Hindawi
Journal of Sensors
Volume 2021, Article ID 6685686, 18 pages
https://doi.org/10.1155/2021/6685686

Research Article

Vision-Based System for Assisting Blind People to Wander Unknown Environments in a Safe Way

Andrés A. Díaz-Toro,[1] **Sixto E. Campaña-Bastidas**,[1] **and Eduardo F. Caicedo-Bravo**[2]

[1] *School of Basic Sciences, Technology and Engineering ECBTI, Universidad Nacional Abierta y a Distancia UNAD, Pasto 520001, Colombia*
[2] *School of Electrical and Electronic Engineering EIEE, Universidad del Valle, Cali 76001, Colombia*

Correspondence should be addressed to Andrés A. Díaz-Toro; andres.diaz@unad.edu.co

Received 1 December 2020; Revised 18 February 2021; Accepted 20 February 2021; Published 5 March 2021

Academic Editor: Eduard Llobet

Vision is the principal source of information of the surrounding world. It facilitates our movement and development of everyday activities. In this sense, blind people have great difficulty for moving, especially in unknown environments, which reduces their autonomy and puts them at risk of suffering an accident. Electronic Travel Aids (ETAs) have emerged and provided outstanding navigation assistance for blind people. In this work, we present the methodology followed for implementing a stereo vision-based system that assists blind people to wander unknown environments in a safe way, by sensing the world, segmenting the floor in 3D, fusing local 2D grids considering the camera tracking, creating a global occupancy 2D grid, reacting to close obstacles, and generating vibration patterns with an haptic belt. For segmenting the floor in 3D, we evaluate normal vectors and orientation of the camera obtained from depth and inertial data, respectively. Next, we apply RANSAC for computing efficiently the equation of the supporting plane (floor). The local grids are fused, obtaining a global map with data of free and occupied areas along the whole trajectory. For parallel processing of dense data, we leverage the capacity of the Jetson TX2, achieving high performance, low power consumption, and portability. Finally, we present experimental results obtained with ten (10) participants, in different conditions, with obstacles of different height, hanging obstacles, and dynamic obstacles. These results show high performance and acceptance by the participants, highlighting the easiness to follow instructions and the short period of training.

1. Introduction

Electronic Travel Aids (ETAs) for assisting blind people, especially vision-based aids, have taken an approach based on autonomous vehicles since both have to deal with similar challenges, such as real-time performance, handle previously unseen environments, be robust to different conditions and dynamic environments, and be safe for the user, for the people, and for objects around it. In this sense, algorithms in the field of autonomous vehicles can be used for assisting blind people in navigation tasks, such as scene understanding, object detection, segmentation, path planning, localization, and mapping.

The data that these algorithms process must be accurate and reliable. Vision sensors have become a tendency in the field of self-driving cars due to its high accuracy, high frame rate, low weight, small size, and low price. However, visual data is dense so we need to use high-performance processors like the general-purpose graphics processing units GPGPUs that work in parallel way, achieving to speed up the performance up to 10 times with respect to implementations in CPU. Besides to having high-performance processor, the system needs to be wearable, which means to have small size and low weight in order to be hand-carried easily or coupled to the user's body. After processing the information of the sensors, a control module decides the most appropriate actions for each situation. For autonomous vehicles, the control is carried out with inputs to its motors while in the case of blind people, their guidance to free space (or to a goal position) can be done with audio feedback or vibration patterns generated by an haptic belt. Although autonomous vehicle scenarios are limited as compared to humans and both tasks do not face identical goals, navigation aids for the blind can be inspired by the notable advances in the former field.

In this work, we consider the Manhattan world assumption and apply a technique for segmenting the floor based on inertial sensors and normal vectors computed from depth data that is obtained from a stereo camera. The system detects changes in height with respect to the ground plane and builds an incremental occupancy grid in two dimensions. Once we have information about free space, we map it to motion instructions using a reactive navigation algorithm, achieving to guide the user around an unknown environment without collisions. The global occupancy grid has the advantage to store information during the whole trajectory in an incremental and compressed way, which enables the system to work in more complex tasks such as path planning to guide the user to objects of interest (the integration of an object detector and a path planning algorithm are left as future work). Moreover, the system does not need to insert in the environment neither RFID tags nor beacons and does not communicate with an external server for processing data, and the distance for detecting obstacles can be configured. These obstacles can be dynamic and can be located at a height over the user's waist, which is not possible with a white cane.

The contributions of this system are as follows:

(1) An algorithm for segmenting at real time the floor (ground plane) based on inertial sensors, normal vectors, the gravity vector, and RANSAC (random sample consensus)

(2) An algorithm that builds at real time an occupancy 2D grid for identifying free space. This grid is built in incremental way by fusing local data, generating a global occupancy grid

(3) An algorithm for reacting to close obstacles and for generating motion commands through an haptic belt

(4) The experimental results that define the performance and acceptance by the participants of the implemented system, under different conditions.

2. Related Work

Electronic Travel Aids (ETAs), proposed during the last years for assisting blind people, include a wide spectrum of sensors, feedback, and processors. The most common sensors are ultrasonic sensors [1–5], structured light cameras [6–8], stereo cameras [9, 10], and monocular cameras [11]. Each sensor has some drawbacks that limit the functionality and pervasiveness of the system. For example, ultrasonic sensors have difficulty for estimating orientation of obstacles due to the bean width, which is about 15°. Monocular cameras cannot provide scale in the maps. Structured light cameras, like the Kinect, are affected by other infrared sources such as sunlight in outdoors. Stereo cameras have low accuracy, especially with low-light conditions and with low texture. In this sense, the fusion of sensors has allowed to solve some problems and provide more complete functionalities. For example, in [12, 13], ultrasonic sensors together with a structured light camera are used for avoiding obstacles, including small and transparent ones. In [14], an ultrasonic sensor is used to estimate the distance of objects at the level of eyesight and integrates a monocular camera to recognise objects and read out text around the user. In [15], ultrasonic sensors are used together with RFID readers and tags in order to identify indoor objects. For feedback, the senses of hearing and touch are the most common ones used for replacing the sense of sight. For audio feedback, there have been tested tones (beeps) [16], stereo tones [17], augmented reality sound sources [18], and voice commands [4, 13, 19]. The general disadvantage is related with overloading the sense of hearing, especially in dynamic environments, like in [17]. For tactile feedback, Braille displays [20, 21], haptic belts [22–25], vests [26], and gloves [27] are used. Braille displays are difficult to interpret in dynamic environments, and the other ones need a period of training for interpreting the commands adequately [22]. The system [24] guides a blind person from his/her current position to a goal point using a tactile belt with eight vibrating motors that generates directional and rotational patterns. A stationary laser scanner tracks the user inside a room by fitting an ellipse to the torso. ROS (robot operating system) provides a local navigation planner for computing an obstacle free path, robust to possible deviation of the user and changes in the environment. The authors in [5] use both vibration motors and audio feedback for guiding blind people to free space. Four motors and six ultrasonic sensors are located in a shoe, together with a liquid detector sensor. The processor is other component that has an important impact in the functionality of the system. It must be small, light (wearable), and able to process great amount of data at real time, especially when dense data is delivered by sensors like cameras. The authors of [28] use online computer vision service (Microsoft Cognitive Service) for describing images and narrating the scene around the user, based on machine learning and deep learning. It reduces the development cost of this prototype. The systems [7, 29] use a laptop carried in a backpack, for processing dense data, specially coming from a camera. The system [25] employs a mobile device with two video cameras and a quad core processor for triggering vibration patterns. Other systems use high-performance embedded processors like a vision processor in [30], the Jetson TX1 in [16], the Jetson TX2 in [31], and the Raspberry Pi in [13, 14, 28].

In the remaining of this section, we are going to describe the methodology of some outstanding vision-based systems. Corners are detected in [32], and depth data of these features is use for estimating the position of obstacles. A neural network is used in [6] to classify six-line profiles extracted from the depth image for defining free space, obstacles, upstairs, and downstairs. The system described in [33, 34] works in any illumination condition, either indoor or outdoor spaces, employing an infrared camera, a stereo camera, and inertial sensors. For indoors, planar regions are computed considering normal vectors, followed by a region growing step and the selection of the plane with the largest distance to the camera. For outdoors, the ground surface is segmented using [35], a global 3D model for copying with depth uncertainty and superpixels for estimating confidence in dynamic environments.

In [7], the floor is extracted from the scene, initially using only depth data for computing vector normals and RANSAC for defining the equation of the plane. Next, color information is used for improving and expanding the initial estimation. Polygonal floor segmentation or watershed segmentation is selected automatically according to the type of scene the system is dealing with. The system [30] uses an ARM processor, and a fabricated vision processor for computing in real time a point cloud, normal vectors, and then, planes are classified using information from inertial sensors. These planes are improved with region growing which groups similar neighboring points. The system [29] estimates egomotion with visual odometry, identifies normals that are parallel to the gravity vector, computes a plane using RANSAC, builds a 3D voxel map, builds an occupancy 2D grid, and runs a path planning algorithm (D$*$ Lite algorithm) to guide the user to a target.

The authors of [16] demonstrated that transferring technology from autonomous vehicles to assistive tools for blind people is feasible. They build an extended version of the Stixel algorithm for working in indoor and outdoor environments, using the Jetson TX1 for processing data. The system [20] is wearable, performs at real time, segments the floor using also the Stixel World algorithm, and includes purposeful navigation to objects of interest such as empty chairs. It uses a linear classifier for classifying objects considering depth data. In [17], geographic and semantic information is provided to the user using sounds. In order to avoid obstacles, all Stixels within 9 m are represented as water droplets with loudness and phase difference related to the distance and direction of the detected obstacles. Vehicles are represented with horn sounds while pedestrians are represented with bell sounds for providing contextual information.

Our system uses a stereo camera with integrated inertial sensors (the camera computes its pose with a visual-inertial SLAM algorithm), a high-performance processing device, and haptic feedback for guiding the user to free space without collision. It works at real time by leveraging the capacity of the Jetson TX2 for parallel processing of dense data, with low power consumption, high portability, and without the need to connect to an external server for carrying out complex computations. The system segments the floor by evaluating normal vectors and orientation of the camera obtained from depth and inertial data, respectively. Next, RANSAC is applied for computing efficiently the equation of the supporting plane (floor). A global occupancy 2D grid is built in incremental way and a reactive navigation algorithm is executed for avoiding obstacles. The system is not limited to use only in previously fitted areas, since it does not use neither tags nor beacons. The main features of our system will be explained in more detail in the next section.

3. Methodology

The camera employed is the stereo camera ZED Mini, from Stereolabs. We evaluated the system with the camera located in the chest (see Figure 1(a)) and in the head (see Figure 1(b)), with the optical axis (z-axis) pointing down with an angle α with respect to the horizon. The height of the camera with respect to the floor is d. We get the camera orientation as the XYZ Euler Angles (roll, pitch, and yaw angles) and its position as a vector $[t_x, t_y, t_z]$. The camera was set to deliver depth and color images of 672 x 376 pixels, at a rate of 15 fps.

Algorithm of Figure 2 presents the main processes carried out by the proposed system for assisting blind people in wandering unknown environments in a safe way. We begin with a segmentation of the 3D point cloud, for identifying points that belong to the floor. For this process, we apply rotations using an initial estimation of orientation from the inertial sensors integrated to the camera. These rotations make the normals of points that belong to the floor parallel to the gravity vector. Next, we compute a 3D point cloud and normal vectors and apply two conditions (explained later) for an initial selection of points that belong to the floor. Since this estimation contains outliers, we employ RANSAC for defining the equation of the plane that best fits to data. We proceed to transform the whole point cloud in order to move the points that belong to the floor to the xz-plane ($y = 0$). Then, we concatenate the transformations given by the camera, apply a transformation to x- and z-components of the points, project the points to a 2D grid, and generate a global occupancy grid by evaluating the y-component of the points. Finally, we implement a reactive navigation algorithm that uses the global occupancy grid for creating commands in the haptic belt. Next, we give more details about these processes.

4. Initial Segmentation of the Floor

Initially, we estimate a 3D point cloud by processing depth data on GPU, using CUDA (Compute Unified Device Architecture) for implementing the kernels. Given the z-coordinate, we compute the x- and y-coordinates using the following equations:

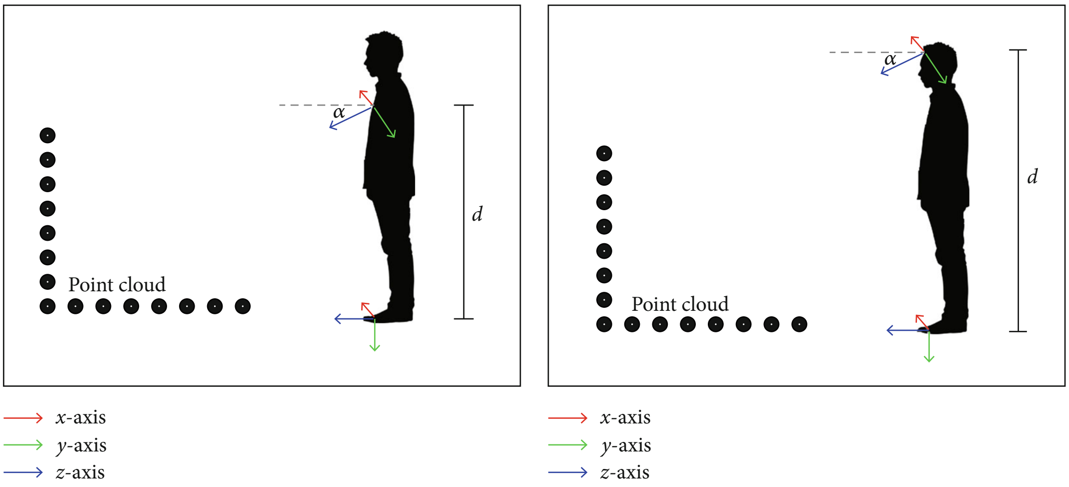
$$x = \frac{u - c_x}{f_{\mathrm{x}}} z, \tag{1}$$

$$y = \frac{v - c_y}{f_y} z, \tag{2}$$

where (x, y, z) is the Cartesian coordinate for a 3D point, (u, v) is a coordinate in the image plane, (c_x, c_y) is the coordinate of the principal point of the image and (f_x, f_y) is the focal length in x and y. Both, the principal point and the focal length are included in the intrinsic parameters defined in the calibration parameters of the ZED Mini. These parameters are stored in constant memory while 3D coordinates (x, y, z) with respect to the current camera pose are stored in texture memory.

The first transformation is a composite rotation around the x- and z-axis in order to have the y-axis parallel to gravity vector. Equation (3) presents this transformation while Figure 3 depicts it graphically.

$$T_{w_{ini} c_{ini}} = \begin{bmatrix} Rot(x,-\alpha) Rot(z, \beta) & [000]' \\ [000] & 1 \end{bmatrix}. \tag{3}$$



(a) (b)

Figure 1: Camera located (a) in the chest and (b) in the head of the user. The camera is pointing down, defining an angle α with respect to the horizon (upper coordinate system). Its height with respect the floor is d. We want to move the points associated to the floor to the xz-plane of the lower coordinate system so we can differentiate obstacles from floor by evaluating the height of the points.

```
Algorithm: Pseudocode for assisting blind people to wander known environments
Data: RGB data, depth data and camera pose data
Result: Global Occupancy 2D grid and commands through the haptic belt
while system is running do
    get data from camera;
    apply rotations Rot $(x, -\alpha)$*Rot $(z, \beta)$;
    build a local point cloud and normal vectors;
    apply the two conditions proposed for initial segmentationof the point cloud;
    if activePoints>th_aPoints then
        apply RANSAC for computing the equation of the plane;
    end
    apply transformation for moving points that belong to the floor to the$xz$-plane;
    concatenate current camera transformation forgetting global coordinates;
    build/expand the occupancy 2D grid;
    run the reactive navigation algorithm;
    generate commands through the haptic belt;
end
```

Figure 2: Pseudocode for assisting blind people to wander unknown environments.

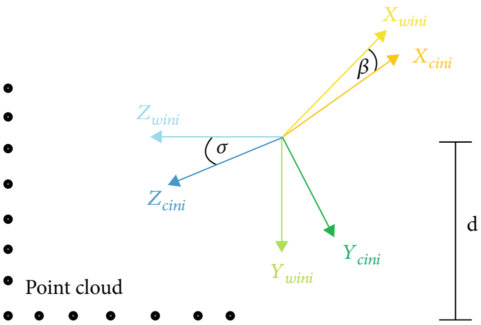


Figure 3: Initial transformation of composite rotations around x- and z-axes. The height of the camera with respect to the floor is d and is initially unknown.

Once the points are referenced to w_{ini}, normal vectors are computed on GPU, considering 3D points located on the right and down the current pixel, as

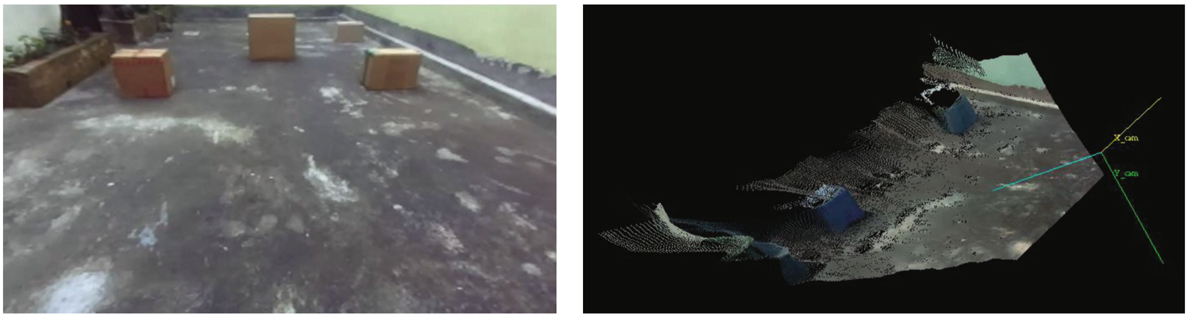
$$n_{w_{ini}}(u,v) = \left(P_{w_{ini}}(u+1,v) - P_{w_{ini}}(u,v)\right) \times \left(P_{w_{ini}}(u,v+1) - P_{w_{ini}}(u,v)\right). \tag{4}$$

The unit normal vectors $\widehat{n}_{w_{ini}}(u,v)$ are computed as

$$\widehat{n}_{w_{ini}}(u,v) = \frac{n_{w_{ini}}(u,v)}{\left|n_{w_{ini}}(u,v)\right|}. \tag{5}$$

Both, the 3D points referenced to w_{ini} and the unit normal vectors are stored in texture memory on GPU. Points can be drawn with color of the RGB images (see Figures 4(a) and 4(b)) or with color associated to unit normal vectors (see Figure 5), for example, in green if a point has a normal with high component in y-direction.

Now that the y-axis (y_{wini}) is parallel to the gravity vector, we define two thresholds, Th_y and Th_{ny}. Points with a y -component greater or equal to the threshold Th_y and with



(a) (b)

Figure 4: (a) RGB image of a paved courtyard with boxes used as obstacles. (b) Point cloud with color of the RGB image. The camera is represented as a coordinate system with the current camera pose.

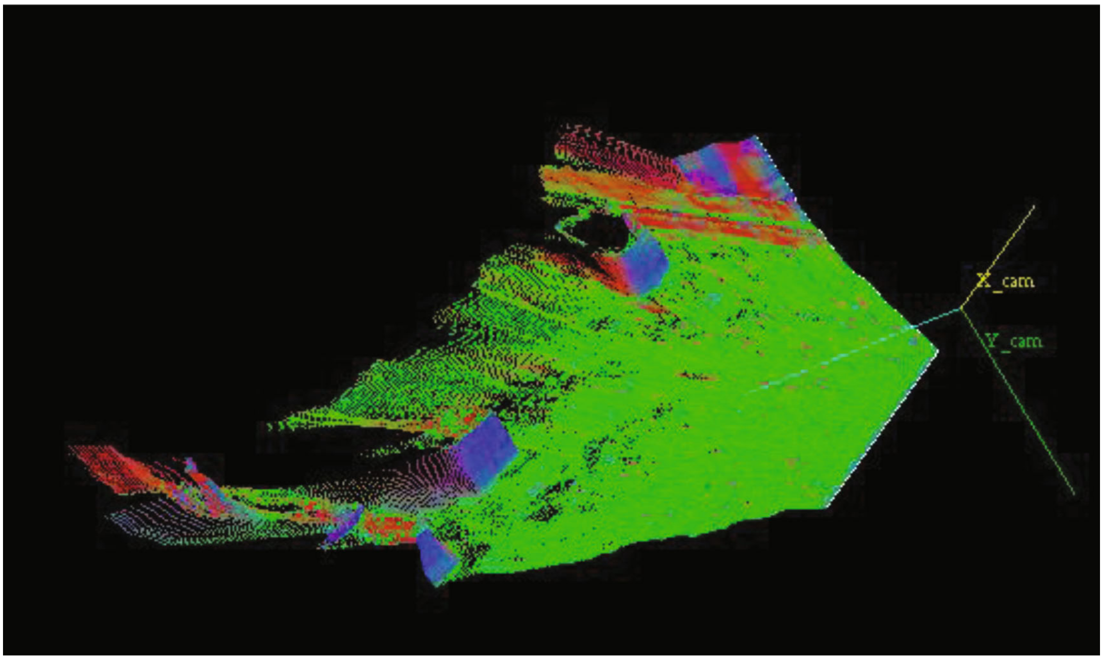


Figure 5: Point cloud of a paved courtyard with boxes used as obstacles. The color is associated to unit normals, computed after applying the rotation defined in Equation (3). The camera is represented as a coordinate system with the current camera pose.

normals parallel to the gravity vector ($\hat{n}.y \geq Th_{n_y}$) are identified and stored in an array, in global memory of GPU. Summarising, the conditions for selecting these points are listed next.

(i) The unit normal vector, $\hat{n}$, must be parallel to the gravity vector, so its y-component must be greater or equal to 0.80: $\hat{n}.y \geq Th_{n_y} = 0.80$

(ii) The 3D point must have a distance in y-direction over a threshold, so its y-component must be greater or equal to a threshold: $P.y \geq Th_{y_{chest}} = 0.80\text{m}$ for the camera in the chest and $P.y \geq Th_{y_{head}} = 1.00\text{m}$ for the camera in the head.

Figure 6 shows the points that fulfil these conditions, for the point cloud presented in Figure 5. The number of points is denoted as *activePoints*.

Points that belong to the floor and to parallel surfaces close to it are selected in this initial segmentation. Considering this, RANSAC is used to compute the equation of the plane that best fits to points associated to the floor since it is robust to outliers.

5. Estimation of the Supporting Plane Using RANSAC

The equation of a plane in 3D space can be defined with a normal vector $\boldsymbol{n}$ and a known point in the plane P_1. Let P be any point in the plane, so the vector $r = P - P_1$, which is defined as

$$r = P - P_1 = (x - x_1, y - y_1, z - z_1), \tag{6}$$

which is lying on the plane. Since this vector and the normal vector $n = (a, b, c)$ are perpendicular each other, the dot product is zero:

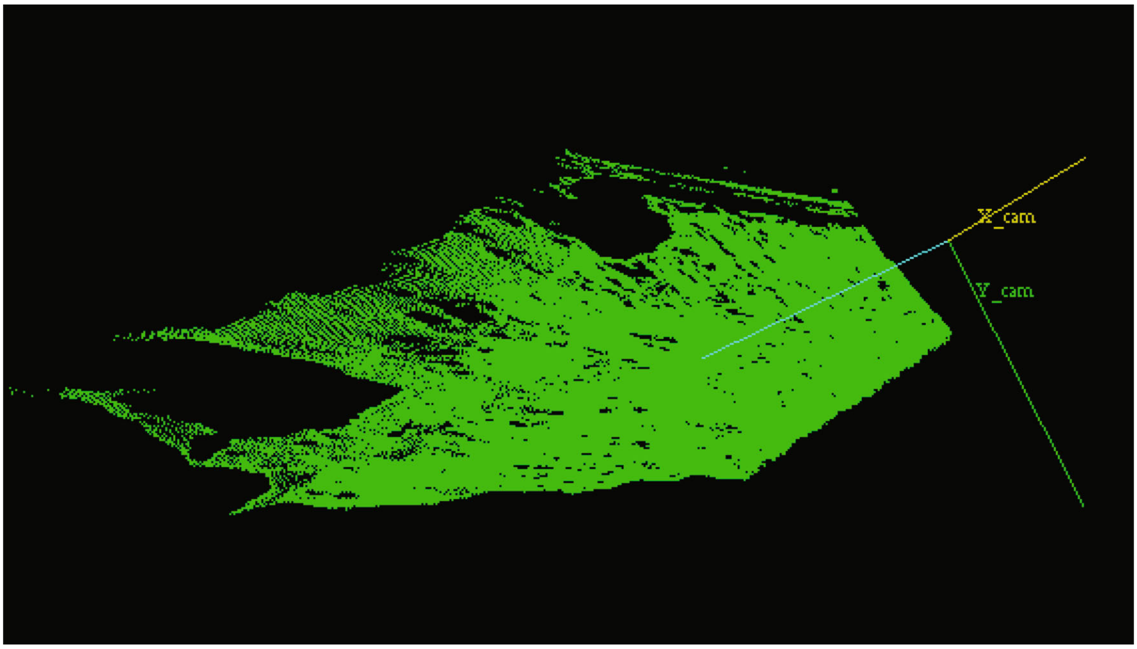


FIGURE 6: 3D points that fulfil the conditions for its unit normal vector, $\hat{n}.y \geq Th_{n_y} = 0.80$, and for its y-component, $P.y \geq Th_{y_{chest}} = 0.80$m (camera located in the chest). Its number is denoted as *activePoints*. The camera is represented as a coordinate system with the current camera pose.

$$n \bullet r = 0. \tag{7}$$

By calculating the dot product, we get

$$(a, b, c) \bullet (x - x_1, y - y_1, z - z_1) = 0, \tag{8}$$

$$ax + by + cz + d = 0, \tag{9}$$

with $d = -(ax_1 + by_1 + cz_1)$. Any point P that belongs to the plane must fulfil this equation. Now, we select three points randomly, P_1, P_2, P_3, (see Figure 7), compute two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, and a normal vector $\boldsymbol{n}$ with the cross product. This process is expressed in Equations (10) and (11).

$$v_1 = P_2 - P_1 v_2 = P_3 - P_1, \tag{10}$$

$$n = v_1 \times v_2. \tag{11}$$

We normalize the normal vector $\boldsymbol{n}$ for getting a unit normal vector using the following equation:

$$\hat{n} = \left(\frac{n.x}{|n|}, \frac{n.y}{|n|}, \frac{n.z}{|n|} \right). \tag{12}$$

Now, we compute the d value of Equation (9), considering a known point, for example P_1, and the unit normal vector $\hat{n}$.

$$d = -(\hat{n}.x * P_1.x + \hat{n}.y * P_1.y + \hat{n}.z * P_1.z). \tag{13}$$

Finally, for each point P_m of the initial segmentation, we verify if it fulfils the equation of the plane 9 by computing $diff_m$ as is presented next.

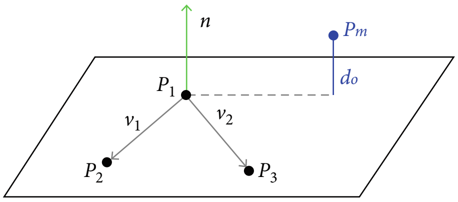


FIGURE 7: Normal $\boldsymbol{n}$ (green vector) and orthogonal distance $d_o(P_m)$ from point P_m to plane (blue segment). It is computed for each point P_m of the initial segmentation and with three points, randomly selected, using RANSAC.

$$diff_m = \hat{n}.x * P_m.x + \hat{n}.y * P_m.y + \hat{n}.z * P_m.z + d. \tag{14}$$

$diff_m$ is the orthogonal distance $d_o(P_m)$ from point P_m to plane. If abs$(diff_m)$ is close to zero (under a threshold, $th_{RANSAC} = 0.06$ m), then the point P_m belongs to the plane and a counter increases in one. After k iterations, the selected model i is the one with the highest counter. This model is defined by $(\hat{n}_i, d_i)$, where $\hat{n}_i$ is the unit normal vector of the plane and d_i is the distance to the origin, for model i. The random search on the space of solutions is done taking advantage of parallel processing on GPU. The number of iterations k for avoiding possible outliers in the selection of the points is

$$k = \frac{\log(1 - q)}{\log(1 - w^{n_r})}, \tag{15}$$

where the parameter q corresponds to the probability that a good model is computed, w represents the proportion of inliers with respect to the total number of points, and n_r is

Algorithm: Pseudocode for computing the equation of the plane that best fits input data, using RANSAC

```
 Data: Point cloud resulting from an initial segmentation of the floor (with outliers)
 Result: Parameters $\hat{n}_i$ and $d_i$ of the equation of the plane that best fits input data
Compute maximum number of iterations $k$, Equation (15);
for ($i = 1$ to $k$) do
    Select 3 points randomly from input data;
    Build vectors $v_1$ and $v_2$, Equation (10);
    Compute unit normal vector $\hat{n}_i$ , Equation (11);
    Compute $d_i$ using $\hat{n}_i$ and a point from the 3 points previously selected, Equation (13);
    for (all points $P_m$ in the input data) do
        Compute distance $diff_m$ from point $P_m$ to plane $i$, Equation (14);
        if ($diff_m < Th_{RANSAC}$) then
            $\text{counter}_i += 1$;
        end
    end
end
Find $i$ for max ($\text{counter}_i$);
Define model $i$ ($\hat{n}_i$ , $d_i$) as the model that best fits to data;
```

Figure 8: Pseudocode for computing the equation of the plane that best fits input data, using RANSAC.

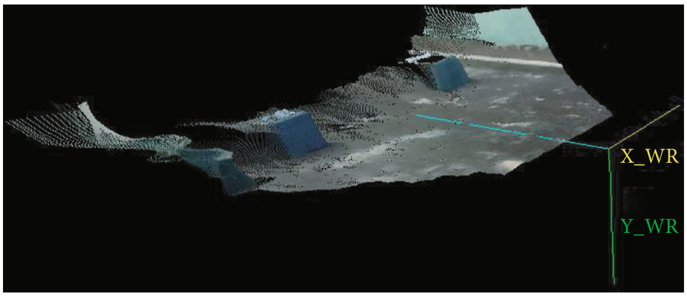


Figure 9: Point cloud transformed in order to move points that belong to the floor to the xz-plane, based on the parameters of the plane computed with RANSAC.

the minimum number of points that are required to compute a model. The numerical values of these parameters are $q = 99.9\%$, $w = 40\%$, and $n_r = 3$, resulting $k = 104$ iterations. Algorithm of Figure 8 summarises the steps for computing this model.

Once we have the equation of the plane computed with RANSAC, we apply the transformation defined in Equation (16) to the whole point cloud in order to refine the initial rotations for getting normals of points that belong to the floor parallel to the gravity vector and move them to the xz -plane, as can be seen in the example of Figure 9.

$$T_{w_R w_{ini}} = \begin{bmatrix} Rot(x,-\alpha_R)Rot(z,\beta_R) & [000]' \\ [000] & 1 \end{bmatrix} * \begin{bmatrix} eye(3,3) & [d * n\wedge.x, d * n\wedge.y, d * n\wedge.z]' \\ [000] & 1 \end{bmatrix}. \quad (16)$$

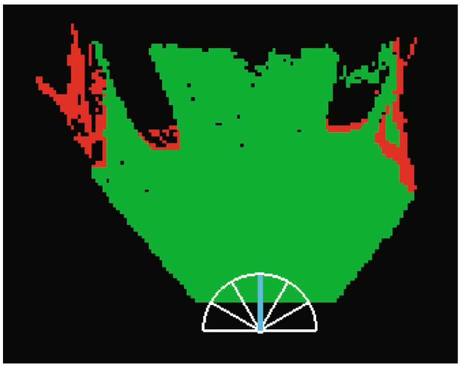


Figure 10: Occupancy grid built for the point cloud presented in Figure 9. Occupied cells are drawn in red while free cells are drawn in green. Cyan line represents the orientation of the user while the semicircle is a region where obstacles activate the vibrating motors of the belt.

where α_R and β_R are the angles that make the normal vector $\hat{n}$ of the computed plane parallel to the y-axis (Y_{wR}) and eye represents the identity matrix. The transformation on the right of Equation (16) corresponds to a translation of d in direction of the unit normal vector $\hat{n}$. For avoiding wrong estimations of the plane, RANSAC is applied only if the number of points resulting from the initial segmentation (*activePoints*) is greater than a threshold ($th_{aPoints} = 90000$). A low value of *activePoints* happens when the user is close to an obstacle and the floor is not visible. In these cases, the equation of the plane that was computed with enough *activePoints* in the closest pose to the current one is used for applying the transformations.

6. Building the Occupancy 2D Grid

The occupancy 2D grid is made up of free and occupied cells that define regions where the user can navigate without colliding with obstacles. A global grid stores data from all the poses where the user has been by merging data considering positional tracking of the camera. Until this point, we have applied the transformation $T_{w_{ini}c_{ini}}$ defined in Equation (3) and $T_{w_R w_{ini}}$ defined in Equation (16). Now, we are going to consider the transformation $T_{c_{ini}c_n}$ between the initial frame c_{ini} and a posterior frame c_n as is shown next.

$$T_{w_R c_n} = T_{w_R w_{ini}} T_{w_{ini} c_{ini}} T_{c_{ini} c_n}, \text{for } n = 1, 2, 3, \cdots, \qquad (17)$$

where

$$T_{c_{ini} c_n} = \left(T_{w c_{ini}}\right)^{-1} T_{w c_n}. \qquad (18)$$

$T_{w_R c_n}$ is computed and stored in constant memory on GPU in each camera pose using the transformation T_{wc_n} given by the camera. From the transformation $T_{w_R c_n}$, we get the position of the user in the 2D grid as $x_{user} = T_{w_R c_n}(1, 4)$, $z_{user} = T_{w_R c_n}(3, 4)$ and the orientation of the user as $atan2(T_{w_R c_n}(3, 3), T_{w_R c_n}(1, 3))$. Now, we use this pose in 2D for modifying the x- and z-components of the point cloud obtained after transformation of Equation (16), leaving the y-component unchanged. Next, we defined a threshold in the y-component of 0.20 m (height is the notation used for the y-component hereinafter) of the transformed point cloud. If the absolute value of the height of a point is over the threshold, it is classified as obstacle point; otherwise, it is classified as free point. Next, we project the points to the occupancy 2D grid, which means we evaluate only the x- and z-coordinates of each point for defining the cell where it is located. The number of free points P_f and obstacle points P_o projected into a cell defines if that cell is occupied of free, according to the next rules. If $P_f > P_o \wedge P_f > th_{P_f} = 200$, then the cell is considered free. Otherwise, if $P_o > P_f \wedge P_o > th_{P_o} = 20$ (for avoiding outliers), then the cell is considered occupied. th_{P_f} and th_{P_o} are thresholds in the number of free and obstacle points, respectively. The resulting occupancy grid for the point cloud presented in Figure 9 is shown in Figure 10. The size of the grid is up to 20×20 m, and the size of the squared cells is 0.05 m.

With the occupancy 2D grid, the system can alert the user about a possible collision and the bearing of a dangerous obstacle when it enters inside a predefined region around the user. This warning region is defined in order to give the user enough time to move to free space and avoid collisions. Besides, the system has the capacity to detect obstacles at different heights, even over the user's waist, in which a white cane is not able to do, for example, hanging lamps, branches of trees, suspended decorations, among others. Finally, since the grid is built incrementally by processing data taken from different poses, more complex tasks can be carried out, such as path planning for guiding the user to objects of interest seen previously in the environment.

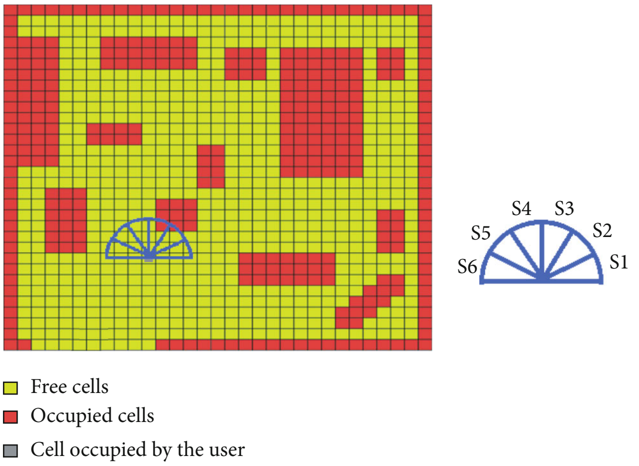


Figure 11: Semicircle with six conical sections for detecting and avoiding obstacles in unknown environments. The pairs S1-S2, S3-S4, and S5-S6 define the direction that the user can traverse by generating vibrations.

7. Reactive Navigation

The user must be informed if an obstacle is close, with enough time to react and move in order to avoid it. For tackling this problem, we use the occupancy 2D grid, define a semicircle in front of the user (with a radius r_o), and divide the semicircle in six equal conical sections (cones with an opening of 30°) for identifying the bearing of the obstacles. We compute in GPU, for each occupied cell, its orientation and distance to the user. Occupied cells inside the semicircle are considered dangerous. A section with occupied cells indicates that there is an obstacle in that direction, and the user cannot traverse it.

For avoiding obstacles, we evaluate the six sections and create vibration patterns according to the next rules. If S1 and/or S2 have/has occupied cells, then the motor on the right will vibrate. If S3 and/or S4 have/has occupied cells, then the frontal motor will vibrate. If S5 and/or S6 have/has occupied cells then the motor on the left will vibrate. When one or both sections of the previous pairs have/has occupied cells, a path cannot be defined through this pair. Generalizing, the pairs S1-S2, S3-S4, and S5-S6 define the direction that the user can traverse by generating vibrations. In the example of Figure 11, the right and frontal motors vibrate since pairs S1-S2 and S3-S4 contain occupied cells.

8. Haptic Feedback

Since the user is not able to get information of the environment using his or her visual sense, we use a vision-based system for creating an occupancy 2D grid and for generating vibration patterns that require of the touch sense of the user

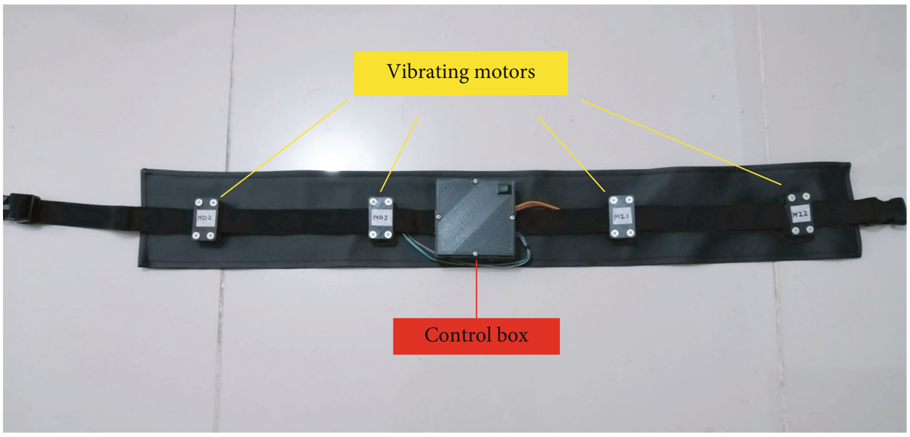


Figure 12: Belt with four vibrating motors and the control box (only three motors are used for avoiding obstacles). The belt is located in the user's waist. The communication is through USB connection.

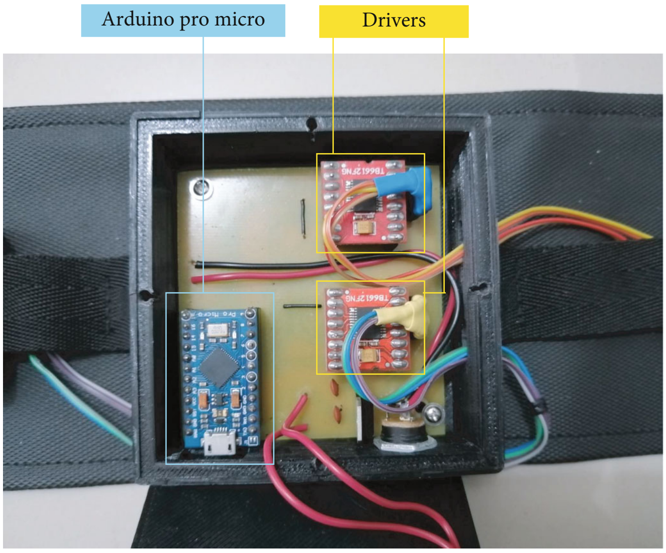


Figure 13: Components of the control box. It is made up of an Arduino Pro Micro and two motor drivers. It is fed by the main battery of the system.

for decoding this information and moving according to a reactive navigation algorithm that avoids collisions. These warnings are represented by unique vibration patterns generated by an haptic belt that the user wears in the waist. This device that is shown in Figure 12 guides the user to walkable space with haptic commands that indicate a turn to one side, go ahead or stop, and scan. A lateral motor vibrating indicates that an obstacle has been detected in the respective side within a radius r_o. The frontal motor vibrating indicates that there is an obstacle in front of the user within a radius r_o. The motor on the back is not used for wandering mode. When the three motors vibrate, a path is not available neither in lateral sides nor in front of the user, so he or she should stop and scan an optional path.

The Arduino Pro Micro generates a PWM signal for controlling the activation of the vibrating motors through drivers with reference 6612FNG. The Arduino Pro Micro and the Jetson TX2 developer kit communicate using a USB connection. The belt is fed from the main battery. The components of the control box are labelled in Figure 13.

Table 1: Results in accuracy in haptic perception.

Participant	Gender	Sighted	Accuracy
1	F	Yes	80.00%
2	M	Yes	95.00%
3	M	Yes	95.00%
4	M	Yes	85.00%
5	F	Yes	92.50%
6	M	Yes	87.50%
7	F	Yes	90.00%
8	F	No	100.00%
9	F	No	90.00%
10	F	No	92.50%
Average in accuracy			90.75%

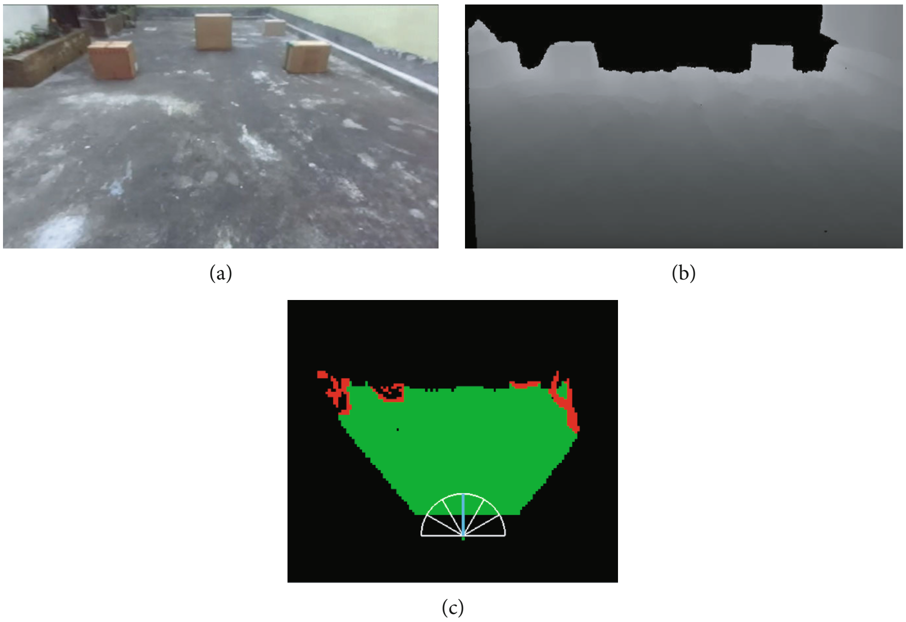


Figure 14: (a) RGB and (b) depth images of a paved courtyard from the first frame. (c) Occupancy 2D grid for this frame. The camera is located in the chest. The angle of the camera α with respect to the horizon is approximately 30°, the size of the squared cells is 0.05 m, the horizontal field of view of the camera is 80°, the maximum depth is 3.00 m, and the radio of the semicircle for activating the motors is 0.80 m.

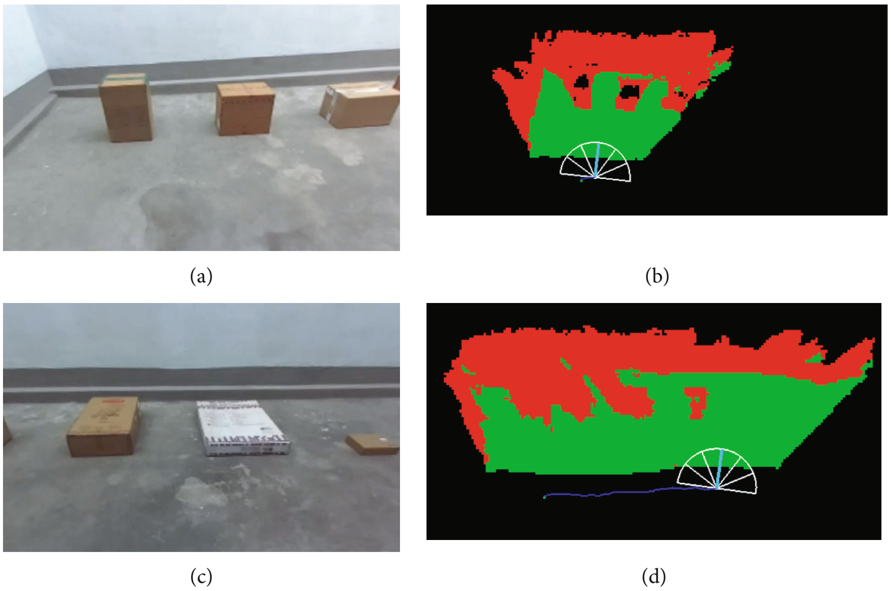
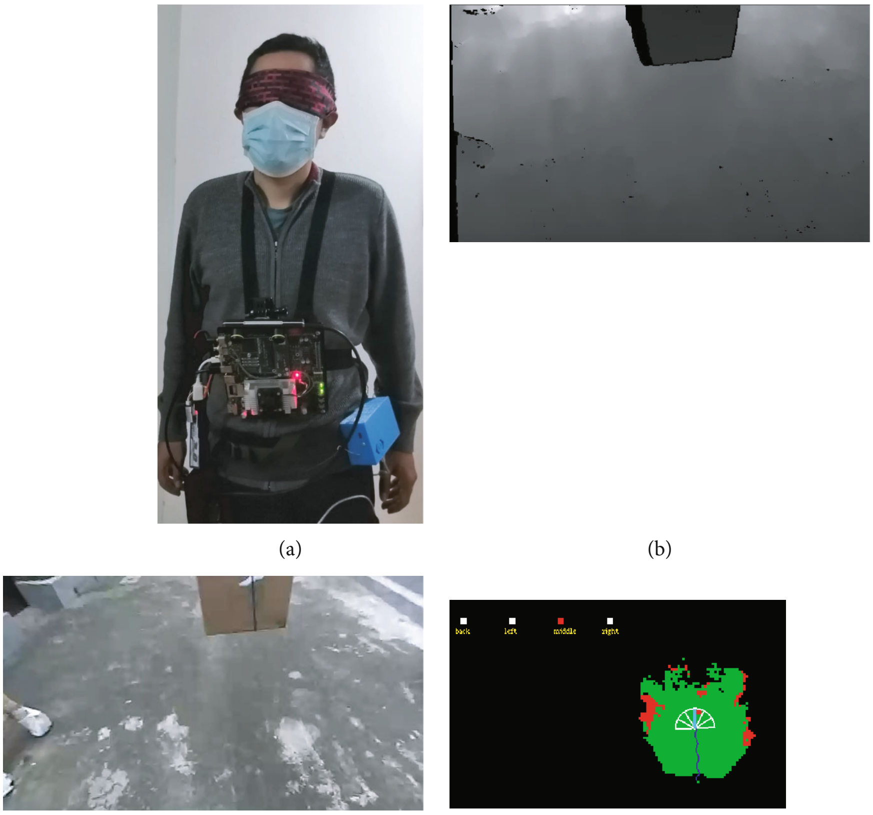


Figure 15: Six boxes separated each other 0.60 m and ordered in descendent height from left to right. (a) RGB image focusing the initial three boxes and (b) occupancy 2D grid for a short lateral displacement. (c) RGB image that shows the remaining boxes on the right and (d) occupancy 2D grid for a longer lateral displacement. The motion is lateral but always facing the boxes. The angle of the camera α with respect to the horizon is approximately 30°, and the size of the squared cells is 0.05 m.

9. Results

In this section, we are going to describe the experiments carried out for estimating the accuracy in haptic perception, the optimal distance to obstacles for activating the motors, the range of measurement in depth, the minimum height of objects to be considered obstacles, the maximum height of hanging obstacles detected by the camera when it is located in the chest and in the head, the effectiveness to detect dynamic obstacles, the average walking speed, the number of collisions in a period of time, the easiness to follow instructions, and the portability, among other parameters that define the performance of the system.

For estimating the accuracy in haptic perception, ten participants of different ages and gender, sighted and blind, wore the haptic belt in his or her waist. Three sets of ten vibration



(a) (b)

(c) (d)

FIGURE 16: Experiment carried out with the camera located in the chest and with a hanging obstacle. (a) Location of the camera. (b) Depth and (c) color images of a box hanging at a height of 0.75 m. (d) Occupancy grid with the projection of the box and with activation of the frontal motor.

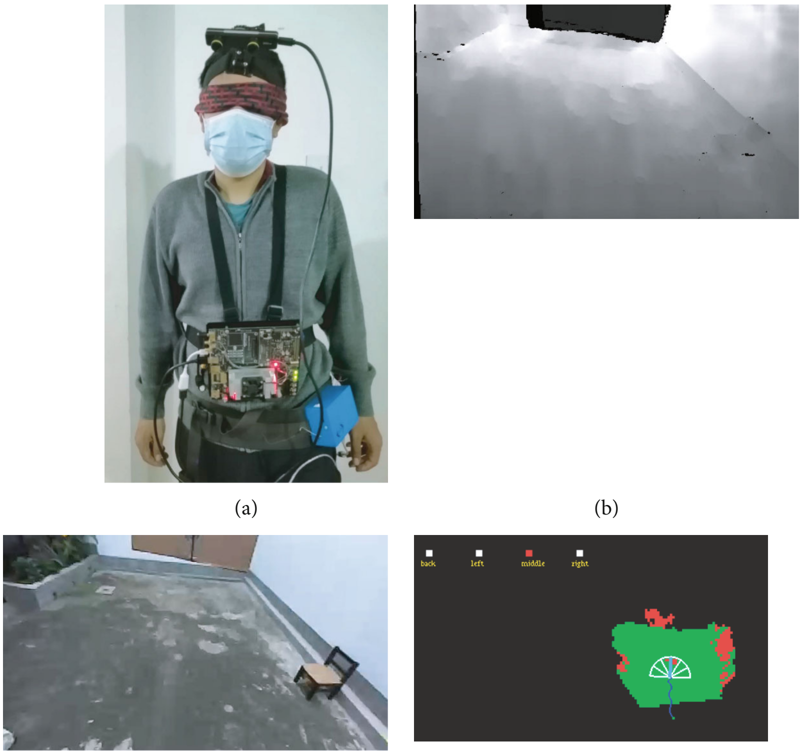
patterns were generated randomly. Each pattern is one of the eight possible combinations obtained with three vibration motors (left, frontal, and right motors). In the first set, the participant was informed about the real combination. In the remaining two sets, the participant was asked about the state of the three motors (pattern). Table 1 summarises the results. Note that the average accuracy in haptic perception is 90.75%, which is appropriate for guiding blind people to walkable space in an unknown environment.

The environment where the next experiments were carried out is a paved courtyard of rectangular shape, with approximate size of 9 m long and 5 m width. It is an outdoor place surrounded by high walls, two small gardens, and some windows. In Figure 14(a), most of this space can be seen while in Figure 14(b), the corresponding depth image is presented. In Figure 14(c), the occupancy 2D grid for this frame is shown. The whole grid can have 400 × 400 cells that cover 20 × 20 m (400 m^2). Each squared cell has a size of 0.05 m.

From this frame, the horizontal field of view of the camera *hfov* was measured and corresponds to 80°. The maximum measurement in depth was set to 3.00 m (see Figures 14(b) and 14(c)) since smaller distances are more accurate than longer ones. The minimum depth was set to 0.40 m for avoiding outliers in the occupancy grid (for example, depth data associated to user's hands or to white canes). Note in Figure 14(c) that part of the garden, the right wall and two boxes, are in the range of measurement and are classified as obstacles. The radio of the semicircle r_o that defines the space where an obstacle is considered dangerous was set to 0.80 m. This value provides to the user enough time to avoid obstacles and, at the same time, allows him or her to pass across small walkable spaces.

For estimating the minimum height of objects detected by the system as obstacles $\min_h$, we put six boxes of different heights, and the user moved in lateral way facing the boxes. The heights of the boxes from left to right are 0.53 m, 0.38 m, 0.30 m, 0.20 m, 0.11 m, and 0.07 m. Figures 15(a) and 15(c) show these boxes. Figures 15(b) and 15(d) present the occupancy 2D grid for a short and long lateral displacement, respectively. Note that the four initial boxes on the left are detected by the system as obstacles (represented as occupied cells) while the remaining two boxes are not detected (represented as free cells). According to this experiment, the minimum height is 0.20 m.

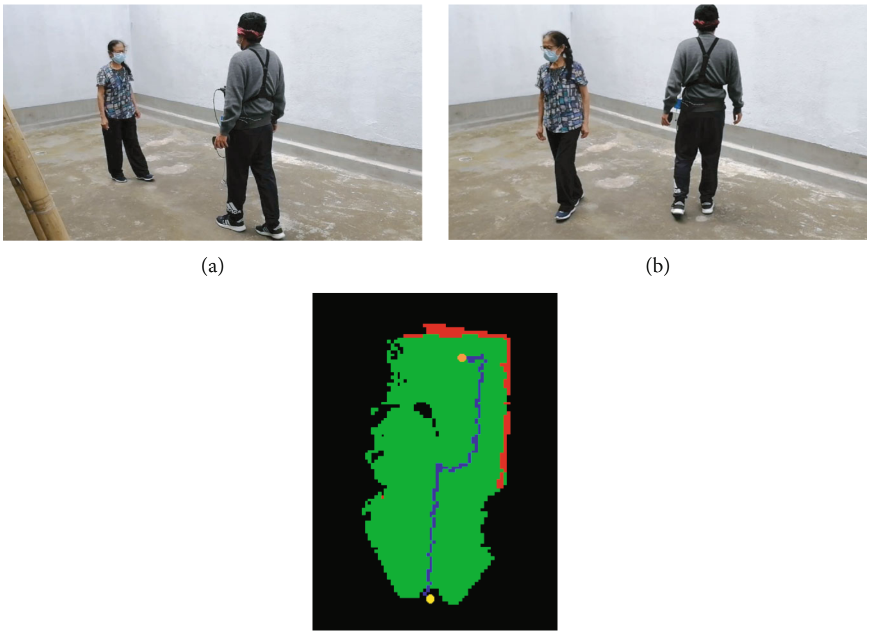
Now, we present the experiments carried out for determining the maximum height that a hanging obstacle can have for been detected by the system. A box was hanged at different heights, from 0.25 m to 1.5 m, with variations of 0.25 m. A blindfolded participant moved following a straight path until the system indicated the presence of an obstacle through the haptic belt or until a researcher warned the closeness of the obstacle (when the box was out of the field of view



(a) (b)

(c) (d)

Figure 17: Experiment carried out with the camera located in the head and with a hanging obstacle. (a) Location of the camera. (b) Depth and (c) color images of a box hanging at a height of 1.50 m. (d) Occupancy grid with the projection of the box and with activation of the frontal motor.



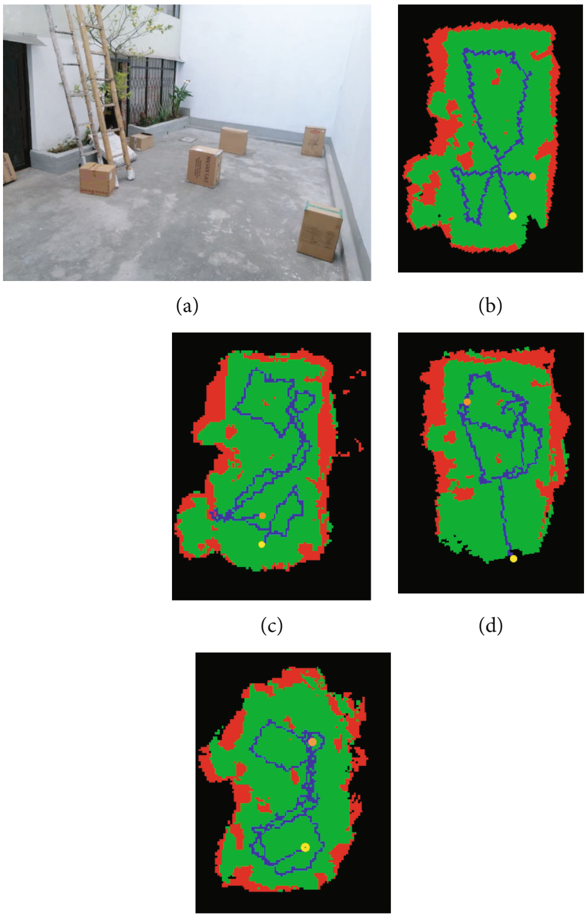
(a) (b)

(c)

Figure 18: (a) Person walking to obstruct the straight path of the blindfolded participant. (b) Moment when the obstruction occurs, and the blindfolded participant avoids the obstacle. (c) Trajectory followed by the participant. The yellow point is the starting location and the orange point is the final location.

Table 2: Parameters of performance and for configuring the system.

Parameter	Description	Value
cam_{res}, cam_{fps}	Camera resolution and frames per second	672 x 376 pix., 15 fps
$depth_{range}$	Range of measurement in depth	[0.40 m; 3.00 m]
$hfov$	Horizontal field of view of measurements	80°
th_{ny}	Threshold in y-component of unit normal vector	0.80
th_{y_chest}	Threshold in y-component of points (c. chest)	0.80 m
th_{y_head}	Threshold in y-component of points (c. head)	1.00 m
k, th_{RANSAC}	Maximum iterations and threshold for RANSAC	104 iter., 0.06 m
$size_{gridx}$, $size_{gridz}$	Size of the grid in x and z direction	20 m, 20 m
$size_{cell}$	Size of one squared cell	0.05 m
r_o	Radio of the semicircle for detecting obstacles	0.80 m
$\min_h$	Minimum height for detecting an obstacle	0.20 m
$maxHang_{Cam\_chest}$	Maximum height of a hanging obstacle (c. chest)	0.75 m
$maxHang_{Cam\_head}$	Maximum height of a hanging obstacle (c. head)	1.50 m


(a)

(b)

(c)

(d)

(e)

Figure 19: (a) Place where the experiments were carried out. (b) Trajectory followed by a blindfolded and (c) a blind participant with the camera located in the chest. (d) Trajectory followed by a blindfolded and (e) a blind participant with the camera located in the head. Each experiment lasts 3 minutes. The blue line represents the trajectory. The yellow point is the starting location and the orange point is the final location.

Table 3: Summary of performance in wandering an unknown environment with the camera in the chest and in the head of the user. F: female, M: male.

Participant	Blind or blindfolded	Gender	Distance (m)		Average speed (m/s)		Number of collisions	
			Chest	Head	Chest	Head	Chest	Head
1	Blindfolded	F	26.00	27.32	0.14	0.15	0	0
2	Blindfolded	M	27.08	27.56	0.15	0.15	0	1
3	Blindfolded	M	29.20	28.63	0.16	0.16	1	1
4	Blindfolded	M	30.55	31.44	0.17	0.17	0	0
5	Blindfolded	F	27.23	28.14	0.15	0.16	1	0
6	Blindfolded	M	29.39	30.21	0.16	0.17	0	1
7	Blindfolded	F	29.12	27.87	0.16	0.15	0	0
8	Blind	F	36.42	34.68	0.20	0.19	1	0
9	Blind	F	41.80	37.17	0.23	0.21	1	2
10	Blind	F	32.84	33.60	0.18	0.19	0	1

Table 4: Easiness to follow instructions and portability, with the camera in the chest and in the head of the user. L: low, M: middle, H: high.

	Camera	Participant									
		1	2	3	4	5	6	7	8	9	10
Easiness to follow instructions	Chest	H	H	H	H	**M**	H	H	H	H	H
	Head	H	H	H	H	**M**	H	H	H	H	H
Portability	Chest	**M**	H	H	H	**M**	H	**M**	**M**	**M**	H
	Head	**M**	H	H	H	**M**	H	**M**	**M**	**M**	H

of the camera). The system detected the box up to 0.75 m height. The box hanging at a height equal to or greater than 1.00 m was not detected because of the low height of the camera (located in the chest) and its inclination with respect to the horizon that focused its field of view mostly to the floor. Figure 16(a) presents the camera located in the chest, while Figures 16(b) and 16(c) present the depth and color images, respectively. Note that the hanging box (height of 0.75 cm) is projected to the 2D occupancy grid, and the frontal motor is activated when the occupied cells are inside the warning region (see Figure 16(d)).

Considering this limitation, we evaluated the option to locate the camera at the level of the head. For this configuration, the system detected obstacles up to 1.5 m. Figure 17(a) shows the camera located in the head. Figures 17(b)–17(d) are the depth image, color image, and occupancy grid, respectively. Note that at that height of the box, occupied cells are also generated in the grid. This is an important advantage of using the camera at the level of the head and one of the main differences from navigating with a white cane.

In this experiment, we evaluated the performance of the system in presence of a dynamic obstacle. The blindfolded participant moved following a straight path until a dynamic obstacle (a person) that moved in the opposite direction obstructed his path. The participant in that moment received a vibration pattern and avoided the person. The radio of the semicircle for detecting obstacles is set to 0.80 m. Figures 18(a) and 18(b) show the participant and a person that obstructs his path, and Figure 18(c) presents the occupancy grid with the trajectory followed by the participant. Note the abrupt change in direction of the participant when the dynamic obstacle (person) enters into the wandering region, and the belt is activated. The same experiment was developed with the camera located in the head, getting similar results.

Table 2 presents these and other parameters of performance and for configuring the system, together with a short description and their values.

The next experiment consists in wandering an unknown environment (see Figure 19(a)) by blind and blindfolded people. We evaluated the performance of the system with the camera initially located in the chest and later in the head, pointing to the floor, with an angle α with respect to the horizon (see Figures 1, 16(a), and 17(a)). There are several boxes of different heights but over 0.20 m height. The participant is trained during five minutes before the experiment is executed. Then, the participant is located in a starting point without knowing about the configuration of the environment. The person begins to move using the vibration patterns of the belt for avoiding collisions with obstacles. This process lasts three minutes and is under the supervision of a researcher. We measured the average walking speed and the number of collisions, and we asked the user for information about the easiness for carrying the system and for avoiding obstacles by following haptic instructions. Ten participants of different ages and gender, blindfolded and blind, wandered this space wearing the system in both configurations (camera in the chest and in the head). Figures 19(b) and 19(c) correspond to the occupancy grid and trajectory followed by a

Table 5: Size and weight of each component of the system.

Component	Size (cm)	Weight (g)
Processor	17x17x5	469
Camera	12.5x2.7x3	63
Control box of belt	11.7x8.1x5.2	180
Belt	Adjustable long: 10 (width)	100
Battery	13.7x4.3x1.7	213
Harness	Adjustable along the chest	100
	Total weight	1125

blindfolded and a blind participant, respectively, with the camera in the chest while Figures 19(d) and 19(e) are for a blindfolded and a blind participant, respectively, but in this case with the camera in the head. Similar results with respect to average speed, number of collisions, easiness to follow instructions, and portability were obtained with both configurations, as can be seen in Tables 3 and 4.

All participants said that after five minutes of the initial training, they felt more comfortable and more confident, and it was reflected in a higher walking speed during the final experiment. The average walking speed considering the ten (10) participants was 0.17 m/s for both locations of the camera (in the chest and in the head). Note that blind participants had a higher walking speed than blindfolded ones since the former participants are used to tackling these situations. Moreover, in 55% of the experiments, the participants (blind and blindfolded, with the camera in the chest or in the head) did not have collisions. This fact validates the radio of the warning region, which gives enough time to the participants to avoid an obstacle. The few collisions were due to distractions since the vibration patterns require to pay complete attention all the time. A blindfolded participant told us that she was thinking about a different subject during few seconds and that produced a collision with an obstacle. This participant evaluated the easiness to follow instructions as "middle." The remaining participants (90%) said that the vibration patterns are intuitive, easy to identify and to understand. This opinion agrees with the high accuracy in haptic perception, defined previously in Table 1. A blind participant told us that in a previous experiment with a system that used ultrasonic sensors and audio feedback, she felt overloaded with voice commands, which was confusing and stressing. However, due to the simplicity and intuitiveness of our haptic feedback, the instructions were easy to follow, reducing the stress, the mental workload, generating a low number of collisions, a short period of time for training, and without interfering with the sound of the environment.

For the next assessment of portability, we present in Table 5 the size and weight of each component of the system. Note that the processor is the heaviest component.

The female participants felt a little uncomfortable with the size, weight, and location of the processor, even when it was located under the chest. In contrast, male participants claimed to feel comfortable wearing the system. A possible solution to this problem is presented in "Discussion." The camera in the head does not cause discomfort to any of the participants due to its low weight and size. However, in one experiment, the cable was short and affected the natural position of the head. This problem was solved with a longer cable. Some experiments can be seen in the Video S1 in the supplementary material.

Jetpack SDK 4.3 was used to flash the Jetson TX2 developer kit. OpenCV 4.1 and CUDA 10 were installed with the Jetpack. The ZED SDK 3.2.2 was installed for working with the stereo camera ZED Mini. OpenGL 4.6 was employed for drawing the occupancy 2D grid. The library RS-232 was used for communicating with the haptic belt.

Table 6 presents the time that the system takes for executing the different processes. The number of frames (each frame has associated a depth image, a color image and pose data) that can be processed in a second is 5.96, which is appropriate for this application considering the average walking speed. Note that the most time-consuming process has to do with RANSAC, applied to the initial segmentation of the floor. This process varies from approximately 50 ms to 80 ms according to the number of points that fulfil the two criteria of the initial segmentation.

We could reduce the total time by taking off the display of data since this task is optional, and it does not affect the wandering mode of the system. The processing rate without displaying data is 7.56 fps. We could also reduce the time for getting data from the camera by using directly GPU memory of the camera instead of CPU memory, which is more efficient, since we do not have to go back on the CPU. This improvement is left as future work.

10. Discussion

Our system processes dense data on an embedded computer, segments the floor, and detects obstacles within 3 m of scope and at 7.56 fps (without displaying data), so it is not necessary to send data to a high-performance computer located in a remote point, unlike [28]. This update rate is sufficient in the context of vision-based navigation aids for blind people, considering the walking speed of a blind person, which in our experiments was in average of 0.17 m/s. Table 7 presents a comparison of systems that process depth data for segmenting the floor, using portable processors, in order to allow blind people to perceive the environment appropriately and move in a safe way.

Although embedded platforms have less computational capacity than laptops, these processors provide sufficient performance for the required tasks, as was proved in the systems of Table 7.

Our system uses tracking data (provided by the camera ZED Mini) in order to merge depth data from different poses, getting a global representation of the environment with memory of spaces previously visited, unlike [20, 30]. This representation is an occupancy 2D grid that enables us to include, in a future work, a module for object detection, like YOLO [36], for defining the location of objects of interest in the grid, and a module for path planning, like in [37], efficient and robust to dynamic environments and to user's movement, for computing at real time an optimal path.

Table 6: Performance with respect to time.

Process	Time (ms)
Get depth, color, and tracking data from camera. Validate data.	40.27
Display depth and color images. Concatenate transformation.	8.39
Compute point cloud referenced to camera frame.	1.72
Compute point cloud and normals, with transformations that make normal vectors parallel to gravity vector. Apply the two criteria for initial floor segmentation.	6.14
Apply RANSAC for estimating the equation of the plane that best fits to points of the initial segmentation of the floor.	73.84
Build a global occupancy 2D grid.	3.72
Execute a reactive navigation algorithm for avoiding obstacles.	0.22
Activate the haptic belt according to close obstacles in the grid.	0.01
Display the grid, the pose of the user, the trajectory, and the activation of the motors in OpenGL.	32.32
TOTAL time for a frame (on average)	167.60

Table 7: Comparison of performance with respect to time.

System	Processor	Dense data	Processing time (fps)
[34]	Laptop	Color and depth	7
[32]	Laptop	Color and depth	8 to 10
[33]	Laptop	Color and depth	10 (outdoor), 15 (indoor)
[29]	Laptop	Color and depth	Faster than 30
Ours	Jetson TX2	Depth	5.96, 7.56 (without visualization)
[20]	Embedded computer	Depth	10
[16]	Jetson TX1	Depth	More than 10

The haptic belt has shown in the experiments to provide identifiable patterns (accuracy in haptic perception of 90.75%) to guide the user to walkable space with little training and without interfering with sounds of the environment, allowing the user to avoid static and dynamic obstacles efficiently. Moreover, the camera located in the head has an important advantage with respect to locating it in the chest: obstacles hanging at heights up to 1.50 m can be detected and avoided. Moreover, the walking speed increased, and the number of collisions decreased as the training time increased. The walking speed achieved in our experiments goes from 0.14 m/s to 0.23 m/s, with an average speed of 0.17 m/s. This speed is similar to the one reported in [20] that goes from 0.09 m/s to 0.23 m/s in a maze navigation task with a depth camera, an embedded computer, and haptic feedback.

Another important factor for avoiding collisions is to pay complete attention all the time since a distraction of few seconds is enough to collide with obstacles, considering the distance for activating the vibration motors of 0.80 m and the average walking speed of 0.17 m/s. Besides wandering unknown environments in a safe way, these patterns can be used for path following when the path planning module is included, like in [29].

The techniques used for segmenting the floor, like the two criteria for an initial segmentation followed by RANSAC, worked appropriately at high speed on GPU, achieving to detect obstacles with height over 0.20 m. However, the accuracy of the occupancy grid depends on the accuracy of the depth data, so environments with low-light conditions and low-textured and very reflective floors are not well segmented. In this sense, the white cane could be a complement to our system, especially for detecting small obstacles and negative obstacles (e.g., holes in the ground). We left as future work a comparison of our methodology to segment the floor with Stixel world, which has shown outstanding results in [16, 20].

The female participants complained about the size, weight, and location of the processor while the male participants said that it was comfortable and did not interfere with their natural movement. We will test, in a future work, the Nvidia Jetson Xavier NX developer kit, located in the waist, as part of the haptic belt. This super computer has a similar price than the Nvidia Jetson TX2 developer kit, but it has better performance (more CUDA cores), is smaller, and lighter.

11. Conclusions

In this paper, we have shown that transferring technology from autonomous cars to the field of assistive tools for blind people is feasible due to (1) both have similar requirements such as real-time performance, work in unknown environments, robust to changing environments, and safe and (2) the increase in accuracy and portability of 3D vision sensors and in the computing power and portability of embedded processors. The aforementioned advances have allowed to build wearable systems that are able to process dense data at high frame rates. This is the case of our system that carries out efficiently a 3D segmentation of the floor and creates a global representation of the environment, which together with a reactive navigation algorithm guides the user to walkable space employing vibration patterns generated by a haptic belt. The system detects efficiently dynamic and static obstacles, at the level of the floor or hanging at heights up to 1.50 m. The participants highlighted the easiness to follow instructions with the haptic belt and the short period of training. Moreover, a global representation enables us to implement more complex tasks such as location of objects of interest in the occupancy grid together with path planning for proving purposeful navigation to the user, which is a posterior objective of this project. Finally, we plan to complement navigational information using haptic feedback with auditory descriptions of the scene when the user requests it.

We think that our system can improve the quality of life of blind people, making easier and safer the navigation in unknown environments and providing them more independence in daily activities.

Data Availability

Some SVO files captured with the camera ZED Mini during the experiments and used to support the findings of this study are available from the corresponding author upon request. Researchers who are interested in our code, called AssistNavBP v1.1, can find it in this public repository: https://bitbucket.org/pnbp/assistnavbpv1.1/src/master/.

Conflicts of Interest

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

Acknowledgments

This research was supported by the GOOGLE LARA 2018 (Latin America Research Awards), by FUNDACIÓN CEIBA with the program Bécate Nariño (grant CFRT-2019) and by UNAD (project code Ecbtiext042019).

Supplementary Materials

Video S1: AssistNavBPv1.1 with new experiments. Vision-based system for assisting blind people to wander. Link: https://youtu.be/rUWETxEx5yY. *(Supplementary Materials)*

References

[1] D. Tudor, L. Dobrescu, and D. Dobrescu, "Ultrasonic electronic system for blind people navigation," in *2015 E-Health and Bioengineering Conference (EHB)*, pp. 1–4, Iasi, Romania, 2015.

[2] M. F. Saaid, A. M. Mohammad, and M. S. A. Megat Ali, "Smart cane with range notification for blind people," in *2016 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pp. 225–229, Selangor, 2016.

[3] H. Sharma, M. Tripathi, A. Kumar, and M. S. Gaur, "Embedded assistive stick for visually impaired persons," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, Bengaluru, India, 2018.

[4] S. S. Bhatlawande, J. Mukhopadhyay, and M. Mahadevappa, "Ultrasonic spectacles and waist-belt for visually impaired and blind person," in *2012 National Conference on Communications (NCC)*, pp. 1–4, Kharagpur, 2012.

[5] K. Patil, Q. Jawadwala, and F. C. Shu, "Design and construction of electronic aid for visually impaired people," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 2, pp. 172–182, 2018.

[6] V. Filipe, F. Fernandes, H. Fernandes, A. M. R. Sousa, H. Paredes, and J. Barroso, "Blind navigation support system based on Microsoft Kinect," *Procedia Computer Science*, vol. 14, pp. 94–101, 2012.

[7] A. Aladrén, G. López-Nicolás, L. Puig, and J. Guerrero, "Navigation assistance for the visually impaired using rgb-d sensor with range expansion," *IEEE Systems Journal*, vol. 10, no. 3, pp. 922–932, 2016.

[8] A. Ali and M. A. Ali, "Blind navigation system for visually impaired using windowing-based mean on Microsoft Kinect camera," in *2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME)*, pp. 1–4, Beirut, Lebanon, 2017.

[9] H. Takizawa, A. Kitagawa, and M. Aoyagi, "Stereovision cane system: obstacle detection and seat recognition for the visually impaired," *IIEEJ Transactions on Image Electronics and Visual Computing*, vol. 6, no. 2, pp. 74–81, 2018.

[10] T. Schwarze, M. Lauer, M. Schwaab, M. Romanovas, S. Bohm, and T. Jurgensohn, "An intuitive mobility aid for visually impaired people based on stereo vision," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 409–417, Santiago, Chile, 2015.

[11] Q. Nguyen, H. Vu, T. Tran, and Q. Nguyen, "A vision-based system supports mapping services for visually impaired people in indoor environments," in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, pp. 1518–1523, Singapore, 2014.

[12] J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Smart guiding glasses for visually impaired people in indoor environment," *IEEE Transactions on Consumer Electronics*, vol. 63, 2017.

[13] H. Hakim and A. Fadhil, "Navigation system for visually impaired people based on rgb-d camera and ultrasonic sensor," in *Proceedings of the International Conference on Information and Communication Technology*, pp. 172–177, New York, USA, 2019.

[14] M. P. Arakeri, N. S. Keerthana, M. Madhura, A. Sankar, and T. Munnavar, "Assistive technology for the visually impaired using computer vision," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1725–1730, Bangalore, India, 2018.

[15] V. V. Meshram, K. Patil, V. A. Meshram, and F. C. Shu, "An astute assistive device for mobility and object recognition for visually impaired people," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 5, pp. 449–460, 2019.

[16] M. Martinez, A. Roitberg, D. Koester, R. Stiefelhagen, and B. Schauerte, "Using technology developed for autonomous cars to help navigate blind people," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 1424–1432, Venice, Italy, 2017.

[17] J. Wang, K. Yang, W. Hu, and K. Wang, "An environmental perception and navigational assistance system for visually impaired persons based on semantic Stixels and sound interaction," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1921–1926, Miyazaki, Japan, 2018.

[18] Y. Liu, N. R. B. Stiles, and M. Meister, "Augmented reality powers a cognitive assistant for the blind," *eLife*, vol. 7, 2018.

[19] R. Jafri, R. L. Campos, S. A. Ali, and H. R. Arabnia, "Visual and infrared sensor data-based obstacle detection for the visually impaired using the Google project tango tablet development kit and the Unity engine," *IEEE Access*, vol. 6, pp. 443–454, 2018.

[20] H. Wang, R. Katzschmann, S. Teng, B. Araki, L. Giarré, and D. Rus, "Enabling independent navigation for visually impaired people through a wearable vision-based feedback system," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 6533–6540, Singapore, 2017.

[21] T. Kurata, M. Kourogi, T. Ishikawa, Y. Kameda, K. Aoki, and J. Ishikawa, "Indoor-outdoor navigation system for visually-impaired pedestrians: Preliminary evaluation of position measurement and obstacle display," in *2011 15th Annual International Symposium on Wearable Computers*, pp. 123-124, San Francisco, CA, USA, 2011.

[22] A. Cosgun, E. A. Sisbot, and H. I. Christensen, "Evaluation of rotational and directional vibration patterns on a tactile belt for guiding visually impaired people," in *2014 IEEE Haptics Symposium (HAPTICS)*, pp. 367–370, Houston, TX, USA, 2014.

[23] R. K. Katzschmann, B. Araki, and D. Rus, "Safe local navigation for visually impaired users with a time-of-flight and haptic feedback device," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 3, pp. 583–593, 2018.

[24] A. Cosgun, E. A. Sisbot, and H. I. Christensen, "Guidance for human navigation using a vibro-tactile belt interface and robot-like motion planning," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6350–6355, Hong Kong, 2014.

[25] J. F. Oliveira, "The path force feedback belt," in *2013 8th International Conference on Information Technology in Asia (CITA)*, pp. 1–6, Kota Samarahan, 2013.

[26] Y. H. Lee and G. Medioni, "Rgb-d camera based wearable navigation system for the visually impaired," *Computer Vision and Image Understanding*, vol. 149, pp. 3–20, 2016.

[27] K. Yelamarthi and K. Laubhan, "Navigation assistive system for the blind using a portable depth sensor," in *2015 IEEE International Conference on Electro/Information Technology (EIT)*, pp. 112–116, Dekalb, IL, USA, 2015.

[28] H. Baskaran, R. L. M. Leng, F. A. Rahim, and M. E. Rusli, "Smart vision: assistive device for the visually impaired community using online computer vision service," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, pp. 730–734, Singapore, 2019.

[29] Y. H. Lee and G. Medioni, "Wearable rgbd indoor navigation system for the blind," in *Computer Vision - ECCV 2014 Workshops*, L. Agapito, M. M. Bronstein, and C. Rother, Eds., pp. 493–508, Springer International Publishing, Cham, 2015.

[30] D. Jeon, N. Ickes, P. Raina, H. Wang, D. Rus, and A. Chandrakasan, "24.1 a 0.6v 8mw 3d vision processor for a navigation device for the visually impaired," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 416-417, San Francisco, CA, USA, 2016.

[31] A. A. Díaz Toro, S. E. Campaña Bastidas, and E. F. Caicedo Bravo, "Methodology to build a wearable system for assisting blind people in purposeful navigation," in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, pp. 205–212, San Jose, CA, USA, 2020.

[32] N. Kanwal, "A navigation system for visually impaired: a fusion of vision and depth sensor," *Applied Bionics and Biomechanics*, vol. 2015, 16 pages, 2015.

[33] S. Caraiman, A. Morar, M. Owczarek et al., "Computer vision for the visually impaired: the sound of vision system," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 1480–1489, Venice, Italy, 2017.

[34] S. Caraiman, O. Zvoristeanu, A. Burlacu, and P. Herghelegiu, "Stereo vision based sensory substitution for the visually impaired," *Sensors*, vol. 19, no. 12, 2019.

[35] P. Herghelegiu, A. Burlacu, and S. Caraiman, "Robust ground plane detection and tracking in stereo sequences using camera orientation," in *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 514–519, Sinaia, Romania, 2016.

[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, 2016.

[37] M. Kapadia, F. Garcia, C. D. Boatright, and N. I. Badler, "Dynamic search on the GPU," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3332–3337, Tokyo, 2013.