Attitude Angle Correlation
Calculating Surface Normal Vectors
Segmentation in Different Directions
Eliminating Correlation in Different Directions

Figure1: The system structure

In our system, the Microsoft Kinect is utilized to obtain depth pictures which generate 3D cloud points. The grey level of a pixel in depth picture represents the distance from the object to the principle plane of the optic centre of the camera. With the function MapDepthFrameToCameraSpace in the Kinect SDK, we can also acquire x and y coordinates other than z coordinates in the depth picture. Set the optic centre of Kinect as origin, the y coordinates represent the perpendicular distance from the object to the origin, and the x coordinates represent the horizontal distance. However, the pitch angle of Kinect or another RGB-D sensor is constantly changing since the visually impaired people are pacing during navigation. A perceiving attitude angle module is equipped to improve the applicability of our system. As shown in Figure 2, the JY-901B attitude angle sensor is mounted on the Kinect, and the kernel module of the attitude angle sensor is MPU6050 which measures angular acceleration in three directions, so the pitch angle and tilt angle of the Kinect is known.

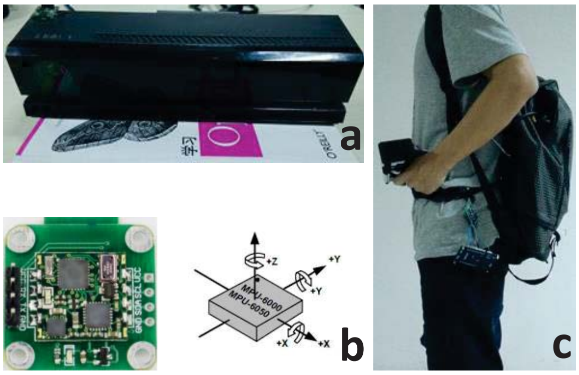


Figure 2: (a) Kinect for Windows V2 and attitude angle sensor JY-901B; (b) The MPU6050 module and its coordinate system; (c) The guiding system and feedback device

The approach based on the estimation of normal vectors of point cloud is detailed described in Section III. In this work, we use C++ programming language in Microsoft Visual Studio 2012 to develop our system. Processing of raw data is performed with help of PCL (Point Cloud Library) [16] and OpenCV [15].

After point cloud processing and scene segmenting, we need to give instant feedback to the users. Our system uses voice prompts and a vibrating belt to feed back to the visually impaired. As shown in Figure 2, the black belt is equipped with seven small vibrators commanded by a laptop in the backpack. The vibrators are working according to different orders for different scene situation. Not only the obstacle information detected but also the scene segmentation result is conveyed through the voice feedback. Vibrating belt is more prompt than voice feedback but voice alerts can provide more explicit information. Since this is not the major part of this paper, we just need to know cooperation of voice and vibration is a valid way of feedback to the blind.

3 Algorithm of scene segmentation

3.1 Point cloud pre-processing

Before implement our algorithm, point cloud pre-processing is significant to ensure qualified data for scene segmentation. Firstly, we down-size the depth picture of the Kinect from 512×424 to 128×106 to save computing time. Despite loss of accuracy, the smaller picture is adequate for our application since the information from the dense depth picture is abundant. Also, blind guiding doesn't require extremely high precision. Noise reduction is required for point cloud processing generally. However, the quality of depth picture from Kinect is pretty high for our application. Experimental results show that the random error of depth measurement ranges from a few millimetres up to about 4cm at the maximum range of the sensor [11]. Therefore, the de-nosing step is skipped in our algorithm to increase the frame rate.

3.2 Adjustment with attitude angle sensor

On the strength of the attitude transducer, x, y and z coordinates can be adjusted to the world coordinate system. Since the camera and the MPU6050 module are assembled together, the MPU6050 coordinate system and camera coordinate system are the same. The MPU6050 module and its coordinate system are shown in Figure 2. Assume a point in the Kinect coordinate system is *(Kx, Ky, Kz)*, and the attitude angle are a, b and c. That is to say, the point *(Kx, Ky, Kz)* rotates around the x-axis by α =a, then rotates y-axis by β =c, and rotates z-axis by γ=b in the end. As Formula (1) shown, multiplying the point coordinate *(Kx, Ky, Kz)* by the rotation matrix can obtain the point *(Wx, Wy, Wz)* in the world coordinate system. Besides, the RGB data are also calibrated with the function MapDepthFrameToColorSpace. In consequence, we can acquire a dense point cloud including x, y, z, RGB after eliminating the influence of angular deflection.

$$\begin{bmatrix} Wx \\ Wy \\ Wz \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} Kx \\ Ky \\ Kz \end{bmatrix} \quad (1)$$

3.3 Estimation of surface normal vectors

The algorithm of scene segmentation is based on the estimation of surface normal vectors of a point cloud. Integral graph and least square method are often used to estimate the normal vector. In our system, the solution to estimate normal vectors is via principal component analysis, which is to establish a covariance matrix from adjacent points of an efficient point. For a point P_i, corresponding covariance matrix C is calculated as Formula (2) shown. K is the number of adjacent points, and $\overline{P}$ is the 3D centroid of adjacent points. λ_i is the i-th eigenvalue, and $\vec{v_J}$ is the j-th eigenvector.

$$C = \frac{1}{k}\sum_{i=1}^{k}(P_i - \overline{P})(P_i - \overline{P})^{\mathrm{T}}, C \bullet \overrightarrow{v_J} = \lambda_i \bullet \overrightarrow{v_i}, j \in \{0,1,2\} \quad (2)$$

In this way, the normal vector $(\overrightarrow{v_0}, \overrightarrow{v_1}, \overrightarrow{v_2})$ of a point P_i is calculated as long as searching for adjacent points is done. In PCL, the normal vectors are calculated through the function computeCovarianceMatrix.

3.4 Segmentation of planes

The main goal of scene segmenting for guiding visually-impaired is to separate a typical indoor or outdoor scene into several planes. The visually-impaired expect the assistant device to be intelligent enough to have him or her known where the walls are and whether there is a slope or stairs. Points on a same plane have a similar normal vector. In this work, Z direction is defined to be parallel to horizontal plane, pointing to the front. Y direction is defined to be perpendicular to water level pointing upwards, and X, Y, Z are in a right handed system.

The procedure of segmentation is shown in Figure 3. Firstly, we use normal vectors in Z direction to differentiate ground and ceilings from other points. In this step, we have to set all the vectors positive in Z direction. Similarly, when processing the point clouds in other directions, it has to be done to eliminate the directional ambiguity. Adaptive threshold method is adopted, which avoids the misjudgement caused by fixed and artificial threshold setting. The self-adaptive threshold is multiplied by a larger factor because ground is considered to be parallel to horizontal plane. Next, normal vectors in X and Y direction are employed to distinguish walls or slope. Multiplying normal vectors in (Z, X) or (Y, Z) directions helps to eliminate correlation in different directions to discriminate walls. Next, remaining points which are not marked are used to detect if there are stairs. In this work, one way to quickly check out if there are stairs is to view the histogram of *y* coordinate in world coordinate system. Several peaks in the histogram will be seen so the number of stairs can be counted to notify the users. Finally, the segmentation result is transferred to the users through voice vibrating belt and voice assembly.

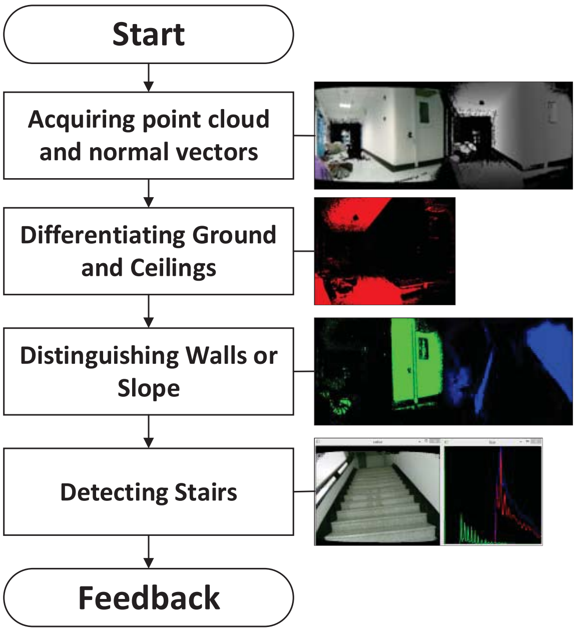
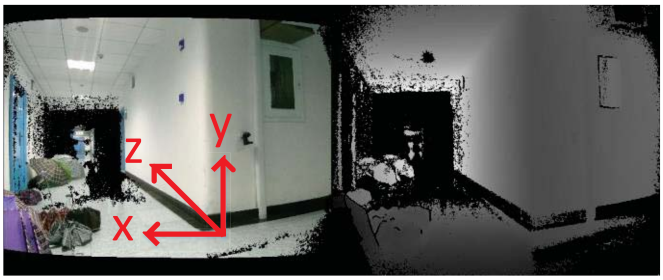

Start
Acquiring point cloud and normal vectors
Differentiating Ground and Ceilings
Distinguishing Walls or Slope
Detecting Stairs
Feedback

Figure 3: The segmentation pipeline

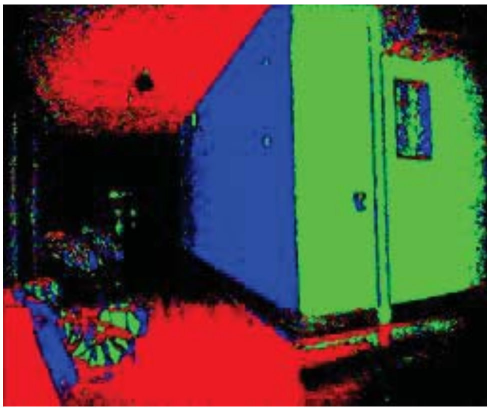
3.5 Segmentation results

Extensive tests were performed to test the algorithm. First, a typical static scene condition without stairs is considered, which is obtained in the laboratory corridor. Afterwards, a scenario with stairs and moving people is analysed.

Figure 4 shows a typical indoor scene containing ground and two walls. The red arrows show the world coordinate system. The left picture in Figure 4(a) is the colour picture mapped from the depth picture on the right. In this experiment, the search radius is 0.03m when searching for adjacent points to calculate normal vectors based the range of the Kinect (0.5-8m) and the ordinary size of wall corner. Shown in the 128×106 pseudo colour picture, the ground and ceilings are marked in red, and the two walls are marked in blue and green. The static indoor scene has been successfully segmented into the three main planes. The connected walls have been correctly segmented into two different planes even though the normal vectors of the two planes are not parallel to coordinate axes.



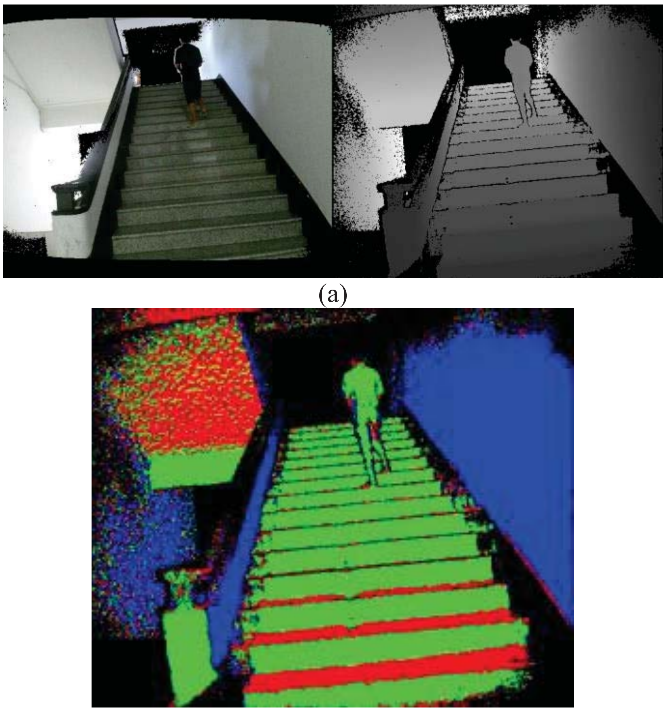
(a)



(b)

Figure 4: (a) Indoor environment depth picture and colour picture; (b) Segmentation result pseudo colorized

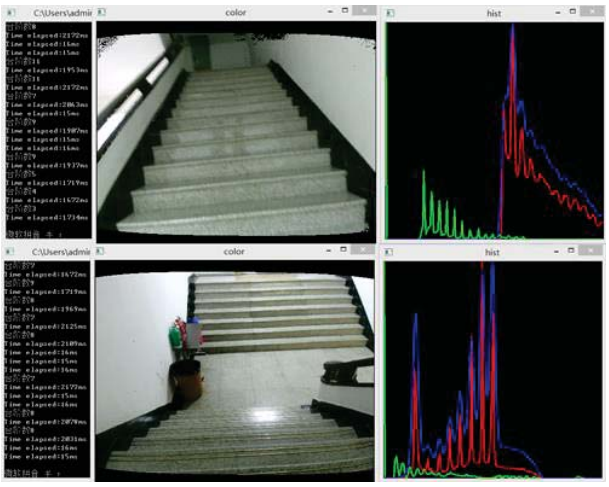
Figure 5 shows an indoor scene with both stairs and moving people. Other than ground and walls segmenting, we use histogram of y in world coordinate system to detect stairs in this work. In the histogram picture, the number of peaks and the space between peaks can be used forecast the step number and width. As shown in Figure 6, the red curve is the y coordinate histogram and the blue curve is the result after Gaussian filtering. This method works well both when ascending and descending the stairs. The whole segmenting program run at approximately 0.6 frames/s on a 2.6GHz laptop. The performance is faster than RANSAC algorithm in estimating normal vectors, but it is not suitable for real-time application yet. With help of CMake [5] and newer versions of the PCLs can take advantage of parallel computation and improve the processing speed greatly.



(a)

(b)

Figure 5: (a) Indoor scene with stairs and moving people; (b) Segmentation result pseudo colorized



```
台阶数8
Time elapsed:2172ms
Time elapsed:16ms
Time elapsed:15ms
台阶数11
Time elapsed:1953ms
台阶数11
Time elapsed:2172ms
台阶数7
Time elapsed:2063ms
Time elapsed:15ms
台阶数9
Time elapsed:1987ms
Time elapsed:15ms
Time elapsed:16ms
台阶数9
Time elapsed:1937ms
台阶数5
Time elapsed:1719ms
台阶数4
Time elapsed:1672ms
台阶数3
Time elapsed:1734ms
```

```
台阶数7
Time elapsed:1672ms
台阶数9
Time elapsed:1719ms
台阶数8
Time elapsed:1969ms
台阶数9
Time elapsed:2125ms
台阶数8
Time elapsed:2109ms
Time elapsed:16ms
Time elapsed:15ms
Time elapsed:16ms
台阶数7
Time elapsed:2172ms
Time elapsed:15ms
Time elapsed:16ms
台阶数8
Time elapsed:2078ms
台阶数8
Time elapsed:2031ms
Time elapsed:16ms
Time elapsed:15ms
```

Figure 6: Stairs detection

4 Surface smoothness assessment

In this part, three methods to assess smoothness are introduced and compared. It can be told by assessing smoothness whether or not the surface is walkable for the visually impaired. Assumed there is a tactile sensor which can express the smoothness of the ground in front of the user, he or she is able to wander freely as if he knows exactly the terrain by touching the device. Thus, safe navigation of the visually impaired can be assured in various environments.

We choose the situation shown in Figure 7 to analyse because it contains both a concrete surface and patches of grass. The first way is using an *Unevenness Point Descriptor (UPD)* in [4]. The descriptor is defined to measure consistent degree of normal vectors as Formula (3) shown. K is the number of adjacent points, and the numerator is sum of surface normal vectors of adjacent points. The bigger the *UPD*, the more consistent of normal vectors. The bigger the *UPD*, the more smooth the surface. *UPD* varies from $\sqrt{2}/2$ to 1. The second way is using variance of y in world coordinate shown in Formula (4), which is the variance of adjacent points' height. The smaller the *VAR*, the more smooth the surface.

$$UPD = \frac{\left|\sum_{i=1}^{k} \overrightarrow{n_i}\right|}{k} \qquad (3)$$

$$VAR = \frac{1}{k}\sum_{i=1}^{k} y_i^2 - \overline{y}^2 \qquad (4)$$

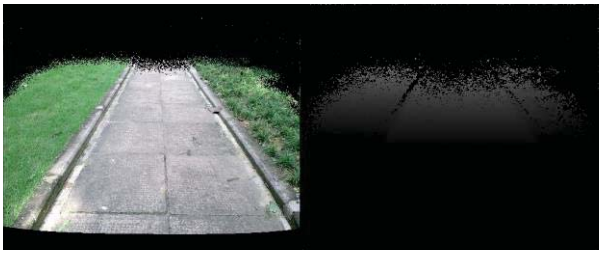
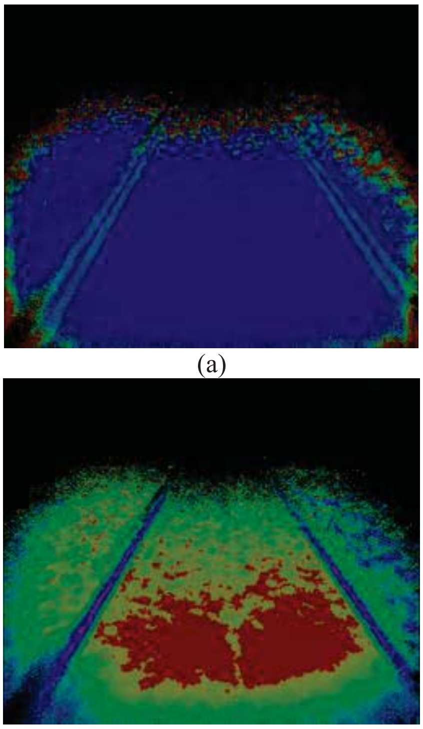


Figure 7: Outdoor environment for surface smoothness assessment

We propose a new way to evaluate the smoothness of point cloud surface. It is defined by *Difference between Normal Vectors (DNV)* of two sizes of adjacent regions. Formula (5) shows the calculation of difference between normal vectors when the search radius are respectively r_1 and r_2. It is easy to understand the normal vectors of different estimation search radius will vary much at the junction of two planes but almost keeps unchanged in a plane. The smaller the *DNV*, the more smooth the surface of adjacent points. *DNV* varies from 0 to 1.

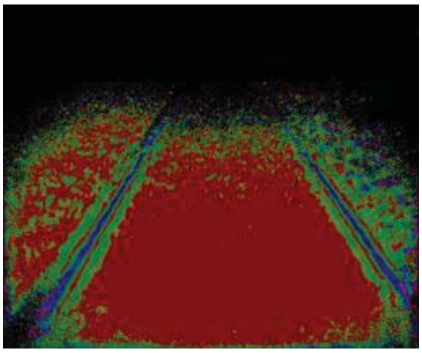
$$DNV(r_1, r_2) = \frac{\left|\vec{n}(r_1) - \vec{n}(r_2)\right|}{2} \qquad (5)$$

The results of surface smoothness assessment by the three ways are shown in Figure 8. For (a), blue represents more smooth and red the contrary and for (b) (c), red represents more smooth and blue the contrary. In UPD and VAR method, search radius is 0.03m. In DNV method, two search radii are respectively 0.03m and 0.1m. Three ways can all be used in assess smoothness of surface, but each one has its superiority and weakness. The UPD and VAR method can measure smoothness from an essential perspective. VAR method is simple but it is sensitive to error points and it is only suitable for one direction analysis, otherwise it should calculate variance of coordinates in other directions or variance of 3D coordinates. UPD method is suitable for surface in different directions but is the slowest method because it has to search for adjacent points and estimate normal vectors for every efficient point. DNV is a way to indirectly measure surface smoothness but it is the fastest method because it just needs a subtraction of normal vectors of two search radii.



(a)

(b)



(c)

Figure 8: Surface smoothness assessment result pseudo colorized in outdoor environment: (a) *UPD* result; (b) *VAR* result; (c) *DNV* result

5 Conclusions and future work

Based on the fact that plenty of obstacle and ground detecting methods for blind guiding are studied by precious research, the approach presented in this paper serves as a supplementary function in navigation. We adopt a commercial RGB-D camera and a low-cost attitude angle sensor, from which we fuse depth pictures and attitude angle of the camera to obtain point clouds in world coordinate system. Making use of point clouds and surface normal vectors estimated, our method can segment ground, ceilings, walls and stairs correctly, which will benefit the visually impaired a lot. Besides, a new way to evaluate smoothness of surfaces is proposed, and three valid methods are compared.

In the future, we aim to incessantly enhance our navigation system for guiding the visually impaired. (1) The implementation of the algorithm is not yet optimized, so we are looking forward to improve the frame rate with program parallel data processing. (2) The approach in outdoor environment is limited because the Kinect will be influenced in direct sunshine. We plan to integrate binocular camera in the system to improve its applicability in various environment. (3) Finally, we intend to make a tactile feedback device which would help to express the terrain situation.

References

[1] A. Aladrén, G. Lopez-Nicolás, L. Puig, et al. "Navigation assistance for the visually impaired using RGB-D sensor with range expansion", *IEEE System Journal*, (2014).

[2] M. Bellone, A. Messina, G. Reina. "A new approach for terrain analysis in mobile robot applications", *Mechatronics (ICM), 2013 IEEE International Conference on*. *IEEE*, pp. 225-230, (2013).

[3] M. Bellone, G. Reina, N. I. Giannoccaro, et al. "3D traversability awareness for rough terrain mobile robots", *Sensor Review*, **vol. 34(2)**, pp. 220-232(13), (2014).

[4] M. Bellone, G. Reina, N. I. Giannoccaro, et al. "Unevenness point descriptor for terrain analysis in mobile robot applications", *International Journal of Advanced Robotic Systems*, **vol. 10(4)**, pp. 261-270, (2013).

[5] CMake. [Online]. http://www.cmake.org/

[6] S. Cockrell, G. Lee. "Using the XBOX Kinect to detect features of the floor surface", *Electronic Thesis or Dissertation*. *Case Western Reserve University*, (2013).

[7] M. A. Fischler, R. C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Readings in Computer Vision*, **vol. 24(6)**, pp. 726-740, (1987).

[8] A. Hub, J. Diepstraten, T. Ertl. "Design and development of an indoor navigation and object identification system for the blind", *Acm Sigaccess Accessibility & Computing*, pp. 47-152, (2003).

[9] L. A. Johnson, C. M. Higgins. "A navigation aid for the blind using tactile-visual sensory substitution", *Engineering in Medicine and Biology Society*, *2006. EMBS '06. 28th Annual International Conference of the IEEE. IEEE*, **vol. 2006**, pp. 6289 – 6292, (2006).

[10] Kinect for Windows. [Online]. http://www.microsoft.com/en-us/kinectforwindows/

[11] K. Khoshelham. "Accuracy analysis of Kinect depth data", *ISPRS – International Archives of the Photogrammetry*, *Remote Sensing and Spatial Information Sciences*, **vol. 3812**, pp. 133-138, (2011).

[12] X. Lu, R. Manduchi. "Detection and localization of curbs and stairways using stereo vision", *IEEE International Conference on Robotics and Automation (ICRA '05*, pp. 4648-4654, (2005).

[13] N. Molton, S. Se, J. M. Brady, et al. "A stereo vision-based aid for the visually impaired", *Image & Vision Computing*, **vol. 16(4)**, pp. 251-263, (1998).

[14] H. Mori, S. Kotani, K. Saneyoshi, et al. "The Matching Fund Project for Practical Use of Robotic Travel Aid for the Visually Impaired", *Advanced Robotics*, **vol. 18(5)**, pp. 453-472, (2004).

[15] OpenCV. [Online]. http://opencv.org/

[16] Point Cloud Library (PCL). [Online]. http://pointclouds.org/

[17] R. B. Rusu, S. Counsins. "3D is here: Point Cloud Library (PCL)", *Robotics and Automation (ICRA)*, *2011 IEEE International Conference*, **vol. 47**, pp. 1-4, (2001).

[18] J. Smisek, M. Jancosek, T. Pajdla. "3D with Kinect", *Advances in Computer Vision & Pattern Recognition*, **vol. 21(5)**, pp. 1154-1160, (2011).

[19] M. Tanaka. "Robust parameter estimation of road condition by Kinect sensor", *SICE Annual Conference (SICE)*, *2012 Proceedings of IEEE*, pp. 197-202, (2012).

[20] World Health Organization. [Online]. http://www.who.int/mediacentre/factsheets/fs282/en/

[21] Xtion PRO. [Online]. http://www.asus.com.cn/Commercial_3D_Sensor/Xtion_PRO/

[21] M. Zöllner, S. Huber, H. Jetter, et al. "NAVI-a proof-of-concept of a mobile navigation aid for visually impaired based on the Microsoft Kinect", Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction, **vol. 4**, pp. 584-587, (2011).