

Research Article

Real-Time Obstacle Detection System in Indoor Environment for the Visually Impaired Using Microsoft Kinect Sensor

Huy-Hieu Pham,^{1,2} Thi-Lan Le,² and Nicolas Vuillerme^{1,3,4}

¹Université Grenoble-Alpes, AGIM/AGEIS, 38706 La Tronche, France

²International Research Institute MICA, HUST-CNRS/UMI-2954-GRENOBLE INP and HUST, 1 Dai Co Viet, Hanoi, Vietnam

³Institut Universitaire de France, 75231 Paris, France

⁴LAI Jean-Raoul Scherrer, University of Geneva, Switzerland and University of Grenoble Alpes, 38041 Saint-Martin-d'Hères, France

Correspondence should be addressed to Huy-Hieu Pham; hieuhuy01@gmail.com
and Nicolas Vuillerme; nicolas.vuillerme@agim.eu

Received 27 July 2015; Revised 22 October 2015; Accepted 25 October 2015

Academic Editor: Hai-Feng Ji

Copyright © 2016 Huy-Hieu Pham et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Any mobility aid for the visually impaired people should be able to accurately detect and warn about nearby obstacles. In this paper, we present a method for support system to detect obstacle in indoor environment based on Kinect sensor and 3D-image processing. Color-Depth data of the scene in front of the user is collected using the Kinect with the support of the standard framework for 3D sensing OpenNI and processed by PCL library to extract accurate 3D information of the obstacles. The experiments have been performed with the dataset in multiple indoor scenarios and in different lighting conditions. Results showed that our system is able to accurately detect the four types of obstacle: walls, doors, stairs, and a residual class that covers loose obstacles on the floor. Precisely, walls and loose obstacles on the floor are detected in practically all cases, whereas doors are detected in 90.69% out of 43 positive image samples. For the step detection, we have correctly detected the upstairs in 97.33% out of 75 positive images while the correct rate of downstairs detection is lower with 89.47% from 38 positive images. Our method further allows the computation of the distance between the user and the obstacles.

1. Introduction

In 2014, the World Health Organization estimated that 285 million people were visually impaired in the world: 39 million are blind and 246 million have low vision [1]. Furthermore, about 90% of the world's visually impaired live in low-income settings and 82% of people living with blindness are aged 50 and above. Generally, these individuals are facing important difficulties with independent mobility that relates to sensing the near-field environment, including obstacles and potential paths in the vicinity, for the purpose of moving through it [2]. The recent advances of computer science now allow the development of innovative solutions to assist visually impaired people. Various types of assistive devices have been developed to provide blind users means of learning or getting to know the environment. A recent literature review of existing electronic aids for visually impaired individuals

has identified more than 140 products, systems, and assistive devices while providing details on 21 commercially available systems [3]. A large number of these systems are based on the Global Position System (GPS) that unfortunately prevent them to be effectively and efficiently employed in indoor environment. Indeed, these systems are not able to provide local information on the obstacles that are encountered due to the inaccurate nature and the susceptibility to loss of the GPS signal. Other types of mobility and navigational aids are based on the sonar to provide information about the surroundings by means of auditory cues [4–6]. They use short pulses of ultrasound to detect objects but there are some disadvantages with this. Different surfaces differ in how well they reflect ultrasound and ultrasonic aids are subject to interference from sources of ultrasound. Finally, another type of assistive devices for blind and visually impaired people has been developed based on the stereo vision technique like [7].

With the advances of computer vision algorithms, intelligent vision systems have received a growing interest. The computer vision-based assistive technology for the visually impaired people has been studied and developed extensively. These systems can improve the mobility of a person who has an impaired vision by reducing risks and avoiding dangers. As imaging techniques advance, such as RGB-D cameras of Microsoft Kinect [8] and ASUS Xtion Pro Live [9], it has become practical to capture RGB sequences as well as depth maps in real time. Depth maps are able to provide additional information of object shape and distance compared to traditional RGB cameras. Some existing systems use RGB-D camera and translate visual images into corresponding sounds through stereo headphones [10, 11]. However, these systems can distract the blind user's hearing sense that could limit their efficient use in daily life.

In this paper, we present a Microsoft Kinect-based method specifically dedicated to the detection of obstacles in indoor environment based on 3D image processing with color-depth information (Figure 5). Precisely, our system was designed to obtain reliable and accurate data from the surrounding environment and to detect and warn about near obstacles such as walls, doors, stairs, and undefined obstacles on the floor with the ultimate goal in order to ultimately assist visually impaired people in their mobility. This paper, indeed, is a part of our long-term research on low vision assistance devices. Inside, the main objective is to design and evaluate a complete prototype of an assistive device which can help the visually impaired in their mobility (Figure 2). To achieve this goal, we rely on various themes explored in literature, including obstacle detection using computer vision, embedded system design, and sensory substitution technology. The main novelty of our work is unifying into a single prototype and this paper is result of image processing module.

2. Related Work

In the last decades, obstacle detection has received a great interest. Interestingly, the majority of the existing systems have been developed for mobile robots [12, 13]. In this section, we will only focus on the works related to assistive technology to help visually impaired people. Wearable systems have been developed based on various technologies such as laser, sonar, or stereo camera vision for environment sensing and using audio or tactile stimuli for user feedback. For instance, Benjamin et al. [14] have developed a laser cane for the blind called C-5 Laser Cane. This device is based on optical triangulation to detect obstacles up to a range of 3.5 m ahead. It requires environment scanning and provides information on one nearest obstacle at a time by means of acoustic feedback. Molton et al. [15] have used a stereo-based system for the detection of the ground plane and the obstacles.

With the RGB-D sensor-based computer vision technologies, the scientists are finding incredible uses for these devices that have already led to advances in the medical field. For instance, Costa et al. [16] used the low-cost RGB-D sensors to reconstruct human body. Other systems based on the RGB-D devices (e.g., Microsoft Kinect or ASUS Xtion Pro) are

able to detect and recognize human activities [17, 18]. Other researchers have developed the methods for detecting falls in the homes of older adults using the Microsoft Kinect. For instance, Mundher and Jiaofei [19] have developed a real-time fall detection system using mobile robot and Kinect sensor. The Kinect sensor is used to introduce a mobile robot system to follow a person and detect when the target person has fallen. This system can also send an SMS message notification and make an emergency call when a fall is detected. Stone and Skubic [20] have also presented a method for detecting falls in the homes of older adults using an environmentally mounted depth-imaging sensor. RGB-D sensor based assistive technology can improve the mobility of blind and visually impaired people to travel independently. Numerous electronic mobility or navigation assistant devices have been developed based on converting RGB-D information into an audible signal or into tactile stimuli for the visually impaired persons. For instance, Khan et al. [21] have developed a real-time human and obstacle detection system for a blind or visually impaired user using a Xtion Pro Live RGB-D sensor. The prototype system includes a Xtion Pro live sensor, a laptop for processing and transducing the data, and a set of headphones for providing feedback to the user. Tang et al. [22] presented an RGB-D sensor based computer vision device to improve the performance of visual prostheses. First, a patch-based method is employed to generate a dense depth map with region-based representations. The patch-based method generates both a surface-based RGB and depth (RGB-D) segmentation instead of just 3D point clouds. Therefore, it carries more meaningful information and it is easier to convey the information to the visually impaired person. Then, they applied a smart sampling method to transduce the important/highlighted information and/or remove background information, before presenting it to visually impaired people. Lee and Medioni [23] have conceived a wearable navigation aid for the visually impaired, which includes an RGB-D camera and a tactile vest interface device. Park and Howard [24] presented a real-time haptic telepresence robotic system for the visually impaired to reach specific objects using an RGB-D sensor. In addition, Tamjidi et al. [25] developed a smart cane with SR4000 3D camera for camera's pose estimation and obstacle detection in an indoor environment. More recently, Yus et al. [26] have proposed a new stair detection and modelling model that provides information about the location, orientation, and the number of steps of the staircase. Aladren et al. [27] have also developed a robust system for visually impaired people based on visual and range information. This system is able to detect and classify the main structural elements of the scene.

3. The Proposed System

3.1. Overview of the Proposed System. Figure 1 illustrates the overall structure of our proposed system. The proposed system uses a personal computer (PC) for processing color-depth images captured from a RGB-D camera. An obstacle detection method aims to define the presence of obstacles and to warn the visually impaired users by using the feedback devices such as auditory, tactile, and vibration.

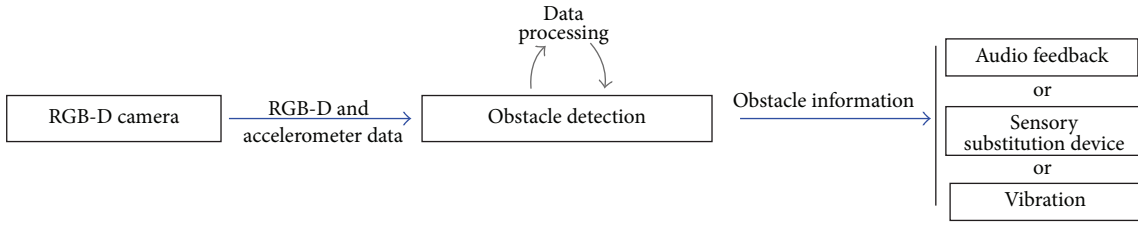


FIGURE 1: Overview of proposed system architecture.

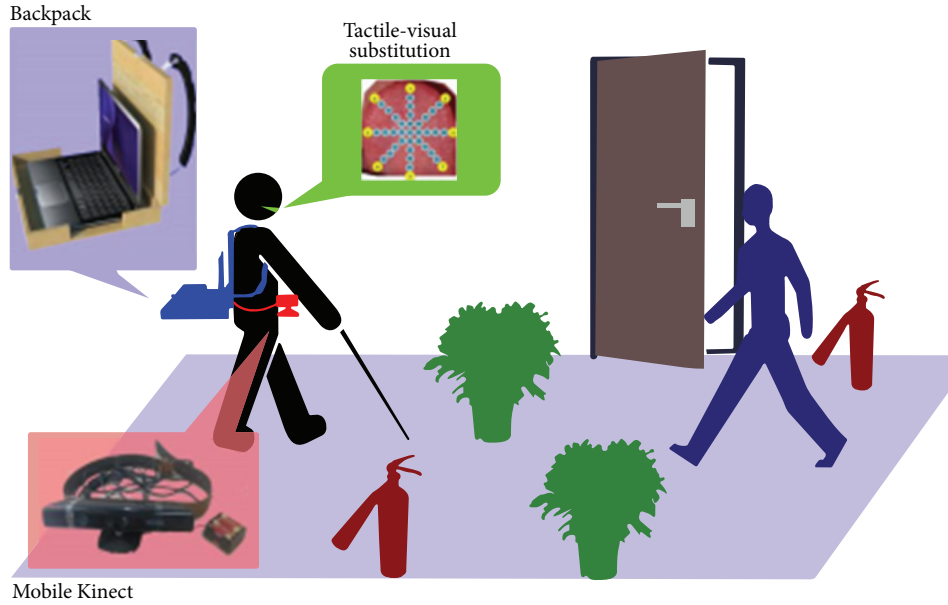


FIGURE 2: Prototype of an obstacle detection and warning for visually impaired people.

In this paper, we focus on obstacle detection using information coming from RGB-D cameras. To warn the visually impaired users, we use a sensory substitution device called Tongue Display Unit (TDU) [32]. In [33], we have presented a complete system for obstacle detection and warning for visually impaired people based on electrode matrix and mobile Kinect. However, this system detects only loose obstacle on the floor. We extend this work by proposing new approach for detecting different types of obstacles for visually impaired people.

3.2. User Requirements Analysis. In order to define the obstacles, we have done a survey with ten blind students in Nguyen Dinh Chieu school in Vietnam. The results of the preliminary study indicated that there are many obstacles in an indoor environment such as moving objects, walls, doors, stairs, pillar, rush bins, and flower pots that blind students have to avoid. In this study, we have defined four frequent types of obstacle that the blind students face in typical indoor environments of their school: (1) doors, (2) stairs, (3) walls, and (4) a residual class that covers loose obstacles (see some examples in Figure 3).

3.3. Obstacles Detection. In this section, we provide the obstacle detection process as illustrated in Figure 4.

The process is divided into five consecutive steps of which the first one is dedicated to the acquisition data. In this step, color-depth and accelerometer data of the scene in front of the user are acquired using a RGB-D camera. These data are then used to reconstruct point cloud in the second step. The third step subsequently filters the obtained point cloud which is fed to the segmentation step. The main goal of the segmentation step is to identify the floor plane. Finally, in the obstacles detection step, we can identify the types of obstacle based on their characteristics.

Step 1 (acquisition data). Various types of RGB-D camera such as Microsoft Kinect and ASUS Xtion Pro can be used in our work. However, in this work, we use Kinect sensor (Figure 5). Kinect is a low-cost 3D camera that is able to work with various hardware models. It is also supported by various framework and drivers. It should be noted, however, that the fundamental part of our system does not need to be changed if we want to use another type of RGB-D cameras in the future.

RGB-D camera captures both RGB images and depth maps at a resolution of 640×480 pixels with 30 frames per second. The effective depth range of the Kinect RGB-D camera is from 0.4 to 3.5 m. The Kinect color stream supports a speed of 30 frames per second (fps) at a resolution of

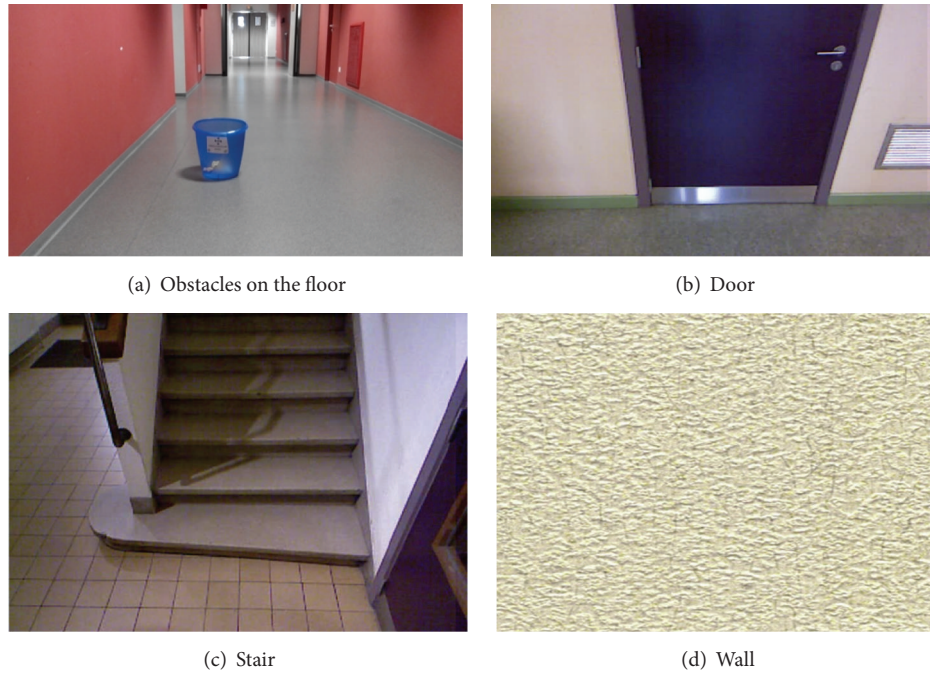


FIGURE 3: Classification of the obstacles.

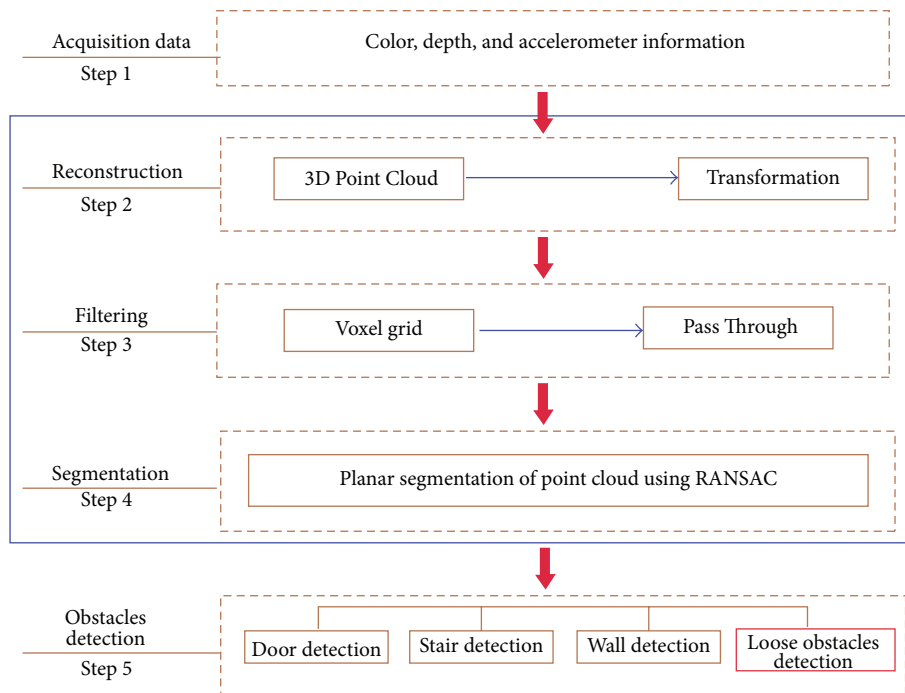


FIGURE 4: Block diagram of obstacle detection process.

640 × 480 pixels [34]. Figure 6 shows an illustration of the viewable range of the Kinect camera.

In this step, in order to capture color and depth information of the scene, we use the standard framework for 3D sensing OpenNI.

Step 2 (reconstruction). Color and depth are combined to create a Point Cloud; it is a data structure used to represent a collection of multidimensional points and is commonly used to represent three-dimensional data. In a 3D Point Cloud, the points usually represent the X , Y , and Z geometric

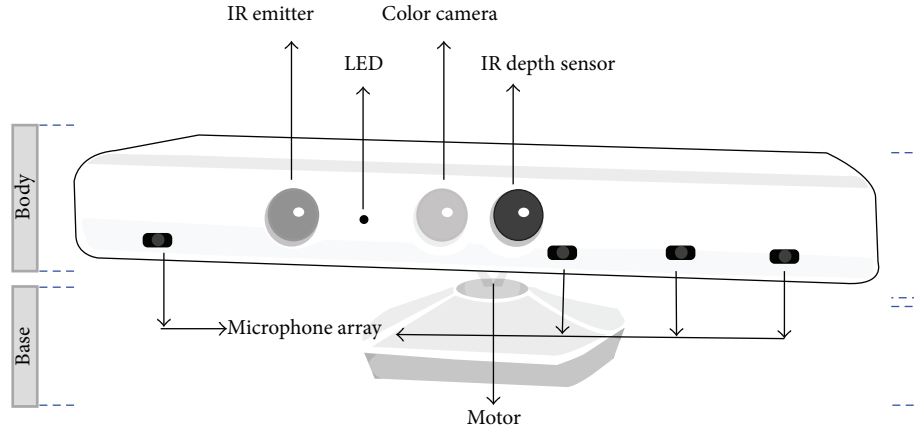


FIGURE 5: The different components of a Kinect sensor.

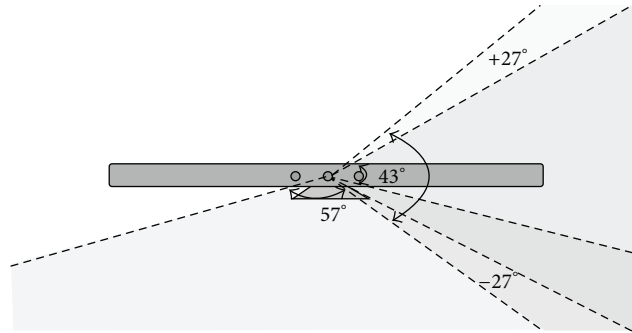


FIGURE 6: The viewable range of the Microsoft Kinect.

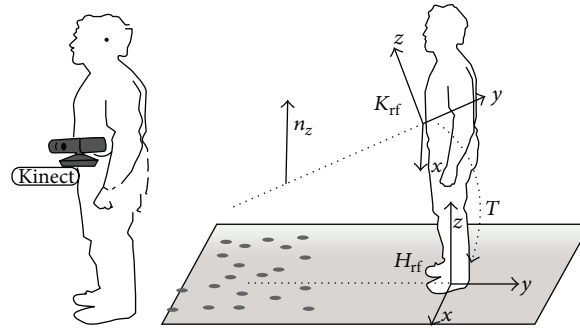


FIGURE 7: Location of the Kinect on the body.

coordinates of an underlying sampled surface. We use the parameters provided by Burrus [35] in order to calibrate color and depth data. Once the Point Cloud is created, it is defined in the reference system of the Kinect, indicated by K_{rf} in Figure 7.

This represents a disadvantage because we have to determine the location of obstacles in the reference system centered at the user's feet, indicated by H_{rf} . Therefore, we apply the transformations geometry including translation, rotation, and reflection to bring Point Cloud from the reference system of the Kinect K_{rf} to H_{rf} . The orientation information computed from accelerometer data and Point Cloud data has been used to perform this task.

Step 3 (filtering). The execution time of the program depends on the number of points in Point Cloud. We thus need to reduce the number of points to ensure that the system can be able to respond fast enough. We will use Voxel Grid Filter to downsample the Point Cloud and, then, Pass Through Filter will remove all points that are located at a position larger than 75 cm in the x -axis (see Figure 8).

Step 4 (segmentation). The next step is Plane Segmentation. Random Sample Consensus (RANSAC) algorithm [36] is used for plane detection in Point Cloud data. RANSAC is an iterative method to estimate the parameters of a model using data that contains outliers. In the present work, RANSAC can

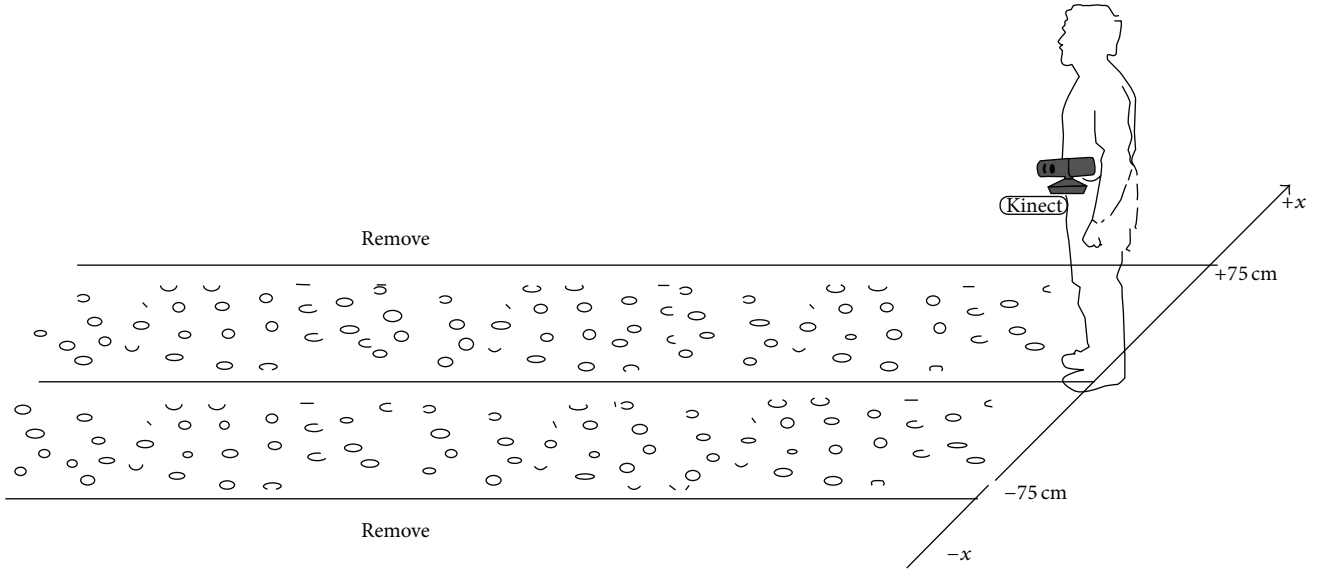


FIGURE 8: Point Cloud after Pass Through.



FIGURE 9: Image of a floor plane.

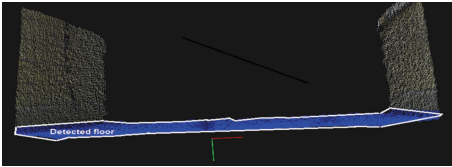


FIGURE 10: Floor detection using RANSAC.

choose a set of points which satisfy the equation of the plane: $ax + by + cz + d = 0$, combined with parallel condition between the floor plane and the xy -plane (see Figure 7). An example of floor image is illustrated in Figure 9 and the detected floor plane is shown in Figure 10.

Step 5 (obstacles detection). Consider the following:

(a) *Obstacles on the Floor Detection.* After performing the floor detection, Euclidean Cluster Extraction is used to determine the clusters on the floor plane. Each cluster is a set of the points in Point Cloud. In a cluster, the distance of each point to the other is smaller than a threshold and it presents for each obstacle [37]. An example of loose obstacle detection is illustrated in Figure 11.

In addition, some classes in PCL library will help us provide the obstacle's size. Each obstacle can be approximated

with a different structure. In our study, each obstacle is approximated by a rectangular parallelepiped (see Figure 12).

(b) *Door Detection.* The detection of door is based on certain conditions on door width as described in [37]. The algorithm can be summarized as follows.

Algorithm 1 (door detection). Consider the following:

- (1) Input is Point Cloud P .
- (2) Estimate the local surface normal n_i of the points p_i belonging to a Point Cloud P .
- (3) Segment the set of planes $V = \{V_1, V_2, \dots, V_N\}$ of potential doors using both the surface normal and the color information.
- (4) Use RANSAC to estimate the parameters of each plane as in the following equation:

$$ax + by + cz + d = 0. \quad (1)$$

Herein, (a, b, c) is the normal n_k of plane V_k .

- (5) Determine the angle between n_k and the normal of the ground plane. This angle should approximate 90 degrees since doors are perpendicular to the floor.
- (6) Determine the dimensions of each plane.
- (7) Check for each plane $V_k \in V$ if its width satisfies the conditions. Remove V_k if this is not the case.

(c) *Staircase Detection.* A staircase consists of at least 3 equally spaced steps as Figure 13.

The authors of Monash University [38] have developed an algorithm to detect the steps of a staircase with the high performance. This algorithm is able to provide the end user with the information as the presence of a staircase (both

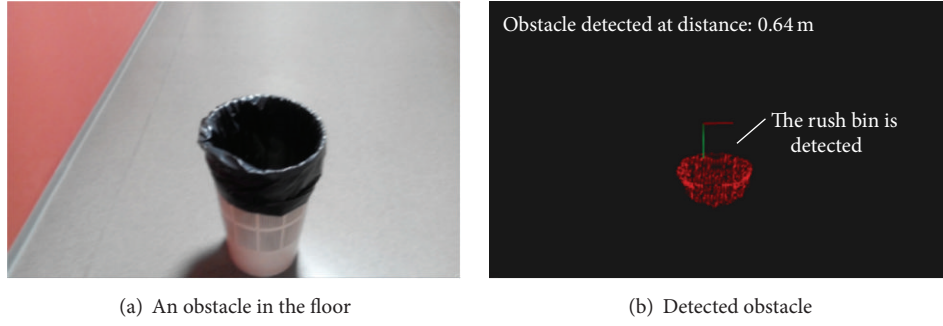


FIGURE 11: The rush bin is detected with the distance between it and the user.

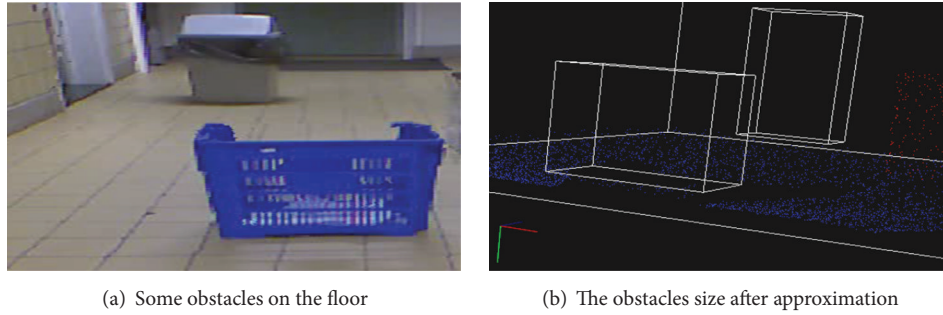


FIGURE 12: An example of the obstacle size approximated by a rectangular parallelepiped.

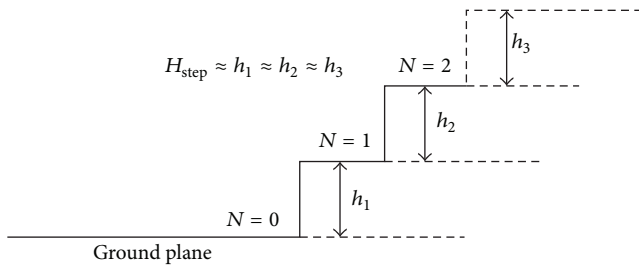


FIGURE 13: Definition of staircase, a staircase has to consist of at least 3 equally spaced steps.

the upstairs and the downstairs) and the number of steps of a staircase in the field of view. The staircase detection algorithm can be summarized as follows.

Algorithm 2. The staircase detection algorithm is as follows:

- (1) Input is Point Cloud P .
- (2) Estimate the local surface normals n_i of the points p_i belonging to a Point Cloud P .
- (3) Filter p_i based on n_i where n_i is parallel to the ground plane.
- (4) Initialize $N = 1$. Determine $P_N = \{p_n\}$, the set of points that are located at height $h = H_N \pm H_{tol}$. Herein, $H_N = H_{step} \times N$ where H_{step} is an estimate of the height of a step and H_{tol} is the tolerance measure.
- (5) Estimate a plane V_N using the set of points P_N in a RANSAC based manner.

- (6) Determine the number of inliers of V_N . If $|V_N| < H_{step}/h$ no step is found and the procedure quits.
- (7) In the other case, increment N by 1 and restart the procedure from step (4).

The value of H_{tol} can be changed. In our program, we can choose $H_{tol} = 1/2 \cdot H_{step}$ to make sure not to miss a step.

(d) **Wall Detection.** With the support of the RANSAC algorithm, we can determine the planes in the scene and calculate the local surface normal of each planes in the Point Cloud. The wall was detected based on the perpendicular characteristic with the floor plane (see Figure 17). In other words, the wall consists of points with a normal that forms an angle of approximately 90 degrees with the z -axis (see Figure 7) and the number of points within a fixed range.

4. Results and Discussion

We have developed the program in C++ using Visual Studio 2010. The implementation extensively makes use of the Point Cloud Library (PCL), an open source for 2D/3D image and Point Cloud processing. PCL supports natively the OpenNI 3D interfaces and can thus acquire and process data from the Microsoft Kinect sensor. Outside the PCL, we also work with some external libraries including Boost, Eigen, OpenCV, FLANN, QHull, and VTK. Our image processing program was tested on a computer, 2.4 GHz. The experiments have been performed in multiple indoor environments with different light conditions.

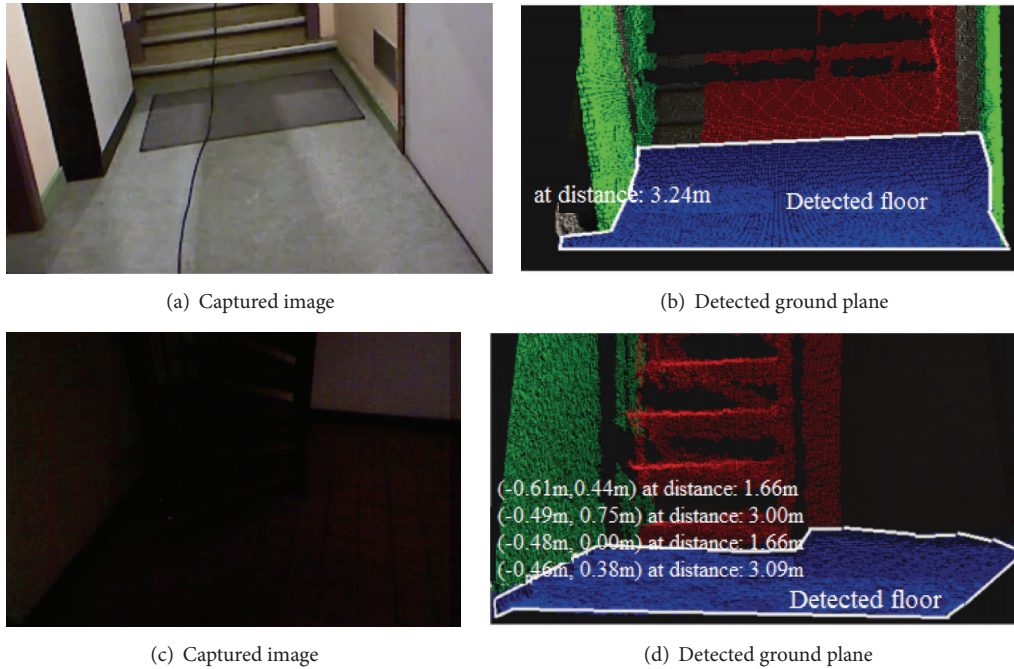


FIGURE 14: Some examples of ground plane detection: (a) color image of the scene; (b) result of ground plane detection.

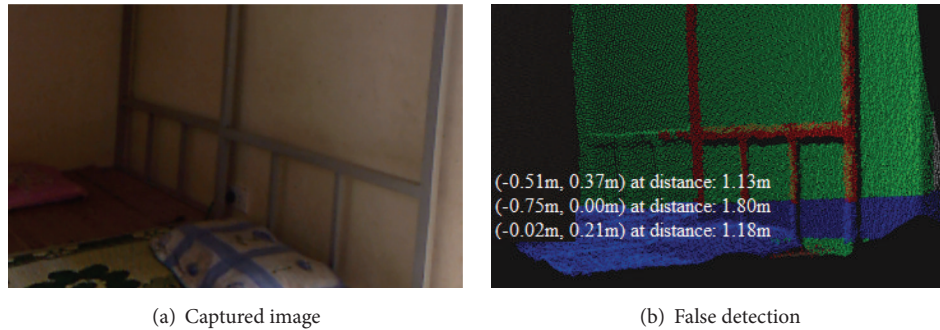


FIGURE 15: A false case of ground detection.

In the obstacle detection process, the floor detection is very important and the performance of our system is based on floor detection. Indeed, in the case that we cannot detect the floor accurately, the system will not be able to detect the obstacles. The program has been tested with a dataset in multiple indoor scenarios. The results showed that the ground plane was detected in most cases in the indoor environment. Figure 14 shows some results of ground plane detection. However, there still exist some situations in which it could fail. For example, the light condition is too strong for Kinect camera.

In another case, when an obstacle consists of a large horizontal plane, in this situation, the horizontal plane of the obstacle could be wrongly identified as the ground plane (see Figure 15).

With regard to the detection of the wall and the loose obstacle on the floor plane (see Figure 16), the result showed

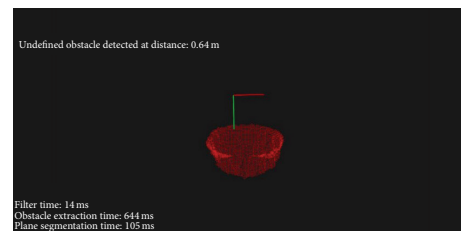


FIGURE 16: An example of loose obstacle detection.

that the wall and the loose obstacle are detected in practically all cases. After measuring the real distance between the user and the obstacles, we further have compared this value with the results obtained by the obstacle detection program and the result showed that the error was negligible (<2%).

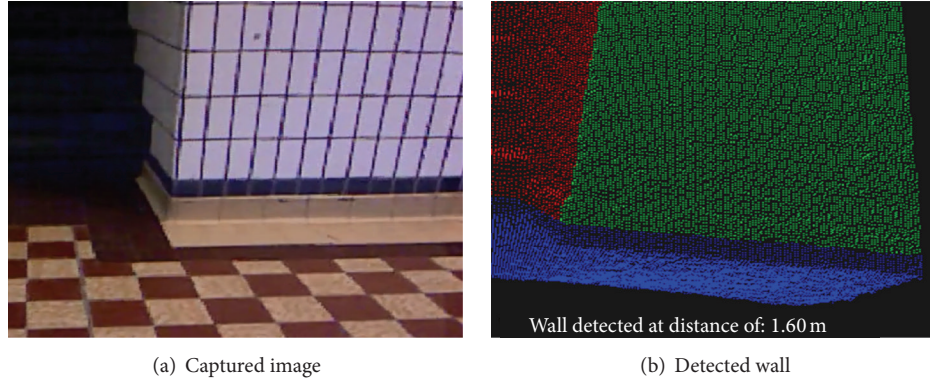


FIGURE 17: Wall detection.

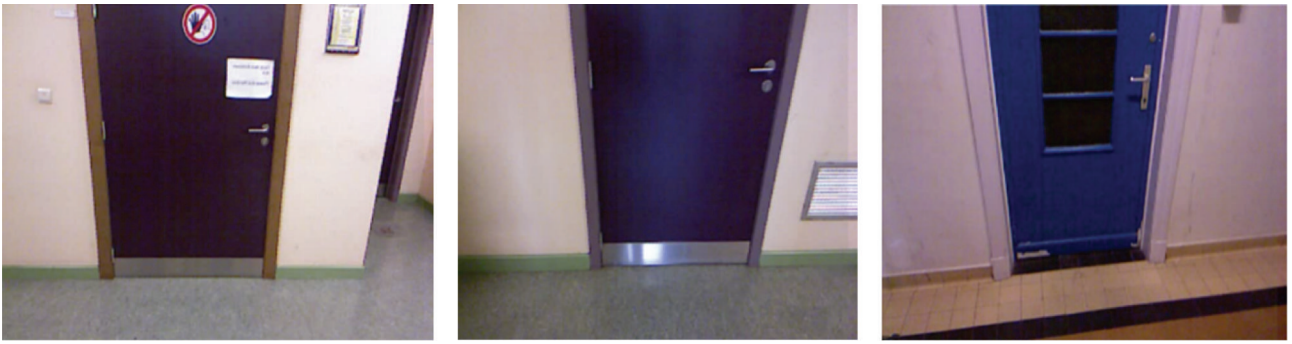


FIGURE 18: The original image of doors.

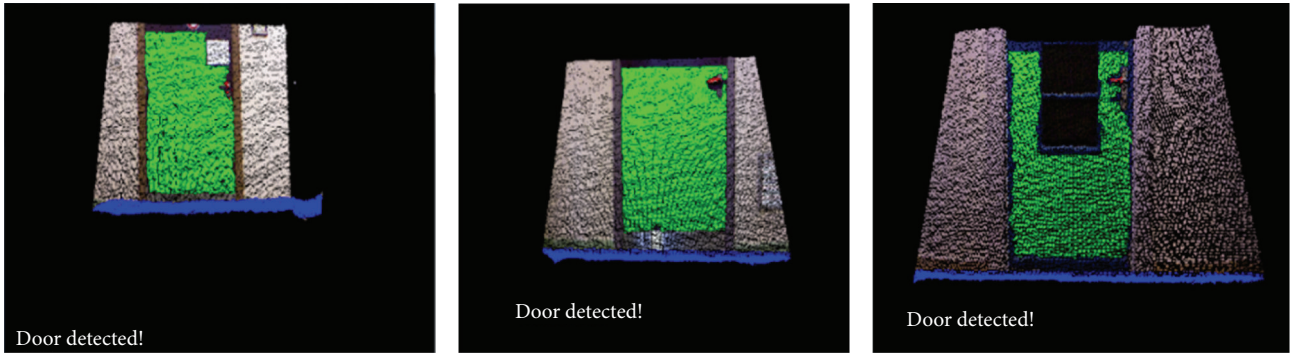


FIGURE 19: Detected doors.

In order to evaluate the performance of the door algorithm, we use the standard measures widely used for classification or identification evaluation, namely, Precision [39]. This is defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (2)$$

where TP and FP denote the number of true positives and false positives, respectively. The doors are detected in 90,69% out of 43 positive images (Figures 18 and 19). The results are summarized in Table 1. We further can compare our results with the performance of some other systems (see Table 2).

With our data, we see that the program also operates well when the camera is approaching the door at an angle

TABLE 1: Performance of the door detection.

Number of images	TP	FP	Precision	Detection time per an image
43	39	4	0.90	0.410 s

of approximately 45 degrees. Figures 20 and 21 show some examples in this case.

In another experiment, we have tested our approach on the dataset with 75 images of upstairs. We also used the standard measures proposed by [39] for evaluating the upstairs algorithm. The results of this experiment are presented in



FIGURE 20: Some examples of doors at an angle.

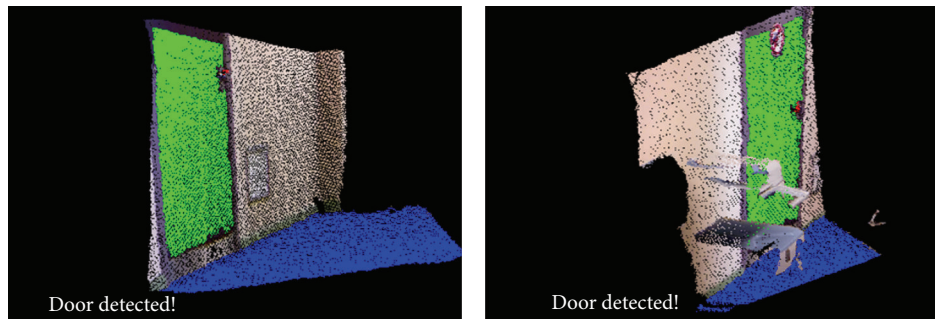


FIGURE 21: Detected doors.



FIGURE 22: Some examples of upstairs.

TABLE 2: The performance of some other approaches.

Authors	Method/technique	Dataset	Result
[28]	2D image processing	76 images	79.00%
[29]	2D/perceptual fuzzy models	390 images	90.50%
[30]	2D/edges and corners features	210 images	91.90%
[31]	2D/Canny edge detection	196 images	80.00%
Our solution	3D RGB-D image processing	30 images	90.69%

TABLE 3: Performance of the upstairs detection.

Number of images	TP	FP	Precision	Detection time per image
75	73	2	0.97	0.433 s

TABLE 4: Performance of the downstairs detection.

Number of images	TP	FP	Precision	Detection time per image
38	34	4	0.89	~0.40 s

Table 3. Some examples of upstairs images are shown in Figure 22 and the detected downstairs are shown in Figure 23.

The program was also capable of detecting the downstairs but with the lower performance. The downstairs are detected

in 89.47% out of 38 positive image samples. Table 4 shows the performance measures of the downstairs detection using our approach. Representative downstairs images and the detected downstairs are shown in Figures 24 and 25. In addition, there are some cases that the program cannot detect downstairs; see Figures 26 and 27.

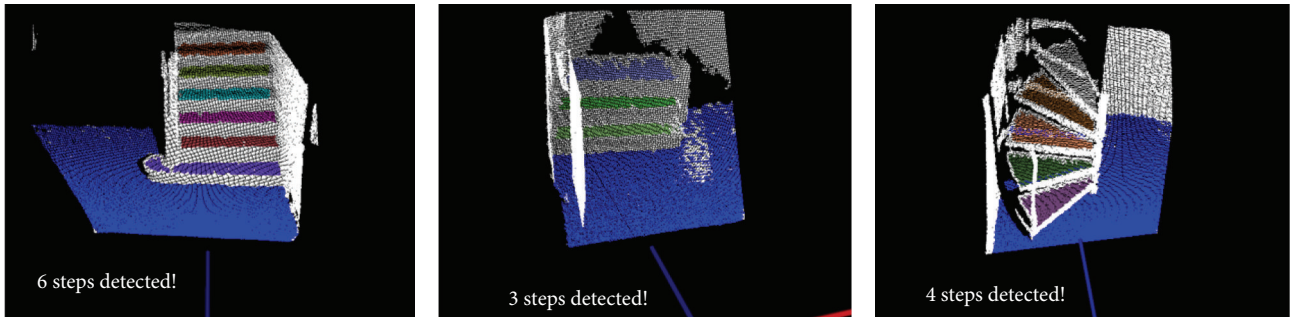


FIGURE 23: Detected upstairs.

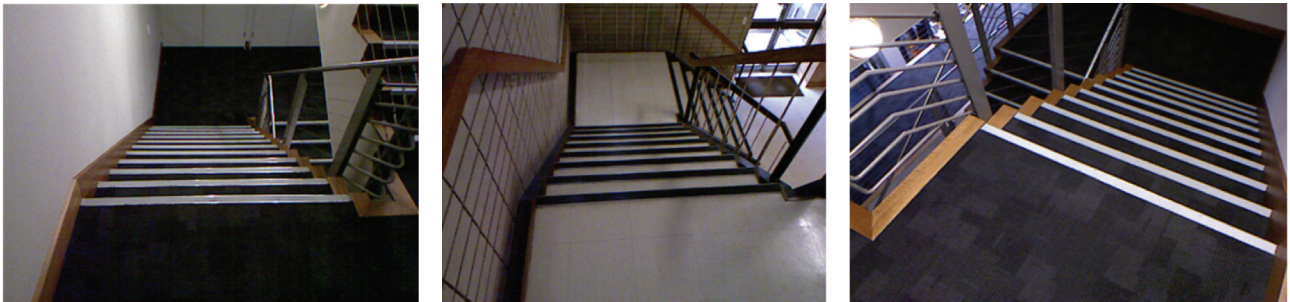


FIGURE 24: Some examples of downstairs.

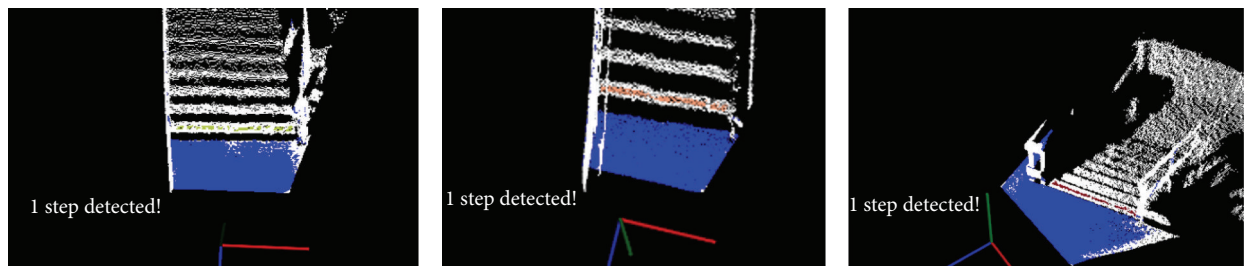


FIGURE 25: Detected downstairs.

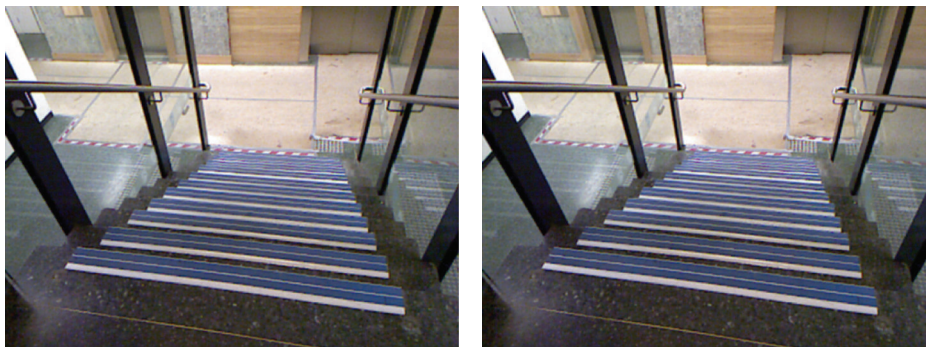


FIGURE 26: Some examples of downstairs.

The results show that our approach can be used for the low-light environments. This feature can overcome the limitations of the monocular or stereo vision technique [7, 8, 16].

The execution time for intermediate processing steps is negligible (about 0.04 s for the floor segmentation time and 0.009 s for the normal estimation time). The detection time per image (see Tables 2 and 3) includes all the steps

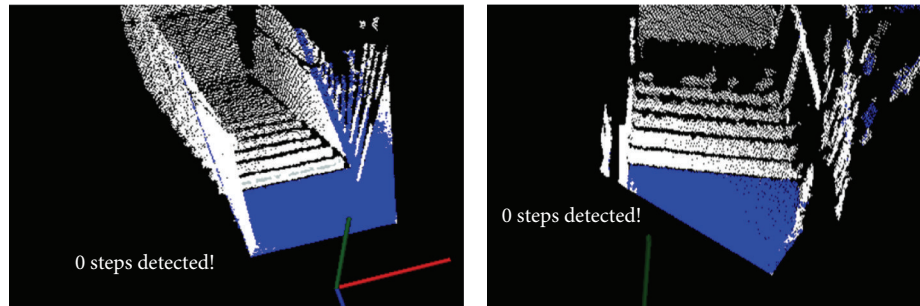


FIGURE 27: Undetected downstairs.

such as reconstruction, filtering, and segmentation. The total execution time of the algorithms is fast enough to develop the real-time applications for visually impaired people; the experiments performed in different indoor environments have produced positive results in terms of accuracy and time response.

5. Conclusions

In this paper, we have presented a Microsoft Kinect-based method specifically dedicated to the detection of obstacles in indoor environment based on 3D image processing with color-depth data. The proposed solution was designed to provide information about the space in front of the user in real time by using color-depth data acquired by Microsoft Kinect sensor. Our main contribution is the obstacle detection module, which combines different algorithms for obstacle detection such as walls, doors, stairs, and the bumpy parts on the floor.

Our proposed system shows good results with our dataset. However, a number of caveats and limitations still have to be taken into account. Firstly, the system is not reliable in all kinds of scenarios or data. For example, when the light condition is too strong, the Kinect cannot capture the depth information. In the future, we plan to combine the color and depth information in order to build a reliable approach for obstacle detection. Secondly, our obstacle detection relies on the floor ground detection. In the case that floor cannot be detected, our system will fail. For this, we have to use different cues for detecting the obstacle candidates. Thirdly, the performance of the downstairs detection program is still low, and we need a new approach for downstairs detection from far distance. Finally, our approach performs obstacle detection in every frame; therefore it does not take into account the results of the previous frame. In the future, we can reduce program execution time by combining four algorithms and apply probabilistic models for estimating the presence of obstacles in a given frame based on the result of previous ones.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by funding by the French national program “Programme d’Investissements d’Avenir IRT Nanoelec” ANR-10-AIRT-05, Institut Universitaire de France, and the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant no. FWO.102.2013.08. The authors would like to thank the members of AGIM laboratory and International Research Institute MICA for helpful comments and suggestions.

References

- [1] WHO, “Visual impairment and blindness,” Fact Sheet no. 282, WHO, 2014.
- [2] J. M. Loomis, A. C. Beall, K. L. MacUga, J. W. Kelly, and R. S. Smith, “Visual control of action without retinal optic flow,” *Psychological Science*, vol. 17, no. 3, pp. 214–221, 2006.
- [3] U. R. Roentgen, G. J. Gelderblom, M. Soede, and L. P. de Witte, “Inventory of electronic mobility aids for persons with visual impairments: a literature review,” *Journal of Visual Impairment and Blindness*, vol. 102, no. 11, pp. 702–724, 2008.
- [4] L. Russel, “Travel path sounder,” in *Proceedings of the Rotterdam Mobility Research Conference*, American Foundation for the Blind, Rotterdam, The Netherlands, May 1965.
- [5] L. Kay, “An ultrasonic sensing probe as a mobility aid for the blind,” *Ultrasonics*, vol. 2, no. 2, pp. 53–59, 1964.
- [6] B. Laurent and T. N. A. Christian, “A sonar system modeled after spatial hearing and echolocating bats for blind mobility aid,” *International Journal of Physical Sciences*, vol. 2, pp. 104–111, 2007.
- [7] L. Chen, B.-L. Guo, and W. Sun, “Obstacle detection system for visually impaired people based on stereo vision,” in *Proceedings of the 4th International Conference on Genetic and Evolutionary Computing (ICGEC '10)*, pp. 723–726, Shenzhen, China, December 2010.
- [8] Microsoft Kinect, <https://www.microsoft.com/en-us/kinectfor-windows/>.
- [9] ASUS Xtion Pro Live, May 2015, <https://www.asus.com/us/Multimedia/>.
- [10] Seeing with Sound—The voice, May 2015, <https://www.seeing-withsound.com/>.
- [11] M. Brock and P. O. Kristensson, “Supporting blind navigation using depth sensing and sonification,” in *Proceedings of the ACM Conference on Ubiquitous Computing (UbiComp '13)*, pp. 255–258, Zurich, Switzerland, September 2013.

- [12] R. Mojtahedzadeh, *Robot obstacle avoidance using the kinect [M.S. thesis]*, Royal Institute of Technology, Stockholm, Sweden, 2011, http://www.researchgate.net/publication/257541372_Robot_Obstacle_Avoidance_using_the_Kinect.
- [13] B. Peasley and S. Birchfield, "Real-time obstacle detection and avoidance in the presence of specular surfaces using an active 3D sensor," in *Proceedings of the IEEE Workshop on Robot Vision (WoRV '13)*, pp. 197–202, Clearwater Beach, Fla, USA, January 2013.
- [14] J. M. Benjamin, N. A. Ali, and A. F. Schepis, "A laser cane for the blind," *Proceedings of the San Diego Biomedical Symposium*, vol. 12, pp. 53–57, 1973.
- [15] N. Molton, S. Se, J. M. Brady, D. Lee, and P. Probert, "A stereo vision-based aid for the visually impaired," *Image and Vision Computing*, vol. 16, no. 4, pp. 251–263, 1998.
- [16] P. Costa, H. Zolfagharnasab, J. P. Monteiro, J. S. Cardoso, and H. P. Oliveira, "3D reconstruction of body parts using rgb-d sensors: challenges from a biomedical perspective," in *Proceedings of the 5th International Conference on 3D Body Scanning Technologies*, Lugano, Switzerland, October 2014.
- [17] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from RGBD images," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 842–849, IEEE, Saint Paul, Minn, USA, May 2012.
- [18] W. Niu, J. Long, D. Han, and Y. F. Wang, "Human activity detection and recognition for video surveillance," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, vol. 1, pp. 719–722, Taipei, Taiwan, June 2004.
- [19] Z. A. Mundher and Z. Jiaofei, "A real-time fall detection system in elderly care using mobile robot and kinect sensor," *International Journal of Materials, Mechanics and Manufacturing*, vol. 2, no. 2, pp. 133–138, 2014.
- [20] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the microsoft kinect," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 1, pp. 290–301, 2015.
- [21] A. Khan, F. Moideen, W. Khoo, Z. Zhu, and J. Lopez, "KinDetect: kinect detecting objects," in *Proceedings of the 13th International Conference on Computers Helping People with Special Needs, Linz, Austria, July 2012*, K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, Eds., vol. 7383, pp. 588–595, Springer, Berlin, Germany, 2012.
- [22] H. Tang, M. Vincent, T. Ro, and Z. Zhu, "From RGB-D to low resolution tactile: smark sampling and early testing," in *Proceedings of the IEEE Workshop on Multimodal and Alternative Perception for Visually Impaired People in Conjunction with IEEE International Conference on Multimedia and Expo Workshops (MAP4VIP-ICMEW '13)*, pp. 1–6, IEEE, San Jose, Calif, USA, July 2013.
- [23] Y. H. Lee and G. Medioni, "A RGB-D camera based navigation for the visually impaired," in *Proceedings of the RSS 2011 RGB-D: Advanced Reasoning with Depth Camera Workshop*, Los Angeles, Calif, USA, June 2011.
- [24] C. H. Park and A. M. Howard, "Real-time haptic rendering and haptic telepresence robotic system for the visually impaired," in *Proceedings of the IEEE World Haptics Conference (WHC '13)*, pp. 229–234, Daejeon, South Korea, April 2013.
- [25] A. H. Tamjidi, C. Ye, and S. Hong, "6-DOF pose estimation of a portable navigation aid for the visually impaired," in *Proceedings of the IEEE International Symposium on Robotic and Sensors Environments*, pp. 178–183, Washington, DC, USA, October 2013.
- [26] A. P. Yus, G. L. Nicolas, and J. J. Guerrero, "Detection and modelling of staircases using a wearable depth sensor," in *Computer Vision—ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III*, vol. 8927 of *Lecture Notes in Computer Science*, pp. 449–463, Springer, 2015.
- [27] A. Aladren, G. Lopez-Nicolas, L. Puig, and J. J. Guerrero, "Navigation assistance for the visually impaired using RGB-D sensor with range expansion," *IEEE Systems Journal*, 2015.
- [28] A. C. Murillo, J. Koščeká, J. J. Guerrero, and C. Sagüés, "Visual door detection integrating appearance and shape cues," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 512–521, 2008.
- [29] R. Munoz-Salinas, E. Aguirre, M. Garcia-Silvente, and A. Gonzalez, "Door-detection using computer vision and fuzzy logic," in *Proceedings of the 6th WSEAS International Conference on Mathematical Methods & Computational Techniques in Electrical Engineering*, December 2004.
- [30] Y. Tian, X. Yang, and A. Ardit, "Computer vision-based door detection for accessibility of unfamiliar environments to blind persons," *Machine Vision and Applications*, vol. 24, no. 3, pp. 521–535, 2013.
- [31] C. Juenemann, A. Corbin, and J. Li, *Robust Door Detection*, Stanford Department of Electrical Engineering, Stanford University, 2010.
- [32] T. H. Nguyen, T. H. Nguyen, T. L. Le, T. T. H. Tran, N. Vuillerme, and T. P. Vuong, "A wearable assistive device for the blind using tongue-placed electrotactile display: design and verification," in *Proceedings of the 2nd International Conference on Control, Automation and Information Sciences (ICCAIS '13)*, pp. 42–47, IEEE, Nha Trang, Vietnam, November 2013.
- [33] V. Hoang, T. Nguyen, T. Le, T. H. Tran, T. Vuong, and N. Vuillerme, "Obstacle detection and warning for visually impaired people based on electrode matrix and mobile Kinect," in *Proceedings of the 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS '15)*, pp. 54–59, Ho Chi Minh City, Vietnam, September 2015.
- [34] A. Jana, *Kinect for Windows SDK Programming Guide*, Packt Publishing, Birmingham, UK, 2012.
- [35] N. Burrus, "Calibrating the depth and color camera," 2014, <http://nicolas.burrus.name/index.php/Research/KinectCalibration>.
- [36] D. Bernabei, F. Ganovelli, M. Di Benedetto, M. Dellepiane, and R. Scopigno, "A low-cost time-critical obstacle avoidance system for the visually impaired," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN '11)*, Guimarães, Portugal, September 2011.
- [37] M. Vlamincx, L. Jovanov, P. V. Hese, B. Goossens, W. Philips, and A. Pizurica, "Obstacle detection for pedestrians with a visual impairment based on 3D imaging," in *Proceedings of the International Conference on 3D Imaging (IC3D '13)*, pp. 1–7, IEEE, Liège, Belgium, December 2013.
- [38] T. J. J. Tang, W. L. D. Lui, and W. H. Li, "Plane-based detection of staircases using inverse depth," in *Proceedings of the Australasian Conference on Robotics and Automation*, Wellington, New Zealand, 2012.
- [39] P. B. Thawari and N. J. Janwe, "Cbir based on color and texture," *International Journal of Information Technology and Knowledge Management*, vol. 4, no. 1, pp. 129–132, 2011.

