Ego-Semantic Labeling of Scene from Depth Image for Visually Impaired and Blind People

Chayma Zatout, Slimane Larabi, Ilyes Mendili, Soedji Ablam Edoh Barnabe
Computer Science Department
USTHB University
BP 32 EL ALIA, 16111, Algiers, Algeria
{czatout,slarabi}@usthb.dz

Abstract

This work is devoted to scene understanding and motion ability improvement for visually impaired and blind people. We investigate how to exploit egocentric vision to provide semantic labeling of scene from head-mounted depth camera. More specifically, we propose a new method for locating ground from depth image whatever the camera's pose. The rest of planes of the scene are located using RANSAC method, semantically coded by their attributes and mapped as cylinders into a generated 3D scene which will serve as a feedback to users. Experiments are conducted and the obtained results are discussed.

1. Introduction

In daily activities, the human being takes advantage of his six senses in order to accomplish these activities. If one of these senses is lost, some remaining senses improve to fill, relatively, the gap left by this absence and enhance the life quality.

With the absence of the sense of seeing, the visually impaired will rely highly on the sense of hearing which becomes thus more important. In many visually impaired aid systems, an audio feedback is used to transmit information and instructions. It's true that this latter enhance the life quality but it prevents the hearing sense to accomplish its usual tasks like detecting instant sound events that can make the visually impaired life in danger.

An alternative solution is to use feedback based on vibration. Although this latter liberates the hearing, it appears less informative and provides only a modest number of possible instructions.

The visually impaired aid systems consist of a set of techniques whose goal is to enhance the visually impaired life in different activities like outdoor navigation [10], indoor navigation [26] [6] [19] [4] [12] [1] [7] [3] [16], localization and grasping objects [22], obstacle avoidness and many other applications [9]. These systems can be traditional like white cane, guide dog or personal assistant; sophisticated by involving advanced technologies and computer science or hybrid by combining the two previous categories as implemented in [6].

From the data gathered from real world, the sophisticated systems generate instructions and signs that can be understandable by the visually impaired people. In these systems, depth or RGB sensors (or both) are often used as input device; and the input data is processed using either image processing techniques [19][16][3], computer vision techniques [22][27] or machine learning techniques [24].

In case of navigation and obstacle avoidness systems, the main task is to detect objects considered as obstacles or the ground that is considered as free space. The nature of output and the way of transmitting it is another challenge. In fact, delivering information or instructions is crucial, they have to be clear, concise, simple to understand and don't requiere lot of concentration and efforts.

We believe, as human beings, that having an impression or a description of our surroundings permit us to do a significant number of tasks.

In this work, we propose an end-to-end system (see figure 1) that enable users to have an impression of their surroundings and become more independent. Our main contributions are: at first, we propose a new method for ground detection from depth image. Secondly, after detecting horizontal planes we propose a semantic labeling based on a promising concept on obstacle classification. The semantic labeling is depicted as cylinders inside a generated 3D scene which will serve as an haptic feedback in order to facilitate the immersion of people in the surrounding.

The rest of this paper is organized as follow: Section 2 is devoted to related works. In section 3, we present our method for ground and planes detection. The semantic labeling and the generated scene are described in section 4.

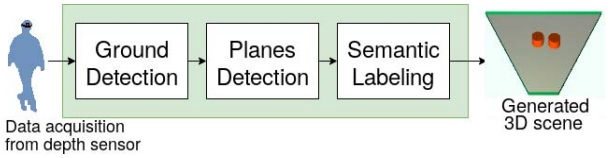

Figure 1. Visually impaired aid system for scene understanding and motion ability improvement. It receives a depth image as input, detects the ground, extracts horizontal planes and generates a 3D scene as output based on the provided semantic labeling.

Conducted experiments are presented in the section 5. The conclusion and future works terminate this paper.

2. Related works

Visually impaired systems based on image processing techniques are simple to implement and efficient when dealing with simple scenes. In general, these systems detect object features like edges as in [19] using Canny filter or corners as in [16] using Harris and Stephens corner detector on RGB image. As in [3], they used an adaptive sliding window and thresholding on a line of depth image as region of interest to detect the optimal moving direction. For complicated scenes, computer vision and machine learning techniques are used. *Thakoor et al.* [22] proposed Attention Biased Speeded Up Robust Features (AB-SURF), an algorithm to localize and recognize specific objects belonging to the data-set in real-time. *Wang et al.* [24] proceed differently, they distinguished free from occupied space by detecting planes. They proposed scene segmentation based on cascaded decision tree using RGB-D image to classify segments whether plane is ground, walls or tables.

When the previous researches seek to detect objects, others tried to detect free space (the ground) in order to avoid obstacles. *Zeineldin and El-Fishawy* [27] proposed an algorithm for obstacle detection based on clustering and enhanced RANSAC algorithm on 3D point cloud. They proposed an enhanced RANSAC algorithm based on not only the computing distance between points but also comparing their normals for floor detection. After extracting the floor, they applied Euclidean segmentation to detect objects.

Floor detection is a crucial step for obstacles detection. Authors in [8] introduced two methods for ground detection based on depth information. The simplest one is robust but assumes that the sensor pitch angle is fixed and has no roll, whereas the second one can handle changes in pitch and roll angles. They use the fact that if a pixel is from the ground plane, its depth value must be on a rationally increasing curve placed on its vertical position. However, this solution causes problems if the floor has significant inclination or declination.

RANSAC algorithm has been used for ground detection with different assumptions. In [5], the authors assumed that the space position of the floor is within z-max value. In [2], the authors distinguish the floor from obstacles and walls based on hue, lighting and geometry image features. If a pixel satisfies a defined criteria, it is labelled as floor-seed. In [15], RANSAC plane fitting is used to determine the ground plane in the 3D space. Because the sensor cannot be fixed, the calculation of the ground information requires an iterative approach. The ground's height is determined by using the V-disparity. In [13], ground plane is detected assuming strong constraints: ground plane must be large enough and Kinect is mounted on the human body such that the distance between ground plane and Kinect (y-axis coordinates) must be in a range of 0.8 to $1.2m$. Authors in [23] considered that ground plane since is located based on the fact that it has a normal perpendicular to the xz plane. However, the scene may present other planes with greater area and verifying this property. Also, in [18], RANSAC procedure is used to find planes, and the relative distance and orientation of each plane with respect to the camera are then tested to determine whether it is floor or not. If the floor is not found with the first cloud, a new one will be captured and the process will be repeated. In this work, as the presented method is devoted to staircases detection, it has not evaluated.

Guo and Hoim [11] proposed and algorithm for support surface (including the ground) prediction in indoor scenes based on RANSAC procedure: after computing surface normals from inpainted depth map and aligning them to the real-world coordinates, they segmented the fitted planes computed by RANSAC based on color and depth gradients. Finally, they designed a hierarchical segmentation and inferred the support structure based on initial estimates of the support and the 3D scene structure. It scored a high accuracy in ground detection on NYU Depth dataset v2 [21]. In this work, we will compare our obtained results with theirs.

Another sensitive module in visually impaired aid systems is the feedback interface. In general, it can be audio feedback [4][12][1][10][22][3][16], vibration-based feedback [17] or a combination of them [6][7]. The audio feedback can be stereo tone [3], beeps [3], recorded instructions [3] and text-to-speech [6][7]. It can deliver some local information (like obstacle in front of you, etc) or instructions to navigate (like go straight, etc), to grasp objects (like up, down, left), etc.

Audio feedback is simple to understand and provides clear and concise information or instructions if these latter are well coded. However, it occupies ears and thus prevent them from doing their job like detecting unsafe instant events such "a car passing by". The vibration-based feedback is an alternative to audio feedback since it does not occupy the sense of hearing. However, the number of pos-

sible instructions provided by vibration can be modest (4 or 8 possible instructions are usually used). Military code as used in [7], enriches the possible directions to take but can be ambiguous like distinction between 5 and 10 o'clock.

The discussed types can sometimes be annoying, distracting and can not be used in some locations as using audio feedback in hospitals when silence is needed or supermarkets when there is huge noise. In these cases headphone [1] [12] or earphone [4] [3] can be used but again, this holds hearing and isolates it relatively from real-world. With a simple informative semantic labeling, the navigation, the grasping and other tasks can be done without dictating instructions. Furthermore, such a labeling allow the visually impaired even blind people to have an impression about the real world and their surroundings. *Horne et al.*[14] proposed a semantic labeling for prosthetic vision for obstacle avoidance and object localization. They proposed a 2D pattern of phosphenes with different discrete level of intensity. In case of navigation, the phosphenes are activated to represent potential obstacles locations thus, free space was represented by gaps. So the user can have an impression about his surroundings and navigate without receiving audio instructions.

3. Ground and planes detection

3.1. Ground detection

Let $(Oxyz)$ be the coordinate system attached to the head-mounted depth camera performing any translation and roll, pitch, yaw rotations allowed by head motion.

Let $p(l, c)$ be the pixel located at row l and column c in the depth image I_d having n rows and m columns and let $P_{l,c}(x, y, z)$ be the associated 3D point in the scene where the coordinates x, y, z are computed after camera calibration.

The first step of our method is to select for each depth z_i and for each column c in depth image I_d, the pixel $p^*(l, c)$ such that its associated 3D point has the $z - component$ equal to z_i and a minimal value of $y - component$ for all $P_{l,c}(x, y, z)$, $l = 1..n$. This allows determining images of all points P of the plane Π_i parallel to the $xy-$plane such that $z = z_i$ as indicated by figure 2. The set of located pixels $p^*(l, c)$ for a given depth z_i defines a curve noted (G_i) whose allure depends on the orientation of the $xz-$plane relatively to the ground (see figure 2).

The second step consists to remove pixels from (G_i) corresponding to objects. In order to facilitate the geometrical illustration, we draw the curves G_i considering that the $xz-$plane of the camera is parallel to the ground. The curve (G_i) will contain many convex and concave parts as shown by figure 3 due to the presence of objects on the ground. By scanning a (G_i) from left to right, we associate a label (cv for convex or cc for concave) to each part. All cv parts

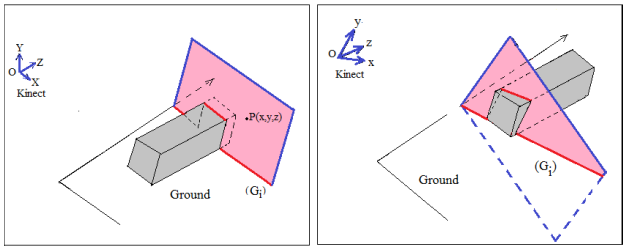

Figure 2. The obtained curve (G_i) in case where: (Left) the $xz-$plane is parallel to the ground, (right) the camera performs yaw and roll rotations.

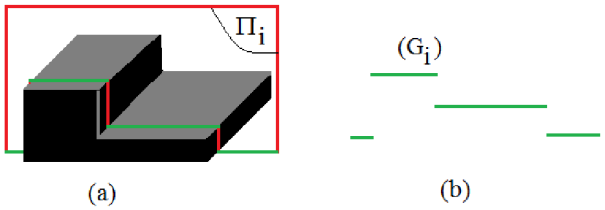

Figure 3. (a) The set of 3D points at given depth with minimal y-coordinate (case where $xz-$plane is parallel to the ground) (colored in green), (b) Pixels in depth image defining the curve (G_i).

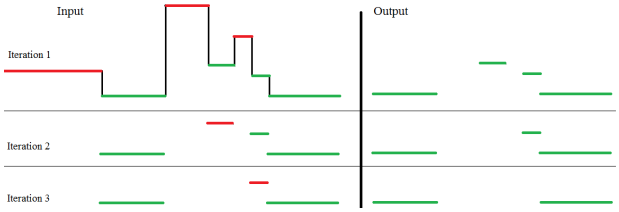

Figure 4. Removing iteratively convex parts (in red color) from G_i to keep only the ground corresponding to concave parts (green color). In the left the input curve (G_i), in the right the output of the algorithm which is used as a new input.

are removed. The labelling and convex parts removal are repeated until there will be no convex parts (see figure 4). The algorithm 1 summarizes the steps to be performed for the computation of (G_i).

In general case, the camera may perform roll, pitch or yaw rotations. We show by figure 5 an example of G_i computation for three values of depth z_i under roll and pitch rotations of the camera.

3.2. Planes detection

Once the ground has been detected, the planes constituting the occupied space are detected using RANSAC and a semantic labeling is affected (fig. 6). After breaking down the depth image into free space (ground) point cloud and occupied space point cloud (fig. 6 (a)), we first applied down sampling on the occupied space point cloud (fig. 6 (b)) as mentioned in [26] to reduce the number of points to be processed and thus decrease the computational complexity of RANSAC. Secondly, we applied RANSAC (fig.

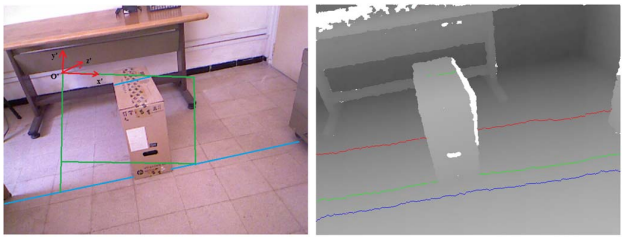

Figure 5. (Left) Acquired image with a roll and pitch rotations of the camera. In this case, the cut plane (in green color) is not orthogonal to the ground. The blue parts correspond to 3D points having the same depth and minimum value of y-coordinates. (Right) The computed curves G_i for three values of z_i. Note that the second curve G_i passes by the bottom of the box but the convex part is located on the high part of the box due to the inclination of the cut plane.

Algorithm 1 Algorithm $DCGD$ (Depth-cut based Ground Detection

Input: $I_d(n \times m) = \{p(l,c), l = 1..n, c = 1..m\}$,
The points cloud $\mathbb{P} = \{P_{l,c}(x,y,z)\}$
Output: The set $\mathbb{G}$ of ground pixels $p^*(l,c)$

$\mathbb{G} \leftarrow \emptyset$;
Determine $\mathbb{Z} = \{z_i / \exists p(l,c) \in I_d, P_{l,c}(x,y,z)\ verify\ z = z_i\}$;
for *each* $z_i \in \mathbb{Z},\ i = 1..Card(\mathbb{Z})$ **do**
  $G_i \leftarrow \emptyset$;
  for *each column* $c = 1..m$ **do**
    for *each* $p(l,c),\ l = 1..n$ **do**
      Determine $p(l,c)$ / $P_{l,c}(x,y,z)$ *verify* $z = z_i$;
    end for
    Select $p^*(l,c)$ *associated to* $P_{l,c}(x,y^*,z^*)$ / $(z^* = z_i)$ *and* $(y^* = Min(y_c oordinate)\ of\ P_{l,c})$;
    $G_i \leftarrow G_i \cup \{p^*(l,c)\}$;
  end for
  Remove from G_i convex parts and keep only concave parts.
  $\mathbb{G} \leftarrow \mathbb{G} \cup G_i$
end for
return $\mathbb{G}$

6 (c)) for plane segmentation on the reduced occupied space point cloud and we identified parallel planes to the ground (fig. 6 (d). To reduce the RANSAC's run time and the number of insignificant possible planes we set the distance error threshold up to $2cm$. This latter may affect the result by accepting some outliers but it will be proportionally handled later when needed.

In order to get the occupied space, we extracted the convex hull (fig. 6 (f)) encompassing each plane parallel to ground. To enhance the RANSAC's result and due to sensibility of the convex hull technique toward noisy data, we

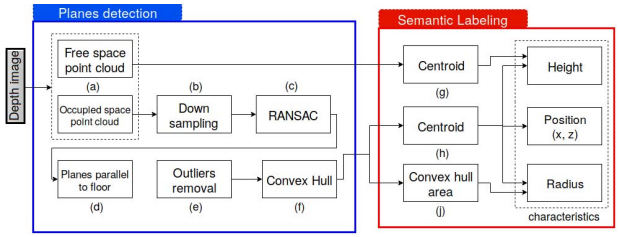

Figure 6. Planes detection and semantic labeling framework.

first projected the plane into its equation and then applied a statistical outlier removal filter (fig. 6 (e)) to refine the projected plane boundaries before extracting the plane's convex hull. This filter provides a Gaussian distribution of the point cloud with a given standard deviation by computing the mean distance from a given point to all its K neighbors and removing it if its mean distance is outside an interval defined by the global distances mean.

At the end of planes detection process, parallel planes to ground are represented by their convex hulls. These latter are used later to extract occupied space characteristics that will be used for our proposed semantic labeling.

4. Semantic labeling

In our first attempt, we seek to semantically label only free space and horizontal planes. Since the ground represents the free and the safe space for the visually impaired and blind people, we label it by considering the surface of the proposed generated scene as the ground. In other words, the lowest surfaces when touching will represent the free space, and the other surfaces represent the planes parallel to ground.

On the other hand, the parallel planes to ground are represented by a cylinder having a specific characteristics that are related to the plane characteristics in the real world (fig. 6). Each cylinder has its position (the coordinates x and z of its center), its height that represents the plane's height from the ground and its radius to represent the area occupied by the plane in the real world . To conduct this, we computed the centroid and the area of the concerned convex hull; and the free space point cloud centroid. The center's position is represented by the plane's centroid coordinates (x and z coordinates). The height is found by subtracting the y-coordinates of the plane's centroid and the ground's centroid. As for the radius, we searched for the radius of a circle having the same area as the area of the convex hull.

Furthermore, to transmit to the user how much a plane is high and how large, we propose a new promising and simple to obtain object classification for visually impaired and blind people regarding the object height and the occupied area. Regarding the height, the first level represents the planes having less than $0.3m$ that can be traversed by feet. The second level represents the planes having less than

$1.6m$ that can be touched by hands. The planes with height higher than $1.6m$ are represented by the third level. As for area, the first degree represents the planes occupying small area with a radius less than $20cm$ that can be explored only by moving hands without effort. The second degree represents the planes occupying a medium area with radius less than $35cm$. This type of plane can be explored by hands but may need stretching the arm. The third area represents the planes with a huge area that can not be entirely explored only by stretching arms but may also require moving around the plane.

5. Experiments

5.1. Datasets

We used two datasets to evaluate the proposed framework:
- NYU Depth dataset V2[21], is comprised of video sequences from a variety of indoor scenes as recorded by both the RGB and Depth cameras from the Microsoft Kinect. It features: 1449 densely labeled pairs of aligned RGB and depth images. Each object in the image is labeled with a class and an instance number. Figure 12 shows some images taken from NYU dataset.
- Our dataset of ground detection for indoor scenes (GDIS dataset) [25] includes different depth and color images acquired by an RGB-D sensor (Microsoft Kinect V1) for various orientations and poses. The ground truth corresponding to the floor is indicated in both images (depth and color). Figures 7, 10, 11 show some examples of the $GDIS$ dataset.

5.2. Evaluation of $DCGD$ Algorithm

5.2.1 Implementation details

The $DCGD$ algorithm can be divided into two main steps namely curves (G_i) construction (fig. 7) and ground detection. The first step is done on one shot by browsing the depth image only once. The complexity of this step having as input a $n \times m$ depth image is $O(n^2)$. Furthermore, a downsampling by depth $step$ can be performed to reduce the complexity and improve the detection quality. The second step includes subdividing of a curve into sub-curves (cv) or (cc) and finding floor from objects. The subdivision (fig. 7) can be performed according to certain criteria such as the permitted error in height between elements in same sub-curve h_err or the distance from the mean. Setting the value of these latter (height and downsampling step) depends on the sensor nature. This step is applied for all the obtained curves; so we need $2k$ iterations including curve subdivision and floor finding from k curves. Thus, the proposed algorithm's complexity is $O(n^2)$.

Another step can be added to reduce noise: some sub-curves can be generated due to noise and affect the floor detection; thus, removing sub-curves having small size $size\_err$ can reduce noise as shown in figure 8.

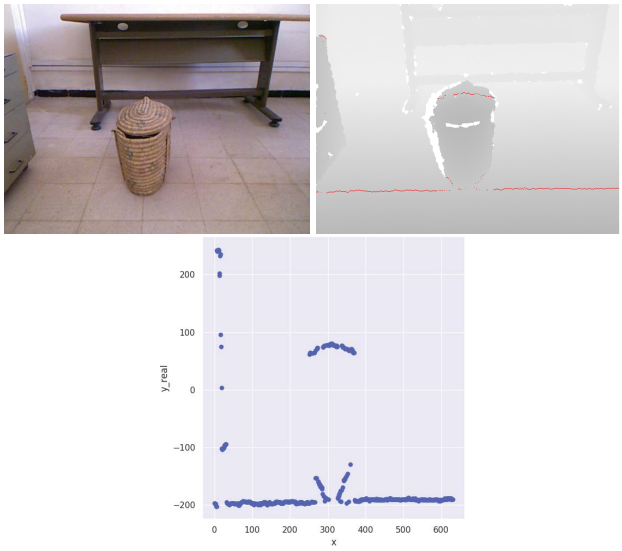

Figure 7. (Top) Color and depth image with located pixels (red color) for a given depth z_i. (Bottom) Curve reconstruction plot and curve subdivision into sub-curves.

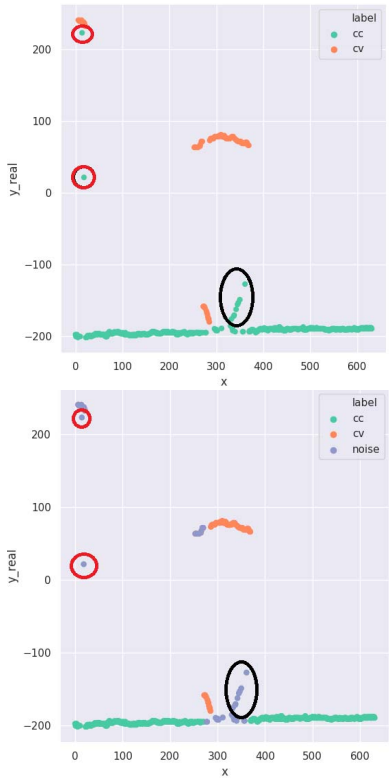

Figure 8. Floor detection: without reducing noise (top) and with reducing noise (bottom). Note that with reducing noise we prevent some false positive cases (circled by red). Note also, other noise area (circled by black) was appeared at the object borders.

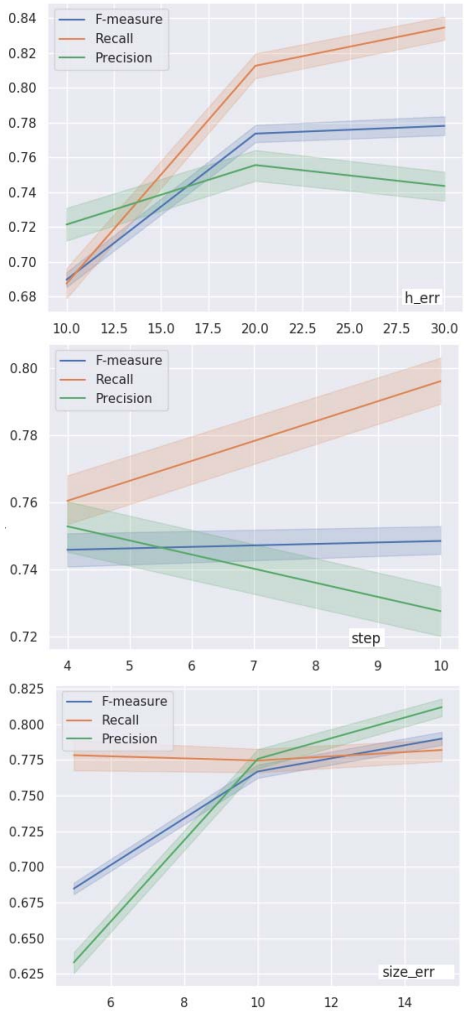

Figure 9. Setting the parameters h_err, $step$, $size\_err$.

5.2.2 Parametric analysis

To evaluate effectiveness of each discussed parameters namely $step$, h_err and $size\_err$, we plotted the algorithm performance applied to NYU Depth dataset V2[21] by varying each parameter as shown in figure 9. For later evaluation, we set $step$ to 4, h_err to 30 and $size\_err$ to 15.

5.2.3 GDIS dataset

We applied our method on GDIS dataset [25]. The ground is located in real time with accuracy. Different scenarios have been tested with different orientations of the Kinect sensor. Figure 10 shows qualitative results for a sample of scenes. Note that some pixels of the ground are missing due to the bad quality of depth image. In addition, as the color image is larger than the depth image, the left and right of the color image did not appear in depth image and thus not processed (see figure 10).

First row of figure 11 gives details of computed curve

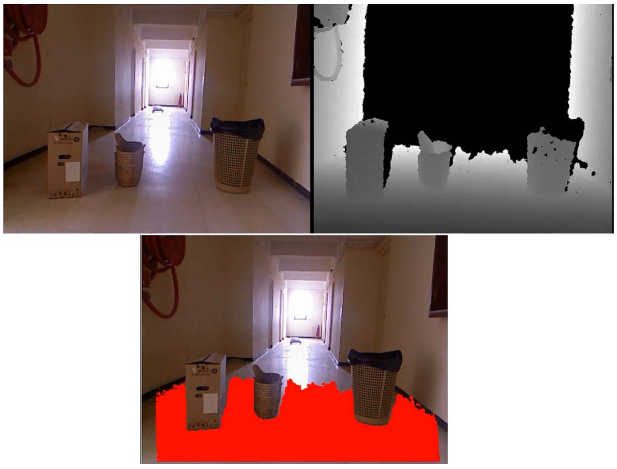

Figure 10. (Top) Color and depth image, (bottom) ground colored with red color.

(G_i) for one value of depth by applying $DCGD$ Algorithm. The convex part corresponding to foot table in drawn (G_i) contains is removed, the remaining parts constitute the ground. Note that pixels defining the curve (G_i) aren't aligned because 3D points at the same depth z_i have in acquired data different values of depths. We selected then all pixels having the value $z_i \pm 10mm$. We note that the area under the seat is detected as ground, which corresponds to the truth. However, if we search to locate obstacles, the plane above this ground's area will be taken into account. In the second row of the same figure 11, three depths are considered. Note that all (G_i) have the same slope corresponding to the orientation of the Kinect sensor relatively to z-axis.

The measures Precision, Recall and F-measure have been computed considering that the ground truth of the floor begins from the far pixel on the depth image. The average of computed measures are: $Precision = 0.98$, $Recall = 0.93$ and the $F - measure = 0.96$.

5.2.4 NYU Depth dataset V2

We evaluated $DCGD$ Algorithm using NYU Depth dataset V2[21]. The proposed algorithm scored a good performance nonetheless, the algorithm fails in some cases where depth images are noisy. Figure 12 shows different results obtained with high, medium and low score of accuracy. Our proposed algorithm scored a highest accuracy (91.84%) for ground detection compared to proposed one in [11] (80.3%). Thus, the $DCGD$ Algorithm is more robust specially when dealing with noisy data compared to [11]. Figures 13 and 14 illustrate respectively the values distribution of different metrics so as ROC curve and the confusion matrix.

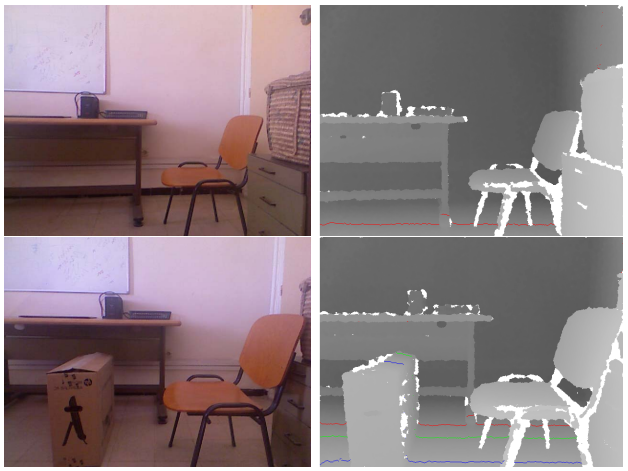

Figure 11. First row: Color and associated depth image of indoor scene. The computed curve (G_i) for depth $z_i = 2000m$ is drawn in red color. Note the presence of an obstacle (table foot) which produces a convex part in (G_i) which is eliminate in the second iteration of the algorithm. The discontinuity of (G_i) is due to occlusion of the area located at the depth z_i by the feet of the seat. Second row: Located curves (G_i) for three depths z_i equal to $1500mm$, $1800mm$, $2000mm$ drawn respectively in blue, green and red color.

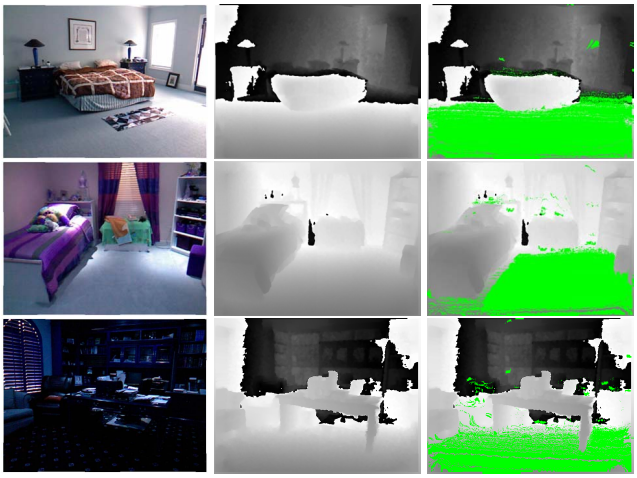

Figure 12. From left to right: Color image from NYU Depth dataset V2 [21], depth image, ground pixels in green color. From top to bottom: Case of high, intermediate and of low score

5.3. Planes detection and the semantic labeling

Once the ground has been detected, the occupied space was segmented and parallel planes to floor are retrieved. It should be noted that in this research, retrieving planes is not our main focus; we have just used algorithms from the state of the art. Planes are detected in nearly real time due to the use of the basic RANSAC implemented by Point Cloud Library (PCL [20]). For experiments purposes, we have taken two frames: with only one obstacle and with two obstacles parallel to floor (fig. 15). Note that by using basic

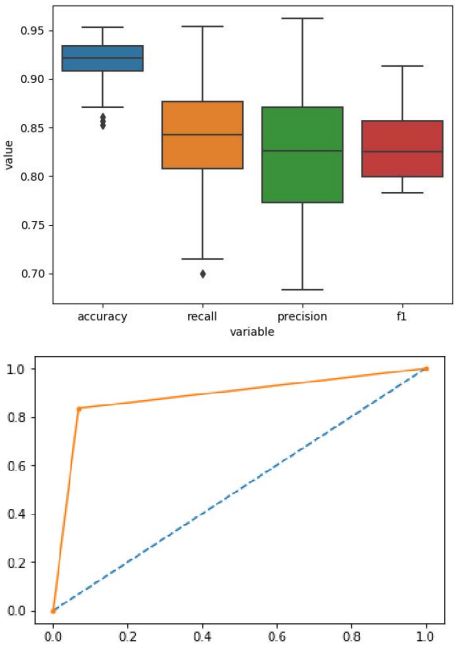

Figure 13. (Top) Values distribution for different metrics: The DCGD algorithm performs well for the majority of scenes (exceed 83%) in terms of the computed evaluation metrics. (Bottom) ROC curve: DCGD performs well in both sensitivity and specificity.

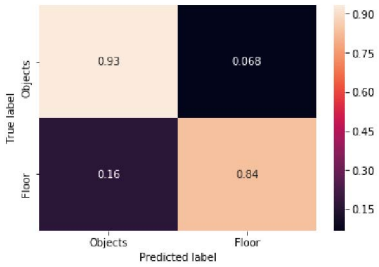

Figure 14. Confusion matrix: DCGD performs well in ground detection with a confusion does not go beyond 2.5%.

RANSAC, some outliers persist: points that do not belong to obstacles but they were considered as part of the detected planes as seen in Figure 15 surrounded by red circles. Fortunately, the applied statistical outlier removal filter implemented by PCL [20] reduced the noise and thus these latter do not affect significantly the plane characteristics computation such as height and radius. The table was not detected as plane parallel to floor, it was detected in fact as perpendicular plane since the table's perpendicular area is larger than the parallel one.

In order to compute how much the obtained characteristics are near to the real wold measurements, we have taken obstacle's measures and then compared them with the computed characteristics. The Mean Absolute Error (MAE) does not exceed $46mm$ for both characteristics, this latter is generally insignificant in regards our proposed labeling.

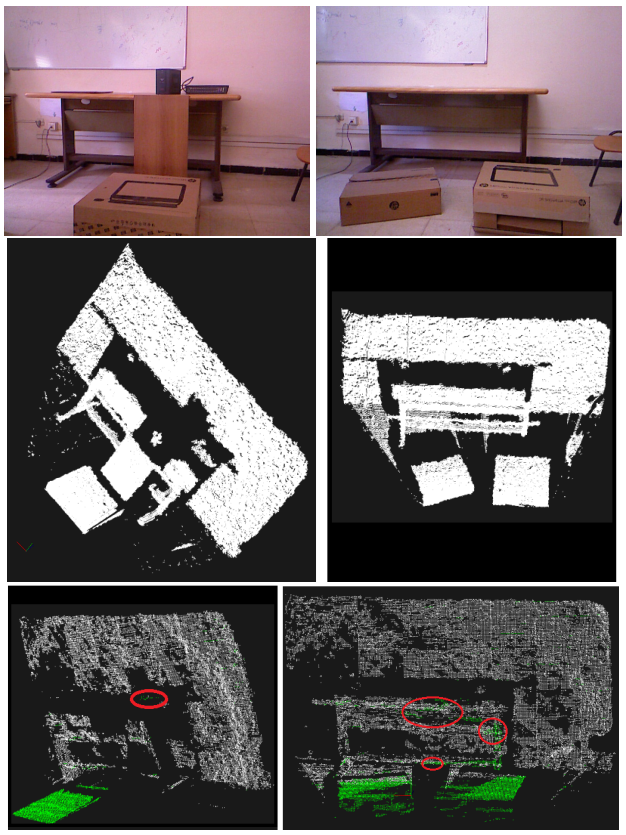

Figure 15. From top to bottom: Color images, Occupied space point cloud, Planes parallel to floor in green color.

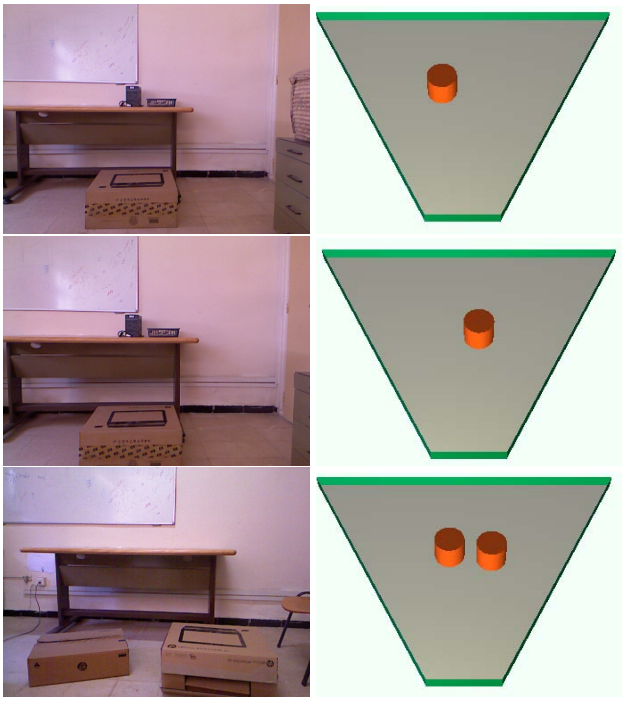

Figure 16. For each row: example of scene and the coding on generated scene.

Figure 16 illustrates the generated cylinders corresponding to provided semantic labeling of three scenes. The first scene was taken in our first position and second scene was taken after few steps ahead. In the two scenes, the generated scene indicates that there is a free space followed by an object having height less than $300mm$ and it can be explored by hands but may need stretching the arm (its radius is less than $350mm$). Note that in the second scene, the position of the cylinder changed to let the user understand that the obstacle became closer. In other words, the area of free space has became smaller. Concerning the third scene, the generated scene indicates that there are two objects, after a free space, having as height less than $300mm$ and they can be explored by hands but may need stretching the arm.

6. Conclusion and future works

In this paper we proposed for ground detection, a new algorithm running in real time and competing with the state of the art in terms of accuracy. In addition, in order to offer to visually impaired and blind people the ability to understand with immersion the scene content, a semantic labeling of scene is proposed and coded in generated 3D scene. The semantic labeling made in this paper concerned only the ground and horizontal planes corresponding to obstacles. Once located from depth image, their attributes are computed and coded on the generated scene using cylinders with different heights and radius. The proposed coding of scene content is elementary but efficient if we consider that the targeted semantic is the space occupancy by objects. There is no difference between table, seat or suitcase. Also, the produced coding is made from one frame and does not address frames registration.

Our future works will be focused on three main tasks:
- By moving the depth camera, the generated cylinders must move on the area of generated scene with new attributes. Only new objects entering in the field of view of the camera will be coded.
- Use the state of the art of object recognition and improve it for labelling more object classes.
- Find the suitable coding, similar to the braille system, that must offer to visually impaired and blind people sufficient information about its surrounding.

References

[1] Reham Abobeah, Mohamed E Hussein, Moataz M Abdelwahab, and Amin Shoukry. Wearable rgb camera-based navigation system for the visually impaired. In *VISIGRAPP (5: VISAPP)*, pages 555–562, 2018. 1, 2, 3

[2] 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, vol. 10, no. 3*, pages 922–932, 2016. 2

[3] Jinqiang Bai, Shiguo Lian, Zhaoxiang Liu, Kai Wang, and Dijun Liu. Smart guiding glasses for visually impaired people in indoor environment. *IEEE Transactions on Consumer Electronics*, 63(3):258–266, 2017. 1, 2, 3

[4] Alexy Bhowmick, Saurabh Prakash, Rukmani Bhagat, Vijay Prasad, and Shyamanta M Hazarika. Intellinavi: Navigation for blind based on kinect and machine learning. In *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pages 172–183. Springer, 2014. 1, 2, 3

[5] Li Bing, Zhang Xiaochen, Pablo Muñoz J., Xiao Jizhong, Rong Xuejian, and Tian Yingli. Assisting blind people to avoid obstacles: A wearable obstacle stereo feedback system based on 3d detection. In *IEEE Conference on Robotics and Biomimetics*, pages 2307–2311. IEEE, 2015. 2

[6] Paulo Costa, Hugo Fernandes, Paulo Martins, João Barroso, and Leontios J Hadjileontiadis. Obstacle detection using stereo imaging to assist the navigation of visually impaired people. *Procedia Computer Science*, 14:83–93, 2012. 1, 2

[7] Paraskevas Diamantatos and Ergina Kavallieratou. Android based electronic travel aid system for blind people. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 585–592. Springer, 2014. 1, 2, 3

[8] Kircali Dogan and Tek Boray. Ground plane detection using an rgb-d sensor. In *Information Sciences and Systems*, pages 69–77. Springer, 2014. 2

[9] Monica Gori, Giulia Cappagli, Alessia Tonelli, Gabriel Baud-Bovy, and Sara Finocchietti. Devices for visually impaired people: High technological devices with low user acceptance and no adaptability for children. *Neuroscience & Biobehavioral Reviews*, 69:79–88, 2016. 1

[10] João Guerreiro, Dragan Ahmetovic, Kris M Kitani, and Chieko Asakawa. Virtual navigation for blind people: Building sequential representations of the real-world. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 280–289. ACM, 2017. 1, 2

[11] Ruiqi Guo and Derek Hoiem. Support surface prediction in indoor scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2144–2151, 2013. 2, 6

[12] Osama Halabi, Mariam Al-Ansari, Yasmin Halwani, Fatma Al-Mesaifri, and Roqaya Al-Shaabi. Navigation aid for blind people using depth information and augmented reality technology. *Proceedings of the NICOGRAPH International*, pages 120–125, 2012. 1, 2, 3

[13] V Hoang, T Nguyen, and T Le. Obstacle detection and warning system for visually impaired people based on electrode matrix and mobile kinect. *Vietnam J Comput Sci (4)*, pages 71–83, 2017. 2

[14] Lachlan Horne, Jose Alvarez, Chris McCarthy, Mathieu Salzmann, and Nick Barnes. Semantic labeling for prosthetic vision. *Computer Vision and Image Understanding*, 149:113–125, 2016. 3

[15] Hsieh-Chang Huang, Ching-Tang Hsieh, and Cheng-Hsiang Yeh. An indoor obstacle detection system using depth information and region growth. *Sensors, 15*, pages 27117–27141. 2

[16] Nadia Kanwal, Erkan Bostanci, Keith Currie, and Adrian F Clark. A navigation system for the visually impaired: a fusion of vision and depth sensor. *Applied bionics and biomechanics*, 2015, 2015. 1, 2

[17] Young Hoon Lee and Gérard Medioni. Rgb-d camera based navigation for the visually impaired. In *Proceedings of the RSS*, 2011. 2

[18] A Perez Yus, G Lopez Nicolas, and J J Guerrero. Detection and modelling of staircases using a wearable depth sensor. In *ECCV 2014: Computer Vision - ECCV 2014 Workshops*, pages 449–463. Springer. 2

[19] R Gnana Praveen and Roy P Paily. Blind navigation assistance for visually impaired based on local depth hypothesis from a single image. *Procedia Engineering*, 64:351–360, 2013. 1, 2

[20] R B Rusu and S Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4. IEEE, May 2011. 7

[21] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2, 5, 6, 7

[22] Kaveri Thakoor, Nii Mante, Carey Zhang, Christian Siagian, James Weiland, Laurent Itti, and Gérard Medioni. A system for assisting the visually impaired in localization and grasp of desired objects. In *European Conference on Computer Vision*, pages 643–657. Springer, 2014. 1, 2

[23] M Vlaminck, Q L Hiep, V N Hoang, H Vu, P Veelaert, and W Philips. Indoor assistance for visually impaired people using a rgb-d camera. In *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. IEEE, 2016. 2

[24] Zhe Wang, Hong Liu, Xiangdong Wang, and Yueliang Qian. Segment and label indoor scene based on rgb-d for the visually impaired. In *International Conference on Multimedia Modeling*, pages 449–460. Springer, 2014. 1, 2

[25] Chayma Zatout and Slimane Larabi. Dataset of ground detection for indoor scenes (gdis dataset). In *http://perso.usthb.dz/ slarabi/DGIS.html*, 2019. 5, 6

[26] Ramy Zeineldin and Nawal El-Fishawy. Fast and accurate ground plane detection for the visually impaired from 3d organized point clouds. *2016 SAI Computing Conference (SAI)*, pages 373–379, 07 2016. 1, 3

[27] Ramy Ashraf Zeineldin and Nawal Ahmed El-Fishawy. Fast and accurate ground plane detection for the visually impaired from 3d organized point clouds. In *2016 SAI Computing Conference (SAI)*, pages 373–379. IEEE, 2016. 1, 2