2006

An Image Understanding System for Detecting Indoor Features

Wenxia Shi
Western University

An Image Understanding System for Detecting Indoor Features

(Spine Title: Detecting Indoor Features)

(Thesis format: Monograph)

by

Wenxia Shi

Graduate Program
in
Engineering Science
Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Engineering Science

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario, Canada

ABSTRACT

The capability of identifying physical structures of an unknown environment is very important for vision based robot navigation and scene understanding. Among physical structures in indoor environments, corridor lines and doors are important visual landmarks for robot navigation since they show the topological structure in an indoor environment and establish connections among the different places or regions in the indoor environment. Furthermore, they provide clues for understanding the image.

In this thesis, I present two algorithms to detect the vanishing point, corridor lines, and doors respectively using a single digital video camera. In both algorithms, we utilize a hypothesis generation and verification method to detect corridor and door structures using low level linear features. The proposed method consists of low, intermediate, and high level processing stages which correspond to the extraction of low level features, the formation of hypotheses, and verification of the hypotheses via seeking evidence actively. In particular, we extend this single-pass framework by employing a feedback strategy for more robust hypothesis generation and verification.

We demonstrate the robustness of the proposed methods on a large number of real video images in a variety of corridor environments, with image acquisitions under different illumination and reflection conditions, with different moving speeds, and with different viewpoints of the camera. Experimental results performed on the corridor line detection algorithm validate that the method can detect corridor line locations in the presence of many spurious line features about one second. Experimental results carried on the door detection algorithm show that the system can detect visually important doors in an image with a very high accuracy rate when a robot navigates along a corridor environment.

Keywords: vision based robot navigation, vanishing point, corridor line detection, door detection, hypothesis generation and verification, feedback mechanism.

ACKNOWLEDGEMENTS

I owe a debt of gratitude to many people who have helped me with my graduate study and research in many ways. First of all, I would like to express my deepest gratitude and appreciation to my advisor, Dr. Jagath Samarabandu, for his excellent guidance and generous support. He offered me such a great opportunity so that I can have my graduate studies at University of Western Ontario. I have benefited a lot from the study and research under his supervision. And I believe I will be able to continue to benefit from this period of experience in my future life.

My greatest and heartfelt thanks must also go to Dr. Vijay Parsa who encouraged and helped me in publishing my first paper. I thank Dr. Hanif Ladak for leading me to the exciting world of image processing, Dr. Mehrdad Moallem for providing such a good course. I would also like to express my appreciation for the help and support of the entire Electrical Engineering and Graduate Studies department staff, including Sandra and Sharon.

I would like to thank to my colleagues, Kamal Ranaweera, Ranga Rodrigo, Zhenhe (Hogan) Chen, Xiaoqing Liu and Chandima Wijesinghe. In particular, I thank Ranga for his valuable suggestions and help in my academic area. I really appreciate Chandima who helped me carry the big tripod and take so many video sequences covering almost all the buildings at the University of Western Ontario. My work would not have been so smooth without your help. I thank Hogan for generously sharing his information and experience with our lab mates.

Last but not least, I am grateful to my parents, my husband Gang Wang and my son Alex for their love and support over the years, for believing in and encouraging me during my hardest time.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Motivation

The capability of identifying physical structures in an unknown environment is very important for autonomous mobile robot navigation and scene understanding. Typical structures in an indoor environment are visually salient features such as corridors, doors, walls, windows, and pathways etc. The challenge of a computer vision system is to be capable of recognizing natural structures and providing the necessary semantic interpretations of their environment. For example, we would like to specify the tasks to be accomplished by the robot in high-level semantic terms, such as "go down the hallway and go through the last door on the left." In order to execute such a task, the robot must be able to identify the objects of interest (here, "hallway and doors") in its perception of the environment.

In our system, we concentrate on identifying corridors and doors since they are the visual landmarks which establish connections among the different places or regions in the environment.

Detection of corridor line locations and the vanishing point provide important information for robot navigation. A corridor line is the intersection line between a wall and the floor. The absolute corridor line locations not only determine the robot's transversal moving range in a corridor, but also provide clues for understanding the image. Vanishing point is a special perspective projection effect. It is well known that with a pinhole perspective projection model, lines parallel to each other in the 3-D space will converge to a common endpoint, called the vanishing point [1] [2] , in the 2-D projection plane. The knowledge of the vanishing point can be used for many purposes on robot navigation, such as inferring the possible directions of advancement, enhancing further analysis of the scene, localizing the robot's position, or building 3-D structure model of the corridor environment.

Doors are one of the most common landmarks for robot navigation since they show the topological structure in indoor environment and mark the entrance/exit of rooms in many office and laboratory environments. Moreover, it is essential to detect doors in indoor scenes to build a map for the environment.

In this thesis, we will develop algorithms to detect corridor lines and doors in corridor scenes using a single video camera. In mobile robot applications it is required to reduce the time consumption as much as possible. So the algorithms are required to work in real-time and perform effectively under variation of illumination, reflection and noises.

1.2 Overview of the System and Results

This dissertation presents two algorithms to detect vanishing point, corridor line locations, and doors respectively. In both algorithms, we utilize a hypothesis generation and verification (HGV) method to detect corridor and door structures using low level linear features. The system diagram of the proposed methods is shown in Figure 1.1. The system consists of low level, intermediate level and high level processing stages. This transformation of signals into symbols consists of the extraction of low level features, the formation of a hypothesis through grouping of multiple features, and finally, verification of the hypothesis about potential structure via certain forms of comparison and confirmation against rules.

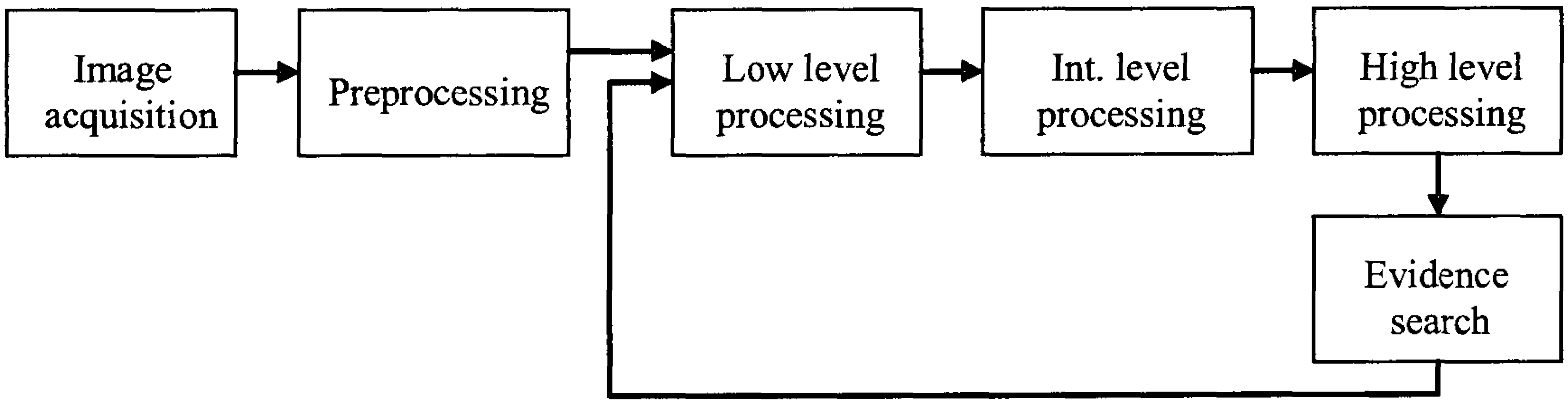


Figure1.1: Block diagram of the system.

In particular, we extend this single-pass framework by employing feedback strategy for more robust hypothesis generation and verification. The feedback scheme has different implementations in the two algorithms. In corridor line detection algorithm, the information is fed back at high level stage to search evidence for hypothesis verification. In door detection algorithm, the feedback mechanism directs the system to search for missing information in windows of interest by tuning down parameter setting, so as to form more robust hypothesis generation. Besides this, the system also needs feedback to search further evidence for hypothesis verification.

The image sources applied in the proposed method are video sequences acquired using a digital video camera mounted on the top of a moving robot or a rolling tripod. More details about image acquisition are described in Chapter 2.

The quality of the video images brings additional challenges to the task of low level feature extraction due to vibration of the moving camera, noise introduced by camera when capturing videos as well as the poor resolution and image quality of consumer grade video cameras used in our experiments. In addition, large variations in brightness and contrast, and reflections also cause difficulties in feature extraction. Therefore, preprocessing of the input image frame is required in order to suppress noise but still preserve sharp edges. We use a Sigma filter at this stage and more details are described in section 4.1.3.

1.2.1 Corridor Line Location Detection Algorithm

Corridor line location detection algorithm is developed for detecting vanishing point and true corridor line locations in the presence of many spurious linear features around the corridor lines. This method is based on the idea that first extracting low level features and generating corridor line hypotheses using these low level features in the image, and then verifying these hypotheses using evidence such as vertical lines and vanishing point information. The proposed procedure is performed in three stages. At low level processing stage, points lying on diagonal lines are selected and vertical lines are extracted. Corridor line hypotheses are generated using the RANSAC algorithm [3] at intermediate stage. At high level processing stage, vanishing point is detected using subtractive clustering algorithm [4], and then information is fed back to search vertical line evidence to confirm or reject corridor hypotheses.

This method has been tested on a large number of real corridor images. The effectiveness and robustness of the proposed method have been validated by experimental results. More details about experimental results and discussions are provided in Chapter 5. Figure 1.2 illustrates the detection results of vanishing point and the corridor lines. The corridor image was grabbed from the video sequence which was captured on the third floor of Thompson Engineering Building at the University of Western Ontario (TEB).

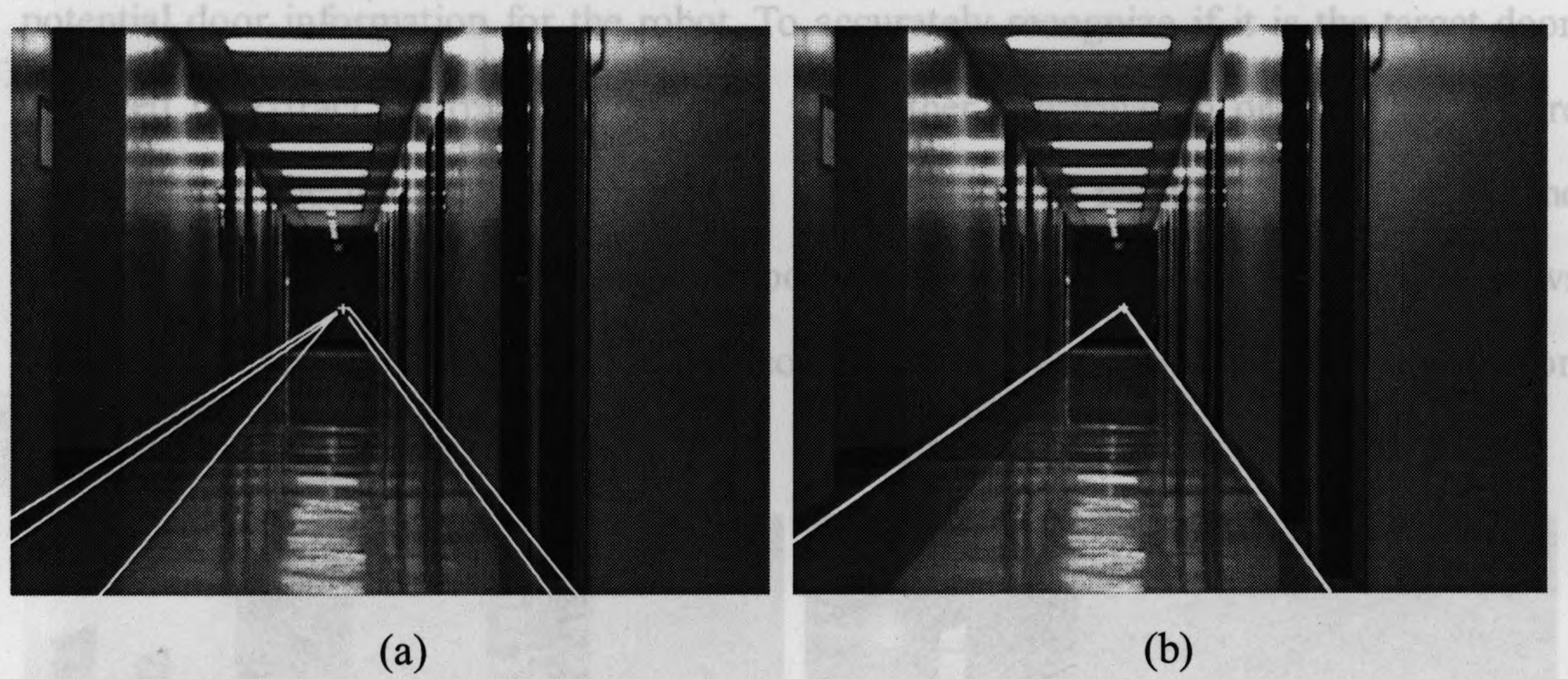
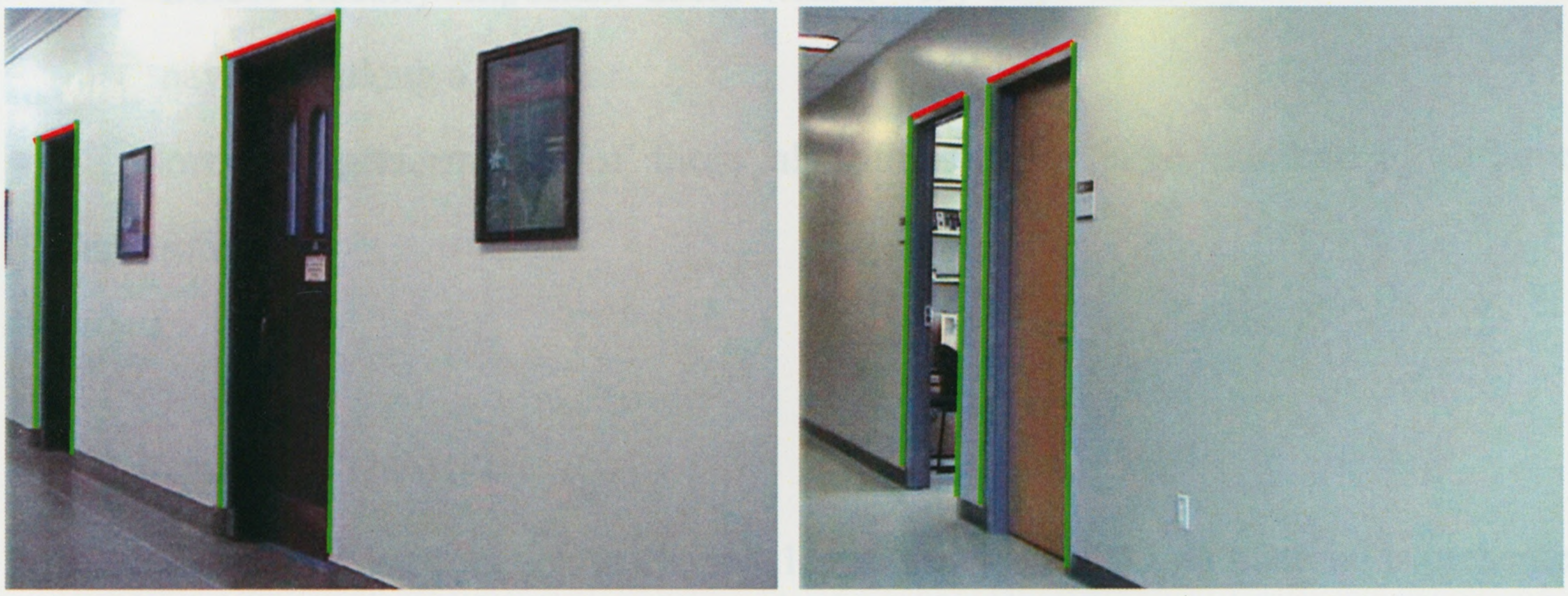

(a) (b)

Figure 1.2 Corridor line location detection results (a) Generated corridor line hypotheses and detected vanishing point (b) Detected corridor lines.

1.2.2 Door Detection Algorithm

The door detection algorithm was developed to detect any potential door structures on left or right walls when a robot moves forward along a corridor. This method is designed in an attempt to reduce the computational cost as much as possible. The basic idea is described as follows: first the vertical lines are grouped to form potential door boundary lines using perceptual grouping rules, and then the information is fed back to search for possible top bars in the window of interest generated by the hypothetical boundary lines. If the top bar is available, then a door hypothesis is generated. The door hypotheses are verified by checking for evidence. This evidence is the presence of at least one of the boundary line of the door hypothesis falling onto the corridor line.

Our algorithm aims to identify visually important doors in the current image frame. The visually important doors are defined as doors which are close to the camera, and can be seen completely (top bars and two boundary lines). We have tested 40 images from different corridor environments. Experimental results demonstrate that the algorithm can effectively detect visually important doors at a very high accuracy rate. Compared with other approaches reported in the literature (Table 5.6), our method has a high detection rate (98.6%) at the cost of a high false alarm (5.71%). The output of our method provides potential door information for the robot. To accurately recognize if it is the target door, we need to extract doorplate information for further confirmation or rejection. Figure 1.3(a) shows the door detection result in the presence of paintings on the wall. The corridor image was captured on the first floor of University College. Figure 1.3(b) shows the detection result of an open door. The corridor image was captured on the third floor, TEB.



(a) (b)

Figure 1.3 Door detection results (a) in the presence of poster and paintings on the wall (b) closed and open doors

1.3 Our Contribution

This dissertation presents two effective and robust algorithms to detect corridor and door structures in an indoor environment. Our main contributions are described as follows:

- Propose a feedback mechanism based hypothesis generation and verification (HGV) method to indoor structure detection applications.
- Implement a corridor detection method using the proposed HGV method that can:
  1. Detect corridor line locations in the presence of many spurious lines in real time.
  2. Obtain vanishing point efficiently and robustly for robot navigation.
  3. Successfully apply the RANSAC algorithm for obtaining corridor hypothesis generation efficiently.
  4. Detect lines efficiently using a modified fast line finder (FLF) algorithm.
- Implement a door detection method using the proposed HGV method that can:
  1. Use feedback scheme to improve the low level processing stage, so as to obtain more robust hypothesis generation and verification. The method can accommodate a wider range of object detection easily by modifying their object models.
  2. Detect visually important doors with a very high accuracy rate when a robot navigates along a corridor.
- Explore the performance of these algorithms in a large number of different environments.

1.4 Thesis Structure

Following the brief introduction in chapter 1, necessary background material is presented in chapter 2. It begins by reviewing the relevant literature on different methods for corridor line detection, vanishing point estimation and door detection applications. It then describes the method and devices for data acquisition in our system.

Chapter 3 begins by describing why straight line segments are chosen as the best primitives available for indoor structure identification, and line extraction problems of video images. Two standard line extraction approaches, Hough Transform and Burns' algorithms, are discussed. The experimental results of these two methods are also compared. This chapter concludes by choosing a Fast Line Finder algorithm (FLF) which is based on Burns' algorithm to extract linear features in the image.

Chapter 4 presents the main contributions of this work. It describes corridor line location detection algorithm and door detection algorithm respectively. The theories, methods of analysis, and implementation details of the proposed algorithms are given.

Chapter 5 presents the experimental results for vanishing point estimation, corridor line location detection and door detection, respectively. Evaluation metric are also developed for quantitatively assessing the accuracy and robustness of the two algorithms.

Finally, in chapter 6, we summarize major contributions of our work and remark on the applicability of our results as well as on future research directions.

Chapter 2
Background Information

2.1 Literature Review

To make an autonomous mobile robot move inside a corridor environment, it is necessary to use a sensory system to extract information about its environment in order to control robot orientation and velocity and successfully navigate while avoiding obstacles. Vision is one of the most powerful human senses. Because of that, a significant research effort in robotics has been dedicated to the use of vision systems for robot navigation. A vision system analyzes the images and produces a *higher level* output, in contrast to the more simple image processing systems, in the form of a description of the scene [5].

2.1.1 Corridor Line and Vanishing Point Detection

For a vision based indoor robot navigation system, it is essential to extract vanishing point and semantically significant line segments for determining the robot orientation and for scene understanding. It is well known that perspective projection maps each set of parallel lines from 3-D space into the set of half lines in the projective plane with a common endpoint called a *vanishing point* [6] [7] [8]. There is a one-to-one correspondence between the points of the projective plane (vanishing points) and directions (sets of parallel lines) in 3-D space. Thus, finding vanishing points which are associated with main directions in the scene becomes an attractive goal because it allows drawing conclusions about its 3-D structure. The determined knowledge of the vanishing point can be used either for inferring the possible direction of advancement [9] [6] [10], or for enhancing further analysis of the scene [11] [1] [12].

Lebegue and Aggarwal [1] [13] proposed an algorithm for detecting and interpreting semantically significant linear features of a monocular images for the navigation of a mobile robot. This method tries to obtain an accurately calibrated system by using inclinometers and a CCD camera, so as to obtain *a priori* of the 3-D directions of corresponding edges. They estimate the position of the vanishing points before they even process the image. Other approaches on using computer vision for robot navigation employed calibrated vision systems to build accurate models of the scene [14] or to detect moving obstacles [15]. However, the calibration process is difficult and the final results are sensitive to calibration errors.

Most research on vision systems for robot navigation has been done without using previous calibration, and the following publications fall into this category. Guerrero and Sagues [16] presented an uncalibrated monocular vision system based on lines for robot navigation. The straight lines are extracted using Burns [17] method and are classified as vertical and non-vertical lines. From non-vertical lines, vanishing points are computed to correct camera orientation. However, using Burns method to extract non-vertical and non-horizontal lines (diagonal lines at any angle) are computationally expensive, and this approach didn't aim to detect corridor lines. Therefore, it is not suitable for our application.

An approach was proposed by Choi *et al.* [18] who described a procedure for estimating lateral position and orientation for the navigation of vision-based wheeled mobile robot. They take the ceiling lamps in a corridor as a landmark. In this method, the angle of the view in the camera is limited. Since this method depends on ceiling lamps in a corridor, it is not a general method that can be applied for a wide variety of corridor environments.

Most methods for vanishing point detection and robot navigation employ generic preprocessing procedures, such as Canny edge detector [19] and Hough Transform [20], for extracting straight line segments [21]. These methods try to find groups of line segments intersecting at a common point of the projective plane and can be further classified with respect to the clustering approach [2].

1) Clustering based on the Hough transform: In this approach, Hough Transform is applied to the set of extracted line segments in order to obtain its

representation in parameter space. The approximate positions of vanishing points are detected by search for local maxima on the set of accumulators [8] [6] [22].

2) Explicit clustering: In this approach, for each pair of line segments a hypothesis is made that it corresponds to a pair of parallel edges in the scene. The resulting hypothetical vanishing points are subjected to an explicit clustering procedure, either on their projections to the Gaussian sphere [11], or in the image plane itself [2] [7]. The centers of the obtained clusters are finally proclaimed as probable vanishing points.

Although these systems operate properly, they are computationally expensive due to the Hough Transform. It is not possible for the Hough Transform to extract even the line features in real time. Also the Hough Transform does not perform well in complex images. Furthermore, these approaches were designed for obtaining the vanishing point and further for other applications, rather than specifically for detecting accurate corridor line locations.

In our project, we aim to detect the true corridor line locations by exploiting the vanishing point information obtained at the intermediate procedure. There are some methods reported in the literature to detect the corridor lines in acquired images.

Li Guan [23] proposed a general program framework for ER1 robot, which consists of sensing, control and motion planning module. In this system, different algorithms are introduced to achieve robust navigation. For direction guidance part, the Hough Transform is performed to get lines having slope between $30^\circ \sim 60^\circ$, $120^\circ \sim 150^\circ$. A mean line segment is calculated as a wall edge for each of the two sloping lines. The intersection of the two edges is the point at infinity, in which direction robot *should* move towards. This approach is simple; however, it is not accurate enough to achieve the true corridor edge. Moreover, Hough Transform doesn't perform well in complicated images.

Rous *et al.* [24] presented an indoor scene analysis technique to extract landmarks, like doors and the floor, for goal-oriented navigation tasks. The detection is based on a priori knowledge of the shape and functionality of searched structures. The core of this method combines region based as well as edge based elements. The segmentation begins with an orientation selective Hough-Transform (OHT) including line segment detection and generates a reticule of convex polygons. Homogeneous polygons with similar color are segmented and merged by a region growing process. Finally a feature extraction and identification is performed to assign regions to known objects. This algorithm works in real-time. However, the approach is applicable for indoor environments with clear linear structures and large homogenous color surfaces. It is not applicable for complex indoor environments with light reflection and shadows.

Vassallo *et al.* [9] proposed a simple and fast method for detecting corridor lines and calculating the vanishing point. This method works well based on the assumption that there is a good contrast between the floor and walls and there is only one such corridor edge. However, in many corridor environments, there are spurious lines around the corridor edge location. Thus, this method is not general for a wide range of corridor environments.

In a typical indoor scene, the corridor environments are often complex with many linear features having similar contrast which are parallel to and close to corridor lines due to edging, patterned tiles and reflective surfaces, etc. No solution so far has been reported in the literature to tackle such situations. With this motivation we developed a procedure for detecting true corridor line locations in the presence of many spurious line features.

The proposed method is based on hypothesis generation and verification (HGV) approach. The HGV is a classical technique for object detection and recognition [25] [26] [27]. The proposed method is performed in three stages consisting of extraction of low level features in the image, the formation of corridor line hypotheses, and finally, verification of the hypotheses using vanishing point and vertical line information in the image. Unlike the previous approaches [2] [21] [23] which extract straight line segments using generic line detection technique, the proposed method generates the hypothetical corridor lines using the RANSAC algorithm [3]. Thus, the corridor line hypotheses and the vanishing point can be achieved in real time. The two most important criteria for a useful vanishing point and corridor detection algorithm are accuracy and speed. The proposed method described in this dissertation meets both criteria.

2.1.2 Door Detection

In the literature, many approaches can be found to detect doors for robotics applications. Some techniques have been reported to detect doors by using the information of ultrasound sensors or laser range finders [28] [29] [30]. A number of approaches combine the sonar data and visual information for detecting doors and robot navigation. A method following this category was proposed in which vision processing provides the system with targets to inspect; and a ring of sonars help navigate the robot through space filled with obstacles as well as confirm or dismiss the hypothesis deduced from vision [31]. Anguelov *et al* [32] proposed a probabilistic framework for detection and modeling of doors from laser range data and omni-directional camera data by capturing shape, color, and motion properties of door and wall objects. In the design proposed by Kortenkamp and Weymouth [33], visual information is added to the sonar information to reduce the ambiguity of places that look identical to sonar sensors. Sonar readings are used to determine gateways in the environment, and then visual cues are extracted from the images in order to distinguish among the different gateways, finally the robot can localize itself in the environment. Approaches like this do not recognize doors as a well-defined entity of the environment, but rather as opening places that mark the transition between one space and another space in the environment [34].

A number of techniques for vision-based door detection in indoor environments have been proposed by many researchers[35] [36] . Kim and Nevatia [37] [38] proposed a method for detecting open doors based on the representation of the door by a generic model. The model is represented by the door's significant surfaces and functional evidence. The line segments are extracted by edge detection and then grouped into door frames consisting of top bar, the left pole and the right pole. The door frames are verified by their functional evidence. Detecting an open door is achieved by detecting objects inside the door since a trinocular stereo system is used for depth determination. The same approach for detecting closed doors does not guarantee the effective recognition of the door since other objects (such as posters and windows) can be ambiguously confused.

Cokal and Erden [39] developed a robot system using image processing and analysis system as well as a neural network (NN) system to recognize open doors and move through the open door with narrow clearance. The output of image processing is used as input data file to a NN based pattern recognition software. The NN determines the existence of a door. Then further process is to interpret the door status by using knowledge based algorithms. The system is developed for a special purpose mobile robot and details on detection rate are not given.

Dedeoglu *et al.* [40] [41] detected closed doors using color information. The perceived colored objects are processed and the door is detected considering its width and height. The door detection rate is 92% with 3% of false positive rate. Details on the color blobs extraction and on how doors are discriminated from other similar objects are not given.

Tomono and Yuta [42] utilize a model-based object recognition method for navigation in an unknown environment, in which the robot navigates itself to a room designated by room number. Doors are recognized using side edges and color based on the door model. Although the method is simple and efficient, the door model is only applicable for the specific corridor environment. It can only detect doors with predefined color.

Nikovski [43] proposed a Memory-based learning scheme for finding a door in a visual scene. This approach has similar learning power to that of a neural network, but has the advantage of not requiring any training phase. The input vectors are panoramic images of the environment. For each new image vector, the algorithm computes its distance from each vector of the training examples to obtain a probability value for determining the existence of a door. Although this method works reliably for the stated purpose, it has problems when there are spurious objects that look very much like doors. Moreover the system has to manage the problem of changes in scale.

Cicirelli *et al.* [34] proposed a method in which the detection of the door has been performed by detecting its most significant components in the image. This method is based on data classification. Two neural classifiers have been trained for recognizing single components of the door. Then a combining algorithm, based on heuristic considerations, checks if they are in the proper geometric configuration of the structure of the door. This system uses together color and shape information for detecting the components of the target. The approach is able to solve the problems of scale changes, perspective variations and partial occlusions. However, this door detection system works well in one office building, and the neural classifiers need to be trained again if the system is applied in another building. Also the method cannot discriminate the objects which have shape and color similar to doors, e.g. posters and lockers etc.

Object detection and recognition has been a challenging problem in computer vision area [44] [45] [46]. A traditional classification of the events in most object recognition system is that of low-, medium-, and high-level image information processing. However, this kind of single-pass strategy of the hypothesis-verify paradigm becomes inadequate as it is easily affected by poor quality data [25]. A feedback control strategy for object recognition was proposed by Mirmehdi *et al.* [25]. In this system, the mechanisms of the traditional image processing systems are improved by introducing control strategies at low, intermediate, and high levels of analysis. They use this feedback paradigm to recognize box shaped objects and bridges from IR images.

For the case of door detection in corridor environment, the quality of the video images acquired by a moving robot are usually poor due to vibration of the moving camera, low light levels and lighting variations. These conditions often affect low level feature extraction. For example, extracting top edge of a door could often fail due to the viewpoint of the camera, reflections as well as the contrast variations between the door frame and the wall. However, none of the above proposed algorithms mentioned this practical issue nor dealt with such situation. In our proposed method, we employ a feedback control strategy similar to the one proposed by Mirmehdi *et al.* [25] to improve the performance of the low level processing, so as to obtain good hypothesis generation at medium level stage. Based on the feedback control mechanism, we develop a simple and effective method for detecting doors when a robot moves along a corridor using a single video camera.

2.2 Image Acquisition

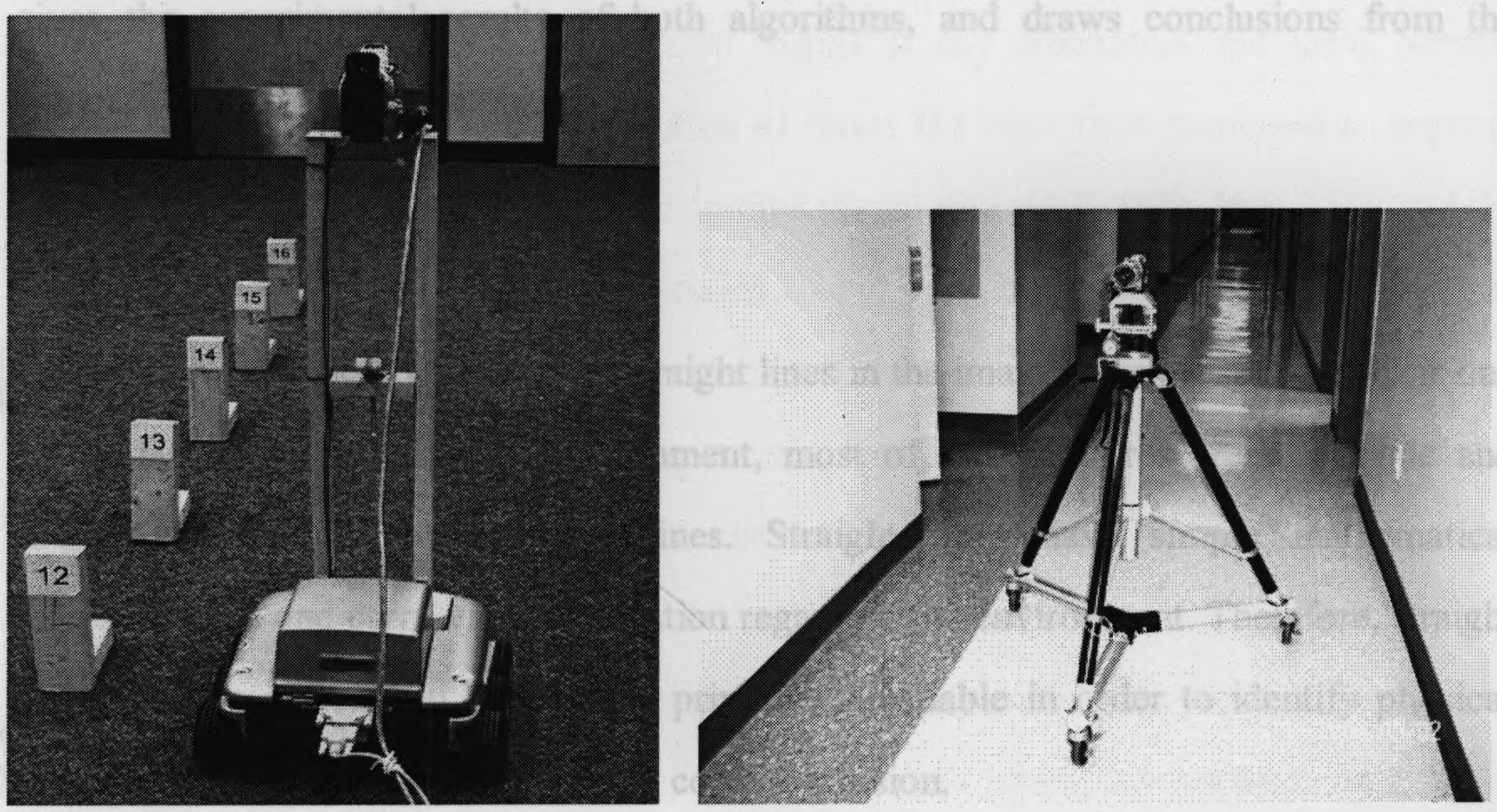
The image sources used by the proposed method are video images rather than still images. The video sequences were acquired using following two methods:

(1) A camera was mounted on the top of a mobile robot traversing in typical indoor environments (Figure 2.1). The robot platform used was Koala, a medium sized mobile robot manufactured by K-Team, Switzerland. The robot was controlled using a personal computer using an RS232 connection. The camera was Canon ZR50MC digital camcorder with resolution 640×480 pixels connected to the computer using an IEEE1394 (Fire wire) cable.

(2) A home video camera was mounted on a rolling tripod (Figure 2.2) pushed by a person along a corridor. The camera model is JVC GR-DV 4000. JVC GR-DV digital camera appears to perform well with both high and low light levels compared to Canon ZR50MC.

We have taken corridor video sequences in Thompson Engineering Building (TEB) and the Spencer Engineering Building (SEB) using the first method, and other video sequences from 50 different corridor environments inside 20 buildings on UWO campus using the second method. The video sequences differ from each other due to the environment in which the camera was placed, the moving speed and the field of view of the camera and time of the day.

Video images acquired using both methods were analyzed in the implementation. The video sequences captured by the first method are much noisier than those by using the second method.



(a) (b)

Figure 2.1 Image acquisitions (a) the Canon camera mounted on the top of the robot (b) the JVC camera mounted on a rolling tripod

Chapter 3
Line Detection Algorithms

Linear feature extraction in an image is an important part at the low level processing stage in our proposed method. This chapter first reviews a number of existing line extraction algorithms. The well known Hough transform and a fast line finder algorithm which is based on the Burns' method are then introduced. The last section of the chapter gives the experimental results of both algorithms, and draws conclusions from the comparison of these two methods.

3.1 Introduction

In our proposed method, we exploit straight lines in the image as key features. This is due to the fact that in an indoor environment, most of the structures are man-made and structure boundaries are straight lines. Straight lines have simple mathematical representations and carry vital information regarding the environment. Therefore, straight line segments are chosen as the best primitives available in order to identify physical structures in indoor environment using computer vision.

The extraction of line segments is one of the most fundamental problems in the field of digital image processing [47]. Over decades, several approaches have been reported in literature and are broadly classified into four categories [48]: pixel connectivity-edge linking based [49], Hough transform (HT) based [20], Burns method based [17], and statistical based method [50].

Nevatia and Babu [49] proposed a line extraction approach based on edge linking and segmentation. The basic idea is to find out local edge pixels, link the found pixels into contours on the basis of proximity and orientation, and then segment the contours into relatively straight line pieces. Other examples based on pixel connectivity-edge linking include work proposed by Zhou *et al* [51], and Nalwa and Pauchon [52]. The main advantage of this kind of method is that connectivity among all the pixels which are identified as linear edge pixels is very much ensured. The main problem of edge linking approaches is that they are sensitive to the output of the edge finder. They tend to be unstable in the presence of clutter, and have trouble bridging gaps.

One powerful approach for straight line detection is the Hough transform and its variants [53] [54]. The original Hough transform technique implemented by Duda and Hart is nowadays called standard Hough transform (SHT). Ballard [55] introduced the Generalized HT that could find arbitrary shapes of any orientation and scale. Several other modifications of the main algorithm of direct HT have been proposed to improve the accuracy [56], [57], [58], [59], the computational time [60], [61], [62], [63], and the memory requirements [64], [65]. Additional analysis has been made for the quantization of the HT space [66], [67] and Bayesian probabilistic scheme based HT [68]. Two surveys of the HT problems and techniques are given in [53], [54].

A third method of line detection due to Burns *et al* [17] utilizes the gradient orientation as the initial organizing criterion to group pixels into a set of line support regions, and then the structure of the associated intensity surface is used to determine the location and properties of the edge. This method can detect low-contrast long linear features in the intensity image, but it is computationally expensive. The work of Burns *et al* was further improved by Nelson [69] who utilizes energy minimization in a landscape derived from spatially extended operators to extract linear features.

Finally, there are statistical based approaches. Mansouri *et al* [50] proposed a hypothesize-and-test algorithm to find line segments of a given length by hypothesizing their existence based on local information, and attempting to verify that hypothesis statistically on the basis of a digital model of an ideal segment edge. This method has some similarities to the approach proposed by Nelson [69], but does not provide an efficient method to fit maximal segments.

Based on the fact that so many algorithms for line extraction have been reported, we wanted to use one of these mature techniques rather than develop a new line extraction algorithm for low level linear feature extraction in our application. We have tested the Hough transform and Burns' method to extract straight line segments in indoor video images.

3.2 Hough Transform Method

The Hough transform (HT) has been recognized as one of the most popular methods to extract parametrized lines from an image [53]. The standard equation for a line in slope-intercept form is:

$$y = ax + b \tag{3.1}$$

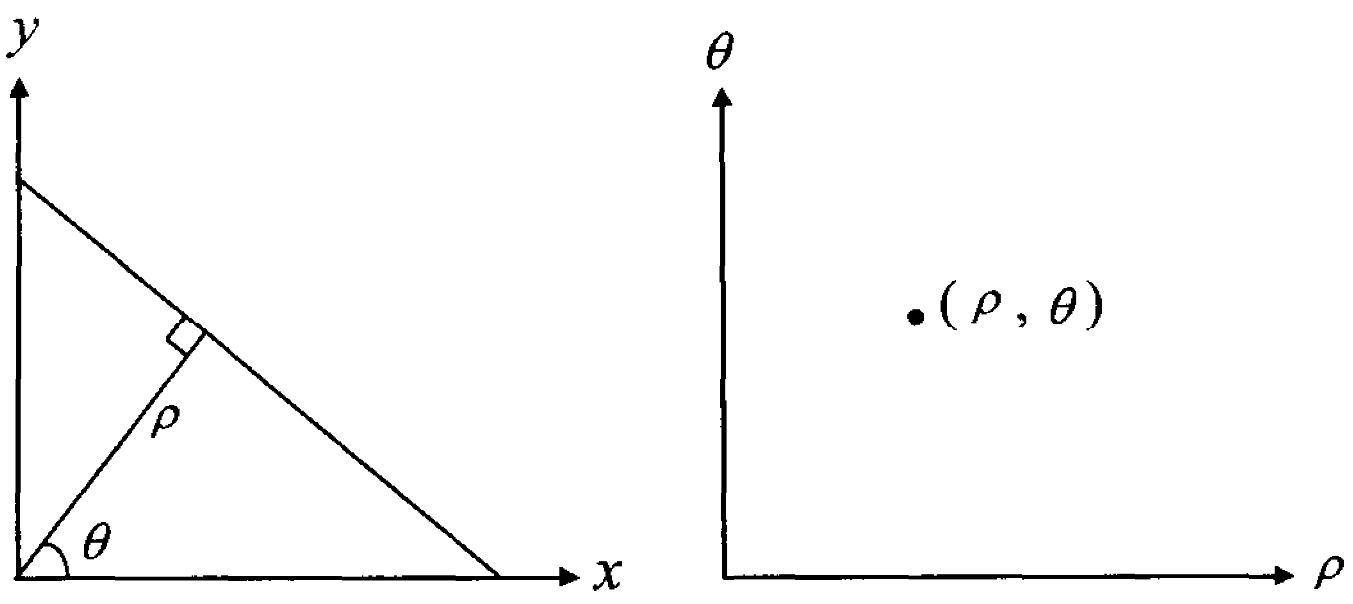
where a is the slope of the line and b specifies where the line intercepts the y-axis. A problem with using this equation to represent a line is that the slope approaches infinity as the line approaches the vertical direction. Duda and Hart [20] suggested using the following equation to define a line:

$$\rho = x\cos\theta + y\sin\theta \tag{3.2}$$

where ρ is the perpendicular distance from the line to the origin, and θ is the angle that the perpendicular line makes with the x - axis.

The HT maps a line in the image space (x,y) into a single point in the (ρ,θ) parameter space (Figure 3.1). This fact can be used to detect straight lines in a given set of boundary points. Suppose we are given boundary points $(x_i, y_i), i = 1, \ldots, N$ for a line. For some chosen quantized values of parameters ρ and θ, map each (x_i, y_i) into the (ρ, θ) parameter space and count $C(\rho, \theta)$, the number of edge points that map into the location (ρ, θ) [70]:

$$C(\rho_k, \theta_l) = C(\rho_k, \theta_l) + 1, \quad \textit{if} \quad x_i\cos\theta + y_i\sin\theta = \rho_k \quad \textit{for} \quad \theta = \theta_l \tag{3.3}$$



(a) (b)

Figure 3.1:The Hough transform (a) A line in the image (b) The corresponding point in parameter space

Then the local maxima of $C(\rho, \theta)$ give the different straight line segments through the edge points; that is, to extract the lines, one just needs to find these peaks in the parameter space.

3.3 A Fast Line Finder Algorithm

Kahn *et al.* [71] developed a fast line finder (FLF) method which is based upon a gradient-based and region-based line extraction algorithm first developed by Burns *et al* [17]. The FLF algorithm remains in the spirit of [17], but it differs significantly in the way pixels are processed, lines are fitted, and its inherent time performance. The algorithm is modified to run much faster (e.g., by an order of magnitude) with respect to the Burns' method [17]. The FLF algorithm has four basic steps. The outlines of implementation procedure are described as follows:

A. Computing Gradient Direction and Magnitude

The first stage computes the direction and magnitude of the image intensity gradient at each pixel. Image derivatives can be implemented for an entire image by using convolution masks (e.g., Prewitt, Sobel) [72] [73]. Figure 3.2 shows the Prewitt and Sobel masks for computing the image derivatives.

-1	-1	-1
0	0	0
1	1	1

(a)

-1	0	1
-1	0	1
-1	0	1

(b)

Prewitt

-1	-2	-1
0	0	0
1	2	1

(c)

-1	0	1
-2	0	2
-1	0	1

(d)

Sobel

Figure 3.2: 3×3 convolution masks (a) Prewitt mask for computing gradient in y direction (b) Prewitt mask for computing gradient in x direction (c) Sobel mask for computing gradient in y direction (d) Sobel mask for computing gradient in x direction

The gradient magnitude can be approximated by Eq. 3.4 for reducing computational cost and resources:

$$m \approx |I_x| + |I_y| \qquad (3.4)$$

where m is the gradient magnitude, I_x and I_y are gradients in x direction and y directions respectively.

B. Coarse Quantization of Gradient Directions into Buckets

At this stage, pixels are coarsely quantized into one of a fixed number of "buckets" based upon gradient direction as shown in Figure 3.3 (e.g., gradient directions from $15\pi/8$ to $\pi/8$ are classified into bucket 1).

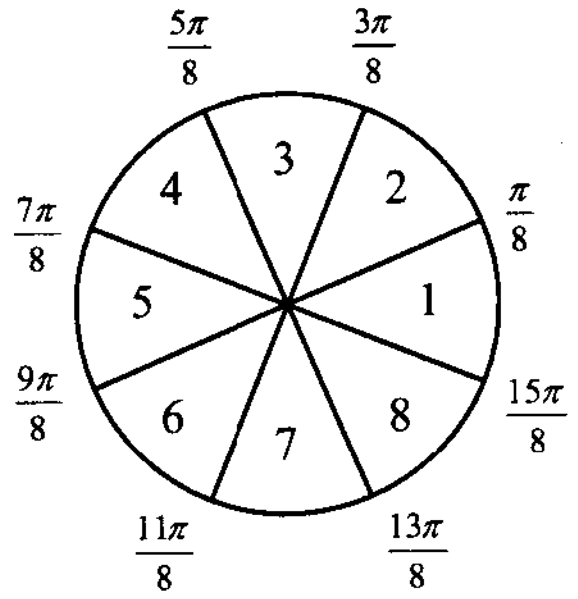


Figure 3.3: A coarse quantization of gradient direction space for assigning bucket labels to pixels

The gradient direction is given by Eq. 3.5:

$$\alpha = \arctan(I_y / I_x) \tag{3.5}$$

This approach is computationally expensive because of functions such as division and arctangent to compute the angular measure. For efficient implementation, a two-dimensional lookup table may be used in which I_x and I_y at each relevant pixel index directly to the associated bucket label. More details may be found in [71].

C. Connected Components Algorithm (CCA) to Form Line Support Regions

A connected components algorithm (CCA) [74] groups adjacent pixels with identical bucket labels into line support regions. Final line lengths can be restricted by filtering line support regions which contain less than a minimum number of pixels.

D. Fitting Lines to Line Support Regions

The principal axis of a line support region provides a good line fit and it requires less computational time than the method used by Burns *et al.* [17]. The principal axis can be determined from the eigenvalues computed from a scatter matrix whose quadratic solution is:

$$\left[\frac{a+c}{2} \pm \sqrt{\frac{(a-c)^2}{4} + b^2}\right] \tag{3.6}$$

where

$$a = \left[\sum wx^2 - \frac{(\sum wx)^2}{\sum w}\right], \qquad b = \left[\sum wxy - \frac{\sum wx \sum wy}{\sum w}\right],$$

$$c = \left[\sum wy^2 - \frac{(\sum wy)^2}{\sum w}\right],$$

w is an optional gradient magnitude weighting factor, and (x, y) are the region pixel coordinates. The small eigenvalue V_S and the large eigenvalue V_L are obtained by

Eq.3.7:

$$V_S = \left[\frac{a+c}{2} - \sqrt{\frac{(a-c)^2}{4} + b^2}\right], \qquad V_L = \left[\frac{a+c}{2} + \sqrt{\frac{(a-c)^2}{4} + b^2}\right] \qquad (3.7)$$

The orientation of the best fitting line is $\vec{V}_S = \arctan(\frac{V_S - a}{b}) + \frac{\pi}{2}$.

The ratio=$\sqrt{V_S}/\sqrt{V_L}$ provides an extremely useful measure of the straightness of the line support region about the fitted line. Small ratios are more "line-like" and large ratios are more circular. The line endpoints are determined by intersecting the principal axis with an upright box bounding its support region.

3.4 Experimental Results and Discussion

In order to choose an appropriate method to extract straight line segments in real indoor images, we tested both the standard Hough transform (SHT) and the FLF algorithm for our application. Testing images are image frames grabbed from video sequences captured by a moving camera. The image size is 640×480 pixels.

3.4.1 Standard Hough Transform

The Hough transform designates a rectangular tile in the (ρ, θ) parameter space as a line. The dimensions of such a tile are $\Delta\rho$ (ρ resolution) and $\Delta\theta$ (θ resolution). We can also use a minimum length criterion $l > l_{\min}$ to filter lines.

Result of applying standard HT is shown in Figure 3.4. In our testing, the $\Delta\rho$ of 10, the $\Delta\theta$ of 20, and the $l_{\min}$ of 35 are used to detect lines. The image is acquired on the third floor of TEB.

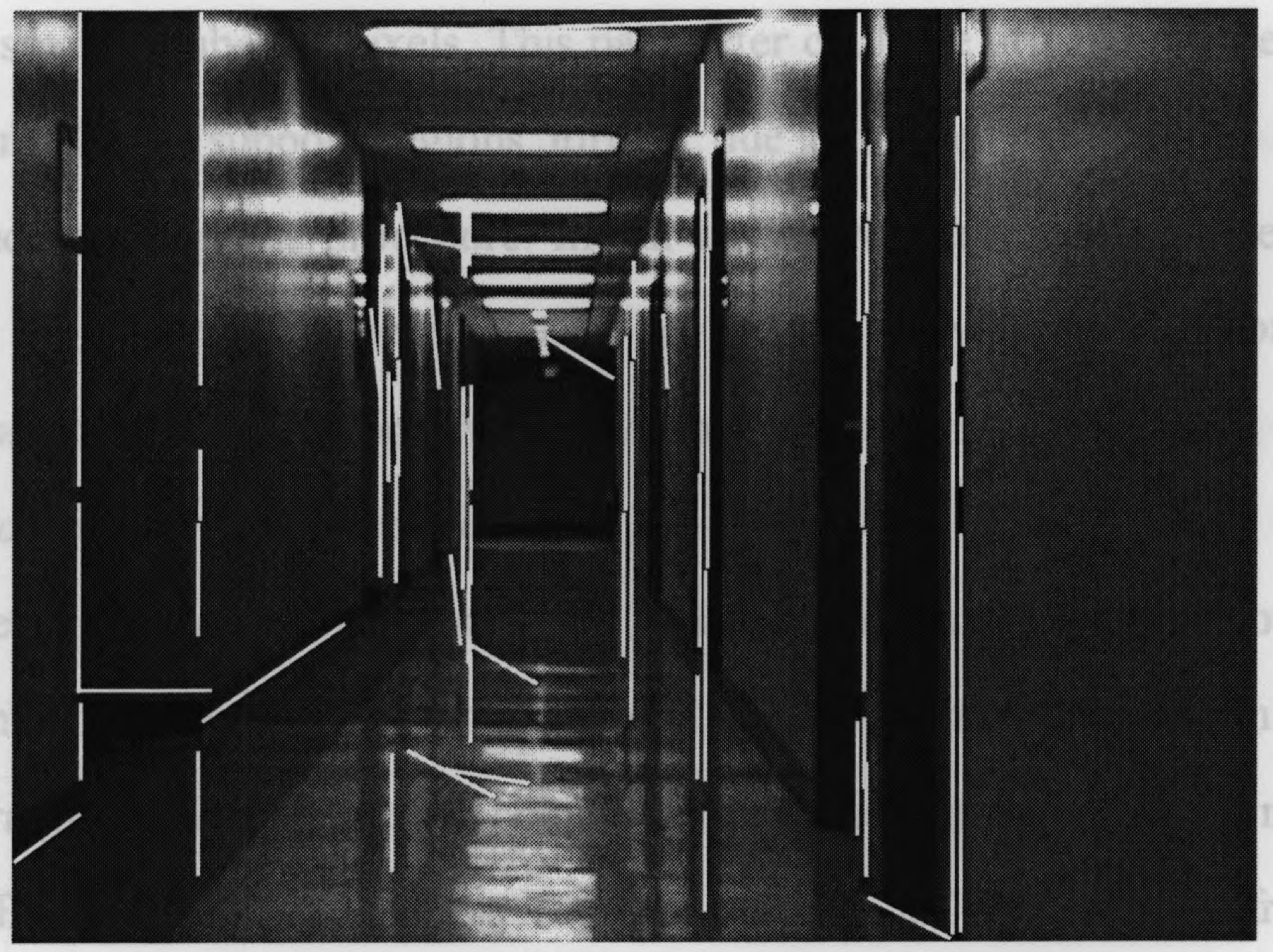


Figure 3.4: Result of applying the standard Hough transform

3.4.2 The FLF Algorithm

When implementing the FLF algorithm, we exploit the following parameters to avoid extracting unnecessary lines from an image and speed up the processing:

A. Gradient Magnitude Threshold

The gradient magnitude threshold is one of the most important parameters which control the amount of overall computation. An appropriate threshold can exclude a large number of pixels from further processing while avoiding significant region fragmentation. In our testing, the gradient magnitude threshold is set at 18.

B. Eigenvalue ratio

There are many reflections and lighting variations in real indoor images. The line support regions formed due to reflections are usually more circular than "line-like". Thus, the ratio $\sqrt{V_S}/\sqrt{V_L}$ is used to remove "circular-like" noisy regions while keeping line support regions with good straightness. In our testing, this parameter is set at 0.25.

C. Region Pixel Count Threshold

The region pixel count threshold is used to eliminate the line support regions containing a small number of pixels. This parameter could be utilized to filter out a large number of small line support regions formed due to noise, extract long straight line segments, and speed up the procedure. In our testing, we are mainly interested in long straight line segments. This parameter is set to180 for vertical and horizontal line segment extraction, and 70 for lines with any other orientations.

D. Orientation Bucket and Direction

When the directional buckets are tuned and centered on the expected orientation of the line segment, the tuning can substantially reduce overall computation and largely avoid line fragmentation at bucket boundaries and the extreme cost of multiple line extractions. Burns *et al* [17] discuss bucket size and bucket boundary fragmentation issues.

Figure 3.5 shows the line extraction result when applying the FLF algorithm.

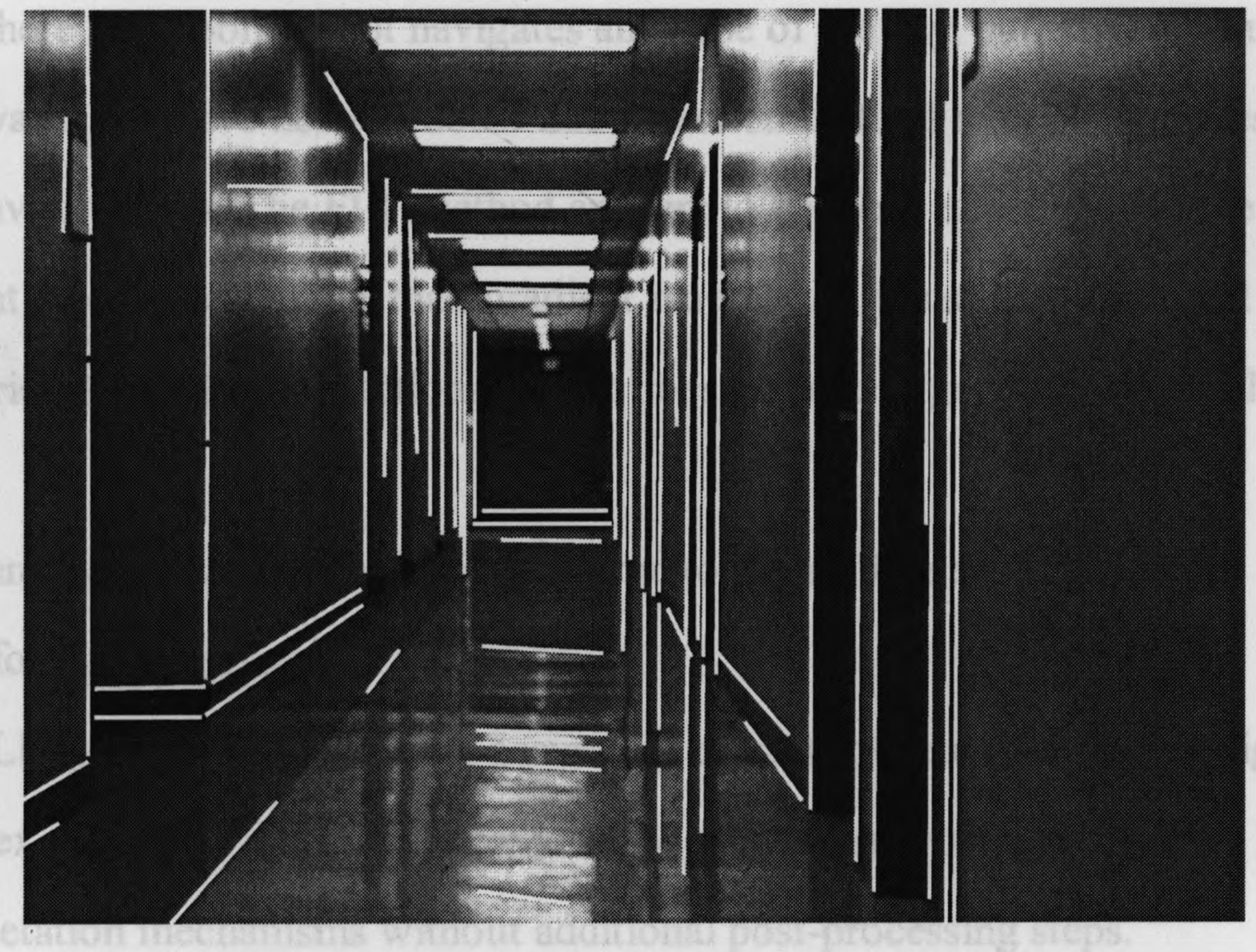


Figure 3.5: Result of applying the FLF algorithm

3.4.3 Comparison and Discussion

We test both SHT and the FLF methods on a number of real indoor images. The performance comparison of these two methods is evaluated based on a qualitative comparison of a few images and shows that the FLF achieves better performance than the SHT (see Figure 3.4 and 3.5).

The standard Hough transform (SHT) cannot obtain good line extraction results for indoor video images in our application. The SHT consists of two stages, an edge detection stage and a line detection stage using the edge detection result. There are several problems with this approach. Firstly, the initial edge detection stage is sensitive to noise. Secondly, the second stage does not use all the information available in the image and therefore incorrect decisions made by the first stage cannot be corrected in the second stage. The SHT also has the problems of computational complexity, coarse resolution and lack of locality. Extensive post processing is often needed.

Usually in indoor environments, there are not only illumination changes depending on areas where the mobile robot navigates and time of the day, but also spurious lines and edges on walls. Thus we need a robust line detection method to cope with these complex imaging environments. The FLF method appears to be more effective than the SHT since the gradient orientation (rather than gradient magnitude) is used as the initial organizing criterion prior to the line extractions, and it does not use edge detection result to extract lines.

In summary, we choose the FLF algorithm to extract line segments in our application due to the following reasons:

1. The FLF is very robust and can accurately extract low-contrast long straight lines in complex images. Thus, it provides edge/line information to the high level interpretation mechanisms without additional post-processing steps.
2. The FLF method can achieve very good performance for the line extraction when prior knowledge of the desired line orientation is available.
3. The FLF can extract parallel, close lines which are formed due to different brightness change.

4. The problem of the FLF is computational cost. However, this can be reduced greatly if orientations of the lines of interest are given.

Chapter 4
Methodology for Indoor Structure Detection

In this chapter we discuss our main contribution, namely, methodology for indoor structure detection. As mentioned in Chapter 1, corridors and doors are the important structures in indoor scenes. We describe two algorithms to detect corridor lines and doors, respectively.

Both algorithms are developed under the assumption that the knowledge about the vertical direction in the scene is available. This constraint is reasonable since it is similar to human vision in a natural way. In addition, even human beings get disoriented when placed in a tilted environment.

Since high-level image understanding algorithms in our implementation are looking for line segments of particular orientations in 3-D, we designed our lower level image-processing stages to take advantage of the prior knowledge of vertical direction. By reasoning in terms of what features the high level interpretation stage will actually use, we were able to design lower level algorithms for better and faster processing. This kind of top-down information can benefit the feature extraction stage by reducing the amount of unwanted features, increasing the sensitivity to good features, and drastically speeding up the computation.

4.1 Corridor Line Location Detection Algorithm

4.1.1 System Overview

As discussed in Chapter 2, corridor line location detection algorithm is developed for detecting true corridor line locations in the presence of many spurious linear features around the corridor lines. The system diagram of the proposed method is shown in Figure 4.1. We utilize a hypothesis generation and verification (HGV) method to detect true corridor locations using low level line features. The system consists of low level, intermediate level and high level processing stages. The low level processing step extracts low level features; the intermediate level step generates corridor line hypotheses through fitting line models using the RANSAC algorithm; and at the high level processing step, the system is directed to search for evidence in an attempt to confirm or reject the hypotheses. The flow chart of the proposed algorithm is shown in Figure 4.2.

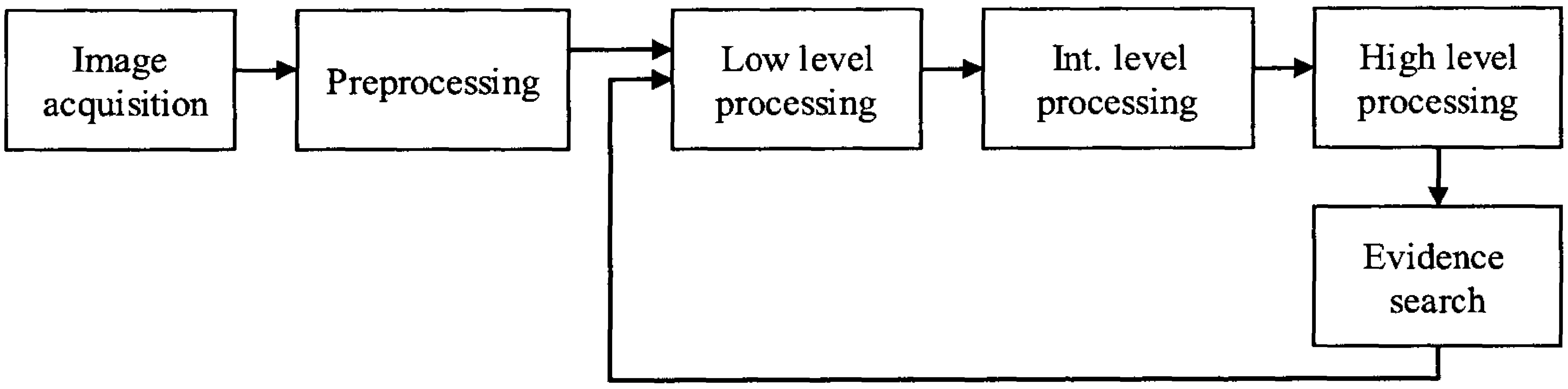


Figure 4.1 Block diagram of corridor line detection system

4.1.2 Image Acquisition

As mentioned in Chapter 2, the video sequences were acquired using a camera mounted on a mobile robot (or a rolling tripod) traversing in typical indoor environments. The optical axis of the camera is basically parallel to the longitudinal corridor axis. The proposed approach is applicable for the cases where autonomous robot is located at any position in the corridor (see Figure 4.3): close to the left side of the wall, in the middle of the corridor, or close to the right side of the wall.

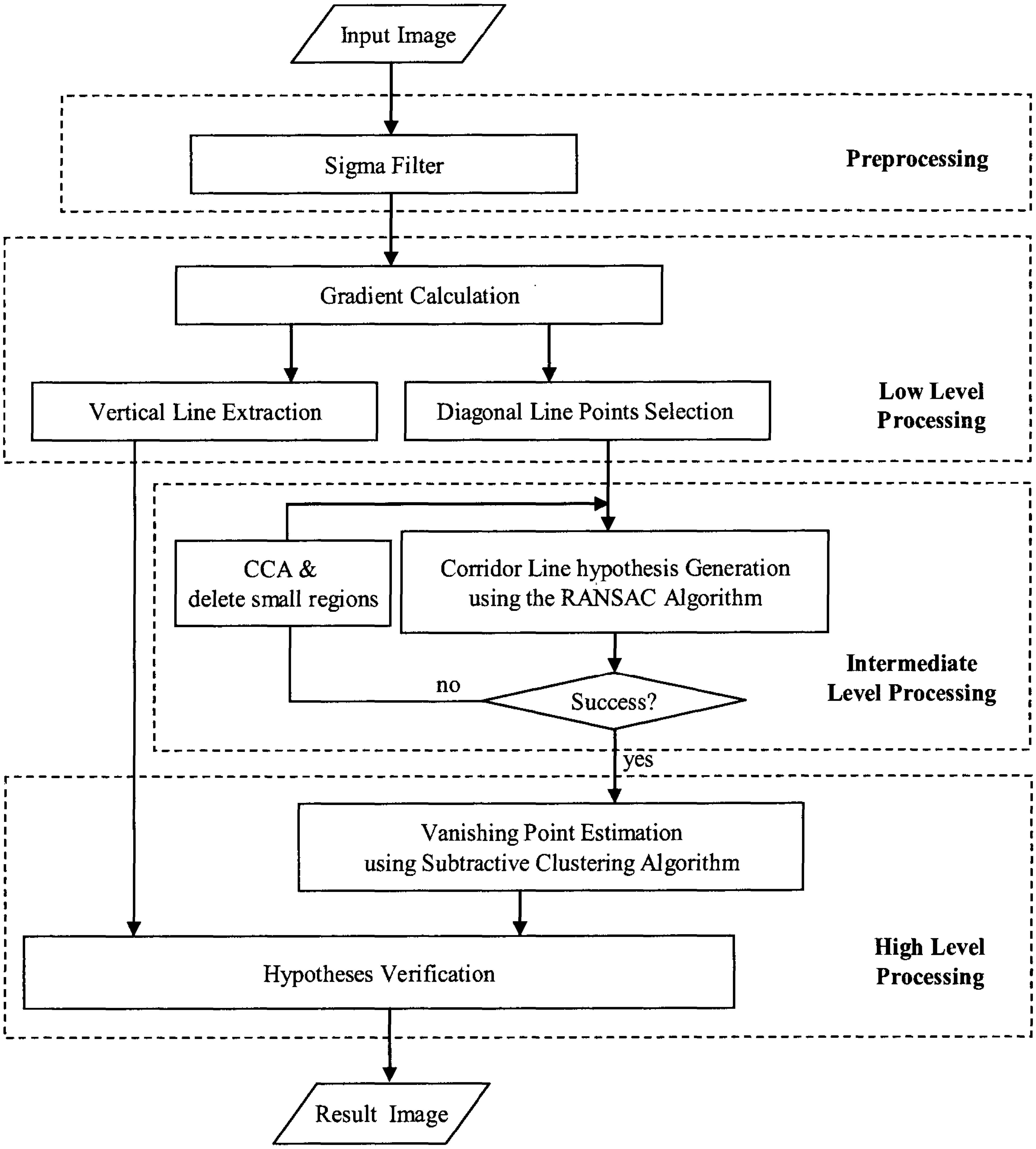


Figure 4.2: Corridor detection algorithm flow chart

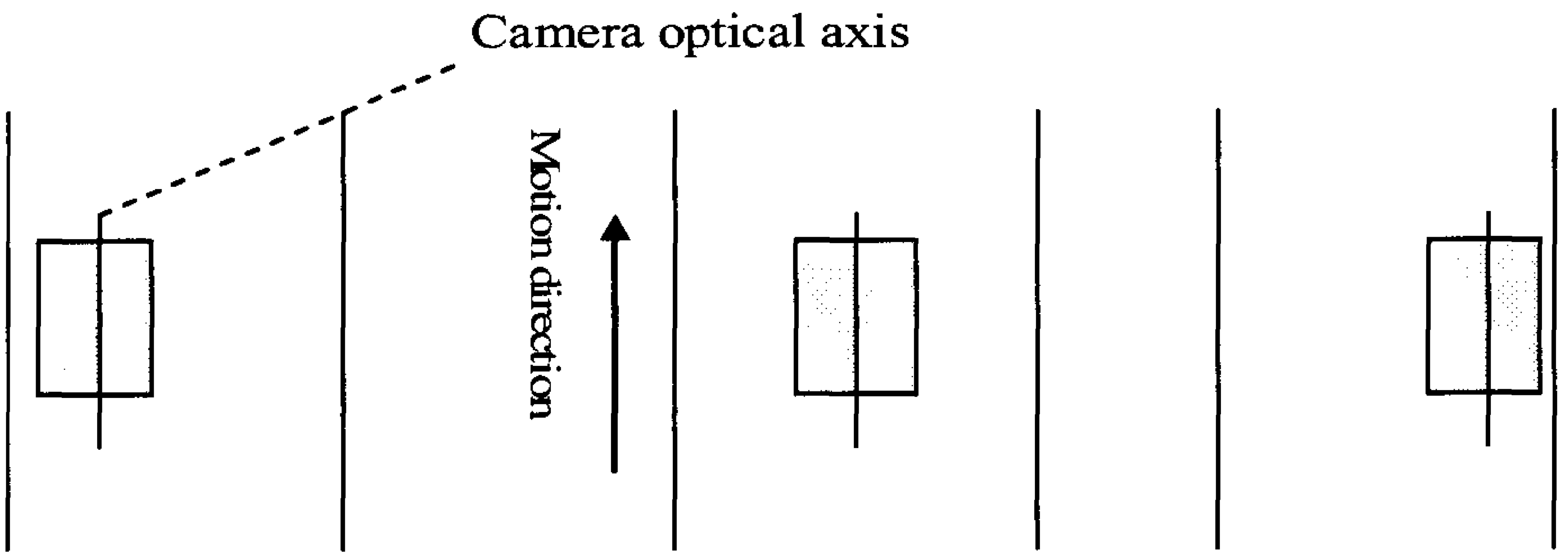


Figure 4.3: Different cases of robot location inside a corridor.

4.1.3 Preprocessing

The quality of the video images grabbed from the video sequences is poor due to the vibration of the motion robot. High noise level is a frequent problem due to low level light. To suppress noise while preserving sharp edges, Lee's sigma filter [75] is applied to the input image as the preprocessing step. The sigma filter takes an average of only those neighboring pixels whose values lie within 2σ of the central pixel value, where σ is the sigma parameter found by trial and error for the image.

Let $x_{i,j}$ be the intensity or gray level of pixel (i, j), $\hat{x}_{i,j}$ be the smoothed pixel (i, j), and K be a prespecified value. The sigma filter procedure is described as follows:

(1) Establish an intensity range ($x_{i,j} + \Delta, x_{i,j} - \Delta$), where $\Delta = 2\sigma$.

(2) Calculate M: the number of pixels whose intensity values lie within the intensity range in a $(2n+1, 2m+1)$ window.

(3) Then $\hat{x}_{i,j}$ = two-sigma average, if $M > K$

=immediate neighbor average, if $M \le K$.

The value of K should be carefully chosen to remove isolated spot noise without destroying thin features and subtle details. For a 7×7 window, K should be less than 4, and it should be less than 3 for a 5×5 window [75].

The two-sigma average is obtained by Eq.4.1:

$$\delta_{k,l} = 1, \quad \text{if } (x_{i,j} - \Delta) \le x_{k,l} \le (x_{i,j} + \Delta)$$

$$= 0, \quad \text{otherwise.}$$

Then

$$\hat{x}_{i,j} = \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} \delta_{k,l}\, x_{k,l} \Big/ \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} \delta_{k,l} \tag{4.1}$$

And immediate neighbor average is calculated by Eq.4.2:

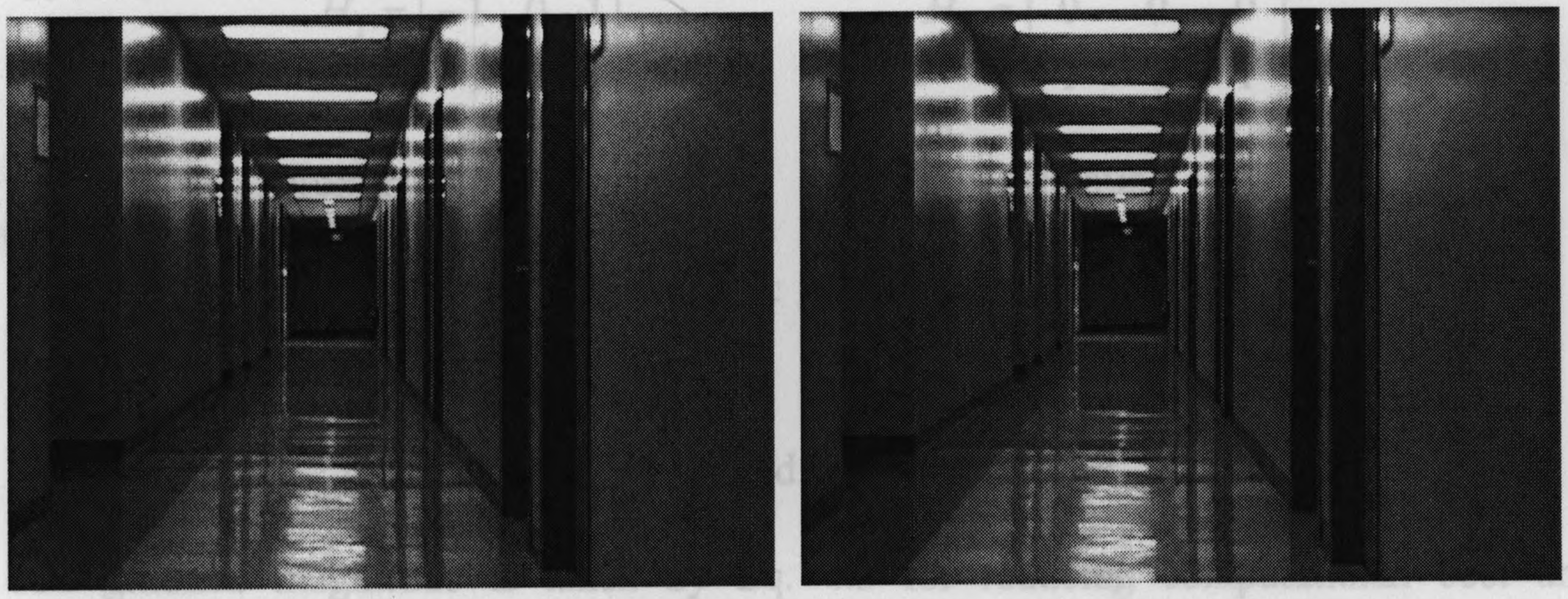
$$\delta_{k,l} = 1,$$

Then

$$\hat{x}_{i,j} = \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} \delta_{k,l}\, x_{k,l} \Big/ \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} \delta_{k,l} \tag{4.2}$$

where $(2n+1, 2m+1)$ is the window size of the sigma filter.

As an illustration, Figure 4.4(a) shows an original corridor image. The result of applying the 7×7 sigma filter is shown in figure 4.4(b).


(a)

(b)

Figure 4.4 The result of the 7×7 sigma filter when applied to a corridor video image.(a) Input test image; (b) The output of the sigma filter.

4.1.4 Low Level Processing Stage

The low level processing step extracts low level features from input images. There are two main procedures in this stage: one is selection of points belonging to diagonal lines and the other is vertical line extraction. Both procedures need to employ the image gradient information. Thus, image derivatives are computed first.

A. Gradient Calculation

Image derivatives are computed by convolving the image with a gradient operator. Let H denote a $p \times p$ mask and define for a digital image U and their inner product at the location (m, n) as the correlation

$$\begin{aligned}\langle U,H \rangle_{m,n} &= \sum_i \sum_j h(i,j)u(i+m,j+n) \qquad (4.1)\\ &= u(m,n) * h(-m,-n)\end{aligned}$$

Here, 3×3 Prewitt convolution masks (Figure 4.5) are used to calculate the image gradient in X and Y directions denoted by I_x and I_y.

$$H_1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad H_2 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

(a)approximating I_x gradient (b)approximating I_y gradient

Figure 4.5: Prewitt filter

So, $$I_x(m,n) = \langle U,H_1 \rangle_{m,n} \text{ and } I_y(m,n) = \langle U,H_2 \rangle_{m,n} \qquad (4.2)$$

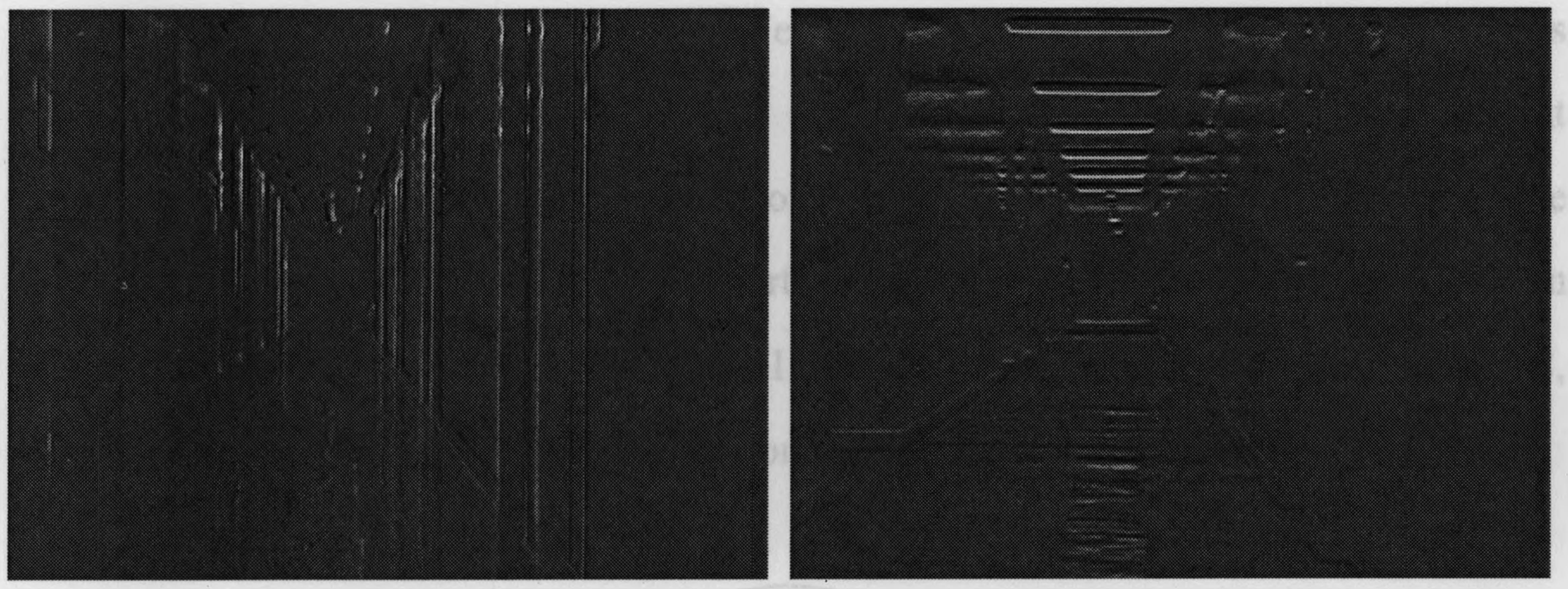
The gradient magnitude is taken by Eq. 4.3 for reducing computational cost and resources:

$$mag(m,n) \approx |I_x(m,n)| + |I_y(m,n)| \qquad (4.3)$$

And the gradient direction is given by

$$\theta(m,n) = \arctan(I_y(m,n)/I_x(m,n)) \qquad (4.4)$$

Figure 4.6 illustrate the gradient in x and y directions of the test image (Figure 4.4(a)).



(a) (b)

Figure 4.6: Gradient of the image. (a)I_x: gradient in x direction; (b)I_y: gradient in y direction

B. Vertical Line Extraction

As discussed in Chapter 3, we choose a fast line finder (FLF) algorithm [71] to extract vertical lines in the image. This is due to the fact that it is able to extract low contrast long straight lines without post processing. The FLF is a gradient orientation-based and region-based line extraction algorithm first developed by Burns *et al.* [17]. The default eight buckets (a set of ranges) for coarse quantization of gradient direction space are shown in Figure 4.7.

To extract vertical straight lines in the image, only pixels whose gradient direction fall into bucket 1 and bucket 5 need to be processed. More details about line extraction can be found in Chapter 3. In order to speed up the line extraction procedure, we make some improvement on the FLF algorithm by adding an erosion operation after the quantization of gradient directions step. A flat vertical linear structuring element is used to remove tiny regions before the Connected Component Algorithm (CCA) step. After employing erosion operation, many isolated edge segments are eliminated, so that the number of line support regions formed by the CCA is substantially reduced. Thus, great speedups can be achieved. The structuring element should be carefully chosen since it is directly related to the computational cost and line quality. A big structuring element would substantially reduce the computational time; however it would affect the line quality. In our application, we choose a vertical structuring element with length 3 for an image size of 640×480 pixels. The vertical line extraction result is shown in Figure 4.8, where the detected lines are superimposed on the original grayscale image.

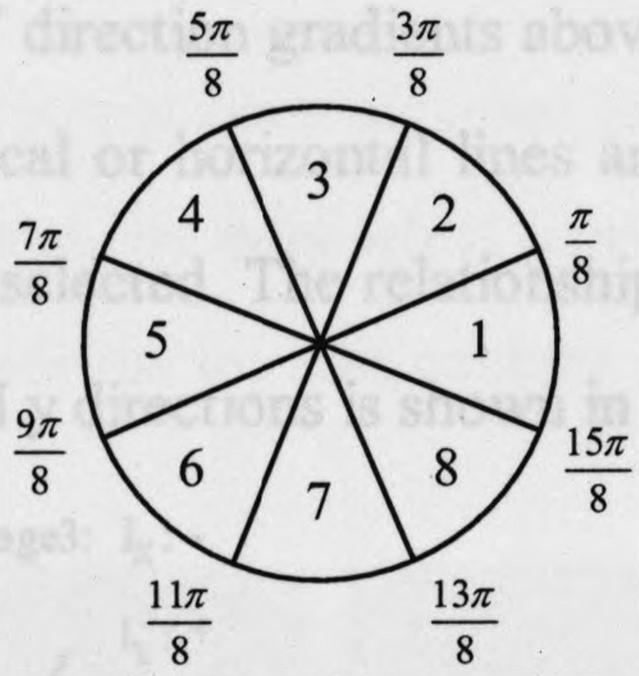


Figure 4.7: The default eight buckets for coarse quantization of gradient direction space

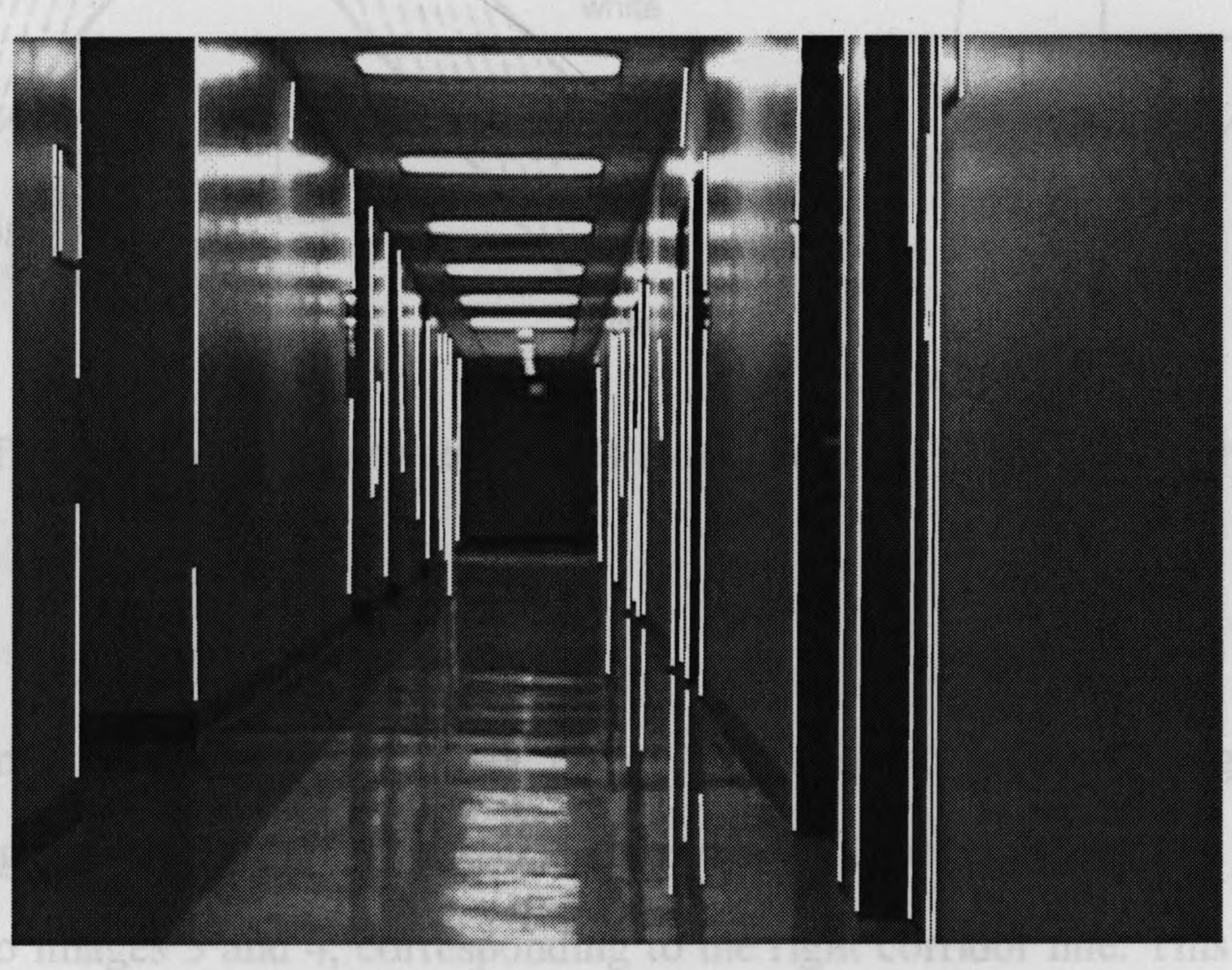


Figure 4.8: Vertical line extraction results

C. Selection of Points belonging to Diagonal Lines

This step aims to select points of interest, that is, points lying on the corridor lines in the image, and then separate them into four binary images. In order to suppress outliers in the selected point set, erosion operation is then applied on these four binary images. The method is implemented based on an assumption that there is relative contrast between the floor and walls.

Points along corridor lines (points along diagonal lines) can be selected by choosing pixels having X direction and Y direction gradients above a certain threshold value. This way, edge points lying on vertical or horizontal lines are discarded and only the points belonging to diagonal lines are selected. The relationship between points along diagonal lines and their gradients in x and y directions is shown in Figure 4.9.

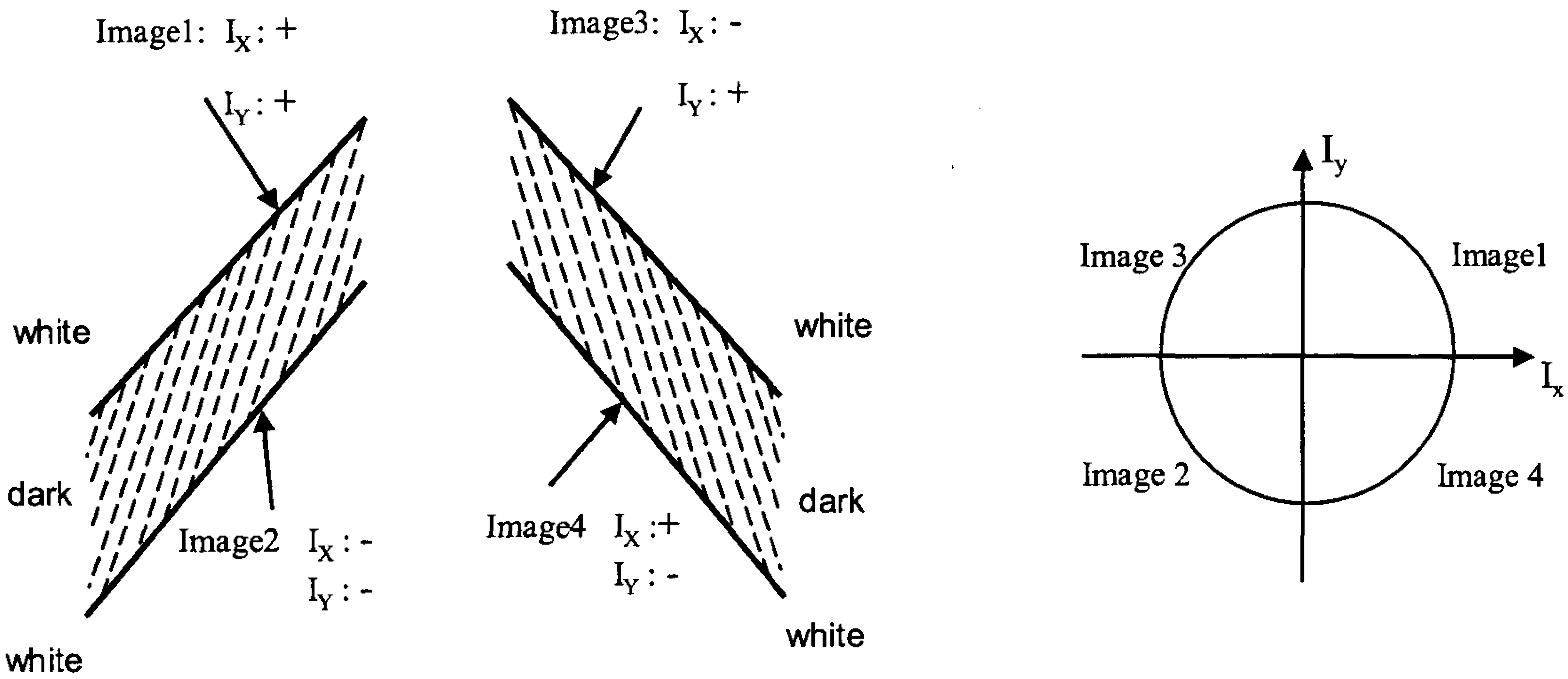


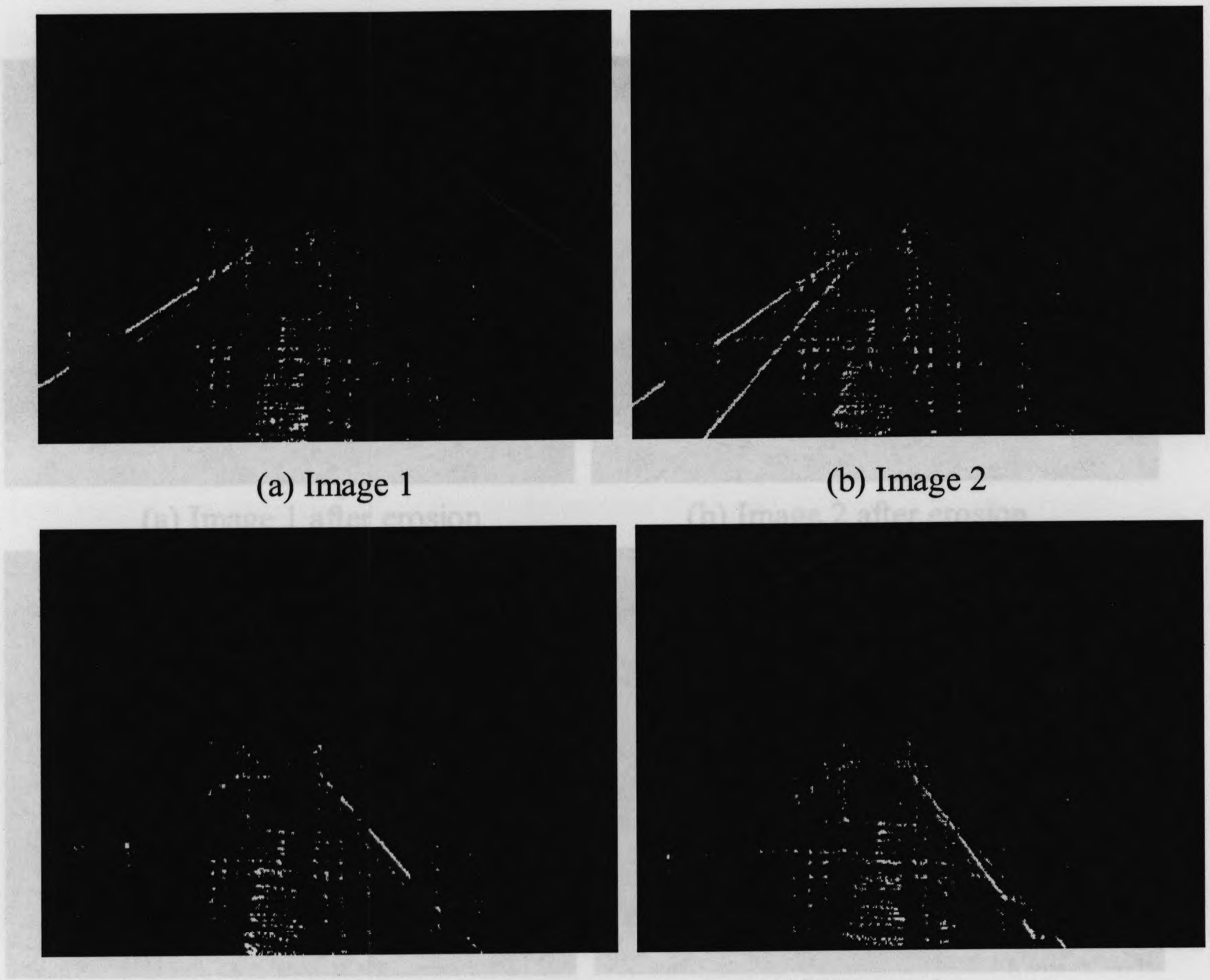
Figure 4.9: The relationship between points along diagonal lines and their I_x and I_y gradients

The points with positive gradient direction are separated into images 1 and 2, corresponding to the left corridor line; and the points with negative gradient direction are separated into images 3 and 4, corresponding to the right corridor line. This procedure is described in Eq.4.5.

$$\begin{aligned}
image_1 &= \{p(m,n) \mid I_x(m,n) > th_1, I_y(m,n) > th_2\} \\
image_2 &= \{p(m,n) \mid I_x(m,n) < -th_1, I_y(m,n) < -th_2\} \\
image_3 &= \{p(m,n) \mid I_x(m,n) < -th_1, I_y(m,n) > th_2\} \\
image_4 &= \{p(m,n) \mid I_x(m,n) > th_1, I_y(m,n) < -th_2\}
\end{aligned} \tag{4.5}$$

where $p(m,n)$ is the pixel at location (m,n) in the image, th_1 and th_2 are gradient thresholds, and I_x, I_y are gradient in x and y directions, respectively.

The selection results of the points along diagonal lines are shown in Figure 4.10. In order to improve calculation speed, we only select points belonging to diagonal lines in the bottom half of the image since the top half contains little information about corridor edges.



(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 4

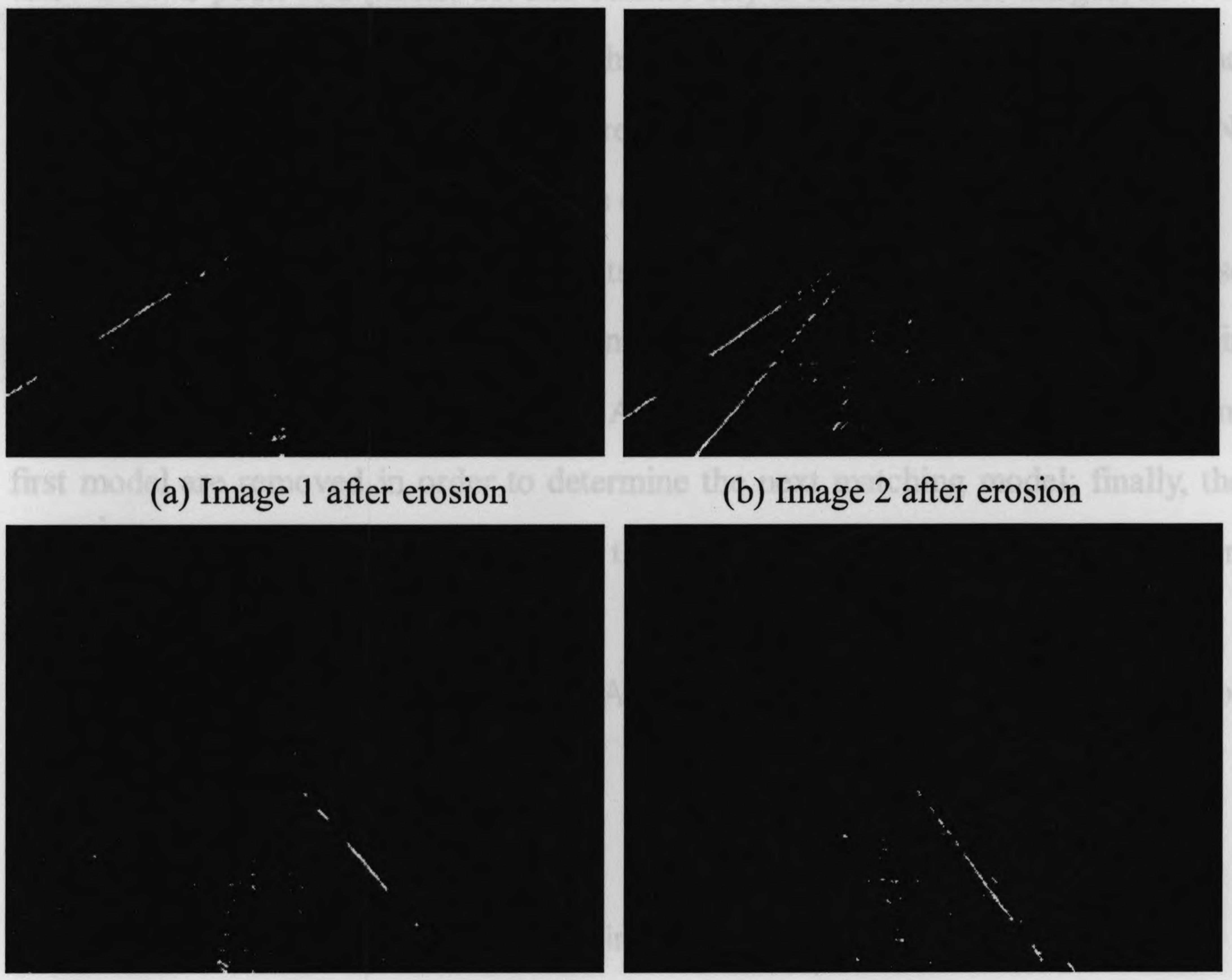
Figure 4.10: Four images containing points belonging to diagonal lines

After selecting and separating points lying on diagonal lines into four binary images, erosion operation is applied on these images to eliminate the spurious edge segments. Two structuring elements (masks) corresponding to diagonal lines are used (Figure 4.11) for erosion operation. Structuring element 1 is used for images 1 and 2, and structuring element 2 is used for images 3 and 4 described in Eq.4.6. Images after erosion operation are shown in Figure 4.12.

$$SE_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad SE_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a) Structuring element 1 (b) Structuring element 2

Figure 4.11: Structuring elements for erosion operation



(a) Image 1 after erosion (b) Image 2 after erosion

(c) Image 3 after erosion (d) Image 4 after erosion

Figure 4.12: The erosion results of four images which contain points belonging to diagonal lines

$$\begin{aligned} Eroded\_image_{(1,2)} &= imerode(\ image_{(1,2)},\ SE_1\ ) \\ Eroded\_image_{(3,4)} &= imerode(\ image_{(3,4)},\ SE_2) \end{aligned} \quad (4.6)$$

4.1.5 Intermediate Level Processing Stage

4.1.5.1 Corridor Line Hypothesis Generation

At this stage, corridor line hypotheses will be generated by fitting lines for the points along diagonal lines in the four eroded images. An efficient and robust method for fitting models in the presence of outliers is the RANSAC algorithm [3]. The RANSAC algorithm works effectively for fitting models between two point sets. The fitting points are called the *inliers,* and the remaining points are called *outliers.* In our applications there are two point sets (inliers set and outliers set) in some corridor images; however, in other corridor images, there may exist three point sets, that is, two inliers sets and one outlier set (see Figure 4.12 (b)). Therefore, we make some modification to the RANSAC algorithm for dealing with different kinds of images in a general way.

The modified RANSAC algorithm fits models for one inliers set or two inliers sets in the presence of many outliers. The first model with the highest number of data points is fitted the same way as the classic RANSAC algorithm; then all data points matching the first model are removed in order to determine the next matching model; finally, the last model is fitted and the process ends until the percentage of remaining points is lower than a ratio.

We summarize the modified RANSAC procedure for interpolating the equations of corridor line hypotheses as below:

R_1: an inlier ratio

R_2: a remaining points ratio which determine if it needs to do RANSAC again

M: total number of data items
N: the number of current data items
dis: distance tolerance

```
$f$ : probability of failure

$k = \left\lceil \frac{\log(f)}{\log(1 - R_1^{\,2})} \right\rceil$

for  $i \leftarrow 1$  to  $k$

    selects 2 data points at random
    estimates line parameter vector  $\vec{x}$ : slope and intercept
    for  $m \leftarrow 1$  to  $N$
        calculates the distance from each point to the line with parameter  $\vec{x}$
        counts  $K \leftarrow$how many data items fit the line within tolerance  $dis$
    end for

    if  $K > R_1$

        save current parameter vector  $\vec{x}$  and data item set  $F$

        if  $(N - K)/M < R_2$
            break;  ← no need to fit lines; exit with success
        else
            remove the data set  $F$ and update current data items
            update N
        end if
    end if
    $i \leftarrow 1$
end for
```

The generated corridor hypotheses are shown in Figure 4.13.

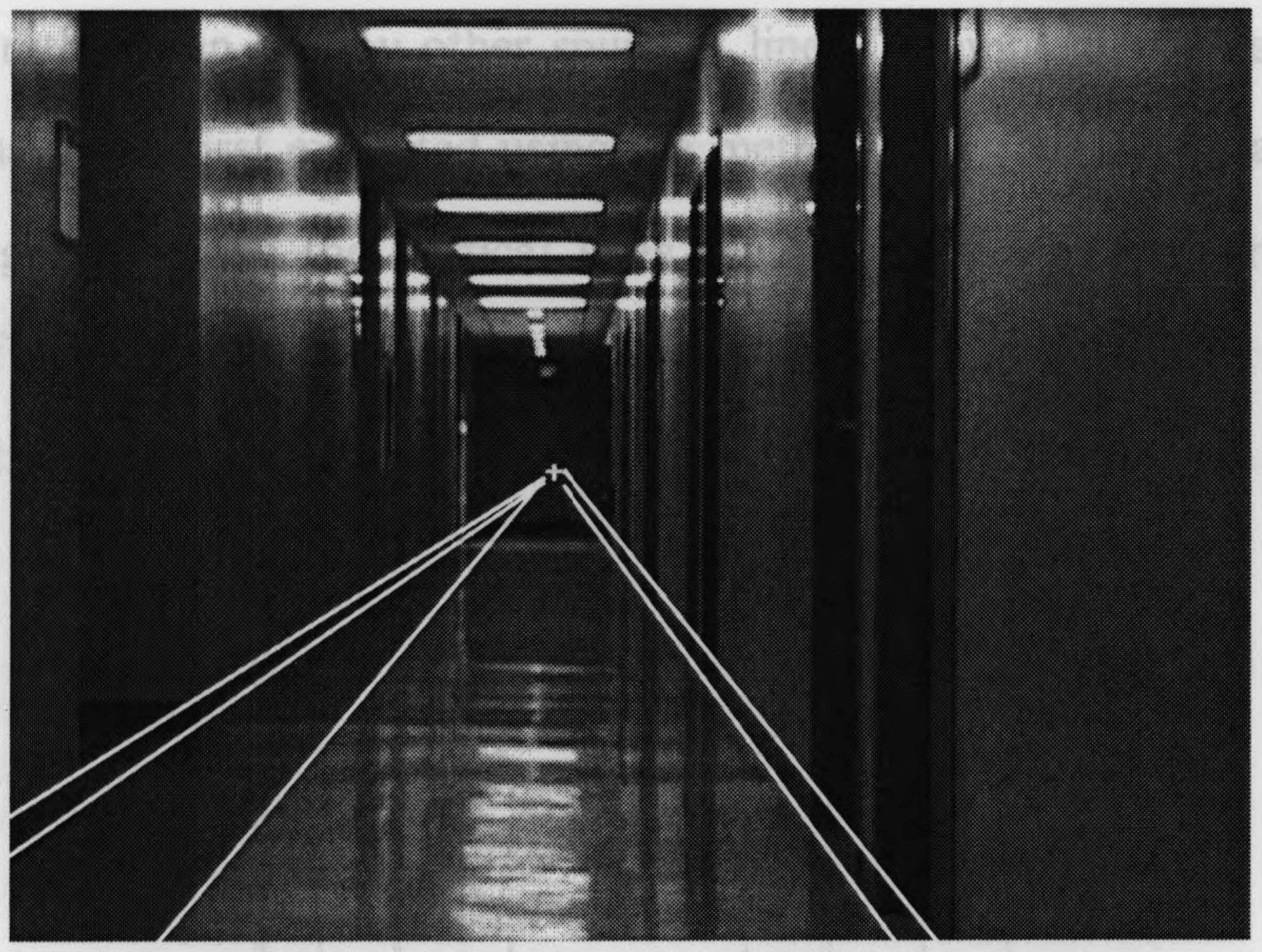


Figure 4.13: Corridor line hypothesis generation

4.1.5.2 CCA and Filtering out Outliers

In most cases, corridor line hypotheses can be successfully generated using above steps. However, if the corridor environment is very complicated, there would be too many outliers in the image. Thus, the RANSAC algorithm would fail to fit hypothesis lines using uniform parameter setting. Therefore, at this point we use the Connected Component Algorithm (CCA) to group adjacent points belonging to diagonal lines into many regions. Small regions which contain less than a minimum number of pixels (region pixel count threshold) are removed from the image. Many outliers can be eliminated by this way. After that the filtered binary images are applied to the RANSAC algorithm again for generating hypothesis lines.

4.1.6 High Level Processing Stage

After the corridor line hypotheses have been successfully generated, the system is directed back to low level stage to search for evidence in order to verify these hypotheses. We choose vertical lines as verification evidence based on the idea that, in a typical corridor scene (Figure 4.14), vertical lines on each side of the wall would fall onto corridor lines rather than on any other spurious lines. To implement this stage, vertical lines in the image are first extracted using the method described above at the low level processing stage; and then the vertical lines are separated into two groups using the information of vanishing point; finally, the hypotheses are confirmed or rejected by the number of vertical lines whose endpoints fall onto the corridor lines.

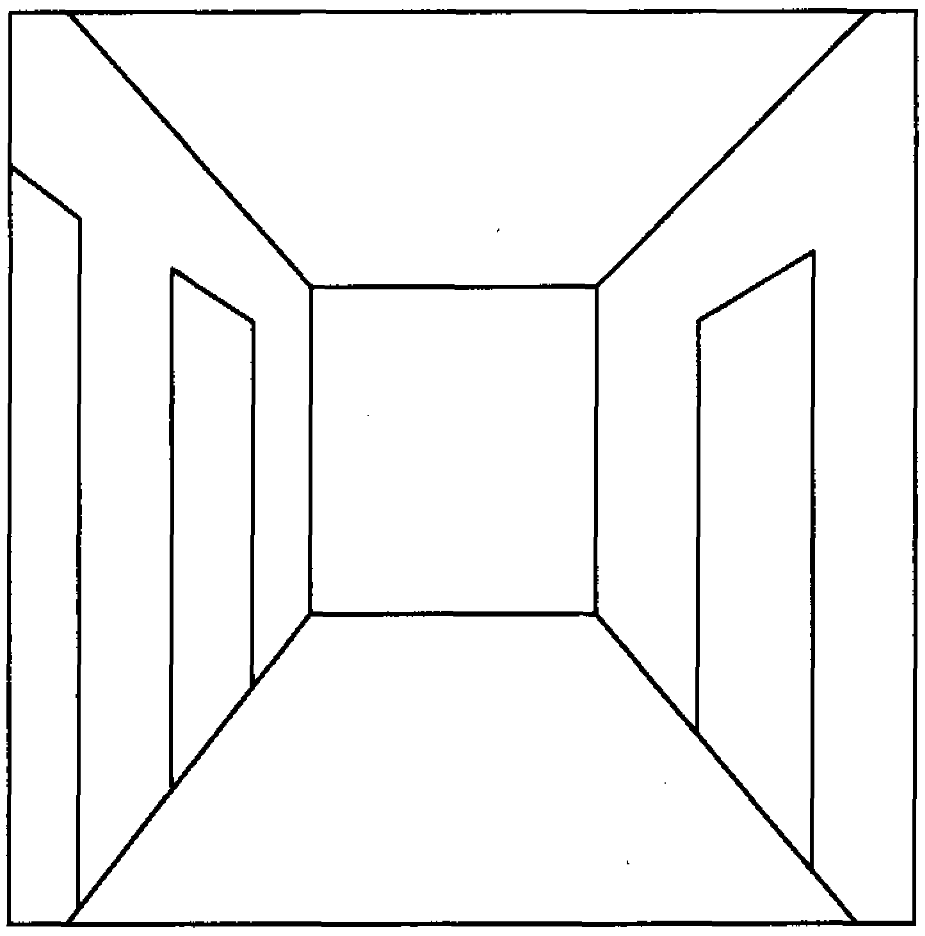


Figure 4.14: Line drawing of a typical corridor scene

A. Vanishing Point Detection

The vanishing point detection at this stage aims to separate vertical lines into two groups corresponding to left and right corridor lines respectively. Information about the vanishing point is also useful for other purposes on robot navigation, such as self-localization and 3-D reconstruction etc. Vanishing point is efficiently and effectively obtained using the subtractive clustering algorithm (SCA) [4].

Subtractive clustering is a fast, one-pass algorithm for estimating the number of clusters and the cluster centers in a set of data. This method assumes each data point is a potential cluster center and calculates a measure of the likelihood that each data point would define the cluster center, based on the density of surrounding data points. The procedure can be described as follows [4]:

(1) Selects the data point with the highest potential to be the first cluster center.

(2) Removes all data points in the vicinity of the first cluster center (as determined by radii), in order to determine the next data cluster and its center location.

(3) Iterates on this process until all of the data is within radii of a cluster center.

In our application, the input set of data is the intersection points of each left corridor line hypothesis with each right corridor line hypothesis (Figure 4.15). The first cluster center is chosen as the estimated vanishing point since it has the highest density of surrounding data points.

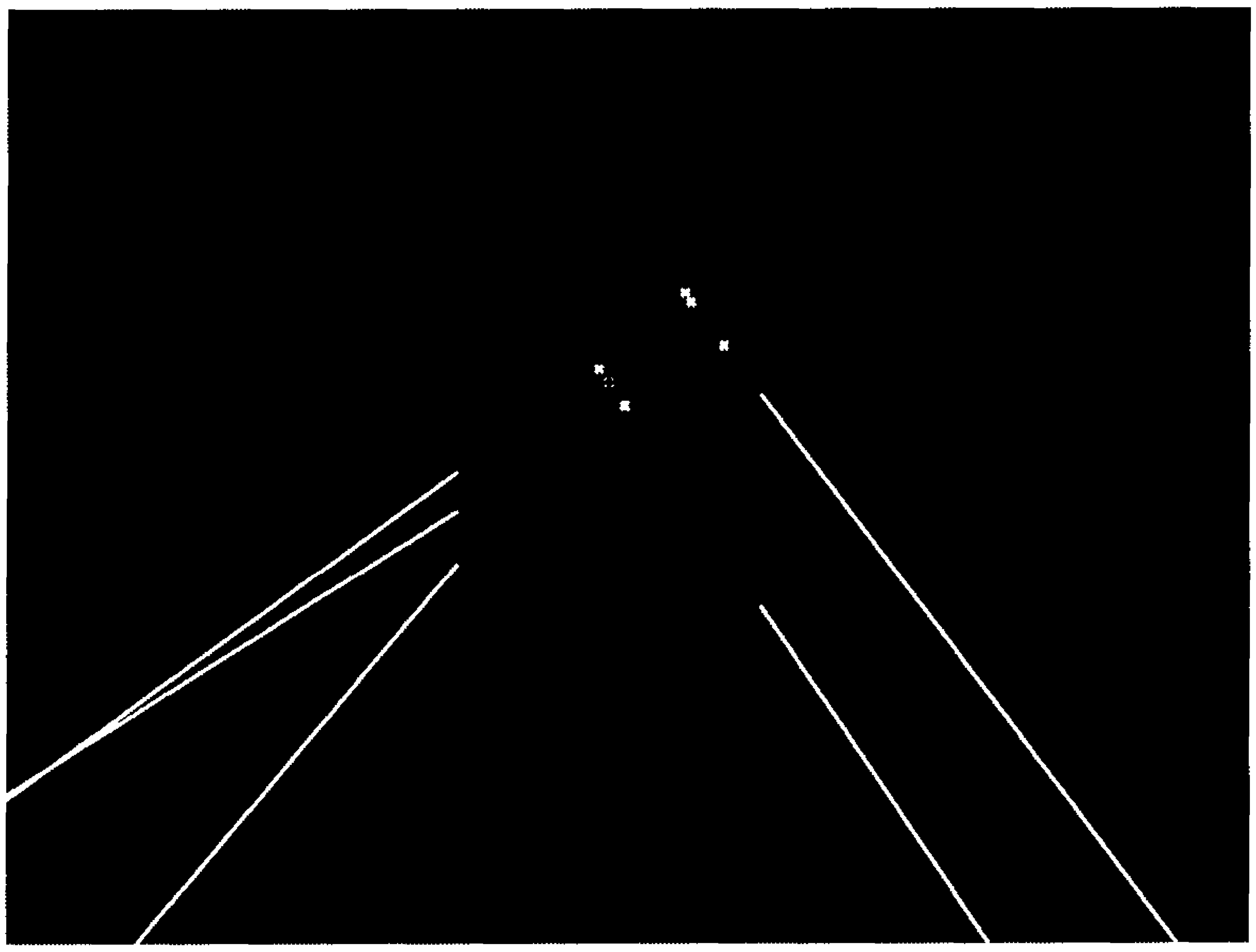


Figure 4.15: Vanishing point detection: "x" points are intersection points of each left corridor line hypothesis with each right corridor line hypothesis; they serve as the input data to the SCA. The "+" point is the detected vanishing point, i.e., the output of the SCA.

B. Hypothesis Verification

Once the vanishing point is achieved, the vertical lines are separated into two groups: the vertical lines on the left side of the vanishing point are grouped into left vertical lines; likewise, the vertical lines on the right side of the vanishing point are grouped into right vertical lines.

After this, we calculate the vertical line percentage for each corridor hypothesis as described in Eq. 4.7. and Eq.4.8.

For the left corridor line hypothesis:

$$\text{vertical line percentage} = \frac{\text{fallen vertical lines}}{LVLG} \tag{4.7}$$

For the right corridor line hypothesis:

$$\text{vertical line percentage} = \frac{\text{fallen vertical lines}}{RVLG} \tag{4.8}$$

where *fallen vertical lines* are the vertical lines which fall onto the corridor line hypothesis, and $LVLG$ (Left Vertical Line Group)or $RVLG$ (Right Vertical Line Group) are left or right vertical lines whose endpoints fall in the area between the vanishing point and the bottom line of the image (Figure 4.16). The hypothetical line whose *vertical line percentage* is big enough (exceeds a ratio threshold) is chosen as the corridor line candidates.

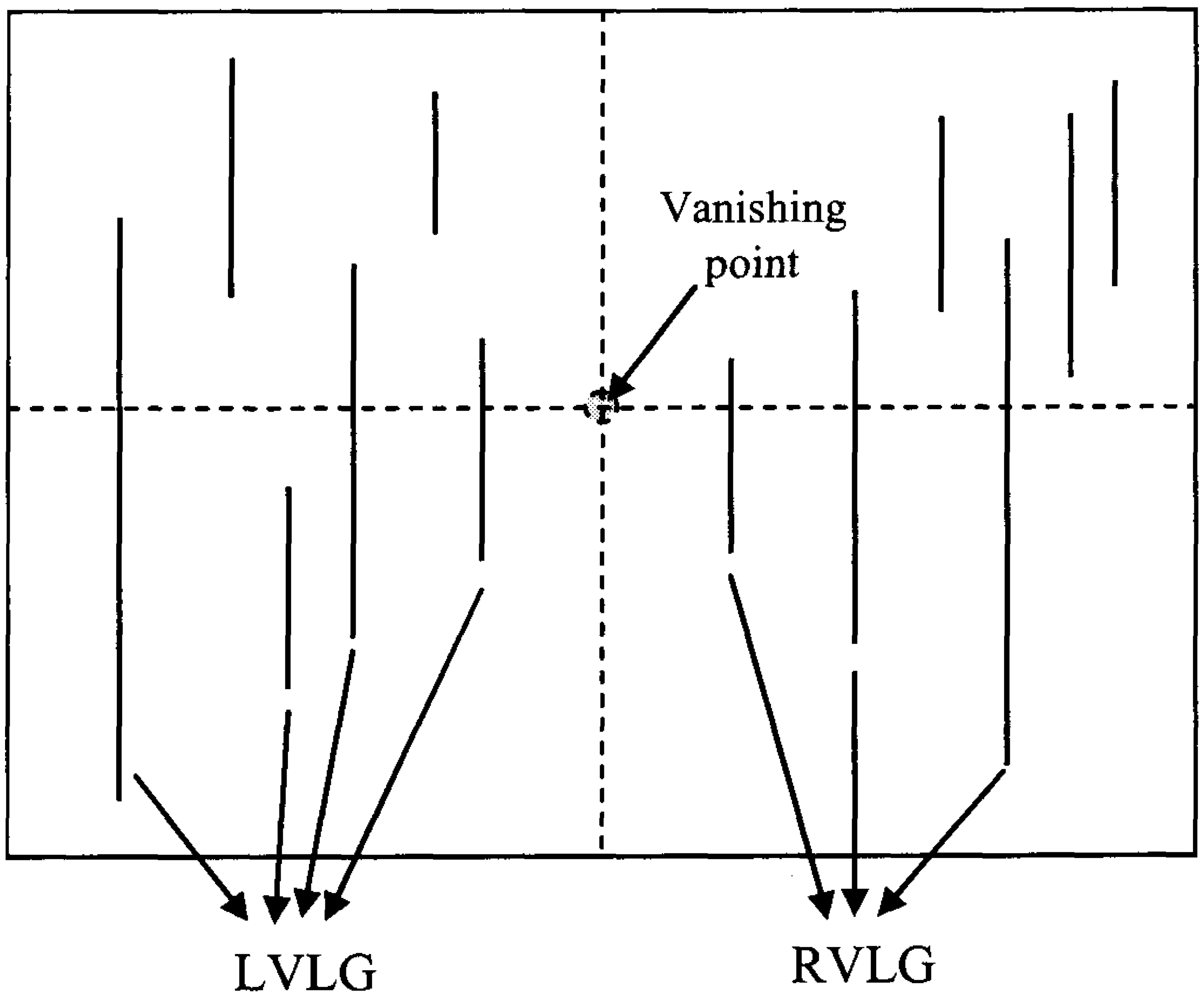


Figure 4.16: Left Vertical Line Group (LVLG) and Right Vertical Line Group (RVLG).

The algorithm can be outlined as follows (using left side as an example):

Assume: *ratio* is a percentage threshold.

```
for $i \leftarrow 1$ to all Left Vertical Line Group (LVLG)
    for $j \leftarrow 1$ to all left corridor hypotheses
        calculate the number of *fallen vertical lines* ;
    end for
end for

for $k \leftarrow 1$ to all left corridor hypotheses
    calculate *vertical line percentage*
    if the *vertical line percentage* > *ratio*
        this is a corridor candidate
    end if
end for
```

What we want to detect is the true corridor line location. In most buildings there are baseboards along the corridor line. The upper edge of the baseboard is collocated with the true corridor line, and they are usually verified as corridor line candidates at the same time. Thus, the bottom line from the verified corridor candidates is chosen for the true corridor line location (Figure 4.17).

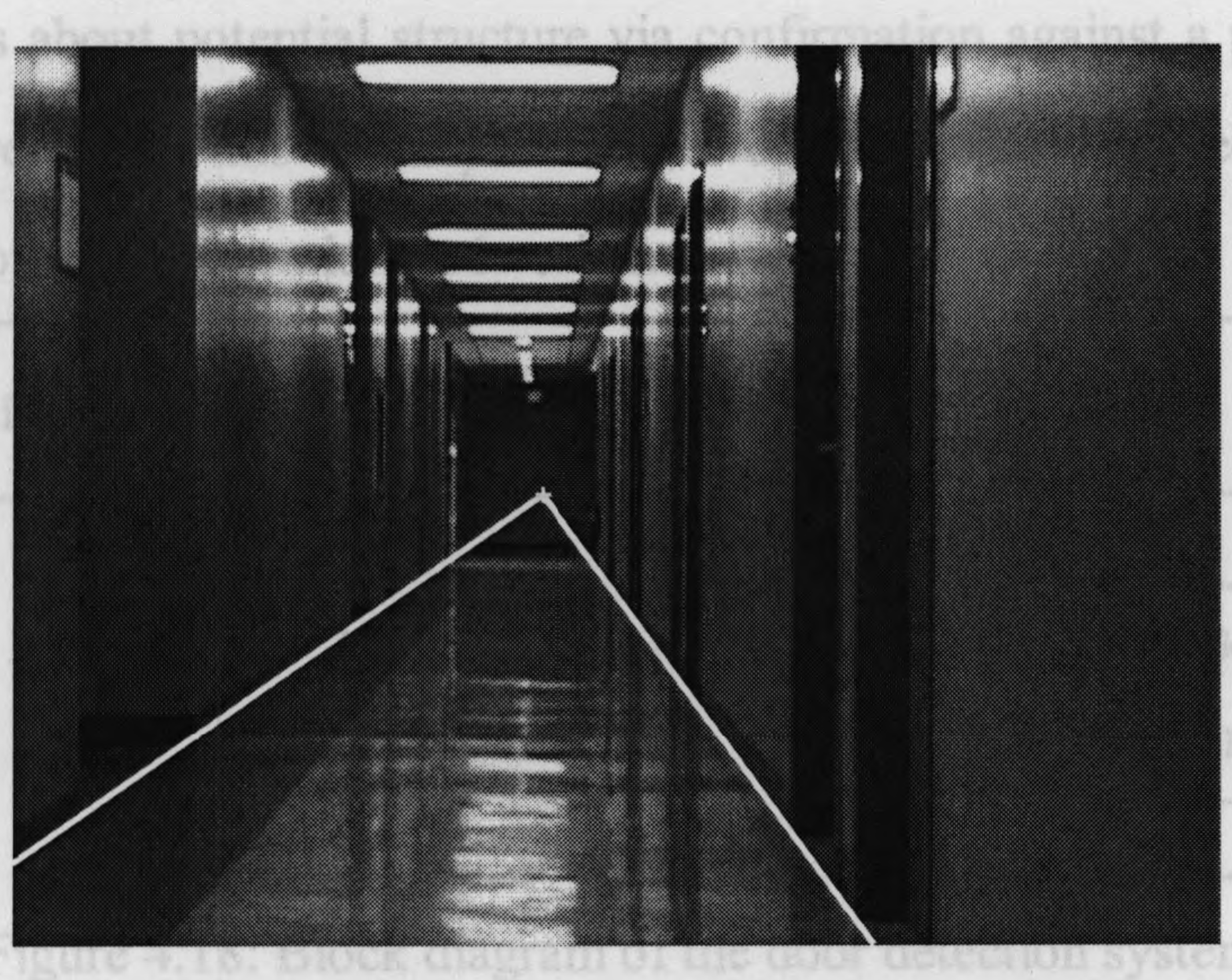


Figure 4.17: Detected corridor lines and the vanishing point

4.2 Door Detection Algorithm

4.2.1 Introduction

Here we present a simple and effective algorithm to detect doors using a single camera for robot navigation and scene understanding. The image sources are the video images captured by a moving robot. This method is used to detect door structures on left or right walls when a robot moves forward along a corridor. Thus, we use a sequence of images as input data. Now our algorithm aims to identify visually important doors in the current image frame. Based on the fact that a door usually spans several frames and can be completely visible in some image frames and partially visible in other frames, an approach that uses a tracking operation in consecutive frames will be later employed to achieve continuous door detection in future work. The tracking algorithm can help solve the problem of partially visible door detection.

We utilize feedback control strategy and HGV method to detect door structures using low level linear features. The system diagram is illustrated in Figure 4.18. The system mainly consists of low level, intermediate level and high level processing stages. This transformation of signals into symbols consists of the extraction of features; the formation of a hypothesis through grouping of multiple features; and finally, verification of the hypothesis about potential structure via confirmation against a predefined model. In particularly, we extend this single-pass framework by employing feedback strategy for more robust hypothesis generation and verification.

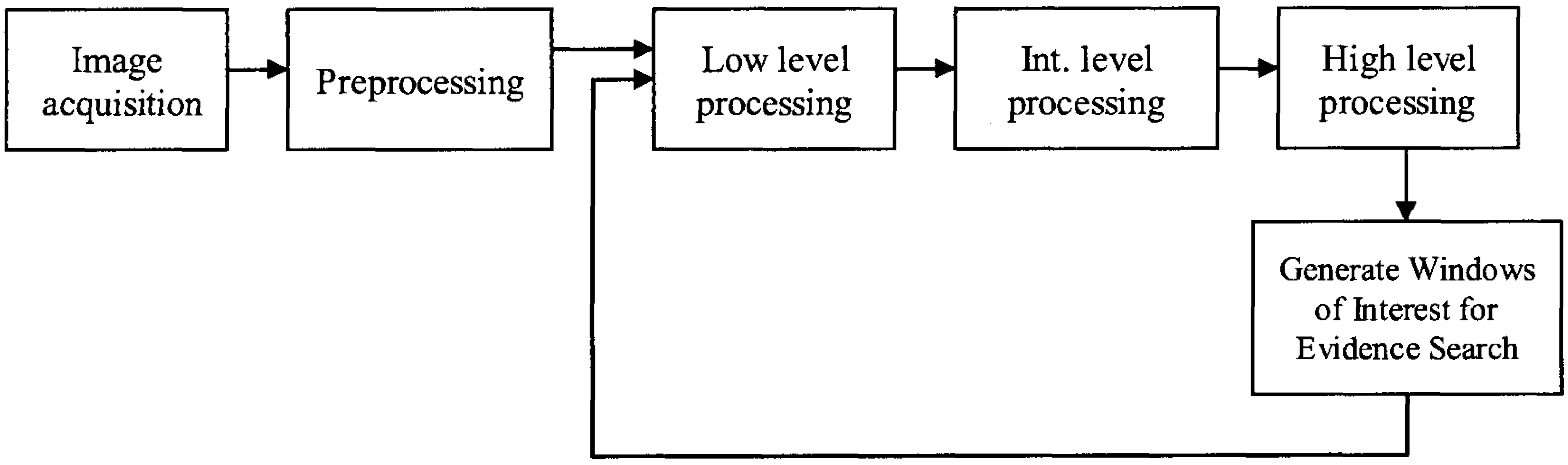


Figure 4.18: Block diagram of the door detection system.

The feedback strategy is employed due to the fact that the single-pass strategy of the hypothesis-verify paradigm becomes inadequate as it is easily thrown off by poor quality data: weak features lead to weak hypotheses. There is also no further search through hypotheses to improve on object matching through focus of attention and missing information analysis [25]. It is essential to use feedback scheme for object recognition in indoor environments based on the following reasons:

Firstly, indoor video image quality presents additional challenges to the task of low level feature extraction due to large variations in brightness, contrast, and reflections as well as the noise introduced by cameras when capturing images. In addition, vibration and shake of the moving camera also cause noises to the image frames.

Secondly, single-pass low level processing is inadequate since some features which we really need can not be extracted using uniform parameters for whole image. So we need feedback to optimize the parameters to search for features of interest.

Finally, low level processing is inadequate as build blocks for the higher level interpretation system. This is due to the fact that the lower levels of processing have no knowledge of the higher level requirements. Consequently, lower levels use local criteria to do their processing regardless of what is required later on. A resulting feature underdetection or the extraction of spurious structures can seriously affect the success of both the hypothesis generation and verification processes.

Therefore, using feedback mechanism incorporated HGV method can be easily expanded to identify various objects in indoor environments.

4.2.1.1 Object Model

In general, a door is composed of a door frame and one or two door panels. Therefore a door can be characterized by three components: a top bar, a left boundary line (left support pole), and a right boundary line (right support pole). We represent door object in terms of its geometric skeleton shape, that is, a door can be detected by finding a upside down U-shape consisting of the top bar, the left boundary line, and the right boundary line.

Currently, our system has been tailored to detect door objects. We believe that our approach can accommodate a wider range of objects easily by incorporating their geometric skeleton shapes into the system.

4.2.1.2 Overview of the Proposed Method

For the door application, some of the top bar edges of doors have very low contrast due to light variation and reflection, and some top bar edges of doors far from the robot are very short due to the perspective projection. However, the top bar edges are the critical features for door detection in our system. Using uniform parameters applied on the whole image for line extraction normally causes two main problems:

(1) Heavy computational cost: If the parameter values (such as gradient magnitude threshold and line length threshold, etc) are set too low, most of the line segments in the image could be extracted. Consequently, it would cost a large amount of time. Furthermore, lots of unwanted line segments are also extracted in the image and this would cause high computational cost for further processing steps.

(2) Missing information: If the parameter values are set too high, it takes much less time than the first case. However, this would cause missing some important information, for example some top bar edges can not be detected at the low level stage.

Based on the above analysis, using uniform parameters for low level feature extraction is not a good choice. Therefore, two parameter settings are applied at low level stage in the system. This method is designed in an attempt to reduce the computational cost as much as possible. The basic procedure is described as follows:

1) Use a parameter setting designed to extract long straight vertical lines at low

level processing stages, by taking advantage of the prior knowledge of vertical orientation;

2) Group these vertical lines into pairs of two support boundary lines to form door hypotheses;
3) Form a window of interest for each door hypothesis. Information is then fed back to the low level stage to adjust the parameter setting for searching for top bar edge in each window of interest. If a top bar edge is extracted, the corresponding U-shape hypothesis is generated;
4) Confirm or reject the U-shape hypotheses by corridor line evidence.

The flow chart of the proposed algorithm is shown in Figure 4.19. It is assumed that the door frame or door panel has relative contrast with respect to the wall.

4.2.2 Image Acquisition

In our system, we use a sequence of images obtained from a digital video camera. The camera is mounted on the robot at an angle from the direction of motion such that it faces to the right or left wall. Figure 4.20 shows the motion of a robot inside a corridor and the camera orientation. The camera is tilted so that the top bars of doors can be captured in the video.

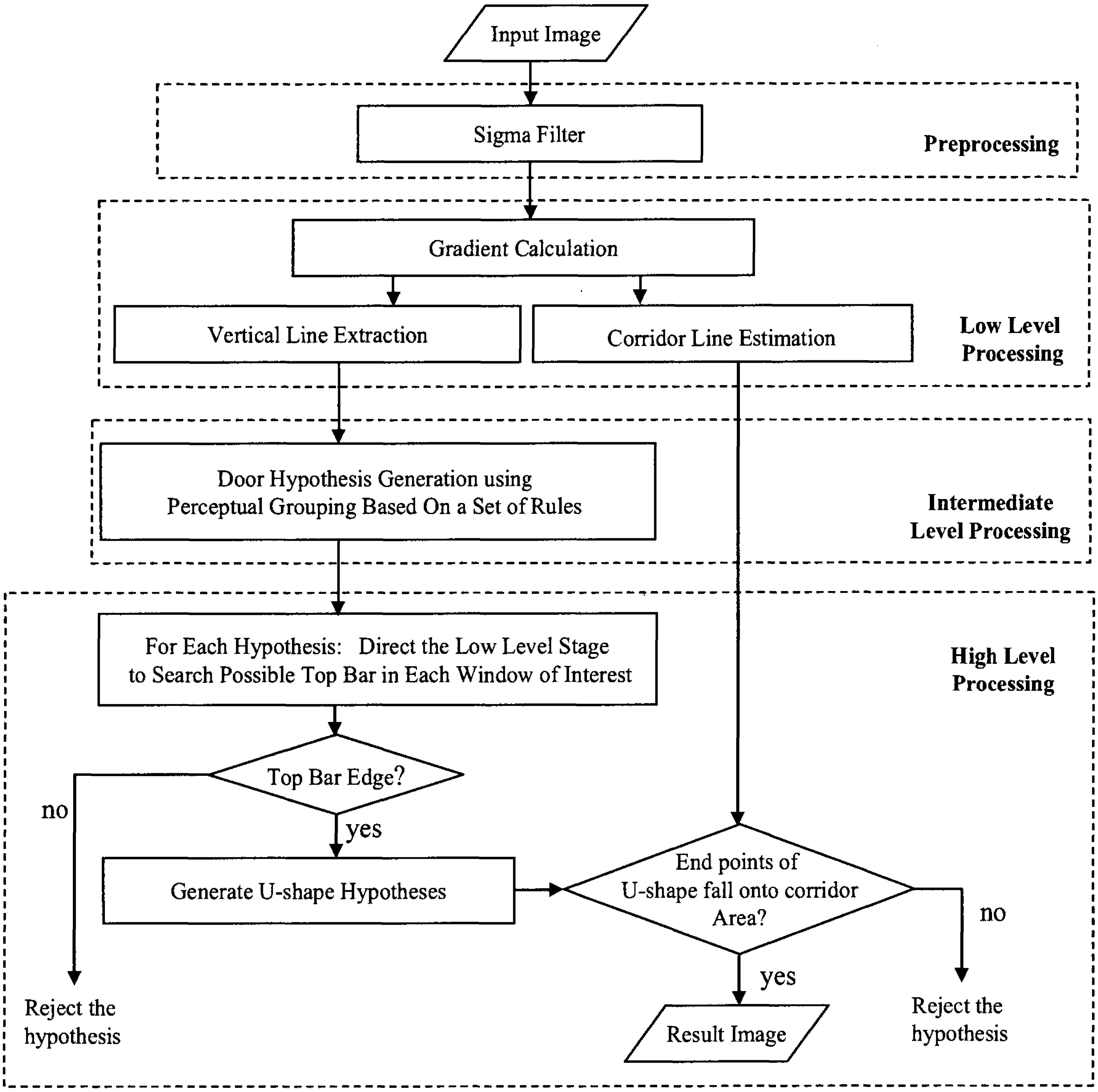

Input Image
Sigma Filter
Preprocessing
Gradient Calculation
Vertical Line Extraction
Corridor Line Estimation
Low Level Processing
Door Hypothesis Generation using Perceptual Grouping Based On a Set of Rules
Intermediate Level Processing
For Each Hypothesis: Direct the Low Level Stage to Search Possible Top Bar in Each Window of Interest
High Level Processing
Top Bar Edge?
no
yes
Generate U-shape Hypotheses
End points of U-shape fall onto corridor Area?
no
yes
Reject the hypothesis
Result Image
Reject the hypothesis

Figure 4.19: Flow chart of the door detection algorithm

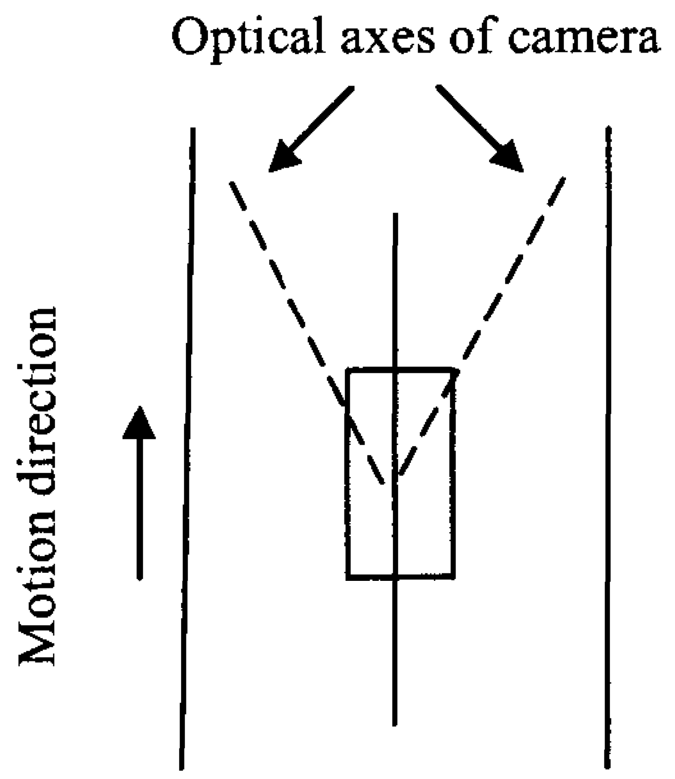


Figure 4.20: Robot motion along a corridor and camera orientation. The camera is mounted on the top of a robot and is tilted towards the left wall or the right wall.

4.2.3 Low Level Processing to Detect vertical Lines

Preprocessing, gradient calculation and vertical line extraction steps in the low level processing stage have been described in section 4.1.3 and 4.1.4. Here, we present further details regarding vertical line extraction for door detection.

As described in Chapter 3, One of the features of the FLF algorithm [71] is its ability to disambiguate lines formed with the different orientations of intensity change. As shown in Figure 4.21, a line formed due to bright to dark change is put in bucket 1, and a line formed due to dark to bright change is put in bucket 5.

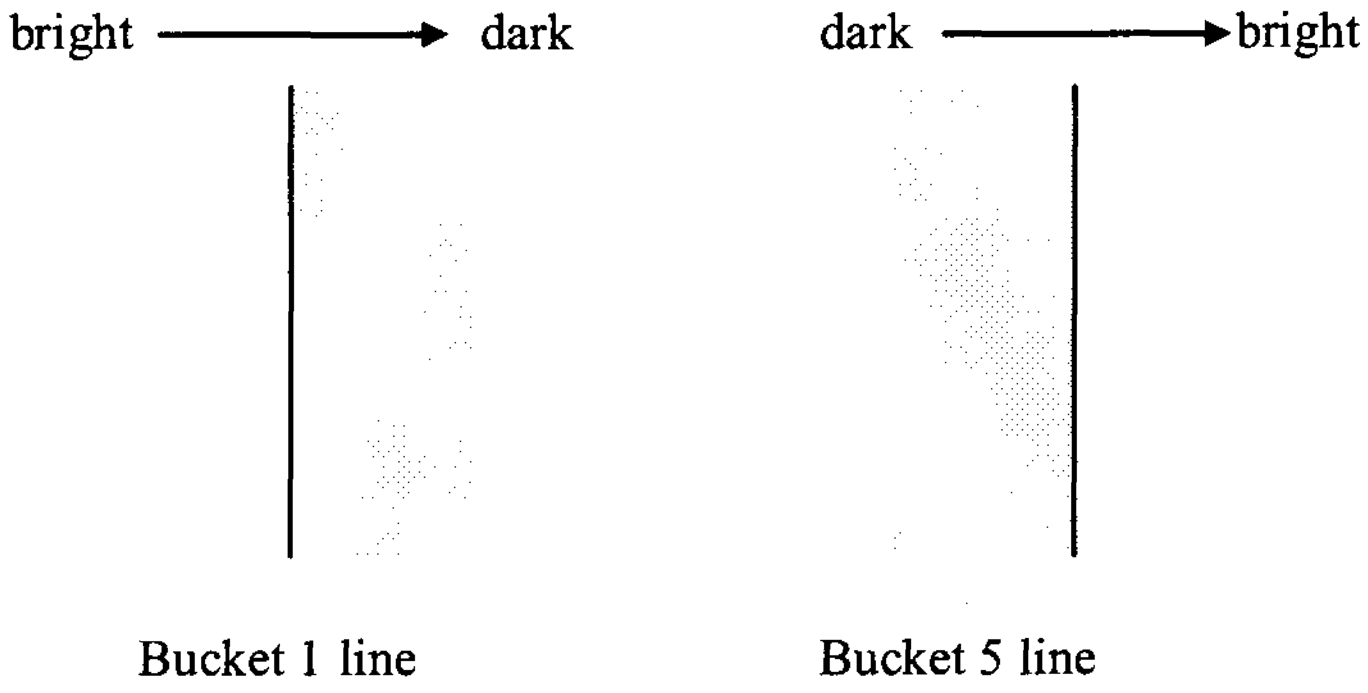


Figure 4.21: Brightness variation of bucket 1 line and bucket 5 line

We extract vertical lines using bucket 1 and bucket 5 in FLF algorithm, and label these lines by number 1 and 5 for further processing. Vertical line extraction results labeled with 1 and 5 of two images are illustrated in Figure 4.22. These two video images are captured from different points of view. Figure 4.22(a) was acquired on the third floor, TEB; and Figure 4.22(b) was acquired on the third floor in SEB.

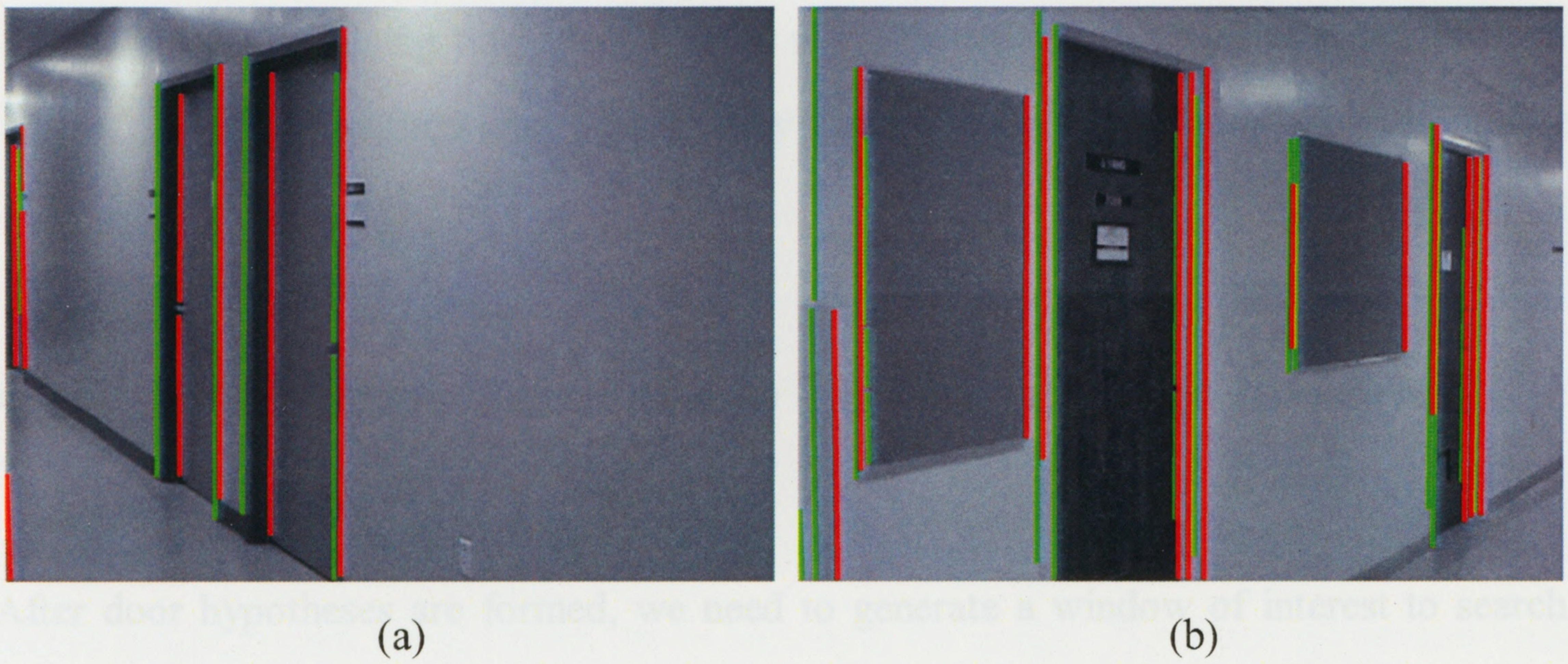


Figure 4.22: Extracted vertical lines of the image: green lines are labeled with 1; and red lines are labeled with 5 (a) facing to the right wall (b) facing to the left wall.

4.2.4 Perceptual Grouping to Generate Hypotheses

Once the vertical line segments in the image have been extracted, these linear primitives need to be perceptually grouped into potential boundary lines of doors so as to form door hypotheses. This is achieved based on the following set of rules:

(1) Group each 1-labeled line with the 5-labeled lines on its right side to form two boundary lines of a door hypothesis. The rationale is that since the left boundary edge that results from wall/door intersection must be from bright to dark brightness variation, the left outer boundary line must be labeled as 1(Figure 4.21). Similarly, the right outer boundary edge resulting from door/wall intersection must be a 5-labeled line. That is to say, only 1-labeled line with a matching 5-labeled line to the right can form a door. It is impossible for two 1-labeled lines or two 5-labeled lines to generate a door. This is applicable for all the images no matter from which point of view they are captured in the corridor (see Figure 4.22).

(2) Reject the line groupings which have too small or too large distances

between the two lines. This is due to the fact that the width of a door (the distance between the left and right boundary lines) should be within a certain range. Using this rule we could significantly reduce the number of spurious line groupings and substantially reduce the computational cost for further processing.

4.2.5 Information Feedback to Low Level stage to Search the Top Bar

After door hypotheses are formed, we need to generate a window of interest to search possible top bar edge for each hypothesis. The window of interest is a rectangle formed by surrounding the top points of the two boundary lines. Figure 4.23 gives examples of windows of interest for top bar searching. The image was captured in TEB third floor.

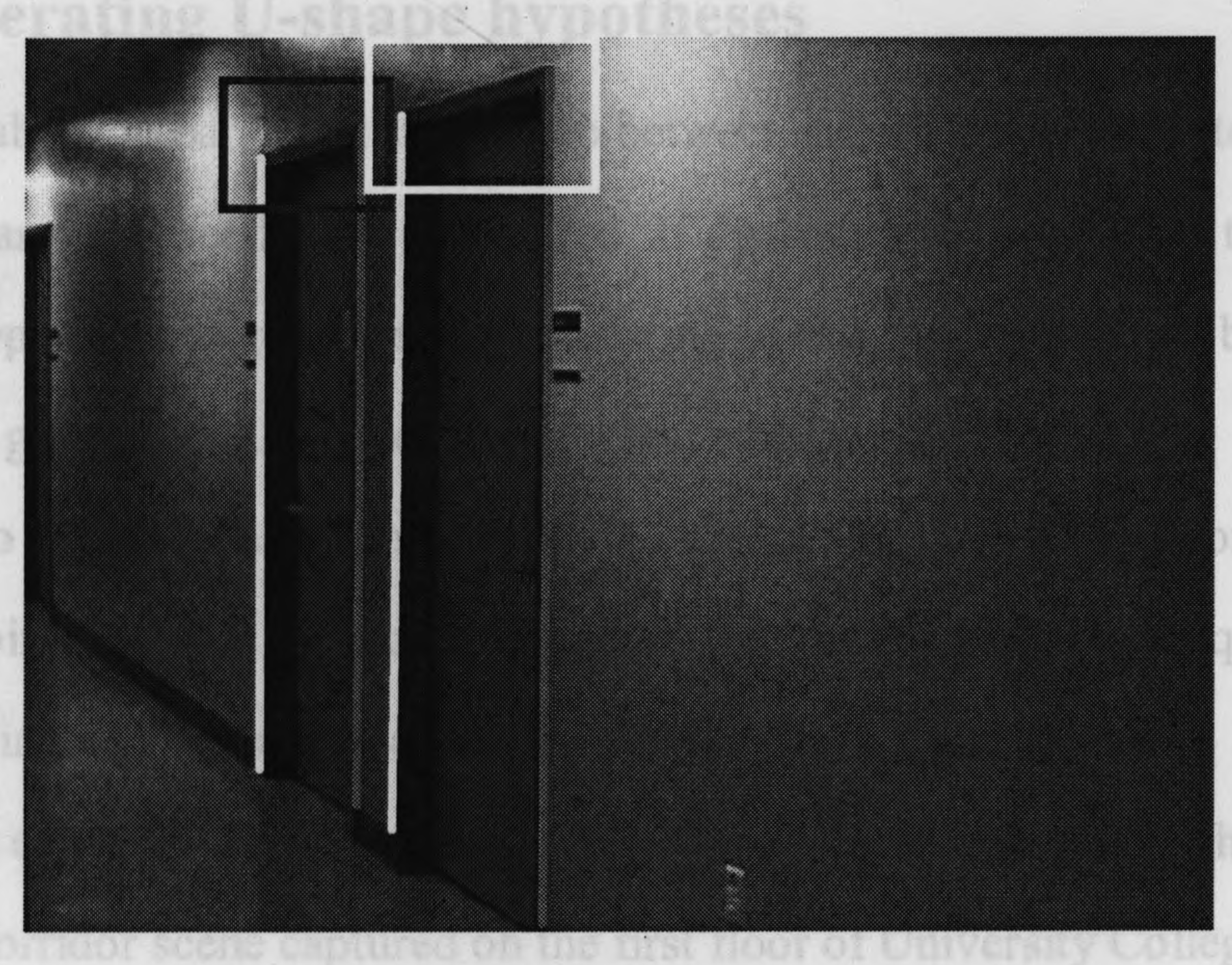


Figure 4.23: Windows of interest for top bar searching

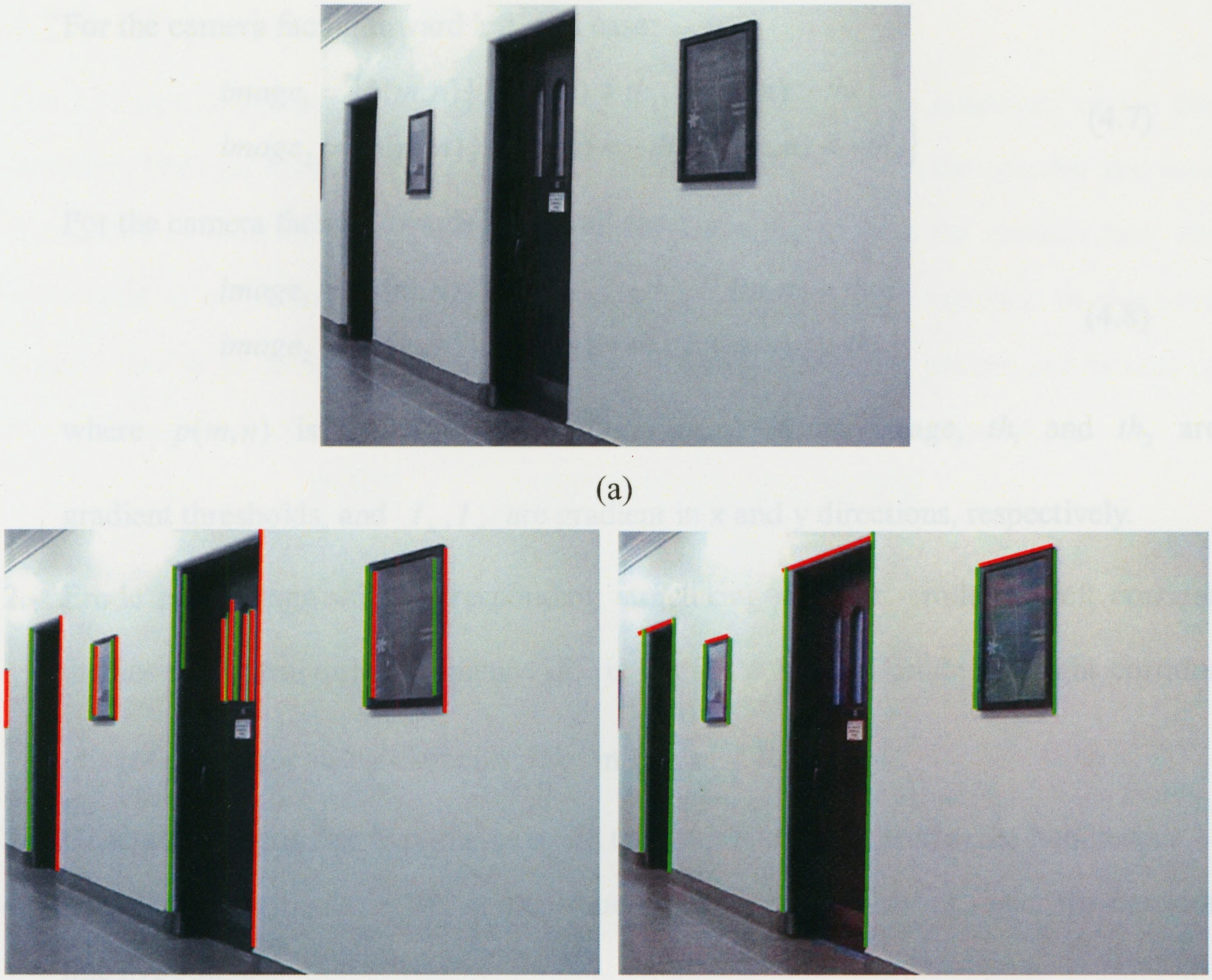
As discussed in Chapter 3, the FLF algorithm performs well on extracting straight lines whose pixel gradient orientation lies at the center of the bucket. As a result, the directional bucket could be tuned and centered on the expected orientation to obtain good line extraction results even with low contrast cases. This is significant for indoor scene application because some of the top bar edges often have low contrast and gradient variation due to the lightning and reflection. We employ this feature to search top bar in each window of interest by tuning the directional bucket and centering it on the desired orientation. Here, the desired orientation is obtained by the line which is determined by the top points of two boundary lines. Furthermore, parameter setting is suitably adjusted for extracting top bar line segment in each window of interest.

We just search for the boundary lines intersected by wall and door which are formed due to bright to dark change from top to bottom in the image. There may be several such line segments in the window of interest depending on the nature of the doorframe, shadows, and reflections, so we choose the longest one as the match.

4.2.6 Generating U-shape hypotheses

For each available top bar, the distance gap between an end point of the top bar and each vertical boundary line need to be calculated. If distance gap is within a threshold value, the U-shape hypothesis is generated by consisting of this top bar and two boundary lines.

In some generated U-shape hypotheses, the vertical boundary lines may be segmented into several line segments due to the occlusion by the doorplate from the camera viewpoint. In order to link the boundary line fragments into a unique straight line, we use a linking and merging operation on the lines with the same label based on proximity and continuity criterion [27]. Figure 4.24 illustrates the U-shape hypothesis results of the corridor scene captured on the first floor of University College.



(a)

(b) (c)

Figure 4.24: U-shape hypothesis results (a) Input intensity image; (b) Extracted vertical lines: green lines are labeled as 1 and red lines are labeled as 5; (c) Generated U-shape hypotheses.

4.2.7 Detecting Corridor Line

This step is somewhat similar to the corridor location algorithm described in section 4.1, but contains few modifications due to the different view point of the camera. The process can be summarized as follows.

1. Selection of points belonging to diagonal lines: The points within the bottom half of the image having X-direction and Y-direction gradients above a certain threshold limit are selected and separated into two binary images.

For the camera facing toward left wall case:

$$\begin{aligned} image_1 &= \{p(m,n) \mid I_x(m,n) > th_1, I_y(m,n) > th_2\} \\ image_2 &= \{p(m,n) \mid I_x(m,n) < -th_1, I_y(m,n) < -th_2\} \end{aligned} \qquad (4.7)$$

For the camera facing towards right wall case:

$$\begin{aligned} image_1 &= \{p(m,n) \mid I_x(m,n) < -th_1, I_y(m,n) > th_2\} \\ image_2 &= \{p(m,n) \mid I_x(m,n) > th_1, I_y(m,n) < -th_2\} \end{aligned} \qquad (4.8)$$

where $p(m,n)$ is the pixel at location (m,n) in the image, th_1 and th_2 are gradient thresholds, and I_x, I_y are gradient in x and y directions, respectively.

2. Erode each image with corresponding structuring element: erode the left corridor images with structuring element SE_1 in Figure 4.11; and erode the right corridor images with structuring element SE_2 in Figure 4.11.
3. Generate corridor line hypotheses using the RANSAC algorithm in the two images.
4. Hypothesis verification: use vertical line evidence to confirm or reject the corridor line hypotheses.

The result of corridor line extraction of Figure 4.24 is illustrated in Figure 4.25:

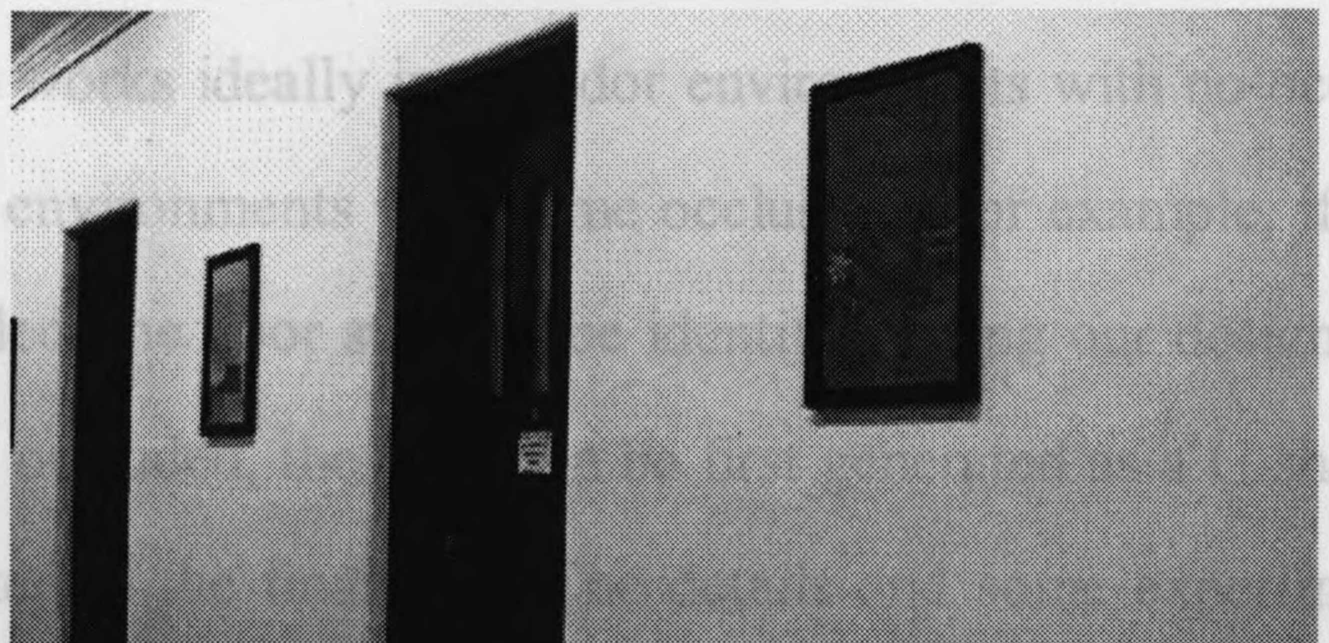


Figure 4.25: Corridor line detection result for door algorithm

4.2.8 Hypothesis Verification

The U-shape hypothesis is verified as a door by checking if the endpoints of the two boundary lines (at least one boundary line) of the U-shape fall onto the corridor line area. If one or two boundary lines of the U-shape hypothesis fall onto the corridor line, this hypothesis is confirmed as a door; otherwise the hypothesis is rejected. In this way, objects similar to door structures such as paintings (Figure 4.24), posters and boards on the wall can be rejected. The final result of Figure 4.24 is shown in Figure 4.26.

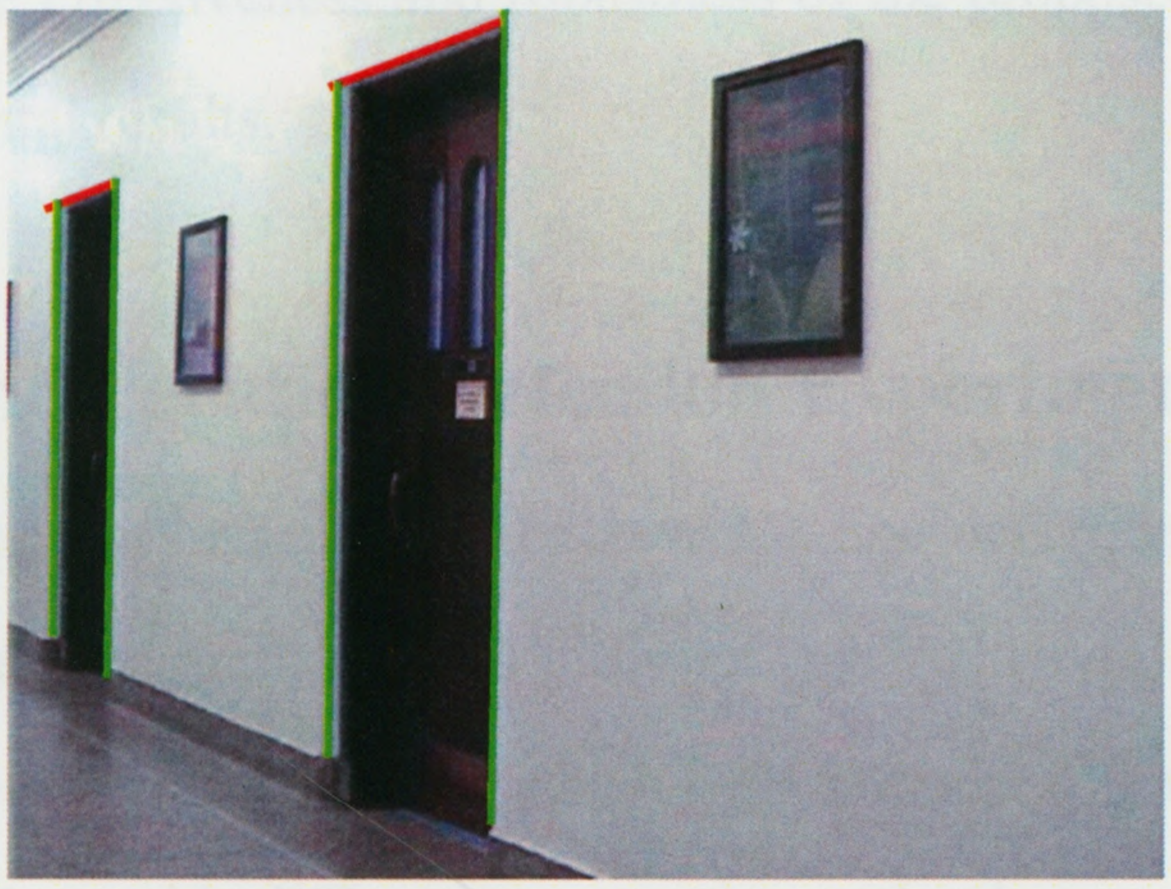


Figure 4.26: Door detection results. Objects similar to door structures such as paintings and posters on the wall are rejected at hypothesis verification stage.

Our algorithm works ideally in corridor environments with no occlusion, however it still works well in environments with some occlusion. For example, if one boundary line of a door is occluded, the door still can be identified using our designed method. If both boundary lines are occluded, the door can be first generated as a U-shape hypothesis and then will be rejected at the final step. The details and some experiment results will be discussed in Chapter 5. Identifying doors in occluded corridor environments will be part of our future work.

Chapter 5
Experiments and Results

In this chapter we present the experiments carried out with the two algorithms described in preceding chapters and discuss the results. First, we present the results of the corridor line detection algorithm, and develop the evaluation metric to measure the performance of this method. Finally, we present the experiments of the door detection algorithm and discuss the results. The effectiveness and robustness of the proposed methods have been validated by experimental results.

5.1 Corridor Line Location Detection Experiments

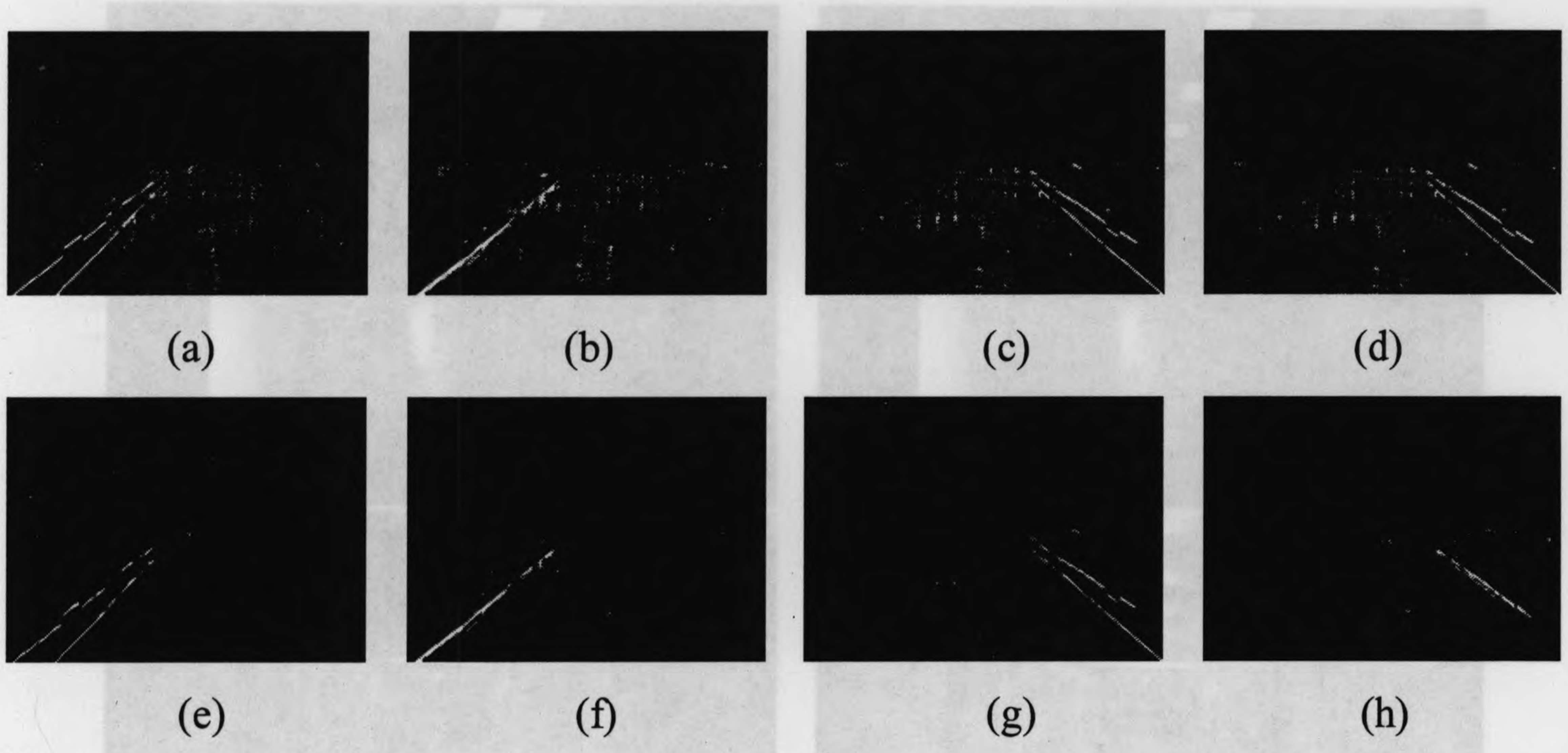
In order to evaluate the performance of the corridor line detection algorithm, we have tested 200 video images acquired from 50 different corridor environments. These image frames are grabbed from video sequences which were taken at different moving speed, at different time of the day, with different perspective and robot position under different lighting conditions.

5.1.1 Illustration of the algorithm procedure

Figure 5.1 to Figure 5.2 show the procedure of the corridor detection algorithm. The image was acquired on the 1st floor of Western Science Center.

5.1.2 Illustration of some results for a variety of corridor environments

Figure 5.3 to Figure 5.5 show the processing results of several video images captured in different corridor environments. From these experiments, we can see that the proposed method can successfully detect the vanishing point and true corridor lines in a variety of corridor environments, such as, different widths and the presence of people. These experiments demonstrate the proposed method is effective and robust in a variety of real corridor scenes.


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

(e) (f) (g) (h)

Figure 5.1: Selection of points lying on diagonal lines and separation into four images. (a)~(d) Selection of points lying on diagonal lines in bottom half of the image (e)~(h) after erosion operation.

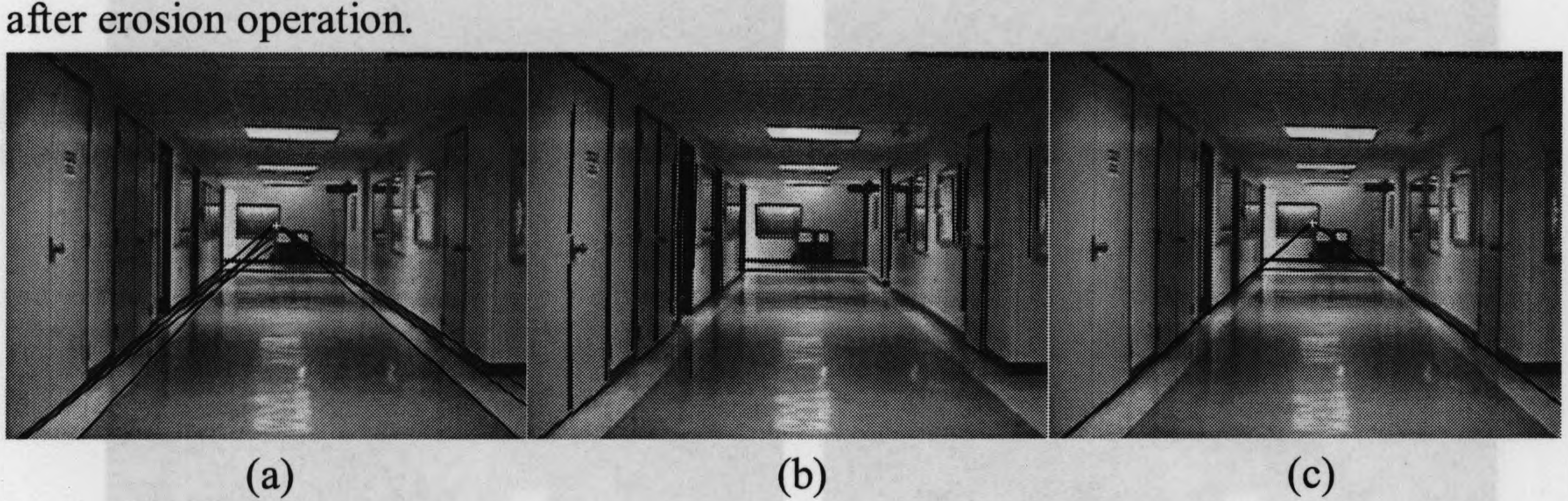
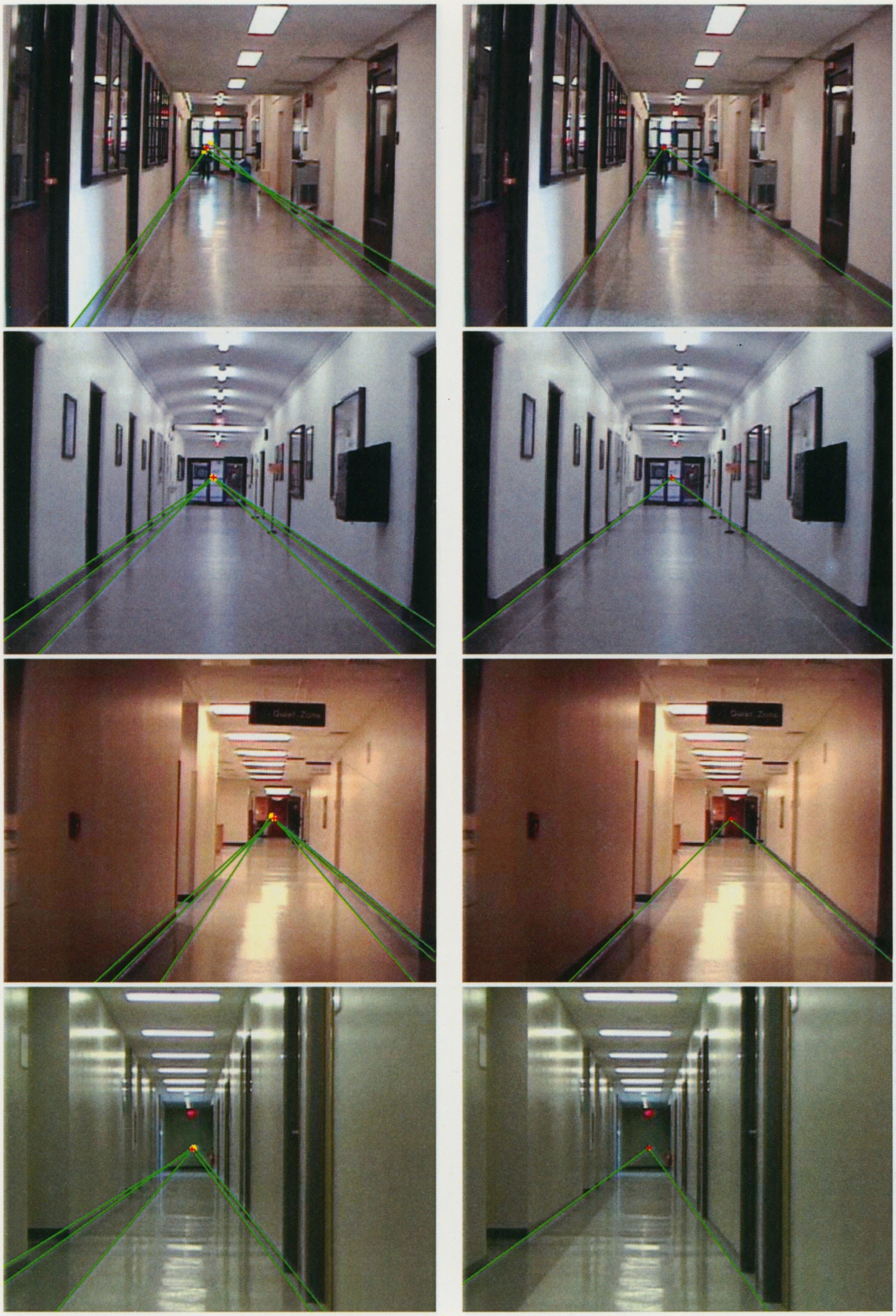

(a) (b) (c)

Figure 5.2: Detecting results (a) Detected vanishing point and generated corridor line hypotheses (b) Extracted vertical lines (c) Detected corridor line locations.

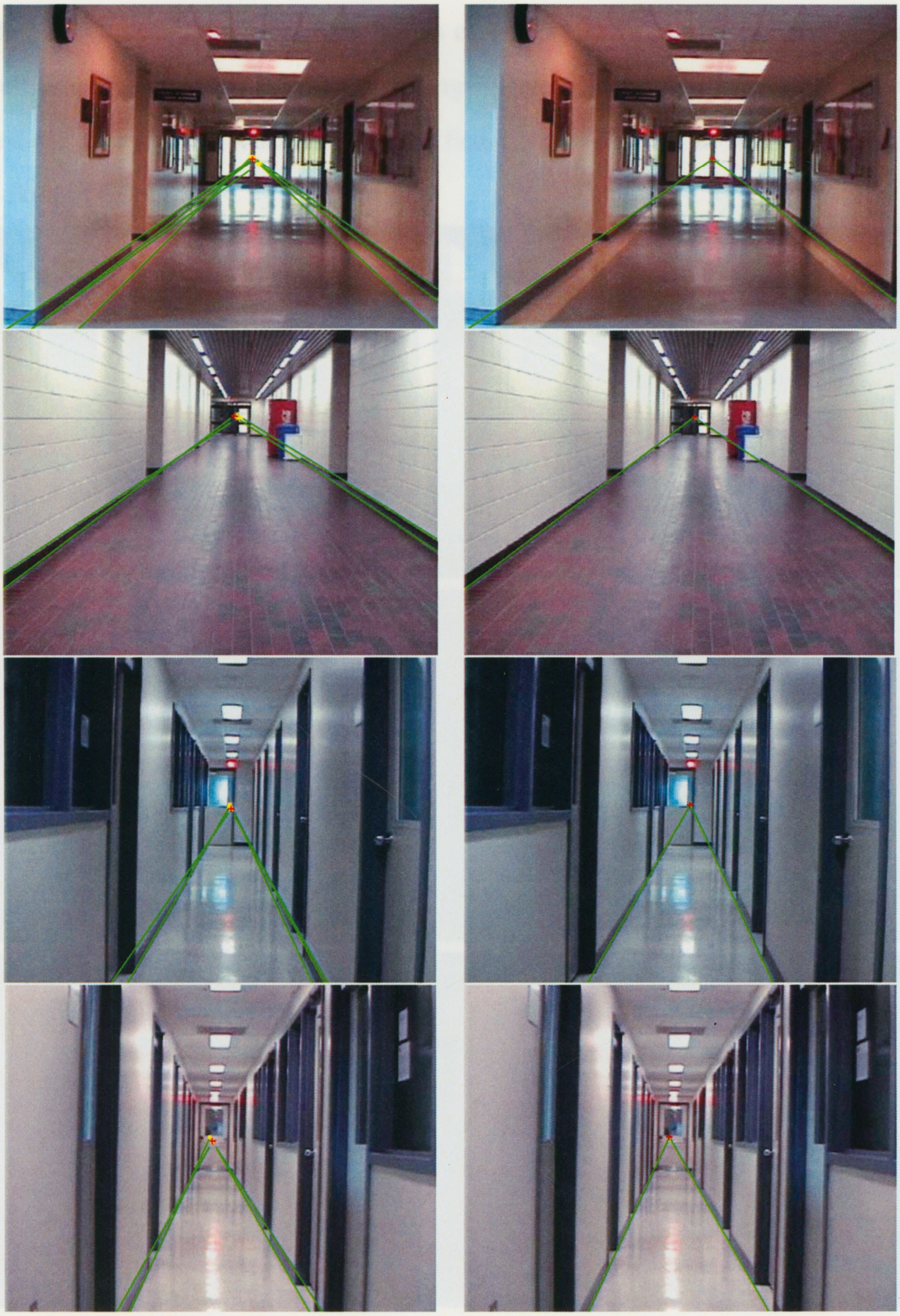
The four images in Figure 5.3 were acquired at following places respectively: the first floor of Physics & Astronomy, the first floor of University College, the first floor of University Hospital, and the third floor of TEB.

Figure 5.4 illustrates the processing results of very wide and narrow corridor environments. The four images were acquired at following places respectively: the first floor of Western Science Center, the third floor of Social Science Center, the first floor of Western Science Center, and the first floor of Western Science Center.



(a) (b)

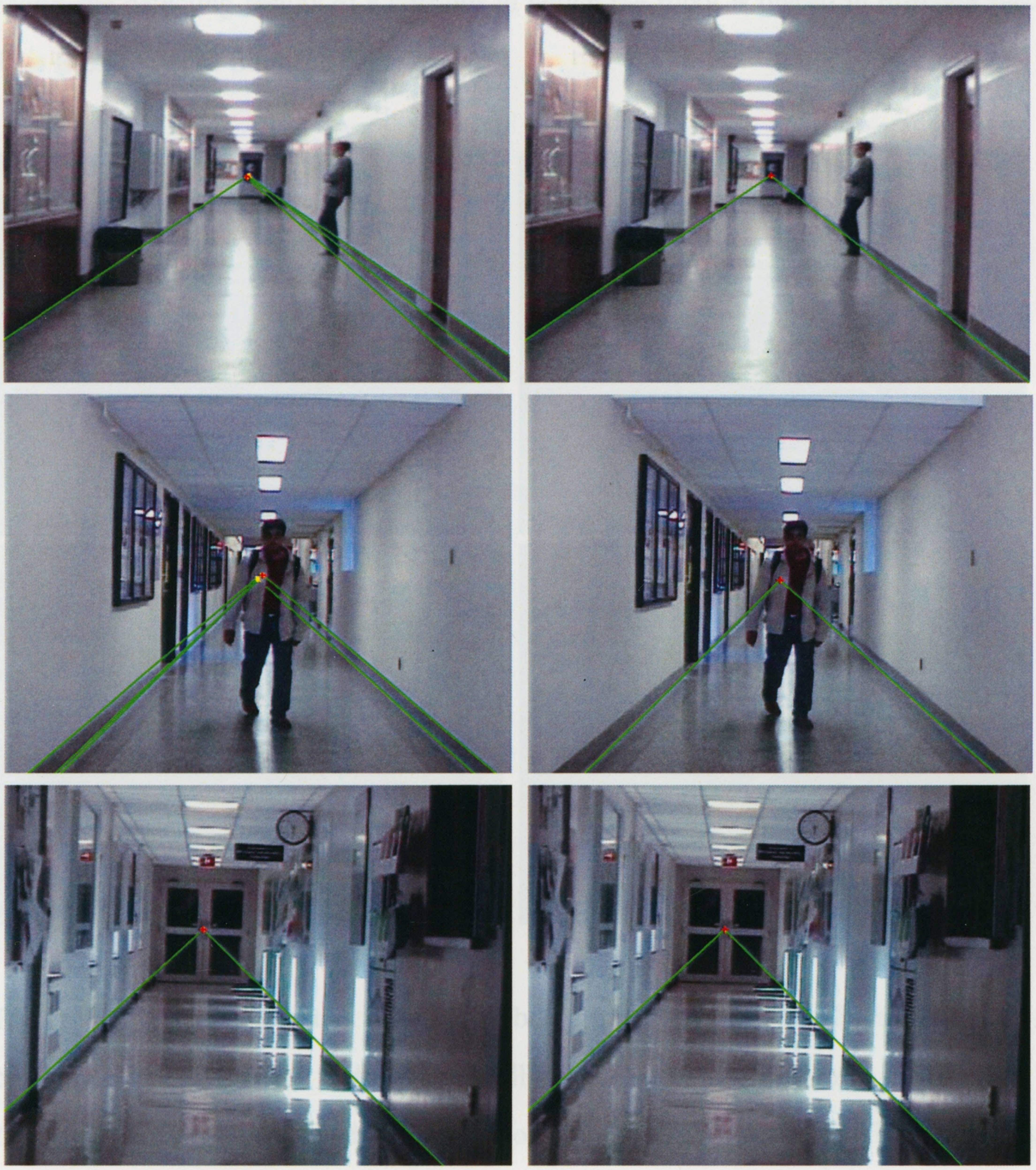
Figure 5.3: Vanishing point and corridor line detecting results (a) Detected vanishing point and generated corridor line hypotheses (b) Detected corridor line locations.



(a) (b)

Figure 5.4: Detecting results of corridors with different width: wide corridors in the first two images and narrow corridors in the last two images (a) Detected vanishing point and generated corridor line hypotheses (b) Detected corridor line locations.

Figure 5.5 shows the processing results of corridor scenes in the presence of people as well as strong reflections. The three images were respectively acquired at following places: the first floor of Thames Hall, the first floor of Physics & Astronomy, and the third floor of SEB.



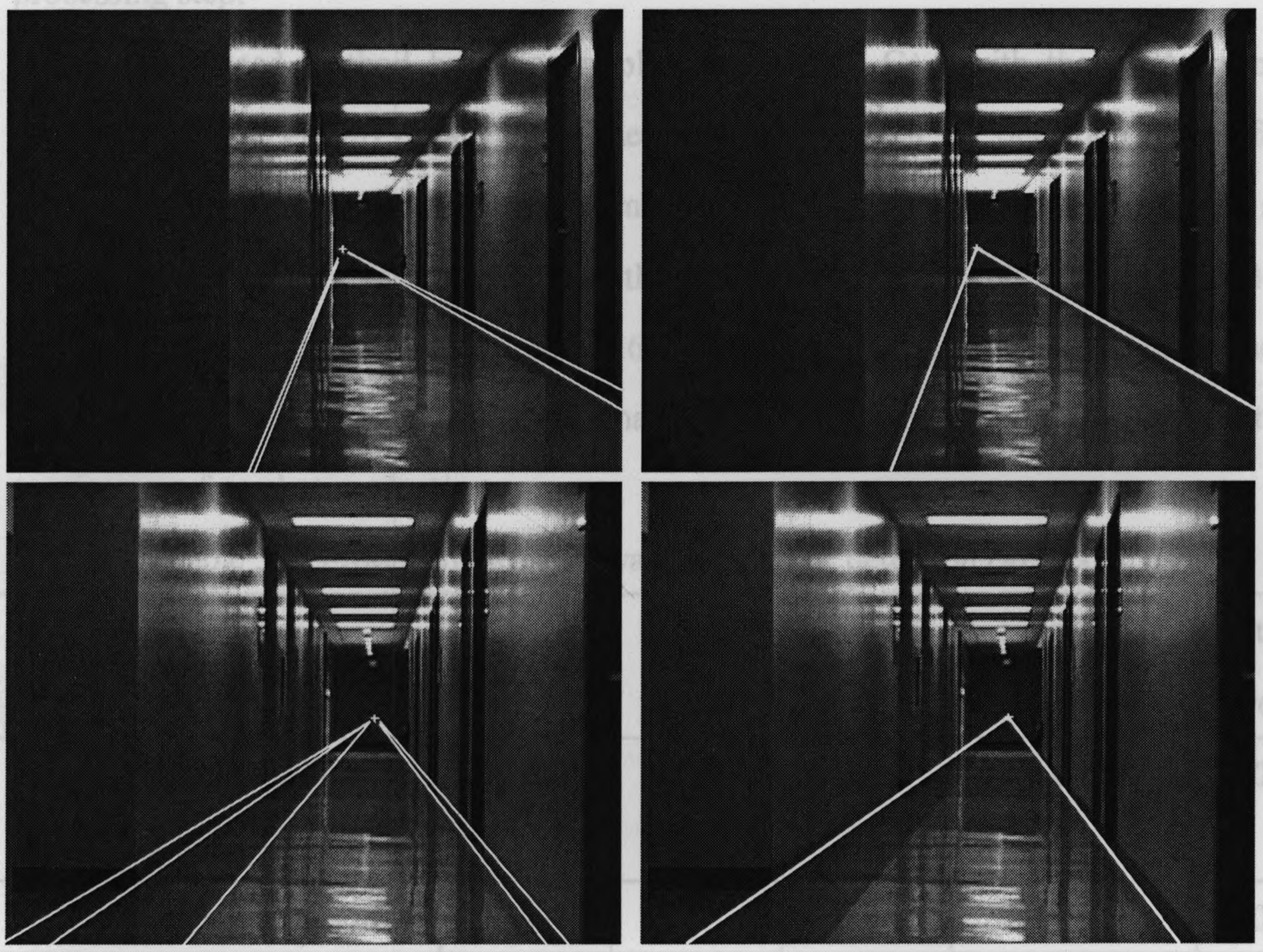
(a)

(b)

Figure 5.5: Detecting results in the presence of people and strong reflections (the last one) (a) Detected vanishing point and generated corridor line hypotheses (b) Detected corridor line locations

5.1.3 Illustration of results for different robot positions inside a corridor

This experiment has been performed for the robot located at different positions inside a corridor, such as close to one side of the wall or in the center of the corridor. Figure 5.6 shows the detecting results of a corridor in which the robot is placed in different places. This is on the third floor, TEB.



(a) (b)

Figure 5.6: Processing results when the robot is positioned in the different places in the corridor: close to the left side of the wall and close to the right side of the wall (a) Detected vanishing point and generated corridor line hypotheses (b) The detected true corridor line locations.

5.2 Performance Evaluation and Discussion

5.2.1 Vanishing point detection

In corridor line detection algorithm, vanishing point (VP) can be quickly and robustly detected at the intermediate step of the procedure. This is useful for robot self-localization and navigation, and also provides necessary information for the further processing step.

The proposed method has been implemented using C++ with the Intel OpenCV (Open Computer Vision) library [76]. The execution time for VP detection ranges from 0.04 to 0.3 seconds depending on the number of features in the image for an image size of 640×480 pixels. Table 5.1 summarize the processing time for 100 images (with 640× 480 pixel resolution) on a laptop with 1.6 GHz Pentium processor and 1 GB memory. The average VP detection time for 100 images is 0.14 seconds which meets the real time requirement for robot navigation.

Table 5.1: Processing time for vanishing point detection (in seconds)

Processing Steps	Image 1	Image 3	Image 5	Average for 100 images	Std. Dev.
Selection of points belonging to diagonal lines	0.01	0.01	0.01	0.0076	0.004
Hypothesis generation	0.11	0.21	0.05	0.1340	0.086
VP estimation	<0.001	<0.001	<0.001	0.0002	0.001
Total time	0.12	0.22	0.06	0.1418	0.291

Table 5.2 summarizes the performance evaluation for 100 corridor video images. The distance error is defined as the Euclidean distance in pixels between the VP ground truth and the processing result, described by Eq. 5.1. The VP ground truth is determined by the intersection point of the corridor line ground truth on each side. The ground truth of corridor lines is obtained by manually choosing the corridor lines in the image.

$$distance\ error = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \qquad (5.1)$$

where $P = (p_x, p_y)$ is the VP ground truth coordinate in the image, and $Q = (q_x, q_y)$ is the processing result coordinate in the image.

The standard deviation s of a data vector X is defined as Eq. 5.2:

$$s = \left[\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \right]^{\frac{1}{2}} \qquad (5.2)$$

where $X = \{x_1, x_2, \ldots, x_n)$ is the distance error vector of VP, and $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$.

Table 5.2: Performance evaluation for the detection of VP for 100 images

(Processed image size: 640×480 pixels)

	Image 1	Image 3	Image 5	Average for 100 images	Stand deviation
Distance error(pixels)	1.4569	3.0815	2.0413	3.2146	3.8658

From Table 5.2, we can see that the detected VP is very close to the true VP. We believe that this error is acceptably small for an indoor robot navigation task and are currently working on utilizing this algorithm for such an application.

5.2.2 Corridor line location detection

Since there have been no attempts reported in the literature to detect true corridor line locations in the presence of some spurious lines, there have been no existing methods that quantitatively evaluate this kind of algorithm. Here, we propose an evaluation metric to compare our detecting results with the ground truth.

As mentioned above, the ground truth of a corridor line was obtained by manually choosing the corridor line in the image. The angular deviation is the angle difference between the detected corridor line and the ground truth. The area deviation is calculated by Eq.5.3. The area error is calculated as shown in Figure 5.7. The numerical results of experiments are illustrated in Table 5.3. Ideally, the angular deviation should be zero, and the error deviation on each side of the corridor should be zero.

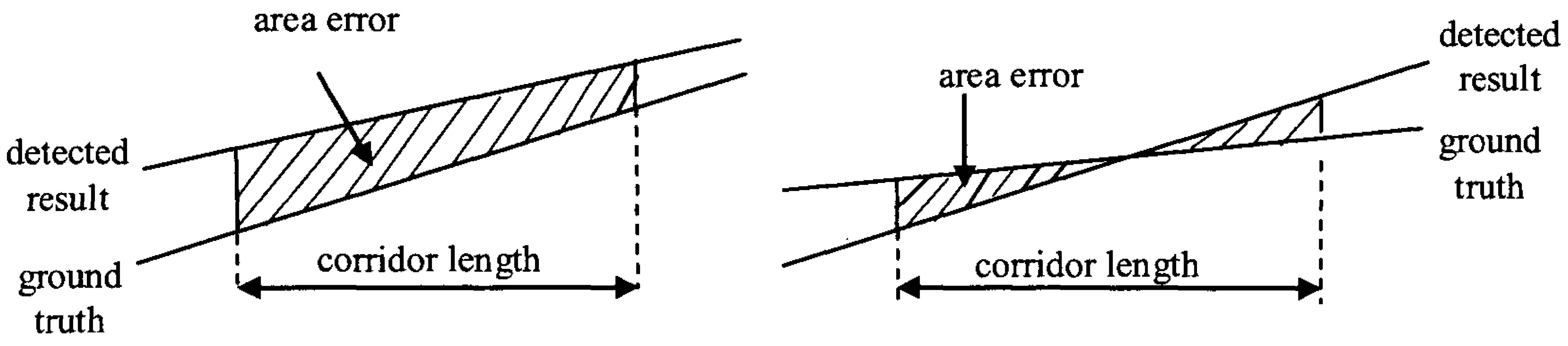


Figure 5.7: Illustration of the area error between the detected result and the ground truth.

$$area\ deviation = \frac{area\ error}{corridor\ length} \tag{5.3}$$

Table 5.3: Comparing detected corridor lines to their ground truth

	Left corridor		Right corridor	
Images	Angular deviation (degree)	Area deviation (pixels)	Angular deviation (degree)	Area deviation (pixels)
Image 1	0.009	1.859	0	0
Image 2	0.015	1.421	0.023	3.011
Image 3	0.008	1.042	0.006	1.625
Average for 100 images	0.017	5.196	0.021	5.436
Stand deviation	0.026	7.950	0.028	8.191

Because the error distribution is not Gaussian, standard deviation values shown in Table 5.3 are unusual. Majority of the error values are distributed less than the average error, but some error values greater than the average have big deviation (greater than stand deviation) with respect to the average value.

Table 5.4: Processing time for corridor line detection (in seconds)

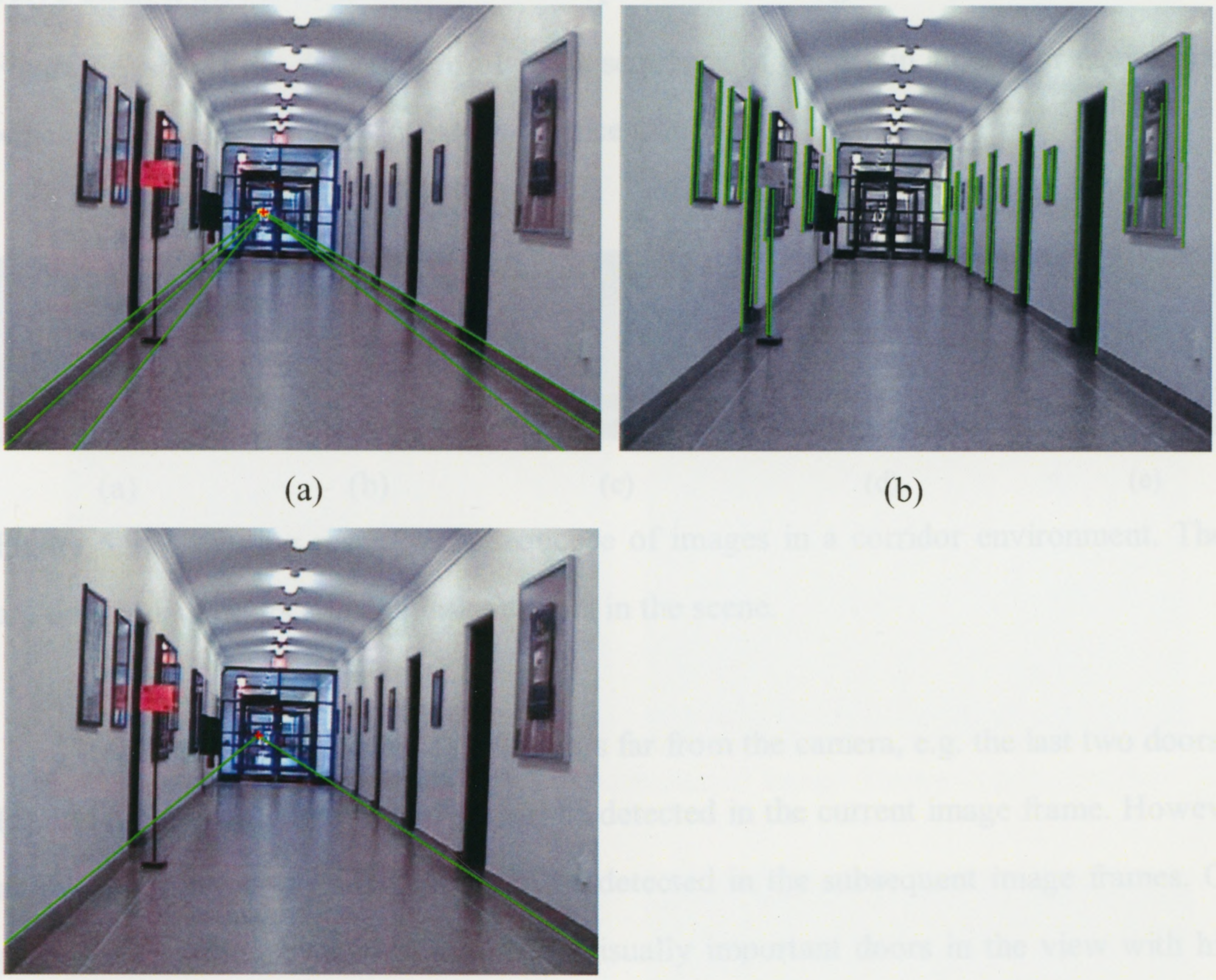
Processing Steps	Image 1	Image 4	Image 5	Average for 100 images	Std. Dev.
Selection of points belonging to diagonal lines	0.01	<0.001	0.01	0.0076	0.0043
Hypothesis generation	0.11	0.291	0.05	0.1340	0.0867
VP estimation	<0.001	<0.0010	<0.001	0.0002	0.0014
Vertical line extraction	1.132	0.6	1.242	0.9361	0.2713
Hypothesis verification	<0.001	<0.001	<0.001	0.0001	0.0010
Total time	1.252	0.891	1.302	1.0782	0.2911

The average processing time of 100 images is about 1 seconds on a laptop with 1.6 GHz Centrino processor and 1 GB memory. These execution times are for 640×480 image. By using more powerful computers and optimizing the implementation, it is possible to reduce the execution time significantly.

The efficiency, effectiveness and robustness of the corridor line detection method are validated by the above experimental results.

5.2.3 Effect due to vertical line availability

In some experiments, the detecting result is the upper edge of the baseboard on the wall, rather than the true corridor line. Figure 5.8 shows this case on the left side of the corridor. This is due to the fact that not enough vertical lines fall onto the corridor area on the left side. Using tracking techniques applied on a sequence of image frames to correct this kind of effect is part of our future work.



(a) (b)

(c)

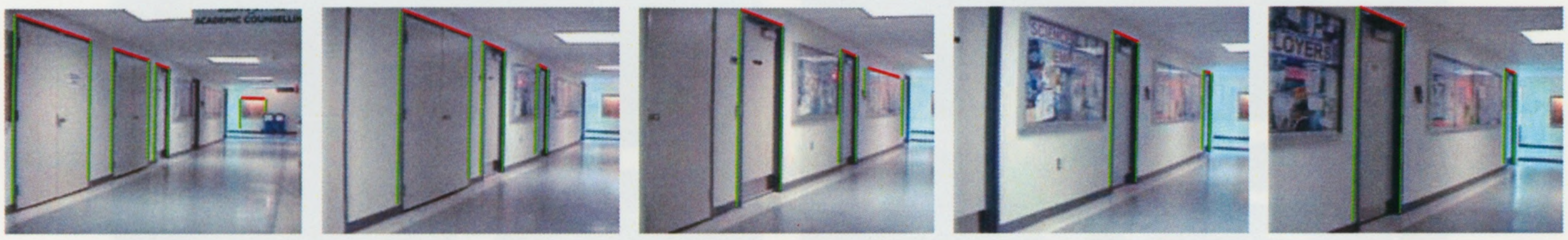
Figure 5.8: Upper edge effect (a) Detected vanishing point and generated corridor line hypotheses (b) Extracted vertical lines (c) The detected corridor line locations.

5.3 Door Detection Experiments

In order to evaluate the performance of the door detection algorithm, 40 video images acquired from 5 different corridor environments have been tested. These image frames cover a wide range of situations: doors more or less close to each other, double-leaf doors, open and closed doors, and partially occluded doors. The experiments validate the effectiveness and robustness of the proposed method.

5.3.1 Illustration of results for a sequence of images in a corridor

Figure 5.9 shows the detecting results of a sequence of images in a corridor environment which is on the first floor of Western Science Center.



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

Figure 5.9: Detecting results of a sequence of images in a corridor environment. There are double-leaf doors and single-leaf doors in the scene.

From Figure 5.9, we can see that doors far from the camera, e.g. the last two doors in (a) and the last one door in (b), cannot be detected in the current image frame. However, as the robot moves forward, they can be detected in the subsequent image frames. Our proposed method can fulfill the task of visually important doors in the view with high accuracy.

5.3.2 Illustration of results for different corridor environments

In order to show the robustness of the door detection method, we have tested it on images of open doors, partially occluded doors without additional modifications. Figure 5.10 illustrates processing results of a sequence of image frames grabbed from the video which was captured on the third floor, University College. These images contain open doors, closed doors, partially occluded doors in the presence of a person.

The processing results for different corridor environments are shown in Figure 5.11. These images were acquired in the following places: the third floor of TEB, the third floor of SEB, and the third floor of University College.

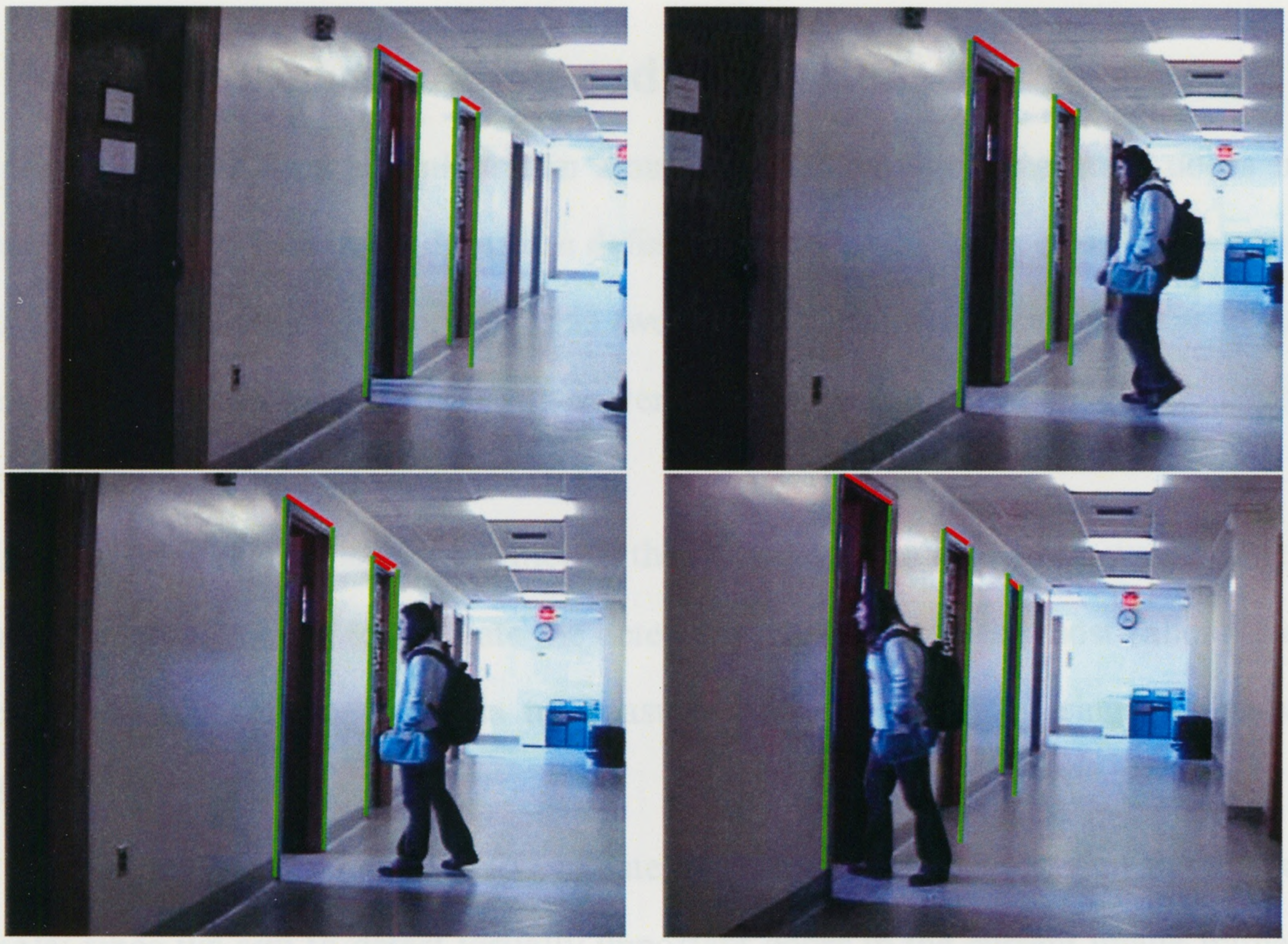


Figure 5.10: Processing results of a sequence of image frames in one corridor.

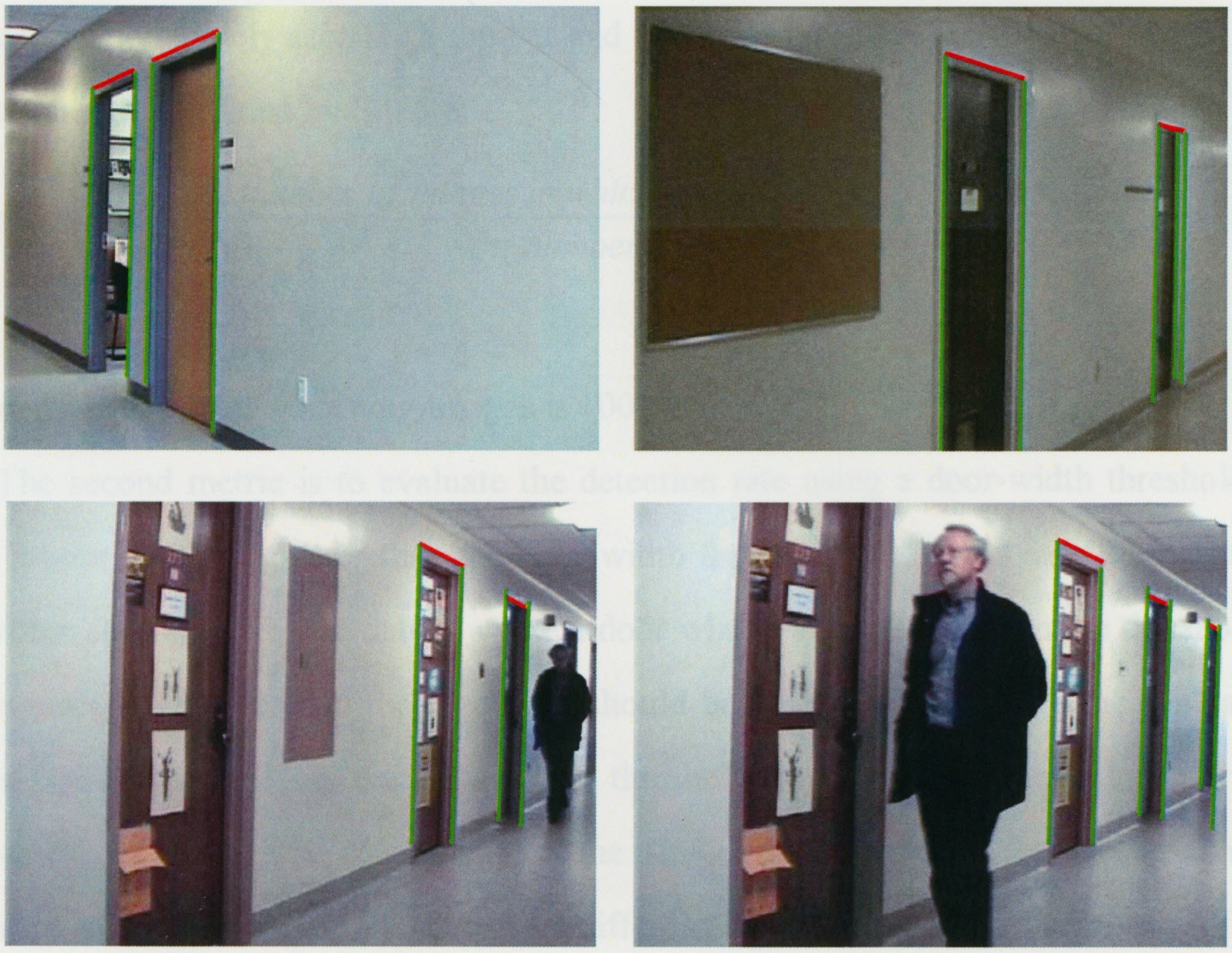


Figure 5.11: Processing results of image frames in different corridor environments.

5.4 Performance Evaluation and Discussion

The proposed door detection algorithm aims to detect visually important doors in an image. The visually important doors are defined as doors which are close to the camera, and can be seen completely (top bars and two boundary lines). Since doors far from the camera are very small and their width is very narrow, these doors are not counted for missed detection in the current image frame.

In order to evaluate the performance of the door detection algorithm, we propose two evaluation metrics: one is to evaluate the detection rate based on the most salient door in the image frame; and the other is to measure the detection rate using a door-width threshold.

The first evaluation metric is to evaluate the detection rate in terms of the following rationale: since we aim to detect visually important doors in the current image frame, for each image we just check if the most salient door (the door that is closest to the camera and can be completely seen with top bar and two boundary lines) is detected or not. The detection rate is defined by Eq. 5.6.

$$\textit{detection rate} = \frac{\textit{the number of images in which the most salient doors are detected}}{\textit{the number of testing images}} \qquad (5.6)$$

The detection rate for 40 testing images is 100%.

The second metric is to evaluate the detection rate using a door-width threshold. A door whose width is greater than the door-width threshold is counted as a door that the algorithm should be able to detect; while a door whose width is less than the threshold is not regarded as a door that the algorithm should be able to detect. Since the evaluation metric would depend on the door-width threshold, we choose three values for the door-width threshold: 2%, 4%, and 5% of the whole image width. We have tested five set of video sequences captured from five different corridor scenes. A summary of the experiment results is shown in Tables 5.5. Table 5.5 shows the actual number of doors within the width-threshold, the correctly detected doors (doors among those detected which are actual doors), and false positives (doors identified by the algorithm but do not belong to the actual doors). The detection rate and the false alarm are also shown in the table, and they are defined by Eq. 5.4 and Eq. 5.5.

$$detection\ rate = \frac{correctly\ detected\ doors}{actual\ doors} \times 100\% \tag{5.4}$$

$$false\ alarm = \frac{false\ positive}{actual\ doors} \times 100\% \tag{5.5}$$

Table 5.5: Results of door detection in five corridor environments

Door-width threshold	Actual doors	Correctly detected doors	False positives	Detection rate (%)	False alarm (%)
2%	96	90	5	93.8	5.21
4%	70	69	4	98.6	5.71
5%	56	56	3	100	5.36

Table 5.6 shows the performance comparison of our proposed method with several other methods reported in the literature.

Tables 5.6: Performance comparison with some existing methods

Author	Method	Number of images	Detection rate (%)	False alarm (%)
Proposed method (4% of the image width)	Feedback based HGV	40	98.6	5.71
Cicirelli *et al.* [34]	Neural network, learning by components	821	92	1.3
Dedeoglu [41]	Color-blob detector	180	92	3
Carinena [28]	Fuzzy temporal rules	Ultrasound information	91	16

Compared with other approaches, our method has a high detection rate as well as a high false alarm. As mentioned in Chapter 1, the presented work aims to detect any potential door structures on left or right walls when a robot navigates along a corridor. Our method acts as the first processing step for detecting doors in real applications. To accurately recognize if it is the target door, we need to extract extra information (e.g. doorplate information) for further confirmation or rejection. Therefore, at current stage we don't want to miss detection of any possible door structures.

In the proposed method, we utilize corridor line evidence for verifying door hypotheses. A detected corridor line which falls onto the corridor area is considered as an accurate detection. The accuracy rate for 40 testing images is 100%.

Experimental results show that our proposed method can detect visually important doors in an image at a very high accuracy rate. Here, we discuss two cases which fail to be detected in the current image frame.

1. Doors too far away from the camera: A door too far away from the video camera looks like two close vertical lines with very narrow top bar edge, due to the viewpoints of the camera. It may not be detected in the current image frame; however, as the robot moves forward, this door will become closer to the camera and visually bigger, and will be identified by the subsequent image frames.
2. Doors too close to the camera: A door too close to the camera would be partially visible (only part of the two boundary lines without top bar edge can be seen). This kind of door cannot be identified in the current image; however this door has already been detected in the previous image frames. A tracking algorithm can be employed to keep tracking and marking the detected doors in the subsequent image frames. In this way, doors can be continuously and successfully detected when a robot navigates along a corridor. Incorporating tracking algorithm in our proposed method is the research work in our future direction.

In conclusion, although doors in above two cases can not be identified in the current image frame, problems can be successfully solved by detecting them in the previous or subsequent image frames.

Chapter 6
Conclusions

6.1 Summary of the Contributions

In this dissertation, we present two algorithms for detecting corridor line and door structures for vision based autonomous mobile robot navigation using a single video camera. A feedback scheme based hypothesis generation and verification (HGV) method and low level line features are utilized to detect these significant structures in the corridor images.

We have tested the two methods on a large number of real video images captured from a variety of corridor environments. Experimental results performed on the corridor line detection algorithm demonstrated that the method is efficient and robust in a variety of corridor environments: under different illumination and reflection conditions, with different robot moving speed, with different position of the robot, and with different contrast of the corridor edge. Experimental results carried on the door detection algorithm shows that the system can detect visually important doors with high accuracy rate in a sequence of corridor images. The door detection method is able to detect closed doors, open doors, partially occluded doors with respect to different viewpoints and different lighting and reflection variations. It successfully discriminates between doors and objects similar to doors, e.g. paintings and posters on the wall in a corridor environment. It can also discriminate between doors and objects dissimilar to doors, such as people, in the corridor scene.

We summarize our major contributions as follows:

- Propose a feedback mechanism based HGV method to indoor structure detection applications, and obtain good performance in improving the low level processing stage.

- Implement and test a corridor line detection method using the proposed HGV method that can:
    a) Detect vanishing point efficiently and robustly for robot navigation.
    b) Robustly detect corridor line locations for robot navigation and scene understanding in real time.
- As far as we know, the corridor line detection algorithm reported in this thesis represents the first attempt to detect true corridor line locations in the presence of many spurious line features. Little research has been done on tackling such situation.
- Implement and test a door detection method using the proposed HGV method that can detect visually important doors with a high accuracy rate when a robot navigates along a corridor.

6.2 Discussion and Future Work

In this section we discuss a number of issues related to our proposed work and identify several future research directions.

- For the corridor line detection algorithm:
    a) There is a slight disparity between the true corridor location and our detection results for some images due to reflections and noise. An energy based line detection algorithm developed in our lab presents a method of solving this problem by making the detected lines cling to the true ones. In the future, we intend to incorporate energy based line detection method to tune the corridor line results, so as to obtain more accurate corridor line location.
    b) In some experiments, the detecting result is the upper edge of the baseboard on the wall, rather than the true corridor line. This is due to the fact that more vertical lines fall onto the upper edge area than the corridor edge area. Using tracking techniques applied on a sequence of image frames to correct this upper edge effect is part of our future work.

c) In our method, we employ vertical lines as evidence to confirm or reject corridor line hypotheses. An issue we need to consider is the case when the vertical line evidences are not available. Further investigations are in the direction of exploring possible approaches that involve searching for extra information using machine leaning based feature detection methods to verify hypotheses.

- For the door detection algorithm:

Our proposed method aims to detect visually important doors in view when a robot navigates along a corridor. It is the first processing step for detecting doors in a corridor environment for the whole robot navigation system. The output of our algorithm provides potential door information for the robot, so that the robot can guide itself to approach the detected object, and adjust the pan and tilt angles of the camera to capture more images around the door from different viewpoint. To accurately recognize if it is the target door, we need to identify the doorplate information by incorporating Text Localization and Extraction Algorithm developed in our lab for further confirmation or rejection. For the current door detection method, we discuss some issues as follows:

a) The proposed method performs well in corridor environments without occlusion or partial occlusion (with one boundary line occluded). Identifying doors in occluded corridor environments will be part of our future work.

b) Our algorithm aims to identify visually important doors in the current image frame. Based on the fact that a door usually spans several frames and can be completely visible in some image frames and partially visible in other frames, an approach that uses a tracking operation in consecutive frames can be a solution to the continuous detection of doors. Using a tracking algorithm we will be able to label the recognized doors and keep tracking them in subsequent image frames. Incorporating tracking algorithm in our proposed method is also planned for future research.

c) In our algorithm we assume that corridors have light colored walls and dark colored doorways. While this is valid for most public buildings, the algorithm needs to be modified if this assumption does not hold. We also assume that an indication of vertical direction is available. Thus, if the camera is tilted, we need to rotate images so that vertical lines in the scene are parallel to the y axis of the image.

Finally, combining the corridor line detection method and door detection method together to detect physical structures for vision based robot navigation remains an issue we will be exploring in the future.

Bibliography

[1] X. Lebegue and J. K. Aggarwal, "Significant line segments for an indoor mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp. 801-815, Dec.1993.

[2] S. Segvic and S. Ribaric, "Determining the absolute orientation in a corridor using projective geometry and active vision," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 3, pp. 696-710, June2001.

[3] R. C. B. M.A.Fischler, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of ACM*, vol. 24, pp. 381-395, June1981.

[4] Chiu.S, "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent & Fuzzy Systems*, vol. 2, no. 3 Sept.1994.

[5] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.

[6] R. Schuster, N. Ansari, and A. Bani-Hashemi, "Steering a robot with vanishing points," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, pp. 491-498, Apr.1993.

[7] G. F. McLean and D. Kotturi, "Vanishing point detection by line clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, pp. 1090-1095, Nov.1995.

[8] E. Lutton, H. Maitre, and J. Lopez-Krahe, "Contribution to the determination of vanishing points using Hough transform," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 430-438, Apr.1994.

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

[10] S. Lili and S. Uchikado, "New visual feedback guidance of a mobile robot via vanishing point," in *2004 IEEE Conference on Robotics, Automation and Mechatronics,* Volume 1 ed 2004, pp. 153-158.

[11] T. Shakunaga, "3-D corridor modeling from a single view under natural lighting conditions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 293-298, Feb.1992.

[12] P. Parodi and G. Piccioli, "3D shape reconstruction by using vanishing points," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 211-217, Feb.1996.

[13] X. Lebegue and J. K. Aggarwal, "Extraction and Interpretation of semantically Significant Line Segments for a Mobile Robot," in *Proc.IEEE Int.Conf.Robotics Automat.* 1992, pp. 1778-1785.

[14] N. Ayache, *Artificial vision for mobile vision stereo vision and multisensory perception*. London: MIT Press, 1991.

[15] D. Nair and J. K. Aggarwal, "Moving obstacle detection from a navigating robot," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 3, pp. 404-416, June1998.

[16] J. J. Guerrero and C. Sagues, "Uncalibrated vision based on lines for robot navigation," *Journal of Mechatronics*, vol. 11, pp. 759-777, 2001.

[17] J.B.Burns, A.R.Hanson, and E.M.Riseman, "Extracting straight lines," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 425-455, July1986.

[18] Kyung-Jin Choi, Sung-Hun Bae, Young-Hyun Lee, and Chong-Kug Park, "A lateral position and orientation estimating algorithm for the navigation of the vision-based wheeled mobile robot in a corridor," in *SICE 2003 Annual Conference,* 3 ed 2003, pp. 2464-2469.

[19] J.Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, June1986.

[20] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and cueves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11-15, 1972.

[21] Myong Ho Kim, Sang Cheol Lee, and Kwae Hi Lee, "Self-localization of mobile robot with single camera in corridor environment," in *IEEE International Symposium on Industrial Electronics,* 3 ed 2001, pp. 1619-1623.

[22] J. A. Shufelt, "Performance evaluation and analysis of vanishing point detection techniques," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 3, pp. 282-288, Mar.1999.

[23] Li Guan, "Sensor-based Robot Navigation and Map Construction," in *http://www.cs.unc.edu/~lguan/COMP290-58.files/FinalReport.htm*.

[24] M. Rous, H. Lupschen, and K.-F. Kraiss, "Vision-Based Indoor Scene Analysis

for Natural Landmark Detection," in *Proc. of IEEE International Conference on Robotics and Automation* 2005, pp. 4642-4647.

[25] M. Mirmehdi, P. L. Palmer, J. Kittler, and H. Dabis, "Feedback control strategies for object recognition," *IEEE Transactions on Image Processing*, vol. 8, no. 8, pp. 1084-1101, Aug.1999.

[26] S. Noronha and R. Nevatia, "Detection and modeling of buildings from multiple aerial images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 501-518, May2001.

[27] P. Vasseur, C. Pegard, E. Mouaddib, and L. Delahoche, "Object detection using perceptual organization and prediction/verification of hypotheses," *IECON 97.*, vol. 3, pp. 1254-1259, Nov.1997.

[28] P. Carinena, C. V. Regueiro, A. Otero, A. J. Bugarin, and S. Barro, "Landmark detection in mobile robotics using fuzzy temporal rules," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 423-435, Aug.2004.

[29] M. R. Masliah and R. W. Albrecht, "The mobile robot surrogate method for developing autonomy," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 2, pp. 314-320, Apr.1998.

[30] A. Tapus, G. Ramel, L. Dobler, and R. Siegwart, "Topology learning and recognition using Bayesian programming for mobile robot navigation," in *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robotics and Systems* 2004, pp. 3139-3144.

[31] S. A. Stoeter, F. Le Mauff, and N. P. Papanikolopoulos, "Real-time door detection in cluttered environments," in *Proceedings of the 2000 IEEE International Symposium on Intelligent Control* 2000, pp. 187-192.

[32] D. Anguelov, D. Koller, E. Parker, and S. Thrun, "Detecting and modeling doors with mobile robots," in *IEEE International Conference on Robotics and Automation,* 4 ed 2004, pp. 3777-3784.

[33] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," in *Twelfth National Conference on Artificial Intelligence* 1994.

[34] G.Cicirelli, T.D'orazio, and A.Distante, "Target recognition by components for mobile robot navigation," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 15, no. 3, pp. 281-297, 2003.

[35] M. Bishay, R. A. Peters, II, and K. Kawamura, "Object detection in indoor scenes using log-polar mapping," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation,* 1 ed 1994, pp. 775-780.

[36] C. Eberst, M. Andersson, and H. I. Christensen, "Vision-based door-traversal for autonomous mobile robots," in *Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 1 ed 2000, pp. 620-625.

[37] D. Kim and R. Nevatia, "A method for recognition and localization of generic objects for indoor navigation," in *IEEE Workshop on Applications of Computer Vision* 1994, pp. 280-288.

[38] D.Kim and R. Nevatia, "Recognition and localization of generic objects for indoor navigation using functionality," *Image and Vision Computing*, vol. 16, pp. 729-743, 1998.

[39] E. Cokal and A. Erden, "Development of an image processing system for a special purpose mobile robot navigation," in *The Fourth Annual Conference on Mechatronics and Machine Vision in Practice* 1997, pp. 246-252.

[40] G. Dedeoglu, M. Mataric, and G. S. Sukhatme, "Incremental,on-line topological map building with a mobile robot," in *Photonics East Conference - Mobile Robots XIV-SPIE 99, Boston, MA* 1999, pp. 129-139.

[41] G. Dedeoglu and G. S. Sukhatme, "Landmark-based matching algorithm for cooperative mapping by autonomous robots," in *Fifth International Symposium on Distributed Autonomous Robotic Systems (DARS 2000).*

[42] M. Tomono and S. Yuta, "Mobile robot navigation in indoor environments using object and character recognition ," in *Proceedings.ICRA '00.IEEE International Conference on Robotics and Automation* 2000, pp. 313-320.

[43] D. Nikovski, "Visual memory-based learning for mobile robot navigation," in *Second International Conference on Computational Intelligence and Neurosciences, Research Triangle Park, North Carolina,* 2 ed 1997, pp. 1-4.

[44] B.Draper, A.Hanson, and E.Riseman, "Knowledge-Directed Vision: Control, Learning, and Integration," *Proceedings of the IEEE*, vol. 84, no. 11, pp. 1625-1637, Nov.1996.

[45] R. A. Brooks, "Model based three-dimensional interpretations of two-dimensional images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 140-149, 1983.

[46] R.Haralick and L.Shapiro, *Computer and Robot Vision* Reading, MA: Addision-Wesley, 1992, pp. 500-503.

[47] J.W.Lee and I.S.Kweon, "Extraction of line features in a noisy image," *Pattern Recognition*, vol. 30, no. 10, pp. 1651-1660, 1997.

[48] D.S.Guru, B.H.Shekar, and P.Nagabhushan, "A simple and robust line detection algorithm based on small eigenvalue analysis," *Pattern Recognition Letters 25*, pp. 1-13, 2004.

[49] R.Nevatia and K.R.Babu, "Linear feature extraction and description," *Computer Vision, Graphics, and Image Processing*, vol. 33, pp. 257-269, 1980.

[50] A.Mansouri, S.Malowany, and M.D.Levine, "Line detection in digital pictures: A hypothesis prediction/verification paradigm," *Computer Vision, Graphics, and Image Processing*, vol. 40, pp. 95-114, 1987.

[51] Y.T.Zhou, V.Venkateswar, and R.Chellappa, "Edge detection and linear feature extraction using a 2-D random field model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 84-95, 1989.

[52] V.S.Nalwa and E.Pauchon, "Edgel aggregation and edge description," *Computer Vision, Graphics, and Image Processing*, vol. 40, pp. 79-94, 1987.

[53] J.Illingworth and J.Kittler, "A survey of the Hough transform," *Computer Vision, Graphics, and Image Processing*, vol. 44, pp. 87-116, 1988.

[54] V.F.Leavers, "Survey - which Hough transform," *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 58, no. 2, pp. 250-264, Sept.1993.

[55] D.H.Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, pp. 111-122, 1981.

[56] T.Van Veenn and F.Groen, "Discretisation Errors in Hough Transform," *Pattern Recognition*, vol. 14, pp. 137-145, 1981.

[57] W.Niblack and D.Petrovic, "On Improving the Accuracy of the Hough Transform: Theory, Simulations and Experiments," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 574-579, 1988.

[58] J.Illingworth and J.Kittler, "The Adaptive Hough Transform," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, pp. 690-698, 1987.

[59] P.Palmer, J.Kittler, and M.Petrou, "Using Focus of Attention with the Hough

Transform for Accurate Line Parameter Estimation," *Pattern Recognition*, vol. 27, pp. 1127-1134, 1994.

[60] N.Kiryati, Y.Eldar, and A.M.Bruckstein, " A Probabilistic Hough Transform," *Pattern Recognition*, vol. 24, pp. 303-316, 1991.

[61] L.Xu, E.Oja, and P.Kultanen, "A New Curve Detection Method: Randomized Hough Transform," *Pattern Recognition Letters*, vol. 11, pp. 331-338, 1990.

[62] B.Gatos, S.J.Perantonis, and N.Papamarkos, "Accelerated Hough Transform Using Rectangular Image Decomposition," *Electronics Letters*, vol. 32, no. 8, pp. 730-732, 1996.

[63] S.J.Perantonis, B.Gatos, and N.Papamarkos, "Block Decomposition and Segmentation for Fast Hough Transform Evaluation," *Pattern Recognition*, vol. 32, no. 5, pp. 811-824, 1999.

[64] D.Ben-Tzvi and M.B.Sandler, "A Combinatorial Hough Transform," *Pattern Recognition Letters*, vol. 11, no. 3, pp. 167-174, 1990.

[65] Y.Zhang and R.Webber, "A Windowing Approach to Detecting Line Segments Using Hough Transform," *Pattern Recognition*, vol. 29, pp. 255-265, 1996.

[66] D.Leung, L.Lam, and W.Lam, "Diagonal Quantization of the Hough Transform," *Pattern Recognition Letters*, vol. 14, pp. 181-189, 1993.

[67] W.Lam, L.Lam, K.Yuen, and D.Leung, "An Analysis on Quantizing the Hough Space," *Pattern Recognition Letters*, vol. 15, pp. 1127-1135, 1994.

[68] A. Bonci, T. Leo, and S. Longhi, "A Bayesian approach to the Hough transform for line detection," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 35, no. 6, pp. 945-955, Nov.2005.

[69] R.C.Nelson, "Finding line segments by stick growing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 519-523, May1994.

[70] Anil K.Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, New Jersey 07632: Prentice Hall Inc., 1989.

[71] P.Kahn, L.Kitchen, and E.M.Riseman, "A fast line finder for vision-guided robot navigation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 11, pp. 1098-1102, Nov.1990.

[72] D.H.Ballard and C.M.Brown, *Computer Vision*. Englewood Cliffs, New Jersey:

Prentice-Hall Inc., 1982.

[73] R.C.Gonzalez and R.E.Woods, *Digital Image Processing*, 2nd ed. Upper Saddle River, New Jersey 07458: Prentice-Hall Inc., 2002.

[74] C.Ronse, *Connected Components Algorithm for Binary Images*. New York: Research Studies Press, 1984.

[75] J.S.Lee, "Digital Image Smoothing and the Sigma Filter," *Computer Vision, Graphics, and Image Processing 24*, pp. 255-269, 1983.

[76] "Intel Open Source Computer Vision Library," available on *http://www.sourceforge.net/projects/opencvlibrary*.