

SEGMENTATION OF FLOORS IN CORRIDOR IMAGES FOR MOBILE ROBOT NAVIGATION

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

by
Yinxiao Li
August 2011

Accepted by:
Dr. Stanley Birchfield, Committee Chair
Dr. Adam Hoover
Dr. John Gowdy

ABSTRACT

This thesis presents a novel method of floor segmentation from a single image for mobile robot navigation. In contrast with previous approaches that rely upon homographies, our approach does not require multiple images (either stereo or optical flow). It also does not require the camera to be calibrated, even for lens distortion. The technique combines three visual cues for evaluating the likelihood of horizontal intensity edge line segments belonging to the wall-floor boundary. The combination of these cues yields a robust system that works even in the presence of severe specular reflections, which are common in indoor environments. The nearly real-time algorithm is tested on a large database of images collected in a wide variety of conditions, on which it achieves nearly 90% segmentation accuracy.

Additionally, we apply the floor segmentation method to low-resolution images and propose a minimalistic corridor representation consisting of the orientation line (center) and the wall-floor boundaries (lateral limit). Our study investigates the impact of image resolution upon the accuracy of extracting such a geometry, showing that detection of wall-floor boundaries can be estimated even in texture-poor environments with images as small as 16×12 . One of the advantages of working at such resolutions is that the algorithm operates at hundreds of frames per second, or equivalently requires only a small percentage of the CPU.

DEDICATION

I would like to dedicate my dissertation to my beloved parents, who made all of this possible through the endless words of encouragement and undoubted confidence in me.

ACKNOWLEDGMENTS

First of all, I would most like to express my great appreciation for the immeasurable support and guidance given by Dr. Stan Birchfield, my advisor, who guided me and provided enormous support that made my study and research a lot easier. He always come up with deep insight and sparkling ideas on our research topics, which drawn to the myriad of technological improvements and innovations in the area of computer vision.

Additionally, I also want to express my gratitude to Dr. Adam Hoover and Dr. John Gowdy, not only for their input in the preparation of this dissertation, but also for the many hours of quality instruction they made in my graduate studies leading up to this point.

Last but not the least, I would like to thank all helpful discussion and assistance given by the members from the Dr. Birchfield's computer vision group. Particularly, I would like to thank Dr. Zhichao Chen for help in capturing the corridor image database and algorithm development. Also, I would like to thank Vidya Murali for the helpful discussion in the chapter of geometry estimation from low-resolution images.

TABLE OF CONTENTS

	Page
TITLE PAGE	i
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGMENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
 CHAPTER	
1 Introduction	1
1.1 Motivation	1
1.2 Approach	3
1.3 Previous Work	4
2 Floor Segmentation Algorithm	7
2.1 Detecting line segments	7
2.1.1 Detecting and classifying line segments	7
2.1.2 Pruning line segments	9
2.2 Structure Score	10
2.3 Bottom Score	15
2.4 Homogeneous Score	17
2.5 Detecting the wall-floor boundary	18
2.6 Experimental Results	19
3 Applying Floor Segmentation to Estimating Minimalistic Corridor Geometry	26

Table of Contents (Continued)

	Page
3.1 Orientation Line Estimation	26
3.1.1 Median of bright pixels	27
3.1.2 Maximum entropy	27
3.1.3 Symmetry by mutual information	28
3.1.4 Combining the measurements	28
3.2 Wall-floor boundary	28
3.2.1 Width of the corridor	29
3.2.2 Lateral position in the corridor	29
3.3 Experimental Results	30
4 Conclusions and Discussion	41
BIBLIOGRAPHY	43

LIST OF TABLES

Table	Page
2.1 Quantitative comparison of our thresholding method with two standard algorithms. Shown are the percentage of images without spurious pixels on the floor.	15

LIST OF FIGURES

Figure	Page
1.1 Typical corridor images are shown. Strong reflections, shadows, dark areas, and low-contrast in color are commonly seen, which bring difficulties to vision-based mobile robot navigation.	2
1.2 A typical corridor image at seven different resolutions. Even in resolutions as low as 32×24 , it is easy to see the structure of the corridor. For display purposes, we resample the downsampled images so that all of the seven different resolution images are shown in the same size.	3
2.1 Flowchart of the proposed method for floor segmentation.	8
2.2 The wall-floor boundary of typical corridor images is difficult to determine due to strong reflections and shadows. TOP: Two images, with the result of the modified Douglas-Peucker line fitting algorithm applied to Canny edge detection overlaid. The line segments are classified into two groups: vertical (blue) and horizontal (yellow). BOTTOM: Line segments have been pruned according to length and the vanishing point, as described in the text, to reduce the influence of reflections and shadows (Best viewed in color).	10
2.3 TOP-LEFT: A typical corridor image. TOP-RIGHT: Pixels with gradient magnitude greater than a threshold are shown in white. BOTTOM-LEFT: Using the separating curve from Figure. 2.4, the edge pixels mostly follow the boundaries of the wall, door frames, and floor. BOTTOM-RIGHT: The original image thresholded by a value determined by the separating curve, thus revealing the structure of the corridor.	12
2.4 Linear-based classification of pixels on horizontal line segments. The x coordinate is the intensity of the pixel, while the y coordinate is its gradient magnitude. From the training data, some pixels are edge points (red stars), while others are nonedge points (blue circles). A polynomial kernel based classifier separates the two groups of points by an optimal curve (black). . . .	14
2.5 Results of two standard thresholding algorithms on the same image as the previous figure: Ridler-Calvard [29] (left), and Otsu [27] (right). Notice the spurious pixels on the floor due to reflection and shadows.	15

List of Figures (Continued)

Figure	Page
2.6 Results of our proposed threshold method on the sixteen typical corridor images in Figure 1.1.	16
2.7 Horizontal and vertical line segments detected in two images. The red ellipses highlight the horizontal segments with the highest bottom score, due to their proximity to the bottom endpoints of nearby vertical line segments (Best viewed in color).	17
2.8 The result of graph-based segmentation [11] on two corridor images used in Figure 2.7, with each region assigned a random color. Note that the floor is almost a homogeneous area in both images, while the wall and doors are divided into several smaller regions.	18
2.9 TOP LEFT: Detected wall-floor boundary by our approach. TOP RIGHT: Ground Truth of the floor drawn manually. BOTTOM LEFT: Blue shade area shows the misclassified area. BOTTOM RIGHT: Red shade area indicates the ground truth. Then error rate is $r_{err} = 100 \frac{blue}{red} \%$, where the number of pixels in each colored region are indicated in the numerator and denominator.	20
2.10 Examples of floor successfully detected by our algorithm. Note the variety of floor materials, floor reflectivity, relative pose of the floor with respect to the robot, and lighting conditions (Best viewed in color).	21
2.11 Results of our algorithm on images downloaded from the web (Best viewed in color).	22
2.12 Examples for which our algorithm fails to properly detect the floor. From left to right, the failures are caused by strong texture on the floor, texture on the wall, an overly dark image from poor image exposure, and excessive bright lights at the end of the corridor (Best viewed in color).	22
2.13 TOP ROW Original Images. MIDDLE ROW Failure results from Lee et al. [16]. BOTTOM ROW Our algorithm working on three failure examples (Best viewed in color).	23
2.14 Results of our algorithm on video sequence 1 (Best viewed in color).	24
2.15 Results of our algorithm on video sequence 2 (Best viewed in color).	24

List of Figures (Continued)

Figure	Page
2.16 Results of our algorithm on video sequence 3 (Best viewed in color).	25
3.1 TOP TWO ROWS: The orientation line estimate (vertical green line) for the images shown in Figure 1.2. Down to a resolution of 16×12 , the results remain essentially unchanged. Only at the lowest resolution of 8×6 is the technique unable to recover the orientation line accurately. BOTTOM TWO ROWS: The wall-floor boundary found by the algorithm described in Chapter 2 for the different resolution images of Figure 1.2. The accuracy degrades slightly until the resolution of 32×24 , after which the errors become more pronounced.	32
3.2 The three-line model estimate of the corridor found by combining the orientation line with the wall-floor boundary, on the same images. As before, the structure of the corridor remains intact even in the resolution of 32×24 , with only slight errors visible in 16×12	33
3.3 Corridor structure reconstruction from the wall-floor boundary, displayed as a top-down view. The first row in blue shows the ground truth location of the walls (Cartesian conversion of polar laser readings), and the next 6 rows in red show the reconstruction results from the wall-floor boundaries on different resolution images. Each row represents a different run of the robot in the same corridor, with the robot placed at a different lateral position in the corridor for each run (The position of the robot is shown by the dotted line at 0).	34
3.4 TOP: Normalized root mean square error (NRMSE) for estimating the lateral position of the robot for three runs in a single corridor. The structure was accurately captured in all three cases. BOTTOM: Mean NRMSE for the estimation of the corridor width. There is not much difference in estimation error rates across the different resolutions, and in fact the error drops in some cases for 32×24 and 16×12 sizes due to the removal of noise and artifacts by downsampling.	35
3.5 Comparison between our results (three yellow lines) in the first and the third rows and those of Kong et al. [15] (pink region) in the second and the fourth rows. Our algorithm achieves more accurate estimation of both the orientation line and the wall-floor boundary in indoor scenes, particularly at low resolutions.	36
3.6 Additional results for other corridors, including one without ceiling lights in the first row. In some case, the low-resolution image yields more accurate results.	37
3.7 Minimalistic geometry estimation at four different resolutions. Sequence 1.	38

3.8	Minimalistic geometry estimation at four different resolutions. Sequence 2.	. . .	39
3.9	Minimalistic geometry estimation at four different resolutions. Sequence 3.	. . .	40

Chapter 1

Introduction

1.1 Motivation

Segmenting the floor in an image is an important problem for indoor mobile robot navigation. By detecting which pixels belong to the floor, a mobile robot would be able to avoid obstacles and determine the geometry of the scene. This information would be useful in constructing a map of the environment or to guide the robot within an environment for which a map has already been constructed. Additional reasons for floor segmentation include problems such as computing the size of the available space in a room.

One of the challenges for indoor corridor floor segmentation is the specular reflections that are common on the floor. The reflections on the floor may come from the ceiling lights, door frames, or walls, that even make it difficult sometimes for a human observer to distinguish the floor area. Several typical corridor images are shown in Figure 1.1, illustrating a variety of poses and illumination conditions resulting from the ceiling lights and walls. In addition, some floors are textured while others are uniformly colored, thus further adding to the difficulty of accurate floor segmentation.

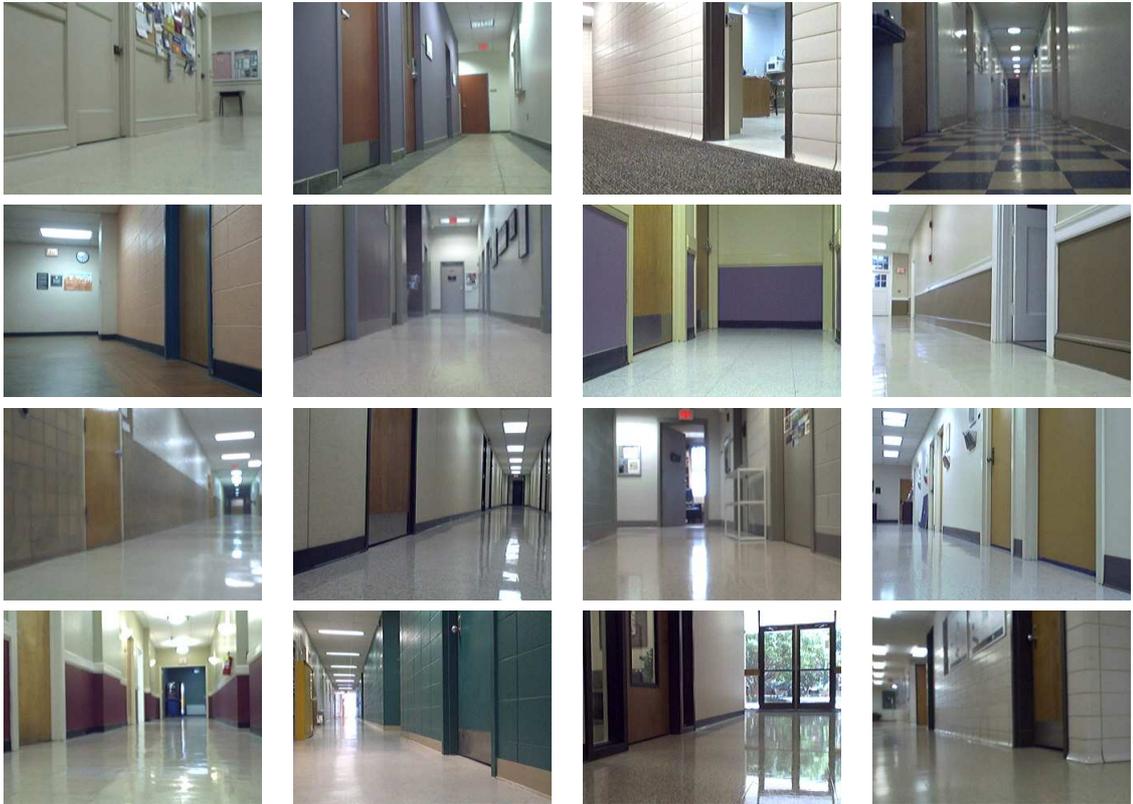


Figure 1.1: Typical corridor images are shown. Strong reflections, shadows, dark areas, and low-contrast in color are commonly seen, which bring difficulties to vision-based mobile robot navigation.

One especially noteworthy contribution of our approach is its proper handling of specular reflections. As seen in the figure above, it is not uncommon for indoor scenes to contain significant amounts of reflection of light off the floor, particularly when the overhead lights are bright, the sun is shining through a window, and/or the floor is particularly shiny. These reflections can confuse homography-based approaches, because they cause pixels on the ground plane to violate the ground plane constraint. Reflections are also known for being difficult to model, causing spurious intensity edges and altering the color appearance of the floor.

In addition, inspired by psychological studies that the human visual system does not require high-resolution images to ascertain information about the environment for basic navigation, also known as the “selective degradation hypothesis” by Leibowitz [17], we apply our floor

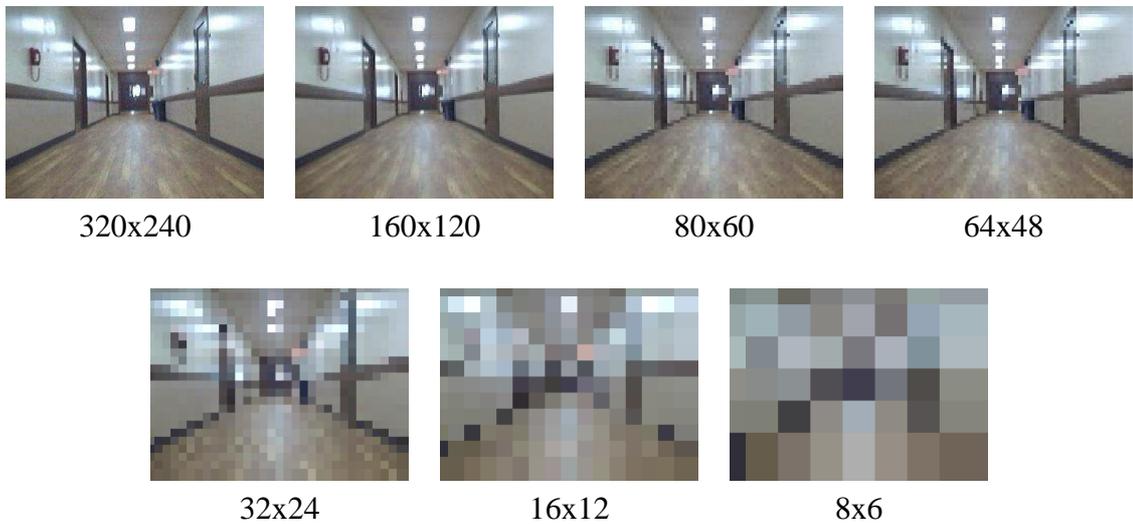


Figure 1.2: A typical corridor image at seven different resolutions. Even in resolutions as low as 32×24 , it is easy to see the structure of the corridor. For display purposes, we resample the downsampled images so that all of the seven different resolution images are shown in the same size.

segmentation to the down-sampled images and surprisingly notice that the results are accurate even in images with a resolution as low as 32×24 . One typical corridor image and its down-sampled images are shown in Figure 1.2.

1.2 Approach

Our floor segmentation method for mobile robot applications only requires a single image. Unlike existing techniques, the approach does not make use of the ground plane constraint and therefore does not use homographies, optical flow, or stereo information. As a result, it does not require the camera to be calibrated, not even for lens distortion. Inspired by the work of McHenry et al. [21] that combines multiple visual cues to detect the difficult material of transparent glass, our technique combines multiple cues to enable the wall-floor separating boundary to be estimated in the image. By combining these visual cues, our approach is often able to ignore the spurious edges, distinguishing between edges arising from the structure of

the scene and those produced by the specular reflections. Another contribution of this work is the introduction of a rather large database of more than 400 indoor corridor images from dozens of different buildings, exhibiting a wide range of conditions. On this challenging dataset, our algorithm is able to successfully detect the floor on nearly 90% of the images. Several results are shown for our floor segmentation in various environments. The algorithm and results are shown and described in Chapter 2.

Floor segmentation algorithm is then applied in Chapter 3 to extract the geometry information from indoor scene. In Chapter 3, we propose a minimalistic representation of a corridor with three lines that capture the center of the corridor, the left wall-floor boundary, and the right wall-floor boundary — the intersection of the orientation (center) line with the wall-floor boundary detected is the vanishing point. To detect this representation, we combine our floor segmentation with Murali and Birchfield [24, 25] that using median of bright pixels, maximum entropy and symmetry to determine the center of the corridor in low-resolution images. Besides showing the results of our floor segmentation, we will also subsequently show the results of our method on low-resolution images, which does not degrade too much with reduction in resolution. The errors seen in estimation of our corridor percepts change little as resolution drops up to 20 times the standard resolution in each direction, effecting a reduction of more than 99% of the original image information.

1.3 Previous Work

A significant amount of research has focused upon the obstacle avoidance problem. In these techniques, the primary purpose is to detect the free space immediately around the mobile robot rather than the specific wall-floor boundary. Most of these approaches utilize the ground plane constraint assumption to measure whether the disparity or motion of pixels matches the values that would be expected if the points lie on the ground plane. Sabe et al. [30] use stereo

cameras to accomplish this task, while the methods of Stoffler [33] and Santos-Victor [31] rely upon optical flow. An alternate approach was pursued by Lorigo et al. [19], who used a combination of color and gradient histograms to distinguish free space from obstacles.

Only a handful of researchers have considered the floor segmentation problem itself. Similar to the obstacle avoidance approaches, the techniques employed tend to utilize the ground plane constraint. Kim and colleagues [14, 7] and Zhou and Li [38] apply planar homographies to optical flow vectors, while Fazl-Ersi and Tsotsos [10] rely on stereo homographies. The approach of [38] computes only a sparse representation of the floor by classifying sparse feature points, while the other two approaches make a pixelwise decision to result in a dense floor representation.

In the computer vision community, some promising results have been achieved recently for related problems. Lee et al. [16], for example, have developed a method that is capable of performing geometric reasoning on a single indoor image. The early line-drawing interpretation work of Guzman [12], they are able to separate the walls from the floor and ceiling using intensity edges and geometric constraints. In another piece of work, Hoiem et al. [13] also assign labels to pixels based upon image data and class priors, primarily for outdoor scenes. Although the results of these approaches are promising, none operate in real time, thus limiting their application to robotics at the moment. Moreover, the work of [16] requires the ceiling to be visible, which is often not the case when the camera is mounted on a mobile robot that is low to the ground.

Regarding to applying our floor segmentation algorithm on low-resolution images, Torralba et al. [35, 34] in their recent work have presented several psychovisual experiments to show that 32×24 bit images are sufficient for human beings to successfully performs basic scene classification, object segmentation, identification . They have shown results on an extensively constructed database of 70 million images that make a powerful argument in favor of low-resolution vision for non-parametric object and scene recognition. Emphasis has been laid on

minimalistic representations for solving visibility based robotic tasks by Tovar et al. [36] and O’Kane and LaValle [26]. Their work focuses on identifying simple and generic solutions to basic robot exploration / navigations tasks.

Chapter 2

Floor Segmentation Algorithm

In this chapter, we describe our algorithm for floor segmentation [18], illustrated in Figure 2.1. For each of the input images, horizontal and vertical line segments are detected. The horizontal line segments are then evaluated as to whether they lie on the wall-floor boundary using three visual cues. Note that our algorithm operates on a single image, without stereo or motion information.

2.1 Detecting line segments

2.1.1 Detecting and classifying line segments

The first step of the approach is to detect intensity edges by applying the Canny edge detector [4] to the grayscale image. Then a robust line fitting method is applied to the intensity edges to obtain a set of line segments. We use the Douglas-Peucker algorithm [8], with the modification described in [6] to improve the retention of small line segments that occur at the bottom edge of doors. Empirically, there is no big difference between the modified version and the original version. But we believe that in some specific cases, it is worthy to detect the concavity of doors to accurately segment the floor. Each line segment is defined by two endpoints in the image.

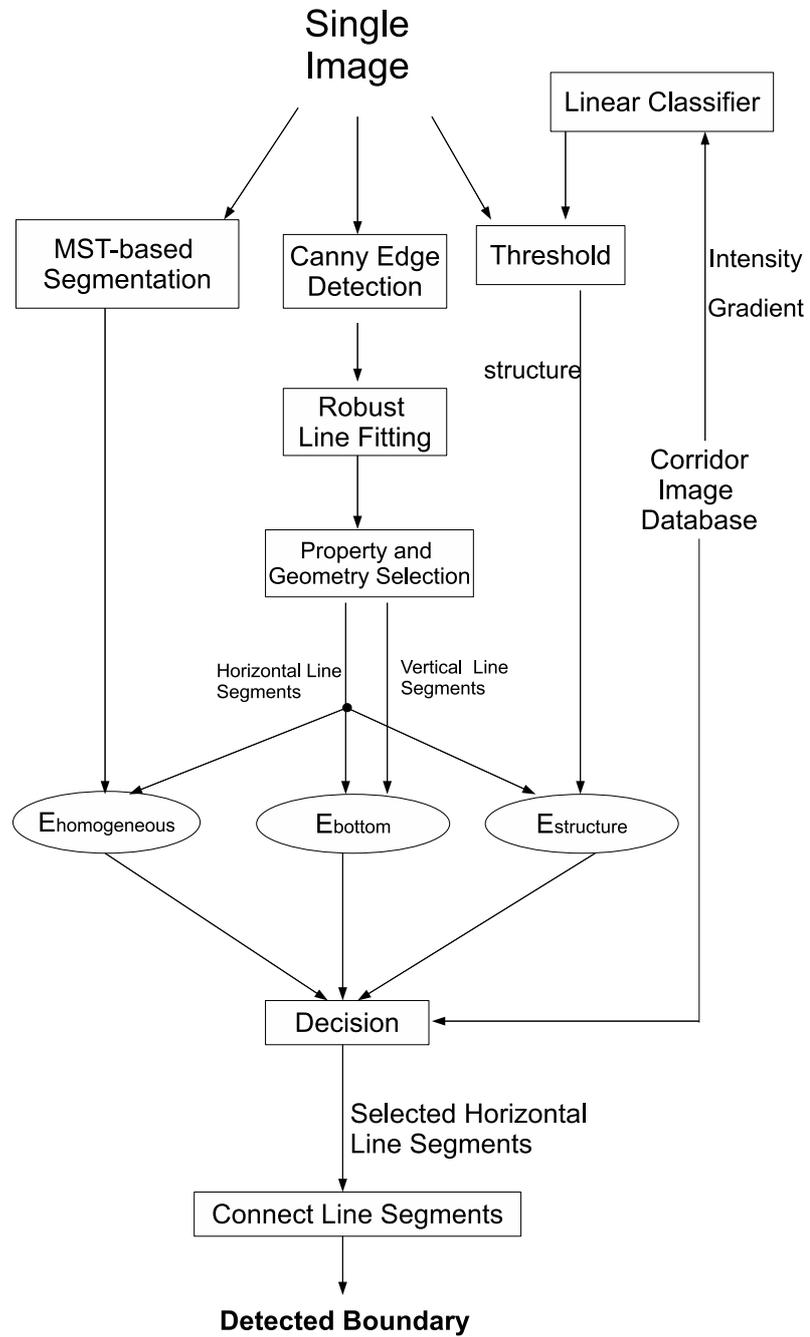


Figure 2.1: Flowchart of the proposed method for floor segmentation.

Line segments are divided into two categories: vertical and horizontal. Based on over three hundred corridor images and the ground truth, we determined a tight slope range to the vertical line segments, so that a line segment is classified as vertical if its slope is within $\pm 5^\circ$ of the vertical direction. Horizontal line segments are given a wider slope range: A line segment is classified as horizontal if its slope is within $\pm 45^\circ$ of the horizontal direction. All other slopes are discarded.

2.1.2 Pruning line segments

Due to the noisy conditions of real-world scenes, the procedure just described often produces spurious line segments that are not related to the wall-floor boundary. We apply two additional steps to prune such segments. First, we discard segments whose length is less than a threshold (60 pixels for vertical lines, and 15 pixels for horizontal lines). Then we compute the intersections of the horizontal line segment pairs, after which we compute the mean of the y coordinate of the intersections inside the image to yield an estimate of the vanishing line. For any pair of horizontal line segments, the intersection point is calculated by the cross product between the two line extensions, using homogeneous coordinates:

$$\begin{bmatrix} wv_x \\ wv_y \\ w \end{bmatrix} = \sum_{i,j} \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} \times \begin{bmatrix} a_j \\ b_j \\ c_j \end{bmatrix}, \quad (2.1)$$

where each horizontal line is described by $ax+by+c=0$, and the intersection point $[v_x \ v_y]^T$ is determined by dividing by the scaling factor w . Once the intersection point has been detected, all horizontal line segments that lie above the vanishing line are discarded. The result of detection, classification, and pruning of line segments is illustrated in Figure 2.2.

Not all of the horizontal line segments that remain from the pruning step will be related to the wall-floor boundary. To determine the likelihood that a horizontal segment ℓ_h is near this

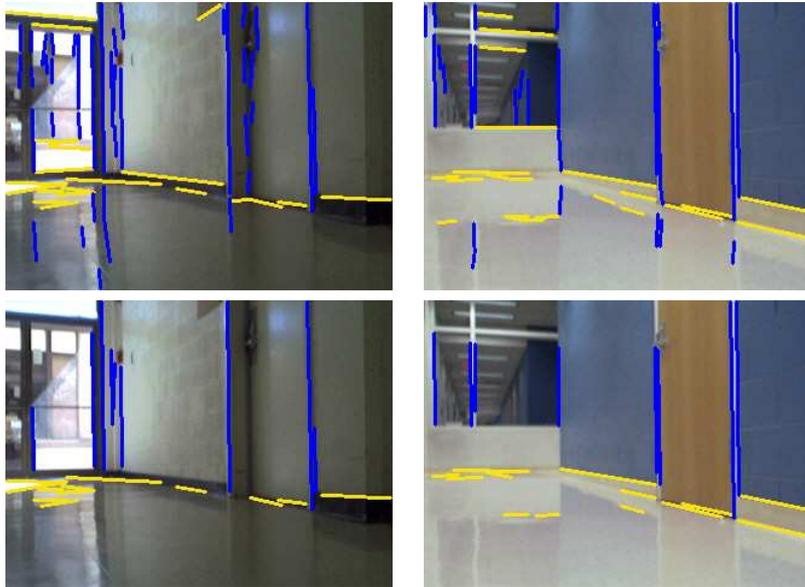


Figure 2.2: The wall-floor boundary of typical corridor images is difficult to determine due to strong reflections and shadows. TOP: Two images, with the result of the modified Douglas-Peucker line fitting algorithm applied to Canny edge detection overlaid. The line segments are classified into two groups: vertical (blue) and horizontal (yellow). BOTTOM: Line segments have been pruned according to length and the vanishing point, as described in the text, to reduce the influence of reflections and shadows (Best viewed in color).

boundary, we compute a weighted sum of scores for three individual visual cues:

$$\Phi_{total}(\ell_h) = w_s \bar{\phi}_s(\ell_h) + w_b \bar{\phi}_b(\ell_h) + w_h \bar{\phi}_h(\ell_h), \quad (2.2)$$

where w_s , w_b , and w_h are the weights, and $\bar{\phi}_s(\ell_h)$, $\bar{\phi}_b(\ell_h)$, and $\bar{\phi}_h(\ell_h)$ are the three individual scores, which are now described.

2.2 Structure Score

We have found that a surprisingly effective cue for distinguishing the walls from the floor in typical corridor environments is to simply threshold the image. This approach works especially well in environments in which the walls are darker than the floor, and it also takes

advantage of the fact that the baseboard is often darker than the floor due either to its painted color, shadows, or collected dirt. In fact, even when the walls, doors, and floor are nearly white, the technique is more effective than one might at first expect. This is partly due, perhaps, to the shadows under the doors that appear no matter the color of the surfaces in the corridor. In some environments, the walls are lighter than the floor, in which case thresholding will still distinguish between the two but with reverse binary labels compared with those situations just described.

The structure image is therefore defined as the binary image resulting from thresholding the image:

$$\mathcal{S}(x, y) = (I(x, y) > \tau_{LC}). \quad (2.3)$$

Since the true edges are generally aligned with the structure of the corridor, we compute the structure score ϕ_s of a line segment ℓ_h by measuring the distance of each pixel in the line segment to the nearest non-zero pixel in the structure image:

$$\phi_s(\ell_h) = \sum_{(x,y) \in \ell_h} d[(x, y), \mathcal{S}], \quad (2.4)$$

where $d[(x, y), \mathcal{S}]$ computes the distance between the point (x, y) and the structure image \mathcal{S} . For fast computation, we use the chamfer algorithm to compute the distance [3]. The score is normalized using a Gaussian distribution with a standard deviation σ_s :

$$\bar{\phi}_s(\ell_h) = \exp \left\{ -\frac{\phi_s(\ell_h)}{2\sigma_s^2} \right\}, \quad (2.5)$$

where $\sigma_s = 10$.

An important step is to determine the value of the threshold τ_{LC} to use. One possible approach would be to use the gradient magnitude. For example, the top-right image of Figure 2.3 shows the pixels whose gradient magnitude exceeds a threshold. The result clearly reveals the

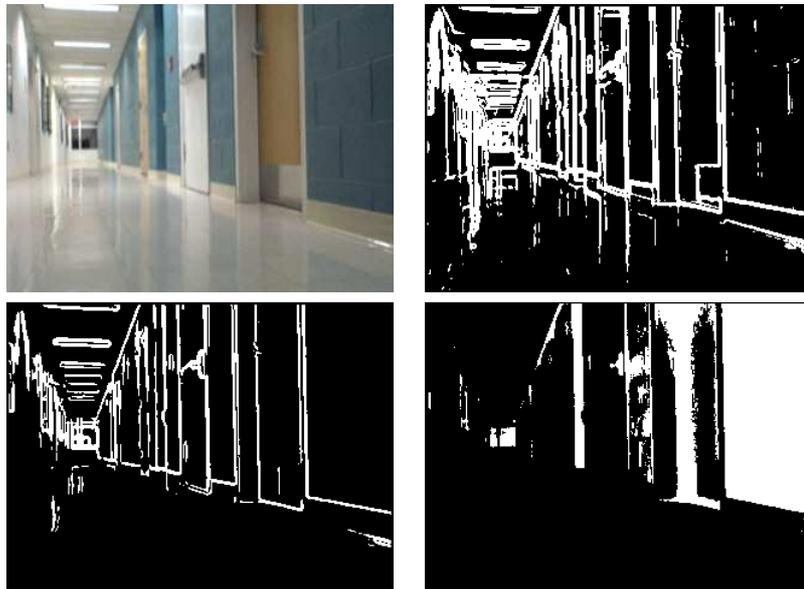


Figure 2.3: TOP-LEFT: A typical corridor image. TOP-RIGHT: Pixels with gradient magnitude greater than a threshold are shown in white. BOTTOM-LEFT: Using the separating curve from Figure. 2.4, the edge pixels mostly follow the boundaries of the wall, door frames, and floor. BOTTOM-RIGHT: The original image thresholded by a value determined by the separating curve, thus revealing the structure of the corridor.

edges of the doors, lights, and wall, so that a human observer looking at this type of image could infer the structure of the scene with little difficulty, as was observed by Lee et al. [16]. As a result, we tried to threshold the image using the average graylevel intensity of these strong edges. However, the distracting intensity edges caused by the reflections on the floor skew the computed threshold in such a way as to reduce the quality of the thresholded image. To fix this problem, we first discard these intensity edges in a manner described in a moment in order to result in a relatively clean thresholded image. The bottom-right of Figure 2.3 shows the result of the improved thresholding procedure.

We now describe our approach to determining the threshold value τ_{LC} . The intensity edges that arise due to reflections on the floor tend to have very high intensity values but quite low gradient magnitude values, the latter being because of the inherent blur that occurs because floors are not perfectly reflective surfaces. To test this hypothesis, we used our database of over

400 images. We manually selected over 800 points on these images that lie on true edges on the walls in the world (i.e., they lie on door frames and so on), and we also randomly selected the same number of points that are not on true edges (i.e., they are on the ceiling or the floor). An SVM-based classifier [5] was used to find the best separating hyperplane to distinguish between the two classes of data using the intensity value and the gradient magnitude of each point. Figure 2.4 shows the training values along with the separating curve. (The hyperplane in the higher dimensional space defined by the polynomial kernel function becomes a curve when projected back into the original feature space.) For all environments, the equation of the SVM-based separating curve is

$$\begin{bmatrix} a & b & 1 \end{bmatrix} \begin{bmatrix} 3.5 & 26 & 475 \\ 26 & 50 & -125 \\ 475 & -125 & 10 \end{bmatrix} \begin{bmatrix} a \\ b \\ 1 \end{bmatrix} = 0, \quad (2.6)$$

where a denotes the intensity I of a pixel and b denotes the gradient magnitude $|\nabla I|$.

From the figure, it is clear that taking both the gradient magnitude and intensity value into account yields better separation than using either alone. We define the set ξ' to denote the pixels in the image whose intensity value and gradient magnitude cause them to lie above and to the left of the SVM-based separating curve.

The average intensity of the pixels in this set determines the threshold that is used for extracting the main structure in a corridor image:

$$\tau_{LC} = \frac{1}{|\xi'|} \sum_{(x,y) \in \xi'} I(x,y). \quad (2.7)$$

For comparison, Figure 2.5 shows the output of two standard algorithms based on the gray-level histogram, Ridler-Calvard [29] and Otsu [27], on the same image. Compared with our approach, the standard techniques mistakenly label reflective pixels on the floor, due to the

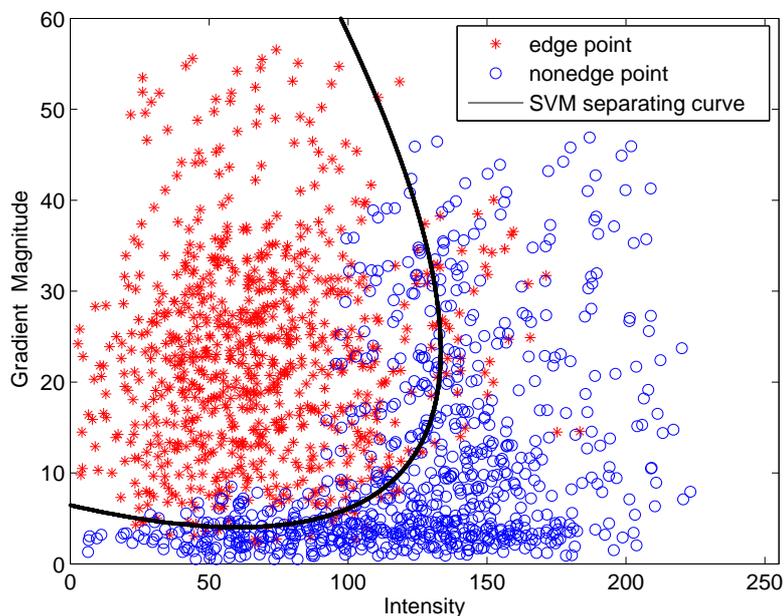


Figure 2.4: Linear-based classification of pixels on horizontal line segments. The x coordinate is the intensity of the pixel, while the y coordinate is its gradient magnitude. From the training data, some pixels are edge points (red stars), while others are nonedge points (blue circles). A polynomial kernel based classifier separates the two groups of points by an optimal curve (black).

failure of the simplified model of a bimodal gray-level histogram to accurately capture the subtle complexities of indoor scenes. Table 2.1 provides a quantitative comparison using the images from our corridor image database. The table shows the percentage of images for which the thresholded result does not contain spurious pixels on the floor. Figure 2.6 shows the results of this procedure on a variety of corridor images. As can be seen, this simple heuristic-based thresholding procedure works surprisingly well on most of the images in our database. However, it fails when the floor is checkered, or the illumination conditions are poor. However, it is important to note that contrast reversals do not pose a problem in practice. In other words, when the floor is darker than the walls the thresholding will cause the floor to be white (in the thresholded image) while the walls are black, but even so, the distance from pixels on horizontal line segments near the wall-floor boundary to the structure image will

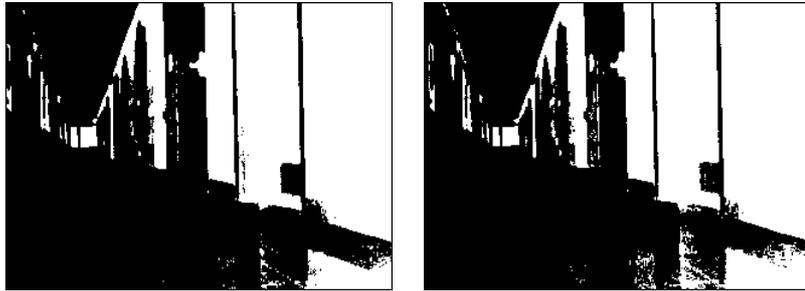


Figure 2.5: Results of two standard thresholding algorithms on the same image as the previous figure: Ridler-Calvard [29] (left), and Otsu [27] (right). Notice the spurious pixels on the floor due to reflection and shadows.

remain low. This is because pixels near the wall-floor boundary are also near the boundary of the structure image, whether the structure image is white on the walls and dark on the floor, or vice versa.

	Ridler-Calvard [29]	Otsu [27]	Ours
correctness	62%	66%	82%

Table 2.1: Quantitative comparison of our thresholding method with two standard algorithms. Shown are the percentage of images without spurious pixels on the floor.

2.3 Bottom Score

The vertical line segments provide an important cue to provide independent evaluation of whether a given horizontal line segment is likely to be on the wall-floor boundary. First, we discard all vertical line segments whose bottom point does not extend below the middle of the image. This step helps to ignore vertical line segments that arise due to texture on the wall or ceiling, since the camera on the robot is low to the ground and facing straight ahead. Then, we sort the remaining vertical segments according to their x coordinate and connect their bottom endpoints to yield a polyline that extends from the left side of the image to the right side. Even though this polyline is a rough approximation of the wall-floor boundary, it is fast to compute and generally reliable enough to help guide the evaluation. The bottom score of a horizontal



Figure 2.6: Results of our proposed threshold method on the sixteen typical corridor images in Figure 1.1.

line segment ℓ_h is computed as the distance of all of its pixels to the polyline ℓ_b :

$$\phi_b(\ell_h) = \sum_{(x,y) \in \ell_h} d[(x,y), \ell_b], \quad (2.8)$$

where $d[(x,y), \ell_b]$ computes the distance between the point (x,y) and the polyline. To normalize the score, we use the Gaussian distribution with a standard deviation σ_b :

$$\bar{\phi}_b(\ell_h) = \exp \left\{ -\frac{\phi_b(\ell_h)}{2\sigma_b^2} \right\}, \quad (2.9)$$

where $\sigma_b = 30$.

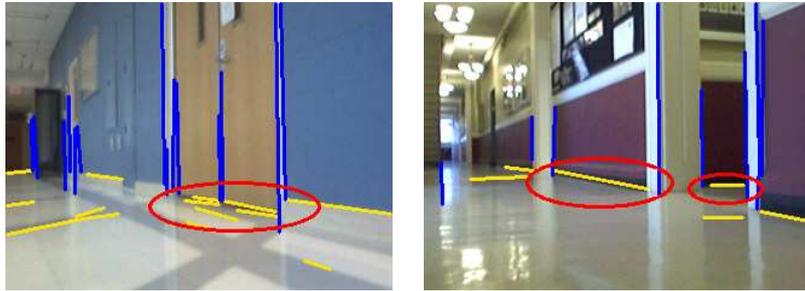


Figure 2.7: Horizontal and vertical line segments detected in two images. The red ellipses highlight the horizontal segments with the highest bottom score, due to their proximity to the bottom endpoints of nearby vertical line segments (Best viewed in color).

Figure 2.7 illustrates this computation for two typical corridor images. The horizontal line segments that benefit most from this computation are highlighted with red ellipses. We can see that this computation is especially helpful for reducing the likelihood of considering line segments on the wall to be part of the wall-floor boundary.

2.4 Homogeneous Score

In many cases, the floor of a typical corridor environment is fairly homogeneous in its color throughout. In contrast, there tend to be moderate to high amounts of texture on the wall regions due to decorations, posters, door knobs, kick plates, nameplates, windows, and so forth. Similarly, the lights in the ceiling cause texture in that region as well. To take advantage of this information, we perform color-based segmentation of the image to support those horizontal line segments which are located just above large homogeneous regions, since the floor is generally the largest homogeneous region in the image. In other words, line segment just above a large homogeneous region are weighted higher.

We employ the MST-based segmentation algorithm of Felzenszwalb and Huttenlocher [11] because it is computationally efficient, requires few parameters (e.g., the minimum size of a region), and produces reasonable results. The results of this algorithm on a couple of typical



Figure 2.8: The result of graph-based segmentation [11] on two corridor images used in Figure 2.7, with each region assigned a random color. Note that the floor is almost a homogeneous area in both images, while the wall and doors are divided into several smaller regions.

corridor images are shown in Figure 2.8. Notice that the floor is the largest homogeneous region in both images, which is often the case in our image database. Occasionally, disturbance from reflection or texture on the floor prevent this cue from being successful, which helps to motivate the need for multiple cues.

The homogeneous score of a horizontal line segment is computed as

$$\bar{\phi}_h(\ell_h) = \frac{|\mathcal{R}|}{|\mathcal{R}_{\max}|}, \quad (2.10)$$

where $|\mathcal{R}|$ denotes the number of pixels in the region \mathcal{R} just below the line segment, and \mathcal{R}_{\max} is the maximum region size among all the segments found by the MST-based segmentation.

2.5 Detecting the wall-floor boundary

Each horizontal segment ℓ_h for which $\Phi_{total}(\ell_h) > \tau_\phi$, where τ_ϕ is a threshold, is retained. Also, we discard overlapped retained line segments by selecting the lowest one. These remaining line segments are then ordered from left to right in the image, and their endpoints are connected. At the left and right borders of the image, the lines are extended. This results in a polyline stretching across the image defining the boundary between the wall and the floor.

2.6 Experimental Results

To test the performance of our algorithm, an image database of more than 400 corridor images was taken in 20 different buildings exhibiting a wide variety of different visual characteristics. The images were captured by a Logitech QuickCam Pro 4000 webcam mounted about 30 cm above the floor on an ActivMedia Pioneer P3AT mobile robot. The images were processed by an algorithm implemented in the C++ programming language on a 2.4 GHz Core 2 processor (Dell XPS M1330 laptop). Although the computation time varies somewhat according to the number of detected line segments, the algorithm runs at approximately 7 frames/sec using unoptimized code.¹

To evaluate the likelihood that a horizontal segment ℓ_h that can be accepted as a part of the wall-floor boundary, we compute a weighted sum of scores for three individual visual cues:

$$\Phi_{total}(\ell_h) = w_s \bar{\phi}_s(\ell_h) + w_b \bar{\phi}_b(\ell_h) + w_h \bar{\phi}_h(\ell_h), \quad (2.11)$$

where w_s , w_b , and w_h are the weights. For all environments, the weights for the individual scores are chosen empirically, which are $w_s = 1.6$, $w_b = 0.75$ and $w_h = 1.0$, respectively, and the total threshold is $\tau_\phi = 2.7$.

To evaluate the algorithm, the images in the database were manually labeled by clicking on a number of points and then fitting a B-spline curve to yield a ground truth wall-floor boundary. We define the error of the algorithm applied to an image as the number of pixels misclassified as floor or non-floor, normalized by the total number of ground truth floor pixels. Equivalently, the error can be computed as the sum, over all the columns $x = 0, \dots, width - 1$ in the image, of the difference between the ground truth y coordinate $y_{GT}^{(x)}$ and the estimated y

¹See http://www.ces.clemson.edu/~stb/research/floor_detection for videos of the results and database.

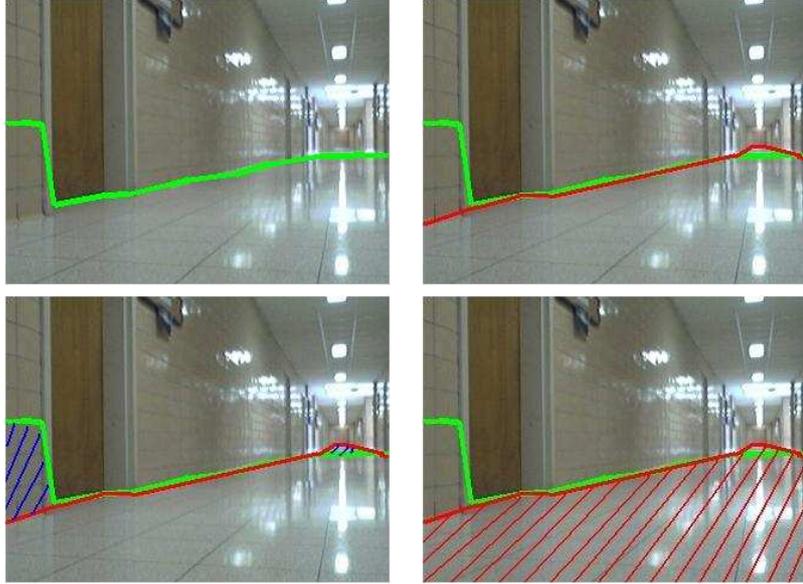


Figure 2.9: TOP LEFT: Detected wall-floor boundary by our approach. TOP RIGHT: Ground Truth of the floor drawn manually. BOTTOM LEFT: Blue shade area shows the misclassified area. BOTTOM RIGHT: Red shade area indicates the ground truth. Then error rate is $r_{err} = 100 \frac{blue}{red} \%$, where the number of pixels in each colored region are indicated in the numerator and denominator.

coordinate $\hat{y}^{(x)}$:

$$r_{err} = \frac{\sum_x |\hat{y}^{(x)} - y_{GT}^{(x)}|}{\sum_x |height - y_{GT}^{(x)}|}, \quad (2.12)$$

where the image is of size $width \times height$, and the subtraction in the denominator arises from the convention that the y coordinate is with respect to the top of the image. We set a threshold of 10%, so that the segmentation for an image is considered a failure for a particular image if $r_{err} > 0.1$ for that image. Using this convention, our approach correctly detects the floor region in 89.1% of the image database. This evaluation method is illustrated in Figure 2.9.

Figure 2.10 presents the results of our algorithm on some typical corridor images. The first row displays wall-floor boundaries that extend upward from left to right in the image, while the second row shows the reverse situation. In the third row, both sides of the corridor are visible, so that the boundary extends in both directions. And the fourth row shows floors with

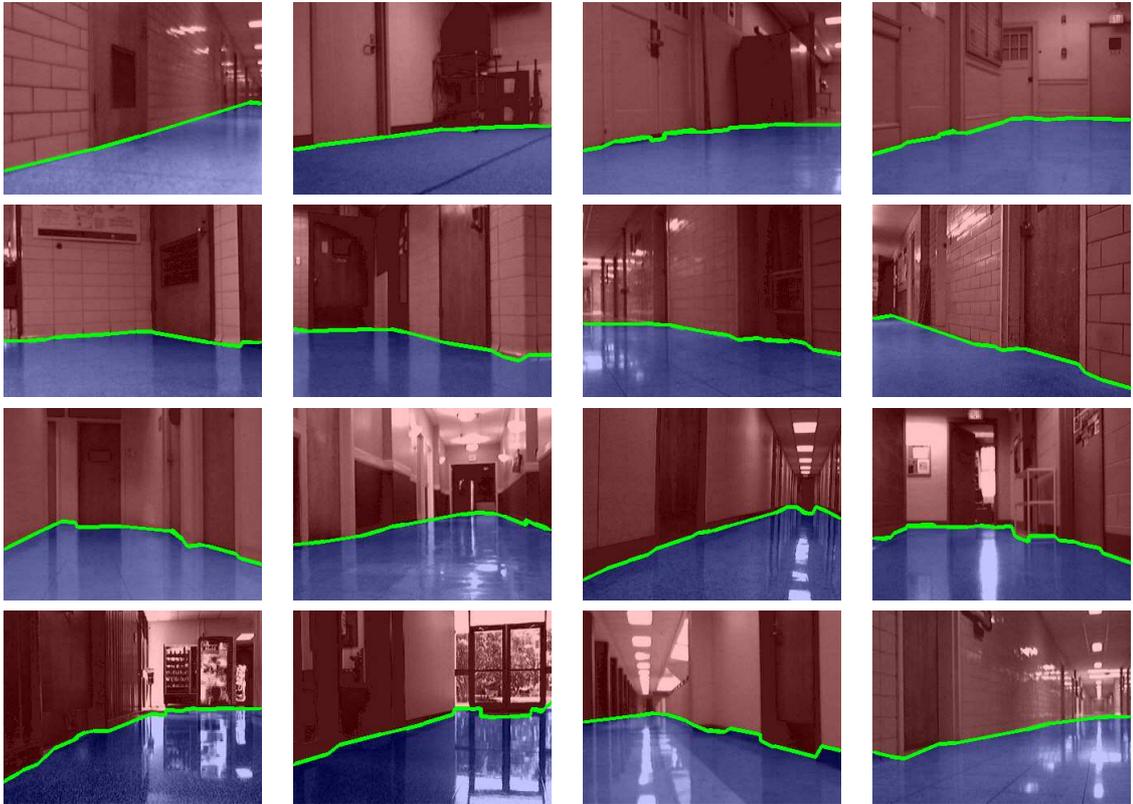


Figure 2.10: Examples of floor successfully detected by our algorithm. Note the variety of floor materials, floor reflectivity, relative pose of the floor with respect to the robot, and lighting conditions (Best viewed in color).

extremely strong reflections on the floor, where the floor and wall are again distinguished using only low-level information. From these results, we can see that our approach is capable of detecting floors in corridors under different illumination conditions and perspectives. In addition, Figure 2.11 shows some successful results on images downloaded from the internet, showing the versatility of the approach. Finally, we apply our algorithm to the video sequences with a low-pass filter added to consecutive frames. Three sample results are shown in Figures 2.6, 2.6, 2.6.

Some examples where the algorithm fails are shown in Figure 2.12. In the first image from the left, the checkered floor lead to many horizontal line segments that are mistakenly interpreted by our score model, and the graph-based segmentation detects many small pieces

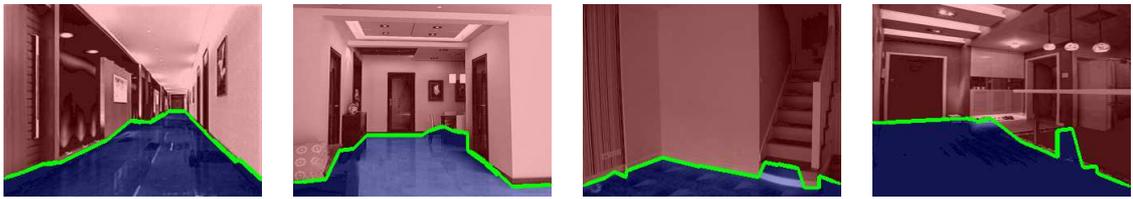


Figure 2.11: Results of our algorithm on images downloaded from the web (Best viewed in color).

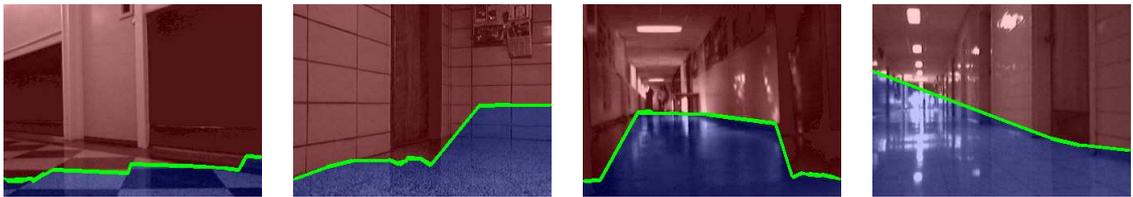


Figure 2.12: Examples for which our algorithm fails to properly detect the floor. From left to right, the failures are caused by strong texture on the floor, texture on the wall, an overly dark image from poor image exposure, and excessive bright lights at the end of the corridor (Best viewed in color).

on the floor region rather than a single homogeneous segment. For the second image, the wall texture also results in many horizontal line segments that distract the algorithm. For the third image, the shadows and reflection dominate the line segment detection due to the poor gain control of the camera, making the final result less precise. And for the fourth image, the far glass door with no absolute edges makes it difficult for even a human observer to precisely locate the wall-floor boundary, though the results of the algorithm are even worse due to the lack of horizontal segments in that region of the image.

It is difficult to compare these results with existing techniques. The work of Hoiem et al. [13] is aimed primarily at outdoor environments, while the recent work of Lee et al. [16] is designed to reconstruct indoor scenes when the ceiling is visible. Neither system is real time. Nevertheless, Figure 2.13 shows some successful results of our algorithm working on three of the failure examples given in the latter paper. Perhaps the closest work to our own in terms of purpose and scope is that of Kim and colleagues [14, 7], which requires two image frames to segment the floor from the rest of the scene. Because their approach does not contain a specific

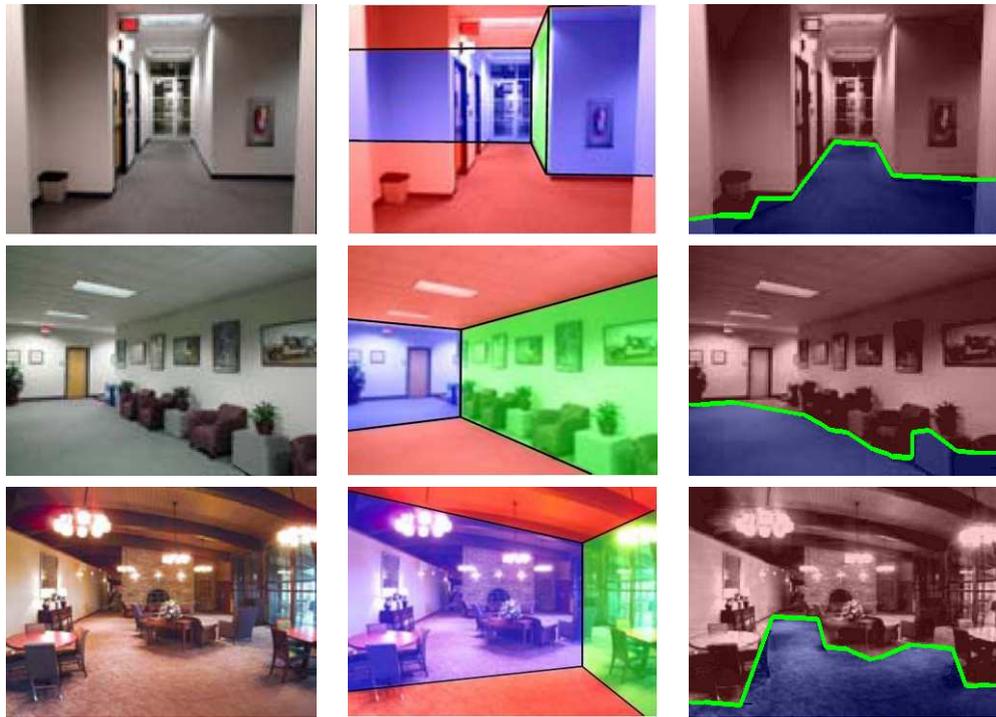


Figure 2.13: TOP ROW Original Images. MIDDLE ROW Failure results from Lee et al. [16]. BOTTOM ROW Our algorithm working on three failure examples (Best viewed in color).

mechanism to handle strong reflections, it is doubtful that it would work successfully on the many images in our database that contain such extreme lighting conditions. Similarly, the obstacle avoidance system of Lorigo et al. [19] is not designed to handle strong reflections. The output of our system on videos of three different corridors are shown in Figures 2.6 through 2.6.

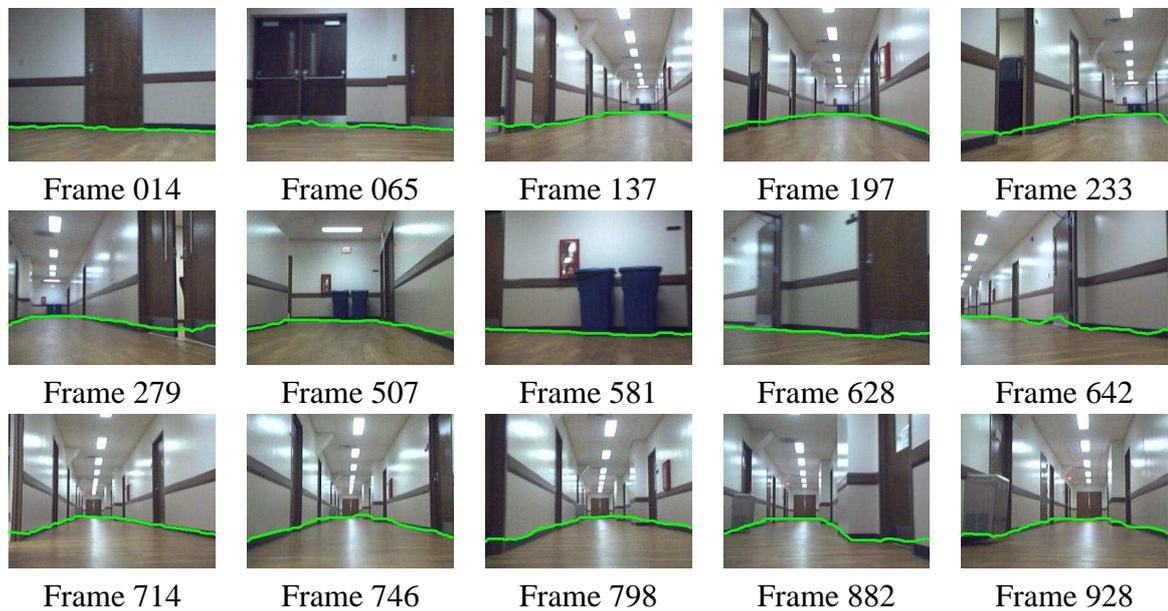


Figure 2.14: Results of our algorithm on video sequence 1 (Best viewed in color).

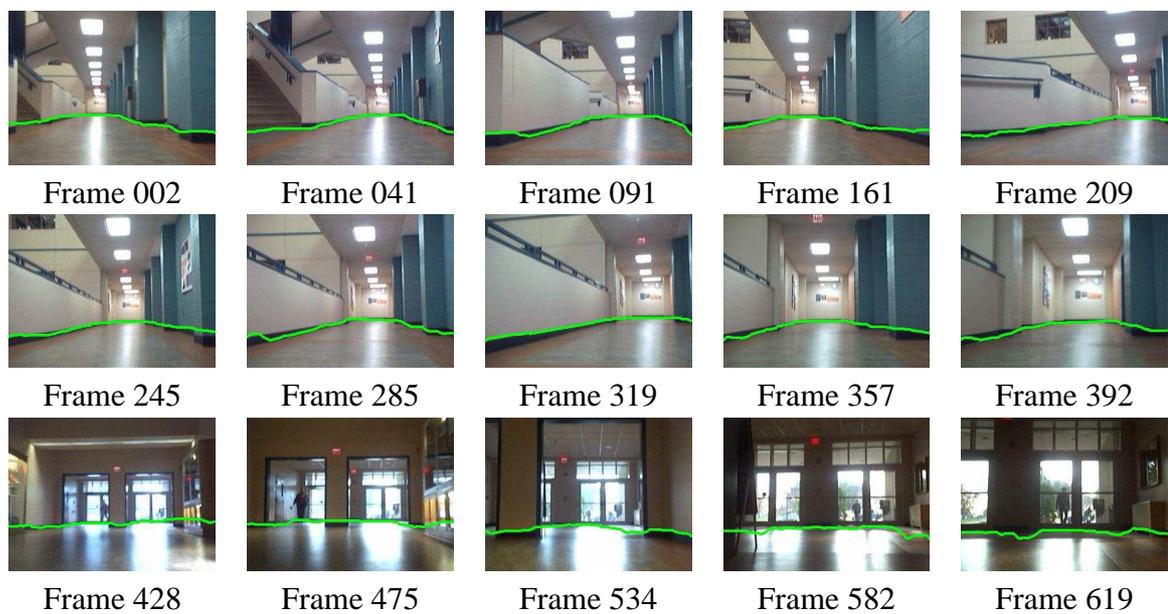


Figure 2.15: Results of our algorithm on video sequence 2 (Best viewed in color).

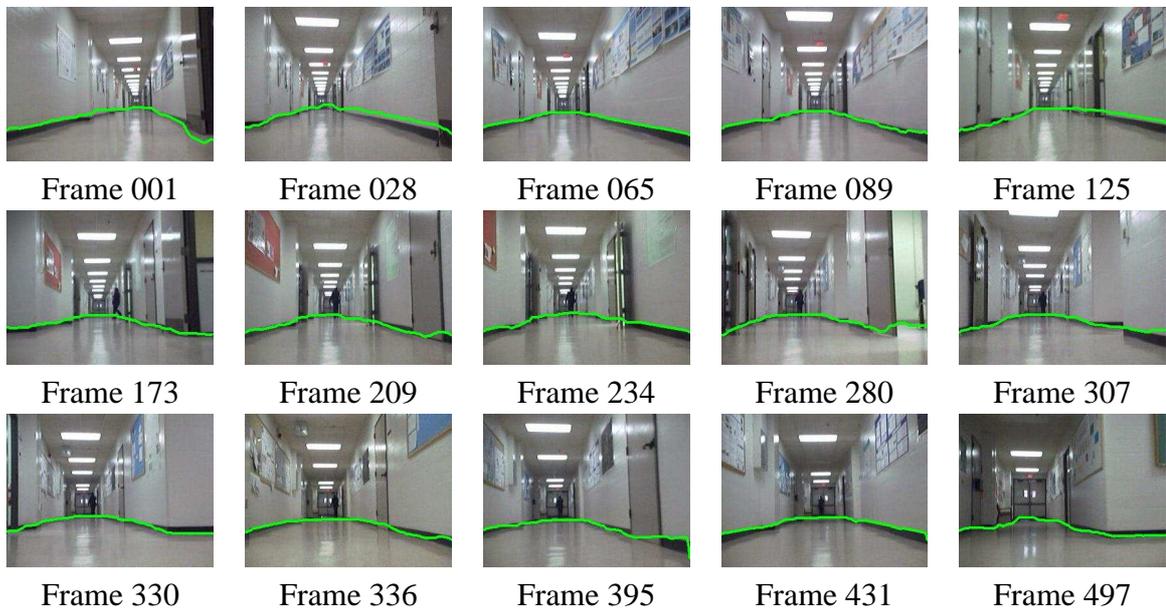


Figure 2.16: Results of our algorithm on video sequence 3 (Best viewed in color).

Chapter 3

Applying Floor Segmentation to Estimating Minimalistic Corridor Geometry

In this chapter, we describe a technique to recover minimalistic geometry representation of an indoor scene [23]. The representation involves three lines separating the scene into left-wall region, right-wall region, and the floor area. The technique combines our floor segmentation algorithm (described in the previous chapter) with an orientation line estimation algorithm [24].

3.1 Orientation Line Estimation

We model the geometry of a corridor by three lines in the image. A vertical line indicates the orientation line, or centerline, of the corridor, which passes through the vanishing point. The wall-floor boundary is then captured by two diagonal lines that meet at the same point on the orientation line. Our approach, then, consists of two steps. First is the estimation of the

orientation line in the image by combining multiple cues. Second is the estimation of the wall floor boundary.

In this section we describe the orientation line estimation by Murali and Birchfield [24]. This approach uses the median of bright pixels (ceiling lights), maximum entropy, and maximum symmetry measures in the corridor image to determine its center and therefore the orientation. This approach has several advantages over existing techniques: It is simple, computationally efficient, and yields good results even for low-resolution images.

3.1.1 Median of bright pixels

The ceiling lights, which are usually symmetric with respect to the main corridor axis, provide an important cue. Two difficulties are the low resolution of the image and the fact that sometimes the lights are not in the center of the corridor but rather on the sides. A simple technique that overcomes these difficulties is to apply the k -means algorithm [20] to the graylevel values the image, with $k = 2$. The median horizontal position of the brighter of the two regions is calculated, yielding an estimate of the center of the corridor.

3.1.2 Maximum entropy

It has been found empirically that, as a general rule, entropy is maximum when the camera is pointing down the corridor [25]. The reason for this perhaps surprising result is that such an orientation causes scene surfaces from a variety of depths to be visible, yielding an increase of image information at this orientation. In contrast, when the robot is turned so that it faces one of the side walls, the range of visible depths is much smaller and therefore the variety of pixel intensities usually decreases. A similar observation has been noted by other researchers in the context of using omnidirectional images [2, 9].

3.1.3 Symmetry by mutual information

Symmetry is defined as the invariance of a configuration of elements under a group of automorphic transformations [37]. In the real world, objects are not perfectly symmetric and moreover, the symmetries are distorted by perspective projection, occlusion, and other imaging effects. Therefore, a continuous value of symmetry is more appropriate, which can be formed by comparing the two regions on either side of the axis using mutual information. Mutual information is a measure of the amount of information that one random variable contains about another random variable.

3.1.4 Combining the measurements

The estimations are combined as a weighted average: $f(I) = \alpha_l(I)f_b(I) + \alpha_h(I)f_e(I) + \alpha_s(I)f_s(I)$, where f_b , f_e , and f_s are normalized value of median of bright pixels, maximum entropy, and symmetry by mutual information, respectively. Because of the reliability of the bright pixels, we set $\alpha_l = 0.8$, $\alpha_h = \alpha_s = 0.1$. Sample results are shown in the first two rows in Figure 3.1.

3.2 Wall-floor boundary

We adapt the floor segmentation method described in Chapter 2 which has been shown to be successfully robust to reflections on the floor. For the seven different resolutions, we compute the minimum acceptance length of the horizontal line segments l_h as $l_h = \ln \eta d$, where $d = \sqrt{w^2 + h^2}$ is the length of the diagonal of the image, w and h are the width and height of the image, respectively, and $\eta = 5$ is a scaling factor. Sample result are shown in the last two rows in Figure 3.1.

According to the floor segmentation method, there are three different scores (structure score, homogeneous score, and bottom score) that contribute to the final wall-floor boundary detection. When applying the method to different resolutions, we notice the structure score always shows the best accuracy while the bottom score always fails when decreasing the resolution. Therefore, we adapt the weights for the three scores according to the resolution so that the $\Phi_{total}(\ell_h)$ is relatively high for line segments near the wall-floor boundary. At the same time, when combining with the orientation line, we compute the intersection of the orientation line and the wall-floor boundary, which is considered as the vanishing point. Then we apply the line-fitting algorithm to both half wall-floor boundaries separated by the vanishing point. Using the slopes and the computed vanishing point, it is easy to find the two terminal points on the image border. Finally, we connect the vanishing point, two terminal points, as well as the orientation line and obtain the structure of the corridor. Some sample results are shown in Figures 3.2.

3.2.1 Width of the corridor

The distance between the two end-points in the wall-floor boundary yields the width of the corridor (in pixels). We use a homography obtained during a calibration process to transform to world coordinates.

3.2.2 Lateral position in the corridor

The position of the orientation line with respect to the wall-floor boundaries will give the lateral position in the corridor. The laser readings were converted from polar to Cartesian to give a top-down measurement of the corridor at every instance of the image collection. A homography between a square pattern on the floor of the corridor and its image (4 points) gives the image to top-down calibration in feet. Several examples are shown in Figure 3.3.

3.3 Experimental Results

Estimating the pose of the robot or the orientation of the robot in a typical indoor corridor is one of the necessary tasks for robot exploration/navigation. While many authors have contributed to this work, by estimating vanishing points in a corridor [32, 1, 22, 28], the emphasis is on clustering detected lines, which perform poorly in low-resolution and textureless images because lines are not easily detected in such images. A more recent approach by Kong et al. [15] approaches the problem similarly but uses texture orientation rather than explicit line detection. In their approach, Gabor filters yield texture estimates, and an adaptive voting scheme allows pixels to decide the confidence of an orientation. Not only is their approach much more computationally intensive than ours, but with indoor low-resolution images the results are significantly less accurate. See Figure 3.5 for some examples.

For orientation, we collected data for 4 different buildings, 8 unique corridors (1 training + 7 for testing). For every unique corridor, at equally spaced intervals along the corridor (15 ft), we rotated the robot from -20° to $+20^\circ$ and collected corresponding odometry (heading), laser readings (span of -90° to $+90^\circ$) and images. We ran the entropy detector, light detector and symmetry detector on the images and compared with ground truth (odometry and/or laser). Since a linear relationship exists between the detected pixel location corresponding to the center of the corridor and the robot orientation as explained in previous sections, we use either the estimate f_l or $(f_h + f_s)/2$.

For wall-floor boundary and corridor reconstruction, we collected data for 11 distinct corridors in 6 different buildings. For each corridor, we drove the robot three times (middle, left, right separated by 1.5 feet) along each corridor and collected images along with their corresponding laser readings (-90° to $+90^\circ$ sweep). The Normalized root mean square error (NRMSE) for estimating the lateral position of the robot for three runs is illustrated in 3.4. Sample results of our algorithm on indoor database are shown in Figure 3.6. The output of

our algorithm on videos of three different corridors are shown in Figures 3.7 through 3.9. Empirically, we found that when approaching the end of a corridor, the angle between the two wall-floor boundary lines increases toward 180° . Therefore, this could be used to detect the end of a corridor.

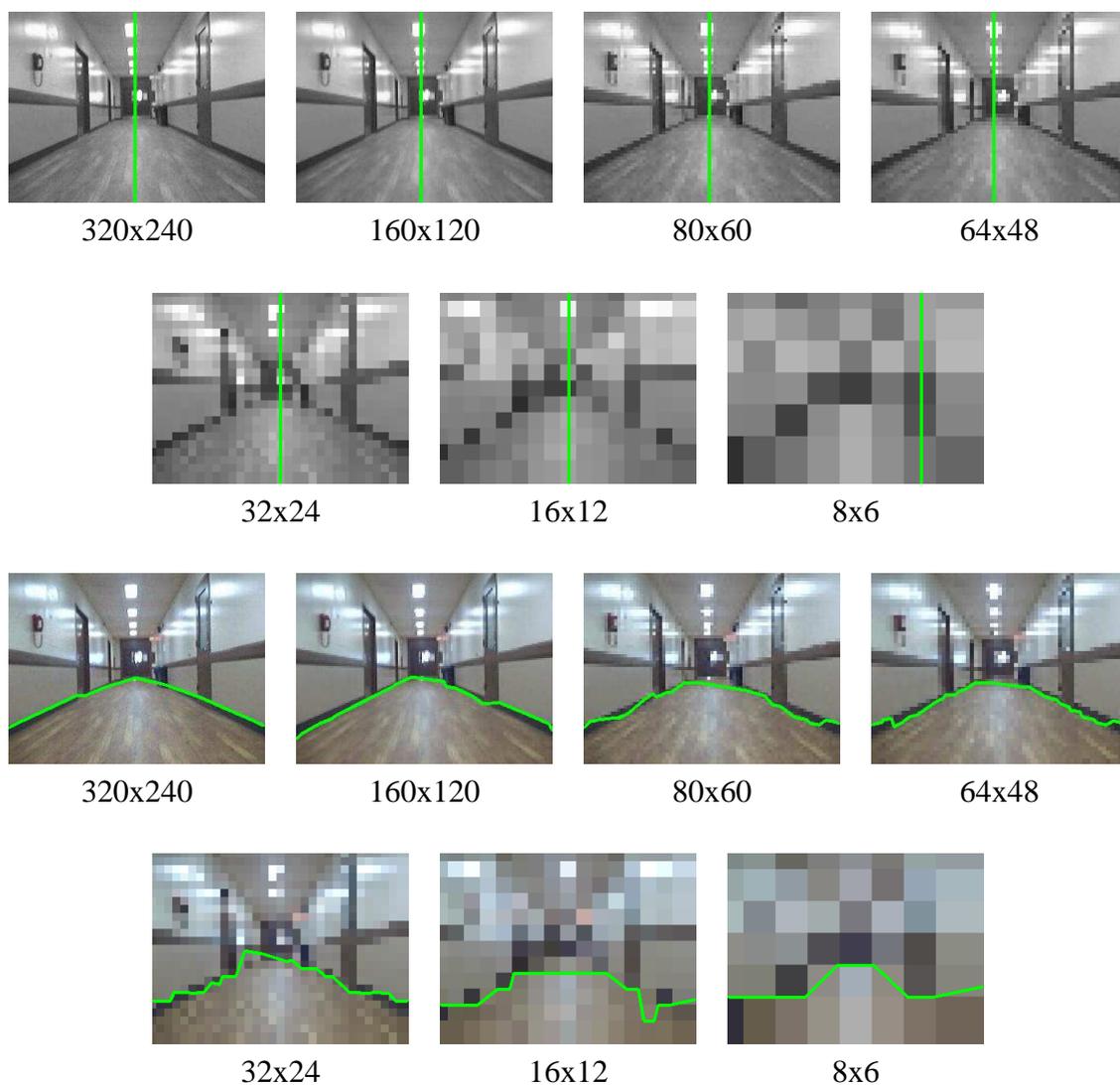


Figure 3.1: TOP TWO ROWS: The orientation line estimate (vertical green line) for the images shown in Figure 1.2. Down to a resolution of 16×12 , the results remain essentially unchanged. Only at the lowest resolution of 8×6 is the technique unable to recover the orientation line accurately. BOTTOM TWO ROWS: The wall-floor boundary found by the algorithm described in Chapter 2 for the different resolution images of Figure 1.2. The accuracy degrades slightly until the resolution of 32×24 , after which the errors become more pronounced.

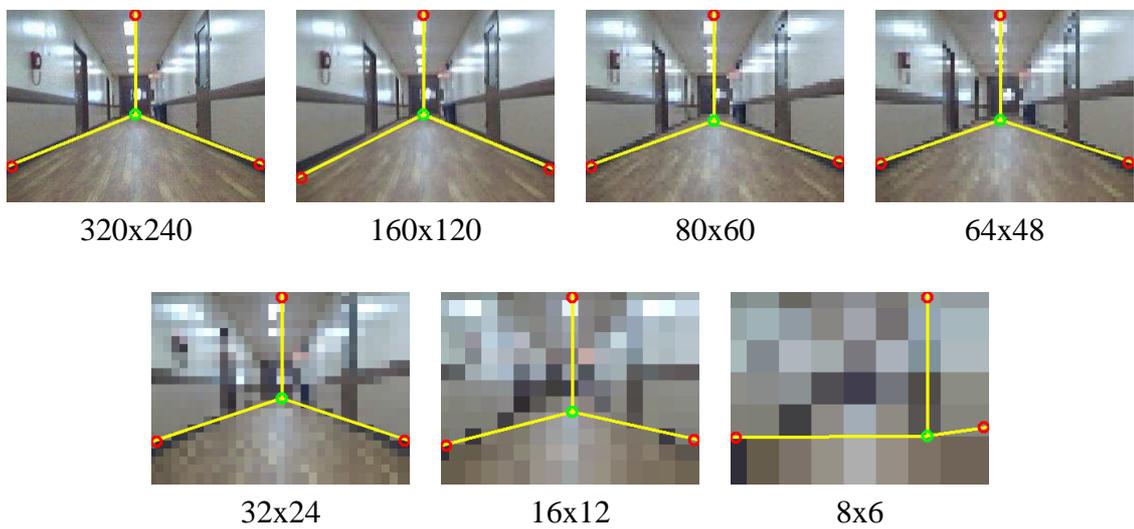


Figure 3.2: The three-line model estimate of the corridor found by combining the orientation line with the wall-floor boundary, on the same images. As before, the structure of the corridor remains intact even in the resolution of 32×24 , with only slight errors visible in 16×12 .

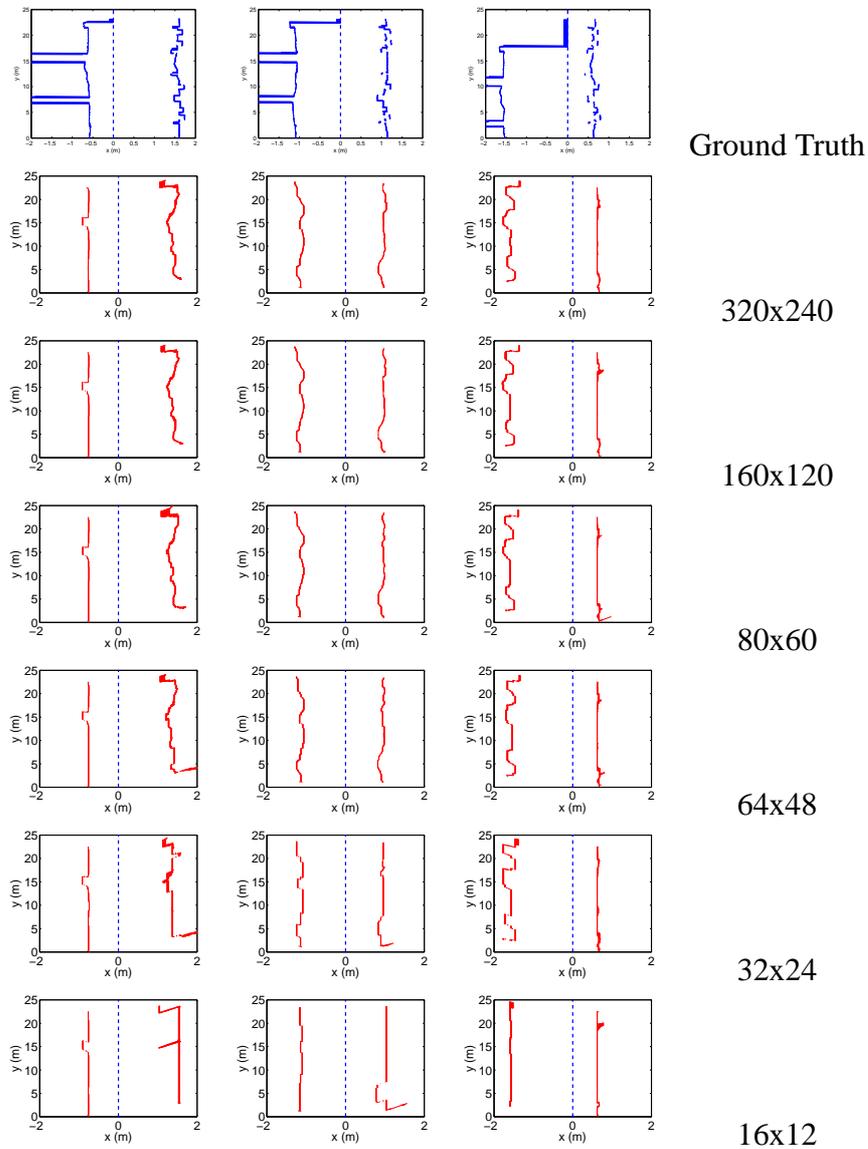


Figure 3.3: Corridor structure reconstruction from the wall-floor boundary, displayed as a top-down view. The first row in blue shows the ground truth location of the walls (Cartesian conversion of polar laser readings), and the next 6 rows in red show the reconstruction results from the wall-floor boundaries on different resolution images. Each row represents a different run of the robot in the same corridor, with the robot placed at a different lateral position in the corridor for each run (The position of the robot is shown by the dotted line at 0).

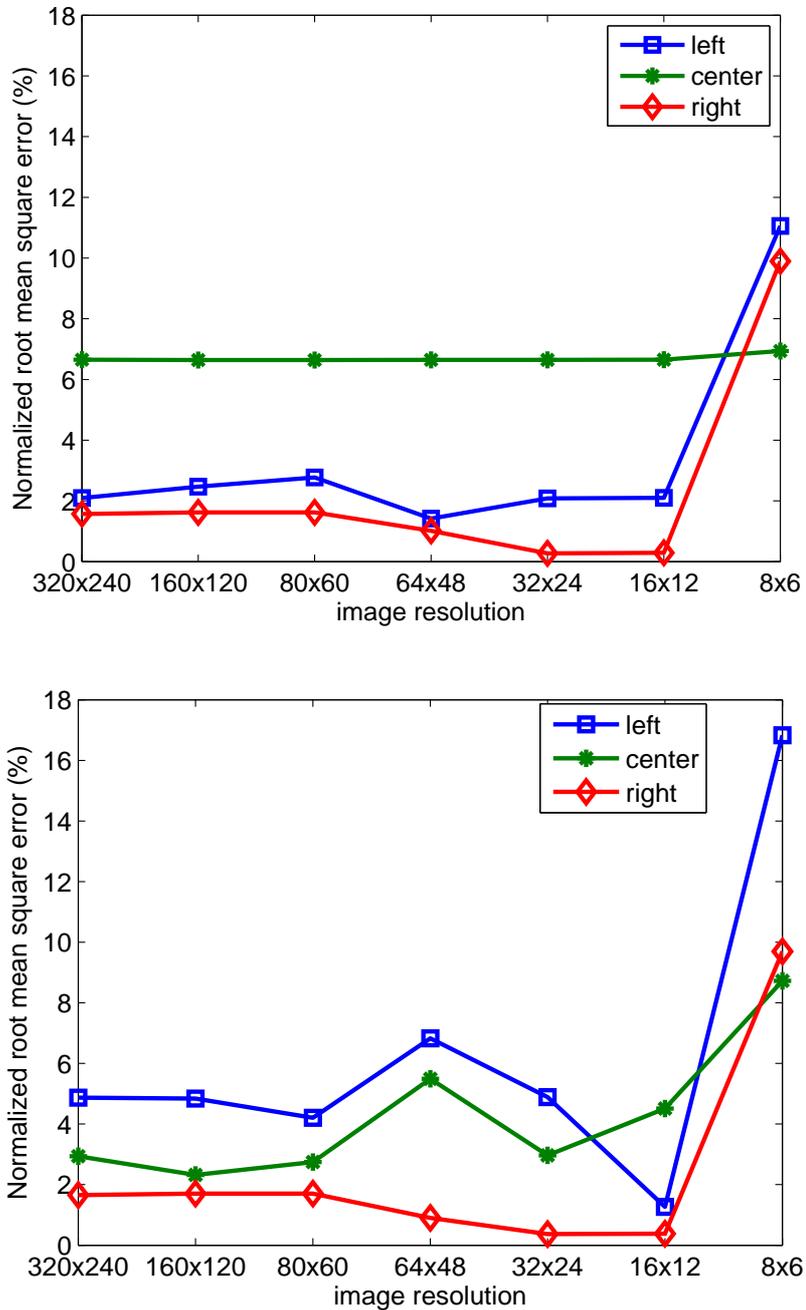


Figure 3.4: TOP: Normalized root mean square error (NRMSE) for estimating the lateral position of the robot for three runs in a single corridor. The structure was accurately captured in all three cases. BOTTOM: Mean NRMSE for the estimation of the corridor width. There is not much difference in estimation error rates across the different resolutions, and in fact the error drops in some cases for 32x24 and 16x12 sizes due to the removal of noise and artifacts by downsampling.

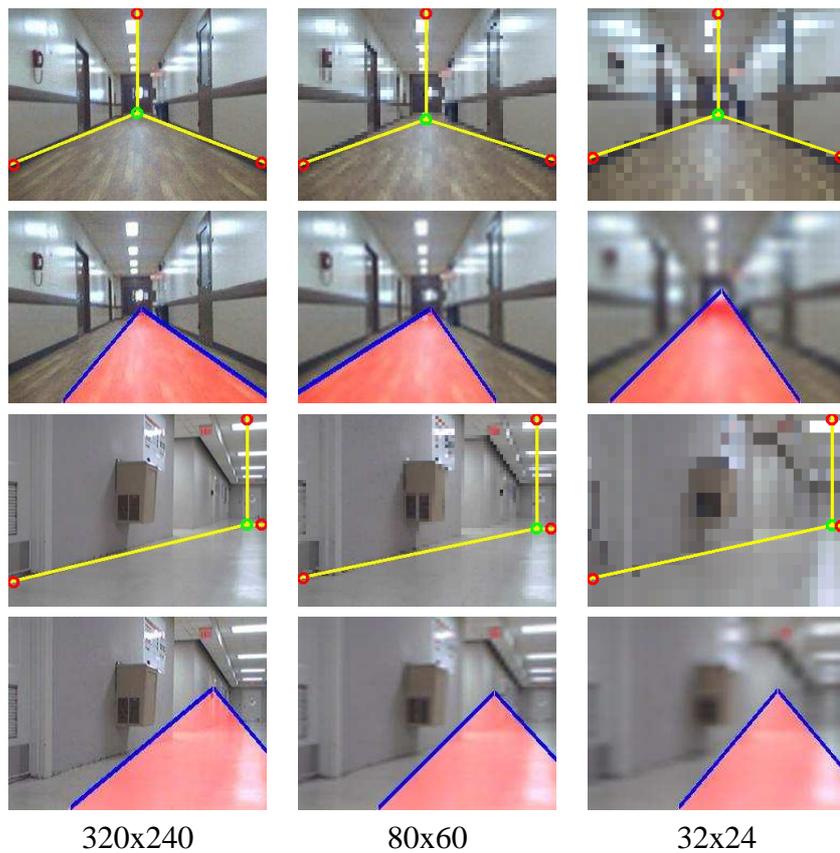


Figure 3.5: Comparison between our results (three yellow lines) in the first and the third rows and those of Kong et al. [15] (pink region) in the second and the fourth rows. Our algorithm achieves more accurate estimation of both the orientation line and the wall-floor boundary in indoor scenes, particularly at low resolutions.

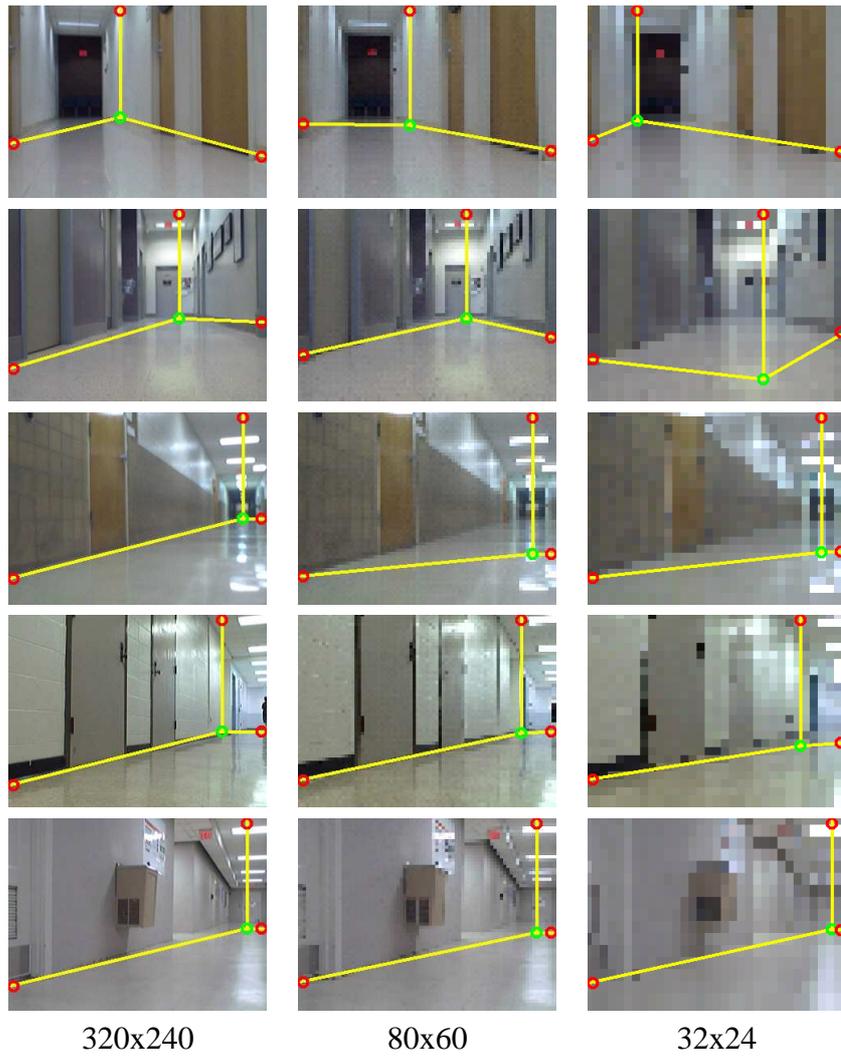


Figure 3.6: Additional results for other corridors, including one without ceiling lights in the first row. In some case, the low-resolution image yields more accurate results.

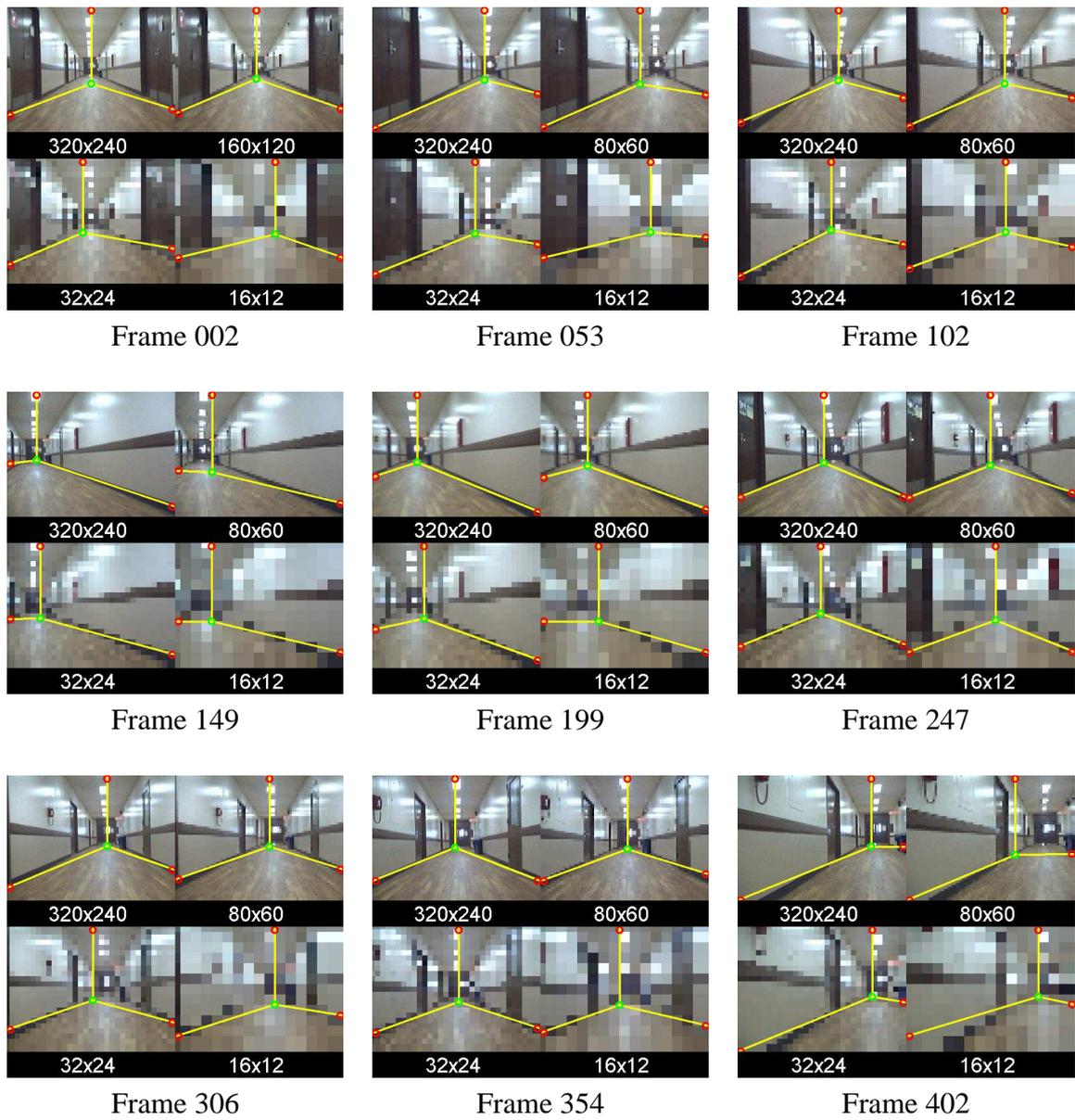


Figure 3.7: Minimalistic geometry estimation at four different resolutions. Sequence 1.

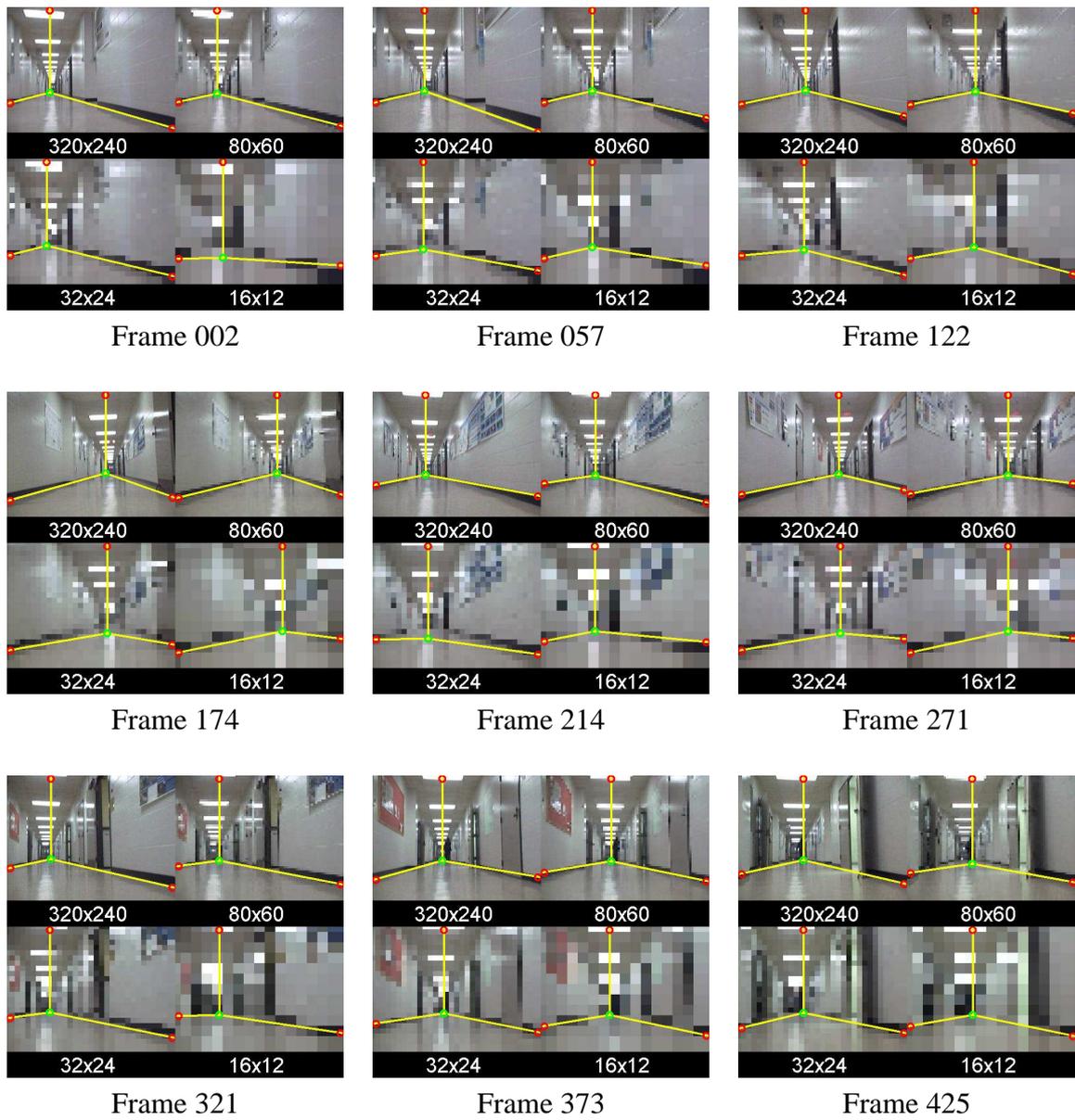


Figure 3.8: Minimalistic geometry estimation at four different resolutions. Sequence 2.

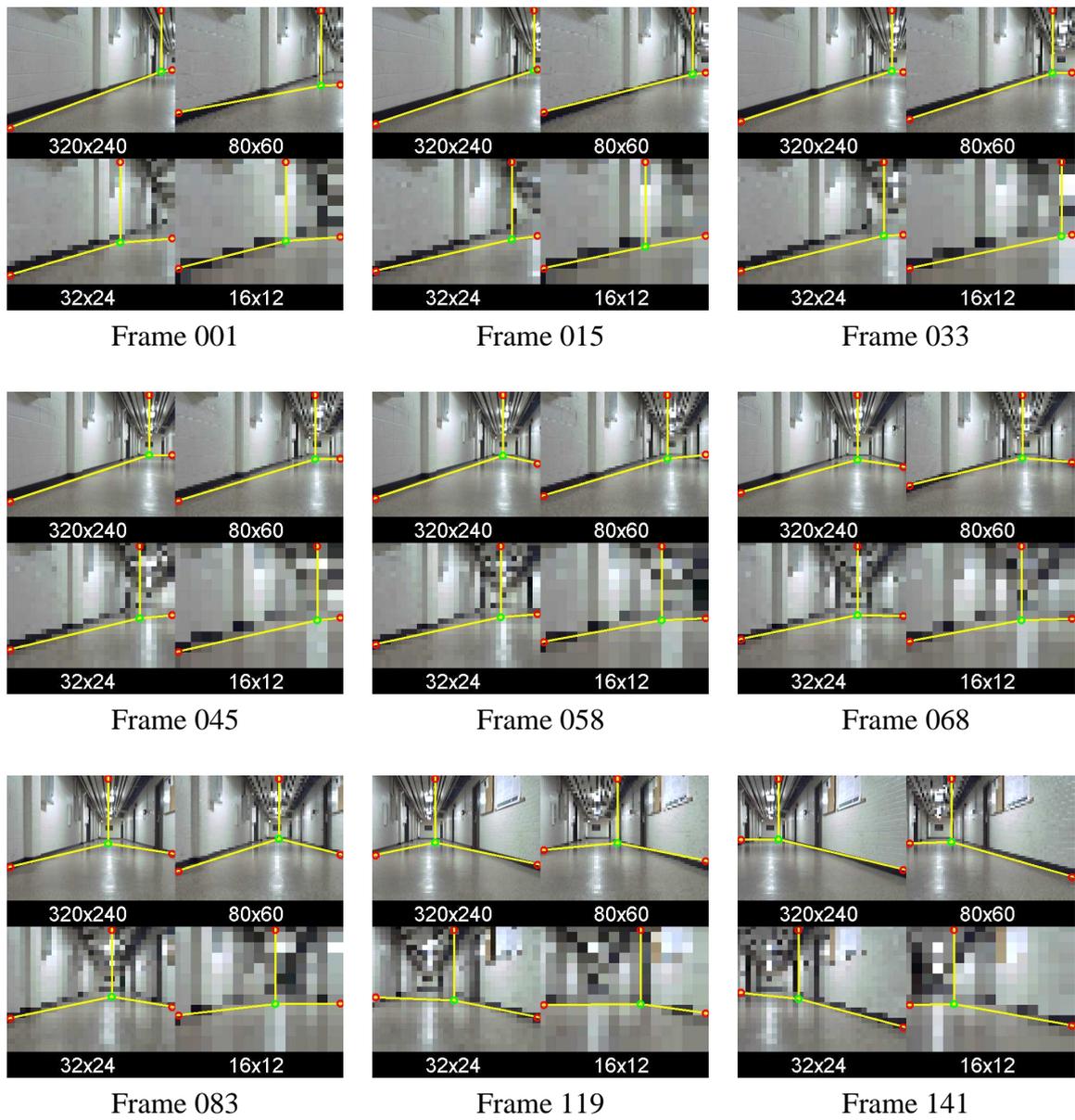


Figure 3.9: Minimalistic geometry estimation at four different resolutions. Sequence 3.

Chapter 4

Conclusions and Discussion

We have presented an image-based floor segmentation algorithm using an uncalibrated camera. The floor is detected by a camera mounted on a mobile robot, which maintains a low perspective of the scene. The novel approach combines the results of applying three different visual cues to test the validity of horizontal line segments detected in the image. Our approach achieves nearly 90% detection of the wall-floor boundary on a rather large database of over 400 images captured in a variety of environments exhibiting difficult conditions such as extreme reflection. The algorithm is suitable for nearly real-time mobile robot applications using an off-the-shelf camera. One limitation of the current approach is its tendency to get confused when the floor is highly textured, or when the image is especially dark due to poor gain control.

In addition, applying our floor segmentation algorithm to extract a minimalistic geometric representation of a typical indoor corridor environment using low resolution images. Motivated by the “selective degradation hypothesis”[17], our approach exploits the redundancy of image information in order to extract useful information for mobile robotic tasks with minimal processing. The proposed algorithm was tested on images from several different corridors, showing that the accuracy of the estimation of the orientation line or corridor geometry

changed very little even when more than 99% of the original information was discarded by downsampling the image to an extremely low resolution. The extension results can be seen as an exploration into identifying how much information is needed for basic mobile robot tasks such as corridor exploration and navigation. By reducing the resolution required for these basic tasks, the CPU time is freed for other tasks that potentially require higher resolutions and more involved processing.

Future work includes integrating more visual cues into the floor segmentation algorithm and applying Adaboost training to obtain a set of parameters for each visual cues.

Bibliography

- [1] E. Bayramoglu, N.A. Andersen, N. Kjolstad Poulsen, J.C. Andersen, and O. Ravn. Mobile robot navigation in a corridor using visual odometry. In *International Conference on Advanced Robotics*, pages 1–6, Jun 2009.
- [2] B. Bonev, M. Cazorla, and F. Escolano. Robot navigation behaviors based on omnidirectional vision and information theory. *Journal of Physical Agents*, 1(1):27–35, September 2007.
- [3] G. Borgefors. Distance transformation in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.
- [4] J. Canny. A computational approach for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [5] C. Chang and C. Lin. Libsvm – a library for support vector machines. <http://www.csie.ntu.edu.tw/~Ecjlin/libsvm/>, 2011.
- [6] Z. Chen and S. T. Birchfield. Visual detection of lintel-occluded doors from a single image. In *Workshop of International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [7] X.-N. Cui, Y.-G. Kim, and H. Kim. Floor segmentation by computing plane normals from image motion fields for visual navigation. *International Journal of Control, Automation, and Systems*, 7(5):788–798, 2009.
- [8] D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.
- [9] F. Escolano, B. Bonev, P. Suau, W. Aguilar, Y. Frauel, J.M. Saez, and M. Cazorla. Contextual visual localization: Cascaded submap classification, optimized saliency detection, and fast view matching. In *IEEE International Conference on Intelligent Robots and Systems*, 2007.
- [10] E. Fazl-Ersi and John K. Tsotsos. Region classification for robust floor detection in indoor environments. In *Proceedings of the 6th International Conference on Image Analysis and Recognition*, 2009.

- [11] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, September 2004.
- [12] A. Guzman. Decomposition of a visual scene into three dimensional bodies. In *Proceedings of Fall Joint Computer Conference*, 1968.
- [13] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *International Conference on Computer Vision (ICCV)*, 2005.
- [14] Y.-G. Kim and H. Kim. Layered ground floor detection for vision-based mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.
- [15] Hui Kong, Jean Yves Audibert, and Jean Ponce. General road detection from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [16] D. Lee, M. Hebert, and Y. Kanade. Geometric reasoning for single image structure recovery. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [17] H. W. Leibowitz, C. S. Rodemer, and J. Dichgans. The independence of dynamic spatial orientation from luminance and refractive error. *Perception and Psychophysics*, 25(2):75–79, 1979.
- [18] Y. Li and S. T. Birchfield. Image-based segmentation of indoor corridor floors for a mobile robot. In *Proceedings of the IEEE Conference on Intelligent Robots and Systems (IROS)*, pages 837–843, 2010.
- [19] Liana M. Lorigo, Rodney A. Brooks, and W. E. L. Grimson. Visually-guided obstacle avoidance in unstructured environments. In *IEEE Conference on Intelligent Robots and Systems*, volume 1, pages 373–379, September 1997.
- [20] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.
- [21] K. McHenry, J. Ponce, and D. Forsyth. Finding glass. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [22] G. F. McLean and D. Kotturi. Vanishing point detection by line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1090–1095, Nov 1995.
- [23] V. Murali, Y. Li, and S. Birchfield. Extracting minimalistic corridor geometry from low-resolution images. In *IEEE Conference on Intelligent Robots and Systems (IROS) (In review)*, 2011.
- [24] V. N. Murali and S. T. Birchfield. Autonomous exploration using rapid perception of low-resolution image information. *Autonomous Robots (In review)*.

- [25] Vidya N. Murali and Stanley T. Birchfield. Autonomous navigation and mapping using monocular low-resolution grayscale vision. In *Workshop on Visual Localization for Mobile Platforms (in association with CVPR)*, June 2008.
- [26] Jason M. O’Kane and Steven M. LaValle. Almost-sensorless localization. In *Proc. IEEE International Conference on Robotics and Automation*, 2005.
- [27] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.
- [28] Long Quan and Roger Mohr. Determining perspective structures using hierarchical Hough transform. *Pattern Recognition Letters*, 9(4):279 – 286, 1989.
- [29] T. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man and Cybernetics*, 8(8):630–632, 1978.
- [30] K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara. Obstacle avoidance and path planning for humanoid robots using stereo vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.
- [31] J. Santos-Victor and G. Sandin. Uncalibrated obstacle detection using normal flow. *Machine Vision and Applications*, 14(2):158–177, 1995.
- [32] S. Segvic and S. Ribaric. Determining the absolute orientation in a corridor using projective geometry and active vision. *IEEE Transactions on Industrial Electronics*, 48(3):696–710, Jun 2001.
- [33] N. O. Stoffler, T. Burkert, and G. Farber. Real-time obstacle avoidance using an MPEG-processor-based optic flow sensor. In *International Conference on Pattern Recognition*, 2000.
- [34] A. Torralba. How many pixels make an image? *Visual Neuroscience*, 26(1):123–131, 2009.
- [35] A.B. Torralba, R. Fergus, and W.T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, November 2008.
- [36] Benjamn Tovar, Luis Guilamo, and Steven M. Lavalle. Gap navigation trees: Minimal representation for visibility-based tasks. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, pages 11–26, 2004.
- [37] Hermann Weyl. *Symmetry*. Princeton Univ. Press, Princeton, NJ, 1952.
- [38] J. Zhou and B. Li. Robust ground plane detection with normalized homography in monocular sequences from a robot platform. In *Proceedings of the International Conference on Image Processing*, 2006.