# FEATURE RECOGNITION AND OBSTACLE DETECTION FOR DRIVE ASSISTANCE IN INDOOR ENVIRONMENTS

A thesis submitted in partial fulfilment of the requirements for the

Degree

of a Master of Science in Computer Science and Software Engineering

at the University of Canterbury

by Chunhui Zheng

Supervised by Richard Green

University of Canterbury

2012

# Abstract

The goal of this research project was to develop a robust feature recognition and obstacle detection method for smart wheelchair navigation in indoor environments. As two types of depth sensors were employed, two different methods were proposed and implemented in this thesis. The two methods combined information of colour, edge, depth and motion to detect obstacles, compute movements and recognize indoor room features. The first method was based on a stereo vision sensor and started with optimizing the noisy disparity images, then, RANSAC was used to estimate the ground plane, followed by a watershed based image segmentation algorithm for ground pixel classification. Meanwhile, a novel algorithm named a standard deviation ridge straight line detector was performed to extract straight lines from the RGB images. The algorithm is able to provide more useful information than using the Canny edge detector and the Hough Transform. Then, the novel drop-off detection and stairs-up detection algorithms based on the proposed straight line detector were carried out. Moreover, the camera movements were calculated by optical flow. The second method was based on a structured light sensor. After RANSAC ground plane estimation, morphology operations were applied to smooth the ground surface area. Then, an obstacle detection algorithm was carried out to create a top-down map of the ground plane using inverse perspective mapping and segment obstacles using a region growing-based algorithm. Both the drop-off and open door detection algorithms employ the straight lines extracted from depth discontinuities maps. The performance and accuracy of the two proposed methods were evaluated. Results show that the ground plane classification using the first method achieved 98.58% true positives, and the figure improved with the second method to 99%. The drop-off detection algorithms using the first method also achieved good results, with no false negatives found in the test video sequences. The system provided the top-down maps of the surroundings to detect and segment obstacles correctly. Overall, the results showing accurate distances to various detected indoor features and obstacles, suggests that this

proposed colour/edge/motion/depth approach would be useful as a navigation aid through doorways and hallways.

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgements

First, I would like to thank my supervisor Dr. Richard Green for sharing his valuable time to help me whenever I needed him and for giving me lots of helpful information to complete this project. The continuous support and vital encouragement that he gave kept me going all the time.

I would also like to thank my industry mentor Ian Palmer from Dynamic Controls, Christchurch. I thank him for providing me with the vital resources for this project and with insight into the wheelchair industry.

Moreover, I would like to thank NZi3 who kindly provided me with a comfortable workspace and the use of modern computer equipment. NZi3 gave me an opportunity and a work environment to challenge myself to do this project and prove that I can do things on my own.

I also appreciate the funding received from the Foundation for Research, Science and Technology (FRST) for this project. I would like to thank Richard and Ian again at this time. I thank Richard for helping me to put forward my research proposal to Dynamic Controls and Ian for transforming my proposal into a practicable project to get the FRST funding.

Finally, I would like to thank all the people that have helped and supported me, especially my family, my friends and my classmates. I would have not finished this project without their guidance and helpful supports.

# Chapter 1
# Introduction

Some disabled people have difficulty in driving a standard powered wheelchair manually and safely. Disabled people such as those with spasticity or cognitive impairment cannot easily control a joystick or a switch to operate a wheelchair. Therefore, smart wheelchairs have been developed to provide these disabled people with various types of assistance such as obstacle avoidance, path planning and performing other specific navigation tasks.

Smart wheelchair technologies have been researched for more than two decades and usually smart wheelchairs employ a collection of sensors such as sonar, infrared sensors, laser range finders or visual sensors with technologies used for mobile robots.

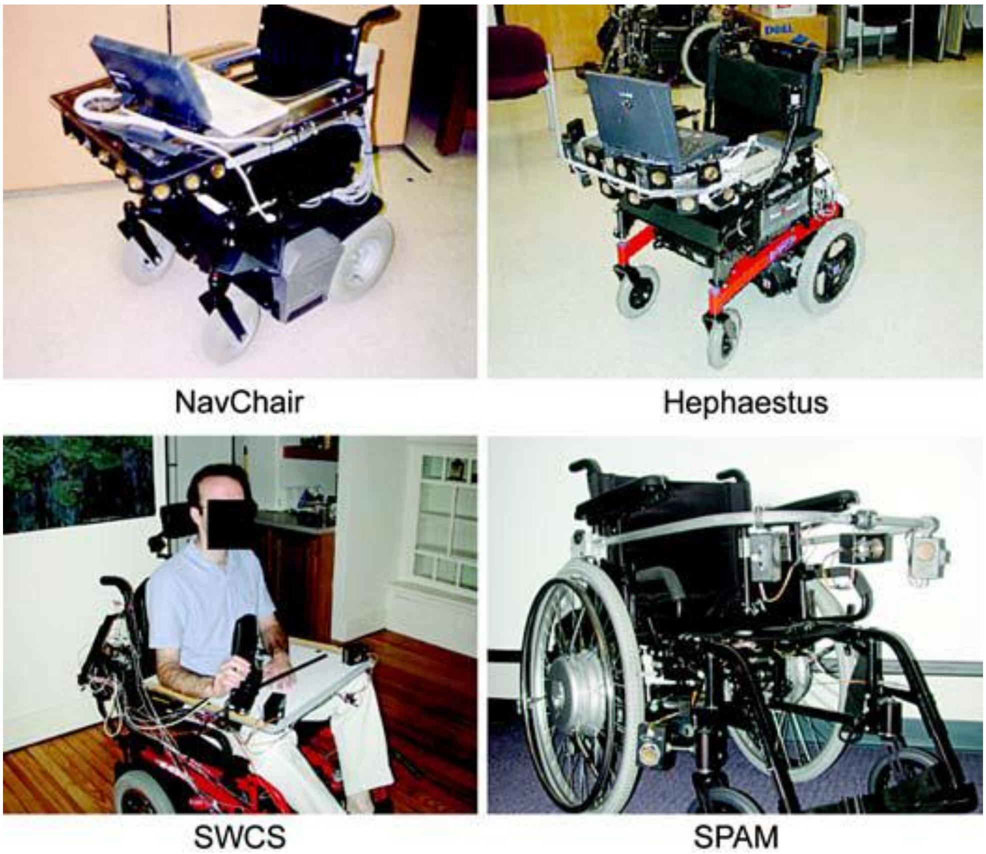


Figure 1.1: **Gallery of smart wheelchairs.** SWCS = Smart Wheelchair Component System, SPAM = Smart Power Assistance Module. (Figure reproduced from [1]).

A study shows that in 2008 there were 1.4 to 2.1 million people in the United Sates needing a smart powered wheelchair [2]. Although smart wheelchairs are widely desired [2, 3], there are only a few smart wheelchairs commercially available [1].

Simpson [1] revealed several issues that limit smart wheelchairs from becoming widely used: 1) the trade-off between the cost and accuracy of the sensors, 2) the lack of a standard communication protocol between wheelchair input devices and add-on modules, and 3) the lack of clinical acceptance. Among these issues, the trade-off between the cost and accuracy of the sensors is the most important problem. Until reliable obstacle and hazard avoidance technologies with cheap sensors are developed, the smart wheelchairs will remain in the research stage for many years.

Therefore, this research project focuses on investigating and developing a robust feature recognition and obstacle detection method for smart wheelchair navigation in indoor environments. In this project, two different methods have been proposed and developed. One is based on a passive stereo vision sensor, and the other is based on an active structured light sensor. The two types of sensors both provide high resolution and accurate depth images; hence, both of them are employed and compared in this research. The two methods combine information on colour features, edge maps, range information and motion analysis to provide robust indoor room feature recognition and obstacle detection. The system is able to calculate motion information, detect ground plane and obstacles, recognize static indoor features such as drop-offs, stairs up, open doors, and compute distances to various detected indoor features and obstacles.

This chapter presents an introduction to the thesis. Section 1.1 provides a summary of the objectives and then Section 1.2 represents an overview of the thesis. Finally, Section 1.3 gives a list of major contributions.

## 1.1 Objectives of the Thesis

The goal of this research project is to develop robust vision-based indoor feature recognition and obstacle detection through doorways and hallways where obstacles and hazards are present.

This project has the following objectives:

1. To develop a robust method for indoor static feature recognition. This includes ground plane detection, doorway and hallway detection, stairs up and drop-off detection.
2. To develop a robust method for obstacle detection and segmentation.
3. To provide motion information of the wheelchair from vision information alone.

## 1.2 Thesis Overview

The thesis is organized as follows:

**Chapter 2** – provides background information on depth data acquisition technologies, and discusses the theories behind stereo vision and structured light sensors.

**Chapter 3** – provides an overview of the vision-based obstacle and hazard detection methods. Vision-based obstacle detection approaches are classified into three categories: knowledge-based, motion-based and depth sensor based. Related works for ground plane estimation and indoor feature detection are also reviewed.

**Chapter 4** – explains the proposed stereo vision based feature recognition and obstacle detection method and the proposed structured light based feature recognition and obstacle detection method.

**Chapter 5** – presents the results of the proposed stereo-vision-based feature recognition and obstacle detection methods and proposed structured-light-based feature recognition and obstacle detection method. Comparisons between the two depth sensors are also made.

**Chapter 6** – provides a conclusion of the thesis. Future development and improvement are also discussed in this chapter.

## 1.3 Contributions

The contributions of the thesis are listed as follows:

- Developing a method combining information on colour, edge, depth and motion to detect ground plane, obstacles, drop-offs and stairs up, compute movements and measure the distance to the detected obstacles and hazards using a stereo vision sensor.
- Developing a method combining information on edge, depth and motion to detect ground plane, obstacles, drop-offs and open doors, and measure the distance to the detected obstacles and hazards using a structured light sensor.
- Developing a method using disparity interpolation/smoothing/noise removal to optimize the noisy disparity images generated from a stereo camera.
- Developing a ground pixel classification algorithm using watershed based image segmentation and RANSAC ground plane estimation.
- Proposing and developing a novel algorithm named "standard deviation ridge straight line detector" for extracting straight lines from the RGB images. This algorithm is able to provide more useful information than using the Canny edge detection and Hough Transform.
- Proposing and developing a novel drop-off detection algorithm and a novel stairs detection algorithm using the proposed standard deviation ridge straight line detector algorithm.
- Developing an obstacle detection algorithm which creates a top-down view of the ground plane using inverse perspective mapping and segment obstacles using a region growing-based algorithm.
- Developing a drop-off detection algorithm and an open door detection algorithm by extracting straight lines from a depth discontinuities map using a structured light camera.

# Chapter 2
# Depth Data Acquisition

## 2.1 Introduction

Smart wheelchairs require sensors to observe their surroundings and detect obstacles. Depth sensors such as sonar, infrared sensors, laser range finders and visual sensors are often used by smart wheelchairs to obtain depth data.

Simpson [1] indicated that the ideal sensors for smart wheelchairs should be accurate, low cost, small, lightweight, with low power consumption and not influenced by environment changes. However, no individual sensor exists that satisfies all these requirements.

Both sonar sensors and infrared sensors are small and low cost, but neither of them is adequate for drop-off detection. In addition, the accuracy of sonar sensors can be influenced by the change of direction to the objects, smooth or sound absorbent objects; and the accuracy of infrared sensors can be influenced by light absorbent objects, refractive or transparent surfaces and sunlight [1]. Furthermore, the infrared sensor cannot work under sunlight.

Compared with sonar and infrared sensors, laser range finders (LRFs) are more accurate for drop-off detection. However, the 3D LRFs are very expensive and large-sized, and cannot be commonly used with smart wheelchairs. The 2D LRF can only detect a 2D plane in the 3D environment; hence, it has difficulty in detecting overhanging objects such as chairs, tables and stairways. Although using a Pan-Tilt-unit to rotate a 2D LRF can achieve 3D scanning, it is time consuming to move the pan [4].

To overcome the drawback of using a single type of sensor, smart wheelchairs (e.g. Wheelesley [5], OMNI [6], CCPWNS [7]) often combine different types of sensors to improve the performance of obstacle and drop-off detection. Although sensor integration provides more accurate and robust depth data, it also leads to difficulties such as a more complex algorithm and higher computation power [8].

With the development of computer vision technology, cameras have become more suitable for use with smart wheelchairs assisting navigation. Smart wheelchairs (e.g., Rolland [9], MAid [10]) already use computer vision to identify natural or artificial landmarks in the environment. In general, cameras are relatively small, cheap and can provide great sensor coverage. In addition, images captured by cameras can provide more information than other types of sensors. Nevertheless, extracting useful information from images is a difficult problem.

This chapter provides background information on depth data acquisition technologies. The techniques and sensor devices used for producing depth image are reviewed briefly in section 2.2. Sections 2.3 and 2.4 focus on stereo vision and structured light techniques, and explain the theories behind them.

## 2.2 Depth Imaging

Cameras can be used to generate depth data. In a depth image, each pixel value shows the distance from the camera to a point in the scene. Depth images can be produced by single cameras using *structure from motion* techniques or range cameras such as time-of-flight, stereo vision and structured light.

### 2.2.1 Structure From Motion

Structure from motion refers to the process of obtaining depth information from sequential images by motion analysis. This method has low accuracy and requires large computational overheads to solve the problem of finding correspondence between images. Moreover, the method requires a moving camera to generate data.

### 2.2.2 Time-of-flight Camera

A time-of-flight (TOF) camera acquires depth information by measuring the time-of-flight of light signals between the camera and objects. A TOF camera has the advantage of its efficient distance algorithm and high frame rate. However, its image resolution is low and its accuracy of measurements can be influenced by sunlight, a scene with multiple reflections, or other TOF cameras [11].

### 2.2.3 Stereo Camera

A stereo camera system captures two or more images at the same time from different perspectives. The system generates depth information by finding corresponding points from a pair of images. The principle of stereo vision is explained in section 2.3. Compared with the structure from motion and time-of-flight cameras, a stereo camera can obtain higher resolutions and consequently more accurate depth images, but nevertheless, a stereo camera has several disadvantages. First, it has difficulty obtaining depth information in non-textured areas. Second, identifying corresponding points in different images can be complex and computationally expensive. Moreover, depth values from the stereo camera can be very noisy without good ambient lighting.

### 2.2.4 Structured light Camera

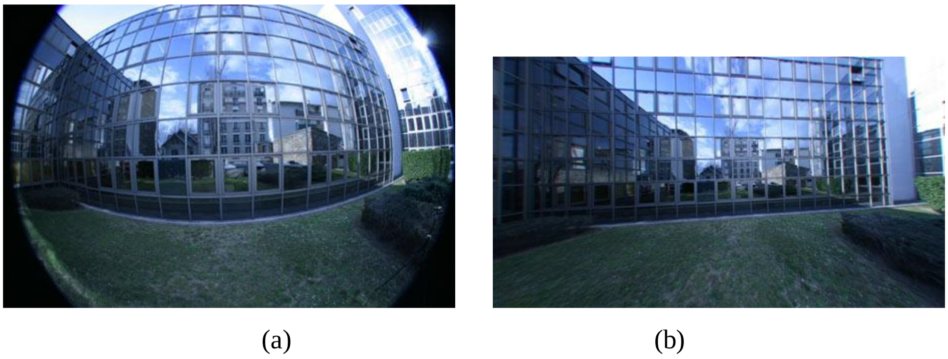
Structured light cameras emit light patterns onto the surface of an object, and resolve distance by calculating the level of distortion in the pattern. Structured light cameras can generate high resolution and accurate depth images, and are still effective in poorly illuminated and minimally textured environments. However, structured light cameras have poor performance under sunlight.

## 2.3 Stereo Vision

Stereo vision simulates the human's vision system and in a biological vision system, the binocular parallax can infer objective distance. By the same principle, in a stereo vision system, two or more images of the same scene are taken from different cameras. The depth information is obtained in the form of disparity which is the correspondence between points in one image and the same points in the other image.

### 2.3.1 Stereo Imaging

Bradski and Kaehler [12] indicated that four steps are involved in stereo imaging when two cameras are used. First, lens distortion must be removed from the images. An undistorted image is shown in Figure 2.1. Second, the image rectification is involved. The image planes of different cameras are reprojected into the same plane, which makes the images become row-aligned and rectified. Third, a correspondence process occurs, where the pair of rectified images are compared to generate a disparity map. The more corresponding pixels that are detected, the more depth points can be obtained. Fourth, triangulation techniques are used to convert disparity values into depth information.



(a) (b)

Figure 2.1: **Distortion removal.** (a) Lens with fisheye distortion. (b) Distortion removed from (a). (Figure reproduced from [13] ).

The geometry of a stereo camera is shown in Figure 2.2, where $T$ is the distance between the cameras' centres of projection, and $f$ is the focal length of the cameras. Both baseline $T$ and focal length $f$ are assumed to be known. $x^l$ and $x^r$ are perspective projections of object $P$ onto the stereo camera image planes. The disparity value $d$ of the point $P$ is the horizontal shift between $x^r$ and $x^r$. Therefore, the distance to the object $P$ along the camera z-axis is expressed as:

$$Z = \frac{fT}{d} \quad \text{where} \quad d = x^l - x^r \qquad (2.1)$$

As depth and disparity are inversely related, high depth resolution is only obtained on objects that are relatively close to the camera, as shown in Figure 2.3.

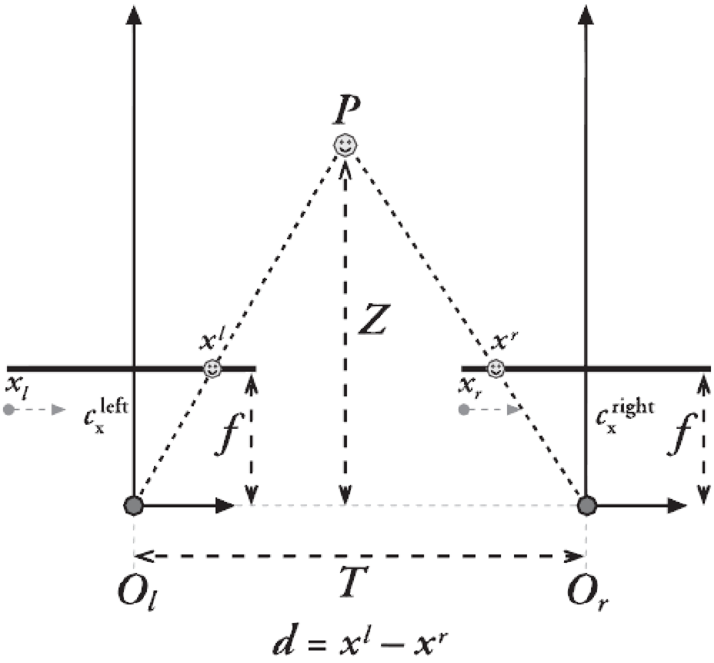


Figure 2.2: **Geometry of the stereo camera.** (Figure reproduced from [12]).

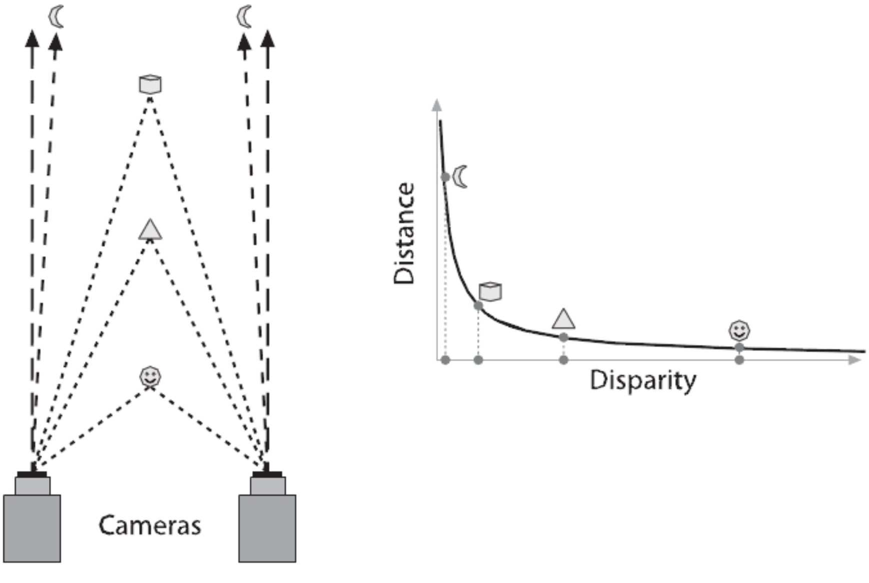


Figure 2.3: **Inverse relationship between depth and disparity.** (Figure reproduced from [12]).

### 2.3.2 Bumblebee 2 Stereo Camera

The Bumblebee 2 stereo camera was developed by Point Grey Research Inc. This stereo camera system has the advantage of accurately pre-calibrated for removing lens distortion and camera misalignment. A stereo vision system using two Logitech web cameras is compared with the Bumblebee 2 stereo cameras in [13]. The results show that the Bumblebee 2 stereo cameras perform significantly better than the Logitech web cameras. When using the Bumblebee 2 stereo camera, the camera's intrinsic and extrinsic parameters are usually retrieved from the internal memory of the Bumblebee 2 stereo camera. These parameters include focal length, baseline distance and principal point, which are normally evaluated during calibration calculations when using two web cameras. In addition, the Bumblebee 2 stereo cameras can retrieve more disparity values than Logitech web cameras. Depth values were significantly more accurate when using a Bumblebee 2 stereo camera.

## 2.4 Structured Light

Figure 2.4 illustrates a light pattern projected by a structured light camera used to generate depth information. However, the visible light pattern projected on the objects in the scene can cause changes in the colour and texture of the objects’ surfaces. In order to avoid the disadvantages, invisible or imperceptible structured light techniques are used to optimise the structured light camera's processing [14]. The use of infrared light is one of the most commonly applied methods.

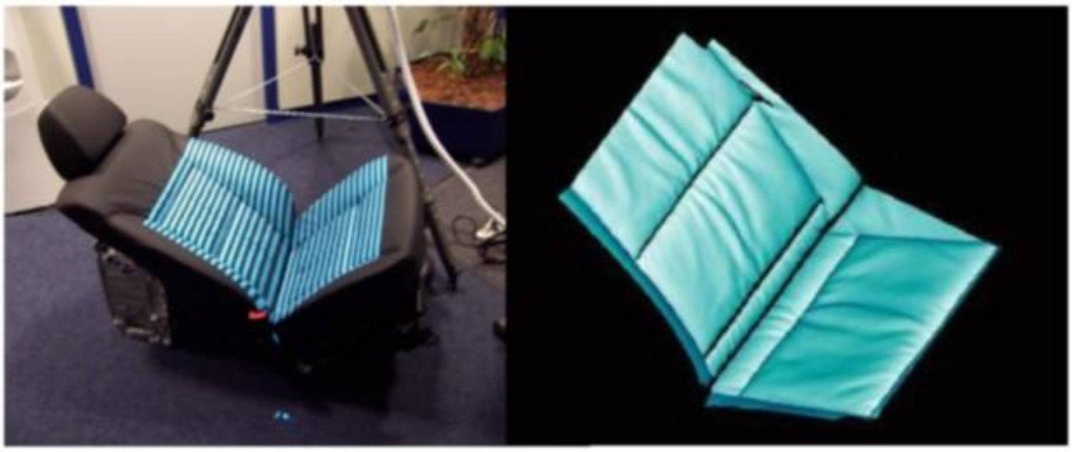


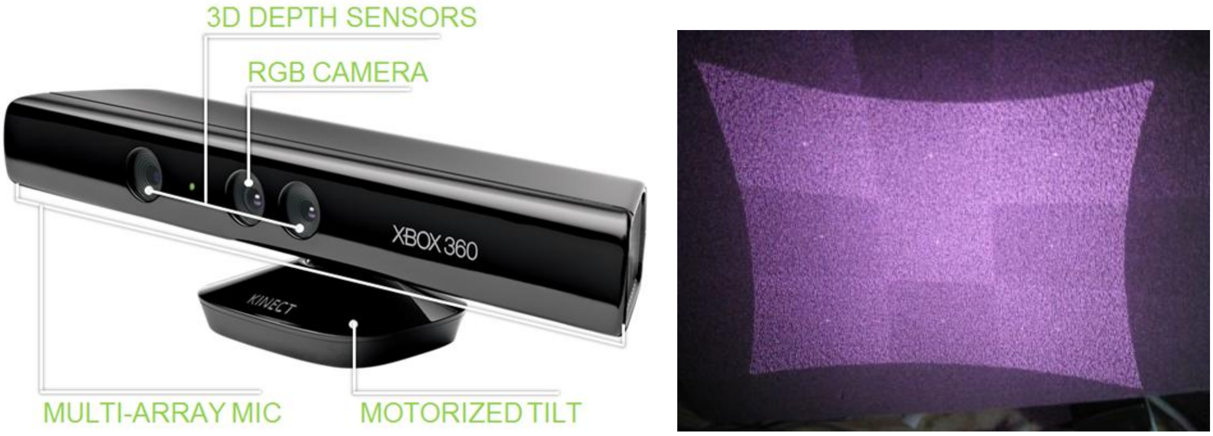
Figure 2.4: **3D survey of a car seat.** (Figure reproduced from [15]).

### 2.4.1 Kinect camera

Structured light cameras have been available for years, but the high prices of the structured light cameras limit their application in autonomous navigation [16]. However, Microsoft launched a commercial low-cost structured light sensor Kinect for the Xbox 360 gaming console in November 2010 [17]. It is still able to generate accurate depth information about the surrounding environment. Therefore, the Kinect sensor has become an ideal substitute for other depth sensors.

As shown in Figure 2.5, the Kinect sensor consists of the following:

- Colour camera
- 3D depth sensing system
- Multi-array microphones
- Accelerometer
- Motorized tilt



(a) (b)

Figure 2.5: **Kinect sensor and its IR pattern.** (a) Kinect sensor. (b) Structured infrared pattern projected by Kinect depth sensors.

The Kinect sensor obtains 3D information about the surrounding environment by combining data from a standard RGB camera and infrared based depth sensors. A structured infrared pattern of dots in Figure 2.5 (b) is projected on to the environment and viewed by an infrared camera. The structured light pattern appears distorted after it reaches the objects in the scene. The depth information of those objects can be retrieved according to the scale of distortion.

## 2.5 Summary

In this chapter, several depth data acquisition technologies are discussed briefly with specific focus on stereo vision and structured light cameras. Table 2.1 shows a summary of the comparisons between the sensors that are often used by smart wheelchairs. Cameras can provide the most significant information, so they have been widely used on smart wheelchairs recently. Table 2.2 lists the advantages and disadvantages of several techniques and range cameras that can be used to produce depth images. As both stereo cameras and structured light cameras can provide accurate and high resolution depth images, both of them are employed and compared in this research project to find a more suitable sensor for smart wheelchairs. Stereo vision and structured light systems are introduced in sections 2.3 and 2.4. A comparison between a stereo vision camera Bumblebee 2 and a structured light camera Kinect can be found in Chapter 5.

|  | Advantages | Disadvantages |
|---|---|---|
| Sonar sensor | - Small, very cheap | - Influenced by direction change, smooth or sound absorbent objects<br>- Not adequate for drop-off detection |
| Infrared sensor | - Small, very cheap<br>- Can work without light | - Cannot work under sunlight<br>- Influenced by light absorbent objects, refractive or transparent surfaces<br>- Not adequate for drop-off detection |
| 2D LRF | - Small, cheap, accurate | - Difficult to detect overhanging objects |
| 3D LRF | - Accurate | - Large, very expensive |
| Camera | - Great area coverage<br>- More information can be retrieved | - Difficult to extract useful information |

Table 2.1: **A summary of the comparisons between different types of sensors.**

|  | Advantages | Disadvantages |
|---|---|---|
| Structure from motion | - Cheap hardware | - Low accuracy<br>- Large computation<br>- Requires camera shifting to generate data |
| TOF camera | - High frame rate | - Low image resolution<br>- Influenced by the sunlight, multiple reflections, other TOF cameras |
| Stereo camera | - Accurate<br>- High resolution | - Low accuracy of home-made stereo cameras<br>- High cost of commercial stereo cameras<br>- Large computation<br>- Cannot work without ambient light |
| Structured light camera | - Cheap, accurate<br>- High resolution | - Cannot work under sunlight |

Table 2.2: **A summary of the comparison between the four depth imaging techniques.**

# Chapter 3
# Obstacle and Hazard Detection Overview

## 3.1 Introduction

Autonomous navigation for vehicles and mobile robots has been widely researched in the last two decades. An extensive survey of vision-based navigation is presented in [18]. For navigation in an unknown environment, obstacle and hazard detection is extremely important as the navigation is achieved by observing the obstacles and hazards in the environment. Moreover, obstacle and hazard detection is essential for path planning and other complex navigation tasks. When driving in an indoor environment, various potential hazards are present, including steps, upward stairs, walls, furniture, people, doorways and ramps. Table 3.1 lists several kinds of hazards in indoor environments.

| **Hazards** | | **Examples** |
|---|---|---|
| Drop offs | | Downward stairs, steps |
| Obstacles | Static | Walls, furniture |
| | Dynamic | People, doors |
| | Transparent | Glass doors, glass walls |
| Overhangs | | Table tops |
| Inclines | | Wheelchair ramps |
| Narrow regions | | Doorways, elevators |

Table 3.1: **Common hazards in indoor environments.** (Figure adapted from [19]).

According to Burschkal et al. [20], objects around a mobile robot are classified into four regions (Figure 3.1). Positive obstacle (object B) in region II and negative obstacle (object D) in region IV need to be detected. These objects can be walls, tables, drop-offs and gaps. Object C in region I and object A in region III can be ignored, because they do not affect the movement of mobile robots. These objects can be ceiling lights and small bumps.

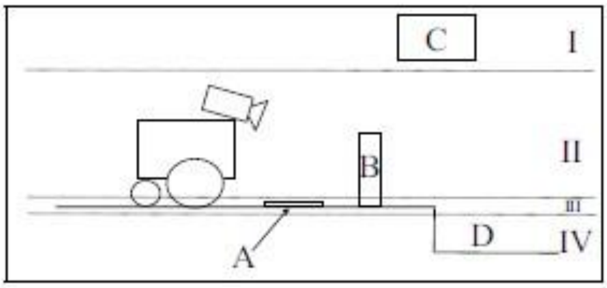


Figure 3.1: **Classification of objects detected by a mobile robot.** (Figure reproduced from [20]).

The following section provides an overview of vision-based obstacle detection, followed by literature reviews of ground plane detection. The feature recognition techniques for indoor features such as doorways, hallways, drop-offs and stairs are discussed in section 3.4.

## 3.2 Obstacle Detection

Various vision-based obstacle detection approaches have been proposed in the literature, which can be classified into one of the following three categories: 1) *knowledge-based*, 2) *motion-based*, and 3) *depth-camera-based* method.

### 3.2.1 Knowledge-based Obstacle Detection

Knowledge-based methods exploit *a priori* knowledge of the obstacle [21-23]. Features such as colour, edge, shape, textures and so on are used to detect obstacles. Ulrich and Nourbakhsh [21] presented an appearance based obstacle detection method with a monocular camera. The method is based on the colour appearance of each image pixel. Objects with different colour appearance from the ground are classified as obstacles. Three assumptions are made to perform robust obstacle detection. First, obstacles have a different appearance from the ground. Second, the ground plane is relatively flat. Third, there is no overhanging obstacle. Figure 3.2 shows some of their experimental results. Based on Ulrich and Nourbakhsh's work, Tada et al. [22] introduced an appearance based obstacle detection approach for corridor passing. They showed that two boundary lines could be generated by the two side walls and corridor floor. The obstacles between the two boundary lines are detected based on their appearance. The size and the position of obstacles are also used for determining the actions of avoidance or stopping when the obstacles are detected; otherwise the movement keeps to the middle of the corridor. Viet and Marshall [23] presented an innovative algorithm using textures and colours to detect obstacles effectively. They revealed that colours were cheap to discover and process, and in addition, textures can improve the performance of colour classification.

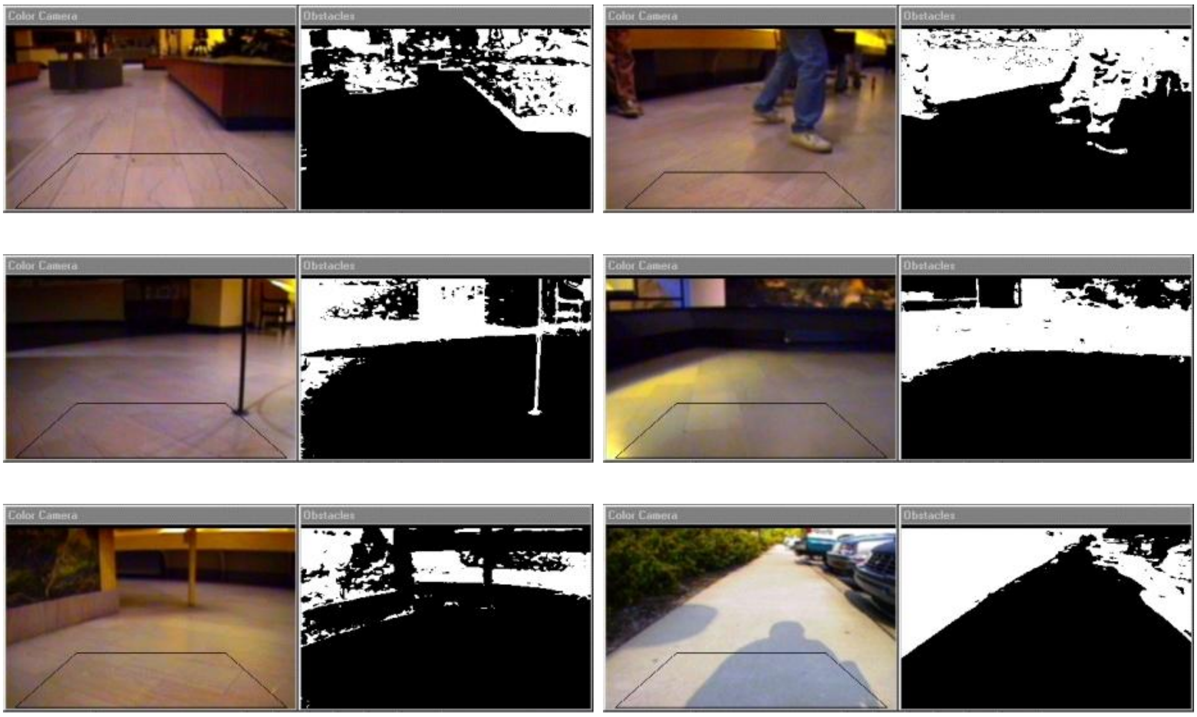


Figure 3.2: **Ulrich and Nourbakhsh's appearance based obstacle detection.** (Figure reproduced from [21].)

Knowledge-based obstacle detection methods can be efficient in a simple environment without clutter, but can easily fail when the background environment contains many similar colours, different textures and many geometric lines and edges [24]. A problem with colour-based obstacle detection is that the colour of an object depends on illumination and reflection, for example, appearance colour in daytime differs from colour at night time. Another problem of edge and line based obstacle detection is that the threshold value for the edge detection needs to be carefully chosen to obtain robust and reliable obstacle detection. Moreover, previous work on knowledge-based obstacle detection shows that this kind of approach is unable to detect overhanging obstacles such as the edge of a table.

### 3.2.2 Motion-based Obstacle Detection

Motion-based obstacle detection is proposed by many researchers [25-28]. Motion information is extracted from successive images, and can be calculated from optical flow. As illustrated in Figure 3.3, optical flow describes the motion with vectors at feature points in a captured image. When a vehicle or a mobile robot is moving forward, the pixels in the centre of the image tend to flow more slowly than the pixels near the edge (see Figure 3.3 (c)).

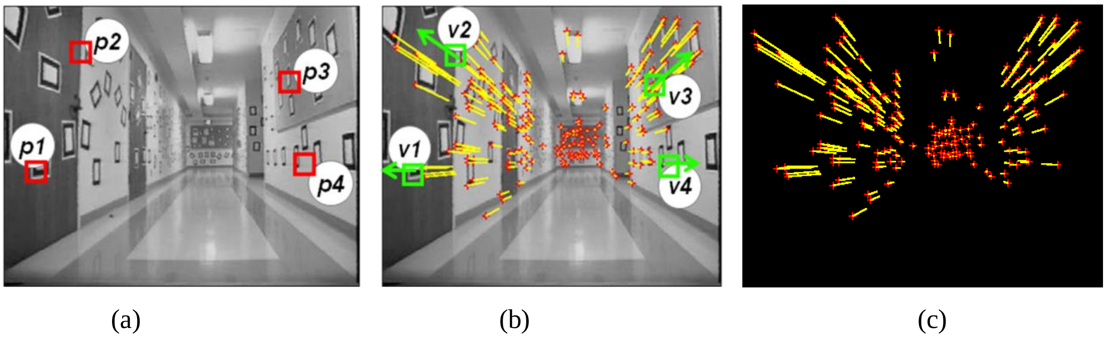


Figure 3.3: **Optical flow.** (a) Strong features such as corners are selected from each frame. (b) Selected features are tracked over time and their movement is converted into vectors. (c) Flow vectors as the camera moves down the hallway. (Figure adapted from [12]) (Original images courtesy of Jean-Yves Bouguet).

Motion information can also be used for calculating the remaining time to contact with a surface [29]. As shown in Figure 3.4 (a), if an object is travelling on a collision course with a robot with constant relative speed $v$, and the distance between robot and object is $r$, the time-to-impact ($TTC$) is calculated by:

$$TTC = \frac{r}{v} \tag{3.1}$$

Another way to calculate $TTC$ is to use the velocity vector $v$ and the vector to a point on the object $r$. If $\phi$ is the angle between the two vectors (see Figure 3.4 (b)), the $TTC$ can be calculated by:

$$TTC = \frac{cos\phi \times sin\phi}{\phi} \tag{3.2}$$

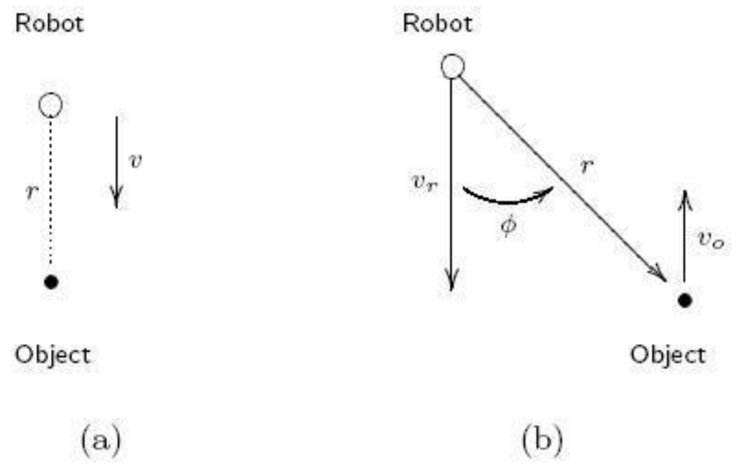


Figure 3.4: **Time to Contact.** (a) The robot and the object are on a straight collision course. (b) The robot and the object are not on a collision course. (Figure reproduced from [29].)

Some motion-based obstacle detection algorithms recover the depth information of the environment. Demonceaux et al. [30] introduced a motion-based road obstacle detection algorithm which comprises three steps. First, the 3D shape of the road is reconstructed by a five-step method which includes training, road detection, localization, tracking, and 3D reconstruction. Second, road motion analysis is performed by a wavelets approach. Finally, obstacle detection is achieved based on a Bayesian hierarchical model. Figure 3.5 illustrates their results of the obstacle detection. However, those vehicles far away from the camera have not been detected owing to the low relative motion between the road and these vehicles (see Figure 3.5 (c, d)). Wekel et al. [31] also proposed an obstacle detection system based on ground motion estimation and stereo matching as shown in Figure 3.6. The system converts the robot’s camera view of the scene into a top-down “bird’s-eye” view. Then, the images are segmented, and each segment is classified as a ground plane or obstacles. However, in order to generate a bird's-eye view map, their system requires a known object (a chessboard) on the ground plane as illustrated in Figure 3.6 (a) for initial calibration. Furthermore, Naito et al. [27] presented an algorithm to detect obstacles and estimate 3D information of the scene by extracting the optical flow along the edge of the image. Wang et al. [32] also developed an obstacle detection system that could estimate mobile robot ego-motion and obstacle depth according to the optical flow.

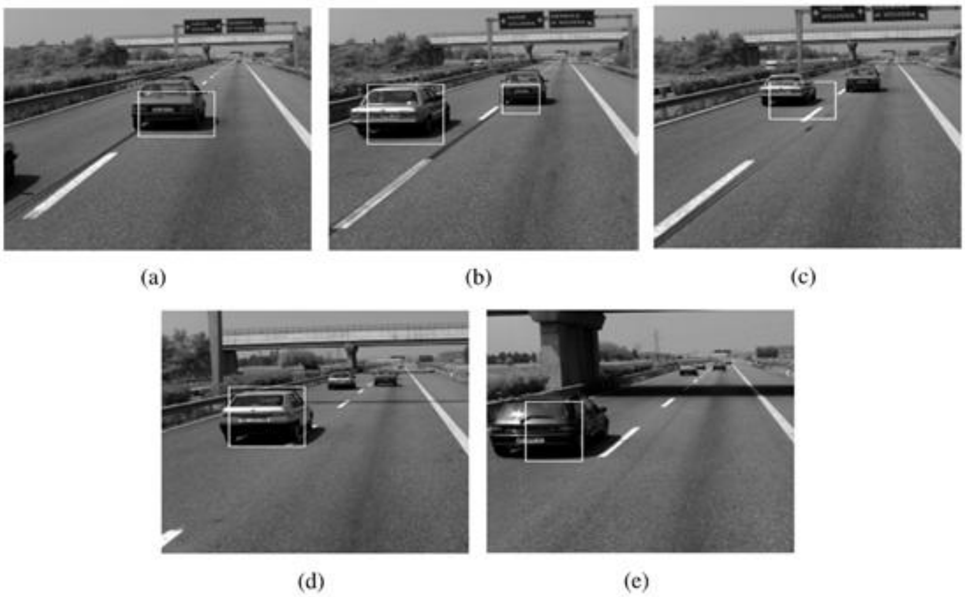


Figure 3.5: **Demonceaux et al.'s motion based road obstacle detection.** (Figure reproduced from [30].)

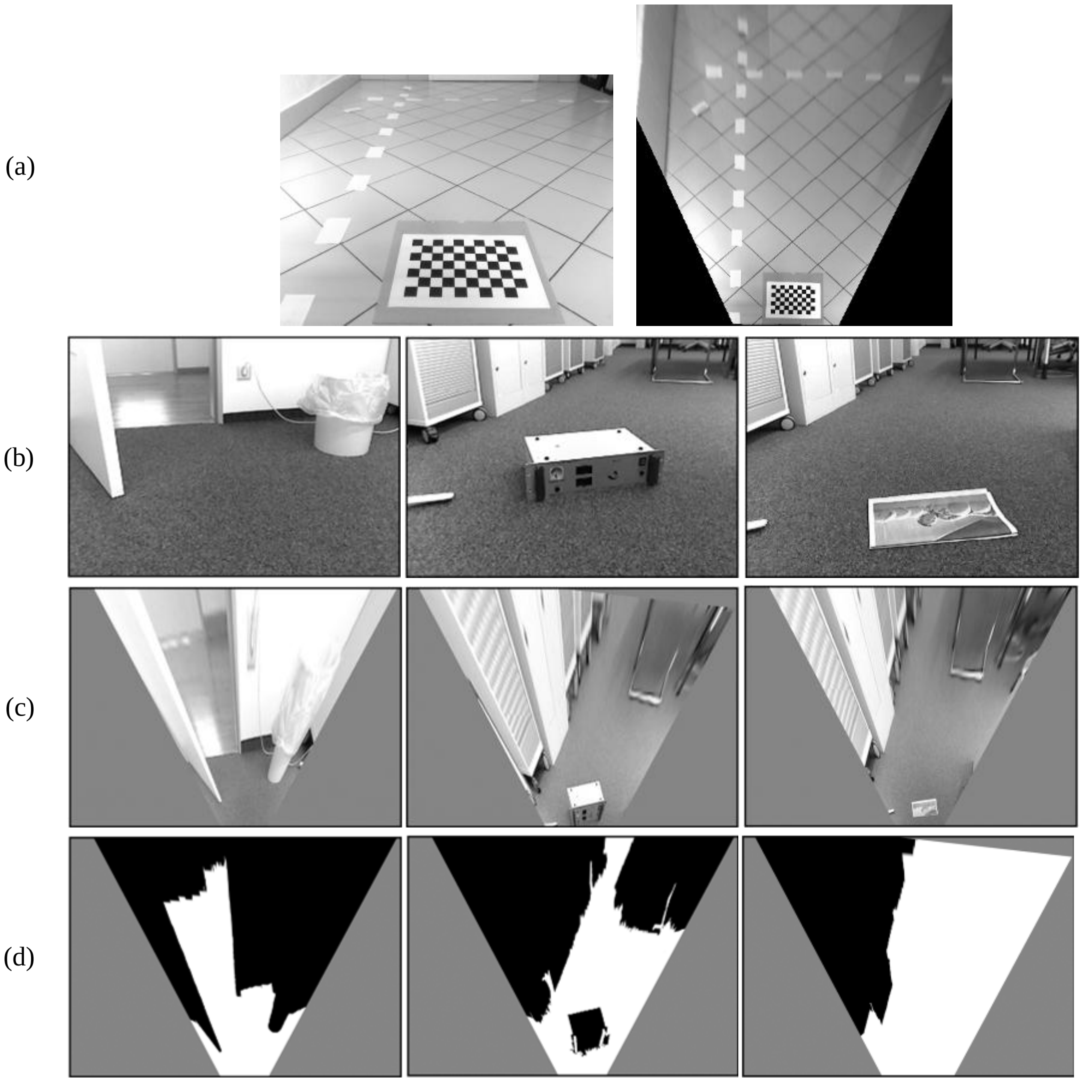


Figure 3.6: **Wekel et al.'s obstacle detection.** (a) System initial calibration. (b) Raw images. (c) Bird's-eye view. (d) Obstacle and ground plane classification. (Figure reproduced from [31].)

Other motion-based obstacle detection methods extract 2D information instead of 3D reconstruction. Sarcinelli-Filho et al. [25] developed a fast algorithm to compute optical flow for obstacle detection without restoring range information. Unreliable optical flow vectors are filtered out based on their measured confidence. According to the optical flow, the objects in the image are segmented from each other. Then, time to contact is calculated for each object, and used to control the speed and the action of the mobile robot. Shen et al. [28] proposed an algorithm for obstacle detection in an outdoor unstructured environment using optical flow without 3D reconstruction. Optical flow at feature points is calculated, and then rotation of the camera and FOE (focus of expansion) are evaluated and refined separately. Finally, inverse time to contact is calculated and obstacles are detected. Figure 3.7 shows some of their experimental results.

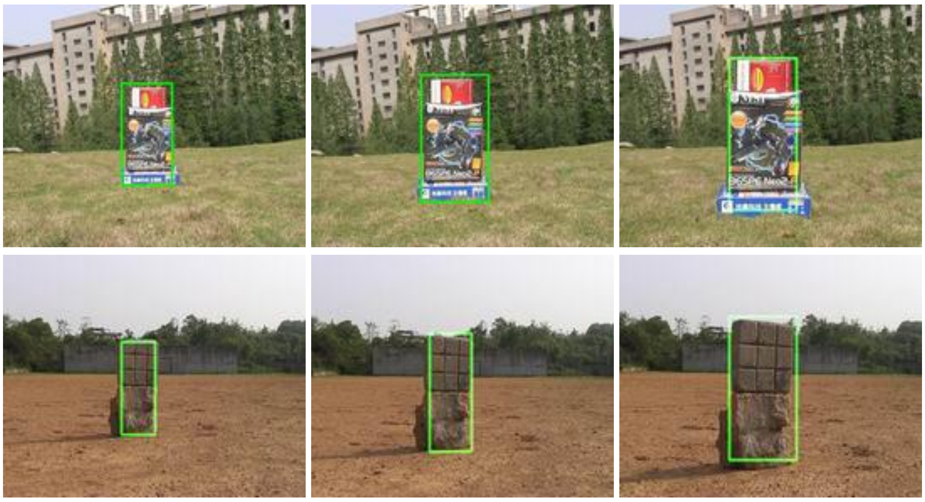


Figure 3.7: **Shen et al.'s optical flow based obstacle detection.** (Figure reproduced from [28].)

Although solving a 2D problem of obstacle detection is relatively simpler and faster than solving a 3D problem [33], Shah [34] listed several weaknesses of 2D-based obstacle detection. First, the 2D-based obstacle detection only solved the problem of 'directions to avoid'. Second, as there is no depth information of the scene, the 2D-based method cannot provide an absolute position of objects in the world or the relationships between these objects. Third, there is no information about the camera ego-motion or camera-to-world coordinate transformations. Shah (2009) also revealed that the 3D-based obstacle detection could deal with all the problems that the 2D-based faces.

Nevertheless, motion-based obstacle detection itself has several weaknesses. Sun et al. [24] enumerated three factors that can affect the computation of motion information. First, significant pixel displacement between successive images by fast movement of the camera can cause errors to optical flow calculation. Second, lack of textures in the images can lead to unreliable motion information. Third, shocks and vibration of the camera can also influence the accuracy of motion information. Apart from these factors, the motion-based methods are computationally expensive for calculating coordinate transformations between different views, as multiple images are taken at different times and from different places. Moreover, motion-based obstacle detection cannot be used to detect static obstacles if the camera is stationary.

### 3.2.3 Depth-camera-based Obstacle Detection

Depth cameras such as stereo vision cameras and structured light cameras can recover the depth information in images even when the camera is stationary. Many depth-sensor-based obstacle detection methods are based on ground plane estimation. Burschkal et al. [20] presented a real-time obstacle avoidance system for mobile vehicles with a stereo camera. There are three main steps in such navigation systems, and Burschkal et al. revealed that first the ground surface is removed from the image after the ground plane has been observed. Then, the remaining disparities are divided into connected components in image and disparity space (see Figure 3.8). At last, the resulting connected components are shown with a proposed path on the ground surface. Cucchiara et al. [35] also proposed a stereo vision based obstacle detection for autonomous guided vehicles. Ground plane is estimated by examining the vertical coordinate of the disparity map, in which the disparity of the ground increases linearly. Then, the ground surface is removed from the disparity map to leave only obstacles. Finally, the time to collision with the closest obstacle on the trajectory of the vehicle is calculated.

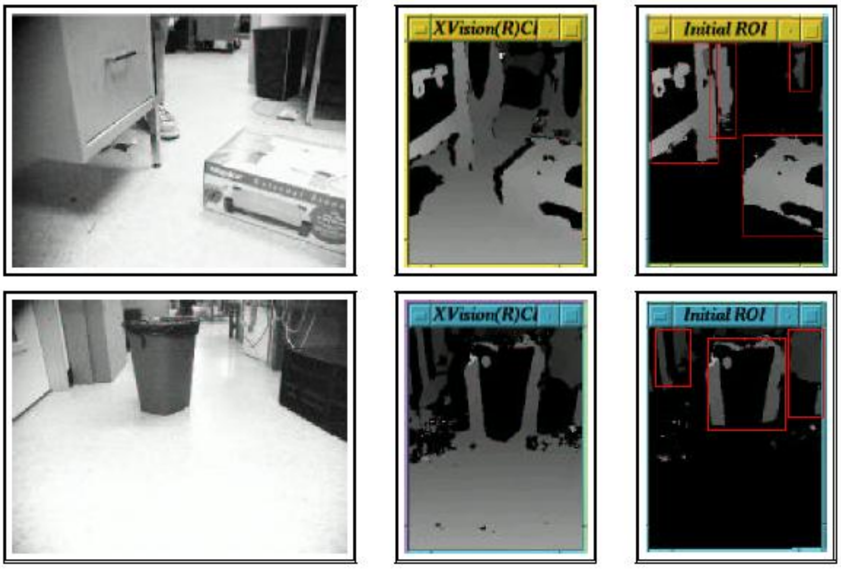


Figure 3.8: **Burschkal et al.'s stereo vision based obstacle detection.** (Figure reproduced from [20].)

Compared with motion-based methods, depth-camera-based obstacle detection has the advantage of depth information being derived without prior knowledge of the scene, and it is more accurate and less sensitive to the environmental changes. However, this depth information alone cannot provide motion information of the camera or other dynamic objects.

### 3.2.4 Multi-approach-based Obstacle Detection

In order to avoid problems with one single method, current vision-based obstacle detection approaches often combine multiple algorithms. Low and Wyeth [29] suggested using range information and other visual cues such as colour, texture and edge recognition to obtain better results for obstacle detection. Fazli et al. [36] presented an advanced method using stereo vision with a shadow removal technique for obstacle detection and revealed that their proposed method is effective and has higher performance. Some previous works [37-39] combined both stereo vision and optical flow to detect moving obstacles from a moving platform. Foggia et al. [37] presented a real-time system that combined both stereo vision and optical flow to separate the camera's ego-motion and the motion of the objects. Ess et al.'s [39] system was able to detect and track moving objects in a highly dynamic scene. The system estimated the trajectories of each object, and predicted their future position.

## 3.3 Ground Plane Detection

One of the most important geometric features is the ground plane, and many ground plane detection approaches are proposed for a variety of computer vision applications. Most of these approaches are based on image pairs from monocular image sequences or stereo vision. Planar homography describes the relationship between two planes in paired images, and Agarwal et al. [40] revealed a survey that reviewed and compared several planar homography estimation techniques.

Ohnishi and Imiya [41] developed an algorithm for dominant plane detection by optical flow. Among the sequential images, the algorithm combined a pair of corresponding points on the dominant plane with affine transformations. In addition, dense planar flow is obtained by calculating the affine coefficient. Then, the planar flow vectors are used for ground plane detection.

Although ground plane can be identified by optical flow with a monocular camera, many problems such as feature matching can be solved much more easily using depth values from depth camera systems.

Ground plane estimation normally uses plane fitting and v-disparity methods. The plane fitting technique is used to find the best surface that covers depth points. There are several plane fitting algorithms, for example, RANSAC and the least squares fitting method. Recently, the v-disparity method has been widely used for ground plane estimation, and several ground plane estimation methods are reviewed in the following sections.

### 3.3.1 Least Squares Fitting

Least squares fitting [42] is a process to find a curve or a surface that best fits a given set of data points by minimizing the sum of the squared residuals. A residual represents the difference between an observed data point and the fitted curve or surface. The sum of the squared residuals is used instead of the residual absolute values, so that the residuals are continuously differentiable. Figure 3.9 illustrates some results of fitting a set of data points.

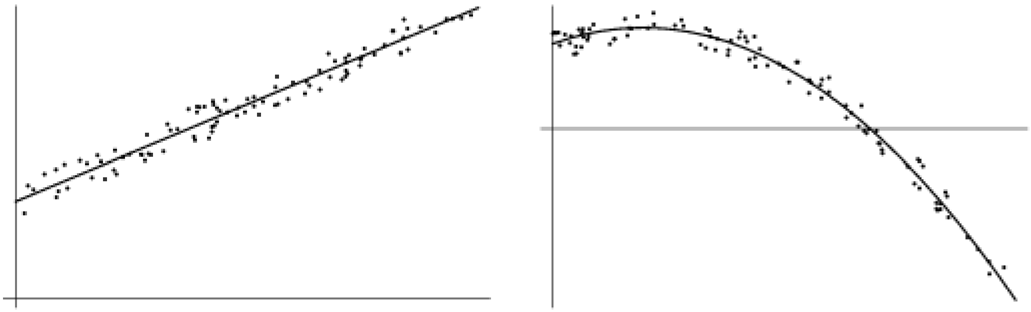


Figure 3.9: **The result of fitting a set of data points.** (a) Fitting with a linear function. (b) Fitting with a quadratic function. (Figure reproduced from [42]).

The linear least squares fitting algorithm is the simplest and most commonly used form of linear regression. It provides the results of finding the best fitting straight line through a given set of data points. The standard linear least squares measures residuals as the vertical distance from the observed data to the fitted value, as shown in Figure 3.10 (a). The total least squares measures residuals is the shortest distance from the observed data to the fitted value, as shown in Figure 3.10 (b). The total least squares may provide a better model fit than linear least squares, and has comparable computational requirements.

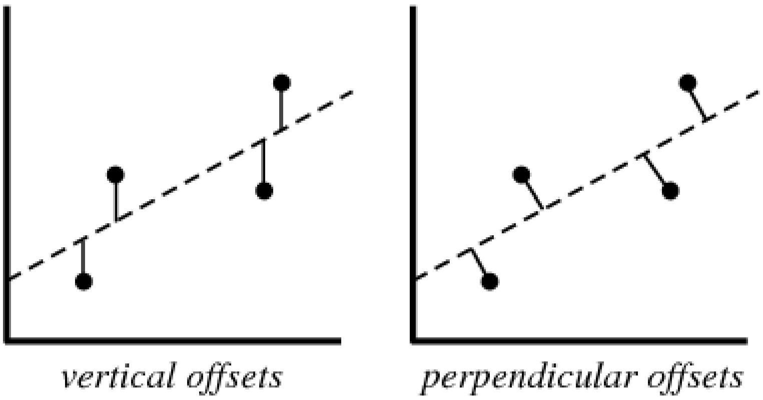


Figure 3.10: **Least squares fitting using vertical residuals and perpendicular offsets (residuals).** (a) linear least squares fitting (b) total least squares fitting. (Figure reproduced from [42]).

The least squares fitting algorithm provides a simple and efficient model of linear regression to find the best fitting linear line. However, among the given data points, some points far away from the others would significantly influence the resulting model, as shown in Figure 3.11 (a). These isolated points should not be taken into account in order to generate a more accurate model. Otherwise, a more suitable plane fitting algorithm such as RANSAC could be implemented.

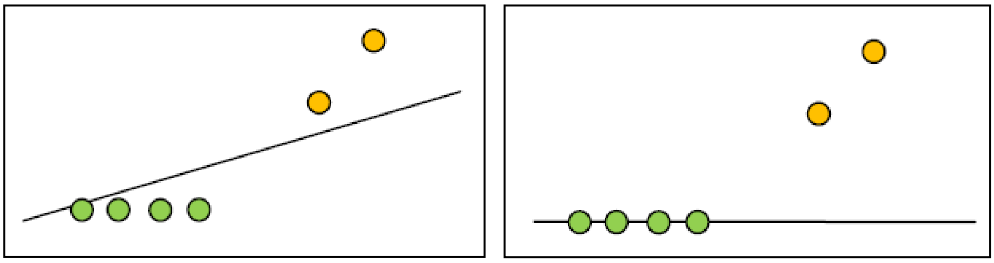


Figure 3.11: **Line fitting.** (a) fitted line with least squares fitting. (b) fitted line with RANSAC.

### 3.3.2 RANSAC

The RANSAC (Random Sampling Consensus) [43] is an iterative method that fits a model to a set of data points. In contrast with least squares fitting, RANSAC is able to take a data set with many outliers and still generate a model with a good fit to the inliers, as shown in Figure 3.11 (b).

For model generating, a random subset of the original data is iteratively chosen as hypothetical inliers. There are several procedures to test the hypothesis in RANSAC. First, a model is selected for the hypothetical inliers. Then, the rest of the data are checked against the chosen model to find fitted points as hypothetical inliers. The higher the number of fitted points which can be found, the better the estimated model used. Since the initial selected model is estimated from initial hypothetical inliers, after more hypothetical inliers have been classified, there is a re-estimation of all hypothetical inliers required to optimise the model. In the end, the re-estimated model can be evaluated by making an assessment of relative hypothetical inliers' error.

The RANSAC based plane estimation algorithm can randomly produce 3D points to determine the relevant plane. Such a procedure is explained in section 4.2.2. Konolige et al. [44] also used the RANSAC to determine the ground plane for obstacle detection in outdoor environments.

Their results are illustrated in Figure 3.12.

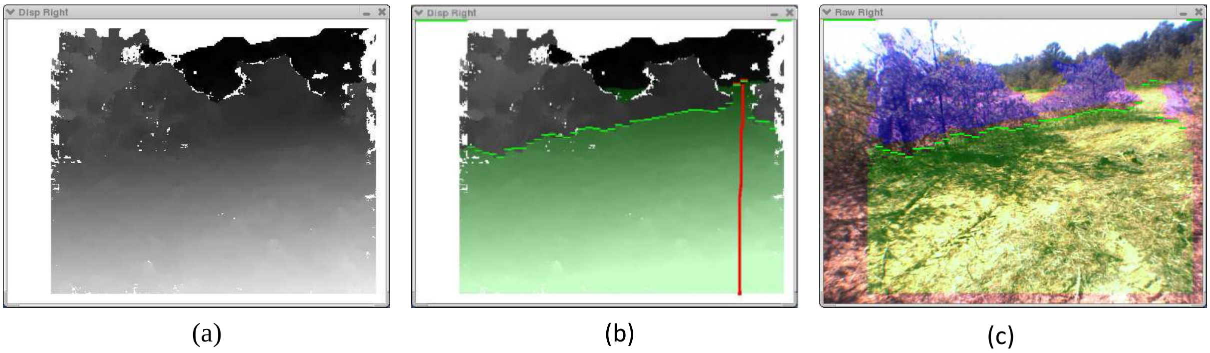


Figure 3.12: **Konolige et al.'s Ground Plane Detection Using RANSAC**. (a) Disparity Image (b) Extracted Ground Plane (c) Ground Plane Overlayed on Original Image. (Figure reproduced from [44].)

Compared with linear least squares and V-disparity algorithm, RANSAC requires more computational resources. There is no computational time limit for RANSAC to determine a best fitted model [45].

### 3.3.3 V-disparity

Labayrade [46] introduced the v-disparity algorithm to extract a 3D plane by using a 2D linear approach. The disparity map is transformed into the v-disparity image. In the v-disparity image, the abscissa axis (w) represents the disparity value; the ordinates axis (v) represents the disparity image row number; the intensity value of each pixel is the number of pixels with same disparity value in the corresponding row of the disparity map. The procedure is illustrated in Figure 3.13. Labayrade shows that planes in the world coordinate system can be represented by line segments in the v-disparity image. Figure 3.14 shows three types of planes in the world coordinate system. Vertical planes are projected as vertical line segments in the v-disparity image, while horizontal and oblique planes are projected as sloping line segments. In addition, the cameras pitch angle can be obtained from these line segments.

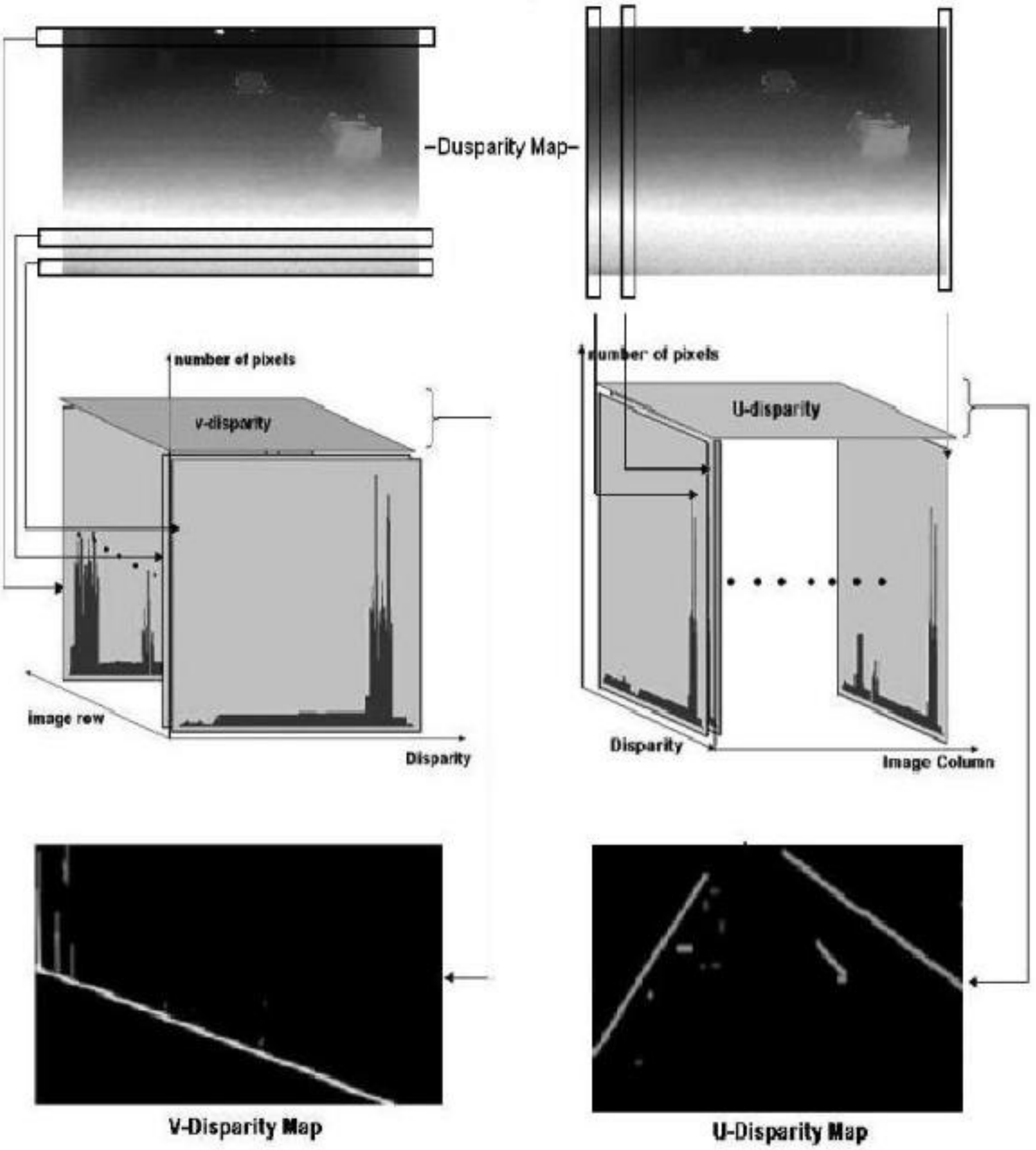


Figure 3.13: **The construction of U and V disparity maps**. (Figure reproduced from [47]).

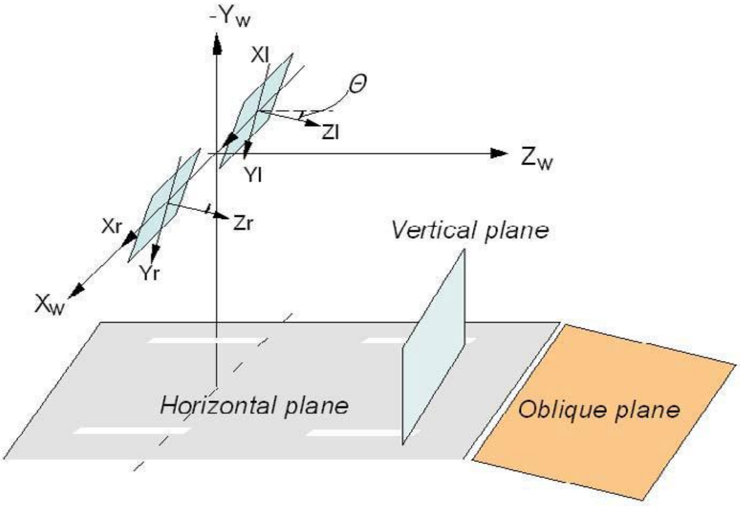


Figure 3.14: **Three basic planes in the world coordinate system**. (Figure reproduced from [48]).

Hu et al. [48] extended Labayrade's work and proposed an "U-V-disparity" algorithm to optimise the recognition of the road side structures and obstacles side surfaces. The u-disparity image is computed in the same way as the v-disparity image, as illustrated in Figure 3.13. Gao et al. [47] revealed that v-disparity can be used to divide the ground plane and obstacles on the ground, while u-disparity improves the accuracy of detection and may provide information on the shapes of obstacles.

Sappa et al. [49] presented a comparative study between a v-disparity based and a least square fitting based road surface approximation from stereo vision data. Their study shows the v-disparity algorithm has a faster processing speed, but it can be significantly influenced by noise. In other words, least square fitting is slower but more robust against noise.

## 3.4 Indoor Feature Detection

Some features are important and helpful for scene analysis, such as the major discontinuities in the images caused by lighting and object boundaries. Feature extraction is useful when detecting indoor features, such as stairs, drop-offs, doorways and hallways. The methods used to detect indoor features are reviewed in the following sections.

### 3.4.1 Doorway and Hallway Detection

Doorway, hallway and window detection are achieved by finding their respective geometric features. A typical doorway is represented by two vertical lines connected by a horizontal line at their top. Edge detection and Hough Transform are used to find straight lines in the image [50]. Once the frame of a doorway is recognized, an open or ajar door is detected using depth maps [51].

Hallway detection focuses on extracting the corridor edge lines along the floor or ceiling. Numerous approaches for corridor edge detection are described in the literature. The Sobel edge detector is one approach used by Vassallo et al. [52]; it is robust against noisy images but has a relatively slow operation speed. The Hilditch thinning algorithm is used for corridor edge lines extraction by Zou et al. [53], then corrected edge lines are obtained by the Hough Transform. Shi et al. [54] presented a method using the Fast line finder (FLF) algorithm to extract vertical lines in the image. The vertical lines produced by two sides of a wall are used as verification evidence for corridor line detection. Lho et al. [55] proposed a corridor lines extraction algorithm that used the patterns of slopes between two points, K-means, and least-square method.

### 3.4.2 Drop-off and Stairs Detection

There has been minimal prior research into detecting drop-offs. Rankin et al. [56] implemented two drop-off detection algorithms, one of which searches for gaps followed by slanting edges in stereo range data, and another looks for gaps in local terrain maps and compares the heights of their edges with nearby pixels. The two algorithms are combined to detect drop-offs. In a later work, Rankin et al. [57] presented a method to identify drop-offs at night-time by combining range data from stereo vision and thermal signatures from infrared cameras.

Murarka, et al. [19] presented a novel motion-based drop-off detection method to assist stereo methods for detecting drop-offs accurately. This method uses motion cues to locate possible drop-off edges. Drop-offs which are on the ground normally have an occluding edge. Murarka, et al. stated that for an occluding edge, new regions appear as the robot moves to the edge, while objects above the edge move faster relative to the edge than objects below the edge. The relative motion information between sequential images enables drop-off detection.

Not only drop-off detection, but also reliable stair detection is vitally important. Some research has also taken place in the stair detection area. Se and Brady [58] described a stereo-vision based algorithm which detects stairs by finding parallel lines in the image using the Hough Transform. In order to classify a stair-up or stair-down, the ground plane parameters from each side of detected stair line are then computed. Lu and Manduchi [59] combined range data from a stereo camera with brightness edge information from the image to detect and localize stairs. Stair lines are detected as they normally look like brightness edges. However, both of the above methods can fail from some viewpoints.

Pradeep et al. [60] presented a stereo-vision based piecewise planar model for step detection. They applied their algorithm to estimate scenes' planar geometry and infer the exits of steps and curbs. Pradeep et al. stated that tensor voting improved normal estimates at each data point and then a plane clustering algorithm was generated to help piecewise planar modelling. Their proposed approach using tensor voting was unlikely to be influenced by the noisy data.

## 3.5 Summary

This chapter provides an overview of obstacle and hazard detection in indoor environments. The approaches of vision-based obstacle detection in prior research are classified into three categories: knowledge-based, motion-based and depth-sensor-based. Comparisons between the three types of obstacle detection approaches are summarized in Table 3.2. Prior research also show that problems with one single approach can be solved by combining multiple approaches. Moreover, three commonly used depth-camera-based ground plane detection methods, namely least squares fitting, RANSAC and v-disparity are discussed. In conclusion, the v-disparity algorithm has a fast processing speed but can be significantly influenced by noise, while RASAC is slower but more robust against noise. Finally, the methods used to detect indoor features such as doorways, hallways, drop-offs and stair detection methods are reviewed.

| | Advantages | Disadvantages |
|---|---|---|
| Knowledge -based | • Efficient in simple environment | • Fail in complex background environment<br>• Cannot detect overhanging obstacles |
| Motion -based | • Can provide motion information of camera or other moving obstacles<br>• Can be used to calculate time-to-impact | • Influenced by fast movement, shock and vibration of the camera.<br>• Fail in low textured area<br>• Expensive computation<br>• Cannot detect static obstacles without camera motion |
| Depth-sensor -based | • Accurate and robust to environmental changes<br>• Depth information being derived without prior knowledge of the scene | • Cannot provide motion information |

Table 3.2: **A summary of the comparisons between the three types of vision-based obstacle detection methods.**

# Chapter 4
# Proposed Feature Recognition and Obstacle Detection Method

## 4.1 Introduction

Sensors play an important role in obstacle detection and feature recognition. In this project, two different methods have been developed. One is based on a passive stereo vision sensor, and the other on an active structured light sensor. The two different methods combine colour, edge, motion information and depth information to effectively analyse visual cues. Section 4.2 focuses on the development of the proposed feature recognition and obstacle detection method using a stereo camera, and Section 4.3 focuses on the development of the proposed feature recognition and obstacle detection method using a structured light camera.

## 4.2 Proposed Method with a Stereo Camera

Figure 4.1 illustrates the process of the proposed method using a stereo camera. Starting from the top left corner, a stereo vision camera system was used to obtain image pairs of the scene and produce disparity maps. Then, a disparity interpolation/smoothing/noise-reduction method was applied to fill and smooth the gaps in the disparity images. Next, the ground plane was estimated using RANSAC or v-disparity. Then, based on the ground plane equation, the 3D position of each pixel was converted from camera view coordinates into world coordinates. Afterwards, the colour image was segmented by watershed transformation or K-mean clustering to classify floor and obstacles in the regional level. Then, obstacles were detected and segmented in the depth image space. In the meantime, a novel algorithm combined a boundary detector and the Hough Transform to extract important lines in the scene. Then, drop-offs and stairs-up were detected by examining the depth value of each line. Simultaneously, the ego-motion of the camera was obtained by calculating optical flow.

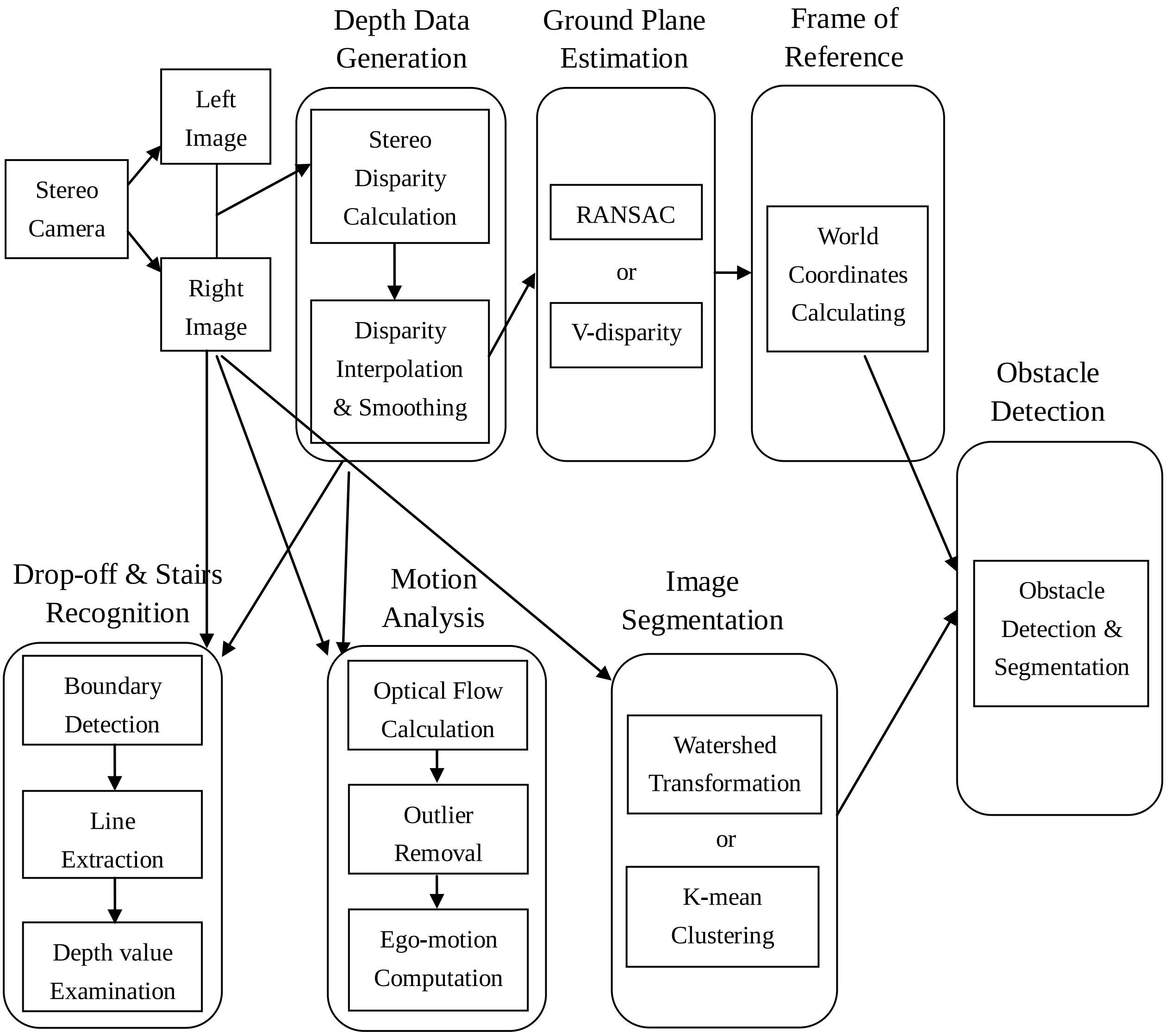


Figure 4.1: **The process of the proposed algorithm using a stereo camera.**

## 4.2.1 Depth Data Generation

As mentioned in Section 2.3.2, the Point Gray Bumblebee 2 stereo camera is pre-calibrated for lens distortion and camera misalignment, and can provide better results than manually calibrating two cameras. Therefore, the Point Gray Bumblebee 2 stereo camera was employed with the Triclops SDK to obtain the stereo image pairs and generate disparity images.

### Disparity Interpolation, Smoothing and Noise Reduction

Disparity images generated by the Point Grey Bumblebee 2 stereo camera are shown in Figure

4.2 (b) where the elements nearest the camera are red and far away are blue. However, stereo processing cannot provide depth for every pixel and often produces incorrect disparity values due to lack of correspondence.

In order to optimize the disparity map, a linear interpolation method was proposed to fill and smooth such "holes" in the disparity image. The proposed linear interpolation method filled the regions of unknown disparity values between two known disparities. This linear interpolation was performed either horizontally or vertically according to the difference between paired disparities. The horizontal paired disparities or vertical paired disparities with smaller difference value were chosen for interpolating. For example, in Figure 4.3, $P$ is a pixel to be interpolated; if $|d_A - d_B| < |d_C - d_D|$ ($d_I$ represents the disparity value of a given pixel *I*), then $d_A$ and $d_B$ are used to estimate the disparity value $d_P$, and vice versa. If an unknown disparity value is between a known value and the edge of the image, no interpolation is applied. Once the linear interpolation was performed, a filter was run through the disparity image and replaced each pixel with the median value of its 11 ×11 neighbouring pixels.

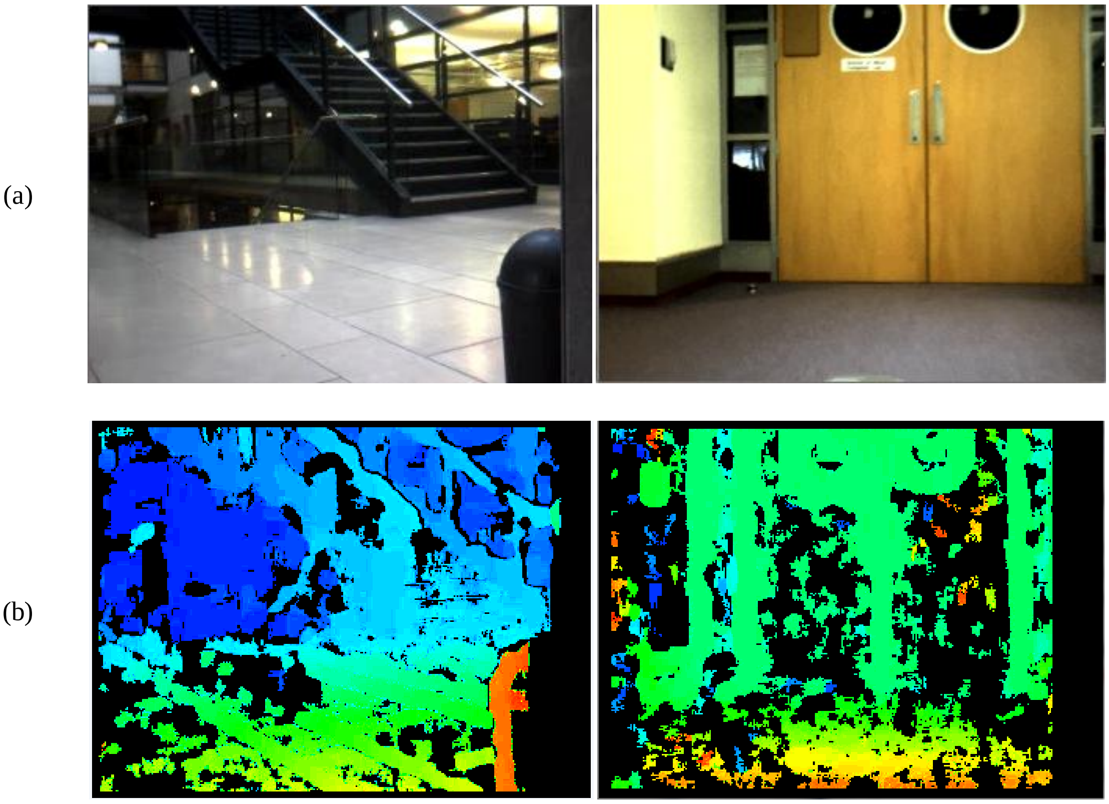


Figure 4.2: **Disparity map interpolation.** (a) Right images. (b) Disparity maps generated by the Bumblebee2 stereo

camera.

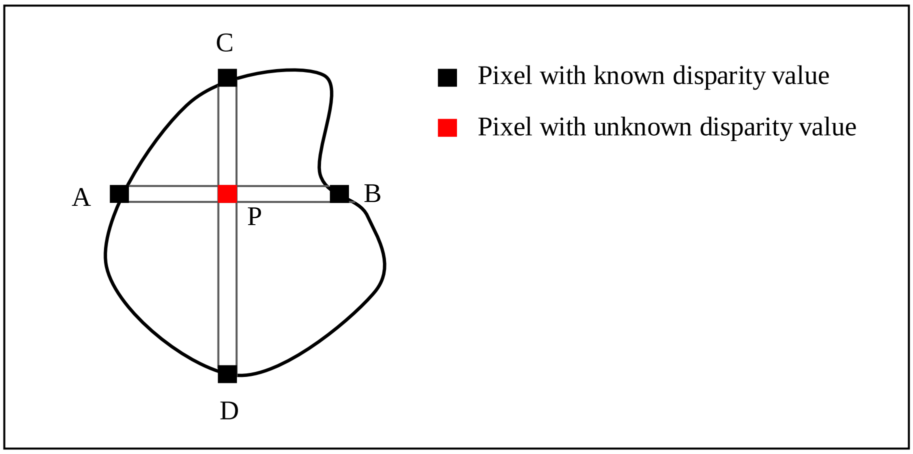

C
A
B
P
D
■ Pixel with known disparity value
🟥 Pixel with unknown disparity value

Figure 4.3: **Region interpolation.**

However, the noisy disparity values even inside an object make the disparity information unreliable. Therefore, noise reduction in the stereo disparity images is necessary before the disparity interpolation. As illustrated in Figure 4.2 (b), the noisy disparity data usually presents as small-sized regions with disparity values that are evidently different from their surrounding pixels. To reduce the noisy disparity values, the algorithm searched for regions of connected pixels which have similar disparity values, and then removed the disparity values which were inside the small-sized regions.

### Stereo vision Disparity-to-depth Calculation

To calculate the depth from the disparity, the equation below was used:

$$Z = \frac{fT}{d} \quad \text{where} \quad d = x^l - x^r \tag{4.1}$$

Then, the distances along the camera's X, Y axes were computed using the projective camera equations:

$$X = \frac{uZ}{f} \tag{4.2}$$

$$Y = \frac{vZ}{f} \tag{4.3}$$

where $(u,\ v)$ represents the pixel position in the 2D image, and $f$ is the focal length of the camera.

### Data Parsing

The stereo camera generated colour and depth values for each pixel on every frame. These values from each frame were parsed into an array of vertices where each vertex described one pixel. As shown in Figure 4.4, each vertex contains its position in the 2D image, the red, green and blue colour values, the 3D position in the camera view coordinate, and the 3D position in the world coordinate.

```
Vertex structure  {
    u, v            :   integer
    x, y, z         :   float
    R, G, B         :   integer
    worldX, worldY, worldZ   :   float
}
```

Figure 4.4: **Structure of a vertex.**

The values `u` and `v` were the pixel position in the 2D image. `x, y, z` were the three coordinates for the pixel position in the camera view coordinate. `R, G, B` represented the three colour channels (red, green, and blue). Moreover, `worldX, worldY,` and `worldZ` were three coordinates for the pixel position in the world coordinate; these values will be assigned after the world reference frame is worked out (Section 4.2.3). Using this parsed information, colour and depth information, the world coordinates of each pixel can be easily retrieved.

## 4.2.2 Ground Plane Estimation

Ground plane estimation is a prerequisite for obstacle detection. Two commonly used stereo-vision-based ground plane estimation methods RANSAC and “v-disparity” were used in this project to locate the ground plane.

**RANSAC**

The RANSAC plane fitting was used to find the ground plane in the 3D space. The procedure was the following:

1) Randomly select $N$ points within the disparity image.
2) From the selected $N$ points, randomly choose three points to fit a plane $ax + by + cz + d = 0$. Check the remaining *N-3* points on whether they fit the estimated plane equation, and calculate the number of fitting points.
3) Repeat step 2 until the number of fitting points is greater than a specified threshold. The ground plane is found.

If the number of fitting points is never greater than the threshold, the ground plane is the plane with largest number of fitting points.

However, RANSAC fails to find a suitable ground plane when there are a great number of obstacles on the same plane in the captured image. To overcome this problem, a constraint was used to limit the angle between the ground plane normal and camera view direction, so that the vertical planes such as walls could be filtered out.

**V-disparity**

The “V-disparity” was also implemented for ground plane estimation. As shown in Section 3.3.3, in a typical v-disparity image, the floor is eventually mapped onto a sloping line. The height of a v-disparity image is the same as the corresponding disparity image, and its width is the difference between maximum and minimum disparity. For each image line, points with the same disparity are accumulated. The grey level of pixels in the v-disparity image indicates the number of points with the same disparity.

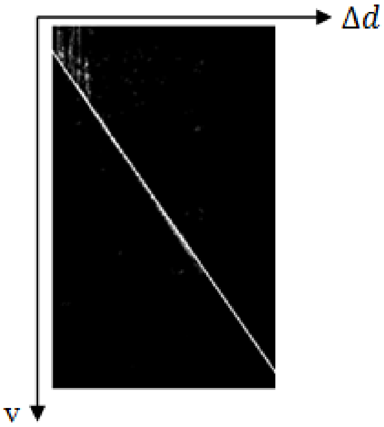


Figure 4.5: **V-disparity image.**

The Hough Transform was then used for finding all straight lines in the v-disparity image, and the line corresponding to the ground plane with the greatest accumulated v-disparity values was chosen. Then, the floor pixels were determined according to the floor line in the v-disparity map. Finally, these detected floor pixels and their disparity values were used to recover ground plane equation $ax + by + cz + d = 0$ by applying the least squares fitting or RANSAC algorithm.

### 4.2.3 Frame of Reference

To obtain a real world representation, it is necessary to know the camera's placement. As the ground plane equation in the camera view coordinates was already known ($ax + by + cz + d = 0$), the approximate height of the camera $H$ above the floor was estimated by projecting the origin of the camera (viewpoint) onto the ground plane.

Assume $P$ is an arbitrary point in the 3D space as shown in Figure 4.6, and its position in the camera view coordinates is known $(u_P, v_P, e_P)$. To obtain the actual z offset between $P$ and $C$ along z-axis, the point $P$ is projected onto the ground plane and its projection $P^{'}$ is calculated by:

$$P^{'}(u_P^{'}, v_P^{'}, e_P^{'}) = \begin{cases} u_P{}' = u_P + a * t \\ v_P{}' = v_P + b * t \\ e_P{}' = e_P + c * t \end{cases} \quad (4.4)$$

where

$$t = -\frac{(au_P + bv_P + ce_P + d)}{a^2 + b^2 + c^2} \tag{4.5}$$

The height of the point $P$ above the floor is equal to $v_P{}'$.

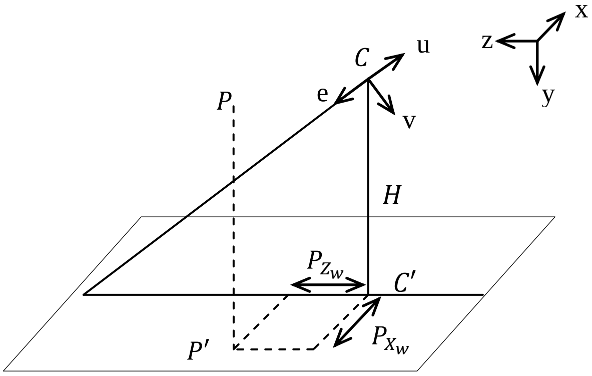


Figure 4.6: **Calculate point P in world coordinates.**

By comparing the distance offsets between the projection of the viewpoint and the projection of the point $P$ on the ground plane, the actual position of $P$ can be retrieved. As a result, the 3D position of each pixel was converted from camera view coordinates into world coordinates.

### 4.2.4 Image Segmentation

The disparity maps were optimized using the interpolation and smoothing method described in Section 4.2.1; however, these interpolated disparity values do not provide real distance measurement and may still contain many outliers. Prior research shows that image segmentation can be used to improve the accuracy of floor and obstacle classification by performing the classification at region level instead of pixel level. Image segmentation provides a method to divide an image into several segments based on their colour or texture.

There are many image segmentation methods, including clustering methods, watershed transformation and region-growing methods. A survey of different colour segmentation methods can be found in [61]. In order to find a suitable method for this system, both watershed transformation and K-mean clustering were implemented.

### Watershed Transformation

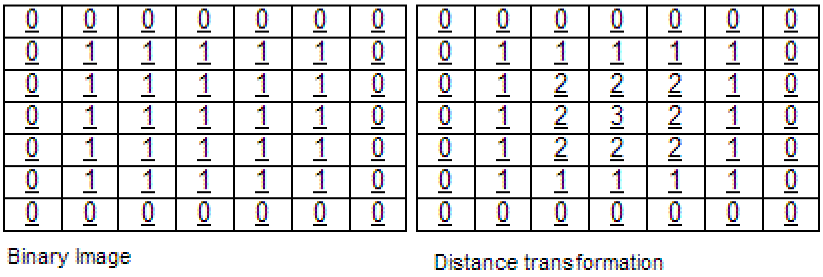
Watershed transformation is an image segmentation method using mathematical morphology. An algorithm similar to the one proposed by Cui and Zhou [62] was employed which consisted of four steps: 1) edge extraction, 2) distance transform calculation, 3) regional minimum extraction and 4) watershed segmentation.

1) Edge extraction

   The binary edge map of the image was extracted using a Canny edge detector.

2) Distance transformation calculation

   The distance transformation of the consequent edge map was calculated in a way that labelled each non-edge pixel with the distance of this pixel to the nearest edge pixel. Figure 4.7 illustrates the distance transformation of a group of binary pixels.



Binary Image

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Distance transformation

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 2 | 2 | 2 | 1 | 0 |
| 0 | 1 | 2 | 3 | 2 | 1 | 0 |
| 0 | 1 | 2 | 2 | 2 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 4.7: **Distance transformation**. (a) Binary image, where value 0 represents edge pixel and value 1 represents the non-edge pixel. (b) Distance transformation, where each value shows the distance of the pixel to the nearest edge pixel.

3) Regional minimum extraction

   A regional minimum is a pixel in the middle of a region that has the longest distance from its nearest edge pixel. A breadth-first search algorithm described in [62] was used to detect the regional minima. The algorithm, in pseudocode, works as follows:

```
Input:  $I$ - distance transform image
Output: $J$ - marker image, where regional minimums are labeled with 1 and
        non-regional minimums are labeled with 0.

$queue$ - a queue contains pixels in marker image that used to control the searching
        process.

foreach pixel $p$ in $J$ do
    $J(p) = 1$          // label all the pixels as regional minimum
end

foreach pixel $p$ in $I$ and $J$ do
    if $J(p) = 1$ then
        foreach pixel $p2$ in the 4-connected pixels of $p$ do
            if $I(p2) > I(p)$ then
                $J(p) = 0$ // label p as non-regional minimum
                add $p$ to $queue$
                exit the sub-for-loop
            end
        end

        while the number of elements in $queue > 0$ do
            $m =$ the element last added into $queue$, and delete it from $queue$
            foreach pixel $m2$ in the 4-connected pixels of $m$  do
                if $J(m2) = 1$ and $I(m) = I(m2)$ then
                    $J(m) = 0$     // label m as non-regional minimum
                    add $m$ to $queue$
                end
            end
        end
    end
end
```

Figure 4.8: **The algorithm used to derive the regional minima.**

4) Watershed segmentation

Image segmentation was fulfilled by extracting watershed lines which separate the catchment basins as illustrated in Figure 4.9. To obtain watershed lines, the algorithm uses the regional minima as starting points to successively flood different catchment basins. Watersheds are the pixels where the water from different regions would merge.

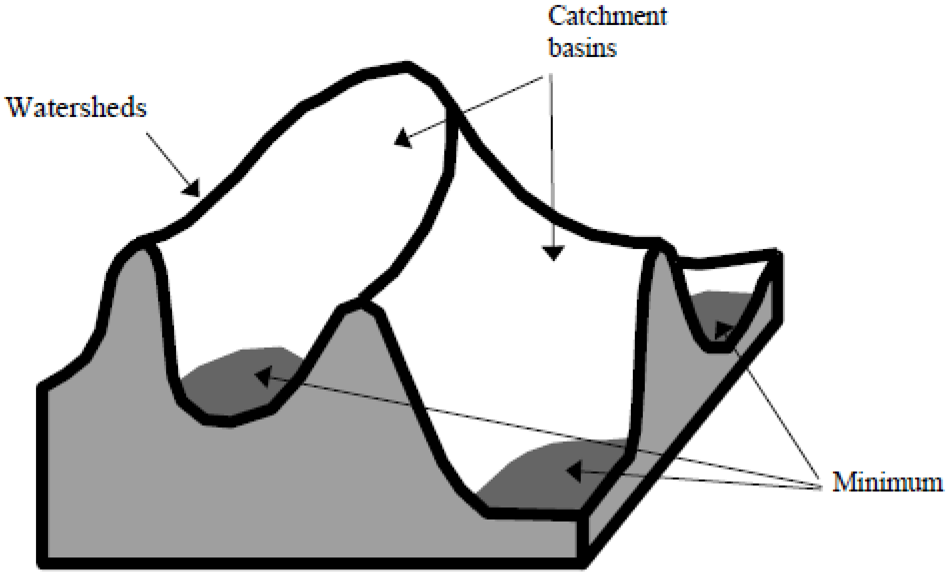


Figure 4.9: Regional minimum, catchment basins, and watershed lines. (Figure reproduced from [62]).

After an image was segmented into separated regions, each region was classified as floor or obstacles if there were sufficient number of pixels belonging to a particular group. The algorithm has also been employed by Fazl-Ersi and Tsotsos [63] for floor detection, and their results are shown in Figure 4.10.

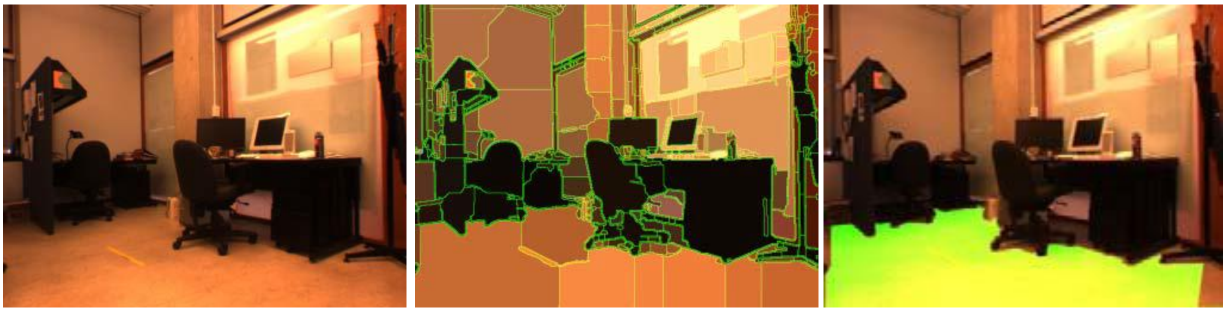


Figure 4.10: **Watershed-based image segmentation.** (a) Original image (b) Segmented image (into 483 regions) (c) Detected floor. (Figure reproduced from [63]).

### K-means clustering in HSV colour space

A K-means clustering based colour segmentation algorithm proposed by Soquet et al. [64] was also employed to segment floors and obstacles. Compared with traditional K-means clustering, the proposed algorithm is able to estimate the cluster number K and the initial position of centroids. In the following paragraphs, the three main steps of the algorithm are described.

First, colour was transformed from RGB to HSV colour space. Each dimension of HSV was quantized with 30 bins for hue, 8 bins for saturation and 8 bins for value as illustrated in Figure 4.11. As a colour with low saturation can be approximated by a grey value, the grey histogram bins were separated from the colour histogram bins. Therefore, there were 8 grey histogram bins, and 30 × 7 colour histogram bins.

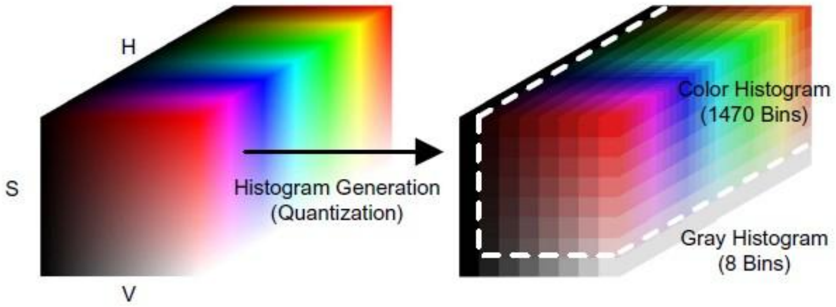


Figure 4.11**: Histogram generation process in HSV colour space**. (Figure reproduced from [64]).

Second, the number of cluster and the initial cluster centre locations were determined by following three sub-steps. a) The first centroid was chosen as the bin that had the maximum number of pixels. b) The distance was calculated between each bin and its closest centroid. A numerical boundary was considered for the distance calculation since the hue dimension had a range between 0° and 360°. The next centroid was determined as the bin that had the maximum value of distance. c) Finding the centroids was continued by repeating step b) until the maximum number of centroids was reached or the chosen centroid had a distance value smaller than a predefined threshold $Th_{max}$.

Finally, based on the results from the last step, each histogram bin was associated with its closest centroid. Then, the new position of each centroid $C_j^{(t+1)}$ was calculated according to the formulae:

$$h_j^{(t+1)} = \frac{\sum_{i=1}^{N_B} h_{i,j}^{(t)} \phi_{(j|\mathcal{B}_i)}^{(t)} B(\mathcal{B}_i)}{\sum_{i=1}^{N_B} \phi_{(j|\mathcal{B}_i)}^{(t)} B(\mathcal{B}_i)} \tag{4.6}$$

$$s_j^{(t+1)} = \frac{\sum_{i=1}^{N_B} s_i \phi_{(j|\mathcal{B}_i)}^{(t)} B(\mathcal{B}_i)}{\sum_{i=1}^{N_B} \phi_{(j|\mathcal{B}_i)}^{(t)} B(\mathcal{B}_i) + \sum_{i=1}^{N_G} \phi_{(j|G_i)}^{(t)} G(G_i)} \tag{4.7}$$

$$v_j^{(t+1)} = \frac{\sum_{i=1}^{N_B} v_i \phi_{(j|\mathcal{B}_i)}^{(t)} B(\mathcal{B}_i) + \sum_{i=1}^{N_G} v_i \phi_{(j|G_i)}^{(t)} G(G_i)}{\sum_{i=1}^{N_B} \phi_{(j|\mathcal{B}_i)}^{(t)} B(\mathcal{B}_i) + \sum_{i=1}^{N_G} \phi_{(j|G_i)}^{(t)} G(G_i)} \tag{4.8}$$

$B(\mathcal{B}_i)$ indicates the number of pixels belonging to colour histogram bin $\mathcal{B}_i$, and $G(G_i)$ indicates the number of pixels belonging to grey histogram bin $G_i$. $\phi_{(j|\mathcal{B}_i)}^{(t)}$ checks if the histogram bin $\mathcal{B}_i$ is associated with the centroid $C_j^{(t)} = (h_j^{(t)}, s_j^{(t)}, v_j^{(t)})$, the value of $\phi_{(j|\mathcal{B}_i)}^{(t)}$ is either 0 or 1. $h_{i,j}^{(t)}$ is the relative hue index between $h_i$ and $h_j^{(t)}$. This step is repeated until the positions of the centroids remain unchanged.

### 4.2.5 Obstacle Detection

For the obstacle detection, first, in the image and depth map space, the ground surface was removed from the image after the ground plane had been observed. The remaining data was divided into connected components according to their 3D positions in world coordinates. The objects that lie directly in front of the camera could influence the movements. Those pixels outside the range $(x_{min},\ x_{max})$ and $(y_{min},\ y_{max})$ were removed from the image. Then, the distance to each detected obstacle was calculated and compared. Finally, the closest obstacle was located.

The algorithm has been improved later in this research with the structured light camera. The improved algorithm projects the obstacles onto a top-down map to provide a better view of the surroundings. More details of the obstacle detection algorithm will be explained in Section 4.3.3.

### 4.2.6 Line Extraction

Doorways, stairs and drop-offs detection were achieved by finding their respective geometric features. Doorways detection was achieved by finding two vertical lines connected by a horizontal line at their top. Stairs detection focused on extracting a set of parallel lines, while drop-offs were detected by looking for a nearly horizontal line. Therefore, a reliable line extraction method is essential for doorways, stairs and drop-offs detection.

A novel algorithm named a standard deviation ridge straight line detector was proposed. This algorithm combines a boundary detector and the Hough Transform to extract lines in the image. This algorithm consists of by three steps. First, detect boundaries in the environment by exploiting the standard deviation ridge detector [65]. Second, perform an edge thinning and short edge removal process to reduce the thick boundary edges to one pixel wide edge elements. Finally, find straight lines in the boundary edge maps with the Hough Transform. The remaining parts of this section explain the standard deviation ridge detector and the process of edge thinning and short edge removal.

### The standard Deviation Ridge Detector

The standard deviation ridge detector was proposed by Hidayat and Green [65]. It is a real-time boundary detector that can retrieve more useful data than an edge detector, such as the Canny edge detector. As shown in Figure 4.12 (b), the Canny edge detector produces too many erroneous edges, and it is likely to fail in finding the most important edges (Figure 4.12 (e)). By comparison, the standard deviation ridge detector tends to detect important boundaries and avoid unimportant divisions (see Figure 4.12 (c, f)).

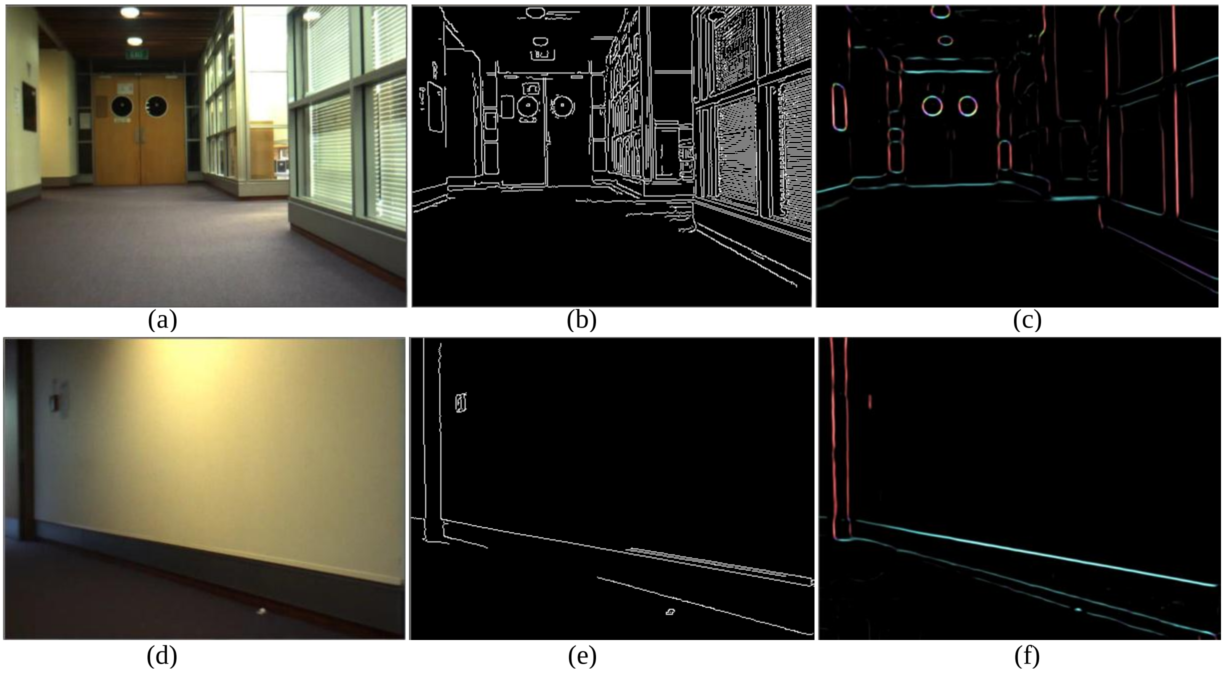


Figure 4.12: **Canny edge detector and Standard deviation ridge detector**. **(**a, d) Raw Image. (b, e) Canny edge detector. (c, f) Boundaries found by the standard deviation ridge detector.

The algorithm can be defined by the equation:

$$b_k(I) = r_k\big(g_k^s(I)\big) - ||g_k^s(I)|| \text{ where } g_k^s(I) = g_k(s_k(I)) \tag{4.9}$$

At the beginning of the algorithm, the input image $I$ is transformed into the CIELAB colour space. Then, four stages are involved with the outputs of each stage as illustrated in Figure 4.13. The four stages are:

1) Calculate the standard deviation of the image $I$, the output is $s_k(I)$.
2) Compute the gradients of the given standard deviation image, the output $g_k(s_k(I))$ is abbreviated as $g_k^s(I)$.
3) Identify the ridges from the gradient image, the output is $r_k\big(g_k^s(I)\big)$.
4) Subtract the gradient magnitude image $||g_k^s(I)||$ from the ridge image $r_k\big(g_k^s(I)\big)$, and get the final result $b_k(I)$ which shows the detected boundaries.

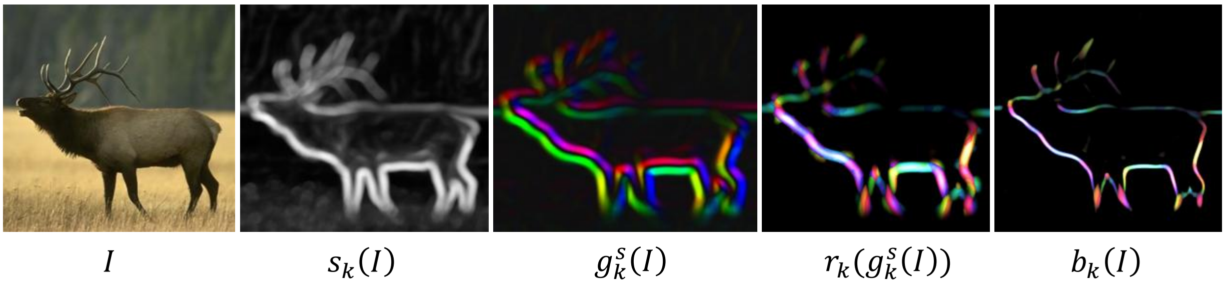


Figure 4.13: **The standard deviation ridge detector at different stages**. For the three right panels, the hue represents gradient/ridge/boundary orientation. k = 8 in all cases. (Figures reproduced from [65]).

### Edge Thinning and Short Edge Removal

The output of the standard deviation ridge detector shows the boundaries with thick edges, so that the edge thinning was carried out to reduce these thick edges to one pixel wide edge elements. First, the output image of the boundary detector was converted to a binary image. Then, an iterative process was performed to remove unwanted pixels until only the skeleton remained. To determine whether an edge pixel should be deleted, its eight nearest neighbours were considered. Figure 4.14 illustrates eight situations in which pixels are considered to be removed. Once the thick edges were turned into skeleton edge lines, the length of each edge was calculated by traversing along the edge pixels; and edges with short lengths were removed.

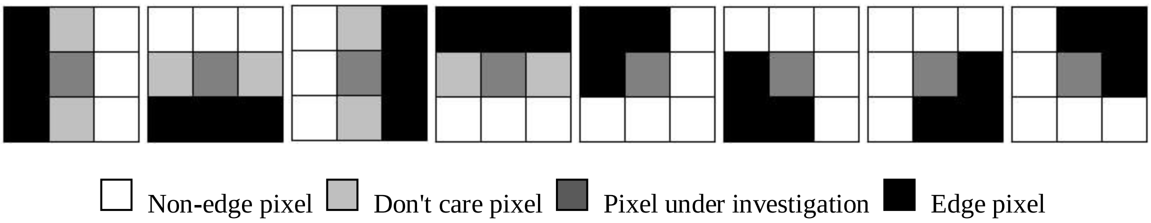


Figure 4.14: **Edge thinning.**

### 4.2.7 Drop-off Detection

A drop-off detection algorithm was proposed which combines information on edge maps and depth information. Lines were extracted from the image first with the proposed standard deviation ridge straight line detector algorithm. The drop-off detection algorithm is based on two assumptions:

1) The edges of the drop-off are straight lines.
2) The direction of the drop-off lines in the image is nearly horizontal.

Based on the second assumption, straight lines with a slope greater than a pre-defined threshold were filtered out to have only the nearly horizontal lines. Then, depth discontinuities were examined at each nearly horizontal line. If the depth value of the region above the line was much greater than the value below the line, a drop-off was found.

### 4.2.8 Stairs Up Detection

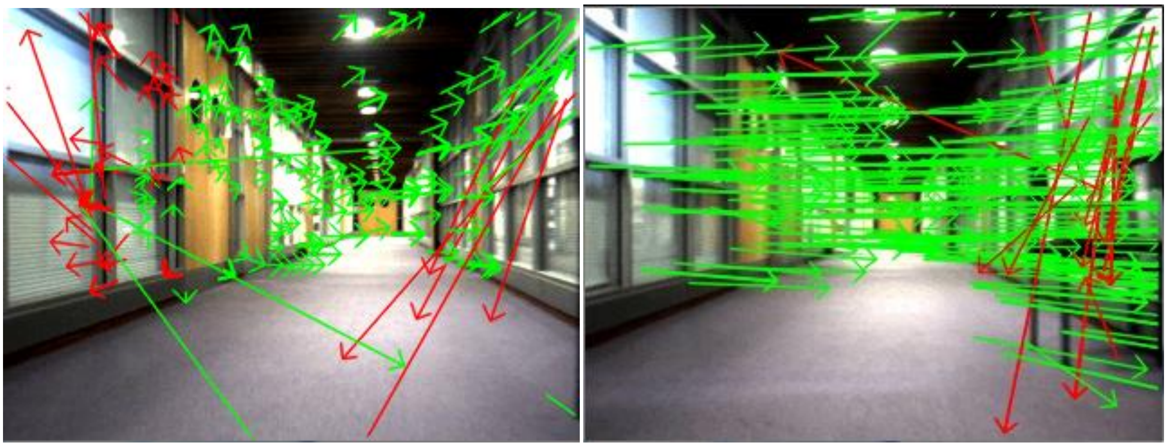
A stairs up detection algorithm was proposed based on three assumptions:

1) The edges of the stairway are straight lines.
2) The edges of the stairway are parallel to each other.
3) The direction of the stair lines in the image is nearly horizontal.

The stairs up detection algorithm searches for a set of parallel line segments; and then checks if the depth of each stair line changes gradually. To detect the stairs up, the horizontal line segments were sorted according to the y coordinate of their left endpoint in the image. Line segments with a large change in the depth around them were removed. Line segments that lie on the floor but not at the edge of the ground plane were also removed as these line segments were likely to have been produced by the textures on the ground plane. Then, line segments at the edge of the ground plane were selected as the possible starting line of a stairway. Each of the remaining line segments above the possible starting line of the stairway were examined from bottom to top based on three constraints. First, the x coordinates of the left and the right endpoints of each line segment should not exceed the two endpoints of the possible starting line too much. Second, the vertical distance between two adjacent line segments should be smaller than a threshold. Third, the depth values of the line segments from bottom to top of the image should increase gradually.

### 4.2.9 Motion Calculation

The motion information of the camera was calculated using optical flow to show the direction of camera movement along a hallway. The method selected feature points for each frame by detecting corners using Shi and Tomasi's method [66], and tracked these feature points using the pyramid Kanade-Lucas optical flow [67]. The flow vectors were calculated by locating the positions of each feature point in the previous and current frame. However, in practice a significant number of outliers may also be generated. As illustrated in Figure 4.15 the vectors with horizontal components to the left are coloured red, while the ones with horizontal components to the right are coloured green.



(a) (b)

Figure 4.15: **Flow vectors detection.** Both of the images contain outliers. (a) The optical flow vectors when the wheelchair is moving forward along a hallway. (b) The optical flow vectors when the wheelchair is turning left.

In order to prune outliers, statistical measures, such as lower (LQR), upper (UQR) and inter (IQR) quartile values were used. Outliers with uncharacteristic magnitude and direction were eliminated by the following steps:

1. Calculate the length and the direction of each flow vector.
2. Sort these vectors by length, and compute the lower, upper and inter quartiles.
3. Remove the flow vectors with lengths that are either less than $\mathrm{LQR} - 1.5 \times \mathrm{IQR}$ or greater than $\mathrm{UQR} + 1.5 \times \mathrm{IQR}$.
4. Divide the remaining flow vectors into nine regions according to their position in the current frame, as shown in Figure 4.16.
5. For the flow vectors in each region, repeat step 2 and step 3 to remove the outliers with uncharacteristic length in the region.
6. Calculate the average direction for each region, and remove the flow vectors in a region whose directions differ from the computed average.

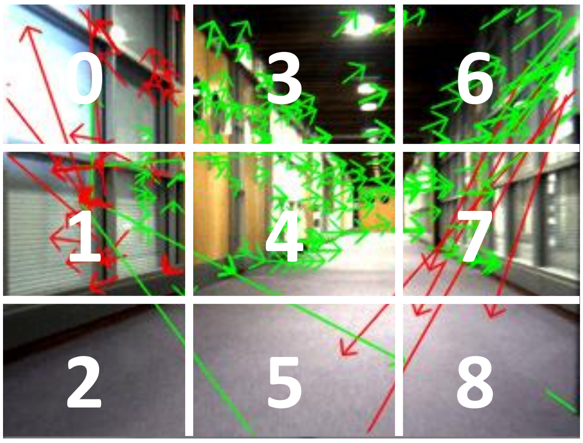


Figure 4.16: **Flow vectors in nine regions.**

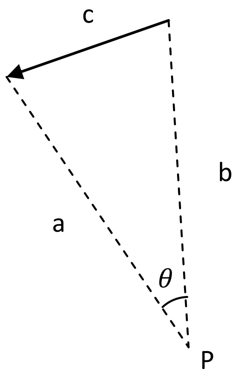
After the outlier pruning, the 2D and 3D positions of each remaining flow point in the previous frame and the current frame were retrieved by integrating with depth information. The depth changes of each flow vector were also calculated. An array of optical flow vectors shown in Figure 4.17 was used to store this information.

```
OpticalFlowVector structure {
    prev_i, prev_j          :   float
    curr_i, curr_j          :   float
    length                  :   float
    prev_x, prev_y, prev_z  :   float
    curr_x, curr_y, curr_z  :   float
    move_x, move_y, move_z  :   float   // camera movements along x, y, z
    slop                    :   int
    region                  :   int
}
```

Figure 4.17: **Structure of a flow vector.**

As the flow vectors with large depth variations are likely produce wrong motion estimation, the flow points that were far away from the camera were pruned out. Then, the average distance from the camera to each region was calculated separately. These distances were then used to compute 6 DOF pose movement of the camera.

The turning degree of the camera was approximately estimated. The difference between the flow vectors on the left side of the image (regions 0, 1, 2 in Figure 4.16) and vectors on the right side of the image (regions 6, 7, 8) were computed to determine if the camera was making left/right turns. Each flow vector on the left and right side of the image was projected onto the ground plane. Then, the turning angle was calculated as illustrated in Figure 4.18.



P - Viewpoint projection on the ground plane.
c - Flow vector projection on the ground plane.
a, b - Distances between Cam and the projection of the two optical flow points.
$\theta$ - Turning angle. $\theta = \operatorname{acos}\left(\frac{a^2+b^2-c^2}{2ab}\right)$

Figure 4.18: **Camera turning degree calculation.**

## 4.3 Proposed Method with a Structured Light Camera

In the last section, a feature recognition and obstacle detection method was proposed by using a Bumblebee stereo camera. However, a low cost structured light camera Kinect was launched which provides more reliable depth measurements under a larger variety of indoor conditions than the Bumblebee stereo camera. Therefore, the Kinect camera was later used in this research instead of the Bumblebee 2 stereo camera for depth data generation. This chapter focuses on the development of the proposed feature recognition and obstacle detection methods using a structured light camera.

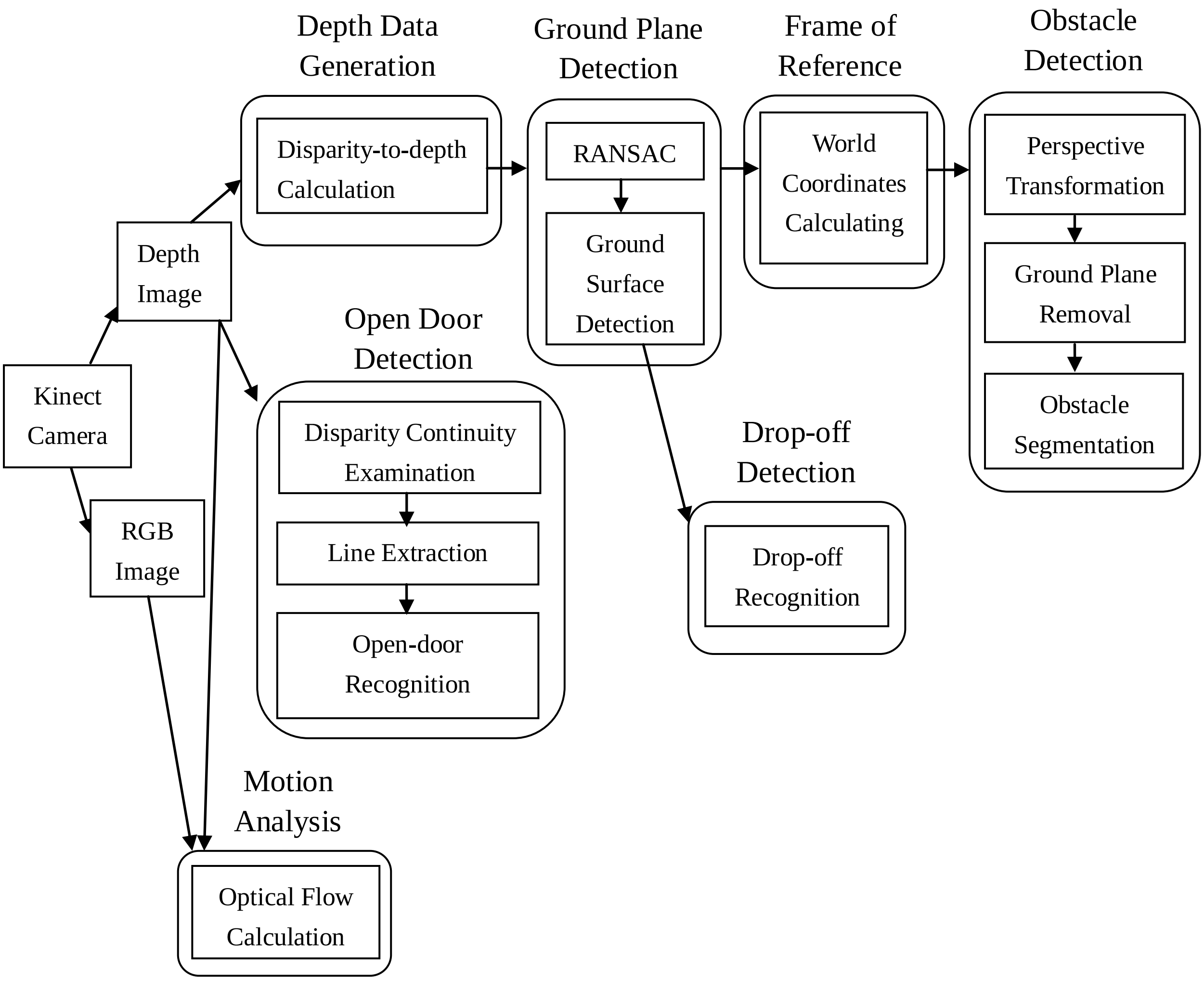


Figure 4.19: **The process of the proposed algorithm using a structured light camera.**

Figure 4.19 illustrates the process of the proposed methods. Starting from the top left corner, the Kinect camera was used to acquire both RGB and depth images of the scene. Then, disparity-to-depth calculation and image registration were performed. Next, the ground plane was estimated using RANSAC. Ground surfaces were processed using the morphology technique or a simplified version of Manduchi et al.'s obstacle detection algorithm [68]. Afterwards, based on the ground plane equation, the 3D position of each pixel was converted from camera view coordinates into world coordinates. The obstacle detection algorithm was divided into three steps: perspective transformation, ground plane removal and obstacle segmentation. Meanwhile, based on the processed ground surface, a novel drop-off detection algorithm was proposed. Simultaneously, open-doors were detected by extracting lines from the disparity continuity map. Moreover, optical flow was calculated to provide motion information of the camera.

Some methods that were implemented with the stereo camera were also used with the structured light camera. Therefore, only the methods that differ from those used with the stereo camera are explained in this section.

### 4.3.1 Depth Data Generation

The low-cost structured light sensor Kinect was employed to obtain both RGB and depth images. Three existing Kinect drivers commonly used are libfreenect [69], OpenNI [70] and Microsoft Kinect for Windows SDK [71]. In this section, the three commonly used Kinect drivers are presented and compared.

**Kinect Drivers**

Table 4.1 shows the capability of the three Kinect drivers. Both OpenNI and libfreenect are open-source drivers for Kinect, while Microsoft Kinect for Windows SDK is for only non-commercial use.

| **Capabilities** | **Libfreenect** | **OpenNI** | **Microsoft beta SDK** |
|---|---|---|---|
| RGB camera access | YES | YES | YES |
| Depth camera access | YES | YES | YES |
| Audio device access | NO | YES | YES |
| Motor controller | YES | NO | NO |
| Accelerometers access | YES | NO | NO |
| Skeleton tracking | NO | YES | YES |
| Hand tracking | NO | YES | NO |
| Gesture detection | NO | YES | NO |
| Scene analyzer | NO | YES | NO |
| Registration for depth and RGB | NO | YES | YES |

Table 4.1: **Capabilities of three Kinect drivers.**

Libfreenect, provided by an open community group OpenKinect, is a core library for accessing the Kinect sensor. The library not only supports access to RGB and depth image, but also provides access to the Kinect motors and accelerometers.

OpenNI “natural interaction” drivers are supported by PrimeSense. OpenNI provides the full capabilities of the sensor, including registration of depth and RGB images, and different depth and RGB resolution support. The depth map returned by OpenNI gives the depth value for each pixel in millimetres; hence, the disparity-to-depth calculation is not necessary. Moreover, the OpenNI also provides the interface for middleware components such as scene analyzer, skeleton and hand tracking.

The Kinect for Windows SDK can only run on a Windows 7 operating system. It provides robust skeletal tracking and advanced audio support. However, the Microsoft beta version has less capability than OpenNI. It also lacks motor controller and accelerometer access which the libfreenect provides.

As Libfreenect provides low level access to the Kinect sensor, it can be a great starting point to experiment with the Kinect camera. OpenNI provides more powerful functionality so that is used to achieve better results and faster processing speed.

### Structured Light Disparity-to-depth Calculation

The raw depth data provided by Kinect camera is an 11-bit disparity value for each pixel. To convert the raw disparity value to an approximate depth value in centimetres, the following calculations are used [72] :

$$z = 12.36 * tan\left(\frac{rawDisparity}{2842.5} + 1.1863\right) \tag{5.1}$$

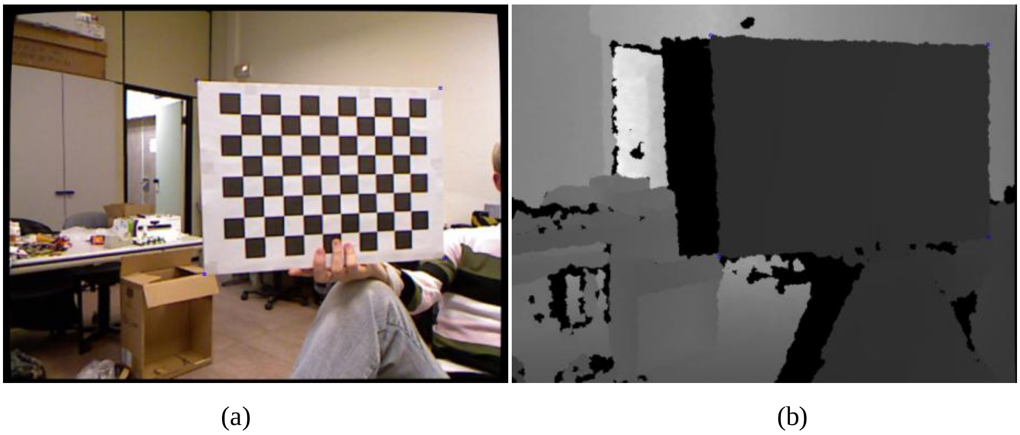
$$x = \left(i - \frac{w}{2}\right) * (z + minDistance) * ScaleFactor * \left(\frac{w}{h}\right) \tag{5.2}$$

$$y = \left(j - \frac{h}{2}\right) * (z + minDistance) * ScaleFactor \tag{5.3}$$

where $rawDisparity$ is the 11-bit disparity value corresponding to pixel $i$, $j$ in the depth image, $w$ and $h$ are the width and height of the depth image, $minDistance$ and $ScaleFactor$ are empirically determined constants, with values $minDistance = -10$ and $ScaleFactor = 0.0021$. The coordinates $(x, y, z)$ represent the pixel $(i, j)$ in 3D position with the depth camera as origin.

### Image Registration

With the Kinect camera, the depth and RGB images are captured by two different cameras from two different viewpoints, so it is necessary to map depth pixels into the corresponding RGB pixels.



(a) (b)

Figure 4.20: **Kinect camera calibration.** (a) RGB image (b) Depth image. (Figures reproduced from [73]).

Burrus [73] shows four steps involved in aligning Kinect depth pixels with colour pixels. First, a standard stereo calibration is applied to estimate the intrinsics of both depth and RGB cameras and relative transformation between the two cameras. As in the depth images, the chessboard patterns on a flat surface cannot be detected (Figure 4.20), so the corners of the chessboard are extracted by using the depth difference. Second, by using the estimated distortion coefficients, RGB and depth images are undistorted. Then, each pixel $(x_d, y_d)$ of the depth image is projected to a 3D space by using the depth camera intrinsics:

$$P_{3D}.x = (x_d - cx_d) * depth(x_d, y_d)/fx_d \tag{2.2}$$

$$P_{3D}.y = (y_d - cy_d) * depth(x_d, y_d)/fy_d \tag{2.3}$$

$$P_{3D}.z = depth(x_d, y_d) \tag{2.4}$$

where $fx_d$, $fy_d$, $cx_d$, $cy_d$ are the depth camera intrinsics. Finally, each 3D point is to be projected back on the colour image to get its colour:

$$P_{3D}^{'} = R * P_{3D} + T \tag{2.5}$$

$$P_{2D\_rgb}.x = \left(\frac{P_{3D}^{'}.x * fx_{rgb}}{P_{3D}^{'}.z}\right) + cx_{rgb} \tag{2.6}$$

$$P_{2D\_rgb}.y = \left(\frac{P_{3D}^{'}.y * fy_{rgb}}{P_{3D}^{'}.z}\right) + cy_{rgb} \tag{2.7}$$

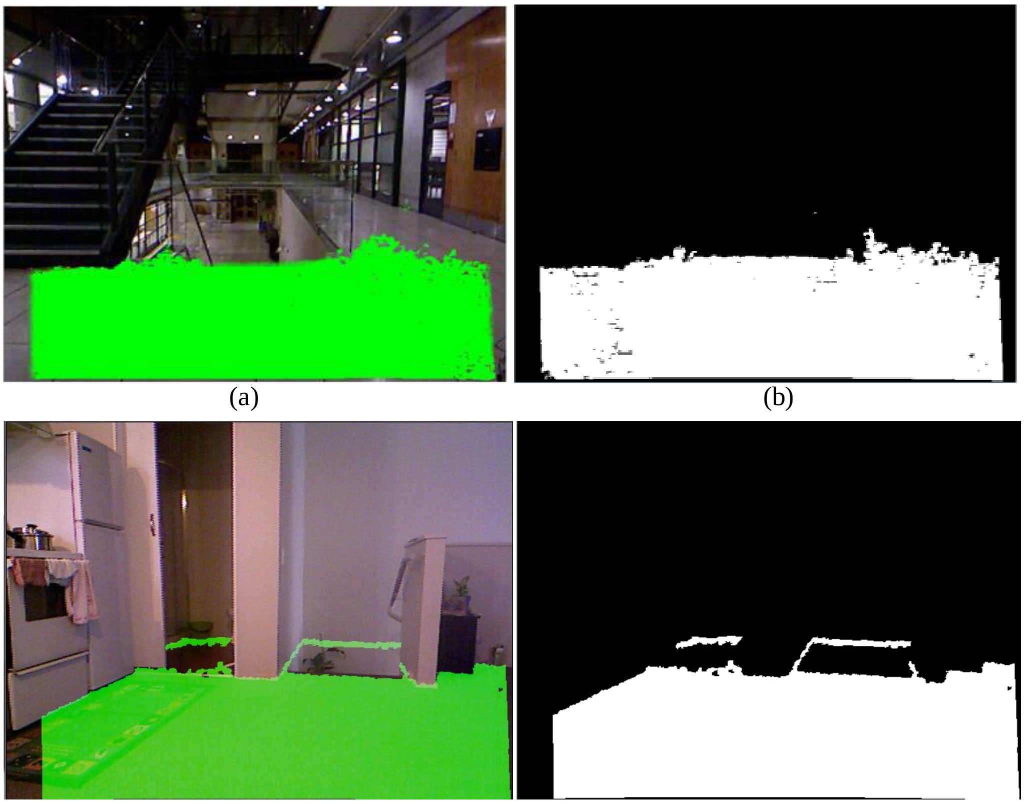
where $R$ and $T$ are the rotation and translation parameters between the two cameras that are estimated during the stereo calibration.

**Data Parsing**

Similarly to the data parse method described in Section 4.2.1, the colour and depth values of each pixel obtained from the structured light camera were stored in an array of vertices. Each vertex contains information about its position in the image plane, its RGB values and its positions in the camera and world reference frames.

### 4.3.2 Ground Plane Detection

Based on the depth information retrieved from the structured light camera, the ground plane was estimated using RANSAC (see Section 4.2.2). Then, the position of each point in the world reference frame was computed with the method in Section 4.2.3. Afterwards, pixels with very low distance from the ground plane were classified as floor pixels. However, in practice, the detected floor pixels may contain many outliers. As shown in Figure 4.21 (b), some pixels that should also be on the ground surface have not been assigned; and in Figure 4.21 (c), the pixels in the intersection between the wall and the ground plane have been incorrectly classified as ground plane.


(a) (b)

(c) (d)

Figure 4.21: **Ground plane pixels with outliers.** (a, c) RANSAC ground plane detection (b, d) ground plane mask.

To reduce the misclassification of the ground plane pixels, further processing was required. The colour image segmentation methods (see Section 4.2.4) exploited to improve the accuracy of floor and obstacle classification with the stereo camera can also be used with the structured lighted camera. However, over-segmentation and under-segmentation are two main problems of the colour image segmentation, and can impact on the results of ground pixel classification. With reliable depth information provided by the structured light camera, two different approaches were adopted to improve the accuracy of ground pixel classification. One is based on a simple morphology technique, and the other is based on a modified version of Manduchi et al.'s obstacle detection method [68]. The processed ground plane mask will be used for obstacle detection.

### Morphology

Morphology is used to close and open gaps in the binary images based on shapes. A structuring element is applied to the image. Based on the types of operation in use, the central pixel of the structuring element will become either maximum or minimum value. The two basic morphological operations are dilation and erosion. The dilation uses the maximum value for expanding the boundary of an object, and the erosion uses the minimum value to reduce an object's boundary. A morphology opening operation was processed on the ground plane mask image to reduce the area where the wall meets the ground plane. The morphology opening operation is simply an erosion operation followed by a dilation operation. In order to fill the gaps in the mask image, the morphology closing operation was performed after the opening operation. The morphology closing operation is a dilation operation followed by an erosion operation.

### Obstacle and Ground Classification in 3D Point Clouds

Another approach used to classify ground plane pixels is based on the obstacle detection method proposed by Manduchi et al. [68]. Manduchi et al.'s method detects and segments obstacles by calculating the spatial relationship between the 3D point clouds. They defined obstacles as two distinct points in 3D space as shown in Figure 4.22.

*Definition 1*. Two surface points $p_1$ and $p_2$ are called *compatible* with each other if they satisfy the following two conditions:

1. $H_{min} < |p_{2.y} - p_{1.y}| < H_{max}$ i.e., their difference in height is larger than $H_{min}$ but smaller than $H_{max}$);
2. $\frac{|p2.y - p1.y|}{||p2 - p1||} > sin\theta_{max}$ (i.e., the line joining them forms an angle with the horizontal plane larger than $\theta_{max}$);

where $H_{min}$, $H_{max}$ and $\theta_{max}$ are constant parameters.

*Definition 2*. Two points $p_1$ and $p_2$ belong to the same obstacle if:

1. They are compatible with each other, or:
2. There exists a chine of compatible point pairs linking $p_1$ and $p_2$.

Figure 4.22: **Manduchi et al.'s definition of obstacles** [68] **.**

The spatial relationships between the 3D point could data are illustrated in Figure 4.23. Manduchi et al. revealed that if a point is compatible with a surface point $p$, the point will belong to the two truncated cones $U_p$ and $L_p$ with vertex at $p$.

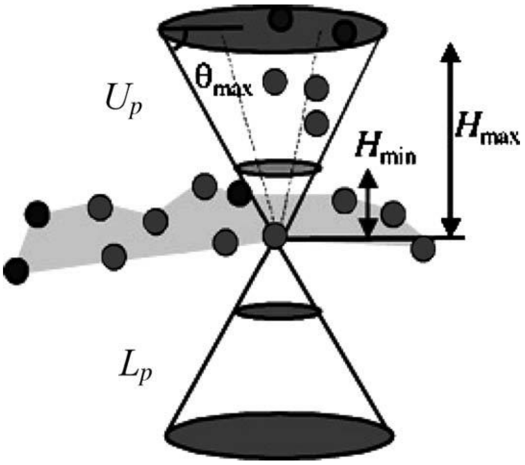


Figure 4.23: **The two truncated cones $U_p$ and $L_p$ for the determination of compatible points**. (Figure reproduced from [68]).

Obstacle points are determined by searching for pairs of compatible points. Instead of examining all image pairs, Manduchi et al. suggested that only a small subset of all point pairs was required to be examined. They calculated the projection of the two cones on the image plane, which are two approximately isosceles triangles (see Figure 4.24). The height of the triangle is $\frac{H_{max}f}{p_z}$ , where $f$ is the focal length of the camera. The base of the triangle is $\frac{2H_{max}f}{\tan\theta_{max}p_z\cos v}$, where $v = arctan\frac{p_x}{p_z}$.

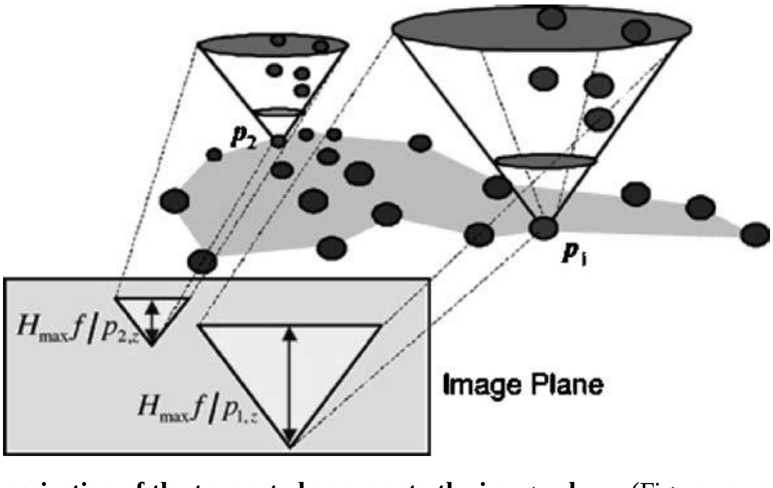


Figure 4.24: **The projection of the truncated cones onto the image plane**. (Figure reproduced from [68]).

In their method, every image point was iterated and compared with its up and down truncated triangles to determine whether it was the ground plane or the obstacle. This process requires a large amount of computations. In this project, as the ground surface has been determined, Manduchi et al.’s algorithm was modified by check the points on the edge of the ground plane mask only.

### 4.3.3 Obstacle Detection and Segmentation

The detection and segmentation of obstacles follow the extraction of the ground plane, and consist of three steps:

1) Create a top-down view of the ground plane based on inverse perspective mapping (i.e. a homography) [74, 75].
2) Remove the ground surface from the depth image to have only the obstacles.
3) Segment the connected components in the depth image, and project each connected component on to the ground plane.

**Inverse Perspective Mapping**

Inverse perspective mapping (IPM) refers to the mapping between two image planes of the same planar surface in the real world. In this project, a homography between the ground plane and the image plane was established to create a top-down view of the ground plane, so that the surroundings of the wheelchair can be observed as if seen from an overhead camera.

Figure 4.25 illustrates a frame of a top-down view of the ground plane, where the image's vertical and horizontal axes represent the distance along z-axis and x-axis in the world coordinates respectively.

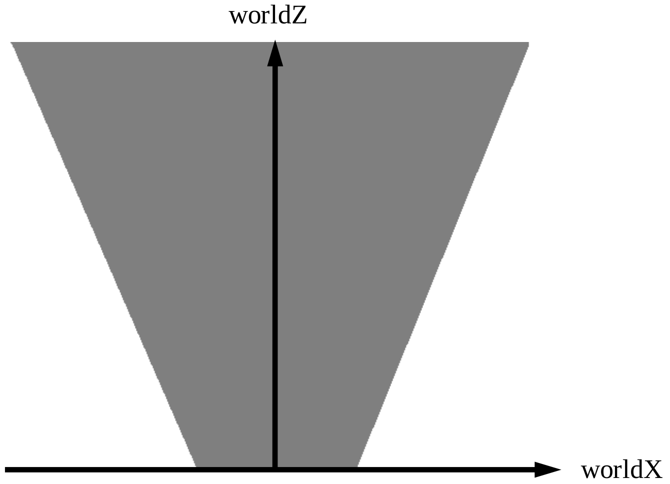


Figure 4.25: **Top-down view of the ground plane.**

In order to build the top-down view of the ground plane, an assumption is made that all the points in the image lie on the ground plane. Points that violate this assumption will appear distorted in the top-down map.

The creation of the top-down view involves five steps:

1) Select four points on the ground plane, so that the coordinates of the four points in the world coordinate system and in the top-down view (r, c) are known.
2) Compute the coordinates of the four points in the camera view coordinate system.
3) Project the four points in the world onto the image plane (u, v) based on the camera's intrinsic parameters.
4) Calculate a 3-by-3 homography matrix from the four pairs of the corresponding points in the image plane and the top-down view of the ground plane.
5) Apply a perspective transformation to the image plane by using the homography matrix to create a top-down view of the ground plane.

### Obstacle Detection and Segmentation

In the depth image space, ground surface was removed from the image according to the processed ground plane mask. The ground surface removal was aimed to isolate obstacle regions. Then, a method was employed that segmented obstacles by dividing the remaining depth data into connected components according to their depth values. The method uses a region growing algorithm to find connected components and retrieves contours of them. The region growing (also known as flood fill) algorithm starts from a specified seed point and fills a region of connected pixels. The connectivity is determined based on the depth closeness of the neighbourhood pixels. The method checks each pixel in the depth image and if a depth pixel has a non-zero value and has not been assigned to any connected component, it is chosen as a seed point for flood filling. The contour of each connected component is detected and stored in an array. Contours with small area are removed and afterwards, the pixels inside each contour are projected into the ground plane which is labelled in the top-down map.

### 4.3.4 Drop-off Detection

The proposed drop-off detection algorithm used for a stereo camera combines an edge map of the RGB image and depth information to detect drop-offs in the environment. Lines with discontinuities caused by lighting and object boundaries were extracted using the proposed standard deviation ridge straight line detector, then the depth continuity was examined at each nearly horizontal line. The method worked well for both stereo and structured light cameras. However, some scenes had many lines in them, and it was inefficient to check every line for depth continuity.

Since the structured light camera produces more reliable depth measurement than the stereo camera, the drop-off algorithm proposed in this section only uses the depth information obtained from the structured light camera. Therefore, this algorithm can be more efficient than the one used with the stereo camera as it does not iterate through all the lines detected in an RGB image.

The ground plane mask after applying the morphology operations was employed. The Canny edge detector and the Hough Transform were used to extract straight lines from the boundaries of the ground plane mask. Meanwhile, a depth discontinuity map was derived from the depth image by examining the depth changes around each pixel. If the depth value of a pixel was much greater or smaller than its neighbour pixels, the pixel was identified as a depth discontinuity. The Hough Transform was then used to find the straight lines in the depth discontinuity map. Then, every line in the two images was examined. For each nearly horizontal line, the average heights of the pixels in the region below the line were computed. If the height was low enough (i.e. on the ground), the drop-off line is detected.

### 4.3.5 Open Door Detection

The straight lines obtained from the depth discontinuity map were also used for open door detection. In a depth discontinuity map, when there are two vertical lines connected with a horizontal line at the top, such an area is normally represented as an open door. However, in practice, the camera cannot always capture the top horizontal line of the door. In addition, the top line of the door does not appear horizontal from a different angle of view. Therefore, the open door detection algorithm concentrates on examining the nearly vertical straight lines from the depth discontinuity map.

In order to detect the open doors, several approaches are used. The nearly vertical line segments were compared with each other. The line segments with very small distances between them were treated as a single line using the following algorithm:

```
foreach straight line $l$ in array $M$ do
    foreach straight line $k$ in array $M$ do
        if $l \neq k$ and $l$ is parallel with $k$ then
            calculate the distance $d$ between $l$ and $k$
            if $d$ < threshold then
                $l$ and $k$ belong to a single line
            end
        end
    end
end
```

Figure 4.26: **The algorithm used to compare straight lines.**

The resulting vertical line segments were classified into two types depending on the depth values of both their sides. The line segments that had greater depth values on their right side were classified as Type 1, and the line segments have greater depth values on their left side were classified as Type 2. Then, the vertical line segments were sorted and examined from left to right.

After the first Type 1 line was found, the next first Type 2 line was grouped with the first Type 1 line. The depth value of the left side of the first Type 1 line was compared with the depth value of the right side of the grouped Type 2 line. When the difference of the two depth values met a pre-defined constraint (i.e. 0.5 - 2 metres), the open door was found.

## 4.4 Summary

This chapter describes two proposed feature recognition and obstacle detection methods. First method is based on the RGB and depth images that captured from a stereo camera. The method combines the information on colour features, edge maps, range information and motion analysis to detect obstacles, calculate camera movements, and recognize indoor features such as ground plane, drop-offs and stairs up. One advantage of this proposed method is the introduction of a standard deviation ridge straight line detector algorithm, and the use of the algorithm for both drop-off detection and stairs up detection. Results of the proposed method are shown in Section 5.4.

Another proposed method is based on the RGB and depth images captured from a structured light camera. The method combines the information on edge maps, range information and motion analysis to detect obstacles, calculate camera movements, and recognize indoor features such as ground plane, drop-offs and open doors. The method concentrates on extracting useful information from the depth image. The main advantage of this proposed method is the use of the structured light camera to take the place of the stereo camera for obstacle detection and indoor feature recognition. Results of the proposed method are shown in Section 5.5.

The evaluations of the two proposed methods are shown in Chapter 5.

# Chapter 5
# Results

## 5.1 Introduction

This chapter presents and discusses the results from the methods proposed in Chapter 4. A comparison between a stereo camera and a structured light camera are revealed in Section 5.3. Results of proposed indoor feature recognition and obstacle detection method with a stereo camera are presented in Section 5.4, and results of proposed indoor feature recognition and obstacle detection method with a structured light camera are presented in Section 5.5. Evaluation of the two proposed methods are measured in terms of accuracy and performance as shown in Section 5.6.

## 5.2 System

The experiment system was developed on a desktop machine with the following profile:

| | |
|---|---|
| CPU | Intel Core i7-2600 at 3.4GHz |
| RAM | 8GB |
| Graphic card | AMD Radeon HD 6870 |
| OS | Windows 7 Professional (64-bit) |

Table 5.1: **The computer system used for evaluation of proposed methods**.

The specifications of the PointGray Bumblebee 2 camera and the Microsoft Kinect camera are listed below:

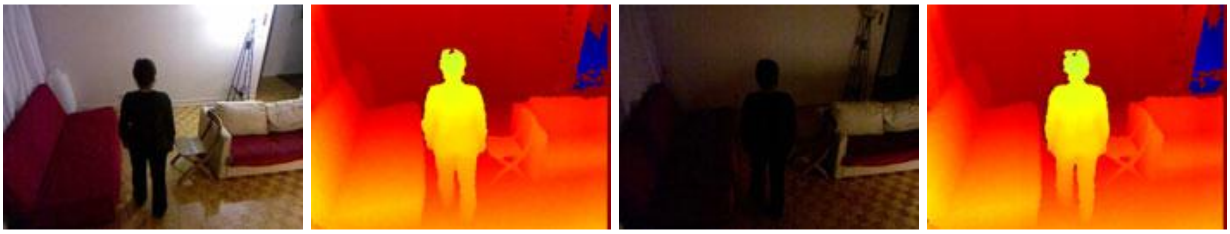
|  | PointGray Bumblebee 2 | Microsoft Kinect |
|---|---|---|
| Interface | IEEE 1394 (FireWire) | USB |
| Frame size | 1280×960 | 640×480 |
| Maximum frame rate | 20 fps | 30 fps |
| Field of View (FOV) | 66° horizontal FOV<br>43° vertical FOV | 57° horizontal FOV<br>43° vertical FOV |
| Working range | 0.5 - 4.5m | 1.2 - 3.5m |
| Market price | $NZ 2000 | $NZ 220 |

Table 5.2: **Specifications of the Bumblebee 2 camera and the Kinect camera.**

## 5.3 Stereo Camera versus a Structured Light Camera

A stereo camera PointGray Bumblebee 2 and a structured light camera Microsoft Kinect were compared. As shown in Table 5.2, the Bumblebee camera is able to produce high resolution images of 1280×960 pixels at 20 fps. It has a wide working range of 0.5 - 4.5 metres and a wide horizontal field of view. However, the Bumblebee camera is expensive and requires extensive tuning of parameters for optimal depth values with minimal noise. In addition, the camera has difficulty obtaining depth information in non-textured areas. Also, stereo is not able to work well without good ambient lighting. Furthermore, the depth calculation of the Bumblebee camera is performed on the host machine and so incurs a higher computational cost than using the Kinect.

In contrast, the Kinect camera has more advantages than the Bumblebee camera for indoor applications. As shown in Table 5.2, the Kinect camera is able to provide a reasonable resolution video of 640×480 pixels at 30 fps. The supported depth range of the Kinect camera from the specification is 1.2 – 3.5 metres. Experiments in [76] show that a useful depth working range of the Kinect camera is around 0.7 – 7 metres, although with marginal accuracy near these limits. The accuracy of Kinect depth data evaluated in [77] show that the error of distance measurement reaches 4cm at the maximum range of 5 metres. In addition, the Kinect camera has a low price of approximately $NZ200 compared to about $NZ2000 for the Point Gray Bumblebee 2 stereo camera at the time of writing. Most importantly, the Kinect camera provides more reliable depth information under a larger variety of indoor conditions than the Bumblebee stereo camera. The camera is even able to work without any ambient lighting as shown in Figure 5.1. Moreover, the Kinect camera calculates the depth value directly using its built-in processor [78]; hence, it does not require large computation.



With normal light conditions

With low light conditions

Figure 5.1: **Comparison of a video image with the corresponding depth map under different light conditions.** (Figure reproduced from [79]).

Images in Figure 5.2 are results from the Bumblebee 2 stereo camera, illustrating the difficulty obtaining depth information in non-textured areas. By comparison, the Kinect camera projects an infrared light pattern to cover the full scene and so retrieves depth information for almost every pixel (see Figure 5.3).

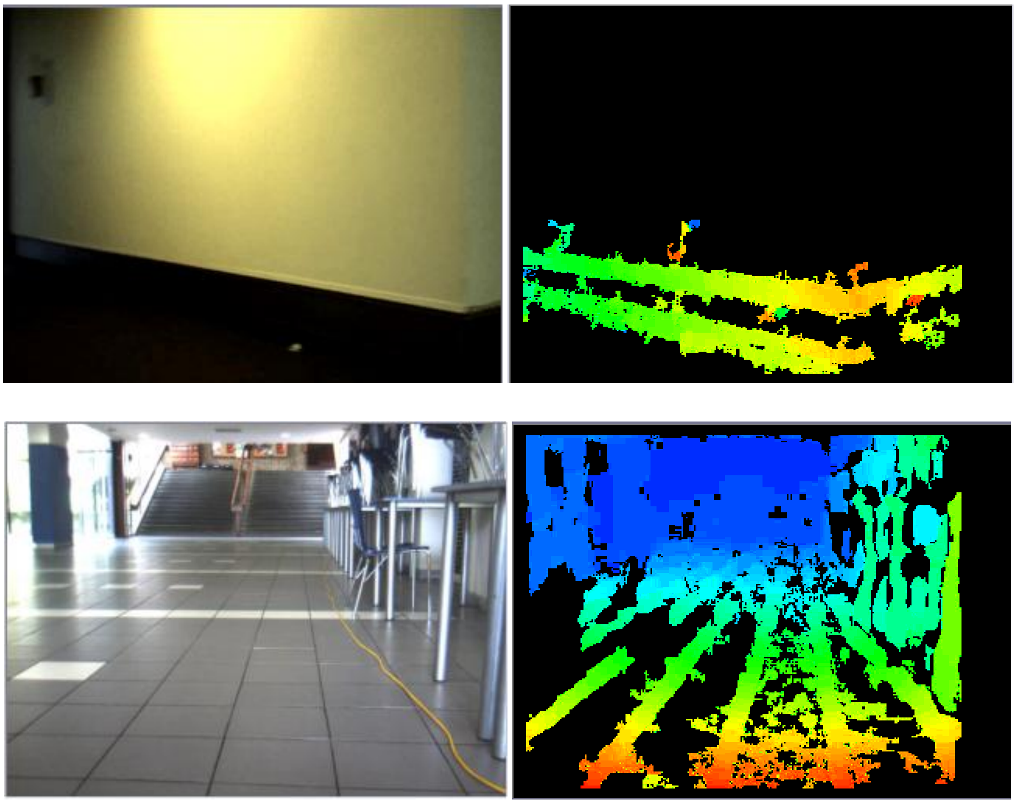


Figure 5.2: **Depth image generated by the Bumblebee2 stereo camera.**

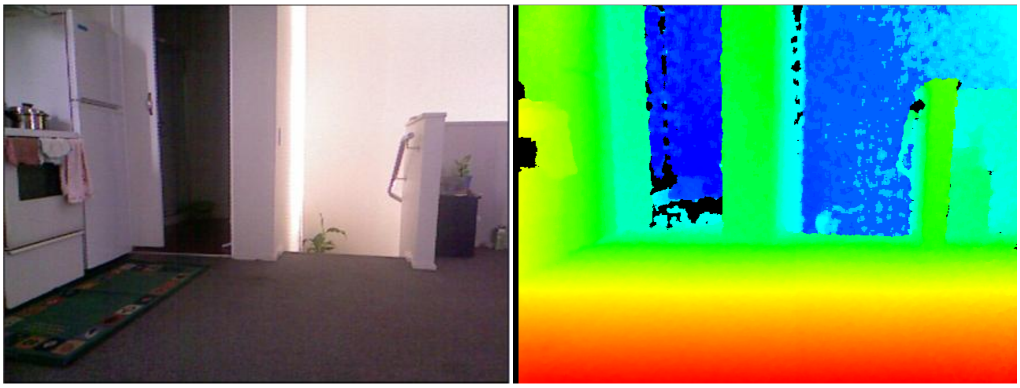


Figure 5.3: **Depth image generated by the Microsoft Kinect camera.**

As mentioned in Chapter 2, ideal sensors for smart wheelchairs should be accurate, low cost, small, lightweight, low power consumption and not influenced by environment changes. Therefore, the Kinect camera is preferable for use with smart wheelchairs in indoor environments compared with the Bumblebee camera.

## 5.4 Results of Proposed Method with a Stereo Camera

### 5.4.1 Depth Data Generation

Images in Figure 5.4 (b) are disparity maps generated by the Bumblebee camera where the elements nearer the camera are in warmer colours and the missing disparity values are black. The majority of missing disparity values are caused by lack of correspondence due to the different viewpoints observed by the left and right cameras. In addition, there is a noticeable amount of incorrect disparity values.

Therefore, the proposed disparity interpolation/smoothing/noise-reduction method was applied to remove the noisy disparity values and fill the disparity holes. Figure 5.4 (c) illustrates the result of noise reduction in the disparity images. Most of the noisy disparity data was eliminated, although some correct disparity values were also removed. Figure 5.4 (d) shows the dense disparity maps generated after linear interpolation and disparity smoothing. However, this interpolation method could also produce invalid disparity values, as the disparity holes due to objects far away from the camera are interpolated by two foreground objects.

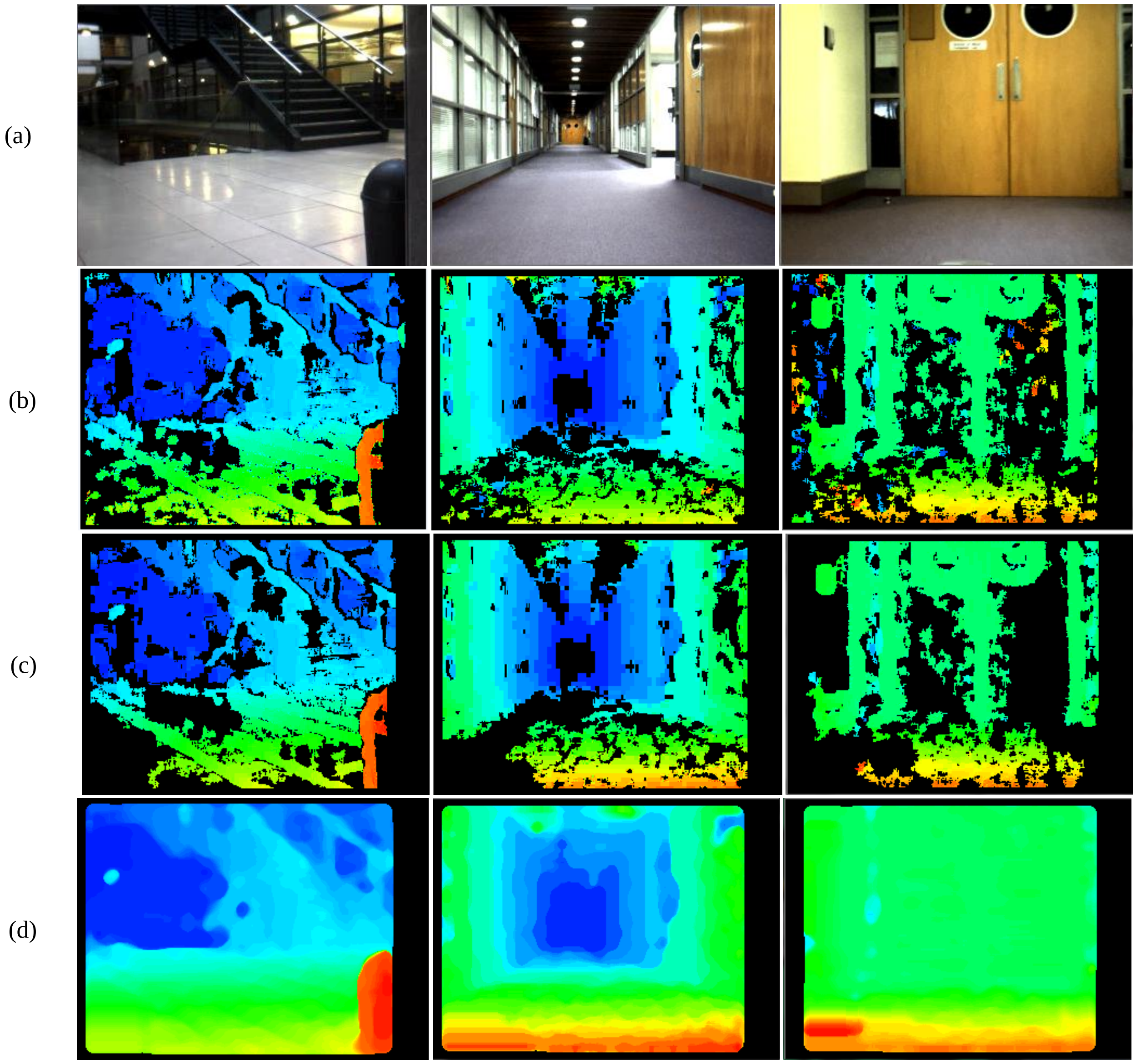


Figure 5.4: **Disparity map interpolation.** (a) Right images. (b) Disparity maps generated by the Bumblebee2 stereo camera. (c) Disparity maps after incorrect disparity values removal. (d) Disparity maps after linear interpolation and smoothing.

### 5.4.2 Ground Plane Detection

Ground plane detection is an essential preparation for obstacle detection. In order to get an accurate result for ground plane detection, the "v-disparity" and RANSAC based ground plane detection algorithms were implemented.

**V-disparity**

Figure 5.5 illustrates the result of ground plane estimation using the "v-disparity" method. The floor is eventually mapped onto a clear line segment in the v-disparity image, and the ground plane is estimated according to the floor line in the v-disparity map. The method works fine if there is enough texture and no camera's roll angle, but it is not able to detect the floor properly when a roll angle exists. Owing to limited time, the implemented algorithm simply assumed there was little or no roll angle of the camera. However, the algorithm described in [80] can be used to estimate the camera's roll angle.

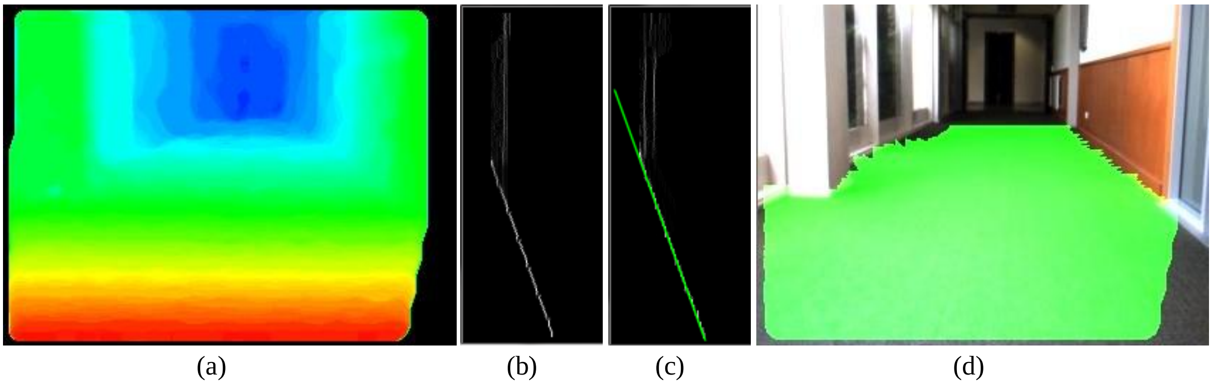


Figure 5.5: **Ground plane detection using "v-disparity"**. (a) disparity map. (b) v-disparity image. (c) line corresponding to floor. (d) marked floor.

### RANSAC

The RANSAC ground plane detection algorithm does not require the estimation of the camera's roll angle. Figure 5.6 shows some results of ground plane estimation using RANSAC. The algorithm was evaluated in different buildings with different kinds of floor material and textures, and with different lightings. The result shows that with sufficiently accurate disparity data, the floor is detected accurately. However, floor detection fails in the highly reflective floor regions. By comparison, the ground plane estimation method described by Chumerin and Hulle [81] relies heavily on the accuracy and the density of the disparity map, as their algorithm does not have a disparity interpolation step.

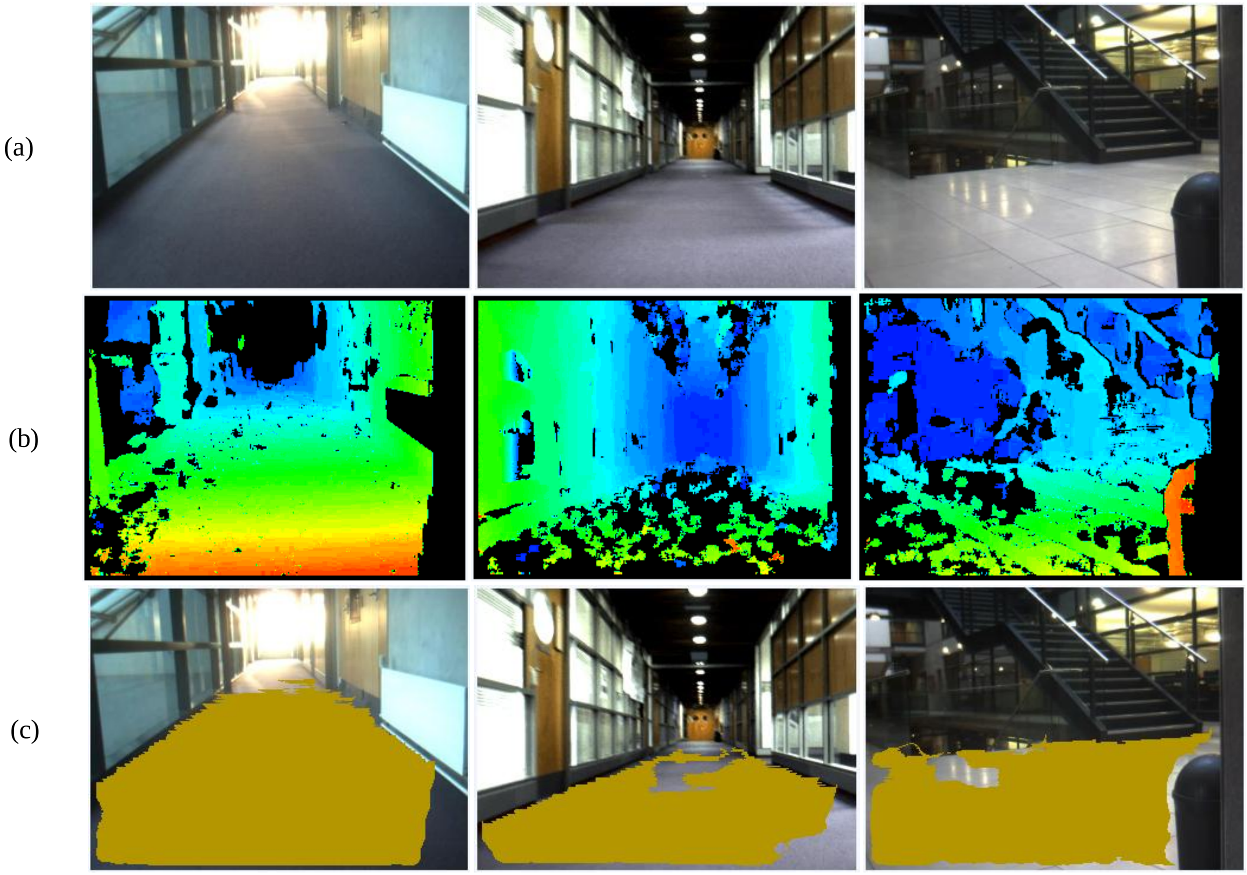


Figure 5.6: **Results of ground plane estimation.** (a) Right image. (b) Raw disparity map. (c) Detected floor area.

### 5.4.3 Image Segmentation

Two image segmentation algorithms have been implemented for floor/obstacle classification. One is k-mean clustering, and another is watershed transformation. The k-mean clustering algorithm generates poor estimates, while the watershed transformation algorithm produces more accurate results and so was used to classify the floor and obstacles in this project.

**K-means clustering in HSV colour space**

Some results of the k-means clustering algorithm are shown in Figure 5.7. After applying the k-means clustering method, the pixels of similar colours were assigned with a random colour. Unfortunately, the algorithm is only efficient if the colour of the object to be segmented differs significantly from the colour of other nearby objects (see Figure 5.7 (b)). Figure 5.7 (d) shows that even when the floor has the same colour; it is divided into several segments by this algorithm. A problem with colour is that it is too easily influenced by illumination. Therefore, the k-means clustering colour segmentation algorithm is unreliable for real world segmentation.

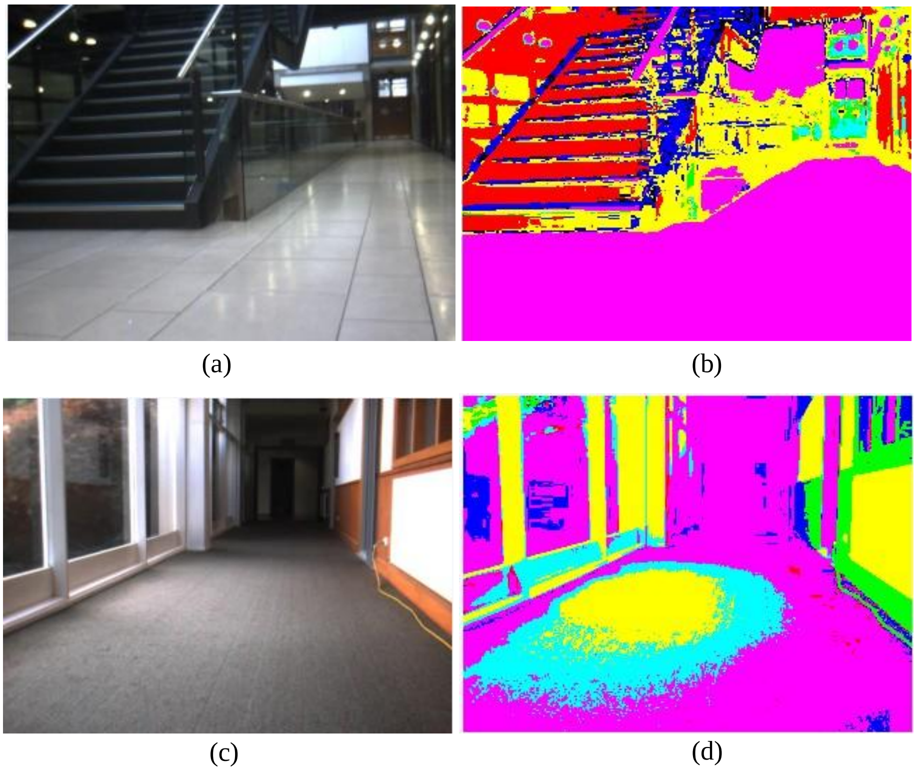


Figure 5.7: **K-means clustering in HSV colour space**. (a, c) RGB images (b, d) Images segmented by k-mean clustering.

### Watershed Transformation

The watershed transformation method described in Section 4.2.4 was carried out, based on the Canny edge detection (see Figure 5.8). The result of watershed segmentation is shown in Figure 5.9 (c), where images are divided into small regions. Each region was classified as ground plane or obstacles depending on how many pixels in the region belong to the ground. If a region contains more than 20 percent ground surface pixels, the region is classified as ground plane. In Figure 5.9 (c), regions that belong to the ground plane are highlighted.

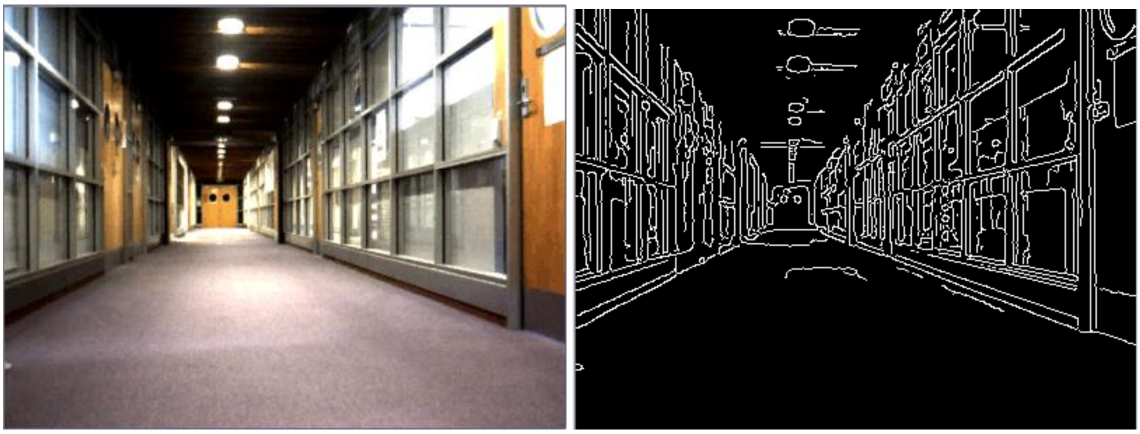


Figure 5.8: **Canny edge detection.**

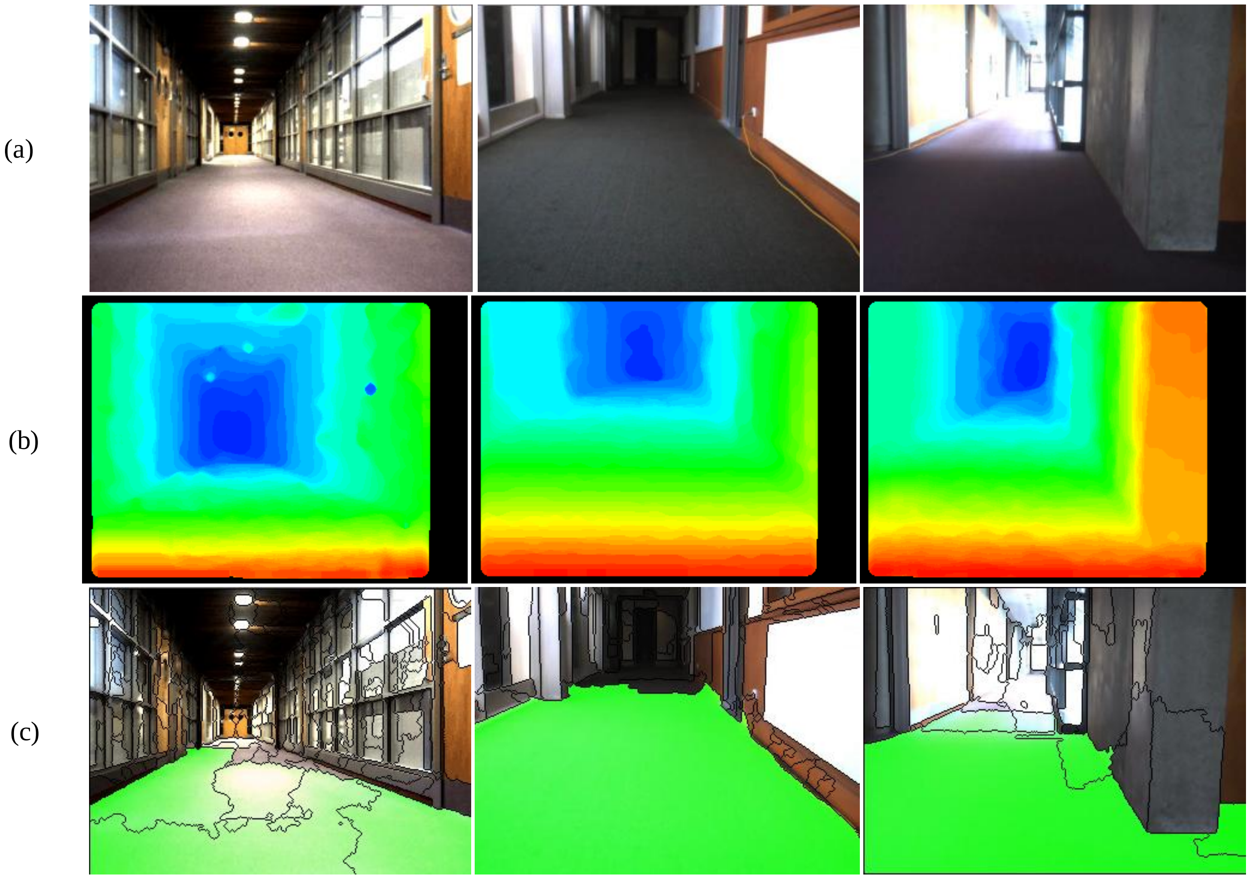
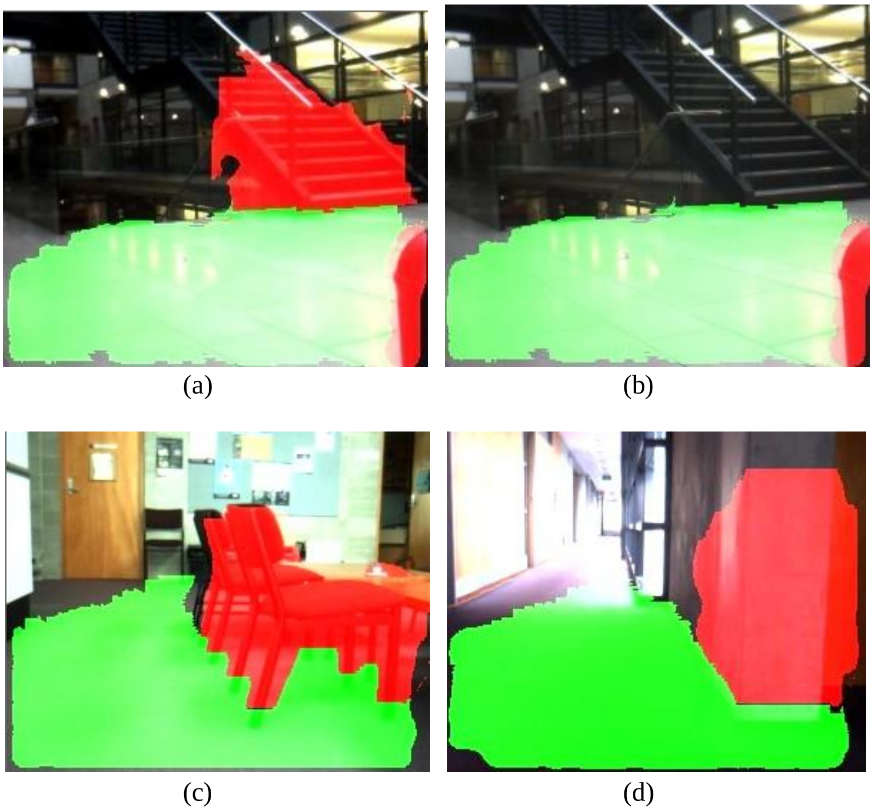


Figure 5.9: **Result of watershed-based ground plane segmentation.** (a) Right image. (b) Disparity map after interpolation and smoothing. (c) watershed-based ground plane segmentation. The three images from bottom left to bottom right are segmented into 303, 96 and 66 regions respectively.

### 5.4.4 Obstacle Detection

Some results of obstacle detection are illustrated in Figure 5.10, where the floor is highlighted with green colour while the detected obstacles are highlighted with red colour. In Figure 5.10 (a), the stairs and the rubbish bin are two obstacles that lie directly in front of the camera. After calculating their distance, the closet obstacle was located (see Figure 5.10 (b) ).



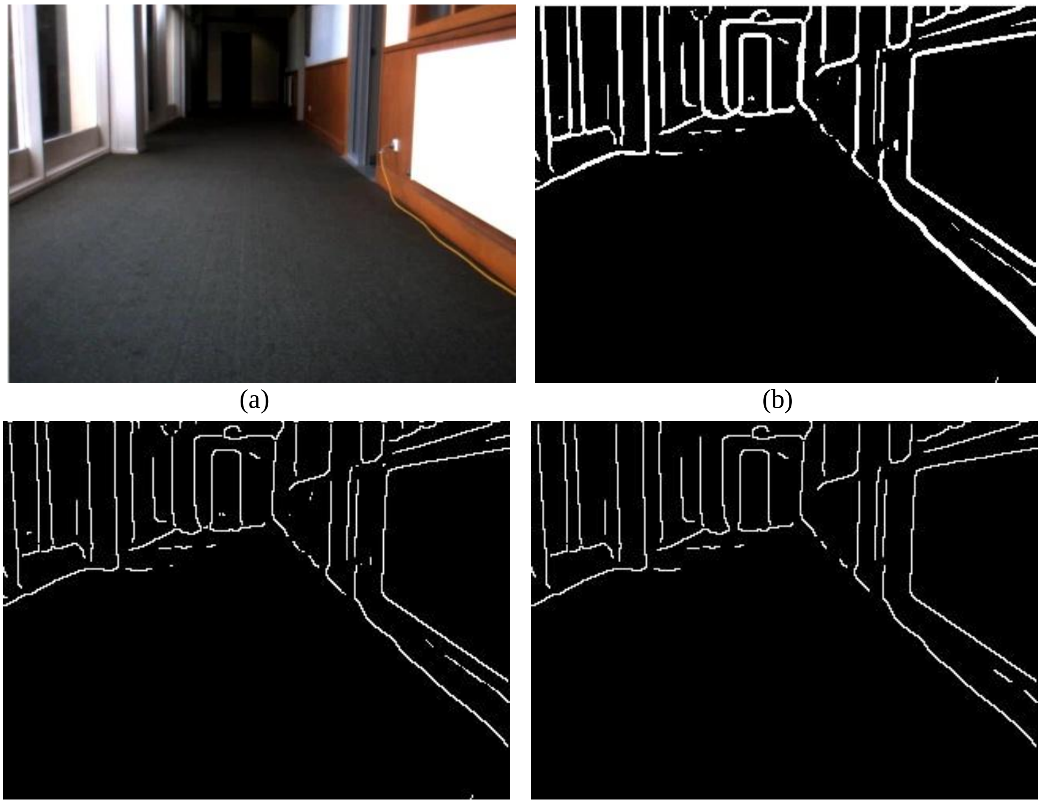
(a) (b)

(c) (d)

Figure 5.10: **Detecting nearest obstacle detection**. (a) locating stairs and rubbish bin (b) locating rubbish bin (c) locating table and chairs (d) locating corridor walls.

### 5.4.5 Line Extraction

The proposed standard deviation ridge straight line detector algorithm was implemented. A boundary image after applying the standard deviation ridge detector was converted into a binary edge map. As shown in Figure 5.11 (a), the map contains thick edges. Figure 5.11 (d) illustrates the resulting skeleton edge lines after edge thinning and short edge removal. Finally, the Hough Transform was carried out to find straight lines. Figure 5.12 illustrates the results of the proposed standard deviation ridge straight line detector algorithm, where lines in the environment are extracted.



(a) (b)

(c) (d)

Figure 5.11: **Edge thinning and short edge removal.** (a) raw image (b) binary edge map (c) edge map after edge thinning (d) edge map after short edge removal.

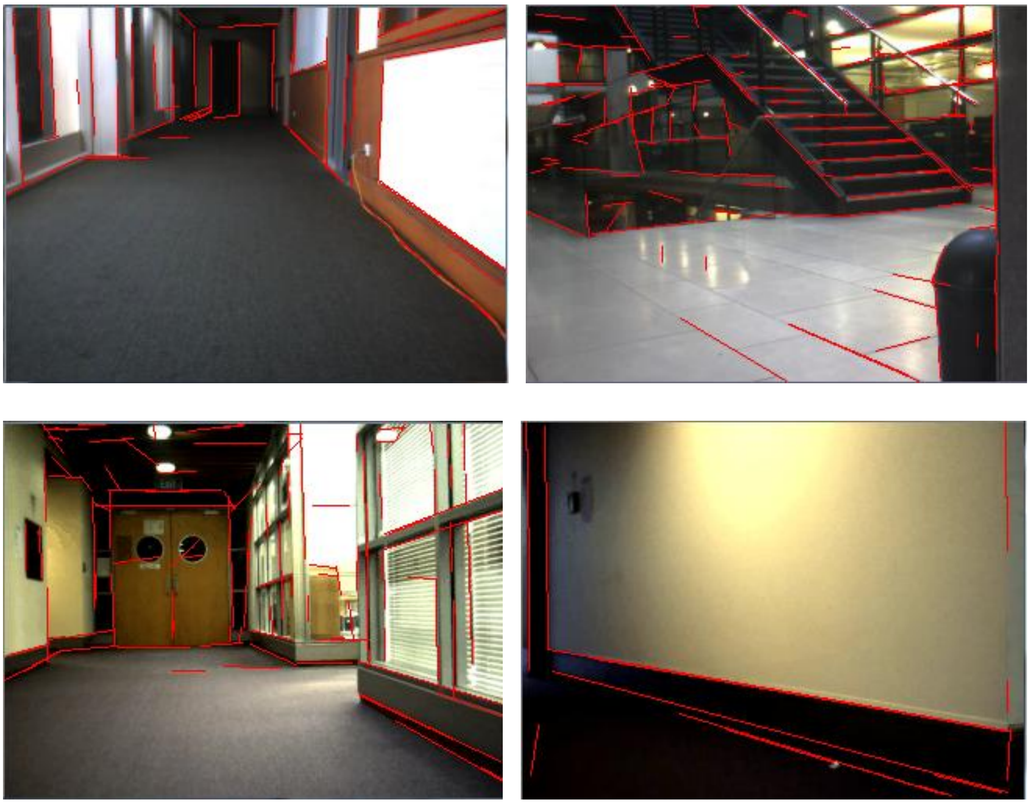


Figure 5.12: **Result of the proposed standard deviation ridge straight line detector algorithm.**

A comparison was made between the proposed standard deviation ridge straight line detector algorithm and a straight line detection algorithm using the Canny edge detector and the Hough transform. As shown in Figure 5.13, straight lines obtained by using the Canny edge detector and the Hough transform contain many erroneous lines, as too many trivial edges are produced by the Canny edge detector. The results demonstrate that the proposed standard deviation ridge straight line detector algorithm is able to extract lines more accurately than the Canny edge detector and the Hough transform.

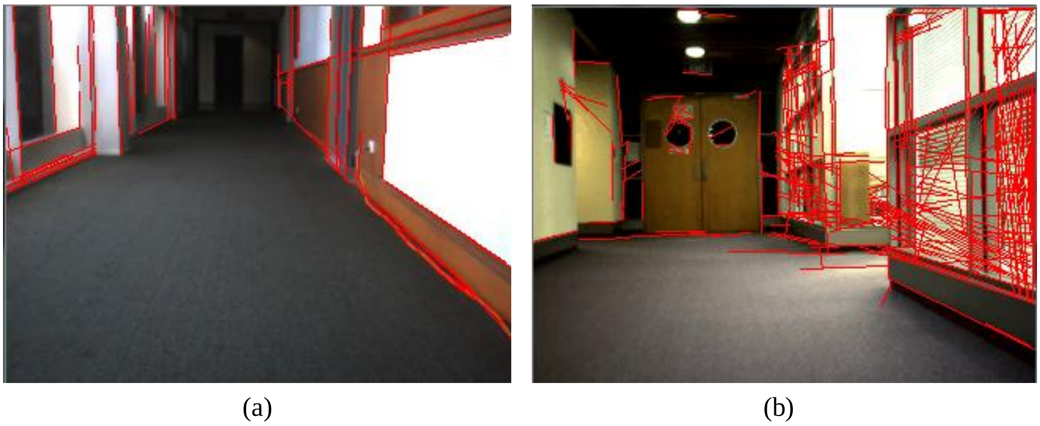
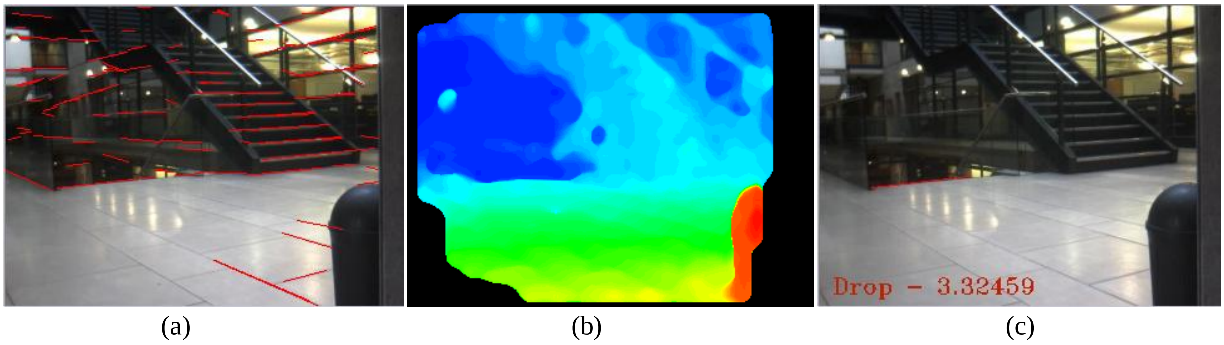

(a) (b)

Figure 5.13: **Result of the Canny edge detector and the Hough Transform.**

### 5.4.6 Drop-off Detection

The proposed drop-off detection algorithm was implemented which makes use of the detected straight lines and depth information. As illustrated in Figure 5.14, the depth discontinuities were examined at each of the nearly horizontal lines to determine the drop-off line. The distance to the detected drop-off was calculated and drawn on the bottom of the resulting image.



(a) (b) (c)

Figure 5.14: **Drop off detection.** (a) nearly horizontal lines (b) disparity map (c) detected drop-off line.

An experiment was carried out to examine the accuracy of the proposed drop-off detection algorithm using a twenty-second video sequence. The drop-off in the video sequence was always correctly detected. However, other false positive results were often found, as the algorithm identified objects such as banisters and tables other than drop-offs. To reduce the high false positive rate, a constraint was defined that the area below the drop-off line must contain sufficient ground plane pixels. As a result, the false positive rate was reduced to less than 6%. Figure 5.15 illustrates a false positive result that is likely to be caused by the incorrectly interpolated depth values.

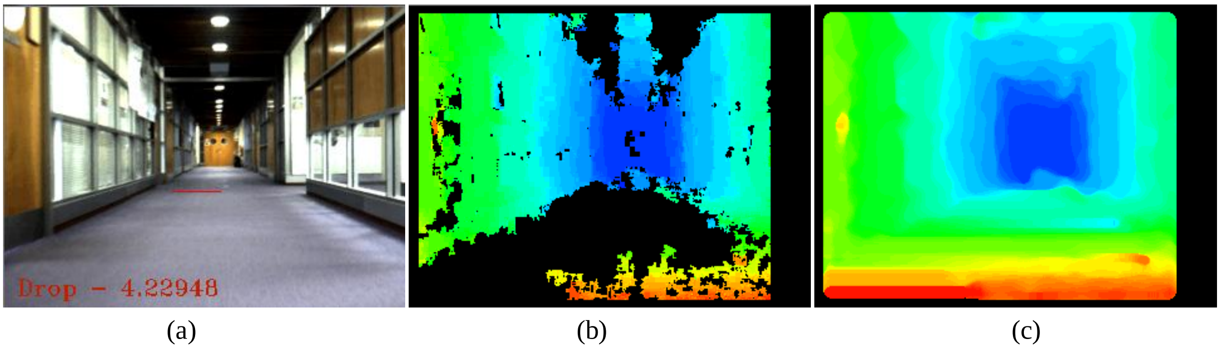
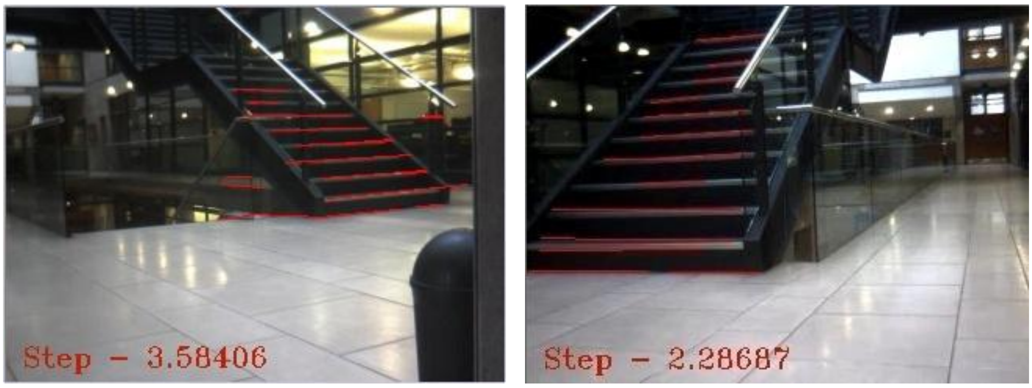


Figure 5.15: **False positive result of the proposed drop off detection.** (a) a false detected drop-off line (b) raw disparity map (c) disparity map after the noise removal, interpolation and smoothing.

### 5.4.7 Stairs Up Detection

Figure 5.16 illustrates the results of stairs up detection, where the stairs are represented as a set of parallel line segments in which depth changes gradually. However, some line segments that do not belong to the staircase are also marked as part of the stairs (see Figure 5.16 (a)). This may be caused by one of the following reasons. First, the starting lines of a stairway had been incorrectly chosen. In Figure 5.16 (a), the line segment in the intersection between the ground plane and the left side of the staircase is selected as a possible starting line of the stairs. Second, as the system compares the x coordinates of the left and the right endpoints of each line segment with the two endpoints of the starting line, as illustrated in Figure 5.16 (a), a short line segment that is above the starting line of the stairs but not a part of the stair lines is also marked as part of the stairs. The problem caused by the second reason can be avoided by comparing the endpoints of each line segment with the endpoints of the adjacent line segment.



(a) (b)

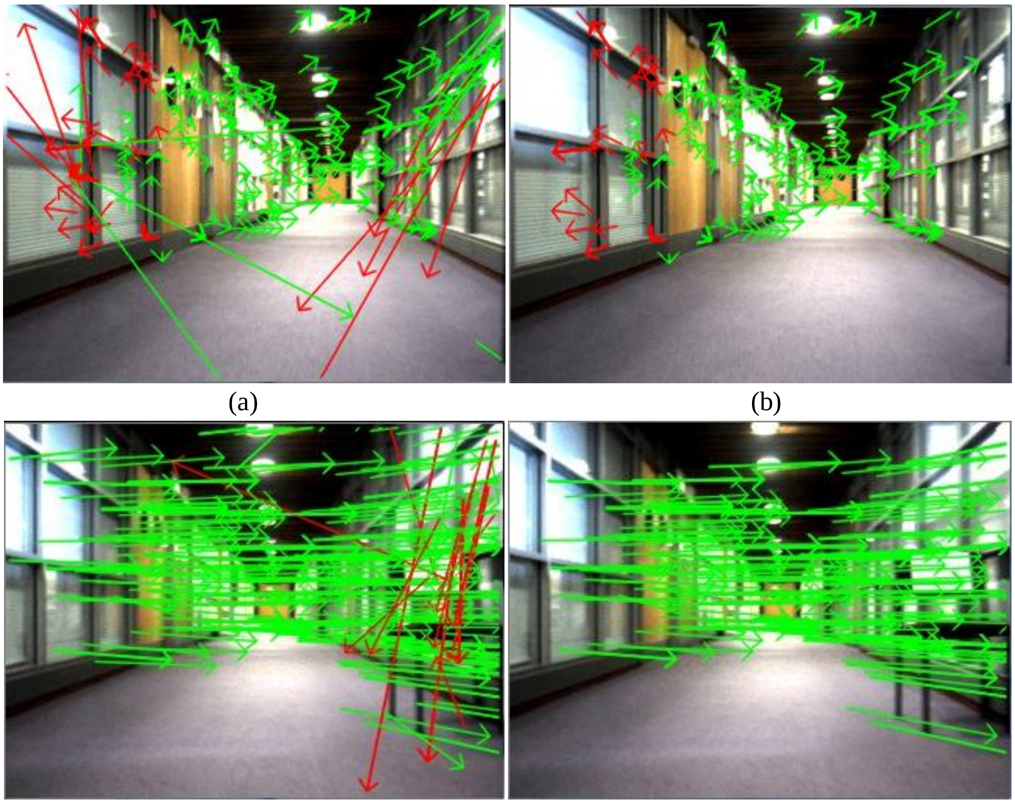
Figure 5.16: **Stairs up detection.** The red number shows the distance to the detected staircase in metres.

Moreover, the incorrect stairs up detection also causes the incorrect distance measurements. To reduce the false positive rate, a more advanced algorithm will to be designed to distinguish the stair lines from others. Moreover, to achieve a better result in stairs detection, the edge linking algorithm described by Cong et al.[82] can be used. However, compared with drop-offs, an up staircase is less harmful for wheelchair navigation, as an up staircase can be simply treated as an obstacle.

### 5.4.8 Motion Calculation

The optical flow algorithm was implemented to calculate the movement of the wheelchair. In practice, every fifth frame taken from a video sequence was used for optical flow calculations. The maximum and minimum number of feature points was also defined. If a sufficient number of feature points were detected, these points were then tracked in the next fifth frame. As feature points may be lost during the movement, if the number of tracked feature points were less than a minimum threshold, new feature points were selected for tracking.

During the tracking process in optical flow, a large number of outliers were produced. To derive meaningful motion information, the proposed algorithm identifies and prune the outliers based on the magnitude and direction of the flow vectors. The results are illustrated in Figure 5.17 (b, d), where outliers with uncharacteristic magnitude and direction are eliminated. However, the algorithm also incorrectly filtered out some of the flow vectors.

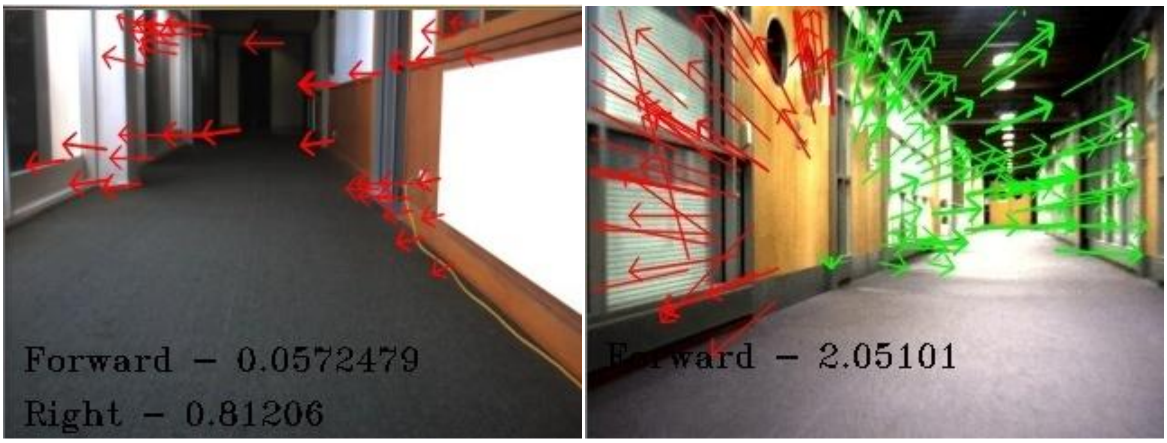

(a) (b)

(c) (d)

Figure 5.17: **Optical flow vectors.** (a, c) Raw unfiltered optical flow vectors generated by the moving camera. (b, d) Optical flow vectors after outlier removal.

Then, the optical flow vectors were used to analyse the motion information of the camera. In Figure 5.18, the camera's ego-motion is illustrated. The number in Figure 5.18 (a) shows the camera turned 0.812 degrees to the right and moved 0.057 metres forward, while Figure 5.18 (b) shows the camera moved 2.05 metres forward.

The estimated motion information was examined approximately. However, the experiment shows that the camera's moving distance measured by the algorithm is often greater than the ground truth. This is mainly caused by the flow points with incorrect disparity values.



(a) (b)

Figure 5.18: **Motion analysis.** (a) optical flow vectors when the camera was turning left. (b) optical flow vectors when the camera was moving forward.

# 5.5 Results of Proposed Method with a Structured Light Camera

## 5.5.1 Depth Generation

OpenNI was used to obtain depth and RGB images from the Kinect camera. Figure 5.19 shows the 3D point cloud data generated by the Kinect camera that contains RGB values for each point.

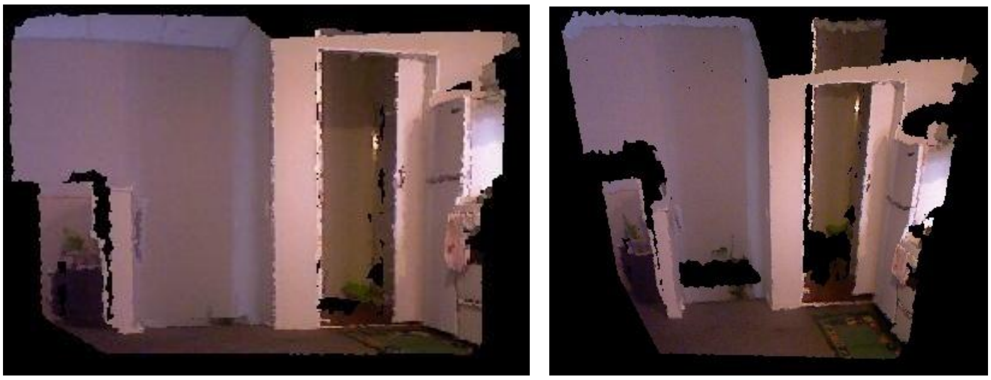


Figure 5.19: **3D Point clouds.**

## 5.5.2 Ground Plane Detection

After applying the RANSAC ground plane detection algorithm, the ground plane was estimated as illustrated in Figure 5.20. Then, morphology technique and Manduchi et al.'s ground/obstacle classification method was implemented to reduce the over- and under-classified ground pixels. A

$5\times5$ structuring element was used for the morphology opening operation, followed by a $3\times3$ structuring element for the closing operation. Manduchi et al.'s algorithm also depends on a set of parameters (the maximum slant angle $\theta_{max}$, the maximum and minimum height $H_{min}$, $H_{max}$ of the cone). According to Manduchi et al.'s experiment, $\theta_{max} = 40°$, $H_{min} = 20mm$ was chosen. However, the larger the value $H_{max}$, the more computation is required. Therefore, $H_{max} = 100mm$ was used in this experiment as it provided desirable results.

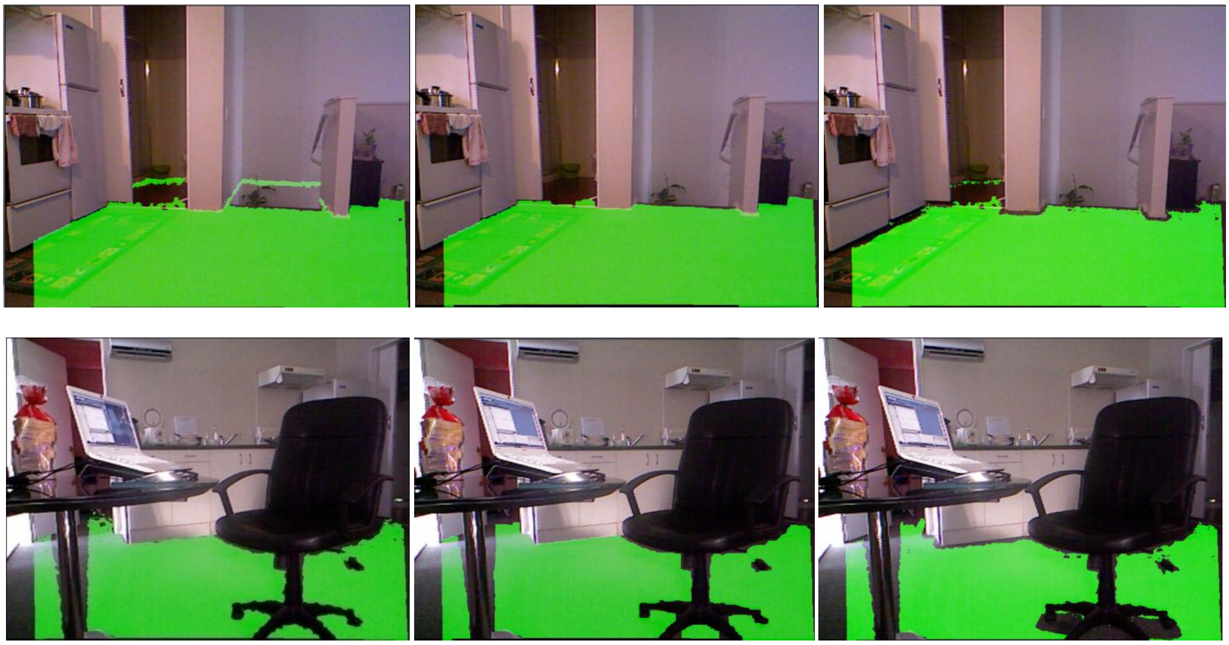


Figure 5.20: **Result of ground plane detection.** (Left) RANSAC ground plane detection. (Middle) Ground pixels after applying the morphology technique. (Right) Ground pixels after applying Manduchi et al.'s algorithm.

The results of the two algorithms are shown in Figure 5.20. Morphology operations smooth the edge of the ground plane mask based on its shape, while Manduchi et al.'s method computes the spatial relationships between the 3D point clouds. Therefore, the ground pixels are more precisely classified by Manduchi et al.'s algorithm. As shown in Figure 5.20, the edges where the floor meets the obstacles and the area under the chair legs are classified as obstacles using Manduchi et al.'s algorithm. Furthermore, Manduchi et al.'s algorithm can also be used to detect and segment obstacles. Figure 5.21 shows the detected obstacles using their algorithm, where the obstacle pixels are clearly labelled. However, Manduchi et al.'s algorithm is computationally inefficient to be used for a real-time application. In comparison, the morphology operations are more efficient and still able to produce useful results.

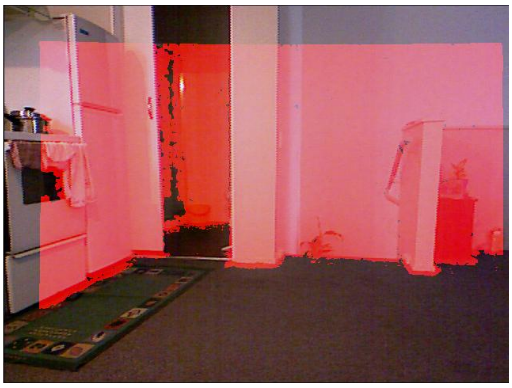


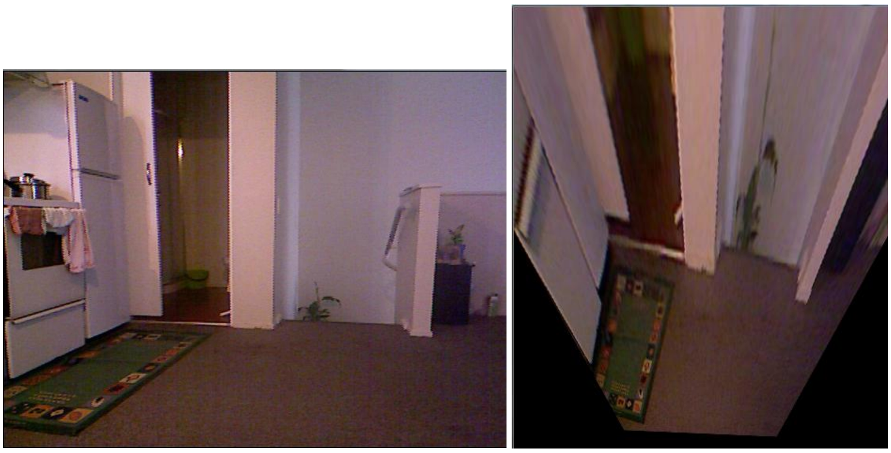
Figure 5.21: **Obstacle detection using Manduchi et al.'s algorithm.**

### 5.5.3 Obstacle Detection and Segmentation

The obstacle detection and segmentation algorithm was implemented which includes top-down map generation, ground plane removal and connected components segmentation.

**Top-down map generation**

The system converts the camera view of the scene into a top-down “bird’s-eye” view as shown in Figure 5.22 using inverse perspective mapping (IPM). The process of the IPM algorithm is illustrated in Figure 5.23. Four points in the top-down map were selected, which were than mapped onto the image plane based on the ground plane equation and the camera's intrinsic parameters. The green quadrangle in the image at the top-right corner of Figure 5.23 is the projection of the four points in the top-down map. It also indicates the position of the ground plane in the image. Then, the four pairs of the corresponding points in the image plane and the top-down map of the ground plane were used to compute a 3-by-3 homography matrix. Afterwards, the matrix was applied to the image plane to create a top-down view of the ground. As non-ground plane pixels appear distorted in the top-down map (see Figure 5.24), pixels that were not lying on the ground were removed from the top-down map.



(a) (b)

Figure 5.22: **Top-down "bird's-eye" view of the scene.** (a) RGB image (b) top-down "bird's eye" view.

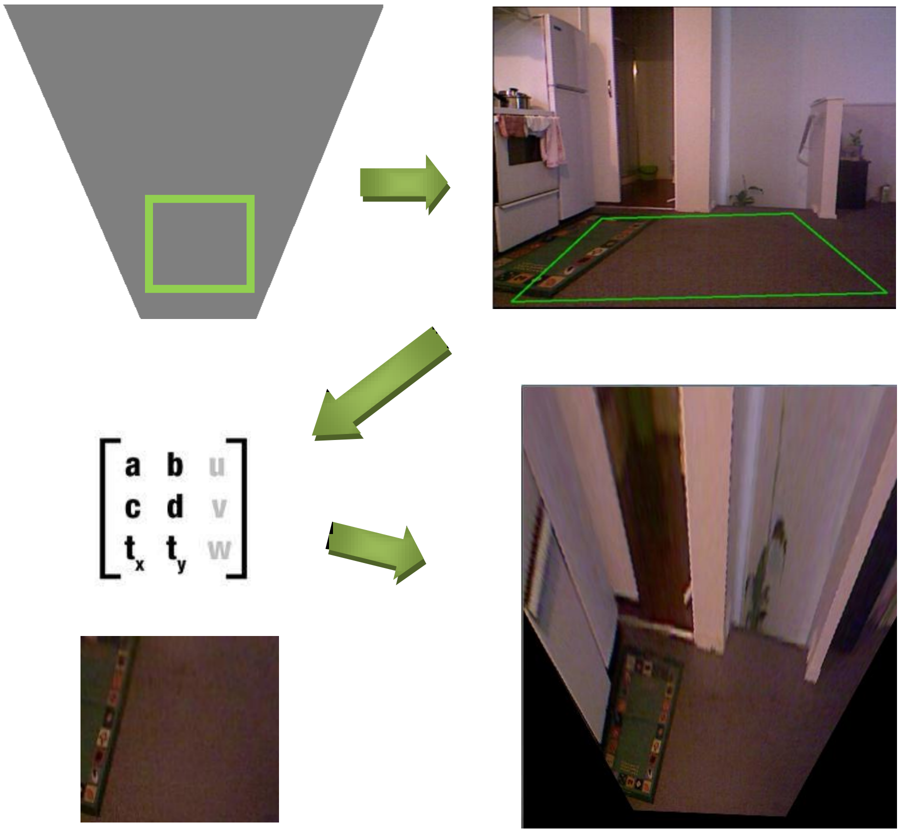


Figure 5.23: **The process of top-down view generation.**

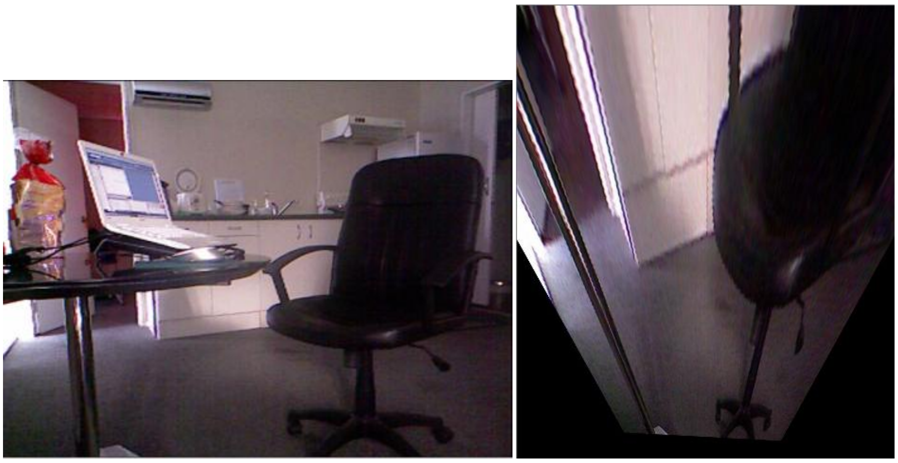
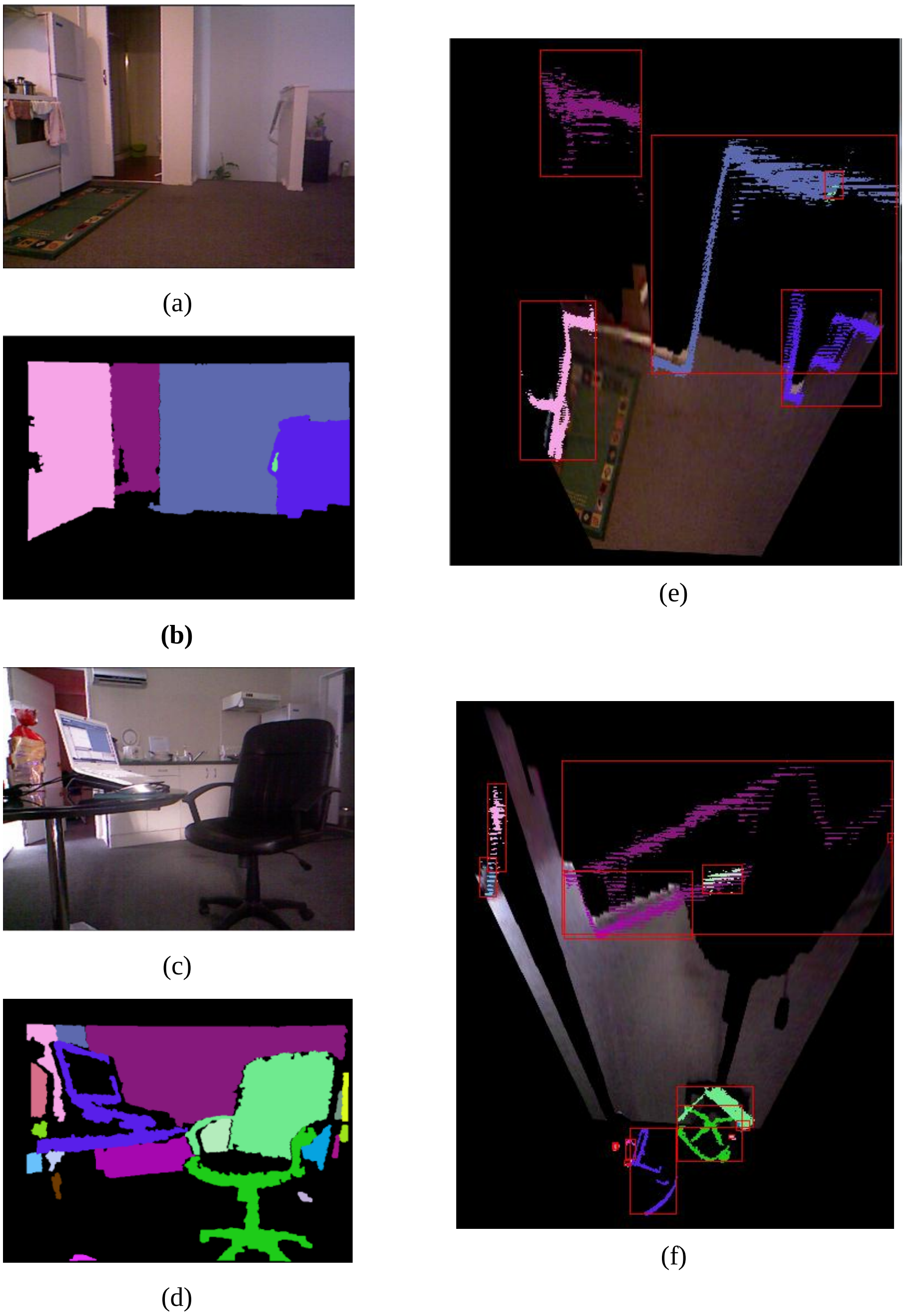


Figure 5.24: **Distorted obstacle points in the top-down view of the ground plane.** (a) RGB image (b) top-down map where a chair appears distorted.

### Ground Surface Removal and Obstacle Segmentation

After the ground surface was removed from the image, the obstacle regions were isolated. The region growing based obstacle segmentation algorithm described in Section 4.3.3 was implemented. The result is shown in Figure 5.25, where each of these obstacles is projected onto the top-down map and is represented by a random colour. As the obstacle segmentation was performed in the depth image space, some background obstacles were divided into many smaller obstacles by the objects in front of them. A boundary cube was computed for each of those obstacles. The rectangles in Figure 5.25 (e, f) are the projection of the boundary cubes on the ground plane. The distance to each obstacle was also calculated.



(a)

**(b)**

(c)

(d)

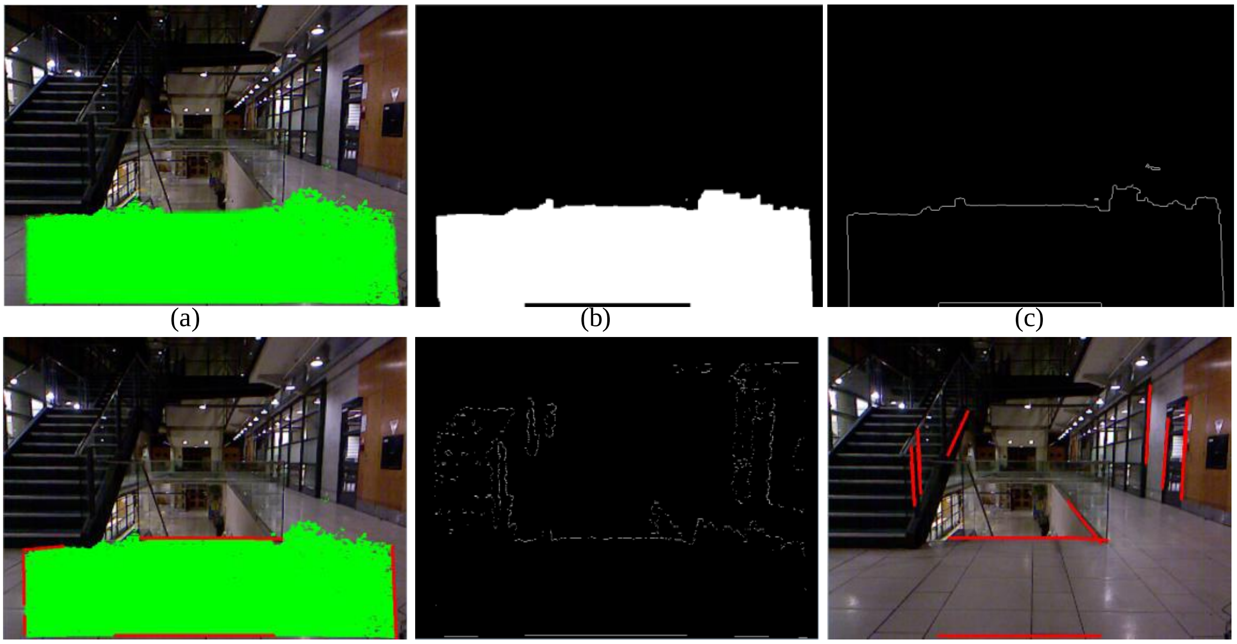
(e)

(f)

Figure 5.25: **Result of obstacle detection and segmentation.** (a, c) RGB image (b, d) Segmented obstacles (e, f) Segmented obstacles in the top-down map.

### 5.5.4 Drop-off Detection

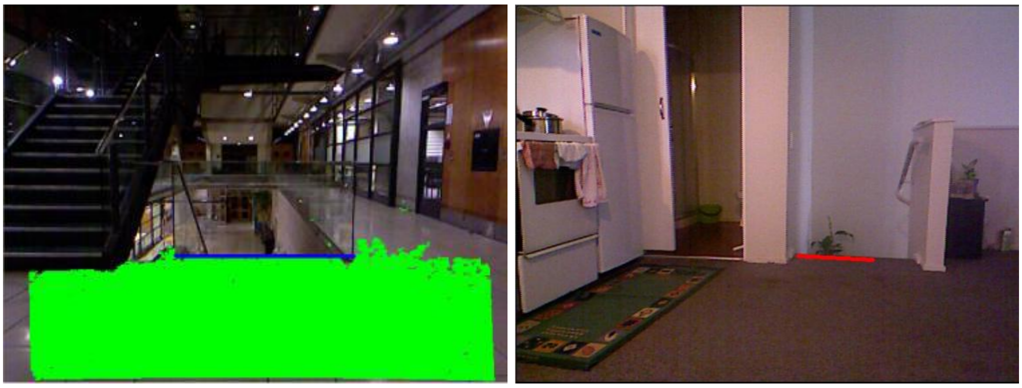
The process of the drop-off detection algorithm is shown in Figure 5.26. Each of the derived straight lines in Figure 5.26 (d) and Figure 5.26 (f) were examined by computing the average heights of the pixels in the region below the line, as a straight line not found from one image may be detected from another. The resulting drop-off line is illustrated in Figure 5.27 (a).



(a) (b) (c)

(d) (e) (f)

Figure 5.26: **Drop-off detection**. (a) detected ground surface (b) ground plane mask after applying the morphology operations (c) extracted boundary of the ground plane mask using the Canny edge detector (d) straight lines on the boundary of the ground plane mask (e) disparity continuity map, pixels with large disparity changes (>0.5m) around then are coloured white (f) straight lines extracted from image (e)



(a) (b)

Figure 5.27: **Drop-off detection**. (a) a drop-off line is detected shown in blue colour. (b) only part of the drop-off is detected.

Results show that the lines extracted from the ground plane mask and the depth discontinuity map often represented only partial drop-off lines, and consequently only part of the drop-off was detected (see Figure 5.27 (b)).

### 5.5.5 Open Door Detection

The open door detection algorithm explained in Section 4.3.5 was implemented. As shown in Figure 5.28, straight lines were extracted by applying the Hough Transform to the depth discontinuity map. Then, a pair of vertical lines represented the door frame were detected by filtering and examining each pair of extracted vertical lines.

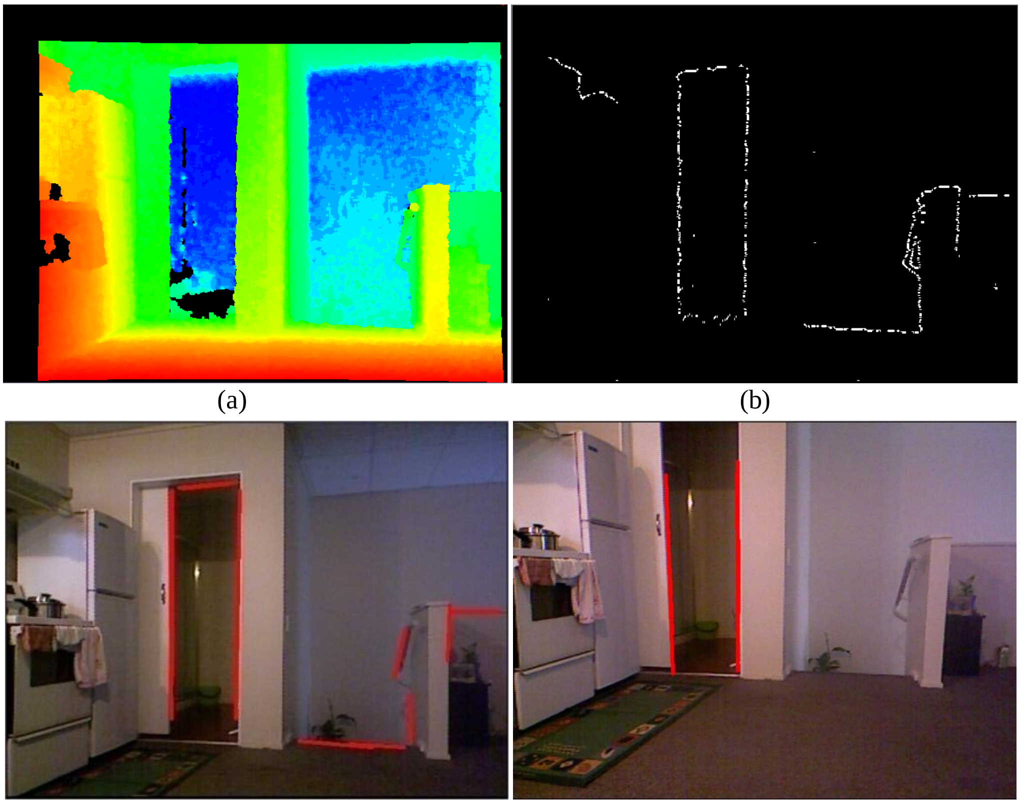

(a) (b) (c) (d)

Figure 5.28: **Open door detection.** (a) Depth map (b) disparity continuity maps (c) the lines extracted using the Hough Transform (d) detected door frame.

## 5.6 Evaluations

### 5.6.1 Ground Classification

An experiment was performed to evaluate the watershed transformation based ground classification algorithm using a stereo camera. Ten images were taken from five different environments. The algorithm was applied to each of those images to generate a ground plane mask image, where ground pixels were represented by 1, and non-ground pixels were represented by 0. In order to measure the pixel level accuracy, a ground truth mask was calculated manually for each of those images, where the ground pixels were labelled. The two mask images were compared pixel by pixel, the number of correctly and incorrectly classified pixels were computed. Moreover, the result of the ground pixel classification without watershed transformation was also compared with the ground truth.

The result of this experiment illustrated in Table 5.3 clearly shows that the watershed transformation increased the accuracy of ground pixel classification. A false positive is a classified ground pixel that is not on the ground plane, and a false negative is a ground pixel that is not detected . The result shows that the average correctly classified rate increased from 87.3% to 98.58% after applying the watershed transformation algorithm, and the average false positive rate is only increased 0.33% (from 0.46% to 0.79%).

| | **Ground/obstacle classification without the watershed transformation** | | | **Ground/obstacle classification with the watershed transformation** | | |
|---|---|---|---|---|---|---|
| | **TP** | **FP** | **FN** | **TP** | **FP** | **FN** |
| 1 | 87.97% | 0.28% | 12.03% | 99.23% | 0.58% | 0.77% |
| 2 | 89.99% | 0.44% | 10.01% | 99.56% | 1.18% | 0.44% |
| 3 | 83.98% | 0.43% | 16.02% | 98.63% | 0.82% | 1.37% |
| 4 | 82.98% | 0.79% | 17.02% | 98.47% | 0.85% | 1.33% |
| 5 | 92.56% | 0.36% | 7.44% | 97.02% | 0.54% | 3.58% |

Table 5.3: **Result of ground/obstacle classification.** TP is true positive, FP is false positive, FN is false negative.

The main reason for the incorrect classification is the over- and under-segmentation of the ground plane. As illustrated in Figure 5.29 (a), a region that contains some ground pixels and some wall pixels is classified as ground plane. Also, as shown in Figure 5.29 (b), the floor is divided into many small regions due to the textures and reflection on the ground surface, and some regions without sufficiently accurate depth values are misclassified.

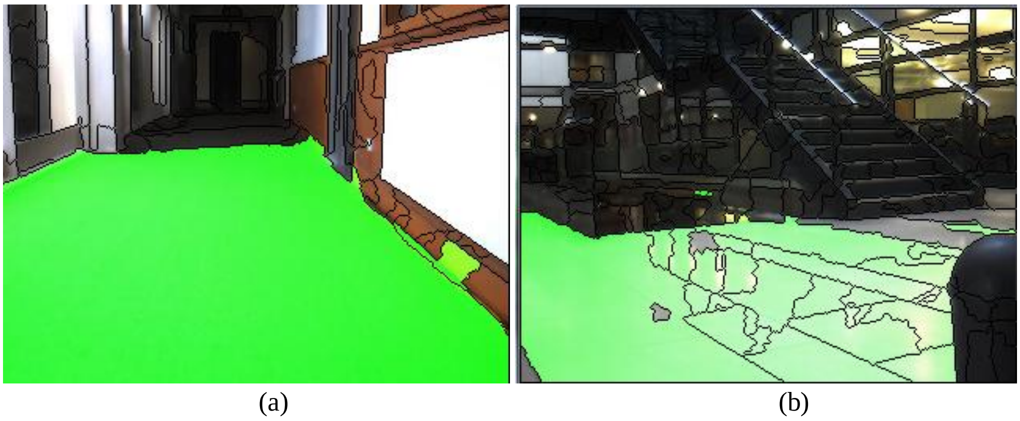


Figure 5.29**: Over- and under- segmentation**. (a) under-segmented image, the edge of the wall is segmented with the same region as the ground (b) over segmented image, the ground plane was segmented into too many pieces.

The morphology technique and the simplified Manduchi et al.'s algorithm that were used for ground pixel classification with a structured light camera were also evaluated. The per-pixel accuracy of the two classification algorithms was measured approximately. The edges where the floor meets the obstacles and the area under the overhanging obstacles were not counted in the correct classification rate, but were taken into account in the false positive rate. Also, the misclassified pixels caused by the empty depth values and pixels out of the camera's best working range were not taken into account. The experiment result is shown in Table 5.8. The percentage of correct classification of morphology operations is around 99%, and Manduchi et al.'s algorithm is around 98.3%. The false positive rate of morphology operations is around 0.5%, while Manduchi et al.'s algorithm has a lower false positive rate (approximately around 0.2%). These numbers are also compared with the algorithm used with the stereo camera. The experiment shows that the algorithms using the structured light camera generate better results. According to the experimental results in [83], their system with monocular vision has worse result with an average true-positive rate of 90.92% and an average false positive rate of 7.78%.

| **Camera Type** | **Algorithm** | **TP** | **FP** |
|---|---|---|---|
| **Structured light** | Morphology operations | 99 % | 0.5% |
| | Manduchi et al.'s algorithm | 98.3% | 0.2% |
| **Stereo** | Watershed based ground classification | 98.58% | 0.79% |

Table 5.4: **Comparison between the three different algorithms .**

### 5.6.2 Drop-off Detection

An experiment was carried out to evaluate the two drop-off detection algorithms. The algorithm with the stereo camera combines the depth information with the straight lines extracted by the proposed standard deviation ridge straight line detector, while the algorithm with the structured light camera employs only the depth information derived from the camera.

Two twenty-second video sequences were used to test the two algorithms respectively. The results in Table 5.8 show that the proposed drop-off detection algorithm with stereo cameras can correctly detect a frontal drop-off with height over 0.3 metres. The detection rate is 100% true positives and 6% false positives. However, the proposed drop-off detection algorithm has difficulty of detecting small drop-offs. By comparison, the drop-off detection algorithm that employs only depth information from a structure light camera has lower detection rate of 90.1% true positives and 14% false positives. The comparison result suggests that the use of multiple approach instead one single approach can increase the accuracy of drop-off detection.

| Camera Type | Algorithm | TP | FP | FN |
|---|---|---|---|---|
| **Stereo** | Stereo and proposed straight line detector based | 100% | 6% | 0% |
| **Structured light** | Depth discontinuity based | 90.1% | 14% | 9.9% |
| **Stereo** | Murarka, et al.'s stereo and motion based [19] | 100% | 7% | 0% |

Table 5.5: **Comparison between the three drop-off detection algorithms .**

In paper [19], an drop-off detection algorithm combined stereo and motion information was proposed. Their experimental results show that with sufficient height of drop-offs, the drop-off detection algorithm achieved accuracy of 100% true positives and 7% false negatives, which were close to the results generated by the drop-off detection algorithm proposed in this research.

### 5.6.3 Stairs up Detection

To evaluate the accuracy of the proposed stairs up detection algorithm, two twenty-second video sequences were used. Although the staircase in the video sequences was always detected, the false positive rate was extremely high (approximately more than 30%) as shown in Table 5.6. The false positive results were often found at obstacles with many edges and lines.

| Camera Type | Algorithm | TP | FP | FN |
|---|---|---|---|---|
| **Stereo** | Stereo and proposed straight line detector based | 100% | > 30% | 0% |

Table 5.6: **Accuracy of the proposed stairs up detection algorithms .**

### 5.6.4 Open Door Detection

To evaluate the accuracy of the open door detection algorithm, three different open doors captured in three ten-second video sequences respectively were used to test the algorithm. Results illustrated in Table 5.7 show that the open door recognition only had 69% true positives and 6% false positives. False positive results were found in incorrectly paired vertical lines.

| Camera Type | Algorithm | TP | FP | FN |
|---|---|---|---|---|
| **Structured light** | Stereo and proposed straight line detector based | 69% | 6% | 31% |

Table 5.7: **Accuracy of the open door detection.**

### 5.6.5 Distance measurement

The accuracy of distances to the detected indoor features was evaluated where Figure 5.30 graphs results of these distance measurements. It shows that by using the Bumblebee stereo camera, within 4 metres, the error of distance to a detected drop-off is within 0.25 metres (6.25%), and within 3 metres, the error of distance to a detected stairs up is within 0.3 metres (10%).

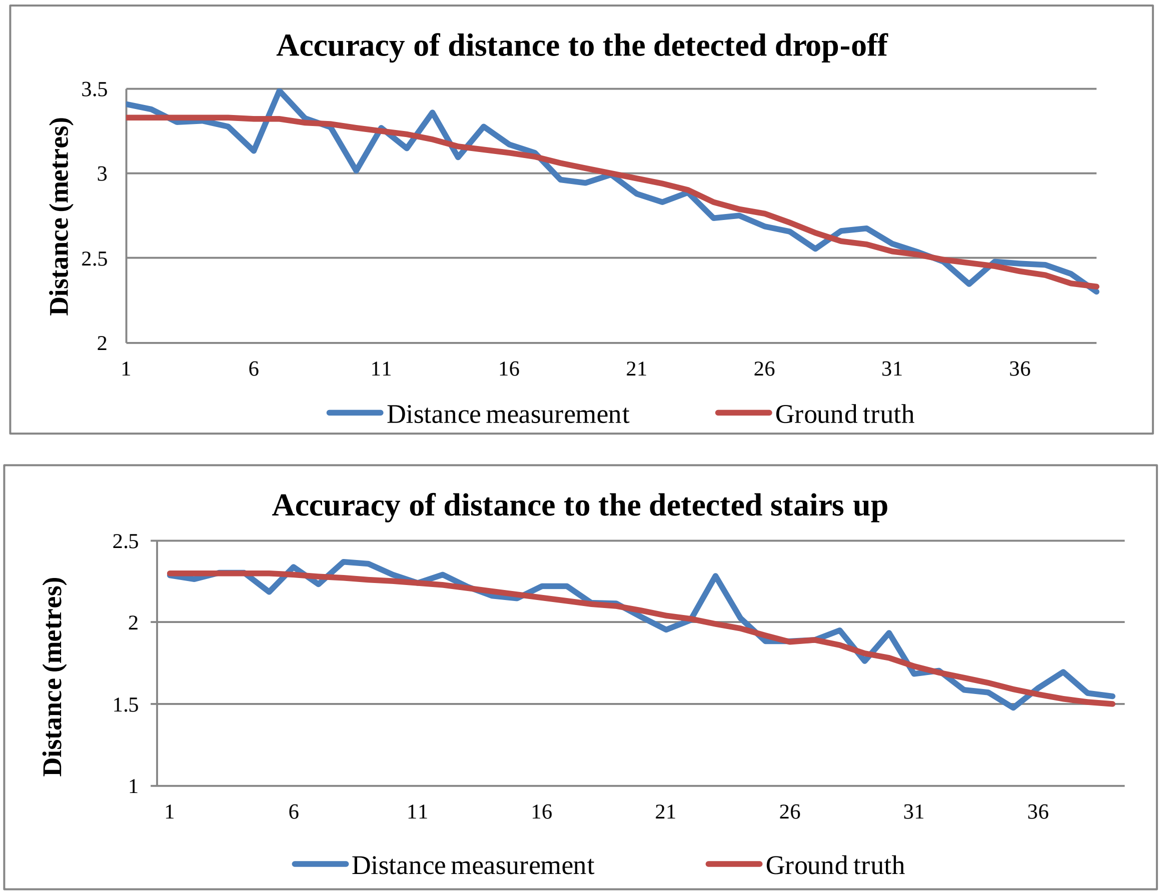


Figure 5.30**: Results of measured distance to the detected indoor features using the stereo camera.**

The accuracy of measured distances to the detected obstacles was also evaluated. The results show that by using a structured light camera Kinect, within 3 metres, the error of distance to a detected obstacle is within 0.1 metres (3.3%).

Several factors influence the accuracy of measured distance to the detected obstacles and indoor features. First, the accuracy of the results is heavily dependent on the depth measurements provided by the depth sensor. For example, the accuracy of the depth measurement by the Kinect camera can be influenced by other IR sources such as direct sunlight coming in through a window. Second, reflective or transparent surfaces can introduce significant depth errors using both types of cameras. Third, line extraction methods may divide an actual corridor line into several line segments which could potentially cause false feature recognition.

### 5.6.6 Performance

The performance of the proposed method with the stereo camera was tested by implementing timing the output for the following procedures: the disparity interpolation/smoothing/noise-reduction, ground plane estimation, watershed based image segmentation, standard deviation ridge straight line detection, drop-off detection, stairs up detection and motion calculation. The system took an input with 320×240 resolution each time. The average performance of each procedure can be seen in Table 5.8.

| Step | Times (ms) |
|---|---|
| Disparity interpolation/smoothing/noise-reduction | 52 |
| RANSAC ground plane detection | 7 |
| Watershed segmentation | 170 |
| Standard deviation ridge straight line detection | 179 |
| Drop-off detection | 2.5 |
| Stairs up detection | 2.6 |
| Optical flow points selection | 225 |
| Optical flow points tracking and motion calculation | 54 |

Table 5.8: **Performance of the proposed method with the stereo camera**.

The performance of the proposed method with the structured light camera was evaluated in the following areas: ground plane estimation, ground pixel classification, frame of reference calculation, obstacle detection, drop-off detection and open door detection. Input images with 640×480 were used. The average performance of each section is shown in Table 5.9.

| Step | Times (ms) |
|---|---|
| RANSAC ground plane detection | 45 |
| Ground pixel classification | 38 |
| Reference frame calculation | 26 |
| Obstacle detection | 180 |
| Drop-off detection | 54 |
| Open-door detection | 79 |

Table 5.9: **Performance of the proposed method with the structured light camera**.

## 5.7 Summary

This chapter presents the results of the two proposed indoor feature recognition and obstacle detection methods. Several experiments were carried out to evaluate the two proposed methods. Sensor comparison in Section 5.3 shows that the structured light camera Kinect is preferable for use with smart wheelchairs in indoor environments compared with the Bumblebee stereo camera. The ground pixel classification using the watershed based image segmentation algorithm with stereo camera achieved an average true positive rate of 98.58%, and the figure was increased to around 99% using simple morphology operations with structured light camera. In addition, results show that the proposed standard deviation ridge straight line detector algorithm is able to extract lines more accurately than the Canny edge detector and the Hough transform. The results of drop-off detection using the standard deviation ridge straight line detector and stereo depth information is accurate, while the drop-off detection using a single depth-sensor-based approach has only 90.1% true positive rate. The stairs up detection using the standard deviation ridge straight line detector produces a high false positive rate (30%). The motion information provided by calculating the optical flow with stereo camera is unreliable because of the incorrectly interpolated disparity values. Moreover, the obstacle detection and segmentation algorithm using the structured light camera provides an accurate top-down map of the ground plane and the obstacles. However, the open door recognition algorithm using the structure light camera only had 69% true positive rate and 6% false positive rate. Results show that the distances to various detected indoor feature and obstacles are accurately computed.

# Chapter 6
# Conclusion and Future Work

## 6.1 Conclusion

This research successfully evaluated two proposed algorithms for indoor feature recognition and obstacle detection. One was through use of a stereo camera, and the other a structured light camera with both methods combining different visual cues to recognize static indoor features, detect obstacles and calculate movements.

The proposed method with the stereo camera combines stereo processing, disparity interpolation, ground plane detection, reference frame calculation, image segmentation, obstacle detection, line extraction, drop-off detection, stairs-up detection and motion calculation to provide useful data for indoor navigation assistance. The main novelty of this proposed method is the use of combined information on colour, edge, depth and motion to derive robust indoor feature recognition and obstacle detection. Another novelty is the introduction of a standard deviation ridge straight line detector algorithm, and the use of the algorithm for both drop-off detection and stairs up detection. As a result, obstacles and static indoor room features such as ground plane, drop-off and stairs up are detected, and camera movements are derived.

The proposed method with the structured light camera combines image registration, ground plane detection, reference frame calculation, obstacle detection, drop-off detection and open door detection. The main novelty of this proposed method is the use of the structured light camera to take the place of the stereo camera for obstacle detection and indoor feature recognition.

Several experiments were performed to evaluate the two proposed methods. These included evaluations for the algorithms of ground pixel classification, obstacle detection, drop-off detection, stairs up detection, open door detection, and evaluation for distances to various detected indoor features and obstacles.

Compared to prior research, the proposed ground plane estimation algorithm with stereo camera is more robust against noise, as the proposed method involves a disparity interpolation/smoothing/noise-reduction algorithm. Results show that the ground pixel classification using the watershed based image segmentation algorithm method with the stereo camera achieved an average true positive rate of 98.58%, and the figure was increased to around 99% using simple morphology operations with the structured light camera. Results show that the proposed standard deviation ridge straight line detector algorithm is able to extract lines more accurately than the Canny edge detector and the Hough transform. The drop-off detection using the standard deviation ridge straight line detector with stereo camera has achieved a significantly better accuracy. In addition, the obstacle detection and segmentation algorithms provide an accurate perspective view of the surrounding with accurately calculated distances to various detected indoor features and obstacles. Compared to prior research, obstacle detection and segmentation using proposed method is more efficient and accurate.

In summary, the proposed algorithm is designed to fulfil the research objectives of:

1. Developing a robust method for indoor static feature recognition. This includes ground plane detection, doorway and hallway detection, stairs up and drop-off detection.
2. Developing a robust method for obstacle detection and segmentation.
3. Providing motion information of the wheelchair.

## 6.2 Future Work

Future research will pursue optimising the following aspects:

1. In this research, only some indoor static features can be recognized including ground planes, drop-offs, stairs up and open doors. Further research will be carried out to extend the current indoor feature recognition to inclined plane recognition and transparent/reflective surface detection.
2. The system proposed in this research calculates only the camera's ego-motion. An extension of this research is integrating optical flow and range information to detect and calculate the

movements of dynamic objects such as moving people. The relative speed of dynamic objects and the moving camera with time-to-impact will be estimated. Future work will combine the depth information obtained from the structured light camera and the motion information to derive more accurate camera movements.

3. Further research will be extended to provide navigation assistance in both the indoor and outdoor environments. The system will merge the two proposed methods, using the stereo camera for outdoor daytime environments and the structured light camera for outdoor night-time environments and indoor environments.

# References

[1] R. C. Simpson, "Smart wheelchairs: A literature review," *Journal of Rehabilitation Research and Development, vol. 42*(4), pp. 423-438, Jul-Aug 2005.

[2] R. C. Simpson, E. F. LoPresti, and R. A. Cooper, "How many people would benefit from a smart wheelchair?," *J Rehabil Res Dev, vol. 45*(1), pp. 53-71, 2008.

[3] L. Fehr, W. E. Langbein, and S. B. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey," *Journal of Rehabilitation Research and Development, vol. 37*(3), pp. 353-360, 2000.

[4] L. Iocchi and S. Pellegrini, "Building 3D maps with semantic elements integrating 2D laser, stereo vision and IMU on a mobile robot," in *Proceedings of the 2nd ISPRS Int. Workshop 3D-ARCH*, 2007.

[5] H. A. Yanco, "Wheelesley: A Robotic Wheelchair System: Indoor Navigation and User Interface," in *Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing*, 1998, pp. 256-268.

[6] U. Borgolte, H. Hoyer, C. Bühler, H. Heck, and R. Hoelper, "Architectural Concepts of a Semi-autonomous Wheelchair," *Journal of Intelligent and Robotic Systems, vol. 22*(3-4), pp. 233-253, 1998.

[7] G. D. Castillo, S. Skaar, A. Cardenas, and L. Fehr, "A sonar approach to obstacle detection for a vision-based autonomous wheelchair," *Robot. Auton. Syst., vol. 54*(12), pp. 967-981, 2006.

[8] A. Aboshosha, "An Introduction to Robot Distributed Sensors Part I," Technical Report, University of Tubingen, 2003.

[9] A. Lankenau and T. Rofer, "A versatile and safe mobility assistant," *IEEE Robotics & Automation Magazine, vol. 8*(1), pp. 29 - 37, 2001.

[10] E. Prassler, J. Scholz, and P. Fiorini, "A robotics wheelchair for crowded public environment," *IEEE Robotics & Automation Magazine, vol. 8*(1), pp. 38-45, 2001.

[11] Wikipedia. *Time-of-flight camera*. Available: http://en.wikipedia.org/wiki/Time-of-flight_camera.

[12] G. R. Bradski and A. Kaehler, *Learning OpenCV : Computer vision with the OpenCV library*, 1st ed. Sebastopol, CA: O'Reilly, 2008.

[13] G. Baguley, "Stereo tracking of objects with respect to a ground plane," Unpublished honours report, Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand, 2009.

[14] D. Fofi, T. Sliwa, and Y. Voisin, "A comparative survey on invisible structured light," *Machine Vision Applications in Industrial Inspection XII*, pp. 90-98, 2004.

[15] Wikipedia. *Structured-light 3D scanner*. Available: http://en.wikipedia.org/wiki/Structured-light_3D_scanner.

[16] K. McDonald and G. Levin. (2011). *Kinect Programming: An Introductory Workshop*. Available: http://futuretheater.net/wiki/Kinect_Workshop.

[17] Wikipedia. *Kinect*. Available: http://en.wikipedia.org/wiki/Kinect.

[18] G. N. DeSouza and A. C. Kak, "Vision for Mobile Robot Navigation: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24*(2), pp. 237-267, 2002.

[19] A. Murarka, M. Sridharan, and B. Kuipers, "Detecting obstacles and drop-offs using stereo and motion cues for safe local motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, 2008, pp. 702 - 708.

[20] D. Burschkal, S. Lee, and G. Hager, "Stereo-based obstacle avoidance in indoor environments with active sensor re-calibration," in *Proceedings IEEE International Conference on Robotics and Automation, 2002. ICRA '02*, 2002

[21] I. Ulrich and I. R. Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence,* 2000, pp. 866-871.

[22] N. Tada, K. Murata, T. Saitoh, T. Osaki, and R. Konishi, "Monocular vision based indoor mobile robot," in *The 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC2008),* 2008, pp. 41-44.

[23] C. N. Viet and I. Marshall, "An efficient obstacle detection algorithm using colour and texture," in *Proceedings of World Academy of Science, Engineering and Technology*, 2009, pp. 123-128.

[24] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28*(5), pp. 694 - 711, 2006.

[25] M. Sarcinelli-Filho, H. A. Schneebeli, and E. M. Caldeira, "Using optical flow to control mobile robot navigation," in *Proceedings of the 15th IFAC World Congress*, 2002.

[26] Z. Chen, N. Pears, and B. Liang, "Monocular obstacle detection using reciprocal-polar rectification," *Image and Vision Computing, vol. 24*(12), pp. 1301-1312, 2006.

[27] T. Naito, T. Ito, and Y. Kaneda, "The obstacle detection method using optical flow estimation at the edge image," *IEEE Intelligent Vehicles Symposium, 2007* pp. 817 - 822 2007.

[28] Y. Shen, X. Du, and J. Liu, "Monocular Vision Based Obstacle Detection for Robot Navigation in Unstructured Environment Advances in Neural Networks – ISNN 2007." vol. 4491, ed: Springer Berlin / Heidelberg, 2007, pp. 714-722.

[29] T. Low and G. Wyeth, "Obstacle detection using optical flow," in *Proceedings of Australasian Conference on Robotics and Automation*, Sydney, N.S.W, 2005.

[30] C. Demonceaux, A. Potelle, and D. Kachi-Akkouche, "Obstacle detection in a road scene based on motion analysis," *IEEE Transactions on Vehicular Technology, vol. 53*(6), pp. 1649-1656 2004.

[31] T. Wekel, O. Kroll-Peters, and S. Albayrak, "Vision based obstacle detection for wheeled robots," in *International Conference on Control, Automation and Systems, 2008. ICCAS*, 2008 pp. 1587 - 1592

[32] X. Wang, K. Ban, and K. Ishii, "Estimation of mobile robot ego-motion and obstacle depth detection by using optical flow," in *Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, 2009, pp. 1770-1775.

[33] G.-S. Young, M. Herman, T.-H. Hong, D. Jiang, and J. C. S. Yang, "New Visual Invariants for Terrain Navigation Without 3DReconstruction," *International Journal of Computer Vision, vol. 28*(1), pp. 45-71, 1998.

[34] S. I. Shah, "Vision based 3D obstacle detection using a single camera for robots/UAVs," Unpublished master's thesis, Georgia Institute of Technology, George W Woodruff School of Mechanical Engineering, Atlanta, Georgia, the United States, 2009.

[35] R. Cucchiara, E. Perini, and G. Pistoni, "Efficient stereo vision for obstacle detection and AGV navigation," in *Proceedings of the 14th International Conference on Image Analysis and Processing*, 2007, pp. 291-296.

[36] S. Fazli, H. M. Dehnavi, and P. Moallem, "An advanced stereo vision based obstacle detection with a robust shadow removal technique," *World Academy of Science, Engineering and Technology,* (67), pp. 699-705, 2010.

[37] P. Foggia, A. Limongiello, and M. Vento, "A moving object and obstacle detection system in real-time AVG and AMR applications," in *ICGST International Conference on Automation, Robotics and Autonomous Systems (ARAS-05),* 2005, pp. 19-24.

[38] J. Klappstein, T. Vaudrey, C. Rabe, A. Wedel, and R. Klette, "Moving Object Segmentation Using Optical Flow and Depth Information," in *Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology*, Tokyo, Japan, 2008, pp. 611-623.

[39] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, "Moving obstacle detection in highly dynamic scenes," in *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 4451-4458.

[40] A. Agarwal, C. V. Jawahar, and P. J. Narayanan, "A survey of planar homography estimation techniques," Technical Report IIIT/TR/2005/12, International Institute of Information Technology, Centre for Visual Information Technology, 2005.

[41] N. Ohnishi and A. Imiya, "Dominant plane detection from optical flow for robot navigation," *Pattern Recognition Letters, vol. 27*(9), pp. 1009-1021, 2006.

[42] Wolfram Math World. *Least Squares Fitting*. Available: http://mathworld.wolfram.com/LeastSquaresFitting.html.

[43] Wikipedia. *RANSAC*. Available: http://en.wikipedia.org/wiki/RANSAC.

[44] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, Martin Fischler, and B. Gerkey, "Outdoor Mapping and Navigation using Stereo Vision," in *Experimental robotics: the 10th International Symposium on Experimental Robotics*, Brazil, 2006.

[45] S. Choi, T. Kim, and W. Yu. (2009) Performance evaluation of RANSAC family. *In Proceedings of the British Machine Vision Conference (BMVC).*

[46] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereo vision on non flat road geometry through "V-disparity" representation," *IEEE Intelligent Vehicle Symposium, vol. 2*, pp. 646-651, 2002

[47] Y. Gao, X. Ai, Y. Wang, J. Rarity, and N. Dahnoun, "U-V-disparity based obstacle detection with 3D camera and steerable filter," in *2011 IEEE Intelligent Vehicles Symposium (IV),* Baden, 2011, pp. 957 - 962.

[48] Z. Hu, F. Lamosa, and K. Uchimura, "A Complete U-V-Disparity Study for Stereovision Based 3D Driving Environment Analysis," in *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, 2005, pp. 204-211.

[49] A. Sappa, R. Herrero, F. Dornaika, D. Gerónimo, and A. López, "Road approximation in Euclidean and v-disparity Space: A comparative study," in *Computer Aided Systems Theory – EUROCAST 2007. 11th International Conference on Computer Aided Systems Theory*. vol. 4739, R. Moreno Díaz, F. Pichler, and A. Quesada Arencibia, Eds., ed: Springer Berlin / Heidelberg, 2007, pp. 1105-1112.

[50] M. Snaith, D. Lee, and P. Probert, "A low-cost system using sparse vision for navigation in the urban environment," *Image and Vision Computing, vol. 16*(4), pp. 225-233, 1998.

[51] H. Uchiyama and W. D. Potter., "Behavior-based perceptual navigation for semi-autonomous wheelchair operations," in *Proceedings of the 2008 International Conference on Artificial Intelligence*, 2008, pp. 700-706.

[52] R. F. Vassallo, H. J. Schneebeli, and J. Santos-Victor, "A purposive strategy for visual-based navigation of a mobile robot," in *Proceedings of the 1998 Midwest Symposium on Systems and Circuits*, 1998, pp. 334-337.

[53] A.-M. Zou, Z.-G. Hou, M. Tan, and D. Liu, "Vision-guided mobile robot navigation," in *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, 2006. ICNSC '06*, Ft. Lauderdale, FL, 2006, pp. 209-213.

[54] W. Shi and J. Samarabandu, "Corridor line detection for vision based indoor robot navigation," in *Canadian Conference on Electrical and Computer Engineering, 2006. CCECE '06*, Ottawa, Ont., 2006, pp. 1988-1991.

[55] H.-S. Lho, J.-W. Kwon, D. Chwa, and S.-K. Hong, "Vision-based corridor line detection using k-means algorithm," in *International Conference on Control, Automation and Systems, 2007. ICCAS '07*, Seoul, 2007, pp. 1052-1056.

[56] A. Rankin, A. Huertas, and L. Matthies, "Evaluation of stereo vision obstacle detection algorithms for

off-road autonomous navigation," in *Proceedings of the 32nd AUVSI Symposium on Unmanned Systems*, 2005.

[57] A. Rankin, A. Huertas, and L. Matthies, "Nighttime negative obstacle detection for off-road autonomous navigation," in *SPIE Defense & Security Symposium: Unmanned Systems Technology IX Conference*, 2007.

[58] S. Se and M. Brady, "Vision-based detection of kerbs and steps," in *British Machine Vision Conference*, 1997, pp. 410-419.

[59] X. Lu and R. Manduchi, "Detection and localization of curbs and stairways using stereo vision," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005*, 2006, pp. 4648-4654

[60] V. Pradeep, G. Medioni, and J. Weiland, "Piecewise planar modeling for step detection using stereo vision," in *CVAVI, European Conference on Computer Vision (ECCV)*, Marseilles, 2008.

[61] J. Vergés-Llahı, "State-of-the-art survey on color segmentation methods," 2005.

[62] Y. Cui and N. Zhou, "Blob analysis using watershed transformation," presented at the Proceedings of the 13th international conference on Industrial and engineering applications of artificial intelligence and expert systems: Intelligent problem solving: methodologies and approaches, New Orleans, Louisiana, United States, 2000.

[63] E. Fazl-Ersi and J. K. Tsotsos, "Region Classification for Robust Floor Detection in Indoor Environments," in *Proceedings of the 6th International Conference on Image Analysis and Recognition*, Halifax, Nova Scotia, Canada, 2009, pp. 717-726.

[64] T.-W. Chen, Y.-L. Chen, and S.-Y. Chien, "Fast image segmentation based on K-Means clustering with histograms in HSV color space," in *2008 IEEE 10th Workshop on Multimedia Signal Processing*, Cairns, Qld, 2008, pp. 322-325

[65] R. Hidayat and R. Green, "Real-time texture boundary detection from ridges in the standard deviation space," in *British Machine Vision Conference*, 2009.

[66] J. Shi and C. Tomasi, "Good Features to Track," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), 1994*, 1994, pp. 593 - 600.

[67] J.-Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker: Description of the algorithm," 2002.

[68] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation," *Auton. Robots, vol. 18*(1), pp. 81-102, 2005.

[69] OpenKinect. (2011). Available: http://openkinect.org/wiki/Main_Page.

[70] OpenNI. (2011). *OpenNI User Guide*. Available: http://www.openni.org/Documentation.aspx.

[71] Microsoft Research. (2011). *Programming guide: Getting started with the Kinect for Windows SDK Beta*. Available: http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/docs/ProgrammingGuide_KinectSDK.pdf.

[72] OpenKinect. (2010). *Imaging Information*. Available: http://openkinect.org/wiki/Imaging_Information.

[73] N. Burrus. *Kinect Calibration*. Available: http://nicolas.burrus.name/index.php/Research/KinectCalibration.

[74] A. M. Muad, "Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system," *TENCON 2004. 2004 IEEE Region 10 Conference, vol. A*, pp. 207 - 210, 2004.

[75] H. A. Mallot, Bulthoff, H., Little, J., Bohrer, and S., "Inverse perspective mapping simplified optical flow computation and obstacle detection," *Biological Cybernetics, vol. 64*, pp. 177-185, 1991.

[76] P. Khandelwal and P. Stone, "A Low Cost Ground Truth Detection System Using the Kinect," in *Proceedings of the RoboCup International Symposium 2011 (RoboCup 2011)*, 2011.

[77] K. Khoshelham and S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors, vol. 12*(2), pp. 1437-1454, 2012.

[78] F. Pece, J. Kautz, and T. Weyrich, "Three Depth-Camera Technologies Compared," in *First BEAMING Workshop 2011*, Barcelona, 2011.

[79] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, "Fall detection from depth map video sequences," in *Proceedings of the 9th international conference on Toward useful services for elderly and people with disabilities: smart homes and health telematics*, Montreal, Canada, 2011, pp. 121-128.

[80] M. Wanduh, "Obstacle detection using V-disparity: Integration to the CRAB rover," Bachelor Thesis, Autonomous Systems Lab, Swiss Federal Institute of Technology Zurich, 2009.

[81] N. Chumerin and M. M. Hulle, "Ground plane estimation based on dense stereo disparity," in *The Fifth International Conference on Neural Networks and Artificial Intelligence*, 2008, pp. 209-213.

[82] Y. Cong, X. Li, J. Liu, and Y. Tang, "A stairway detection algorithm based on vision for UGV stair climbing," in *IEEE International Conference on Networking, Sensing and Control, 2008. ICNSC 2008.*, 2008, pp. 1806 - 1811.

[83] T. Low and A. Manzanera, "Ground-plane classification for robot navigation: combining multiple cues toward a visual-based learning system," in *11th International Conference on Control, Automation, Robotics and Vision (ICARCV 2010)*, Singapore, 2010.