Human navigation assistance with a RGB-D sensor

J. J. Guerrero, A. Pérez-Yus, D. Gutiérrez-Gómez, A. Rituerto, G. López-Nicolás
Universidad de Zaragoza,
Instituto de Investigación en Ingeniería de Aragón (I3A)
{josechu.guerrero, alperez, danielgg, arituerto, gonlopez}@unizar.es

Abstract

This paper focuses on the creation of a human navigation assistance prototype. The system uses a conventional RGB-D camera and a laptop to analyze the environment surrounding the user and provides him with enough information for a safe navigation. The system is designed to work indoors and performs two main tasks: floor and obstacle detection and staircase detection. Both tasks make use of the range and visual information captured by the sensor. The camera points downwards, allowing to acquire relevant navigation information without invading the privacy of other people. The system has been tested in real environments showing good results in the detection of obstacles and staircase.

Resumen

Este trabajo se centra en el diseño de un prototipo de asistencia a la navegación para personas. El sistema se basa en un sensor RGB-D portable conectado a un PC para analizar el entorno alrededor del usuario y facilitarle información para la navegación en este entorno. El sistema está diseñado para trabajar en interiores y realiza dos tareas principales: detección del suelo y obstáculos cercanos y detección de segmentos de escalera. Ambas tareas utilizan la información, tanto de profundidad como visual capturada por el sensor. La cámara está dispuesta mirando hacia abajo para capturar información relevante para la navegación sin interferir en la privacidad de otras personas. El prototipo ha sido probado en entornos reales mostrando buenos resultados en la detección de obstáculos y escaleras.

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. Wearable intelligent systems are great platforms for navigation assistance. Those systems can be very useful for improving or complementing the human abilities in order to better interact with the environment. In this context, project VINEA (Wearable computer VIsion for human Navigation and Enhanced Assistance) aims for the consecution of a personal assistance system based on visual information. This system will help people to navigate in unknown environments and it will complement the human abilities. Possible users of this system will range from visually impaired people to users performing specific tasks that complicate the visibility or accessing to poor visibility 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 present a system that benefits of the use of new and affordable RGB-D cameras to assist the user navigation. Two navigation problems are faced and solved: floor and obstacle detection and staircase detection.

The system uses chest mounted RGB-D camera. The camera points to the floor, capturing the traversable area in front of the user. This configuration allows to capture information important for the navigation (e.g. floor plane, close objects and obstacles) while sensitive information and privacy of other people is out of the field of view of the sensor.

RGB-D sensors provide range and color information. Range information is used to detect and classify the main structural elements of the scene. Due to the limitations of the range sensor, the color information is jointly used with the range information to extend the floor segmentation to the entire scene. In particular, we use range information for closer distances and color information is used for larger distances. This is a key issue not only to detect near obstacles but also to allow high level planning of the navigational task thanks to the longer-range segmentation our method provides. Once we have detected the floor of the scene, we solve the detection and modeling of one common obstacle that a person can come across while moving around: 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. Additionally, we have developed a user interface that sends navigation commands via sound map information and voice commands.

The proposed system has been tested with a user wearing the prototype on a wide variety of scenarios and datasets. The experimental results show that the system is robust and works correctly in challenging indoor environments.

This work is a step forward towards the creation of a human navigation assistance tool. The technical details and evaluations of the detection approaches used here have been individually presented in papers [1] and [2].

2. Related Work

Many different navigation assistance systems for visually impaired have been proposed in the literature [3]. In general, they do not use visual information and they need complex hardware systems, such as wireless communication technology, or ultrasonic and GPS sensors [4]. Other approaches propose the use of colored navigation lines set on the floor and RFID technology to create map information, [5]. Or the creation of a previous floor map of a building to define a semantic plan for a wearable navigation system by means of augmented reality, [6].

Vision sensors play a key role in perception systems because of their low cost and versatility. The work in [7] presents a system for indoor human localization based on global features that does not need 3D reconstruction. A disadvantage of monocular systems is that global scale is not observable from a single image. A way to overcome this problem is using stereo vision such as in [8].

In recent years, RGB-D cameras have gained importance on the fields of computer vision and robotics thanks to their low price and the combination of range and color sensors. They capture color and depth information of the scene simultaneously. The depth information can help to perceive the shape of the scene and it is independent of textures and lightning conditions, however, it is usually limited to about 5 meters. Color information complements this limit, and can include surface details not present in the range data. This is the only sensor used in this work, which benefits from both the range and visual information to obtain a robust and efficient system.

These kind of sensors has been used to find and identify objects in the scene [9, 10]. One step ahead is to integrate range systems in the navigation task. Some examples are [11], where a Kinect sensor is used, [12]

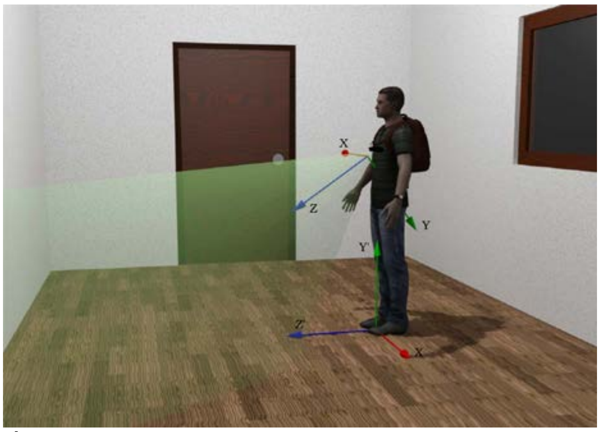


Fig. 1 Wearable camera position: the RGB-D sensor is chest mounted and it looks downwards 45º; the laptop where all the computation is carried on is on the backpack. The image shows the field of view of the sensor (green) and the axis of the scene (X'-Y'-Z') and the sensor (X-Y-Z).

where range information is used to distinguish solid obstacles from wild terrain, or [13], where FAST corner detector and depth information for path planning tasks are used. RGB-D cameras can be also used to estimate the motion and the 3D structure of the scene [14]

Regarding the problems faced in our approach, we see how computer vision has been used before for floor and path-segmentation. The work in [15] presents a system that solves floor-segmentation using hue and light information of the images. In [16], authors use a histogram-based road classifier. In [17], a method to find the drivable surface with appearance models is presented, and [18] shows how the fusion of information, in particular color and geometry information, improves the segmentation of the scene. We exploit this idea by extending the structure estimated from the depth data with the information from the color image.

Stairs detection has also been faced using conventional cameras [19], stereo vision [20] and even laser scanning [21]. We find also approaches using RGB-D as main sensor and machine learning algorithms to perform the staircase detection [22, 23]. Papers [24, 25] use also RGB-D cameras and geometric reasoning to detect the stairs. This is the approach we have consider in our method for stair detection. We start from the traversable area detected with our floor detection approach and detect and model staircase with one or more steps.

3. Prototype setup

There are many options to locate a camera or a RGB-D sensor for a wearable navigation system [26]. The RGB-D device provides range information from active sensing by means of infrared sensor and

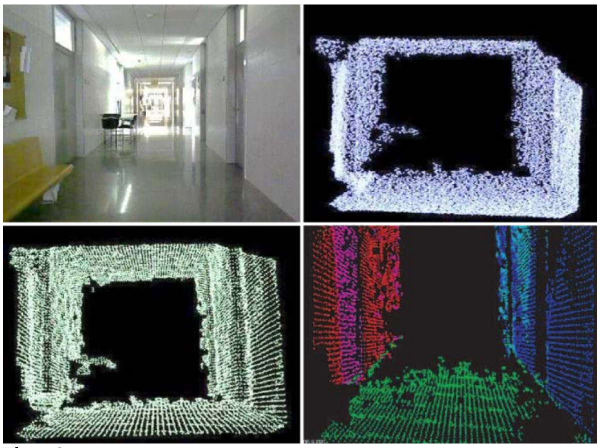


Fig. 3 Process of the floor segmentation in the range data. (Top-left) RGB image. (Top-right) Range data point cloud. (Bottom-left) Filtered range data point cloud. (Bottom-right) Floor segmentation (green) and obstacles (blue and red).

intensity images from a standard camera. We have chosen a chest-mounted system so the sensor remains fixed to the body comfortably for the user that can move freely. The sensor points to the front of the user at all the times being able to detect dangers along the path. We set the camera pointing slightly downwards, 45º down, this way the captured details are mainly the floor plane and obstacles in front of the user. The set up can be seen in Fig. 1.

This camera setup can be classified as *sousveillance*, opposed to *surveillance* (where the camera is fixed to an outside object of the environment). Our configuration shows a great potential to be used for personal safety and security, improved eyesight or augmented reality.

Currently, the RGB-D sensor is connected to a laptop carried in a backpack and performing all the computations. The algorithms are implemented in C++ language for ROS (Robot Operating System), OpenCV library for image processing and PCL (Point-Cloud Library) to process the range data.

4. Floor and obstacle detection

Our approach to detect the floor plane and the obstacles in it is performed in two main steps. First, using the range data, we detect the floor and the objects close to the user. In a second step, the floor plane detected is extended using the image data.

4.1 Floor segmentation

Given the range data, we segment it in planes using the plane model and RANSAC algorithm. Once the planes have been detected, we identify the most important scene planes analyzing the normal vector of each plane and considering that the scene follows the Manhattan World [27] model that assumes that the environment has three main directions which are orthogonal between them. We are able to assign the labels *obstacles* or *floor* to the data. Fig. 2 shows images of the steps of this process.

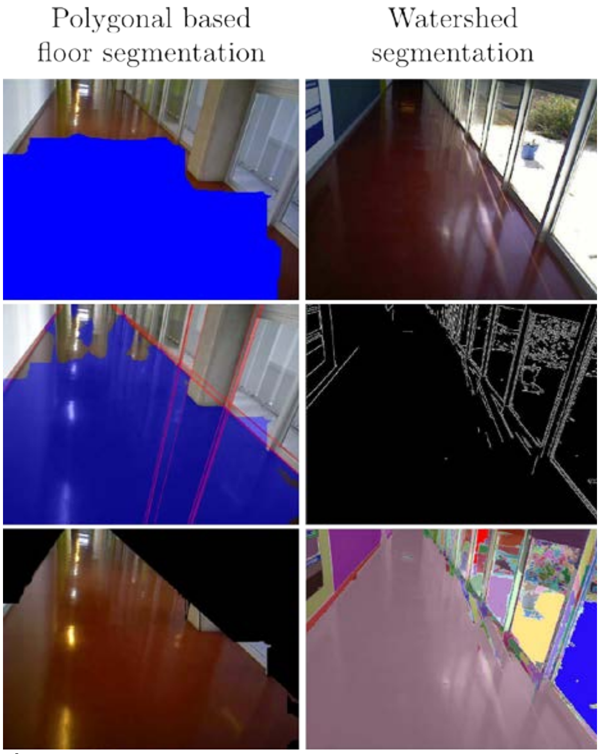


Fig. 2 Process of the segmentation methods used.

4.2 Floor expansion using image information

The maximum reliable distance of the acquired range data is around 3,5 m, enough for obstacle avoidance but not enough for path planning in the guiding assistance. To extend the floor detected and obtain the whole surface of the traversable ground we include the color information. Range data and the RGB data are calibrated, so the detected floor can be projected in the image. We refer to the image projection of the detected floor plane as floor-seed. Starting from this floor-seed region, we will segment the image surface to expand the floor detected in the range data. Two image segmentations will be used depending on the image: Polygonal floor segmentation and Watershed segmentation.

4.2.1 Polygonal floor segmentation. This method uses the lighting, hue and image geometry to segment the image. First, the image is filtered using the shift mean algorithm over a pyramid of images. The result of this step is a smoothed image, where the floor surface is more homogeneous than in the original image while the boundaries with the obstacles are preserved. Next,

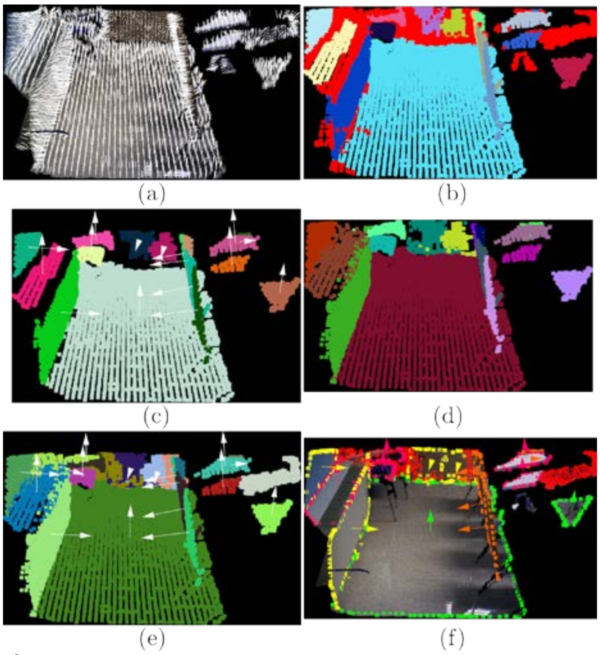


Fig. 4 Planes segmentation and classification for stair detection.

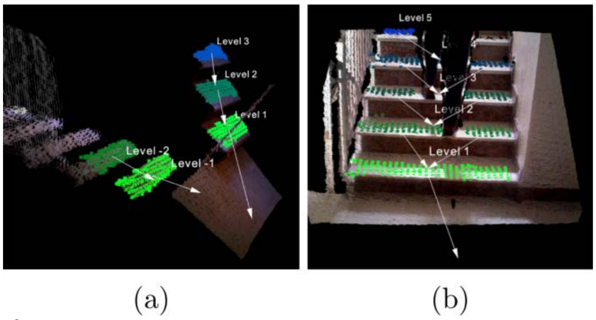


Fig. 5 Connectivity between step candidates: ascending and descending staircases (a), and more than one region per level (b).

we compare the lighting of the filtered image with the lighting of the floor-seed. This is done by comparing the lighting histograms. Pixels satisfying the lighting criteria are then evaluated using a hue criteria. This criteria uses Back Projection to check how well the checked pixels fit the distribution of the hue histogram of the floor-seed. These two criteria allow to select the image pixels with high probability of being part of the floor plane given its lighting and hue values. The final step of this method is a polygonal segmentation. Lines in the image are computed using the Canny edge detector and Hough line transform. Detected lines are extended to image borders and the image is segmented using the polygons defined by these lines. The whole process is shown in the first column of Fig. 3.

4.2.2 Watershed segmentation. When the number of detected lines is too low, too high or the line distribution in the image is too heterogeneous, Watershed segmentation [28] is used. This algorithm takes the binary image resulting from the Canny edge detector as input and produces an image segmentation based in this information. Second column in Fig. 3 shows the process of this method.

Once the image has been segmented with one of the two methods, we use the reference floor-seed area to determine which regions belong to the floor. Segments overlapping with the floor-seed and not overlapping with any obstacle are labeled as floor.

Our method is able to select between both segmentation methods automatically by evaluating if the detected lines are enough to run the polygonal segmentation.

5. Stairs detection

The stairs detection is performed in the range data provided by the sensor. The whole sensor reading is reoriented using the floor plane detected using the process described in Section 4. The origin of coordinates is defined in this plane: y-axis is defined in the plane normal direction, and height 0 is set on the plane surface.

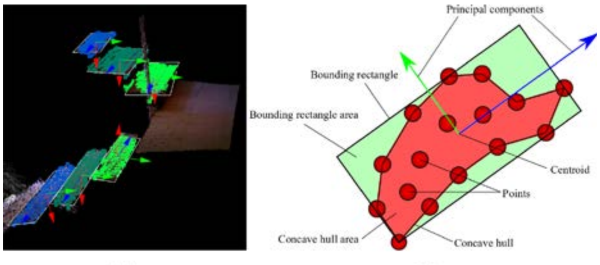
5.1 Segmentation of the scene

A region growing strategy is used to segment the range data. regions are afterwards classified as planar and non-planar using RANSAC. Following this process the planes found are closed regions corresponding to one single element, not a set of uncorrelated points in the scene [24]. The segmentation is performed following the next steps:

Normal estimation (Fig. 4a): For each point and a group of K neighbors, the third component obtained from Principal Component Analysis corresponds to the normal direction. In this step the curvature of the surfaces is also computed.

Region-growing (Fig. 4b): This algorithm starts from a seed, which is the point with minimum curvature, and then expands the region towards the neighboring points that have small angle between the normal and similar curvature value. Points that satisfy the normal and curvature threshold became the new seeds and repeats until the region cannot expand anymore.

Planar test (Fig. 4c): Region-growing produces regions that have a high degree of flatness, but they can also be a curved surface with smooth transitions. RANSAC algorithm seeks for the biggest plane in each region: 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 obstacles.



(a) (b)

Fig. 6 (a) Principal components of the steps (blue-green-red) and the bounding rectangles (white). (b) diagram representing the components.

Planes extension (Fig. 4d): Points not belonging to any region are included in a planar region if the angle between their normal and the planar region normal and their distance to the plane are small.

Euclidean cluster extraction (Fig. 4e): The points not belonging to any region go through a cluster extraction algorithm which establishes connections and forms separate entities, considered obstacles.

Plane classification (Fig. 4f): Once the segmentation stage has succeeded the planes are classified among different classes according to the orientation and relative position of the planes. Planes' normal are compared to the floor normal to detect horizontal and vertical surfaces (walls). Any plane not considered as vertical or horizontal is classified as obstacle.

Horizontal planar regions can be floor, steps or other obstacles. Planar regions with height close to zero are considered *floor*. Regions with positive or negative height that satisfy the Technical Edification Code[1] (13 cm $\leq$ height $\leq$ 18.5 cm) are considered as *step candidates*. The existence of at least one *step candidate* activates the stair detection algorithm.

5.2 Stair detection algorithm

The detection algorithm establishes connections between the *step candidates* to group the stair planes in levels and discard the candidates that do not belong to the staircase. *Step candidates* are analyzed one by one starting from a first step: step candidates whose centroid distance to the floor is below a threshold are considered *first step candidates*. Starting from these *first step candidates*, the connectivity with other *step candidates* is checked using neighbor search and Kd-trees. The first step must be connected to the floor. If no *first step candidate* is detected, the algorithm concludes that there is no staircase.

[1] Ministerio de Fomento. Gobierno de España - Código Técnico de la Edificación, Documento Básico de Seguridad de Utilización y Accesibilidad (DB-SUA, Sección 4.2)

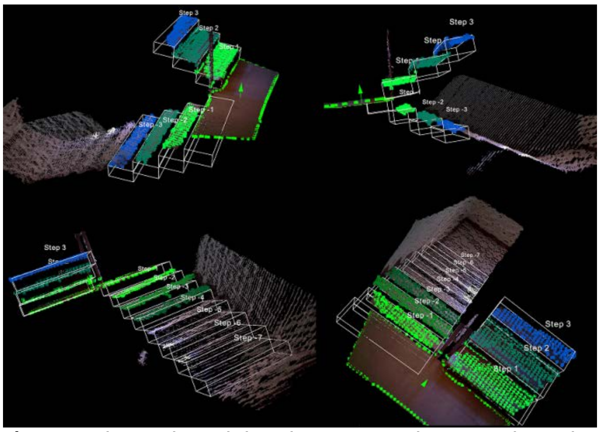


Fig. 7 Estimated model staircase. Top images show the parallelograms corresponding to the found steps. Bottom images show all the steps.

A special case occurs when there is just one step. In this case, strict area and shape conditions need to be verified.

As a result of the stair detection algorithm, a set of connected regions corresponding to different levels is obtained (Fig. 5). When all the candidates have been checked, if the number of stair levels is greater than one, we proceed to model the staircase.

5.3 Stair modeling

Our staircase model uses the next parameters: step width, tread length, riser height and number of steps. We apply Principal Component Analysis (PCA) to each set of points corresponding to the tread of the step in each level of the staircase to compute the width, length and height of each step (Fig. 6(a)) and define the 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. 6(b)).

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 analysis is repeated for all the stair levels, considering the addition of different regions at the same level to form a unique 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. At each level, we will choose the best step as the one with higher extent value among the steps within a valid width range.

Once all the levels have been analyzed, the staircase is modeled. Steps are then modeled as parallelograms whose width is the width of the best step, the height is the average vertical distance between

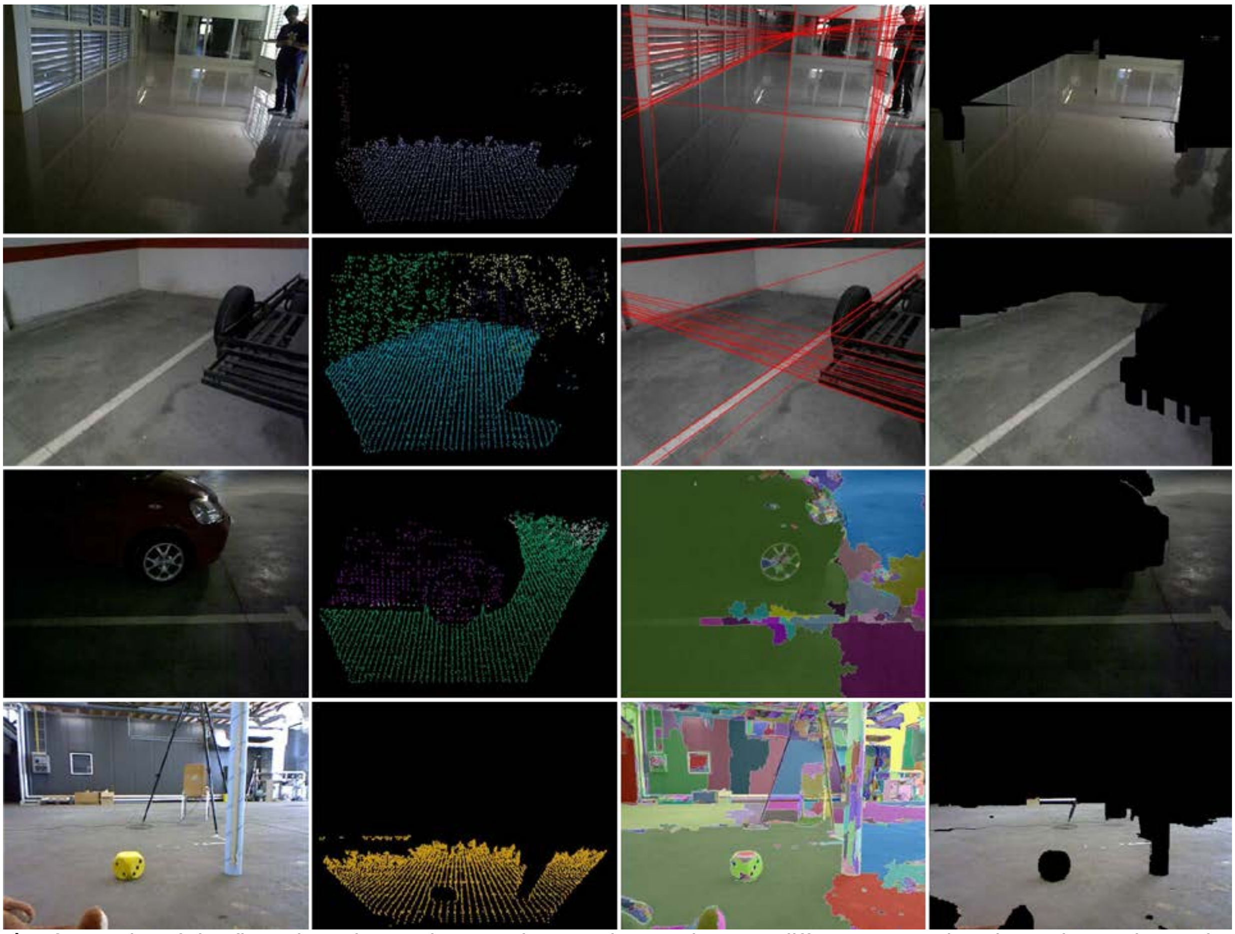


Fig. 8 Results of the floor detection and expansion. Each row shows a different example. First column shows the original RGB image. Second column shows the floor detected in the range data. Third column shows the image segmentation used that is chosen automatically for each image. Finally, fourth column, shows the complete detected floor which is the traversable area to be used to guide the user.

consecutive steps and the length the mean horizontal distance between the edge of every two consecutive steps. 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. 7).

6. User interface

Finally, we propose a simple interface that gives information to the user according to the results provided by the presented algorithms. This interface provides audio instructions and sound map information. Audio instructions will be used only for high level commands, available free path information, or in dangerous situations, where the user could collide with an obstacle. In this case, the system will warn about the situation and will give the necessary instructions. In the rest of cases, the sound map will send stereo beeps whose frequency depends on the distance from the obstacle to the person. We have defined the safety area from the user to any obstacle as two meters. A known drawback of audio systems is that they may block other natural sounds. However, our system does not provide constantly audio instructions or beeps so the possible blocking of natural sounds will only appear sporadically. The user may also regulate the volume of the system so he could hear natural sounds and audio instructions at the same time.

The interface will produce beep sounds depending on the distance from the user to the obstacle. For example, if the left wall is closer to the user than the right one, the user will hear a high frequency beep in his left ear and a low frequency beep in the right ear. If the wall is placed in front of the person, the beep will be Heard in both ears. These beeps allow the user to understand the environment. With this user interface, the user will be able to navigate through an unknown scenario as well as being able to avoid obstacles with no risk of collision.

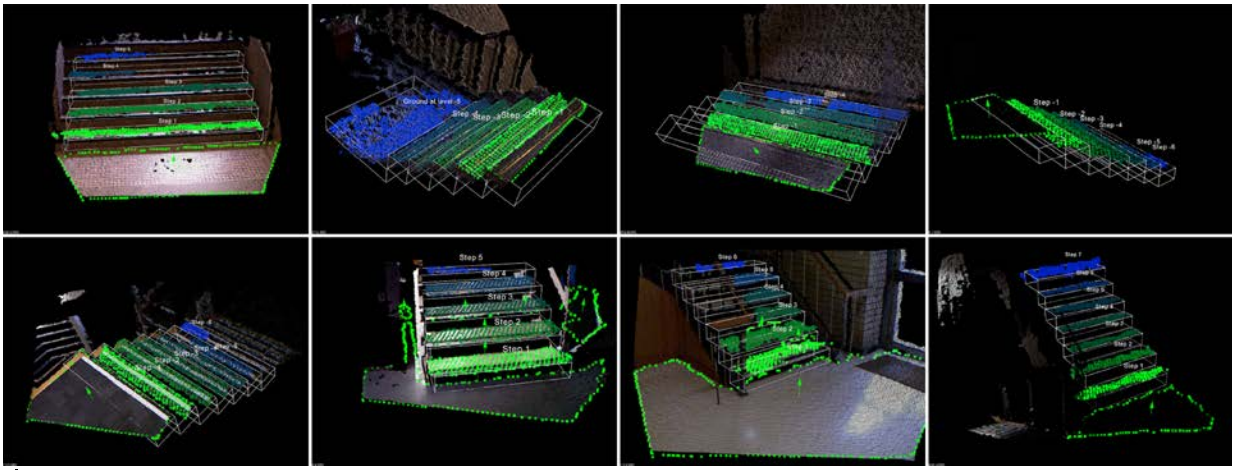


Fig. 9 Results of the stair detection. Last column shows results obtained in dark environments.

7. Experiments

Next sections detail the experiments performed to test the different methods proposed in this work. The methods have been evaluated in real scenarios exhibiting a wide variety of visual characteristics and lighting conditions.

7.1 Datasets used for the experiments

We have tested the algorithm in public and private buildings. The public ones are placed in University of Zaragoza (Spain) and they are: Ada Byron building, Torres Quevedo building and I+D building where Institute of Engineering Investigation of Aragón (I3A) is placed. The private buildings are examples of houses and a garage. Since the number of datasets to test approaches for navigation assistance is almost non-existent we have released our dataset[2], which collects data used in our experiment to be available to the research community. Additionally, scenarios including stairs were also recorded to conduct specific experiments. We have also evaluated our system using the dataset of the Technische Universität München (TUM)[3] and the dataset compiled by Tang et al. compiled in [24].

Table 1 Results of the floor detection evaluated with the annotated ground truth.

Percentages of floor-segmentation with range data				
Scenario	**Precision**	**Recall**	**F1**	**Recall interval**
I3A building	100%	78,62%	87,87%	78,62 ± 4,79 %
Ada Byron bldg.	100%	84,23%	91,43%	84,23 ± 1,08 %
Torres Quevedo	100%	78,95%	88,10%	78,95 ± 3,51 %
Garage	100%	87,63%	93,38%	87,63 ± 1,68 %
München dataset	100%	54,54%	69,01%	54,54 ± 6,82 %
Percentages of floor-segmentation with range and color data				
Scenario	**Precision**	**Recall**	**F1**	**Recall interval**
I3A building	98,94%	96,74%	97,81%	97,00 ± 1,20 %
Ada Byron bldg.	98,97%	95,22%	97,04%	95,00 ± 1,30 %
Torres Quevedo	99,26%	97,38%	98,30%	97,00 ± 1,00 %
Garage	99,62%	93,62%	96,49%	94,00 ± 1,82 %
München dataset	99,09%	96,23%	97,62%	96,00 ± 1,60 %

7.2 Floor and obstacle detection testing

Fig. 8 presents results of our floor detection algorithm on some typical corridor images available in our dataset, and the TUM dataset. Even in the presence of hard conditions (i.e. brightness, reflections), we obtain good results.

A quantitative analysis is shown in Table 1. This table shows the performance of floor detection obtained just with range data and when the whole system is used. For these results, the floor of 150 images has been manually labeled. Table shows precision, recall and F1 statistic values. The recall confidence interval is also computed in the last column at the 95% confidence level.

The precision obtained with range data is 100% in all scenarios. These perfect precisions are caused because of short-range hardware limitations and because the range sensor is unable to obtain range data of regions which are closed to an object's boundary, producing conservative results. On the other hand, recall has low values due to these limitations.

The best recall results using just the range data correspond to sequences where there is no sun light (Garage and Ada Byron bldg.). However, for the rest of sequences the results are weak. Is in those sequences where the use of both range and image data advantages are shown. Range segmentation is limited due to solar light so recall is lower than 80% (55% for the TUM dataset). Adding the color information improves the recall to 95%.

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

[3] http://vision.in.tum.de/data/datasets/rgbd-dataset

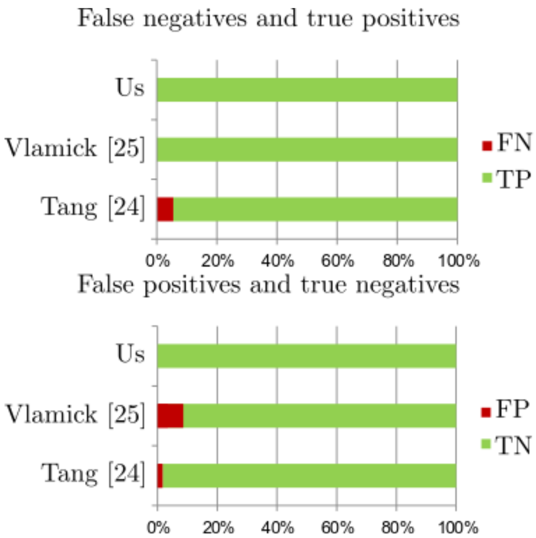


Fig. 10 Comparison of false negatives and false positives between our work and the approaches presented by [24, 25].

Table 2 shows the contribution of each part of the algorithm, range segmentation and color segmentation, to the final floor result. In order to obtain a fair comparison in metric units, we need to project the image's floor without projective distortion to have a top view of it in real magnitude. Otherwise, the farther the segmented region is in the projective image, the less number of pixels it contains (despite representing similar metric area tan closer regions). We have calculated the homography from the image to the floor and we have obtained the number of squared meters segmented by range and color algorithms. Table 2 shows that the expansion of the range segmentation with color segmentation is important in all scenarios. Scenarios where there is no solar light have the highest contribution of range segmentation. Scenarios with medium-low solar light incidence we obtain a contribution of 50% approximately with both kind of segmentations. Those scenarios where the presence of solar light is really high, the color segmentation has the highest contribution, more than 70% of the detected floor is obtained with this part of our algorithm, reducing drastically the limitations of the range data.

7.3 Stair detection experiments

To test the stair detection algorithm we use the Tang dataset. The results with this dataset were successful even in total darkness (Fig. 9). We tested for false positives and false negatives using this dataset and compared our results with methods in [24] and [25] (Fig. 10). We achieve better results with no false negatives as in [25] but also without false positives.

If we look at the step detection rate according to the position of the step in the staircase (Fig. 11) we see how behavior changes when facing ascending staircase

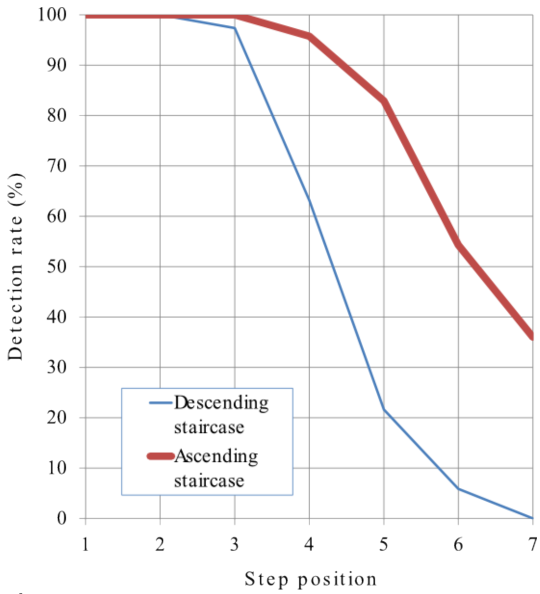


Fig. 10 Step detection rate with the step position in the staircase.

or descending steps. When the user faces a descending staircase the whole staircase can be seen by the sensor, but self occlusion of consecutive steps and quality of the measurements decreases with the distance so the rate detection of further steps decreases. In the case of ascending staircases the steps remain close and visible for the sensor as they rise, although visual angle decreases. In general, steps higher than the seventh position are out of the field of view of the camera.

We have quantitatively analyzed 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 staircases, in both ascending and descending perspectives, from different viewing angles, the results were compared to the real measurements. Real stairs have a length of 30cm and a height of 15cm. The mean values for the computed length and height where 29 cm and 15.4 cm respectively. 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, length and height were 29.39 cm and 15.56 cm respectively in these cases. Some pictures of the experiments with people climbing up/down the

Table 2 Contribution to the final result of the range and color segmentation.

Scenario	Range segmentation	Color segmentation
I3A building	26,53%	73,47%
Ada Byron building	43,34%	56,66%
Torres Quevedo building	54,92%	45,08%
Garage	74,22%	25,78%
München dataset	52,62%	47,38%

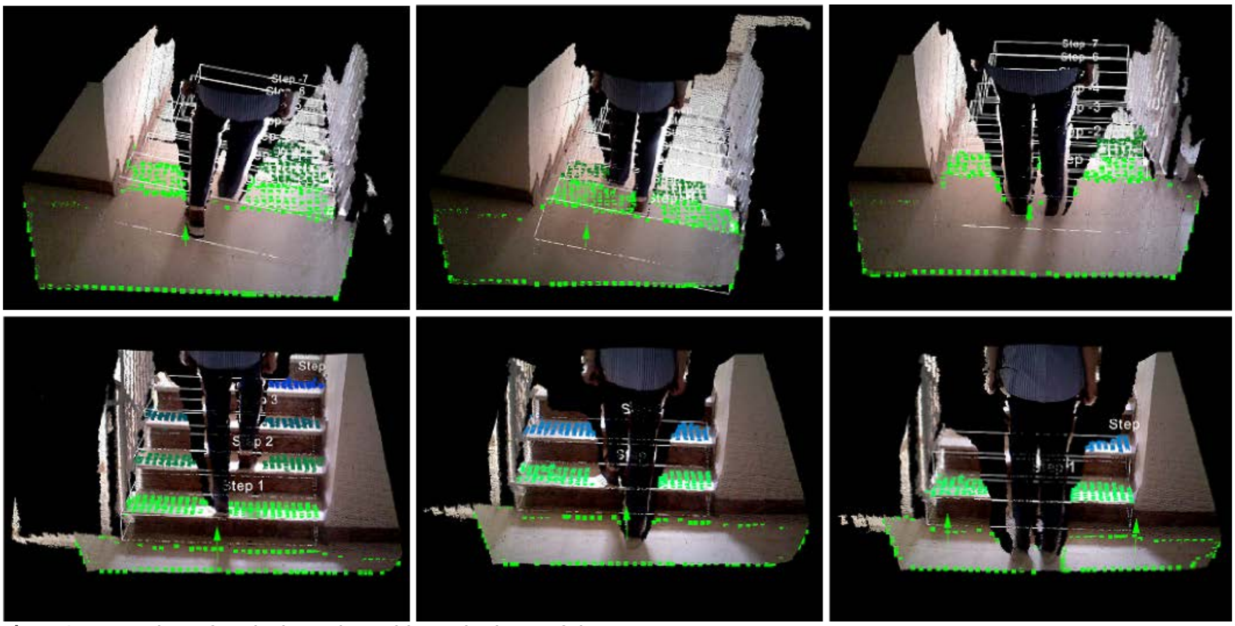


Fig. 12 Examples of stair detection with occlusions of the steps.

staircase can be seen in Fig. 12.

7.4 Computation time

One important point of a navigation system is that it has to be able to run on real time, while the user moves. We have tested the computation times of the method proposed. The whole floor and obstacle detection algorithm (Range data processing, RGB image processing and user interface generation) runs approximately at 0.3 frames/s. The stair detection iteration time ranges from 50 to 150ms. The variation depends on the scene itself: close up captures provides good quality clouds and the segmentation algorithm provides less regions and as a consequence, faster results.

We consider this timing fast enough for indoor navigation assuming walking speeds around 1-1:5m/s. This rate could be improved adding some optimizations to the algorithm or using multi-core processing.

8. Conclusions

In this work we have presented a navigation assistance prototype able to guide a person through an unknown indoor environment avoiding obstacles and detecting staircases. The system uses a chest mounted RGB-D camera that captured the relevant information of the scene without intruding the privacy of nearby people.

The prototype uses the data captured by the sensor to detect the floor plane and close obstacles, and the staircases visible. Floor and obstacles are detected in the range data, allowing to navigate safely in the area close to the user. The floor detected in the range data is extended later in image using the color information. Additionally, the environment is analyzed in the search of staircases close to the user. This analysis is performed also on the range data.

The system has been tested in different real environments showing good better performance than other state-of-the-art techniques and the computations can be run on real time.

9. Acknowledgments

This work has been supported by Spanish project DPI2012-31781.

10. Bibliography

[1] Aladren, A., Lopez-Nicolas, G., Puig, L., & Guerrero, J. J. (2014). 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

[2] Pérez-Yus, A., López-Nicolás, G., & Guerrero, J. J. (2015). Detection and Modelling of Staircases Using a Wearable Depth Sensor. ECCV 2014 Workshops Lecture Notes in Computer Science Volume 8927, pp 449-463

[3] Dakopoulos, D., & Bourbakis, N. G. (2010). Wearable obstacle avoidance electronic travel aids for blind: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 40(1), 25-35.

[4] Guimaraes, C. S. S., Bayan Henriques, R. V., & Pereira, C. E. (2013). Analysis and design of an embedded system to

aid the navigation of the visually impaired. In Biosignals and Biorobotics Conference (BRC), ISSNIP (pp. 1-6).

[5] Seto, T., & Magatani, K. (2009). A navigation system for the visually impaired using colored navigation lines and RFID tags. In Annual International Conference of Engineering in Medicine and Biology Society (EMBC) (pp. 831-834).

[6] Joseph, S. L., Zhang, X., Dryanovski, I., Xiao, J., Yi, C., & Tian, Y. (2013). Semantic indoor navigation with a blind-user oriented augmented reality. In International Conference on Systems, Man, and Cybernetics (SMC), (pp. 3585-3591).

[7] Liu, J. J., Phillips, C., & Daniilidis, K. (2010). Video-based localization without 3D mapping for the visually impaired. In Computer Vision and Pattern Recognition Workshops (CVPRW) (pp. 23-30).

[8] Wong, F., Nagarajan, R., & Yaacob, S. (2003). Application of stereovision in a navigation aid for blind people. In Joint Conference of Information, Communications and Signal Processing, and Pacific Rim Conference on Multimedia. (Vol. 2, pp. 734-737).

[9] Takizawa, H., Yamaguchi, S., Aoyagi, M., Ezaki, N., & Mizuno, S. (2013). Kinect cane: Object recognition aids for the visually impaired. In International Conference on Human System Interaction (HIS) (pp. 473-478).

[10] Wang, Z., Liu, H., Wang, X., & Qian, Y. (2014). Segment and label indoor scene based on RGB-D for the visually impaired. In MultiMedia Modeling (pp. 449-460).

[11] Peasley, B., & Birchfield, S. (2013). Real-time obstacle detection and avoidance in the presence of specular surfaces using an active 3D sensor. In Workshop on Robot Vision (WORV) (pp. 197-202).

[12] Schafer, H., Hach, A., Proetzsch, M., & Berns, K. (2008). 3D obstacle detection and avoidance in vegetated off-road terrain. In International Conference on Robotics and Automation (ICRA) (pp. 923-928).

[13] Lee, Y. H., & Medioni, G. (2011). A RGB-D camera based navigation for the visually impaired. In RSS 2011 RGBD: Advanced Reasoning with Depth Camera Workshop (pp. 1-6).

[14] Gutiérrez-Gómez, D., Mayol-Cuevas, W., & Guerrero, J. J. (2025). Inverse Depth for Accurate Photometric and Geometric Error Minimisation in RGB-D Dense Visual Odometry. In International Conference on Robotics and Automation (ICRA).

[15] Ray, A. K., Behera, L., & Jamshidi, M. (2008). Sonar-based rover navigation for single or multiple platforms: forward safe path and target switching approach. IEEE Systems Journal, 2(2), 258-272.

[16] Li, Y., & Birchfield, S. (2010). Image-based segmentation of indoor corridor floors for a mobile robot. In International Conference on Intelligent Robots and Systems (IROS) (pp. 837-843).

[17] Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., & Bradski, G. R. (2006). Self-supervised Monocular Road Detection in Desert Terrain. In Robotics: science and systems (RSS)

[18] Dal Mutto, C., Zanuttigh, P., & Cortelazzo, G. M. (2012). Fusion of geometry and color information for scene segmentation. Selected Topics in IEEE Journal of Signal Processing, 6(5), 505-521.

[19] Hesch, J. A., Mariottini, G. L., & Roumeliotis, S. I. (2010). Descending-stair detection, approach, and traversal with an autonomous tracked vehicle. In International Conference on Intelligent Robots and Systems (IROS) (pp. 5525-5531).

[20] Lu, X., & Manduchi, R. (2005). Detection and localization of curbs and stairways using stereo vision. In International conference on Robotics and Automation (ICRA) (Vol. 4, p. 4648).

[21] Oßwald, S., Hornung, A., & Bennewitz, M. (2012). Improved proposals for highly accurate localization using range and vision data. In International Conference on Intelligent Robots and Systems (IROS) (pp. 1809-1814).

[22] Filipe, V., Fernandes, F., Fernandes, H., Sousa, A., Paredes, H., & Barroso, J. (2012). Blind navigation support system based on Microsoft Kinect. Procedia Computer Science, 14, 94-101.

[23] Wang, S., & Tian, Y. (2012). Detecting stairs and pedestrian crosswalks for the blind by rgbd camera. In International Conference on Bioinformatics and Biomedicine Workshops (BIBMW) (pp. 732-739).

[24] Tang, T. J. J., Lui, W. L. D., & Li, W. H. (2012). Plane-based detection of staircases using inverse depth. In Australasian Conference on Robotics and Automation (pp. 1-10).

[25] Vlaminck, M., Jovanov, L., Van Hese, P., Goossens, B., Philips, W., & Pizurica, A. (2013). Obstacle detection for pedestrians with a visual impairment based on 3D imaging. In International Conference on 3D Imaging (IC3D).

[26] Mayol-Cuevas, W. W., Tordoff, B. J., & Murray, D. W. (2009). On the choice and placement of wearable vision sensors. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 39(2), 414-425.

[27] Coughlan, J. M., & Yuille, A. L. (1999). Manhattan world: Compass direction from a single image by bayesian inference. In IEEE International Conference on Computer Vision (ICCV) (Vol. 2, pp. 941-947).

[28] Deng, Y., Manjunath, B. S., & Shin, H. (1999). Color image segmentation. In Conference on Computer Vision and Pattern Recognition (CVPR) (Vol. 2).