Universidad
Zaragoza
1542

Trabajo Fin de Máster

Detección y modelado de escaleras con sensor RGB-D para asistencia personal

Autor/es

Alejandro Pérez Yus

Director/es

José Jesús Guerrero Campo
Gonzalo López Nicolás

Escuela de Ingeniería y Arquitectura

2014

Resumen

La habilidad de avanzar y moverse de manera efectiva por el entorno resulta natural para la mayoría de la gente, pero no resulta fácil de realizar bajo algunas circunstancias, como es el caso de las personas con problemas visuales o cuando nos movemos en entornos especialmente complejos o desconocidos. Lo que pretendemos conseguir a largo plazo es crear un sistema portable de asistencia aumentada para ayudar a quienes se enfrentan a esas circunstancias. Para ello nos podemos ayudar de cámaras, que se integran en el asistente. En este trabajo nos hemos centrado en el módulo de detección, dejando para otros trabajos el resto de módulos, como podría ser la interfaz entre la detección y el usuario.

Un sistema de guiado de personas debe mantener al sujeto que lo utiliza apartado de peligros, pero también debería ser capaz de reconocer ciertas características del entorno para interactuar con ellas. En este trabajo resolvemos la detección de uno de los recursos más comunes que una persona puede tener que utilizar a lo largo de su vida diaria: las escaleras. Encontrar escaleras es doblemente beneficioso, puesto que no sólo permite evitar posibles caídas sino que ayuda a indicar al usuario la posibilidad de alcanzar otro piso en el edificio.

Para conseguir esto hemos hecho uso de un sensor RGB-D, que irá situado en el pecho del sujeto, y que permite captar de manera simultánea y sincronizada información de color y profundidad de la escena. El algoritmo usa de manera ventajosa la captación de profundidad para encontrar el suelo y así orientar la escena de la manera que aparece ante el usuario. Posteriormente hay un proceso de segmentación y clasificación de la escena de la que obtenemos aquellos segmentos que se corresponden con "suelo", "paredes", "planos horizontales" y una clase residual, de la que todos los miembros son considerados "obstáculos". A continuación, el algoritmo de detección de escaleras determina si los planos horizontales son escalones que forman una escalera y los ordena jerárquicamente. En el caso de que se haya encontrado una escalera, el algoritmo de modelado nos proporciona toda la información de utilidad para el usuario: cómo esta posicionada con respecto a él, cuántos escalones se ven y cuáles son sus medidas aproximadas.

En definitiva, lo que se presenta en este trabajo es un nuevo algoritmo de ayuda a la navegación humana en entornos de interior cuya mayor contribución es un algoritmo de detección y modelado de escaleras que determina toda la información de mayor relevancia para el sujeto. Se han realizado experimentos con grabaciones de vídeo en distintos entornos, consiguiendo buenos resultados tanto en precisión como en tiempo de respuesta. Además se ha realizado una comparación de nuestros resultados con los extraídos de otras publicaciones, demostrando que no sólo se consigue una eficiencia que iguala al estado de la materia sino que también se aportan una serie de mejoras. Especialmente, nuestro algoritmo es el primero capaz de obtener las dimensiones de las escaleras incluso con obstáculos obstruyendo parcialmente la vista, como puede ser gente subiendo o bajando.

Como resultado de este trabajo se ha elaborado una publicación aceptada en el *Second workshop on Assitive Computer Vision and Robotics* del *ECCV*, cuya presentación tiene lugar el 12 de Septiembre de 2014 en Zúrich, Suiza.

Abstract

The ability of navigating effectively in the environment is natural for people, but not easy to achieve under certain circumstances, such as the case of visually impaired people or when moving at unknown and intricate environments. Our goal in the long term is building a wearable enhanced assistance system to help the people under that circumstances. To deal with this problem we can count on the help of instruments such as cameras, laser scans or other type of sensors and integrate them into the wearable navigation assistant. In this work we have focused on the detection module of the assistant, leaving the rest of the modules such as the interface between the detection and the user to other works.

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 work we propose an algorithm which outputs the region of space available to walk but also 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", "horizontal planes" and a residual class where all members are considered "obstacles". Then the stairs detection algorithm outputs if the horizontal planes 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 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.

As a result of this work there has been a research paper accepted in the *Second Workshop on Assistive Computer Vision and Robotics* of the *ECCV*, whose presentation takes place the 12th of September of 2014 in Zurich, Switzerland.

Contents

Chapter 1

Introduction

In this work we deal with the perception task of a wearable navigation assistant. 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 fast enough for normal usage and works even under partial occlusions of the stairway.

In Section 1.1 we talk about the motivation that encourage us to do this work. Then, in Section 1.2 a review of the different solutions to this problem presented through the years is shown. In Section 1.3 we point out the goals we set when we started to work on this line. Finally, in Section 1.4 the structure of the work is explained, to conclude with a brief comment about our publication related to this work's topic in Section 1.5.

1.1 Motivation

According to the World Health Organization (WHO), in 2012 there were 285 millions of visually impaired people worldwide: 39 million are blind and 246 have low vision. This large amount of people is not negligible and shows the need of portable, practical, and highly functional assistive devices to help out people under this circumstances. The WHO also affirms that the 90% of the world's visually impaired live in low-income settings and the 82% of people living in blindness are aged 50 or above. This means that the devices we pursue to develop should not be high priced or technologically complex.

When thinking about building a platform to help visually impaired people we have to think about the information the user might want or need to receive. Despite their lack of sight, as human beings they can still use their own senses and experiences in many tasks. They also have always had two low-tech level aids:

the white cane and the guide dog. Our research focus in trying to make their life more comfortable, developing a system that complements rather than replaces, increasing and improving the amount of information they can already receive by their own means. For this task we believe the best type of sensor are cameras, due to their portability, price and amount of information they can provide.

Their limitations are bigger when they are in an unknown environment with hardly any previous notion about it. They might know what they want to do but sometimes they will not be able to do it unless they get external help. E.g., they are in a hospital and they want to go to the second floor. If that is the first time they visit that hospital they do not know where the stairs or the elevator might be. A wearable vision-based system like the one we are trying to develop would be extremely helpful in that kind situation. It would also be useful avoiding obstacles, recognizing objects, people, traffic signs, and so on.

But the usage of cameras for this purpose poses some special challenges due to its nature:

- The images that can be acquired during the normal use are highly variable and it is not an easy task to develop an algorithm able to understand the scenes everywhere.
- The changeable lightning conditions or presence of occluding objects harms the recognition of the environment.
- The quality of the images may be out of focus, poorly framed or motion-blurred, making more difficult to do a proper understanding of the scene.
- Sometimes computer vision algorithms are highly time consuming, but for this task they must execute in real time in order to be useful.

Some special cameras can defeat some of this limitations. That is the case, for example, of the omnidirectional cameras, which can capture the whole scene around the user, solving the problem of pointing at the right direction. Additionally, the modern RGB-D cameras provide depth perception of the scene besides the traditional color image, in a synchronized and calibrated way, allowing the developers to have the advantages of both types of data at the same time. Having 3D perception can help recognizing the basic structures of the environment as well as detecting obstacles along the path, enabling a more robust and secure navigation for the user. On the other hand, the poor performance of the depth sensor when it is used outdoors restricts the usage to indoor environments.

In this work we want to take advantage of the RGB-D sensors to develop the perception module of a human navigation assistant, able to recognize basic structures such as walls or floor and to detect obstacles. 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. Specifically, there are two basic structures that every person uses everyday and that are of great significance in indoor environments: doors and stairs. Here we will deal with stairs, leaving the door detection for future works.

Finding stairs along the path has the double benefit of preventing falls and advertising the possibility of reaching another floor in the building. As we mentioned before, we have to think about the information the user might want or need to receive. We consider that the presence of a stair, the location, the orientation, the inclination (up/down), the dimensions of the steps and the number of steps are the basic information the user needs to traverse a staircase. That is why our work goes beyond simple detection and try to model the whole staircase. But as any navigation system, it must be reliable and robust, and that is why we have paid special attention to stair detection and modelling with partial occlusions, being this feature the main novelty of this work.

Although our main concern appears to be the case of visually impaired people, we have to point out that what we plan to develop in the long term can also be applied to many other tasks that involves special visual needs, such as firefighters, policemen, tourists, surgeons and also it can be used for educational or entertainment purposes. It can have its application in robotics too, specially in the case of humanoid robots.

1.2 State of the Art

There has been many publications related to this work's topic through the years, concerning assistance to the visually impaired, scene understanding and stairs detection. Different type of sensors and methods have been used to perform this task. RGB-D cameras are becoming the trending choice among the sensors as since 2010 cheaper devices have been launched to the market, e.g., Microsoft Kinect or Asus Xtion. Despite this tendency, some limitations of this type of cameras, such as the poor performance of the depth sensor outdoors, leaves a gap where conventional cameras can assist. In the next sections we are going to discuss some of the published work related to the topics aforementioned.

1.2.1 Visually Impaired Aids

The study of indoor navigation problems results interesting to two different fields such as robotics and the development of impaired people aids. Although both fields share some common goals, the algorithms to be developed differ in means and purposes to achieve effective navigation. There are two important things to take into account when designing a wearable system, that are: where is the sensor placed and how is the information transferred to the user. Mayol-Cuevas et al. [33] presented an extensive research about the first issue. In Fig. 1.1 some usual configurations are displayed. In our case, solution D is what we chose for our assistant. The reasons for that will be explained later in this work.

About the user interface, our group -as many others- has used audio responses in previous works [1], although vibratory or haptic responses are also widely chosen, even combined. Dakopoulos et al. in 2010 published a survey comparing some of the existing electronic travel aids for the blind in [11] (Fig. 1.2). Despite

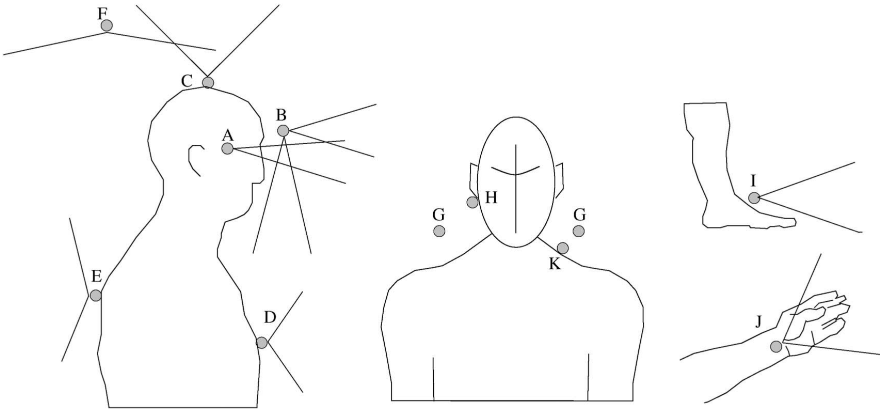


Figure 1.1: Different choices of placements for wearable vision sensors as compiled by [33]

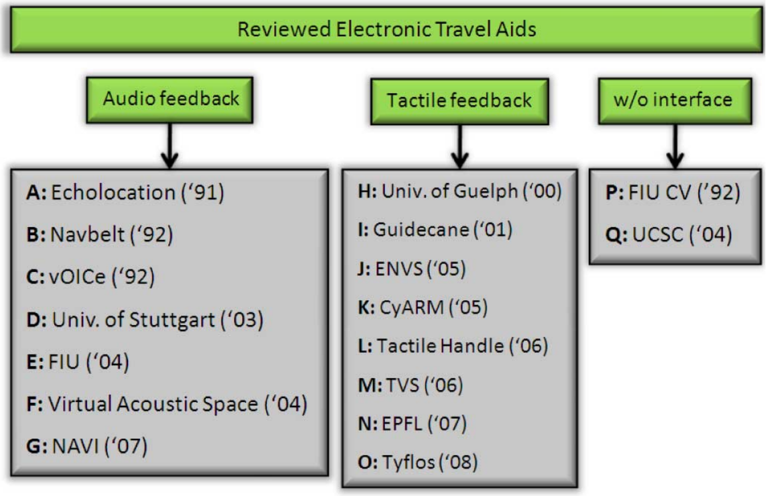


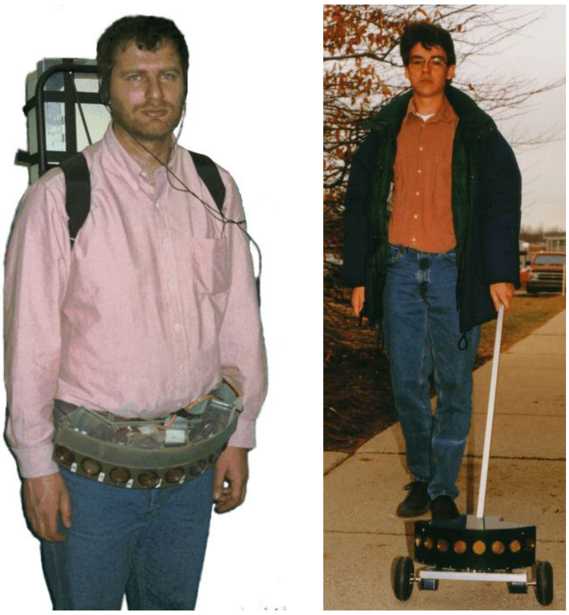
Figure 1.2: Classification of feedback methods in visually impaired aids performed by Dakopoulos et al. in [11]

the necessity of developing an interface, the way of receiving feedback from the system has not been addressed yet and it is no subject for this work.

In the following sections some examples of assistants for the visually impaired people that have been developed in the last years are shown, classified by type of sensors perceiving the information.

1.2.1.1 Assistants without Vision

The first technological aids developed for the blind are mainly sonar-based, but they did not cause a major breakthrough in the community beyond pure research. Although they help detecting obstacles, the poor angular resolution and the limited information they provide is not enough to have great market acceptance. In that direction, researchers from the University of Michigan proposed two sonar-based solutions: the NavBelt [50] and the GuideCane [7] whose prototypes are shown in Fig. 1.3. In 2000, Shoval et al. compared both systems pointing out that GuideCane represents a more helpful proposition in terms of interface because it directs the user as it moves itself before him, like a guide dog [51]. Radar systems perform better than ultrasonic sensors, but they are more complex and expensive and they may suffer from interference problems with other signals inside buildings.



(a) NavBelt (b) GuideCane

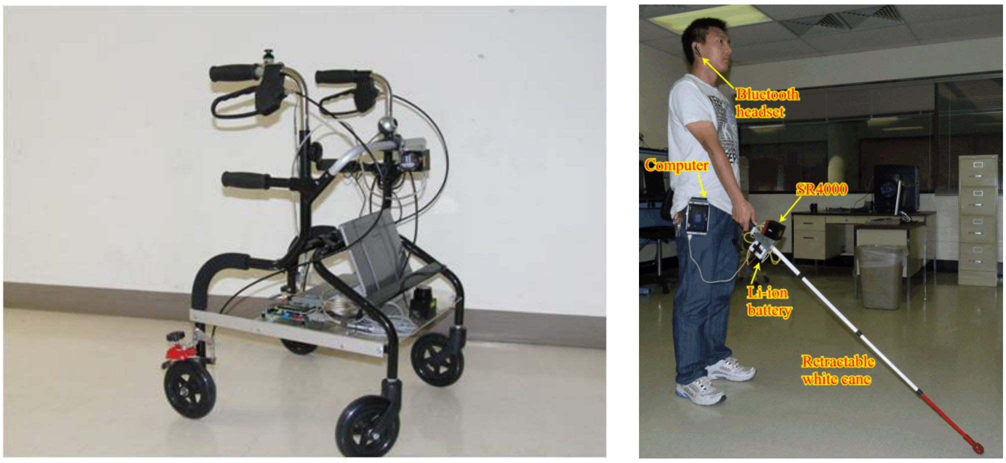
Figure 1.3: Prototypes of visual impaired aids from the University of Michigan

Laser sensors provide good and accurate information, but they are expensive, heavy and involve high power requirements, so they are not the best option for human wearable applications. One example of wearable system to aid blind people is shown in [55], where the main sensor is a 3D laser scanner chest-mounted. Laser scanning is also used in [9], where they propose a robotic system consisting on one PC, two laser range finder, a camera, a microprocessor and a joystick potentiometer, all of it mounted in a trolley walker (Fig. 1.4a). Besides obstacle detection, their algorithm includes face/human detection and steps detection. In [44] the laser sensor is attached to a retractable white cane, giving a new approach to the "Smart Cane" concept (Fig. 1.4b).

Some other common sensors used in this context are RFID [27] or GPS, but they require previously prepared environments or to be used outdoors, requirements that make these systems unsuitable to our current needs.

1.2.1.2 Vision-based Assistants

A better choice of type of sensor for this application due to its portability, price and amount of information able to provide is vision. There exists many different types or configurations of cameras (monocular, stereo, omnidirectional, and so on)



(a) (b)

Figure 1.4: Visual impaired aids examples using laser scanning. (a) Trolley walker developed in [9]. (b) Smart Cane developed in [44]

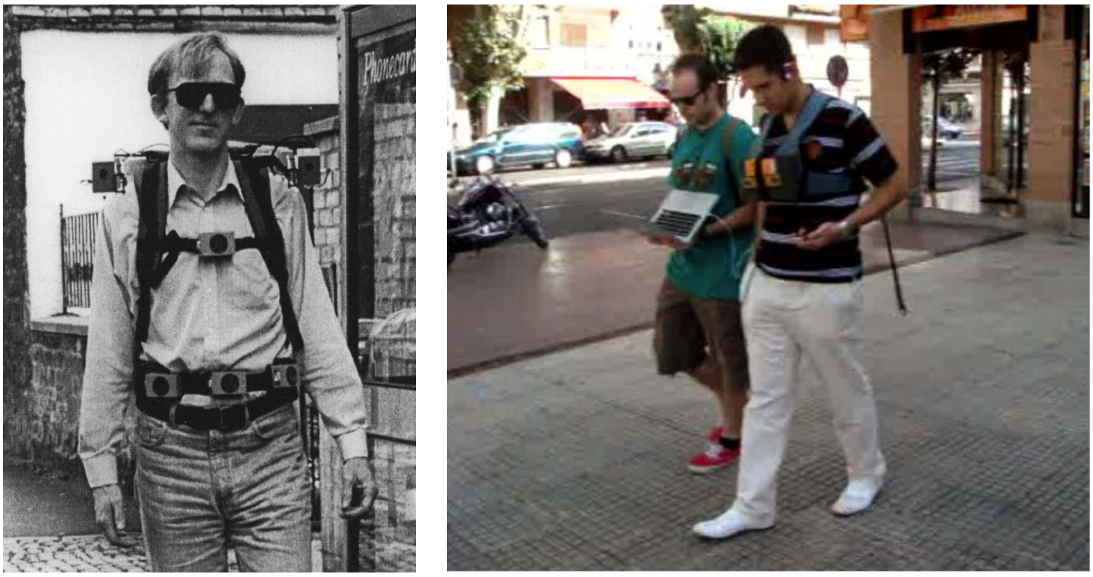
and all of them can have their own applications in this field. Their only drawback is that they may require complex algorithms, but the extensive research in robot applications has already paved the way. An example of how monocular vision can be helpful to detect specific features, such as text signs, doors, elevators or cabinets is shown in [54].

Stereo vision can retrieve 3D information of the scene, which makes easier the detection of obstacles. In 1998, Molton et al. proposed a wearable system composed mainly by a stereo camera to help blind people avoid obstacles [36] (see Fig. 1.5a). The electronics and computing devices as well as the sensors are mounted in a backpack. Specifically, the two cameras that compose the stereo vision system are placed over the shoulders, while on the chest and belt there are some sonar sensors. However, the output they provide is not as purposeful as expected nowadays due to technological constraints.

A more modern approach using a head-mounted stereo vision system is the one shown in [42]. It joins in the same mobility system a SLAM algorithm along with obstacle detection. With the visual odometry and the mapping they perform a traversability analysis of the environment and a tactile interface situated on a vest steers the subjects away from obstacles along the path. The same authors also introduced a module for step detection in [43]. Rodriguez et al. in [46] proposed another obstacle avoidance system using stereo cameras, with acoustic feedback instead of vibration interface (Fig. 1.5b).

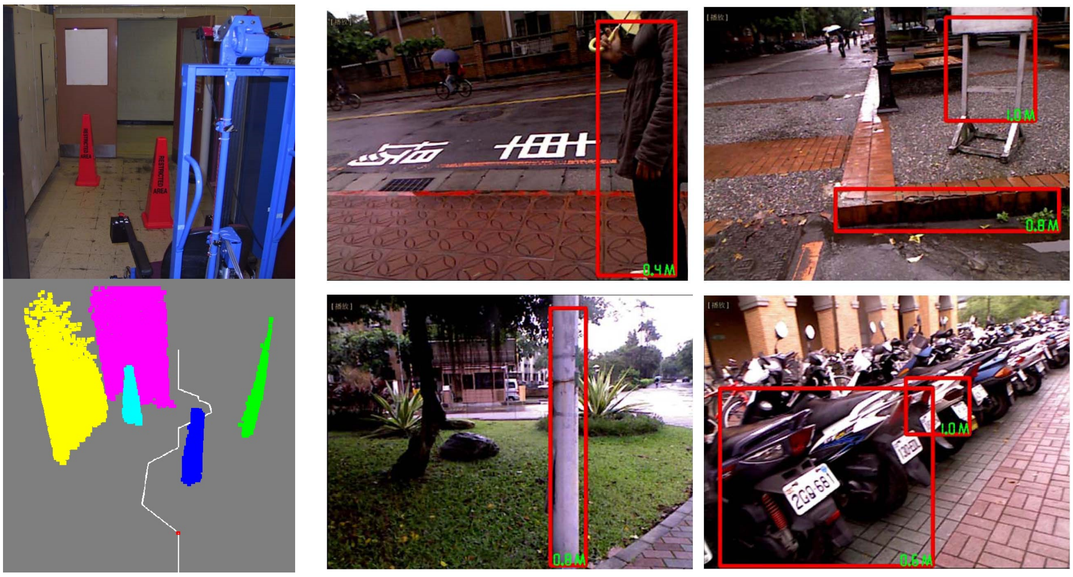
1.2.1.3 Assistants Using Depth Sensing

Stereo vision provides 3D perception but it has some problems capturing depth in texture-less or repeated areas. Range cameras such as Time-Of-flight (TOF) cameras overcome this limitations and work fine in this context for obstacle location, as it can be seen in [8] (Fig. 1.6a) and [28] (Fig. 1.6b).



(a) (b)

Figure 1.5: Visual impaired aids examples using cameras. (a) Stereo vision + sonar developed in [36]. (b) Stereo vision + acoustic feedback developed in [46]



(a) (b)

Figure 1.6: Examples of TOF cameras-based personal guidance systems. (a) Work from [8]. (b) Obstacles detection by [28]

Recently, the introduction in the market of consumer cameras which combine simultaneously range-sensing with conventional image-sensing has made a great impact in what assistance navigation for the visually impaired concerns. They provide a great amount of information, are low-cost and have good miniaturization perspectives. That is the case of [62], which uses a head-mounted configuration or [6], which uses a chest-mounted configuration along with an accelerometer to measure the movement of the subject. [32] introduces the use of a vibrotactile helmet to comunicate with the user besides the head-mounted RGB-D sensor. Also Lee et al. in [29] improved what was published for Pradeep et al. in [42] by replacing the stereo camera with a RGB-D camera. Another example using machine learning techniques for segmentation of the scene is [61].

1.2.2 Scene Understanding

Segmentation and scene understanding have been essential issues in robot and human navigation, because it is not only a matter of avoiding obstacles: in order to perform any relatively complex task it is necessary to recognize the features of your surroundings in order to interact with them. Our case of study is indoor environments, where, like in most human-made scenarios, the basic structure of the scene is made by combination of planes. Range sensors have proven to be extremely helpful for this mission, and many different algorithms have been developed to perform the segmentation. Hoover et al. [24] compiled and performed an evaluation of related early work. Three basic type of approaches are distinguished: RANSAC, Hough Transform and Region Growing.

Rusu et al. [48] introduced a point cloud-based segmentation and mapping of the scene in household environments (Fig. 1.7). Some of the algorithms presented in this article were later included in the Point Cloud Library (PCL) which Rusu et al. presented in [47]. Both the library and some of the algorithms have also been used in our work, such as the normal estimation or region growing. Their acquisition of point cloud data was made with laser scanners and required complex registration operations which can now be avoided with modern RGB-D sensors.

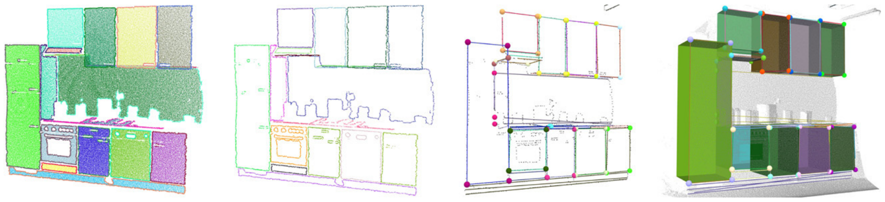


Figure 1.7: Segmentation and mapping in household environments by [48]

Holz et al. proposed an algorithm for real-time plane segmentation using RGB-D-captured data by fast computing of surface normals, clustering and classifying in both normal space and spherical coordinates [23]. Wang et al. proposed another algorithm in [60] which works in two steps: initial segmentation (voxelization and region-growing algorithm) and refined segmentation (plane merging) which proves to be faster than the previously mentioned. Alternatively, Dube et al. [13] used Randomized Hough Transform for the same task. Recently, Holz et al. proposed a new algorithm for fast range image segmentation and smoothing using approximate surface reconstruction and region growing [22]. The main difference to the previous ones is that they make a triangle mesh of the input cloud prior to the segmentation. At the end they do a polygonalization using alpha shapes, resulting a clean smooth segmented reconstruction of the scene.

In [26], Koppula et al. proposed and evaluated a model and learning algorithm for scene understanding that exploits rich relational information derived from the point clouds captured by a RGB-D sensor for object labeling. Not only is able to detect coarse classes like walls or ground, but also individual objects such as a keyboard or a monitor. To achieve that they use a model similar to a Markov Random Field and Machine Learning algorithms (Fig. 1.8). The works were extended in [3]. Although the results are promising, their goals and methods are far from ours at the moment. In this direction we can find other approaches such as [45], [59], or [17].

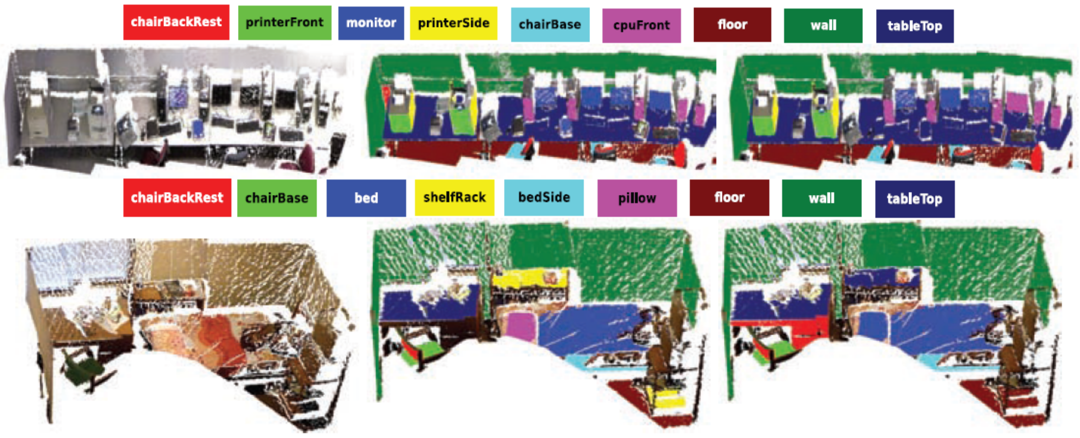


Figure 1.8: Semantic segmentation of the scene by [3]

Silberman et al. claimed that most existing work ignores physical interactions or is applied only to tidy rooms and hallways [52]. Their goals were to analyze typical messy indoor scenes and classify the segments among floor, walls, supporting surfaces and object regions, and to recover support relations among them.

Other special cameras such as omnidirectional cameras can be used to recover the spatial layout of a a scene from one single image, as López-Nicolás et al. show in [30].

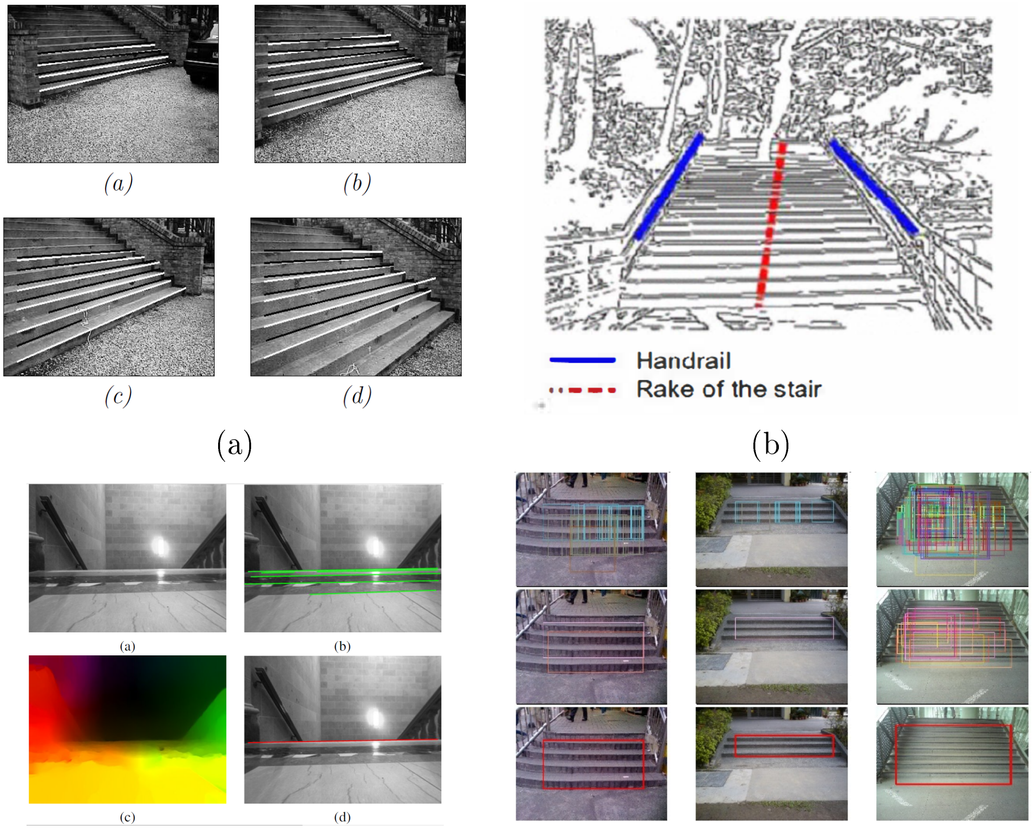
1.2.3 Stairs Detection

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. A brief revision of the existing algorithms is done in this section, classifying the works according to the sensor used.

1.2.3.1 Conventional Cameras

The work of Se and Brady [49] uses normal gray images to detect the presence and estimate orientation and slope of staircases in order to help partially sighted people by using traditional computer vision: Gabor filtering, vanishing point detection and homographies. All the results shown are from outdoor scenes, some of them with the camera quite far from the steps (Fig. 1.9a). However, all of them consist of ascending staircases and it seems that the algorithm might be unable to perform detection of single steps.

Other similar algorithms are [10], which focus in Unmanned Ground Vehicles stair climbing, and [19], which prioritizes the finding of the diagonal lines corresponding to the handrails to select stair candidates, which apparently works fine in the examples shown of outdoor ascending stairs (Fig. 1.9b), but seems unlikely to be exportable to other scenarios. In [20], the same authors proposed a different algorithm to detect stairways using Gabor filter, focusing in indoor environments.



(a) (b)

(c) (d)

Figure 1.9: Stair detection algorithms using conventional monocular cameras.

Hesch et al. [21] decided to focus on detecting descending staircases in both far and near approach, being the latter the most interesting. It uses line detection combined with optical flow computation in order to detect the edges of the first step (Fig. 1.9c). For autonomous tracked vehicles with small size and the camera situated close to the floor might be useful for detecting step-downs, but thinking about a wearable system for human navigation and the situations the sensor may have this approach falls short.

Conventional vision has been widely used together with Machine Learning algorithms. Andersen and Seibel [4] used a head mounted camera and Hough transform to detect lines and train a neural network in order to detect navigation hazards such as doors and staircases. Wang and Wang [58] used Real AdaBoost for training a cascaded classifier (Fig. 1.9d).

1.2.3.2 Stereo Vision

Stereo vision has also been used to perform this task. That is the case of [31], which combines the use of the geometry information provided by the stereo with the RGB data from the images captured. This combination makes the system less prone to error.

Also, the aforementioned [43] uses stereo vision in a more similar way as we do, using the information obtained to estimate normals and planes of the scene. However, they provide the basis for stair finding but not the recognition itself.

Gutmann et al. proposed a stair detection algorithm for humanoid robots in [18] where the stereo vision system segments the scene into planar surfaces. Experimental results are also shown in this work.

1.2.3.3 Laser Scanning

Most of the authors who use laser scanning for finding stairs focus on robot navigation, like [14],[2],[34], [5] and [40].

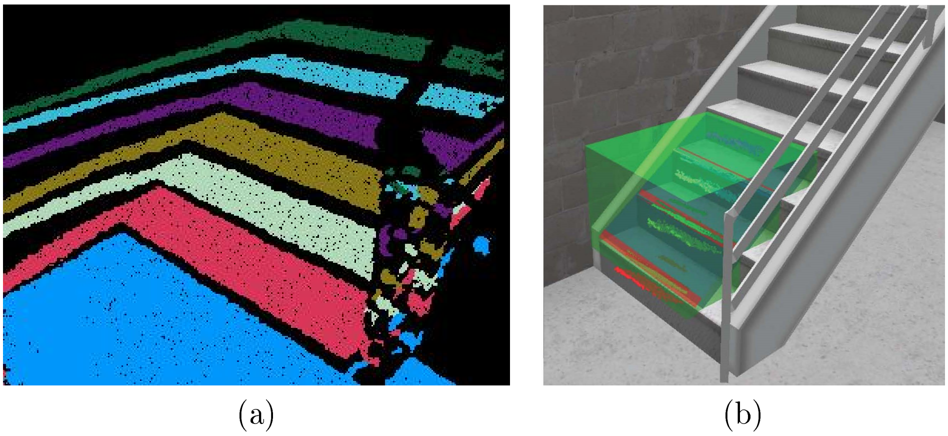
In [25], a small laser range sensor is used to develop a visually impaired assistant. It is attached to the chest of the subject. With this laser scanner, a segmentation of the scene is performed and the coordinates are classified into horizontal or vertical segments, after which the system judges whether there are steps or not. These ideas are also present in our proposal, but our output proves to be much more interesting as they only inform the user of the presence of stairs without giving extra data.

Oßwald et al. in [37] and [38] used the combination of a laser scan and conventional vision applied to humanoid robots stair climbing: the laser to detect the stairs and build a model of the staircase from a distance and the camera, pointing downwards, to detect the next step in front of the robot during stair-climbing. In 2012 they improved their proposal as shown in [39].

1.2.3.4 RGB-D

Some authors who choose RGB-D as main sensor use machine learning algorithms to perform staircase detection. In the case of [15], they use neural networks to detect the presence of obstacles and classify scenes captured by the depth camera among ascending staircase, descending staircase or none. Wang and Tian used a similar approach in [57] with some differences. At first they use Hough transform to detect parallel lines and find out if the images have a group of concurrent lines. After that they use the depth information to recognize among upstairs, downstairs and pedestrian crosswalks using support vector machine (SVM) classifiers. As we have already mentioned, most of RGB-D cameras have problems outdoors with strong sun light, so we do not think the recognition of pedestrian crosswalks is really helpful as it is impossible to assure if it is going to work.

Others preferred the usage of geometrical reasoning instead of machine learning to detect staircases with RGB-D. This is the approach we also consider to solve this problem, because we believe that the existing algorithms using this technology are incomplete and can be improved.


(a)

(b)

Figure 1.10: (a) Stair detection image from [53], where the points correspondent to each step extend beyond the actual step. (b) Model of the staircase from [12].

The algorithm of Tang et al. [53] has some important flaws. First of all, their identification of ground plane starts with the estimation of the height and orientation of the sensor from the ground plane given by an accelerometer, whereas our algorithm does not need that extra information to be able to find the ground. Their iteration to find steps one by one by looking for planes at certain heights is prone to fail as it does not take into account any other shape constraint except having sufficient points. This may cause false positives and that the inliers of the planes correspondent to the steps include more points than the ones the step actually has (Fig. 1.10a). It can be solved using a region growing algorithm instead of just RANSAC for planes. They considered there is a staircase if there are three or more steps, ignoring the possibility of less than three steps, which is also quite common in doorways or other special constructions. The information they provide is also very basic: detects the presence of the staircase, if it ascends or descends and the estimated number of steps. Using point clouds in Euclidean distances allows to measure sizes and distances in a more straightforward manner than using inverse depth measurements in range images as they do. They provided a dataset of some staircases both ascending and descending and other common indoor scenes to detect false positives. We have made a comparison with their results and ours that will be shown later in this work.

Another similar approach is shown in [56]. Their alignment of the reference system of point cloud captured by the sensor to a user reference system is also done in our work. They use that projection to find the floor as the biggest plane parallel to the floor plane, which might not be true if there are some other dominant planes such as big table. In our case, the ground is automatically found at first and it serves as hint to make the projection, so the floor is never missed. While we look for planes just one time and then classify among ground, walls, obstacles, steps, etc., they use a RANSAC algorithm every time they want to find any specific plane, but they might not even be on the scene, being highly time-consuming. They also used the Tang et al. database improving their results, and we compare our performance with them as well.

The two works commented above commit the same mistake of focusing on the detection and ignoring the modelling. With the modelling the actual measurements of the steps can be obtained, information which can be used to verify the detection, to give indications to the user or to analyze the traversability of the staircase.

On the other hand, Delmerico et al. proposed an ascending stairway modelling that introduces some interesting ideas [12]. Their goal is to localize and model stairways to check for traversability and enable autonomous multi-floor exploration. They want to perform this task while the robot is also performing simultaneous localization and mapping of the scene, to include the model of the stairway in the map as well. The model of the staircase is an inclined plane inside a bounding box containing the stairway, and the measurements of the steps and the whole staircase (Fig. 1.10b). 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. 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.

1.3 Objectives

Now that we have explained the motivation that lead us to develop an assistive system for the visually impaired and after performing an extensive review of the related works, it is time to set our main objectives for the current work:

- Choose the best location of the RGB-D sensor given the specification that it has to be wearable.

- Search for the most appropriate libraries and programming languages to perform this task.
- Learn to preprocess the data captured by the sensor in the most efficient way.
- Perform a segmentation of the three-dimensional scene and classify the segments according to their orientation with respect to the user.
- Trace the obstacle-free region of the floor in the image.
- Develop an algorithm that can detect the presence of a staircase, organizing the planes belonging to the steps in a hierarchical way.
- Once we have have found the steps which form a staircase, retrieve and trace the model of the stair.
- Perform experiments in different environments taking into account the time necessary to make all the operations, the quality of the models obtained and if it is possible compare the results with other works.

1.4 Structure

This work is organized as follows:

1. In Chapter 1 we explain the motivation behind this work, the current state of the art and the objectives we set, as well as a brief comment about the structure of this text.
2. In Chapter 2 the first considerations of what we have built here are described, paying special attention to the location of the sensor, the initial preprocessing of the data and the detection of the floor, which is the starting point of the algorithm.
3. In Chapter 3 we show how we perform the segmentation and classification of the elements of the scene, describing each stage and showing examples.
4. In Chapter 4 we explain the stair detection and modelling module, which is the most novel part of this work.
5. In Chapter 5 some results are shown, including captures, comparisons with other works, and quality of the model and computing time analysis.
6. In Chapter 6 we express our final thoughts about this work and some possible lines of work for the future.

1.5 Publication

As a result of this work there has been a research paper sent to the *Second Workshop on Assistive Computer Vision and Robotics (ACVR)* of the *European*

Conference on Computer Vision (ECCV), which has been accepted for oral presentation [41]. The workshop takes place the 12th of September of 2014 in Zurich, Switzerland.

[41] Pérez-Yus, A., López-Nicolás, G., Guerrero, J.J.: Detection and Modelling of Staircases Using a Wearable Depth Sensor. In: Second ECCV Workshop on Assistive Computer Vision and Robotics (2014).

Chapter 2

RGB-D Setup and Data Preprocessing

The different parts of the algorithm and the theory behind them will be explained in this section. The whole process is thought to be iterative, capturing a video sequence in real time and making the computations on the go, but it has been also implemented a single point cloud mode for the experiments.

In Section 2.1 a scheme of how the camera is intended to be worn and the capture of the data is shown. If the floor is significantly present in the cloud, it is automatically found and the 3D coordinates of the point cloud are projected to a user-centered coordinate system as explained in Section 2.2.

2.1 Setup Configuration and Data Acquisition

There are different options to locate the camera in a wearable navigation system [33]. 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 safe 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 safe 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 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. 2.1.

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

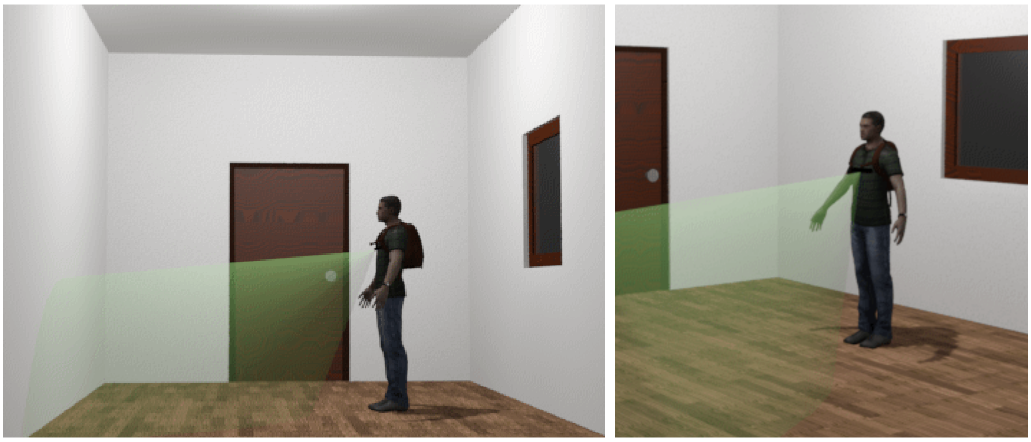


Figure 2.1: 3D rendering which shows how the camera is intended to be worn

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 (640 × 480 points) and thus the first operation will be filtering. First we remove the points with no 3D information from the cloud, and then we pass a 3D voxel grid filter. This is a common algorithm widely used for downsampling point clouds, which also helps removing noise and smoothing the surfaces.

The voxel grid filter consists on a 3D division of the space in a grid of 3D boxes (voxels) inside of which there is only one point (the centroid) instead of the initial set of points contained. The sizes of the edges of the voxels are determined by balancing time consumption and accuracy of the scene. In our case, four centimeters represents the best commitment. This filtered cloud will be used in the remaining algorithm. Fig. 2.2 shows an example of this filtering.

2.2 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. 2.3a.

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 (RANdom SAmple Consensus, [16]) 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. Approximately the floor normal should match (with certain threshold) the $y - axis$ by rotating

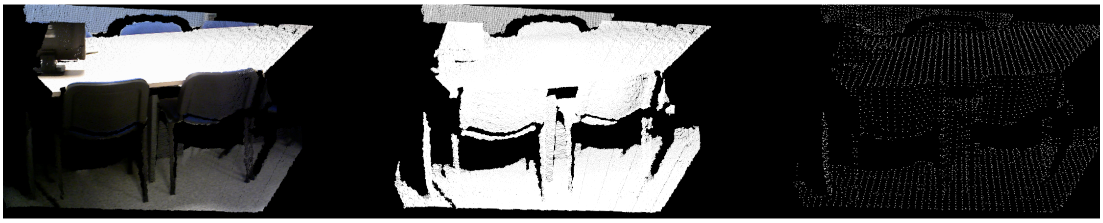
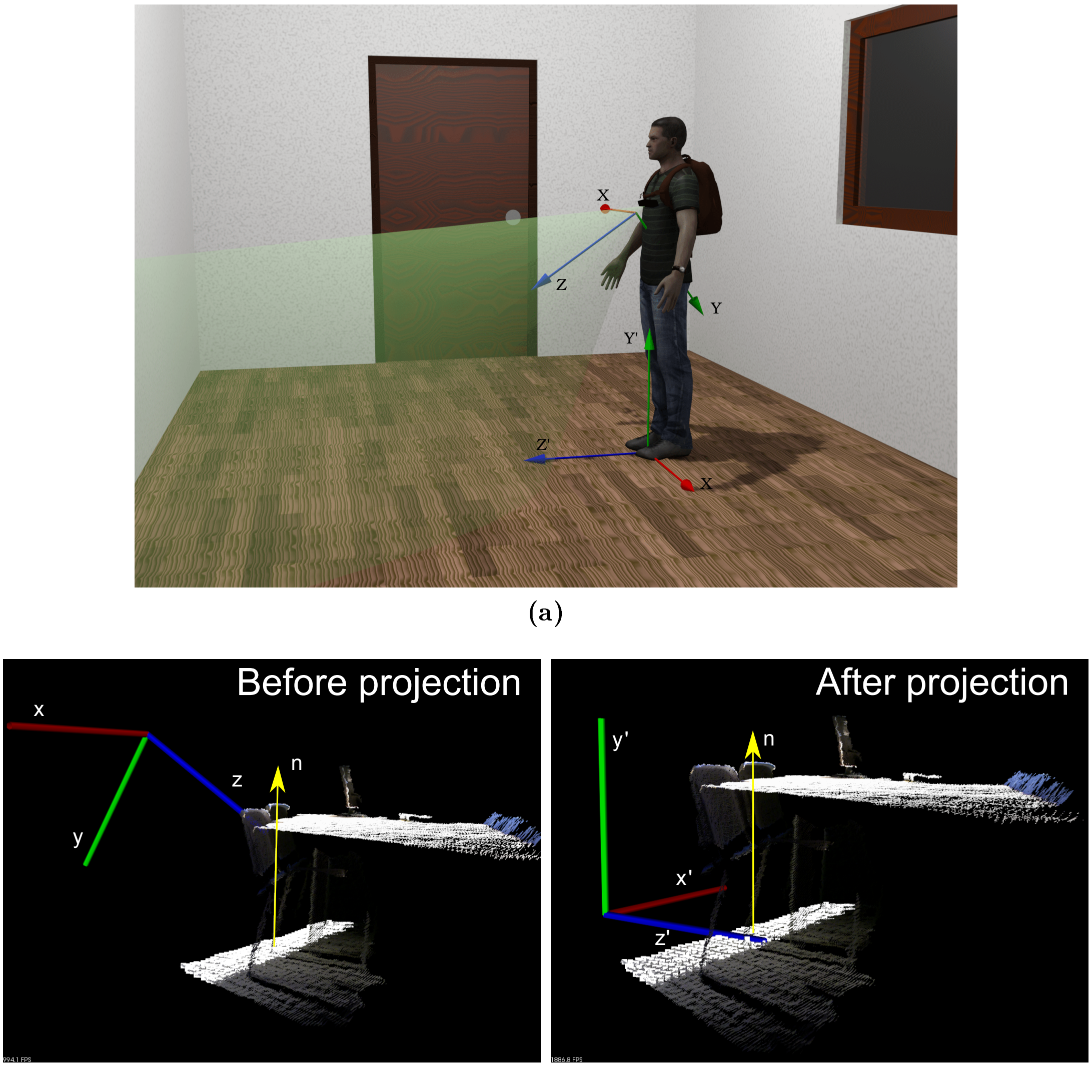


Figure 2.2: Example of colored point cloud captured by the RGB-D sensor (left) and comparison between the set of points before (middle) and after (right) the Voxel Grid with $4cm$ of edge.



(a)

(b)

Figure 2.3: (a) Wearable camera location and axis position before (XYZ) and after ($X'Y'Z'$) the projection to the ground in a 3D render. (b) Projection of the 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.

on $x-axis$ an amount of 135°. The distance of the plane to the camera depends on the height of the user, ranging from 1.1 to 1.4 meters.

With the addition of these constraints, most planes should be discarded, although sometimes the floor can still be mistaken by, for example, a step. As the camera is tilted downwards and the floor should appear close to the subject, if we want to make sure a pass-through filter on z-direction can be added, keeping only the points closest than a value of $1 \sim 2$m and iteratively increase this value until a reasonable amount of points belonging to a good floor candidate are found. In Fig. 2.3b these points are the ones of the floor coloured in white.

If the floor is not found with the first cloud, a new one will be captured and the process will be repeated until a ground plane is found. Otherwise there is no starting point for the incoming parts of the algorithm. As the camera is pointing downwards and the user is supposed to be standing on the floor, it will not last long in any case. The flowchart on Fig 2.4 summarizes the whole process.

Once a set of points belonging to a good floor candidate plane is 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 $\vec{n}=(A,B,C)$ 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 $\vec{n}$ parallel to $\vec{j}=(0,1,0)$ as follows:

$$R_zR_x\vec{n}^T = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{2.1}$$

The entire transformation matrix is then:

$$T = \begin{pmatrix} \cos\gamma & -\sin\gamma\cos\alpha & \sin\gamma\sin\alpha & 0 \\ \sin\gamma & \cos\gamma\cos\alpha & -\cos\gamma\sin\alpha & -D \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

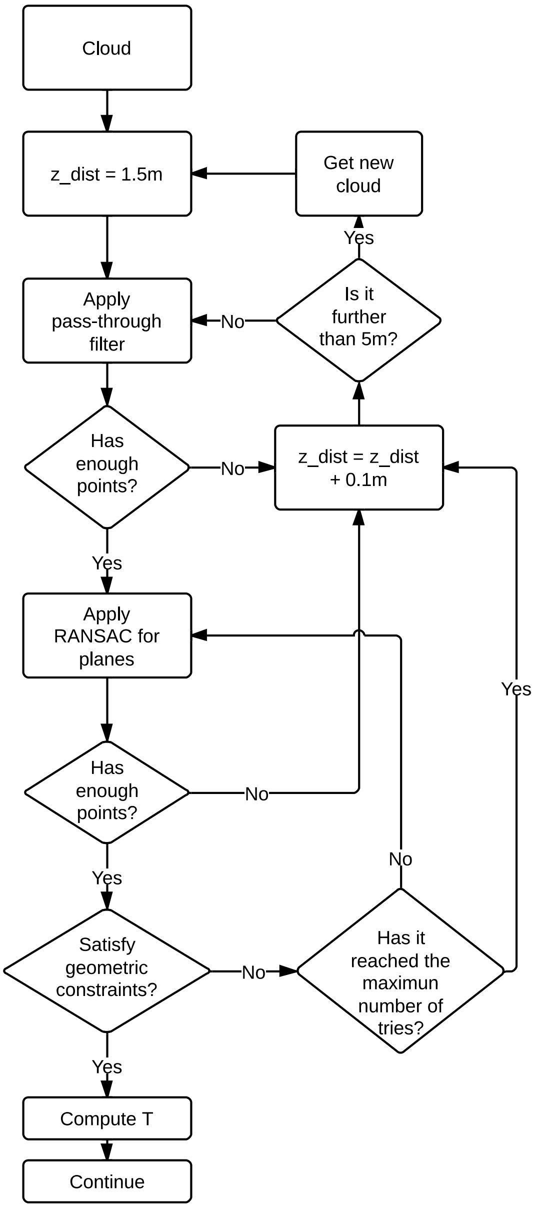


Figure 2.4: Flowchart of the ground detection and projection algorithm

Chapter 3

Segmentation and Classification of the Scene

In order to perform any relatively complex task it is necessary to recognize the features of your surroundings. Before the recognition, a partition of the environment in different segments must be performed, and that is called segmentation. Segmentation has been an essential issue in robot and human navigation through the years. 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 [24].

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 [53]. The remaining points are later merged into existing planes or associated to different clusters of points. In particular, the segmentation module is divided in the following stages:

1. Normal estimation.
2. Region-growing.
3. Planar test.
4. Plane extension.
5. Euclidean cluster extraction.
6. Classification

3.1 Normal Estimation

There are many ways to perform the normal estimation problem. Some of them provide fast results by taking advantage of the organization of the clouds (that is, the cloud has width and height and the position of the points in the cloud corresponds to the order in the image). After the initial filtering, our cloud appears unorganized as we got rid of a large amount of points. Another way of solving this issue consists in obtaining the surface of the point cloud dataset by using surface meshing techniques, after which the normal estimation is immediate. This way adds extra computational load to the algorithm and we try to avoid it.

The solution for estimating the surface normal chosen is based on the analysis of the eigenvectors and eigenvalues (or PCA - Principal Component Analysis) of a covariance matrix created from the nearest neighbors of the query point. More specifically, for each point $\boldsymbol{p}_i$, we assemble the covariance matrix $\mathcal{C}$ as follows:

$$\mathcal{C} = \frac{1}{k}\sum_{i=1}^{k} \cdot(\boldsymbol{p}_i - \overline{\boldsymbol{p}}) \cdot (\boldsymbol{p}_i - \overline{\boldsymbol{p}})^T,\ \mathcal{C} \cdot \vec{\mathsf{v}_j} = \lambda_j \cdot \vec{\mathsf{v}_j},\ j \in \{0, 1, 2\} \tag{3.1}$$

Where k is the number of point neighbours considered in the neighbourhood of $\boldsymbol{p}_i$, $\overline{\boldsymbol{p}}$ represents the 3D centroid of the nearest neighbors, λ_j is the j-th eigenvalue of the covariance matrix, and $\vec{\mathsf{v}_j}$ the j-th eigenvector. The third component obtained from the analysis corresponds to the normal direction. It is necessary to make sure that the normal direction is flipped towards the viewpoint (Fig. 3.1). In this process the curvature of the surfaces is also computed. The output surface curvature is estimated as a relationship between the eigenvalues of the covariance matrix (as presented above), as:

$$\sigma = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{3.2}$$

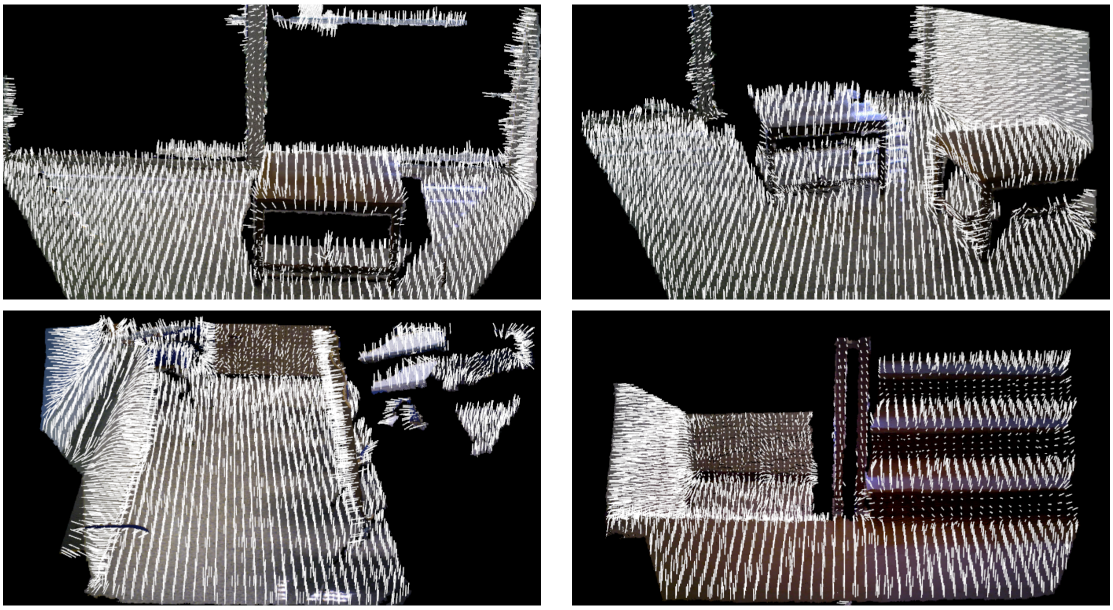


Figure 3.1: Normal estimation. The normals are drawn in the colored point cloud as white vectors pointing towards the viewpoint.

3.2 Region-growing

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 that can be found are smaller than a certain threshold.

The procedure can be seen in more detail in Algorithm 1. The inputs of the algorithm are the point cloud = $\{P\}$ with the corresponding normals = $\{N\}$ and curvatures = $\{c\}$ as computed in Section 3.1, a Neighbour finding function $\Omega(.)$ for which we use a Kd-Tree partition of the cloud, a curvature threshold $c_{th} = 0.5$, an angle threshold $\theta_{th} = 4°$ and a minimum region size of 25 points. In Fig. 3.2 there is an example of the output of the algorithm.

Algorithm 1 Region-growing algorithm

```
Region list $R \leftarrow \emptyset$
Available points list $\{A\} \leftarrow \{1, ..., |P|\}$
while $\{A\}$ is not empty do
  Current region $\{R_c\} \leftarrow \emptyset$
  Current seeds $\{S_c\} \leftarrow \emptyset$
  Point with minimum curvature in $\{A\} \rightarrow P_{min}$
  $\{S_c\} \leftarrow \{S_c\} \cup P_{min}$
  $\{R_c\} \leftarrow \{R_c\} \cup P_{min}$
  $\{A\} \leftarrow \{A\} \setminus P_{min}$
  for $i = 0$ to size ( $\{S_c\}$ ) do
    Find nearest neighbours of current seed point $\{B_c\} \leftarrow \Omega(S_c\{i\})$
    for $j = 0$ to size ($\{B_c\}$) do
      Current neighbour $pointP_j \leftarrow B_c\{j\}$
      if $\{A\}$ contains $P_j$ and $cos^{-1}(|(N\{S_c\{i\}\}, N\{S_c\{j\}\})|) < \theta_{th}$ then
        $\{R_c\} \leftarrow \{R_c\} \cup P_j$
        $\{A\} \leftarrow \{A\} \setminus P_j$
        if $c\{P_j\} < c_{th}$ then
          $\{S_c\} \leftarrow \{S_c\} \cup P_j$
        end if
      end if
    end for
  end for
  Add current region to global segment list $\{R\} \leftarrow \{R\} \cup \{R_c\}$
end while
Return $\{R\}$
```

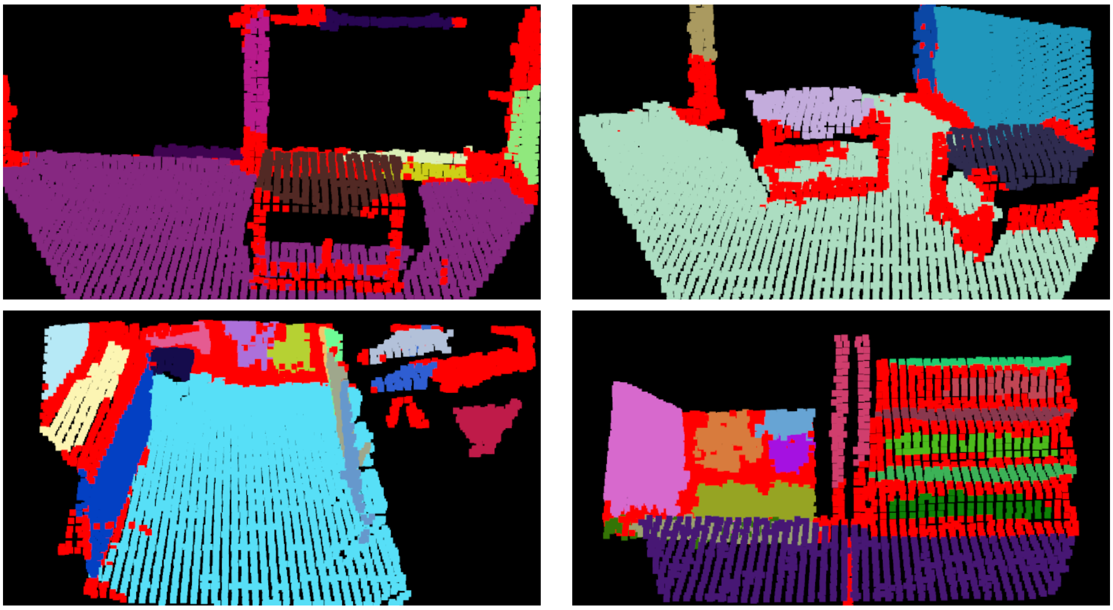


Figure 3.2: Region-growing. The points in red are points with non-assigned region

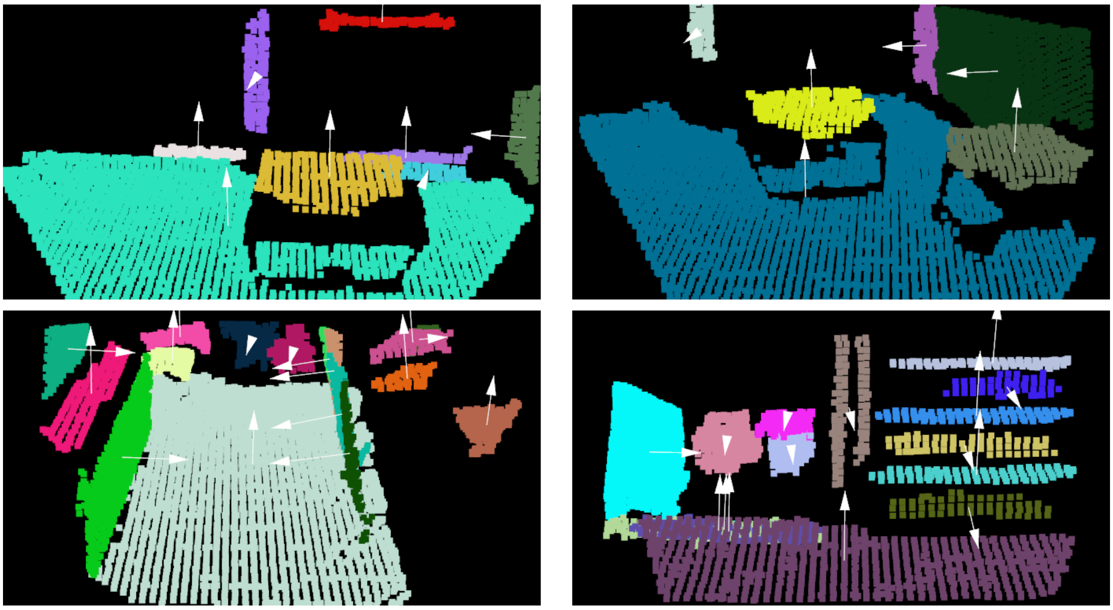


Figure 3.3: Planar test. The regions whose majority of points lies on a plane are considered planes, whose normal is drawn as a white vector.

3.3 Planar Test

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 (Fig. 3.3).

3.4 Plane Extension

In this stage we look for improving the results of the region-growing stage by extending the planar regions to fit in a shape closer to reality. This also allows us to have more control of the process as we divide the problem in two stages.

The remaining points not belonging to any region yet 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 placed near the region (Fig. 3.4). The first two computations can be made using simple 3D geometry operations, and for the later we use a neighbour search from the Kd-tree partition. For example, if the remaining point considered has 2 neighbours of a certain regions in a radius of 15cm, it is considered close to a region.

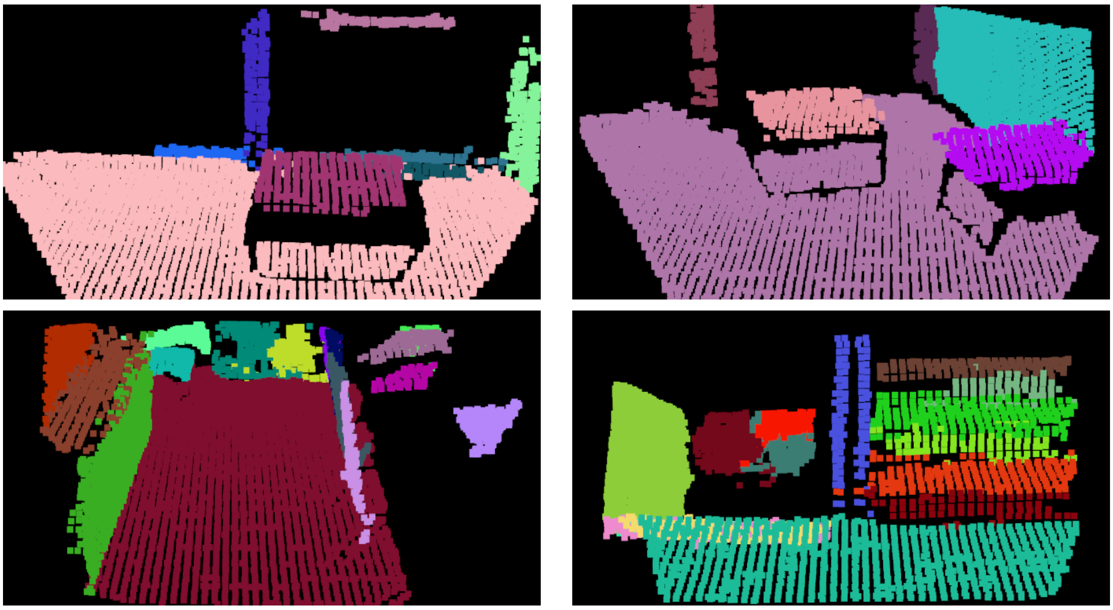


Figure 3.4: Plane extension. The points not belonging to any region become part of the planes found if they satisfy geometric restrictions.

3.5 Euclidean Cluster Extraction

The points still not belonging to any region go through a cluster extraction algorithm which establishes connections and forms separate entities. The algorithm behind this problem is shown in Algorithm 2. The clusters are considered obstacles at first, although the planar testing algorithm can be used again with this clusters as sometimes they are planes (Fig. 3.5).

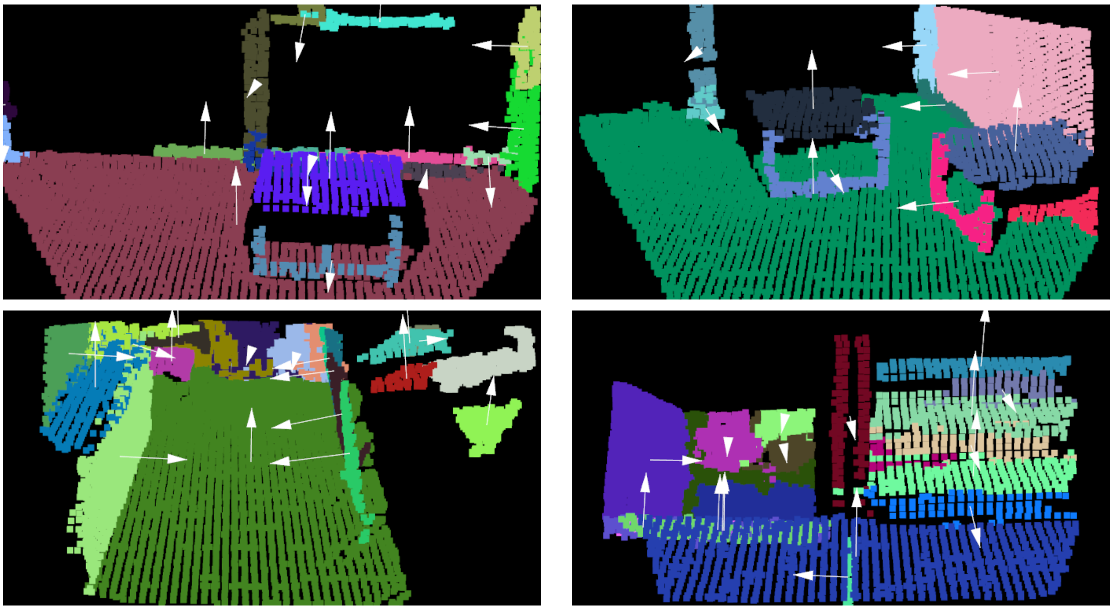


Figure 3.5: Euclidean cluster extraction. Points still not belonging to any region are grouped forming clusters.

Algorithm 2 Euclidean Cluster Extraction

```
Create a Kd-tree representation for the input point cloud dataset $P$;
Set up an empty list of clusters $\{C\} \leftarrow \emptyset$, and a queue of the points that need to be checked $\{Q\}$;
for $i = 0$ to size $(P)$ do
    $\{Q\} \leftarrow \{Q\} \cup \boldsymbol{p}_i$
    for $k = 0$ to size $\{Q\}$ do
        search for the set $P_k^i$ of point neighbors of $\boldsymbol{p}_i$ in a sphere with radius $r < d_{th}$;
        for every neighbor $\boldsymbol{p}_i^k \in P_i^k$ do
            if $\boldsymbol{p}_i^k$ has not been processed yet then
                $\{Q\} \leftarrow \{Q\} \cup \boldsymbol{p}_i^k$
            end if
        end for
    end for
    $\{C\} \leftarrow \{C\} \cup \{Q\}$
    $\{Q\} \leftarrow \emptyset$
end for
```

3.6 Plane Classification

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. 3.6).

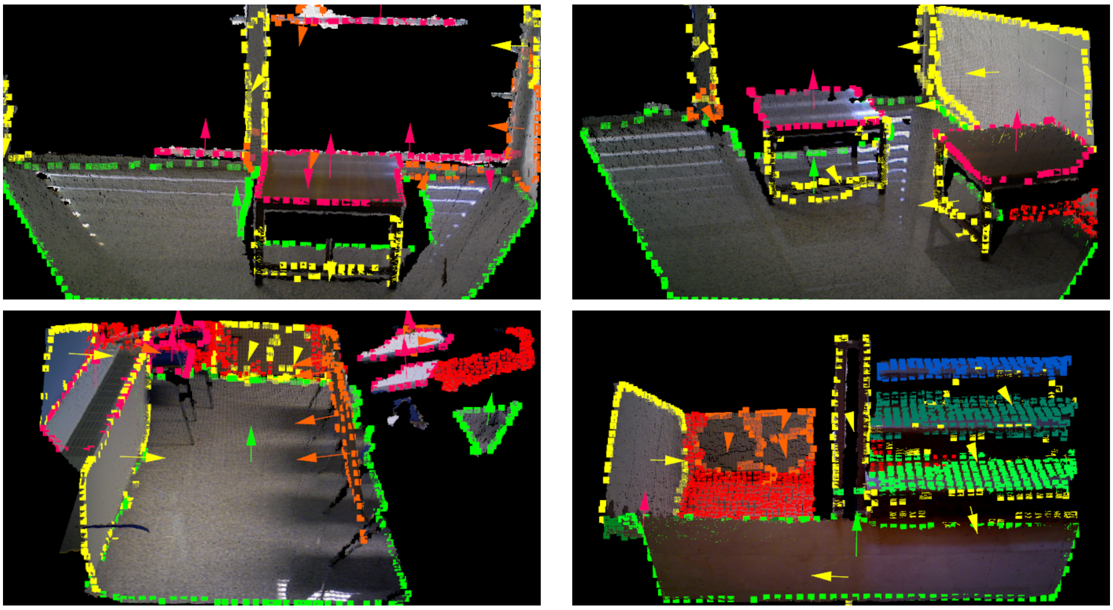


Figure 3.6: Classification. The planes are classified according to their orientation. For each plane, the points of their contours and the normal are displayed in certain color: green for the floor, pink for horizontal planes, yellow for walls, orange for non-vertical and non-horizontal planes. Also, in red the non-planar clusters and in a color gradient from green to blue the steps in the last image.

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 [35] (see Fig. 3.7). 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 very 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. 3.8). 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.

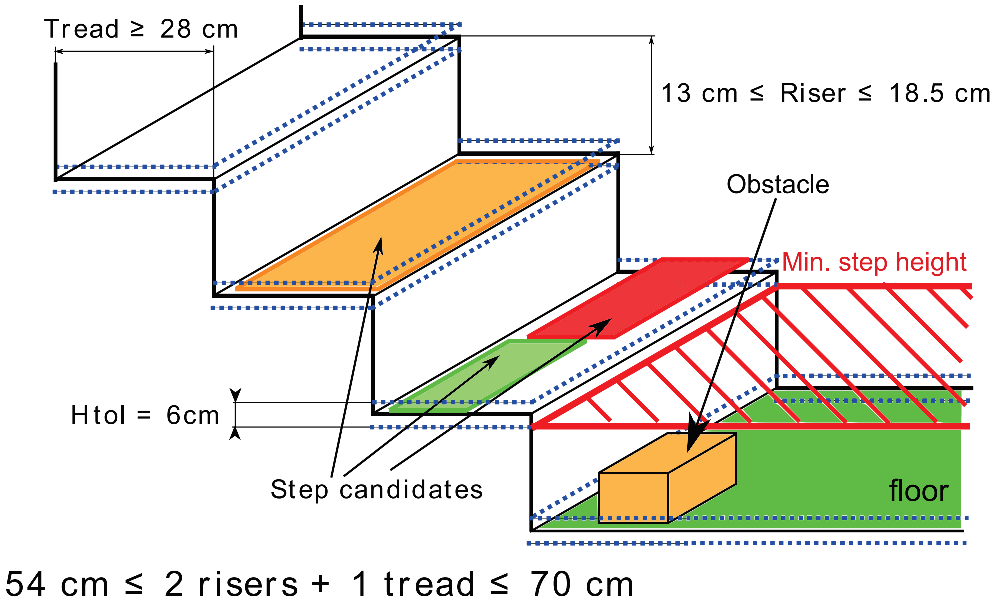


Figure 3.7: Representation of the measurements of a step according to the Technical Edification Code from Spain.

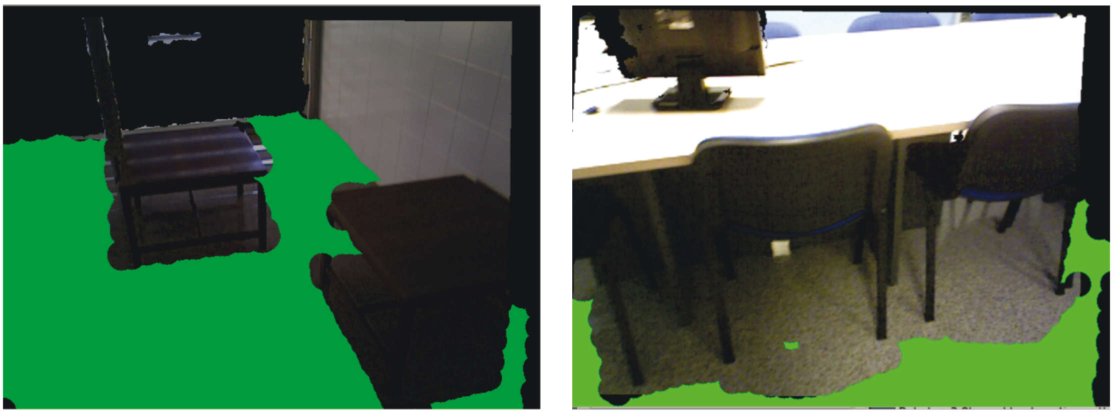


Figure 3.8: In green, the portion of the ground which can be walked over as it has no obstacles above.

Chapter 4

Stair Detection and Modelling Algorithm

In Chapter 3 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.

4.1 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 (Fig. 4.1). 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 between the valid range

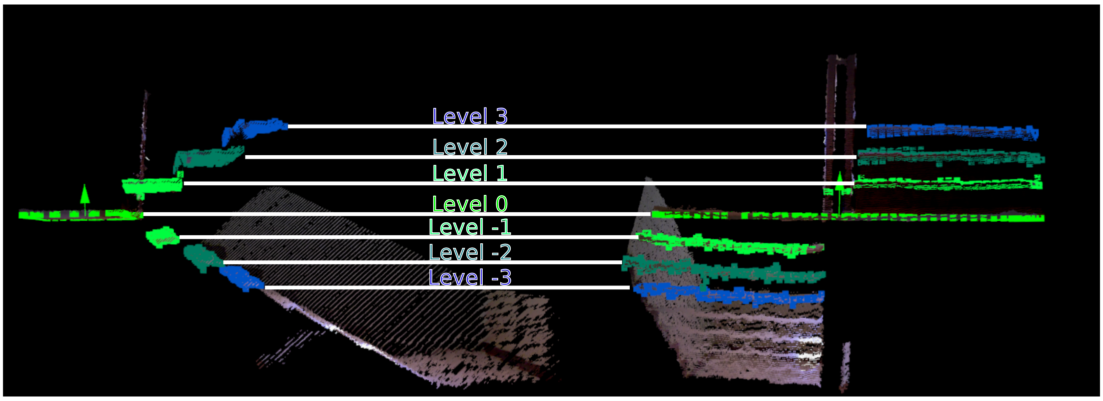
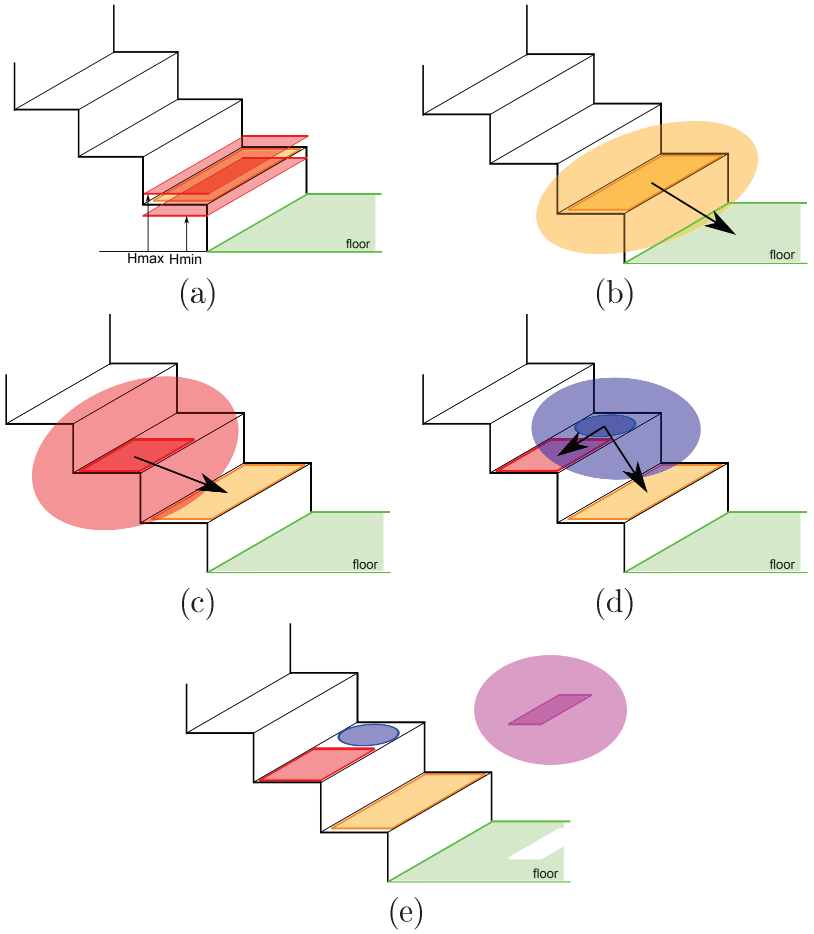


Figure 4.1: Example of one detected ascending stair and a detected descending stair from two perspectives.



(a) (b)

(c) (d)

(e)

Figure 4.2: Explicative sketches for the stair detection algorithm. (a) First step must be in the valid range of heights. (b) First step must be connected to the floor if it is in sight. (c) and (d) The connectivity to previous levels is checked doing a neighbour search. (e) If the candidate is not connected to previous level, it is not part of the staircase.

of heights (between $H_{min} - H_{tol}/2 = 10$ and $H_{max} + H_{tol}/2 = 21.5cm$) to the ground constitute *first step candidates* (Fig. 4.2a). 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 (Fig. 4.2b). 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.

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 a new (Fig. 4.2c) or to the current level (Fig. 4.2d). If they have

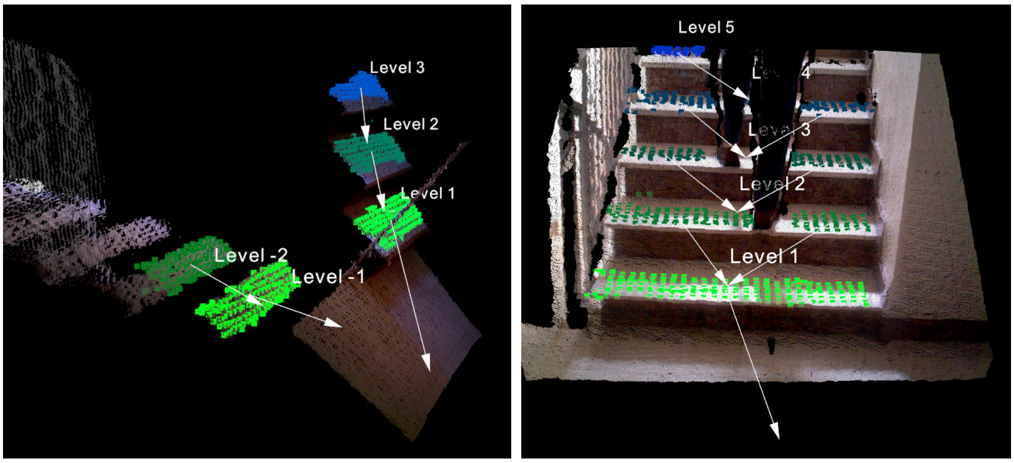


Figure 4.3: Connectivity between step candidates to previous levels: Ascending and descending staircases (left), more than one region per level (right).

no connection to previous levels (e.g. a horizontal plane correspondent to a table) they are classified as obstacles (Fig. 4.2e). As a result, a set of connected regions corresponding to different levels is obtained (Fig. 4.3). When all the candidates have been checked, if the number of levels is greater than one, the algorithm starts the modelling of the staircase.

A special case occurs when there is only one 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. For example, we know the measurements of the steps according to the regulations. If the candidate does not satisfy the conditions because is too small it is an obstacle, if it is too big it is floor at another level. In Fig. 4.4 we show an example where the floor at another level is detected.

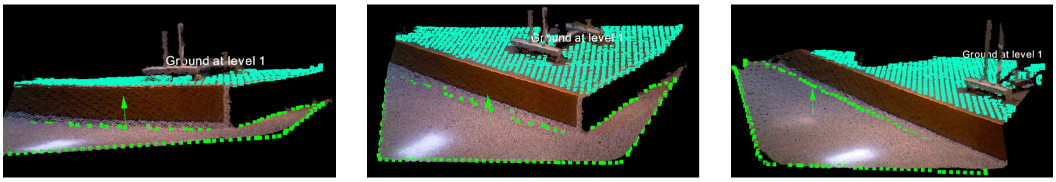


Figure 4.4: Three examples of floor at another level that has been found.

4.2 Stair Modelling

The shape of the staircases can have some differences regarding the presence of absence of the riser or the inclination it might have. We are going to consider a unified model for all the possible cases, consisting on a set of consecutive parallelograms one of each corresponds to a single step (Fig. 4.5). The geometric parameters we are going to retrieve from the point cloud are:

- Width, length, height of every parallelogram.
- Number of steps.
- Orientation of the staircase (main axis).

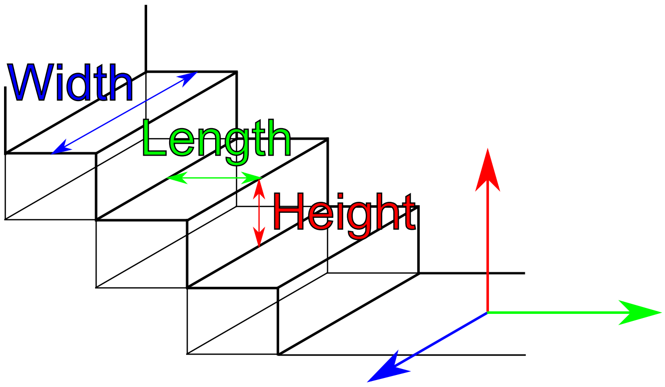


Figure 4.5: Width, length, height and directional axis of the model of the staircase we propose.

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. 4.6).

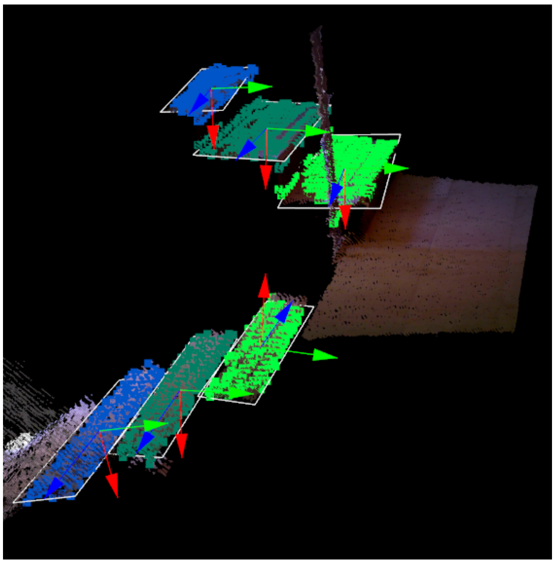


Figure 4.6: Principal components for each step coloured in order (blue-green-red) and bounding rectangle in white.

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 Σ, 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 $\Phi = [\phi_1, \phi_2, \phi_3]$ 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:

$$P_\phi = (P_{\mathbf{x}} - \mu_{\mathbf{x}}) \cdot \Phi \tag{4.1}$$

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 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. 4.7). 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.

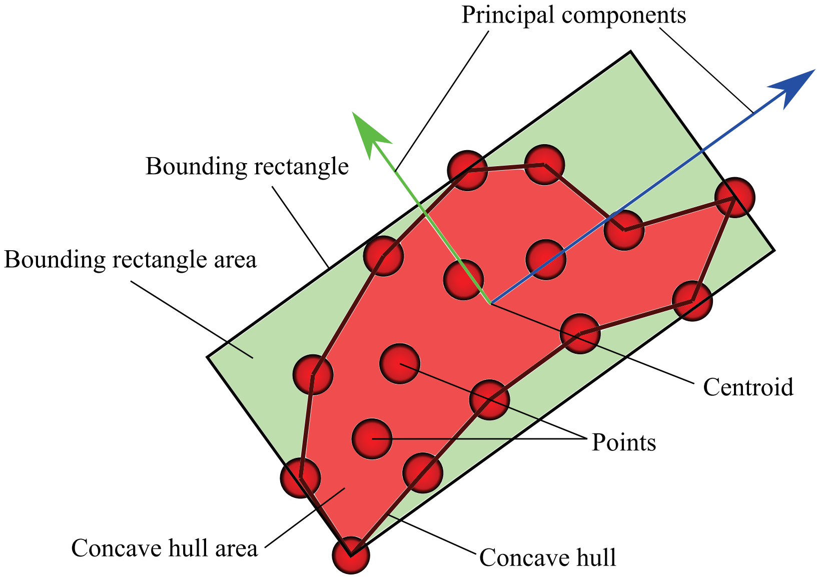


Figure 4.7: Illustrative sketch of the different components.

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:

- Forcing the third principal component to be parallel to the vertical axis, because usually the measurements are noisy enough to produce badly oriented axis.
- Rotating the selected axis as initial guess until the sum of areas of the bounding rectangles of all the steps at the same time is minimized (Fig. 4.8). With this operation you make sure that the axis of the model fits the points of the stair in the best possible way.

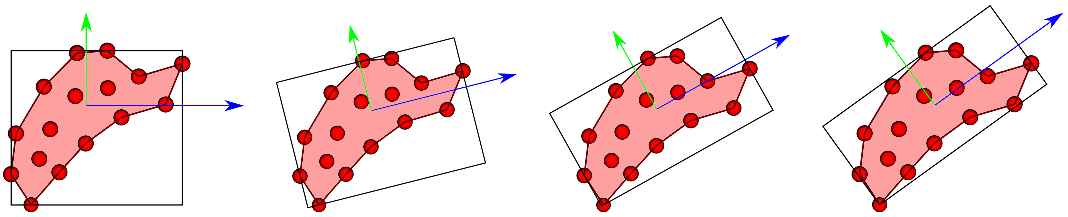


Figure 4.8: Example of the rotation of the selected axis to minimize the area of the bounding box with the points of one single step. It needs to be done with all the steps at the same time in order to obtain the final direction that fits best the staircase.

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 parameters are the following:

- The **width** is the width of the best step.
- The **height** is the average vertical distance between consecutive steps. The centroid of the set of points of each step can be considered.
- The **length** the average 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 (Fig. 4.9) or leaves a gap in descending staircases due to self occlusions (Fig. 4.10). It causes that the length you see in ascending staircases is more than what we need for our model, whereas in descending staircases you miss some valid portion of the step.

Once we have all the parameters, we can use them to validate the staircase detection or discard it. In case of positive results we can trace the model and even extend the information to non-detected steps if the number of steps is known (Fig. 4.11).

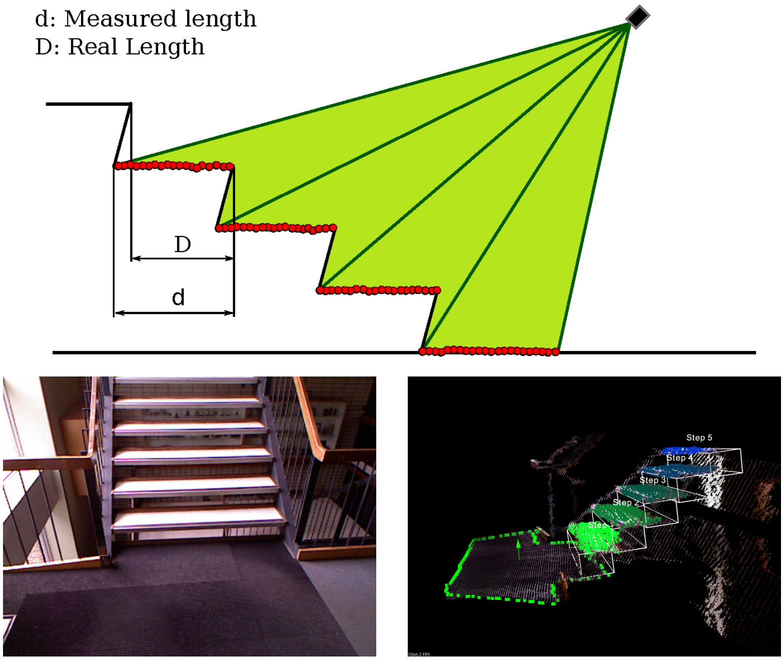


Figure 4.9: From the point of view of the camera, in some cases the measured length is too large to build the proposed model properly.

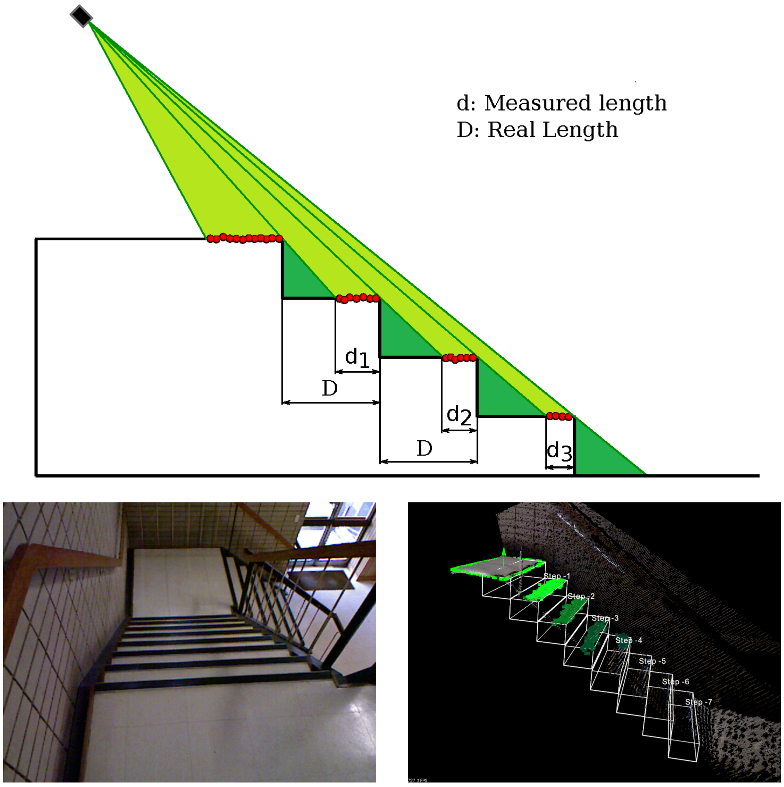


Figure 4.10: The measured length of the steps it is just a fraction of the real length we are going to use in our model.

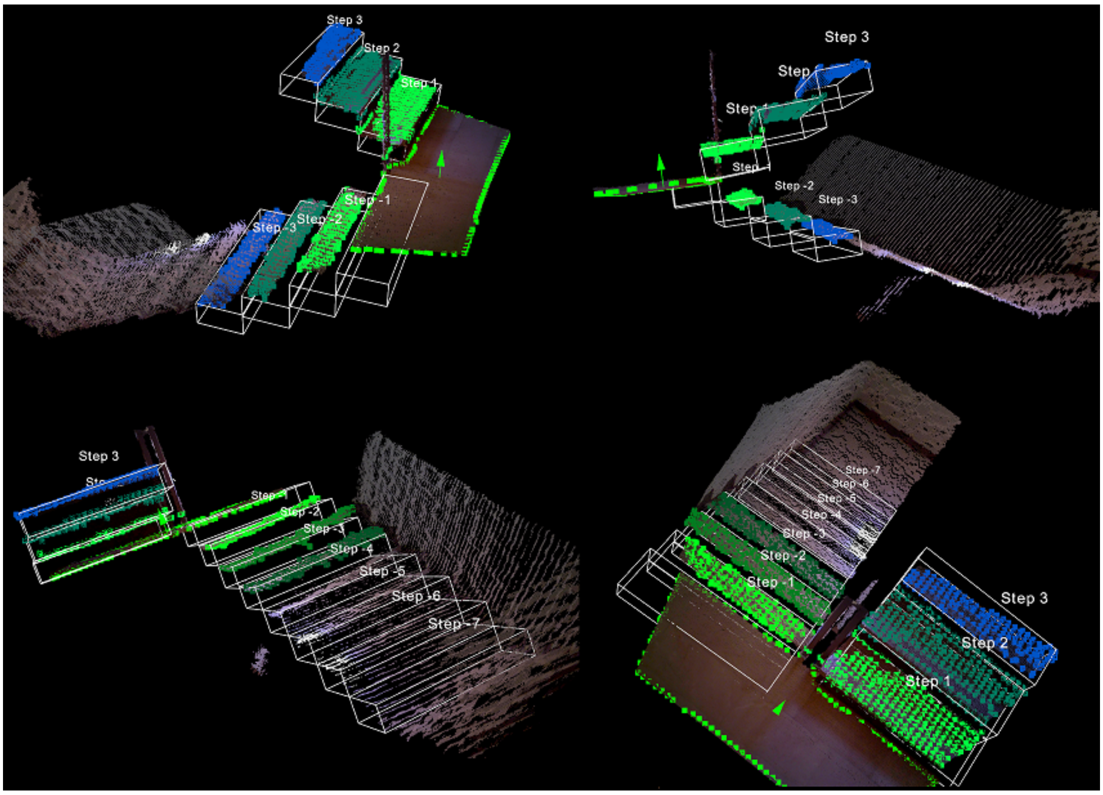


Figure 4.11: Estimated model of the staircase. Top images only draws the parallelograms corresponding to the steps found, whereas at the bottom all the steps are displayed.

Chapter 5

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 [53] 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 information about the calibration parameters of the camera. The results of the test with this dataset were successful even in total darkness (Fig. 5.1). We tested for false positives and false negatives using this dataset and compared our results with the ones from [53] and [56] (Fig. 5.2). We achieve better results with the 0% of false negatives as in [56] 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. 5.3). 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.

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

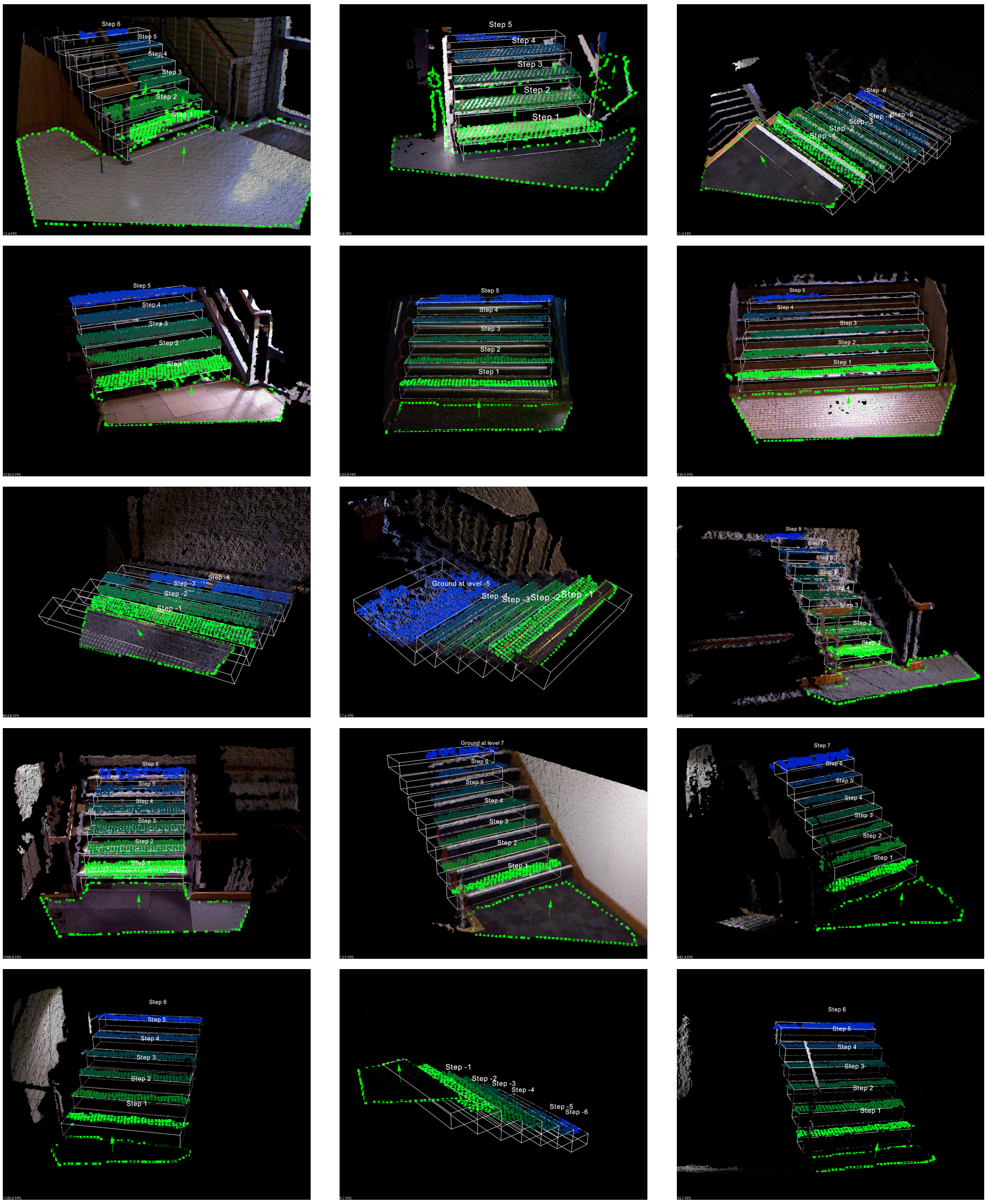


Figure 5.1: Some examples of results obtained with the dataset. The last four are from captures made in darkness.

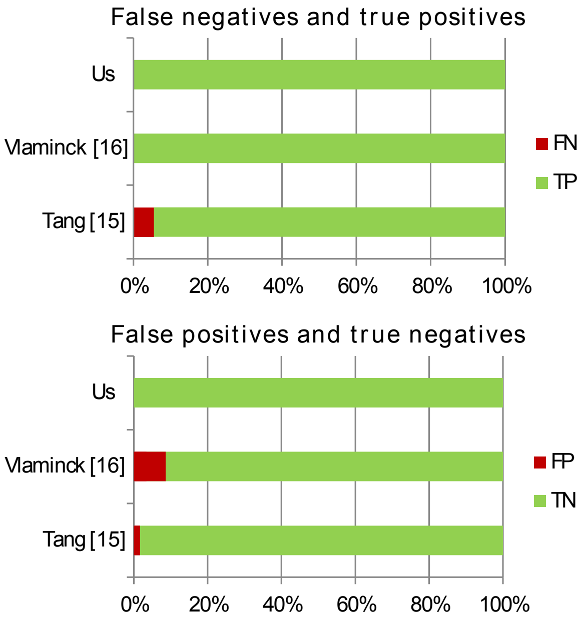


Figure 5.2: Comparison of false negatives and false positives between our work and the presented by [53, 56]

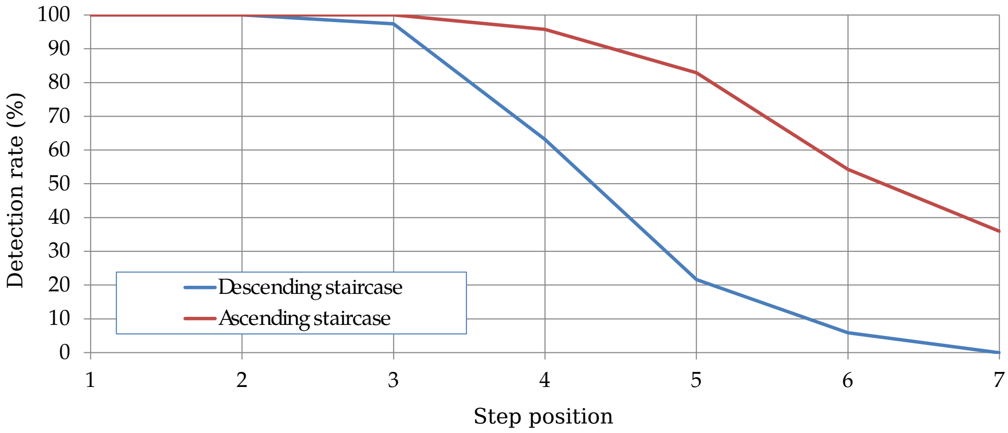


Figure 5.3: Step detection rate with the step position in the staircase.

Table 5.1: 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

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 5.1. 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 descending perspectives, from different viewing angles, the results were compared to the real measurements, as shown in the Table 5.2. 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. 5.4.

Table 5.2: 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

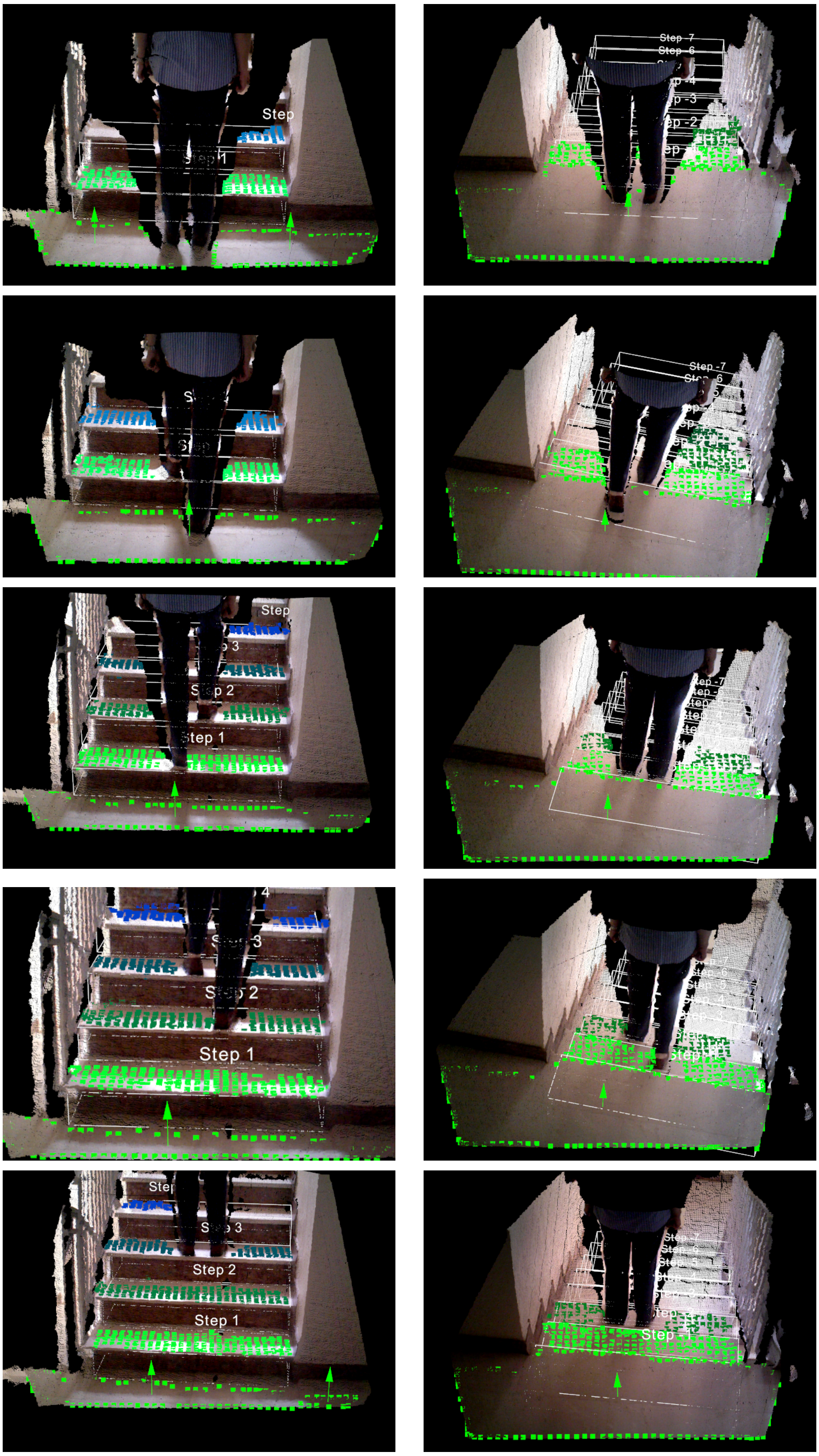


Figure 5.4: Example of a person partially blocking the view of the staircase during ascent or descent.

Chapter 6

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 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.

Bibliography

[1] Aladren, A., Lopez-Nicolas, G., Puig, L., Guerrero, J.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] Anand, A., Koppula, H.S., Joachims, T., Saxena, A.: Contextually guided semantic labeling and search for three-dimensional point clouds. The International Journal of Robotics Research 32(1), 19–34 (2013)

[4] Andersen, J., Seibel, E.: Real-time hazard detection via machine vision for wearable low vision aids. In: Wearable Computers, 2001. Proceedings. Fifth International Symposium on. pp. 182–183. IEEE (2001)

[5] Bansal, M., Matei, B., Southall, B., Eledath, J., Sawhney, H.: A LIDAR streaming architecture for mobile robotics with application to 3D structure characterization. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1803–1810 (2011)

[6] Bernabei, D., Ganovelli, F., Benedetto, M., Dellepiane, M., Scopigno, R.: A low-cost time-critical obstacle avoidance system for the visually impaired. In: International Conference on Indoor Positioning and Indoor Navigation (2011)

[7] Borenstein, J., Ulrich, I.: The guidecane-a computerized travel aid for the active guidance of blind pedestrians. In: IEEE International Conference on Robotics and Automation. vol. 2, pp. 1283–1288 (1997)

[8] Bostelman, R., Russo, P., Albus, J., Hong, T., Madhavan, R.: Applications of a 3D range camera towards healthcare mobility aids. In: IEEE International Conference on Networking, Sensing and Control. ICNSC. pp. 416–421 (2006)

[9] Capi, G., Toda, H.: A new robotic system to assist visually impaired people. In: RO-MAN. pp. 259–263. IEEE (2011)

[10] 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)

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

[12] 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)

[13] Dube, D., Zell, A.: Real-time plane extraction from depth images with the randomized hough transform. In: Computer Vision Workshops (ICCV Workshops). pp. 1084–1091. IEEE (2011)

[14] Fair, M., Miller, D.P.: Automated staircase detection, alignment & traversal. In: Proceedings of International Conference on Robotics and Manufacturing. pp. 218–222 (2001)

[15] 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)

[16] Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), 381–395 (Jun 1981)

[17] Gupta, S., Arbelaez, P., Malik, J.: Perceptual organization and recognition of indoor scenes from RGB-D images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 564–571 (2013)

[18] Gutmann, J.S., Fukuchi, M., Fujita, M.: Stair climbing for humanoid robots using stereo vision. In: Intelligent Robots and Systems.(IROS). Proceedings. IEEE/RSJ International Conference on. vol. 2, pp. 1407–1413 (2004)

[19] Hernandez, D.C., Jo, K.H.: Outdoor stairway segmentation using vertical vanishing point and directional filter. In: International Forum onStrategic Technology (IFOST). pp. 82–86. IEEE (2010)

[20] 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)

[21] 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)

[22] Holz, D., Behnke, S.: Approximate triangulation and region growing for efficient segmentation and smoothing of range images. Robotics and Autonomous Systems (2014)

[23] Holz, D., Holzer, S., Rusu, R.B., Behnke, S.: Real-time plane segmentation using RGB-D cameras. In: RoboCup 2011: Robot Soccer World Cup XV, pp. 306–317. Springer (2012)

[24] 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)

[25] Ishiwata, K., Sekiguchi, M., Fuchida, M., Nakamura, A.: Basic study on step detection system for the visually impaired. In: Mechatronics and Automation (ICMA), IEEE International Conference on (2013)

[26] Koppula, H.S., Anand, A., Joachims, T., Saxena, A.: Semantic labeling of 3D point clouds for indoor scenes. In: NIPS. vol. 1, p. 4 (2011)

[27] Kulyukin, V., Gharpure, C., Nicholson, J., Pavithran, S.: RFID in robot-assisted indoor navigation for the visually impaired. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). vol. 2, pp. 1979–1984 (2004)

[28] Lee, C.H., Su, Y.C., Chen, L.G.: An intelligent depth-based obstacle detection system for visually-impaired aid applications. In: 13th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS). pp. 1–4. IEEE (2012)

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

[30] López-Nicolás, G., Omedes, J., Guerrero, J.J.: Spatial layout recovery from a single omnidirectional image and its matching-free sequential propagation. Robotics and Autonomous Systems (2014)

[31] 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)

[32] Mann, S., Huang, J., Janzen, R., Lo, R., Rampersad, V., Chen, A., Doha, T.: Blind navigation with a wearable range camera and vibrotactile helmet. In: Proceedings of the 19th ACM international conference on Multimedia. pp. 1325–1328 (2011)

[33] 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)

[34] 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)

[35] 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)

[36] Molton, N., Se, S., Brady, J., Lee, D., Probert, P.: A stereo vision-based aid for the visually impaired. Image and vision computing 16(4), 251–263 (1998)

[37] Oßwald, S., Gorog, A., Hornung, A., Bennewitz, M.: Autonomous climbing of spiral staircases with humanoids. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4844–4849 (2011)

[38] Oßwald, S., Gutmann, J.S., Hornung, A., Bennewitz, M.: From 3d point clouds to climbing stairs: A comparison of plane segmentation approaches for humanoids. In: 11th IEEE-RAS International Conference on Humanoid Robots. pp. 93–98 (2011)

[39] 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)

[40] Park, C.S., Seo, E.H., Kim, D., You, B.J., Oh, S.R.: Stair boundary extraction using the 2d laser scanner. In: International Conference on Mechatronics and Automation (ICMA). pp. 1538–1543. IEEE (2011)

[41] Pérez-Yus, A., López-Nicolás, G., Guerrero, J.J.: Detection and Modelling of Staircases Using a Wearable Depth Sensor. In: Second ECCV Workshop on Assistive Computer Vision and Robotics (2014)

[42] Pradeep, V., Medioni, G., Weiland, J.: Robot vision for the visually impaired. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 15–22 (2010)

[43] 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)

[44] Qian, X., Ye, C.: NCC-RANSAC: A fast plane extraction method for navigating a smart cane for the visually impaired. In: International Conference on Automation Science and Engineering (CASE). pp. 261–267. IEEE (2013)

[45] Ren, X., Bo, L., Fox, D.: Rgb-(d) scene labeling: Features and algorithms. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2759–2766. IEEE (2012)

[46] Rodríguez, A., Yebes, J.J., Alcantarilla, P.F., Bergasa, L.M., Almazán, J., Cela, A.: Assisting the visually impaired: obstacle detection and warning system by acoustic feedback. Sensors 12(12), 17476–17496 (2012)

[47] Rusu, R.B., Cousins, S.: 3D is here: Point cloud library (PCL). In: International Conference on Robotics and Automation (ICRA). pp. 1–4. IEEE (2011)

[48] Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3D point cloud based object maps for household environments. Robotics and Autonomous Systems 56(11), 927–941 (2008)

[49] Se, S., Brady, M.: Vision-based detection of staircases. In: Asian Conference on Computer Vision (ACCV). vol. 1, pp. 535–540 (2000)

[50] Shoval, S., Borenstein, J., Koren, Y.: The Navbelt - A computerized travel aid for the blind based on mobile robotics technology. Transactions on Biomedical Engineering 45(11), 1376–1386 (1998)

[51] Shoval, S., Ulrich, I., Borenstein, J., et al.: Computerized obstacle avoidance systems for the blind and visually impaired. Intelligent systems and technologies in rehabilitation engineering pp. 414–448 (2000)

[52] Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: European Conference on Computer Vision (ECCV), pp. 746–760. Springer (2012)

[53] 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)

[54] Tian, Y., Yang, X., Yi, C., Arditi, A.: Toward a computer vision-based wayfinding aid for blind persons to access unfamiliar indoor environments. Machine vision and applications 24(3), 521–535 (2013)

[55] Ueda, T., Kawata, H., Tomizawa, T., Ohya, A., Yuta, S.: Visual information assist system using 3D SOKUIKI sensor for blind people, system concept and object detecting experiments. In: 32nd Annual Conference on Industrial Electronics (IECON). pp. 3058–3063. IEEE (2006)

[56] 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). IEEE (2013)

[57] 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)

[58] Wang, S., Wang, H.: 2D staircase detection using real adaboost. In: 7th International Conference on Information, Communications and Signal Processing (ICICS). pp. 1–5. IEEE (2009)

[59] Wang, Y., Ji, R., Chang, S.F.: Label propagation from imagenet to 3D point clouds. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3135–3142. IEEE (2013)

[60] Wang, Z., Liu, H., Qian, Y., Xu, T.: Real-time plane segmentation and obstacle detection of 3D point clouds for indoor scenes. In: Computer Vision ECCV. Workshops and Demonstrations. pp. 22–31. Springer (2012)

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

[62] Zöllner, M., Huber, S., Jetter, H.C., Reiterer, H.: NAVI–a proof-of-concept of a mobile navigational aid for visually impaired based on the Microsoft Kinect. Springer (2011)