Citation: A. Pérez-Yus, G. López-Nicolás, J. J. Guerrero. Detection and modelling of staircases using a wearable depth sensor. Second International Workshop on Assistive Computer Vision and Robotics (ACVR 2014) in Conjunction with ECCV 2014. Volume 8927 of the Lecture Notes in Computer Science series, 2015

Detection and Modelling of Staircases Using a Wearable Depth Sensor

Alejandro Pérez-Yus, Gonzalo López-Nicolás, and Jose J. Guerrero

Instituto de Investigación en Ingeniería de Aragón,
Universidad de Zaragoza, Spain
alperez@unizar.es, gonlopez@unizar.es, josechu.guerrero@unizar.es

Abstract. In this paper we deal with the perception task of a wearable navigation assistant. Specifically, we have focused on the detection of staircases because of the important role they play in indoor navigation due to the multi-floor reaching possibilities they bring and the lack of security they cause, specially for those who suffer from visual deficiencies. We use the depth sensing capacities of the modern RGB-D cameras to segment and classify the different elements that integrate the scene and then carry out the stair detection and modelling algorithm to retrieve all the information that might interest the user, i.e. the location and orientation of the staircase, the number of steps and the step dimensions. Experiments prove that the system is able to perform in real-time and works even under partial occlusions of the stairway.

Keywords: Stair detection, obstacle detection, segmentation, visually impaired, RGB-D

1 Introduction

The ability of navigating effectively in the environment is natural for people, but not easy to complete under certain circumstances, such as the case of visually impaired people or when moving at unknown and intricate environments. A personal guidance system must keep the subject away from hazards, but it should also point out specific features of the environment the user might want to interact with. In this paper we propose an algorithm which solves the detection of one of the most common features any person can come across during his daily life: the stairs. Finding stairs along the path has the double benefit of preventing falls and advertising the possibility of reaching another floor in the building.

To accomplish that we use a RGB-D sensor mounted on the chest, able to provide simultaneously color and depth information of the scene. The algorithm takes advantage of the depth perception to find the ground automatically and to dynamically recalibrate the ground position in order to project the 3D coordinates to a user-centered system. There is a segmentation process of the projected scene where the resulting segments are tentatively classified among floor, walls, random shapes and possible steps. Then the stairs detection algorithm outputs if the step candidates constitute a stairway, a single step or none. If a stairway is found, the algorithm retrieves how it is positioned with respect to the subject, how many steps can be seen and their approximate dimensions.

What we present here is a new algorithm for human navigation in indoor environments that serve as base for future add-ons to help us to understand better the scene. Our main contribution is a new stair detection and modelling module that provides full information of the staircases present before the subject. Experiments with video recordings in different indoor environments have accomplished great results in terms of accuracy and time response. Besides, a comparison of our results with the ones from other publications like [16, 17] has been performed, showing that the algorithm not only reaches state of the art performance but also includes further improvements. Specifically, our algorithm is the first known to the authors to be able to obtain the measurements of staircases even with people obstructing the view, allowing the extension of the information of the few steps detected to complete the staircase.

2 Related Work

Stairways are inevitably present in human-made environments and constitute a major problem in robot and human navigation. Many different types of sensors e.g. monocular and stereo cameras or laser scanning have been used for detecting stairs, all of them having intrinsic advantages and disadvantages. Some of the most outstanding publications on stairs detection using conventional cameras are [3], [6, 7] and [15], and using stereo vision [9] and [14]. Other authors preferred the use of laser scanning for stairs finding, most of them focused on robot navigation, such as [2], [11] and [13].

In recent years, RGB-D cameras, such as Microsoft Kinect or Asus Xtion Pro Live, have entered the consumer market at a reduced price (around 150€) causing great impact on the fields of computer vision and robotics. Their main feature is that they capture color and depth information of the scene simultaneously. The depth sensor can help perceiving the shape of a staircase more easily and as a consequence it can help performing a more robust detection. Moreover, the depth perception is independent of textures and lightning conditions. On the other hand, depth cameras do not work well outdoors or even indoors with strong sun reflections. In our recent work we have mostly focused on the depth sensing capacities of the RGB-D camera, but the combined use of color and depth information would overcome most limitations, and will be subject for further research [1]. Some authors who choose RGB-D as main sensor use machine learning algorithms to perform staircase detection [5], [18] while others prefer using geometrical reasoning, like [16, 17] and [4]. The later is the approach we also consider to solve this problem.

We believe that the existing algorithms using this technology are incomplete and can be improved. In [16] it is not taken into account the possibility of one or two steps, quite common in doorways or other special constructions. In both [16, 17] they do not model and retrieve the actual measurements of the steps, information which can be used to verify the detection, to give indications to the user or to analyse the traversability of the staircase. They use a RANSAC approach for finding planes in the scene that outcomes of each step a set of points at certain height (which in [16] even extends beyond the actual step) not using any other shape constraint but the sum of sufficient points. Also, our algorithm uses no other sensor than the RGB-D camera itself, instead of an accelerometer to calculate the relative position and orientation of the scene. Our automatic floor finding algorithm is able to find the floor with one single frame and then orient the scene accordingly for the succeeding stages not being deceived by other possible dominant planes in the scene.

Delmerico et al. proposed an ascending stairway modelling that introduces some interesting ideas [4]. Their goal is to localize and model stairways to check for traversability and enable autonomous multi-floor exploration. In order to build up a complete model of the stairway they align the point clouds from different views relying on the robot's estimated pose, which is complicated in human navigation. In addition, the stair edge detection, which is the starting point of their algorithm, is based on abrupt changes in depth that only appears in ascending staircases when the sensor is lower than the steps, i.e. a small robot. However, that collapses with our idea of a chest-mounted sensor. Moreover, the incapacity of the algorithm to detect descending stairs and their requirement of a minimum of three steps for detecting a stairway leaves a margin of improvement.

3 System Setup and Floor Detection

The first module of the proposed algorithm is presented in this section. In Section 3.1 it is explained how the system is intended to be worn and the data acquired. The floor finding algorithm and the coordinates projection is the subject of the Section 3.2.

3.1 Setup Configuration and Data Acquisition

There are different options to locate the camera in a wearable navigation system. Mayol-Cuevas et al. [10] presented an extensive research about this topic. The two most common choices are head-mounted and chest-mounted. The first one has the advantages of being intuitive as it resembles the eyes location, allows the subject to simply stand and scan the environment and makes harder to have the field of view obstructed. On the other hand, it is continuously moving and it adds more complexity to implement a robust and stable navigation system. Moreover, it is less secure as the user might be looking away from his most immediate hazards, as it cannot be controlled. A chest-mounted system remains fixed to the body in a comfortable and secure manner, allowing the user to move freely knowing that the assistant will warn of any danger along the path. For these considerations we have chosen a chest-mounted system as the best option.

The camera will be slightly pointing downwards, at approximately 45° down. As the RGB-D sensor employed has a 45° of vertical field of view, it should be enough to locate the obstacle-free path in front of the subject and to easily

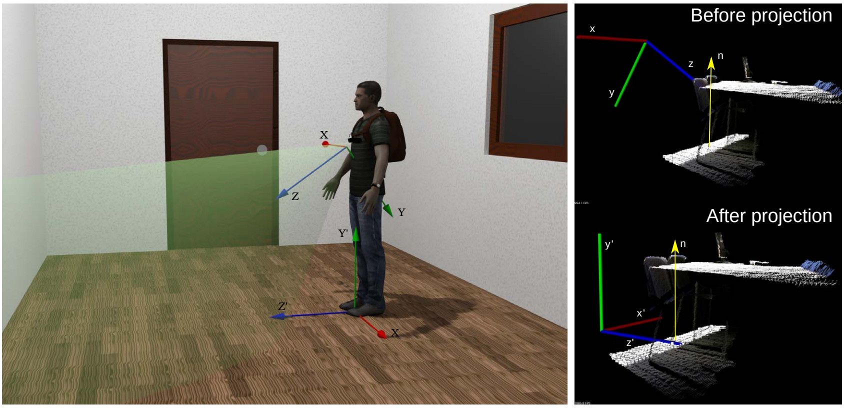


Fig. 1: Wearable camera location and axis position before (XYZ) and after $(X'Y'Z')$ the projection to the ground in a 3D render (left). Point clouds from a real case scenario, where the white points on the floor are those which form the best floor candidate plane and the yellow arrow is the corresponding normal (right).

detect stairs in the scene. Currently, all the computations are operated on a laptop which could be carried in a backpack. A scheme of the configuration is shown in Fig. 1 (left).

The basic type of data used by our system are the so-called *point clouds*, consisting on a set of data of each pixel which contains the 3D location with respect to the camera and the RGB information. We have used Robot Operating System (ROS) as framework and the Point-Cloud Library (PCL) as our main library to deal with this type of data. Video sequences or single point clouds can also be stored to work offline. Capturing the data once the system is running is not highly time-consuming (about 30 Hz).

The amount of data generated by each point cloud is too large to be entirely used and thus the first operation will be filtering. For this we use a 3D voxel grid filter, a common algorithm widely used for downsampling point clouds, which also helps removing noise and smoothing the surfaces. The sizes of the edges of the voxels are determined by balancing time consumption and accuracy of the data. The filtered cloud will be used in the remaining stages.

3.2 Floor Detection

The point clouds have 3D Euclidean measurements of its location in front of the camera, but it is necessary to calculate the relative position between the sensor and the subject in order to convert the raw information acquired to oriented data that would help knowing the absolute position of the objects in the scene. The axis of the coordinate system will be transformed as shown in Fig. 1.

This projection requires to find the plane that most likely corresponds to the floor, which may not be the most dominant. No other sensor has been used for this task, so the only previous knowledge is the approximate location of the camera on the chest, which can vary due to the movement and the height of the subject. A 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. A pass-through filter in $z - axis$ can additionally be used to restrict the search to the proximity of the user (Fig. 1 in the right). If the floor is not found with the first cloud, a new one will be captured and the process will be repeated. As the camera is pointing downwards and the user is supposed to be standing on the floor, the process should not last long.

Once a set of points belonging to a good floor candidate plane are found, the transformation matrix is computed. The normal of the plane has to be parallel to the direction of the $y - axis$, and the origin of coordinates is placed on the floor, at height 0. The fundamental plane equation is $Ax + By + Cz + D = 0$, being the normal vector $\boldsymbol{n} = (A, B, C)$ and D the perpendicular distance from the origin to the plane. The rotation angles of interest are those corresponding to the $x - axis$ (α) and to the $z - axis$ (γ). It is possible to get those angles by computing the rotation needed to make $\boldsymbol{n}$ parallel to $\boldsymbol{j} = (0, 1, 0)$ as shown in (1). The entire transformation matrix is shown in (2).

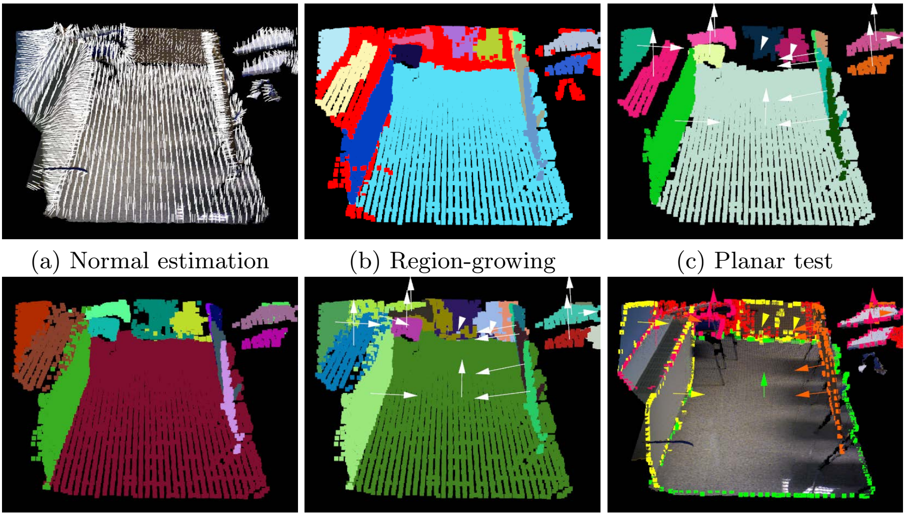
$$R_z R_x \boldsymbol{n}^T = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \boldsymbol{j} \tag{1}$$

$$T = \begin{pmatrix} \cos\gamma & -\sin\gamma\cos\alpha & \sin\gamma\sin\alpha & 0 \\ \sin\gamma & \cos\gamma\cos\alpha & -\cos\gamma\sin\alpha & -D \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

4 Segmentation of the Scene

Segmentation has been an essential issue in robot and human navigation through the years. In order to perform any relatively complex task it is necessary to recognize the features of your surroundings. Our case of study is indoor environments, where, like in most human-made scenarios, the basic structure of the scene is a combination of planes at different orientations. Range sensors have proven to be extremely helpful for this mission, and many different algorithms have been developed to perform the segmentation [8].

In this work a region-growing strategy has been used, enhanced with some refinement functions. Regions are afterwards classified as planar and non-planar using a RANSAC algorithm. We prefer this approach instead of using directly plane detection algorithms, such as RANSAC or Hough transform, because with region-growing the planes found form already a closed region corresponding to one single element and are not a set of uncorrelated points scattered in the scene [16]. The remaining points are later merged into existing planes or associated to



(a) Normal estimation

(b) Region-growing

(c) Planar test

(d) Planes extension

(e) Cluster extraction

(f) Classification

Fig. 2: Example of segmentation and classification from a single frame.

different clusters of points. In particular, the segmentation module is divided in the following stages:

Normal estimation (Fig. 2a): The normal estimation problem is solved using an algorithm derived from the Principal Component Analysis. For each point and a group of K neighbours, the third component obtained from the analysis corresponds to the normal direction, flipped towards the viewpoint. In this process the curvature of the surfaces is also computed.

Region-growing (Fig. 2b): This algorithm starts from a *seed*, which is the point with minimum curvature, and then expands the region towards the neighbouring points that have small angle between the normals and similar curvature value. The neighbouring points which satisfy the normal and curvature threshold became the new seeds and repeats until the region cannot expand anymore. Then, a new initial seed is chosen among the remaining points, and the process starts over until the regions are smaller than a certain threshold.

Planar test (Fig. 2c): Because of how the region-growing algorithm works, most regions are planes or have a high degree of flatness, but they can also be a curved surface with smooth transitions. As the ground, walls, doors or steps are all planes, it is important to test this condition. A RANSAC algorithm looks for the biggest plane in each region and, if most of the points are inliers, it will be considered a planar surface with the plane equation obtained. Otherwise, the regions will be considered as arbitrary obstacles of the scene.

Planes extension (Fig. 2d): The points not belonging to any region are included in a planar region if they have small angle between their normal and the plane normal, they have a small perpendicular distance to the plane and they are situated near the region.

Euclidean cluster extraction (Fig. 2e): The points not belonging to any region go through a cluster extraction algorithm which establish connections and form separate entities, considered obstacles.

Plane classification (Fig. 2f): Once the segmentation stage has succeeded the planes are classified among different classes according to the orientation and the relative position of the planes. The orientation of the plane normals is compared to the normal vector of the floor. If the angle between the planes and the ground is close to 90°, they are tentatively classified as *walls*, whereas planes whose angles are close to 0° are considered horizontal. Any other circumstance is considered obstacle. In this case the term *walls* simply defines a vertical planar structure as no further reasoning has been done yet.

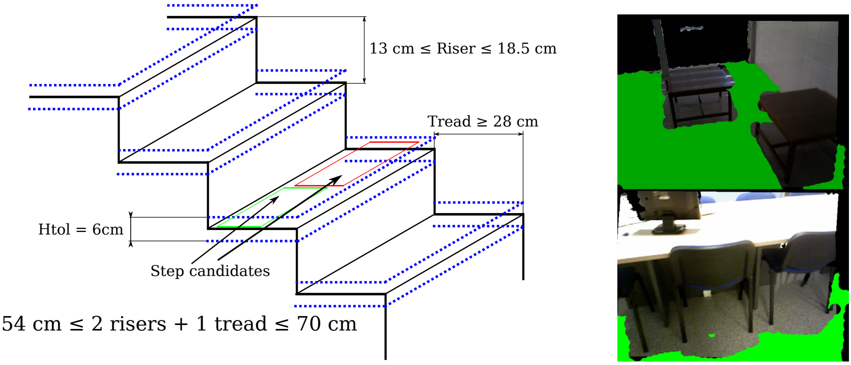


Fig. 3: Representation of the measurements of a step according to the Technical Edification Code from Spain (left). In green, the portion of the ground which can be walked over as it has no obstacles above (right).

Horizontal planes can be ground, steps or other obstacles that should be avoided (e.g. a table). It is common that the floor or steps are composed by more than one planar region as occlusions can happen. The height of the centroid of the planes is then considered: The regions with height close to zero are classified as floor, whereas the regions with positive or negative height are classified as *step candidates* if they satisfy the minimum height requirements regulated by the current Technical Edification Code valid in Spain [12] (see Fig. 3 left). According to the Code, the height of the steps ranges from a minimum H_{min} = 13cm to a maximum H_{max} = 18.5cm. Horizontal regions will be considered as *step candidates* if they are situated above (in ascending stairways) or below (in descending ones) $H_{min} - H_{tol}/2 = 10$cm from the ground. It is necessary to add a tolerance as the measurements can be too noisy. The existence of a set of at least one *step candidate* activates the stair detection algorithm. Other size and shape restrictions are kept to a minimum at this point because they could discard valid portions of steps which might be useful for a better modelling of staircases.

As a result of the segmentation and classification algorithm, the obstacles position can be projected to the ground to remove the non-walkable area from the ground detected (Fig. 3, right). Additionally, if the floor plane equation has significantly changed, a new transformation matrix is computed to not lose track of the orientation of the scene.

5 Stair Detection and Modelling Algorithm

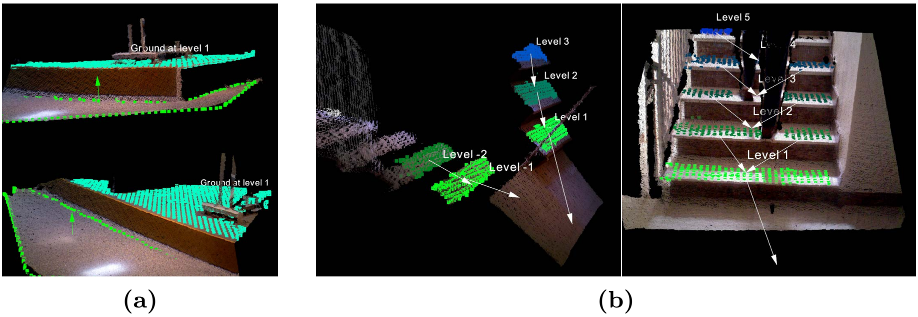
In Section 4 it has been explained how the *step candidates* are obtained. Our stair detection and modelling algorithm is the next phase, using these *step candidates* as input, and providing the full characteristics of the staircase as output. At this moment, the algorithm is functional with both ascending and descending staircases, being capable of detecting one of each at a time. There is no restrictions about the number of steps belonging to a staircase, making also isolated single steps detectable during navigation. Our work goes beyond simple detection and models the staircase even with partial occlusions such as people walking the stairs. That means that every step can be found split in different regions. Spiral staircases can be detected but the modelling part has not been addressed yet.

5.1 Stair Detection

The detection algorithm establishes connections among the candidates to discard the ones that do not belong to the staircase and to group the stair planes in *levels* according to the distance in steps to the floor. The candidates are analysed one by one in a bottom-up strategy, for which it is necessary to select a *first step*. The candidates whose centroid is closer than $H_{max} + H_{tol}/2 = 21.5cm$ to the ground constitute *first step candidates*. If there are more than one, the connectivity to the levels above and below must be tested, otherwise it is immediate. The connectivity between regions has been computed using neighbour search and Kd-trees. The first step must also be connected to the floor if it is present in the image, i.e. if the user has not walked too close to the staircase. In a live video sequence, when there is no floor in sight, as the relative position of the camera and the user has already been computed and we know where the ground is, the connection to the ground does not need to be tested. If no first step candidate satisfies neighbouring conditions, the algorithm determines there is no staircase.

A special case occurs when there is only one possible step. It might either actually be the first step of a staircase, or be just a single step on the way. But it also can be an object which should be considered an obstacle. Here, strict area and shape conditions can be applied in order to determine in which case we are. In Fig. 4a we show an example where the ground at another level is detected.

Once there is a first step, the algorithm takes the remaining step candidates by height and starts testing connectivity and height conditions to determine whether they belong to the current or to a new level. If they have no connection to previous levels (e.g. a horizontal plane correspondent to a table) they are



(a) (b)

Fig. 4: (a) Floor at another level examples. (b) Connectivity between step candidates to previous levels: Ascending and descending staircases (left), more than one region per level (right).

classified as obstacles. As a result, a set of connected regions corresponding to different levels is obtained (Fig. 4b). When all the candidates have been checked, if the number of levels is greater than one, the algorithm starts the modelling of the staircase.

5.2 Stair Modelling

For modelling staircases we consider the following global geometric parameters: the width of the steps, the length of the tread, the height of the riser and the number of steps. To achieve that we use the Principal Component Analysis (PCA). This analysis is applied to each set of points corresponding to the tread of the step in each level of the staircase. Usual staircases have rectangular steps with much more width than length. The first component obtained from the PCA corresponds to the longitudinal direction (width), the second component follows the direction along the length of the step and the third component is orthogonal to the previous two, matching the normal direction of the tread (Fig. 5 left).

Mathematically, it consists in calculating the centroid of the data points, which is the mean value on each axis $\mu_{\mathbf{x}} = (\mu_x, \mu_y, \mu_z)$ and the covariance matrix of the data $\varSigma$, which is a 3×3 matrix as we are in 3D coordinates. The eigenvectors of the covariance matrix are the principal components ϕ_1, ϕ_2 and ϕ_3, being the correspondent to the highest eigenvalue the first component (width), the second highest the second component (length) and the lowest eigenvalue the third component (vertical). If we form a matrix with these vectors in columns we obtain the transformation matrix $\varPhi = [\phi_1, \phi_2, \phi_3]$ which transforms our points $P_{\mathbf{x}}$ from the initial $\mathbf{x} = (x, y, z)$ axis system to P_ϕ in the principal direction axis (ϕ_1, ϕ_2, ϕ_3) with the equation:

$$P_\phi = (P_{\mathbf{x}} - \mu_{\mathbf{x}}) \cdot \varPhi \tag{3}$$

Once we have our cloud transformed to the new axes it is easy to get the minimum and maximum coordinate in each direction to obtain the oriented

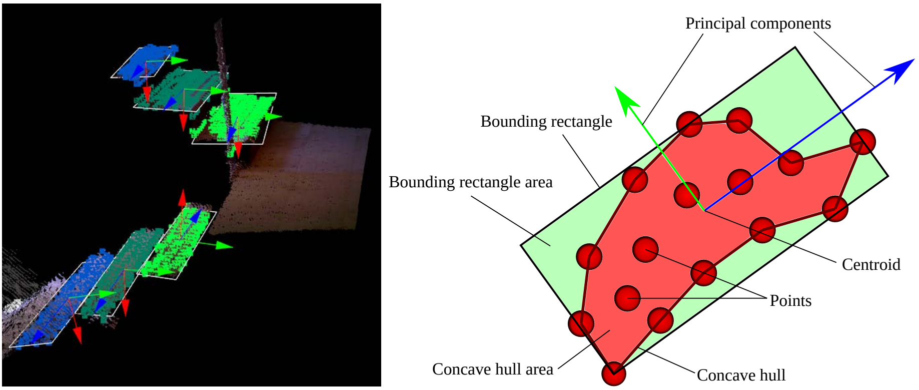


Fig. 5: Principal components for each step coloured in order (blue-green-red) and bounding rectangle in white (left). Illustrative sketch of the different components (right).

bounding box of the step. As the height is small it can be considered negligible, considering the step as a two-dimensional rectangular bounding box (Fig. 5 right). The difference between the maximum and the minimum in the first and second component are the width and the length respectively. We define *extent* as the ratio of the area of the concave hull including the points and the area of the rectangle. The extent is used to measure the quality of the detected step as it relates the area occupied by the points with respect to the area they are supposed to occupy.

The process is repeated with all steps, considering the addition of clouds at the same level as the cloud of the step. Each step has different dimensions and orientations depending on the quality of the measurements, the position of the steps with respect to the camera or the filters performance. We will choose the best step as the one with higher extent value among the steps within the valid width range, and its principal components and width will be considered as initial best guess for the model. The valid width range we choose ranges from the maximum width value detected to that maximum value minus 25cm. The principal direction of the staircase is corrected in two ways. Firstly, by forcing the third principal component to be parallel to the vertical axis. Secondly, by minimizing the sum of the area of the bounding rectangles of every step by rotating the two principal directions of the best step.

The obtention of the bounding boxes and dimensions is repeated for each step with the definitive staircase orientation. The steps will be modelled as parallelograms whose width is the width of the best step, the height is the average vertical distance between consecutive steps and the length the mean horizontal distance between the edge of every two consecutive steps. The definitive length of the steps is computed this way because the vertical projection of the bounding rectangles of two consecutive steps usually overlaps in ascending staircases (due to inclining or non-existent risers) or leaves a gap in descending staircases (due

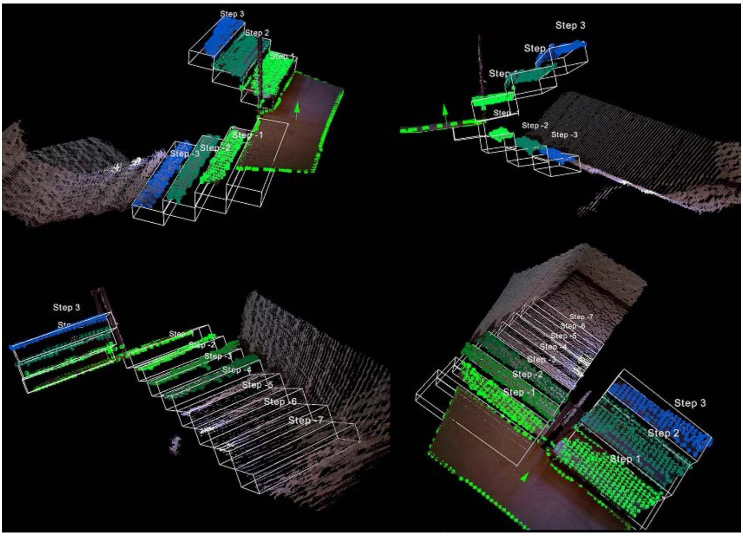


Fig. 6: Estimated model of the staircase. Top images only draws the parallelograms corresponding to the steps found, whereas at the bottom all the steps are displayed.

to self occlusions). Once we have all the parameters, we can use them to validate the staircase detection or discard it, and in case of positive results we can trace the model and even extend the information to non-detected steps (Fig. 6).

6 Experimental Evaluation

The experiments were carried out in a 3.4Ghz computer with 16 Gb of RAM running Ubuntu 12.04, ROS Hydro and the library PCL version 1.7.1. With this framework we were able to capture 640 × 480 3D point clouds in real-time and record video sequences and single frames for later experiments. Although we already had our own recordings from previous research,[1] new scenarios including stairs were also recorded to conduct specific experiments. With our own datasets we could observe that the performance of the system improves when it is used in a real-time video sequence, live or recorded. In this mode, the floor detection algorithm is only used once, and as a result its presence in the image is not required all the time.

Tang et al. compiled a dataset in [16] which includes 148 captures made with a Microsoft Kinect sensor. 90 of them include RGB and depth snapshots of a set of staircases from different poses and the other 58 are normal indoor scenes to test for false positives. The accelerometer measurements of the sensor position were also included but they are not used in our work. From the RGB and disparity range image the point cloud was calculated in each case, using previous

[1] http://webdiis.unizar.es/%7Eglopez/dataset.html

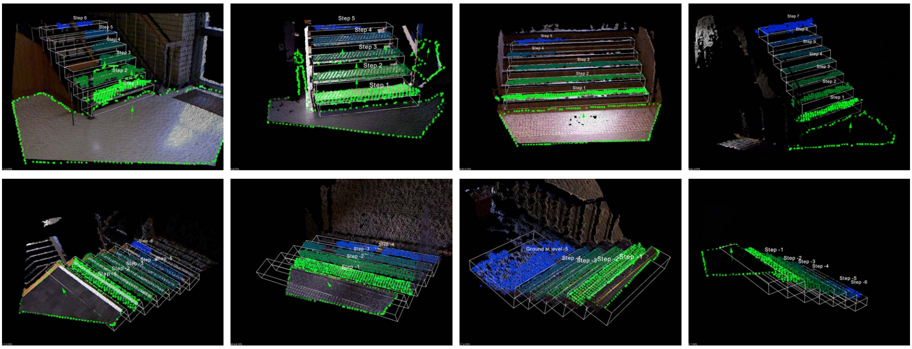


Fig. 7: Some examples of results obtained with the dataset. The last column are from captures made in darkness.

information about the calibration parameters of the camera. The results of the test with this dataset were successful even in total darkness (Fig. 7). We tested for false positives and false negatives using this dataset and compared our results with the ones from [16] and [17] (Fig. 8a). We achieve better results with the 0% of false negatives as in [17] but also reaching a 0% of false positives.

It is also interesting to look at the step detection ratio according to the position of the step in the staircase (Fig. 8b). The behaviour changes when we are facing an ascending staircase or a descending one. Due to the orientation of the chest-mounted sensor, standing before a descending staircase allow us to see the whole staircase but the self occlusion of consecutive steps and quality of the measurements decreasing with the distance harms the detection of steps farther than the third position. In ascending staircases the ratio of detection diminishes in a less prominent way, because the steps remain almost as close to the subject as they rise, although with the penalty of having less and less visual angle. Steps higher than the seventh position are out of the field of view of the camera.

The computing time was also tested to analyse the performance of the system and to compare it to the state of the art. The complete loop iteration time ranges from 50 to 150ms, giving a rate of $7-20$Hz. The variation depends on the scene itself: close up captures provides good quality clouds and the segmentation algorithm provide less regions and as a consequence, faster results. On the other hand, a capture taken to a scene situated far from the camera adds more noise and less smooth surfaces. In general, this timing should be considered fast enough for indoor navigation assuming walking speeds around $1-1.5$m/s. A breakdown of the time distribution is shown in Table 1. This rate could be improved adding some optimizations to the algorithm or using multi-core processing, although no optimization efforts have been done yet.

We have also quantitatively analysed the resemblance of the model to the real staircase. We have excluded the width from the analysis as the view of the stairs may be partial and it is not as relevant as the other measurements. After computing the height and length of a staircases, in both ascending and

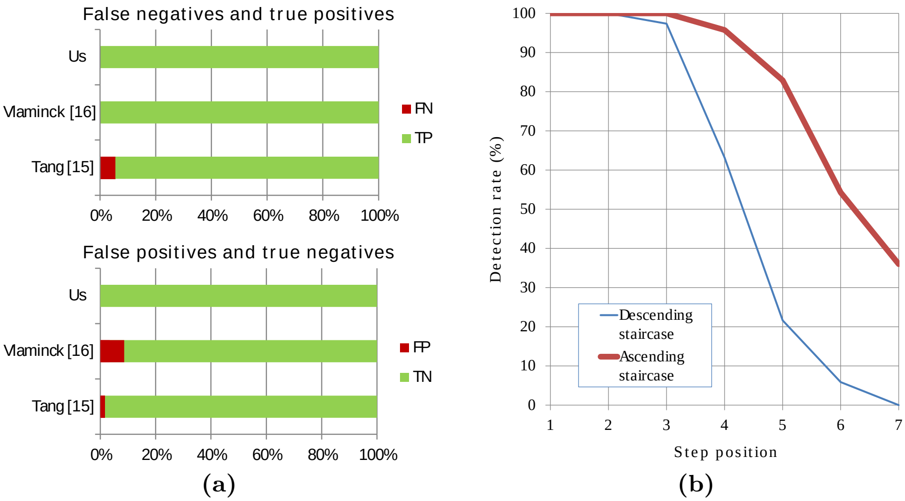


Fig. 8: (a) Comparison of false negatives and false positives between our work and the presented by [16, 17]. (b) Step detection rate with the step position in the staircase.

Table 1: Average time of the stages of the algorithm

Stage	**Time**
Filtering	15ms
Ground extraction	3ms
Normal extraction	13ms
Region-growing	16ms
Plane extension	20ms
Cluster extraction	5ms
Classification	16ms
1 stair detection	5ms

Table 2: Average and standard deviation (in centimetres) of the length and height measured with and without obstacles.

	No obstacles		**Obstacles**		**Real**
	$\bar{\mathbf{x}}$	σ	$\bar{\mathbf{x}}$	σ	$\mathbf{x_r}$
Length	29	2.01	29.39	1.89	30
Height	15.4	1.36	15.56	0.59	17

descending perspectives, from different viewing angles, the results were compared to the real measurements, as shown in the Table 2. As we can observe, the values do not have strong deviation. Half of the experiments were conducted with real people going up and down the stairs. Obstructing the view of the staircase partially does not adversely affect the quality of the model. Some pictures of the experiments with people climbing up/down the staircase can be seen in Fig. 9.

7 Conclusions and Future Work

In this paper we have presented the perception module of a wearable personal assistant oriented to visually impaired people, although it may have applications in other fields such as robotics or special cases of human navigation. Our main

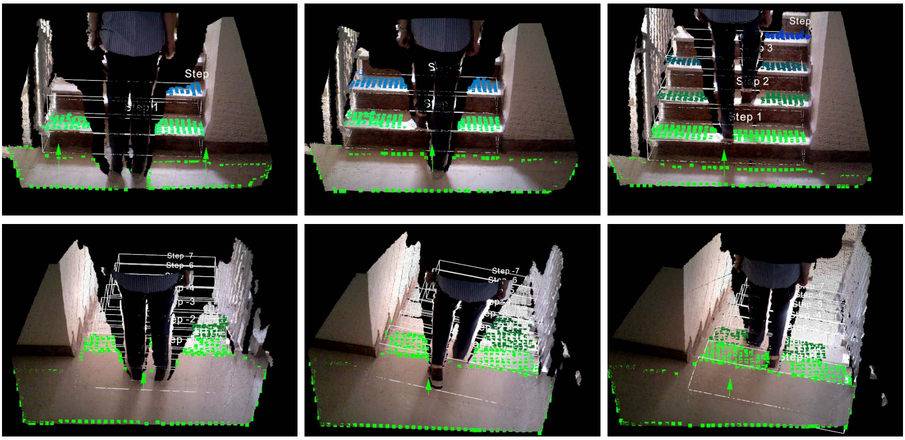


Fig. 9: Example of a person partially blocking the view of the staircase during ascent or descent.

contribution is the stair detection algorithm, which is not only able to detect but also to model staircases with their complete dimensions and position with respect to the user. That would provide the subject with multi-floor navigation possibilities. The experiments prove that the model quality and the computing time are good enough to be used in real-time. The algorithm overcomes some limitations existing in related works, such as the possibility of single step detection or full modelling with partial occlusions caused mainly by other people traversing the staircases.

More detection features are expected to be developed and added to the personal assistant, such as door detection, text sign recognition or people detection. But first we would like to extend the possibilities that a RGB-D sensor can bring to stair detection by combining the depth information with color images. RGB data would help improving the model, counting the steps to extend the staircase model, detecting possible staircases from farther distances where depth measurements are not reliable or when the sun rays affect negatively the depth sensing. It is also required to test the system by users in real scenarios in order to receive feedback for improving our work.

Acknowledgments This work was supported by Ministerio de Economía y Competitividad and European Union under FPI grant BES-2013-065834 and project DPI2012-31781.

References

1. Aladren, A., Lopez-Nicolas, G., Puig, L., Guerrero, J.: Navigation assistance for the visually impaired using RGB-D sensor with range expansion. IEEE Systems

Journal, Special Issue on Robotics & Automation for Human Health PP(99), 1–11 (2014)

2. Albert, A., Suppa, M., Gerth, W.: Detection of stair dimensions for the path planning of a bipedal robot. In: IEEE International Conference on Advanced Intelligent Mechatronics (AIM). vol. 2, pp. 1291–1296 (2001)
3. Cong, Y., Li, X., Liu, J., Tang, Y.: A stairway detection algorithm based on vision for ugv stair climbing. In: IEEE International Conference on Networking, Sensing and Control (ICNSC). pp. 1806–1811 (2008)
4. Delmerico, J.A., Baran, D., David, P., Ryde, J., Corso, J.J.: Ascending stairway modeling from dense depth imagery for traversability analysis. In: International Conference on Robotics and Automation (ICRA). pp. 2283–2290 (2013)
5. Filipe, V., Fernandes, F., Fernandes, H., Sousa, A., Paredes, H., Barroso, J.: Blind navigation support system based on microsoft kinect. Procedia Computer Science 14, 94–101 (2012)
6. Hernández, D.C., Jo, K.H.: Stairway tracking based on automatic target selection using directional filters. In: Frontiers of Computer Vision (FCV). pp. 1–6 (2011)
7. Hesch, J.A., Mariottini, G.L., Roumeliotis, S.I.: Descending-stair detection, approach, and traversal with an autonomous tracked vehicle. In: Intelligent Robots and Systems (IROS). pp. 5525–5531. IEEE (2010)
8. Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., Fisher, R.B.: An experimental comparison of range image segmentation algorithms. Pattern Analysis and Machine Intelligence 18(7), 673–689 (1996)
9. Lu, X., Manduchi, R.: Detection and localization of curbs and stairways using stereo vision. In: International Conference on Robotics and Automation (ICRA). vol. 4, p. 4648 (2005)
10. Mayol-Cuevas, W.W., Tordoff, B.J., Murray, D.W.: On the choice and placement of wearable vision sensors. Systems, Man and Cybernetics, Part A: Systems and Humans 39(2), 414–425 (2009)
11. Mihankhah, E., Kalantari, A., Aboosaeedan, E., Taghirad, H.D., Ali, S., Moosavian, A.: Autonomous staircase detection and stair climbing for a tracked mobile robot using fuzzy controller. In: International Conference on Robotics and Biomimetics (ROBIO). pp. 1980–1985 (2009)
12. Ministerio de Fomento. Gobierno de España: Código Técnico de la Edificación, Documento Básico de Seguridad de Utilización y Accesibilidad (DB-SUA, Section 4.2) (2014)
13. Oßwald, S., Hornung, A., Bennewitz, M.: Improved proposals for highly accurate localization using range and vision data. In: International Conference on Intelligent Robots and Systems (IROS). pp. 1809–1814 (2012)
14. Pradeep, V., Medioni, G., Weiland, J., et al.: Piecewise planar modeling for step detection using stereo vision. In: Workshop on Computer Vision Applications for the Visually Impaired (2008)
15. Se, S., Brady, M.: Vision-based detection of staircases. In: Asian Conference on Computer Vision (ACCV). vol. 1, pp. 535–540 (2000)
16. Tang, T.J.J., Lui, W.L.D., Li, W.H.: Plane-based detection of staircases using inverse depth. Australasian Conference on Robotics and Automation (ACRA) (2012)
17. Vlaminck, M., Jovanov, L., Van Hese, P., Goossens, B., Philips, W., Pizurica, A.: Obstacle detection for pedestrians with a visual impairment based on 3d imaging. In: International Conference on 3D Imaging (IC3D) (2013)

18. Wang, S., Tian, Y.: Detecting stairs and pedestrian crosswalks for the blind by rgbd camera. In: International Conference on Bioinformatics and Biomedicine Workshops (BIBMW). pp. 732–739 (2012)