AUTONOMOUS NAVIGATION AND MAPPING USING MONOCULAR LOW-RESOLUTION GRAYSCALE VISION

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
Vidya N. Murali
August 2008

Accepted by:
Dr. Stanley Birchfield, Committee Chair
Dr. Adam Hoover
Dr. Ian Walker

ABSTRACT

Vision has been a powerful tool for navigation of intelligent and man-made systems ever since the cybernetics revolution in the 1970s. There have been two basic approaches to the navigation of computer controlled systems: The self-contained bottom-up development of sensorimotor abilities, namely perception and mobility, and the top-down approach, namely artificial intelligence, reasoning and knowledge based methods. The three-fold goal of autonomous exploration, mapping and localization of a mobile robot however, needs to be developed within a single framework. An algorithm is proposed to answer the challenges of autonomous corridor navigation and mapping by a mobile robot equipped with a single forward-facing camera. Using a combination of corridor ceiling lights, visual homing, and entropy, the robot is able to perform straight line navigation down the center of an unknown corridor. Turning at the end of a corridor is accomplished using Jeffrey divergence and time-to-collision, while deflection from dead ends and blank walls uses a scalar entropy measure of the entire image. When combined, these metrics allow the robot to navigate in both textured and untextured environments. The robot can autonomously explore an unknown indoor environment, recovering from difficult situations like corners, blank walls, and initial heading toward a wall. While exploring, the algorithm constructs a Voronoi-based topo-geometric map with nodes representing distinctive places like doors, water fountains, and other corridors. Because the algorithm is based entirely upon low-resolution (32×24) grayscale images, processing occurs at over 1000 frames per second.

DEDICATION

This work is dedicated to graduate students and researchers all over the world.

ACKNOWLEDGMENTS

The first thanks must go to my adviser Dr Stan Birchfield, who has been a wall of support throughout. His patience and foresight have incessantly steered me in the right direction while saving me from numerous pitfalls. I must thank Dr Adam Hoover and Dr Ian Walker for teaching me all the important things early on, and for their continuous positive guidance during my research. I also thank them for being on the committee and for their help in directing the research to a credible conclusion.

My fellow lab mates in the Riggs basement, Zhichao Chen and Chris Dunkel have been great mentors, teaching me new things through meaningful discussions. They have always been there to answer my queries and for finding quick workarounds for glitches encountered while working with the robot.

Thanks to all the people who helped develop *Blepo* — the computer vision library in our department. It made all the code possible.

I extend my gratitude to all the great researchers and students of robotics and vision whose work has influenced and inspired me to start on this venture, without whom this work would not have been possible.

I owe a great deal to my parents for being supportive of my decisions and to my sisters Asha and Priya for being an eternal source of creative ideas and arguments.

TABLE OF CONTENTS

Table of Contents (Continued)

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

List of Figures (Continued)

Figure Page

List of Figures (Continued)

Figure Page

List of Figures (Continued)

Figure Page

Chapter 1

Introduction

Psychological studies have shown that human intelligence does not require high-resolution images to ascertain information about the environment for basic navigation. For example the "selective degradation hypothesis", developed by Leibowitz [33], says that some visual abilities such as vehicle steering and speed control remain relatively easy despite loss in visual acuity and color vision. For canonical tasks like walking or moving in a straight line, only a small percentage of what we see is actually useful, and in fact low-frequency information alone is sufficient for success. Motivated by this idea, we describe a system that uses only low-resolution (32×24) grayscale images to navigate a previously unknown corridor environment and to produce a Voronoi-based topo-geometric map of the environment [41]. By discarding 99% of the information captured from a 320×240 camera, the approach is computationally efficient, freeing up much of the CPU for other computation-intensive tasks, such as landmark recognition.

Our approach has a two-fold goal: autonomous navigation in an unknown environment and autonomous mapping. Straight line navigation is achieved using a combination of simple visual competencies such as ceiling lights and homing. Detecting the end of the corridor and turning is achieved by using direct and relative measures of visual information content.

Similarly, a probabilistic combination of information measures is used to detect and map interesting and salient landmarks on either side of the corridor. The navigation path and the detected landmarks are combined to plot a simple roadmap type of topo-geometric map of the environment.

1.1 Mobile robot navigation

Mobile robot navigation has a three-fold fundamental goal: autonomous exploration of an unknown environment, mapping of the environment, and localization in the environment. Typically using on-board computation and standard off the shelf hardware, mobile robots using multiple sensors have been developed for land, sea and aerial navigation and are deployed in the manufacturing, military, security, consumer and entertainment industries.

1.2 Monocular vision as the sensor

As noted by Horswill [25], there is a correlation between the surface structure of an image and the deep structure of objects in the real world. Vision is more powerful than other sensors because vision provides different kinds of information about the environment, while other sensors (such as sonars or lasers) only give us depth. For landmark detection and recognition, vision provides direct ways to do so and is easy to represent because of the close relation to the way humans understand landmarks. In addition lasers are expensive and power-hungry, and sonars cause interference. Vision based navigation can now be achieved using a single off-the-shelf camera which is inexpensive and scalable.

Navigating with a single camera is not easy. Perhaps this is why many approaches rely upon depth measurements from sonars, lasers, or stereo cameras to solve the problem. Granted, knowledge of distances to either wall, the shape of obstacles, and so on, would be directly

useful for localizing the robot and building a geometric map of the environment. Stereo vision has its own difficulties (e.g., it requires texture to compute correspondence, is computationally expensive and produces inaccurate results for many pixels). Indoor environments in particular often lack texture, rendering stereo matching an elusive problem in such places. In contrast, humans are quite good at navigating indoors with one eye closed, even with blurry vision, thus motivating us to find a different solution.

1.3 Low-resolution vision

Low-resolution vision is not entirely new in vision based navigation. As a matter of fact, the older navigation systems predominantly used low-resolution vision because image processing was limited by the technology of the time. For example Horswill [25] used 64×48 and 16×12 images in his work. The selective degradation hypothesis [33] serves to strengthen the axiom that navigation is possible under low visual acuity. The hypothesis itself is a result of several studies and experiments in psychology [2]. Research showed that two modes of vision are present in living creatures: *recognition* and *guidance*, which both contribute to visual perception. But under poor visual conditions such as low-resolution or low illumination, the *recognition* mode deteriorates sharply, while the *guidance* or visually induced self-motion still maintains high efficiency. Schneider [2] showed that animals with poor vision were able to orient visually toward salient visual events. Therefore combining salient events with low-resolution provides a natural structure for perceptive, explorative navigation and mapping. An additional advantage is the gain of CPU time involved in processing, loading and retrieving images in a potential real-time embedded environment.

It should be noted that recent independent work by Torralba *et al.* [56] has also emphasized the importance of low-resolution vision. 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. The application to person detection and localization is particularly noteworthy considering that they have produced good results with low-resolution images that are comparable to that of the Viola-Jones detector which uses high-resolution images. The whole spirit of low-resolution is strengthened by this argument and one can see that low-resolution images (up to the limits of scotopic vision) provide enough information for basic visual processing.

1.4 Structure of this report

The next chapter gives a summary of the work done previously in vision based navigation, the different approaches, the achievements, limitations, and ongoing work. Chapter 3 describes the detailed structure of the algorithm, the percepts, the visual processing for navigation and for mapping. Chapter 5 discusses a number of experiments conducted on the robot in an indoor environment, with supporting plots, graphs and image sequences. These give us a sense of the reliability of the algorithm and its limitations. Chapter 6 gives a summary of the thesis, an outline of future work and concluding discussions.

Chapter 2

Previous work

Vision-based mobile robot navigation has been studied by many researchers. From the early work of the Stanford Cart [40] to the current Aibo (the toy robot built by Sony), navigation has been recognized as a fundamental capability that needs to be developed. According to the survey of DeSouza *et al.* [20], significant achievements have been made in indoor navigation, with FINALE [29] being one of the more successful systems. FINALE requires a model-based geometric representation of the environment and uses ultrasonic sensors for obstacle avoidance. NEURO-NAV [37] is another oft cited system that uses a topological representation of the environment and responds to human-like commands. RHINO [9] is an example of a robust indoor navigating robot. The highly notable NAVLAB [55] is an example of proficient outdoor navigation system which use a combination of vision and a variety of other sensors for navigation and obstacle avoidance. Moravec [40] and Nelson *et al.* [43], however, have emphasized the importance of low-level vision in mobile robot navigation, and Horswill [25] implemented a hierarchical and complete end-to-end vision-based navigational robot based on prior training of the environment.

One approach to navigation has been to use corridor lights, which can achieve robust navigation even in long corridors. In some systems, lights are used as landmarks in a teach/replay

approach, with the camera pointing toward the ceiling [32]. The drawback of such a configuration, of course, is that the robot is blind to anything in front of it, not to mention that the system must be trained beforehand on the environment to be navigated. In another implementation, ceiling lights are used as aids in straight line navigation [31], but here again the camera points toward the ceiling, and the position and orientation of the rectangular lights are used for straight line navigation. Such a computation does not generalize well to environments in which the lights are not of a rectangular shape, or to robots with forward-facing cameras. Choi *et al.* [11] use a forward facing camera for detecting lights, but their approach also relies upon the actual geometrical features of the lamps, and it is restrained by the lights disappearing from the field of view, which is one of the main difficulties of forward-facing cameras. Our approach incorporates computations to handle this difficulty and to automatically detect and handle the end of a corridor, without any prior training of the environment or restriction on light shape.

2.1 Vision based navigation — An overview

Vision is powerful because it is inexpensive, non-intrusive and scalable. The various ways in which vision is used for navigation have been described in detail, by Desouza *et al.* [20]. Vision-based navigation systems can be classified as shown in Figure 2.1 which is a summary of [20]. This thesis aims to form a bridge between map-building systems and mapless systems, thus combining the goal of autonomous exploration and mapping.

2.1.1 Mapless navigation

Mapless navigation using vision predominantly uses primitive visual competencies like measurements of 2D motion (such a optical flow), structure from motion, independent motion detection, estimating time-to-contact and object tracking. While some/all of these can and

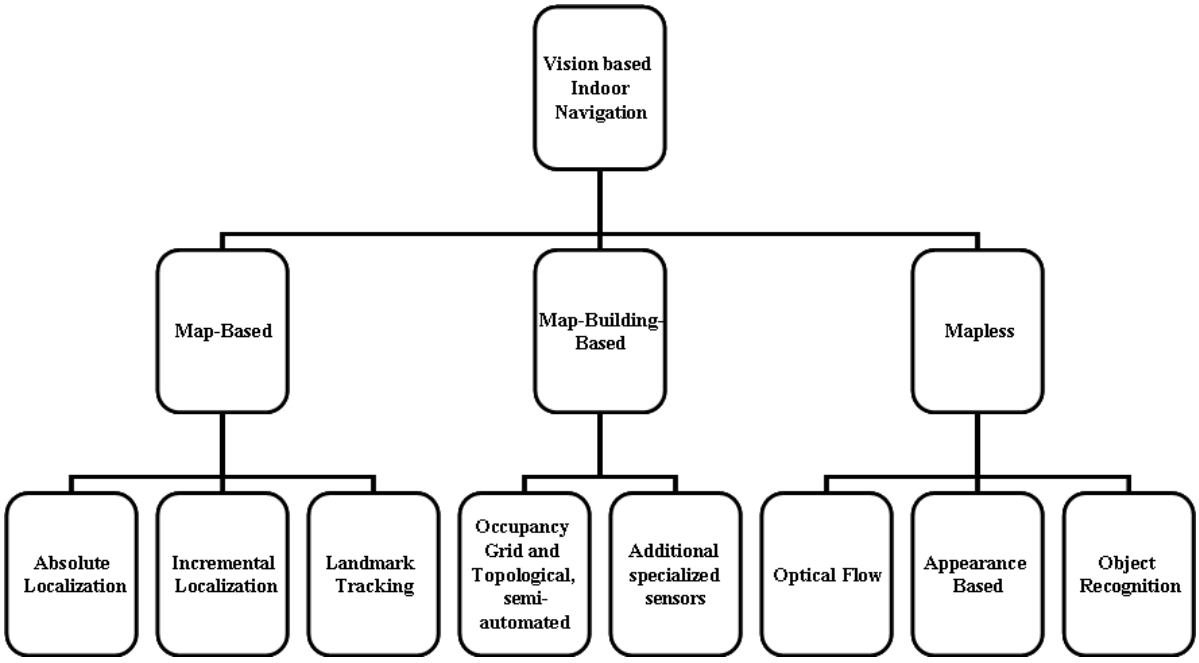

Figure 2.1: A taxonomy of approaches to vision-based navigation, summarizing [20].

have been used to develop a wandering robot, many open points of research need to be mentioned. None of these have been tested before on low-resolution systems (as low as 32×24). All of these visual competencies are known to face problems in textureless environments. These competencies can be used for continuous navigation in a predictable path. At a point of discontinuity such as a corridor end, however, these competencies themselves do not provide a solution.

2.1.2 Map-based navigation

Map-based navigation systems are a complete solution to the goal-based navigation problem. The definition of landmarks is a vital necessity of such a system. Again, historically there have been two types of visual landmarks: Sparse feature based landmarks and higher level abstract landmarks.

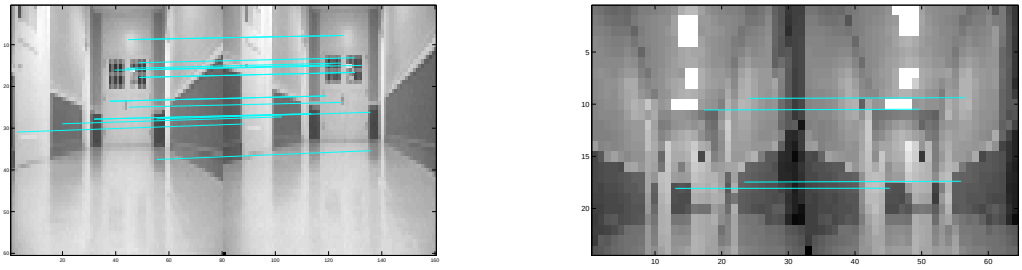

Figure 2.2: LEFT: Two images shown side-by-side with the SIFT matches between them shown as cyan lines. In an indoor environment, only a few SIFT features are found with a 160×120 image. RIGHT: Even fewer SIFT features are found for a low-resolution 32×24 image.

- *Sparse feature based landmarks*: Some of the prominent landmarks used today to represent visual landmarks in SLAM based systems, are based on edges, rotation invariant features, or corners. These are in fact represented by the three popular visual landmark representation techniques: SIFT (Scale Invariant Feature Transform) [50], Harris Corners [22] and Shi-Tomasi feature points [53]. These have the advantage of being robust, scale invariant and sparse [28]. But again the important points to be noted are as follows. These representations are computationally quite expensive. Some work has been done to develop real-time feature detectors, like real-time SLAM [19], GLOH [38] and SURF [4]. FAST [47, 48] is promising for high-speed, feature-based representations, but such approaches often leave little CPU time for other tasks and may not work well with textureless environments. These features work well in well-textured environments with high-resolution. In poorly textured environments with low resolution, sparse features are not robust enough. SIFT in particular is fairly sensitive to resolution and texture, see Figure 2.2.

- *High level abstract landmarks*: Another way of representing landmarks is to use all of the pixels together like the entire image itself, or reduced pixel information. Template

matching is a very simple, common yet powerful landmarks representation/matching technique. Histograms, color maps and other measures are also popular.

2.1.3 SLAM: Simultaneous Localization and Mapping

With regard to mapping, the recent developments in Simultaneous Localization and Mapping (SLAM) have been based primarily upon the use of range sensors [39, 45, 6]. A few researchers have applied this work to the problem of building maps using monocular cameras, such as in the vSLAM approach [27], which is a software platform for visual mapping and localization using sparse visual features. An alternate approach is that of Davison *et al.* [18, 17], who also use sparse image features to build 3D geometric maps.

In these visual SLAM techniques, either a complex matching process for a simple landmark representation [46] or a simple matching process for a complex landmark representation [50] is needed for robust robot localization. In indoor corridor environments, however, the lack of texture poses a major obstacle to such an approach. Indeed, popular techniques such as the Scale Invariant Feature Transform (SIFT) [50] or other feature representations have difficulty in such cases. Moreover, the computationally demanding nature of these algorithms often leaves little room for additional processing, and their design requires higher resolution images.

2.1.4 Map-building based navigation

The whole task of map-building described in modern SLAM, visual or not, always has a manual/tele-operated phase [16, 49, 18]. It is important to note that in most map-building systems, the robot is controlled manually. Autonomous navigation is rare, and autonomous vision-based mapping is even more rare [20]. Notable initiatives include the work done by Matsumoto *et al.* [34], who used omnidirectional cameras with stereo and optical flow to control navigation, and Shah *et al.* [51], who implemented an autonomous navigation system

using a calibrated fish eye stereo lens system. However, these approaches require specialized cameras. Similarly, autonomous vision-based navigation is rare, with many techniques requiring a training phase in which the robot is controlled manually [5, 10, 35, 36, 26]. As a result, efficient autonomous map building of indoor environments using a single off-the-shelf camera has remained an elusive problem.

Team ARobAS of INRIA have made a compelling statement in their annual report[2][54] about the incompleteness of SLAM. They state that the problem of explorative motion strategy of the robot (or reactive navigation) has rarely been a part of SLAM. They argue that autonomous navigation and SLAM cannot be treated separately and that a unified framework is needed for perception, modeling and control. Very few notable initiatives have completely automated the system for collecting the data required to build a map while navigating. Robust perception is a basic necessity for a navigating robot that can be deployed in an environment without human intervention.

[2]http://www.inria.fr/rapportsactivite/RA2007/arobas/arobas.pdf

Chapter 3

Approach: Low-resolution competencies for navigation and mapping

3.1 Algorithm overview

Navigation: The basic navigation system is represented in Figure 3.1. The robot basically has three modes of operation: the centering mode, the homing mode, and the turning at the end of the corridor. The robot navigates in a straight line using ceiling lights (when they are visible) and homing (when the ceiling lights disappear). Control decisions at the end of the corridor are made by entropy, Jeffrey divergence, and time-to-collision. Turning at the end of the corridor is controlled by a search for lights and high entropy.

Mapping: Landmarks are detected using two information measures (Jeffrey divergence and entropy that are measured for the entire navigation path), and marked with respect to the path obtained from navigation (see Figure 3.2). Corrected odometry is used to find the path as explained in section 3.8. The landmark detection is a passive process and does not have any feedback onto the navigation system. Localization is currently not being performed using the

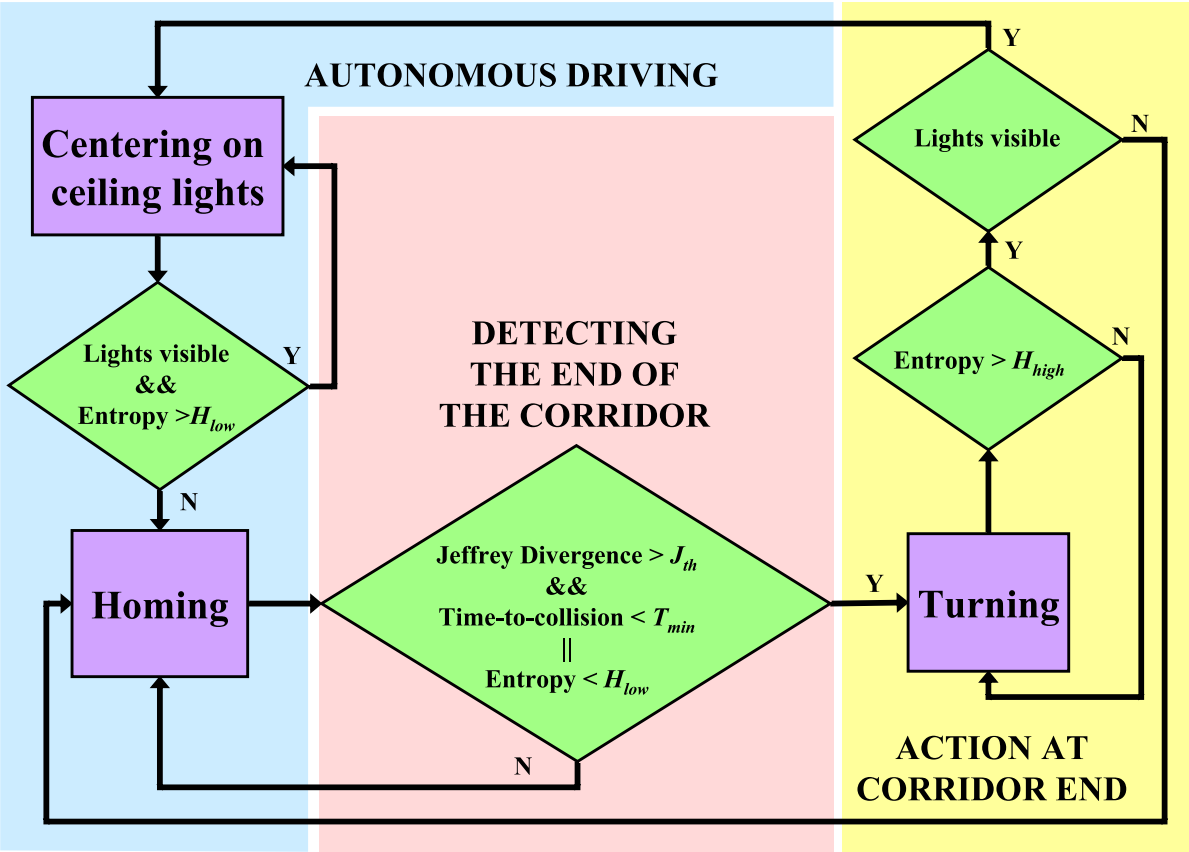

Figure 3.1: Overview of the navigation system. There are three challenges: Autonomous driving, detecting the end, and action at the end of the corridor. These are achieved by the three modes of operation: 'Centering the ceiling lights', 'homing', and 'turning' at the end of the corridor.

detected landmarks. Our ultimate goal is to build a system that uses landmarks for navigation and localization.

The end-end system is a proof of concept of the above described methods and the working of the complete system is demonstrated in the sections that follow where we describe each of these modules in greater detail.

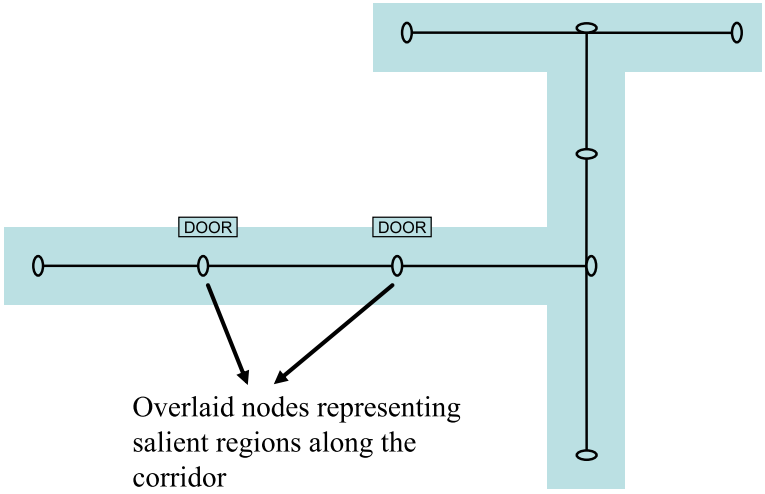

Figure 3.2: A flow diagram explaining the map-building. Salient regions of interest on either side of the corridor are detected using information measures and marked as nodes on the navigation links to form a topo-geometric map.

3.2 Centering using corridor ceiling lights

The image is divided into four adjoining triangles defined by the two diagonals of the image. Assuming that ceiling lights reside in the top triangle, we use the mean horizontal location of the intensities above a threshold to determine whether the robot is traveling in the center of the corridor. Letting the mean horizontal location in the image of the light source be l_{mean}, then the robot is centered by setting the rotational velocity to be a factor of $(l_{mean} - w/2)$, where w is the width of the image. By servoing on this location, the ability to navigate a long corridor with stability is achieved, even without any additional information from odometry or other

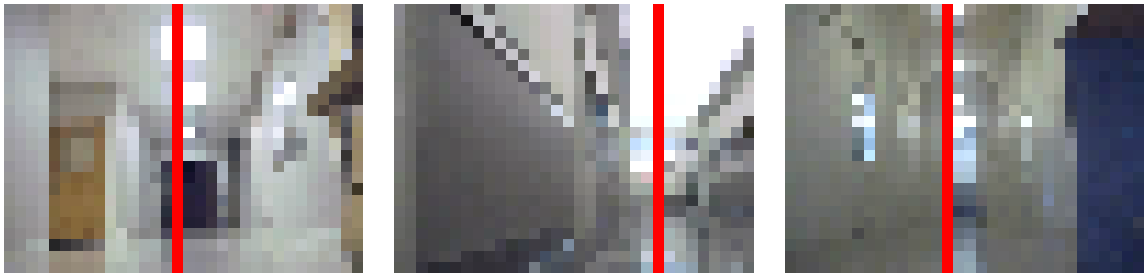

Figure 3.3: Different ceiling lights (in Riggs Hall) and their mean locations (vertical lines) detected by our algorithm. Notice that there is no restriction on the shape or location of lights. LEFT: Fluorescent center lamps in the basement. CENTER: Fluorescent lights on either side pointing toward the reflective ceiling in one wing of floor 1. RIGHT: Sodium vapor center lamps in another wing of floor 1.

sensors. This approach is not only simpler, but also more powerful and more general, than previous approaches that analyze the shape of lights. For example, Figure 3.3 shows a variety of lights that are successfully detected using this method. Note that ceiling lights provide an added advantage over vanishing points because they are affected by translation, thus enabling the robot to remain in the center of the corridor while also aligning its orientation with the walls.

3.3 Distinguishing the corridor by scalar entropy

The entropy of an image is a scalar representing the statistical measure of randomness that can be used to characterize its texture:

$$H(K) = \sum_{p \in K} -p \log p, \tag{3.1}$$

where p is the count value for each bin in the histogram K of the image I (256 bins for a 8-bit graylevel image). The normalized histogram is an approximation of the probability density function of a random variable whose realization is the particular set of pixel values found in the image [23]. The entropy is therefore a measure of the information content in an image.

According to Shannon's theory of entropy [52], the entropy is the measure of information content of a symbol and the rarer its occurrence the greater its information content. When the robot approaches a planar surface, like a blank wall or the surface of an untextured or structured object, the entropy drops as the occurrence of any gray value is very high; this implies that the camera is facing a planar obstacle immediately in front of it. In richly textured images, time-to-collision (TTC) [1] or central image divergence threshold [13] can be used to determine the position of a frontal obstacle. But in an environment devoid of texture and consisting of uniform color, these methods will fail. Using entropy (in addition to the existing methods), therefore, is a promising way to react to a situation where the image does not provide enough information for navigation. Other researchers have used entropy for determining the direction of navigation and for global visual localization using omnidirectional images [8, 21].

Entropy is used in several ways. While driving if the entropy drops sharply, the robot stops and turns away. In the same manner, while turning at the end of a corridor, the robot continues turning as long as either side has low entropy, and the overall entropy is below a threshold H_{low}. That a low entropy value indicates a nearby wall is illustrated in Figure 3.4, where sharp drops in entropy correspond to images where the robot is facing a blank wall.

Entropy can also be used to find corridors. Figure 3.5 shows a plot of entropy values as the robot turns on the spot facing three branches of a T-junction. The entropy is high when the robot is aligned with the corridor, and it drops sharply when the robot faces the wall. The plot of entropy values for other environments is shown along with laser measurements at T-junctions in Figures 3.6, 3.7, and 3.8 corresponding to three different buildings on our campus (Riggs, EIB and Lowry), indicating that entropy is a powerful measure for detecting open corridors. Therefore, entropy can be used to detect the presence of an open corridor for navigation when other metrics fail, whether in textured or untextured environments.

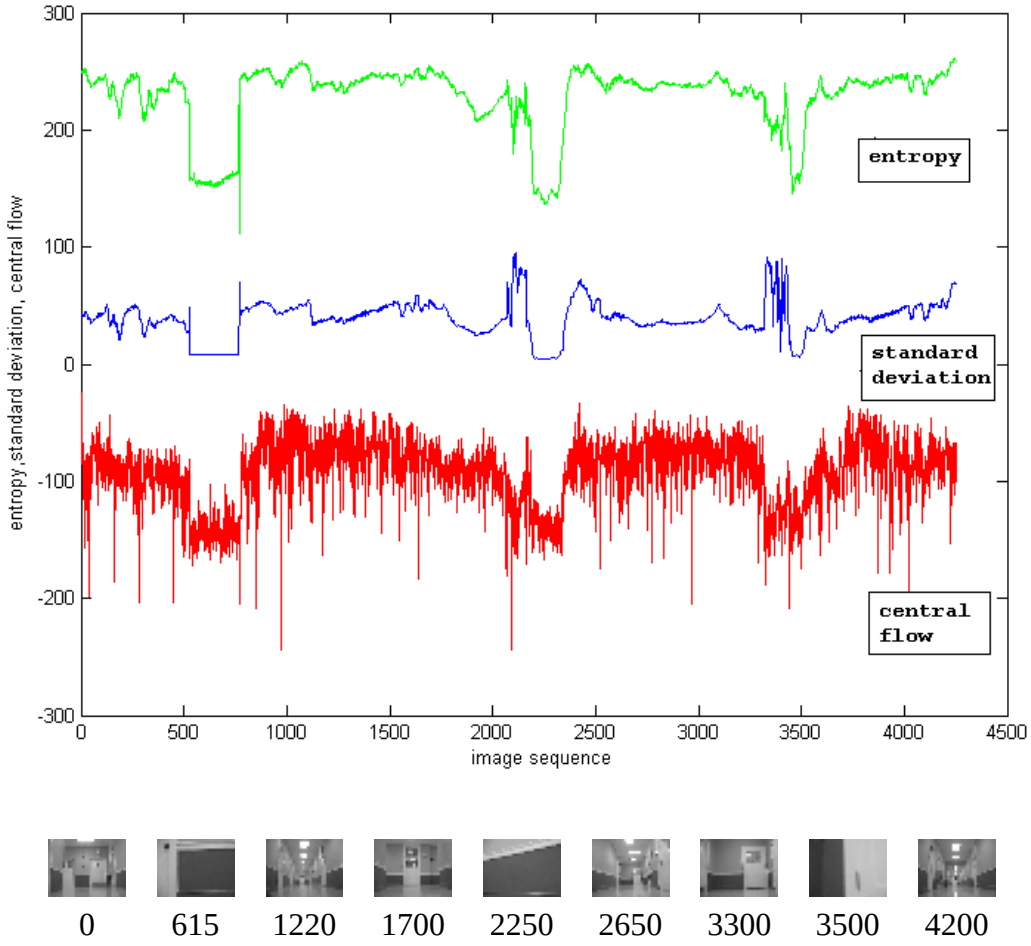

Figure 3.4: TOP: Comparison of image entropy, absolute image standard deviation, and central optical flow of the image, all measured while the robot traveled in a building. The three drops in entropy correspond to three turns, when the robot faced the walls. Notice that the entropy values are more easily distinguished (and less noisy) than those of the other measures. BOTTOM: The images corresponding to the above values are shown.

3.4 Homing mode

When the robot nears the end of a corridor, the lights disappear from the camera's field of view and the overall entropy drops. When either of these occurs, the robot automatically captures the current image and stores it as the 'home' image. Keeping that image in view, the robot navigates toward it using *homing* [42]: The current image is compared with the home image

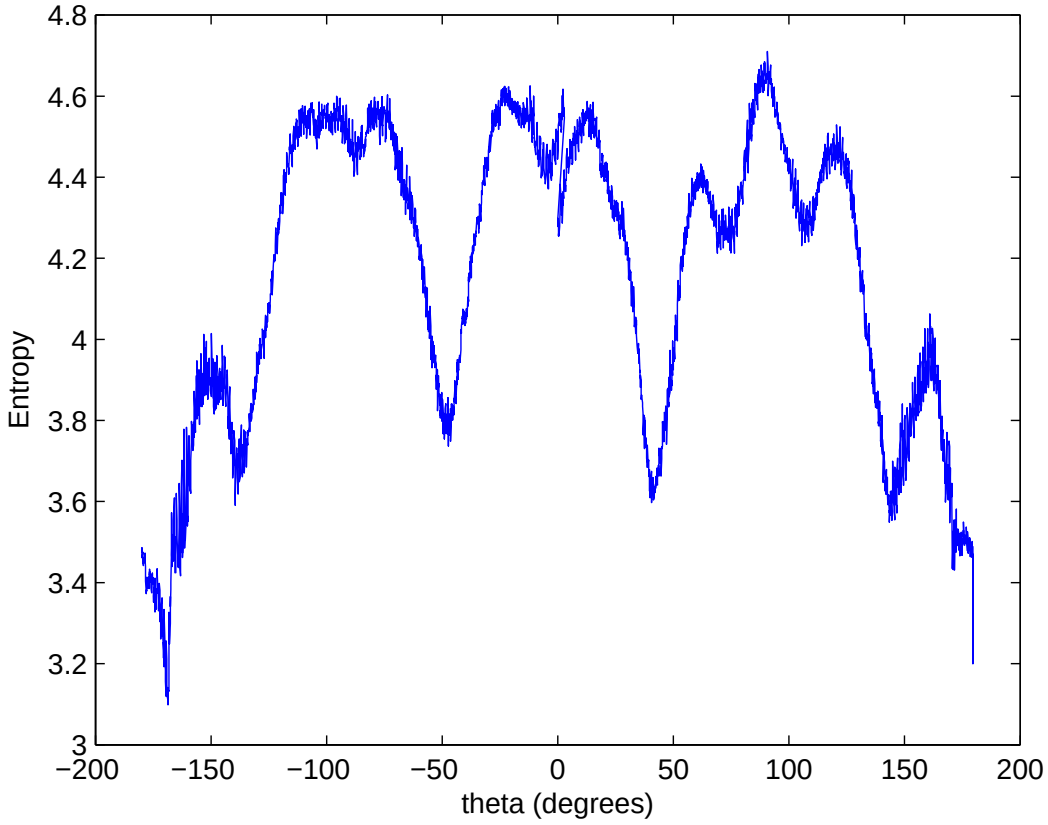

Figure 3.5: Entropy captured as the robot turned in place at the T-junction of two corridors in Riggs floor 3. Entropy is high when the robot faces the length of a corridor and drops sharply on either side, so the three peaks indicate the three corridor directions (at 0, 90, and -90 degrees). Maintaining high entropy allows the robot to avoid the specular reflections of the walls.

after shifting left and right by a maximum disparity of one pixel. The result that yields the lowest sum of absolute difference (SAD) indicates the robot's direction of motion. This keeps the robot in the center of the corridor even when the lights are not visible.

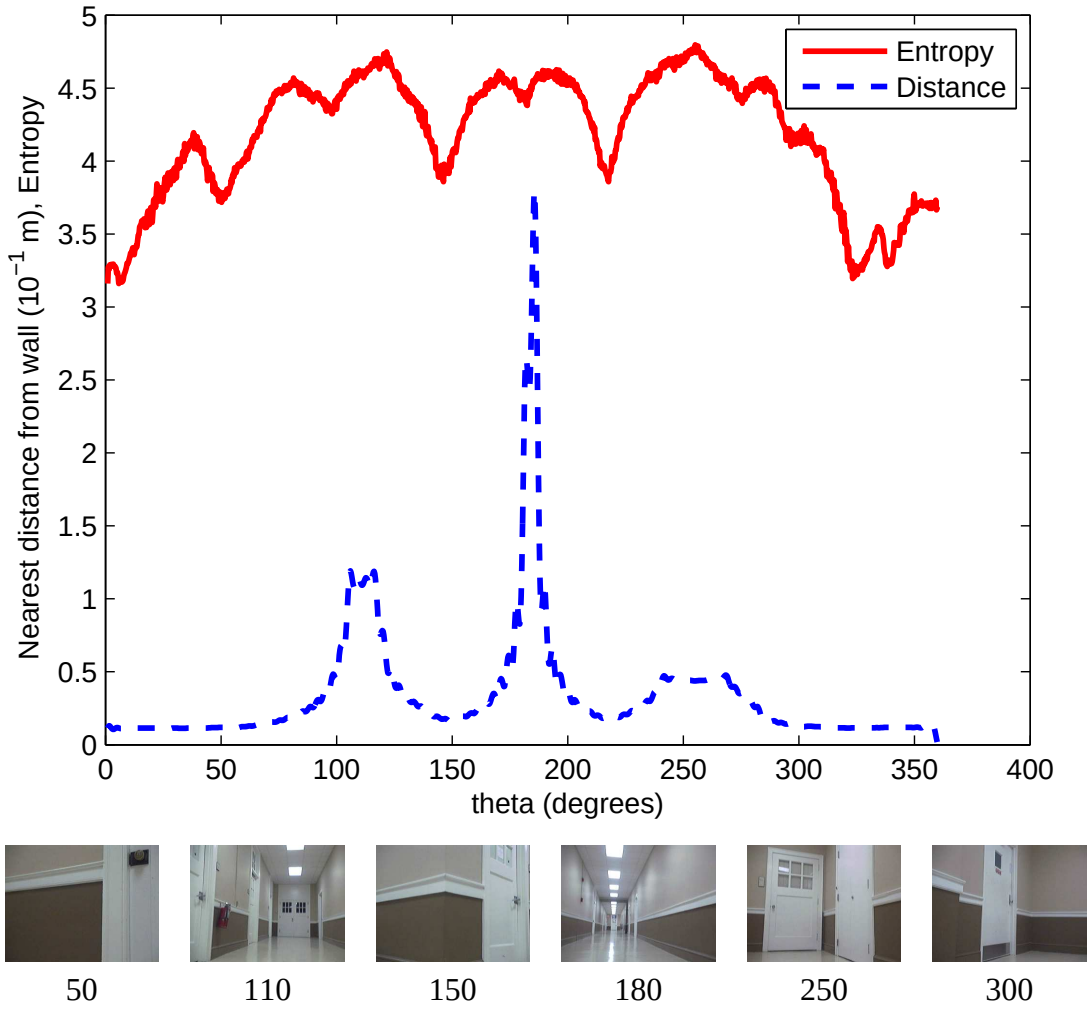

Figure 3.6: TOP: Entropy (red solid line) and distance (blue dashed line) as the robot turned at a corridor T-junction in Riggs floor 3. Distance was measured using a SICK laser scanner. BOTTOM: Images of the corridor approximately showing the orientation with respect to the depth values corresponding to them above.

3.5 Detecting the end of the corridor

The end of the corridor is determined by combining three measures: entropy (described in the previous chapter), relative entropy, and the time-to-collision, in order to navigate in different indoor environments with different levels of texture/information and lighting.

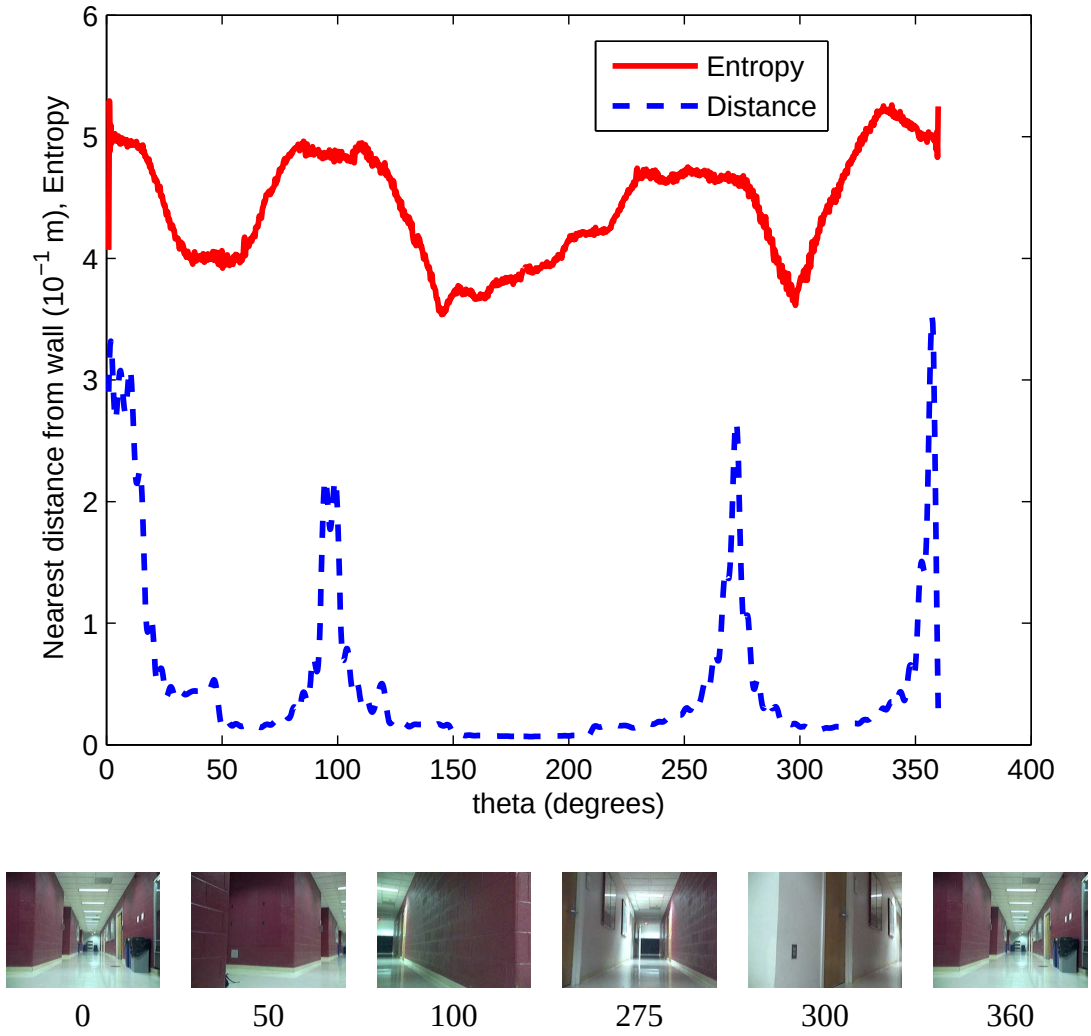

Figure 3.7: TOP: Entropy (red solid line) and distance (blue dashed line) as the robot turned at a corridor T-junction in EIB floor 1. Distance was measured using a SICK laser scanner. BOTTOM: Images of the corridor approximately showing the orientation with respect to the depth values corresponding to them above.

3.5.1 Relative entropy

Considering two discrete distributions with probability functions p_k and q_k, then the Kullback-Leibler distance of p with respect to q is given by

$$D\left(p,q\right)=\sum_{k}p_k\log\left(\frac{p_k}{q_k}\right), \tag{3.2}$$

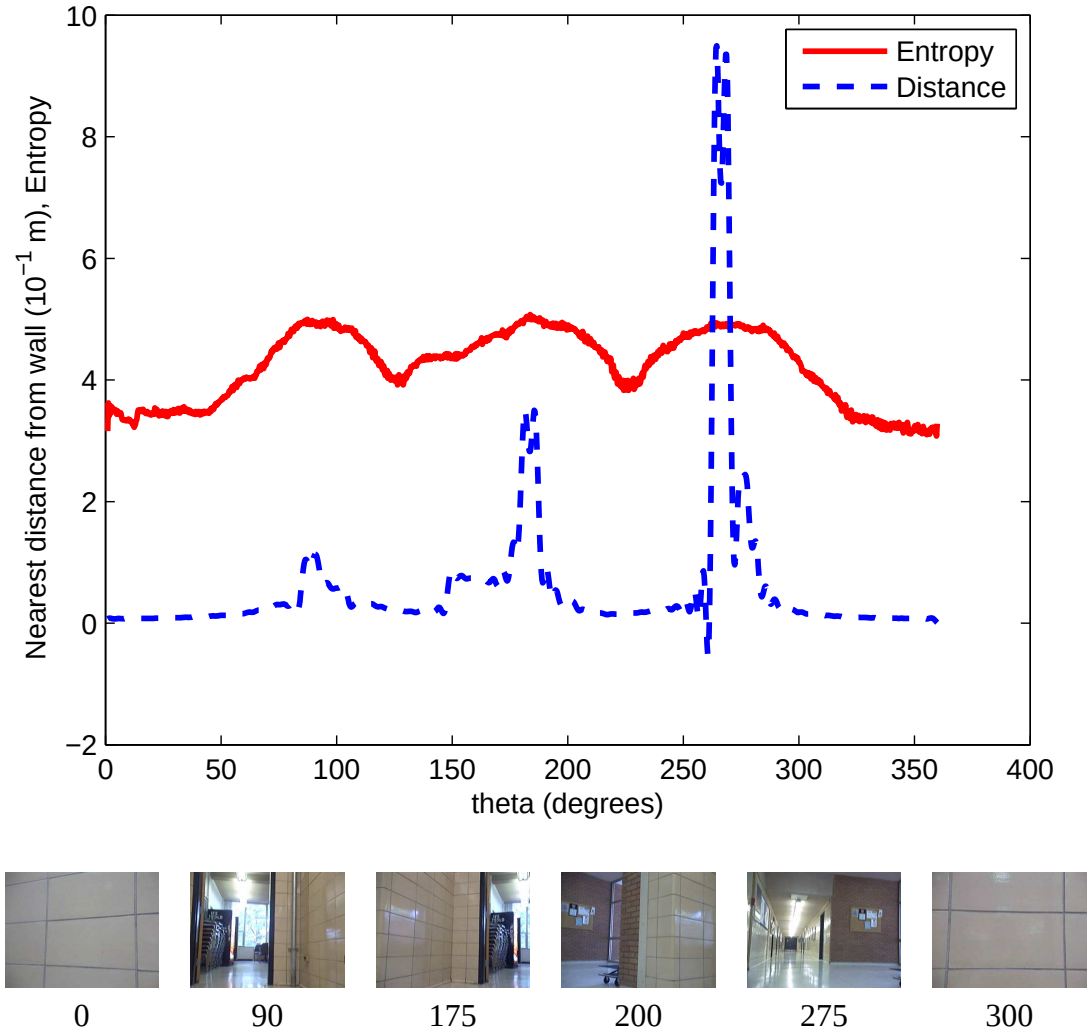

Figure 3.8: TOP: Entropy and distance (measured using a SICK laser scanner) as the robot turned at a corridor T-junction in Lowry (near main entrance, floor 1). BOTTOM: Images of the corridor approximately showing the orientation with respect to the depth values corresponding to them above.

which is a measure of the distance between two distributions [30]. In other words, it is a measure of the inefficiency of assuming that the distribution is q when the true distribution is p [14]. In our application, p_k and q_k represent the intensity histograms of two images, so that the relative entropy D measures how different one image is from the other. One drawback of the Kullback-Leibler measure is that it is not a true distance, because $D(p,q) \neq D(q,p)$. For

a symmetric measure, the Jeffrey divergence is used [57]:

$$J(p,q) = \sum_k \left(p_k \log \left(\frac{p_k}{q_k} \right) + q_k \log \left(\frac{q_k}{p_k} \right) \right). \tag{3.3}$$

Jeffrey divergence has been used previously for vision based robot localization for comparing color histograms in typical SLAM algorithms and has been shown to be a good metric for histogram comparison [57].

As the robot moves toward the end of the corridor in the homing phase described in the previous section, the current image is compared with the home image using Jeffrey divergence. This measures the amount of relative information between the two images, i.e., how different one image is from the other. The divergence value increases steadily as the robot moves, then the value increases rapidly as the robot approaches the end of the corridor. This rapid change signifies that the current image is no longer recognizable as 'home' (see Figures 3.9 and 3.10).

3.5.2 Time-to-collision detector

Time-to-collision (TTC) is defined as the time taken by the center of projection of a camera to reach the surface being viewed, if the relative velocity remains constant [24]. Horn *et al.* [24] have recently described a novel method to determine the time-to-collision using image brightness derivatives (temporal and spatial) without any calibration or tracking. This method computes the time to contact with just two frames of a sequence. Although each individual estimate is noisy, a filtered version (using a median filter of length 15) of the output yields a reliable estimate as the camera approaches the object. Of specific importance is the case of a planar surface for which the algorithm is simple and can be applied to the case of a robot approaching the end of a corridor. For the case of translation motion along the optical axis

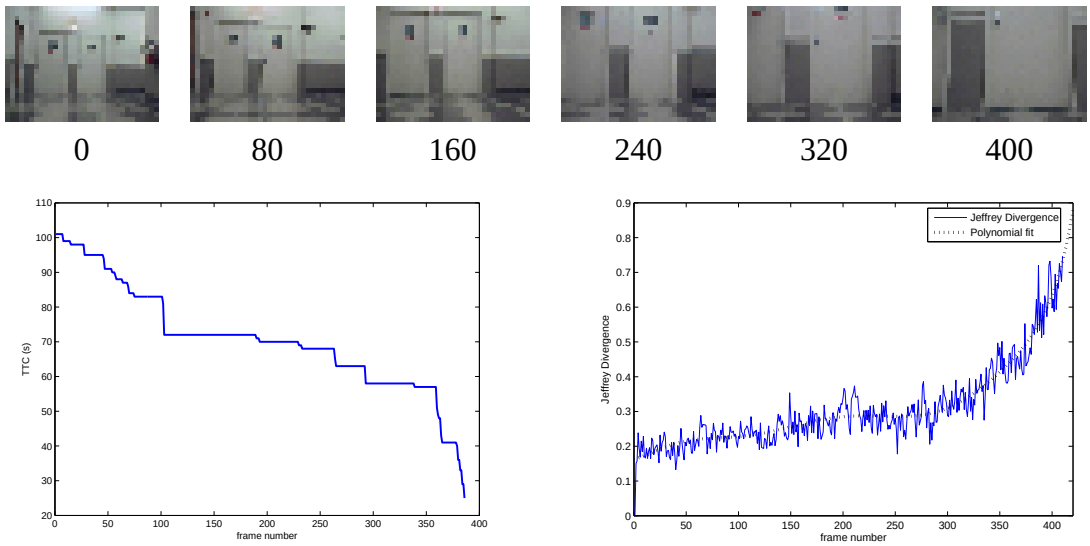

Figure 3.9: Time-to-collision and Jeffrey divergence for an image sequence in which the robot approaches a pair of doors in a textureless environment. TOP: Sample images from the sequence. BOTTOM: Plot of the TTC (left) and Jeffrey divergence (right) versus time. The former decreases, while the latter increases; combining the two enables robust detection of the end of a corridor.

towards a plane perpendicular to the optical axis, the TTC is given by

$$\tau_{TTC} = \frac{-\sum (G(x,y))^2}{\sum G(x,y)\, E_t}, \tag{3.4}$$

If E_x and E_y are spatial image brightness derivatives and E_t is the temporal derivative , then $G(x,y) = xE_x + yE_y$, and the summation is over the desired planar object (in some cases the entire image) [24]. For a detailed derivation from first principles, refer to the Appendix. Figures 3.9 and 3.10 show that the TTC decreases as the robot approaches the end of a corridor. It can be seen from these figures that the metrics describe the approaching end successfully in both textured ('information rich') and relatively textureless environments. By combining Jeffrey divergence and TTC ($J(p,q) \geq J_{th}\ \&\ \tau_{TTC} \leq T_{min}$) , the end of a corridor can be detected reliably.

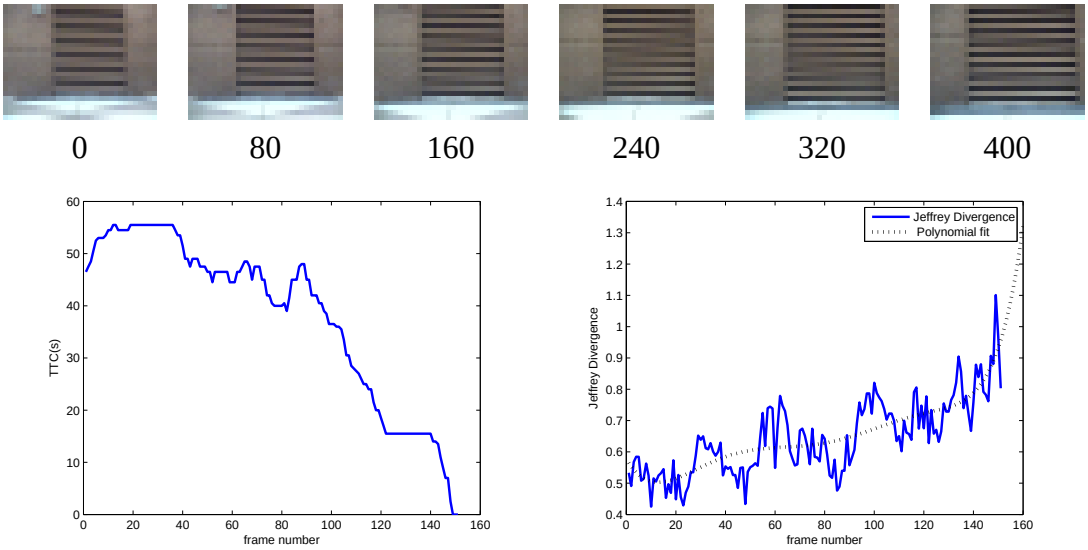

Figure 3.10: Time-to-collision and Jeffrey divergence for an image sequence in a textured environment in which the robot approaches a brick wall with a ventilator. TOP: Sample images from the sequence. BOTTOM: Plot of the TTC (left) and Jeffrey divergence (right) versus time.

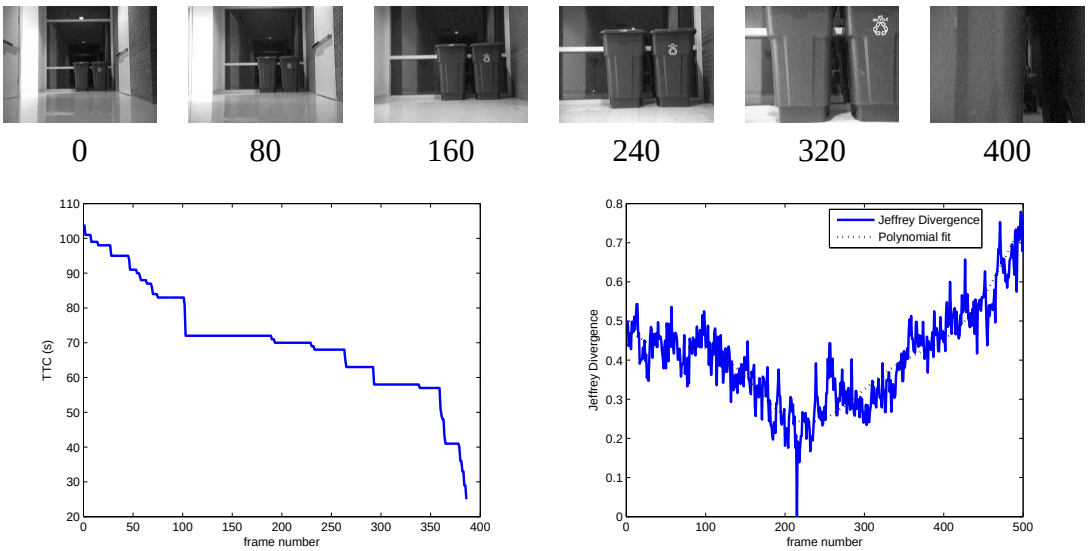

Figure 3.11: Time-to-collision and Jeffrey divergence for an image sequence in which the robot approaches a pair of trash cans with a glass door behind them in EIB, third floor. TOP: Sample images from the sequence. BOTTOM: Plot of the TTC (left) and Jeffrey divergence (right) versus time.

3.6 Turning at the end of a corridor

The robot displays tropism (an orienting response; a reaction to a stimulus source) at the end of each corridor, making an autonomous decision to turn in order to find the new adjacent corridor. While turning, the robot searches for ceiling lights and high overall entropy. The robot enters a rotational search mode until it finds another source light in the ceiling. If it sees the light, it corrects its course and follows the light into the new corridor following the same procedure as above. However, if it does not see any lights on all sides but still senses the presence of a corridor indicated by an entropy value greater than a threshold H_{high} (see Figure 3.5), then it navigates in that direction using 'homing' as described above and the process continues. If lights come into view again, the robot follows the light.

3.7 Autonomous mapping

The same metrics that were used for navigation — entropy and relative entropy — can be used to determine *distinctive/salient* landmarks for map building in an incremental process. Boada *et al.* [7] have shown a popular framework for Voronoi-based maps and localization. The Voronoi-based maps are roadmap methods and are preferred for corridor mapping because of their *accessibility, connectivity, and departability* [12] and can be constructed incrementally by the robot. In this approach, the graph consists of the links which represent the obstacle-free path followed by the robot and the nodes which represent the *distinctive/salient* places along the path.

As the robot drives down the corridor, not all images captured are *salient.* Just as a human driving down a highway often experiences long stretches of monotonous scenery broken by intermittent landmarks, the robot perceives salient regions along either side of the corridor a small percentage of the time. In our approach to mapping, the measures of image saliency

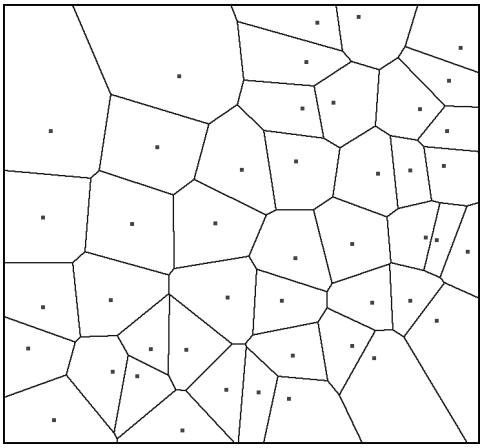

Figure 3.12: Voronoi tessellation of a plane based on a random set of 2D points. (Plot was generated using *VoroGlide*[1]).

described in the following sections indicate the presence of a nearby landmark by a sudden increase in their value. These salient images correspond to locations of landmarks in the scene. The path followed by the robot along the corridor is automatically augmented with salient locations that become nodes in a Voronoi-based graph. Together, these form a map representing the topology of the environment which can be used for later localization and navigation tasks.

3.7.1 Voronoi diagrams

A Voronoi diagram or tessellation is a decomposition of a metric space into a number of non-overlapping spaces based on the objects and boundaries in that space. Given a number of points in a plane the Voronoi diagram divides the plane according to the nearest-neighbor rule. That is, each point representing an object is associated with that region of he plane that is closest to it in the Euclidean sense. If p and q are two points in the plane and δ is the distance function, the domain of point p over point q is given by

$$dom\,\{p,q\} = \{x \in R_2 | \delta\,(x,p) \leq \delta\,(x,q)\} \tag{3.5}$$

The resultant tessellation from equation (3.5) is nothing but a half plane bounded by the perpendicular bisector of the straight line joining p and q [3]. Voronoi tessellations arise in nature and provide a visual interpretation of space (see Figure 3.12). Human intuition is supposedly guided by this visual perception described by the Voronoi diagram. Voronoi diagrams also have interesting mathematical properties like their duality with Delaunay triangles, and they act as a tool to solve computational problems. Voronoi diagrams have been used previously to determine path planning for a point robot. In this case the Voronoi diagram can be used to build a Hidden Markov Model of the landmark sequences to affect localization using the framework. The Voronoi diagram reduces the path planning/localization problem to bi-dimensional trajectory planning providing a simple and efficient path that is safest for the robot to follow. These also include the junction landmarks which form a part of the original Voronoi geometry. The total description is that of the links representing the free path followed by the robot with nodes representing the intersection of the links and additional nodes representing the natural and artificial visual landmarks detected along the two sides of the corridor. It is intuitive to describe landmarks and their *region of influence* using the Voronoi as shown in Figure 3.13. If the width and length of the corridor can be obtained, it is possible to describe actual areas which correspond to the landmark's influence. In many ways this is also crudely representative of the way the human mind identifies landmarks.

3.7.2 Joint probability distribution of distinct landmark measures

For landmark detection only one-sixth of the image is considered on either side (see Figure 3.16), because this narrow region contains landmarks as seen along a corridor. This further implies that only 33% of the 32×24 image is used. We determine distinct landmarks along the hallway by using the measures of image scalar entropy and relative entropy between two

[1] http://www.pi6.fernuni-hagen.de/GeomLab/VoroGlide/

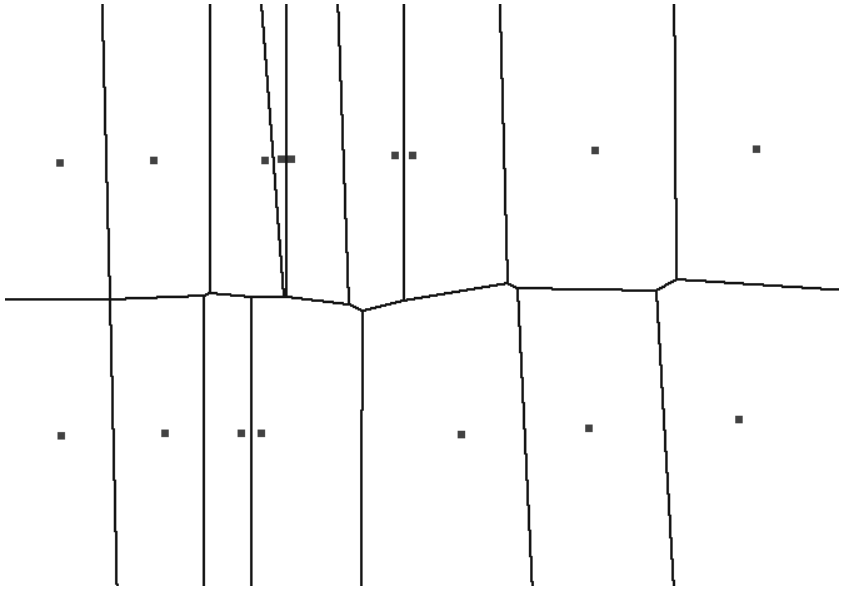

Figure 3.13: Voronoi interpretation of left and right landmarks. The space in the corridor can be tessellated by the points representing the landmarks for geometric mapping (Plot was generated using *VoroGlide*[1]).

subsequent images. Let X be a normalized random variable representing the entropy of the gradient magnitude of i^{th} image seen along the hallway, and let Y represent the Jeffrey divergence between the i^{th} and the $(i-1)^{th}$ image gradients. The joint density of two variables is generally plotted as a 2D surface, but in several cases we are interested in a time domain representation with peaks and valleys [44]. In such cases, the Joint Probability Density (JPD) of the two variables as a function of time t represents the *distinctiveness* of the image/event in the temporal sense as follows:

$$P_t(X,Y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\left(\frac{X^2}{2\sigma_x^2}+\frac{Y^2}{2\sigma_y^2}\right)\right]. \tag{3.6}$$

This can be described as a measure of how information-rich and unique an image is. A landmark is therefore defined as an image that has *interesting*, *recognizable* information that is distinct from the previous image. It is assumed that two consecutive frames in the sequence do not have two different potential landmarks. Considering the speed of the robot and the

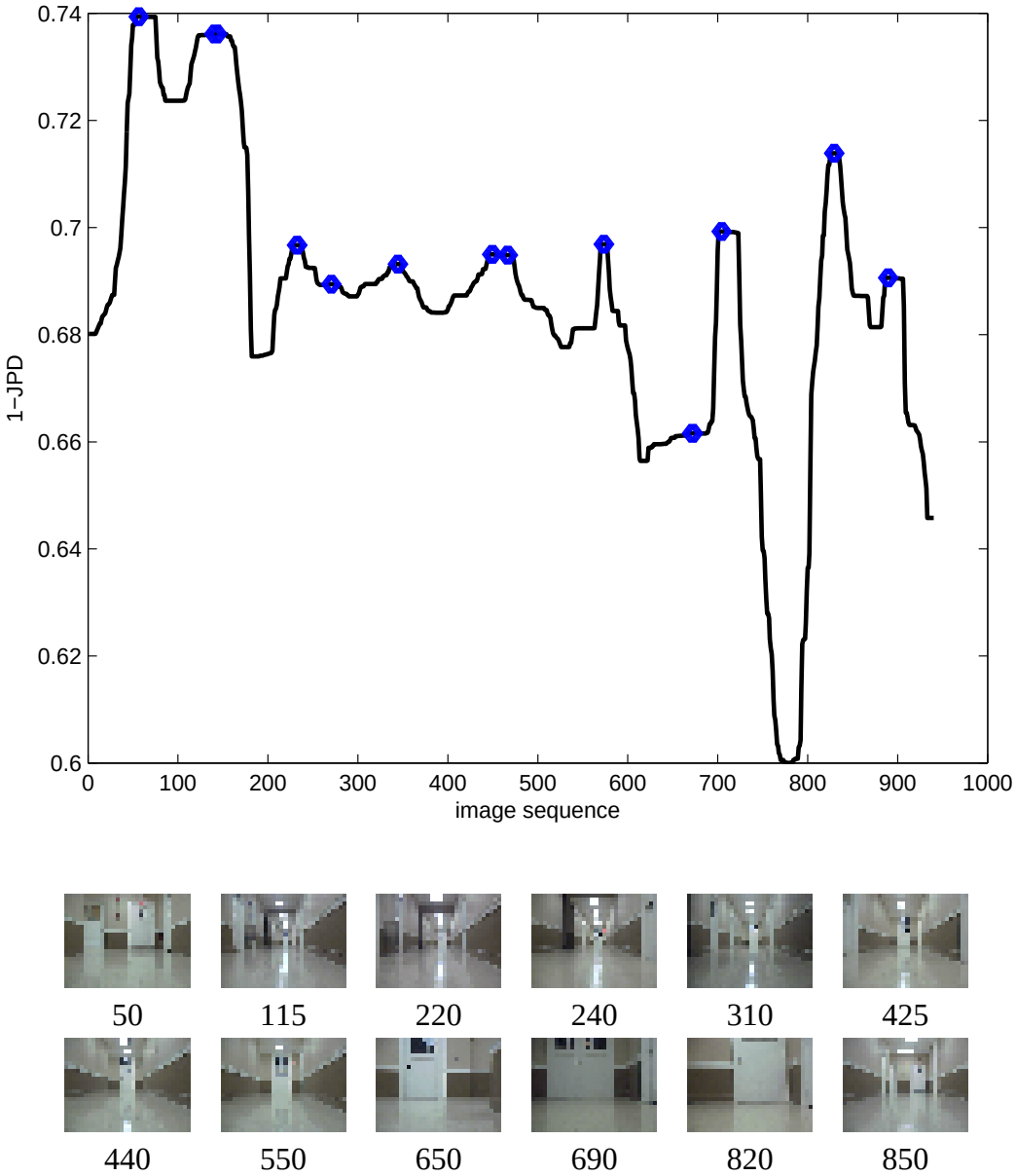

Figure 3.14: TOP: JPD for the left side of the corridor: Each peak corresponding to the local maxima defines a region of high saliency — a landmark. BOTTOM: Images corresponding to the peaks shown in sequence with respect to the peaks from left to right.

capture rate of the camera this assumption is generally true. However, this is a drawback of the algorithm as it may not generalize to different environments.

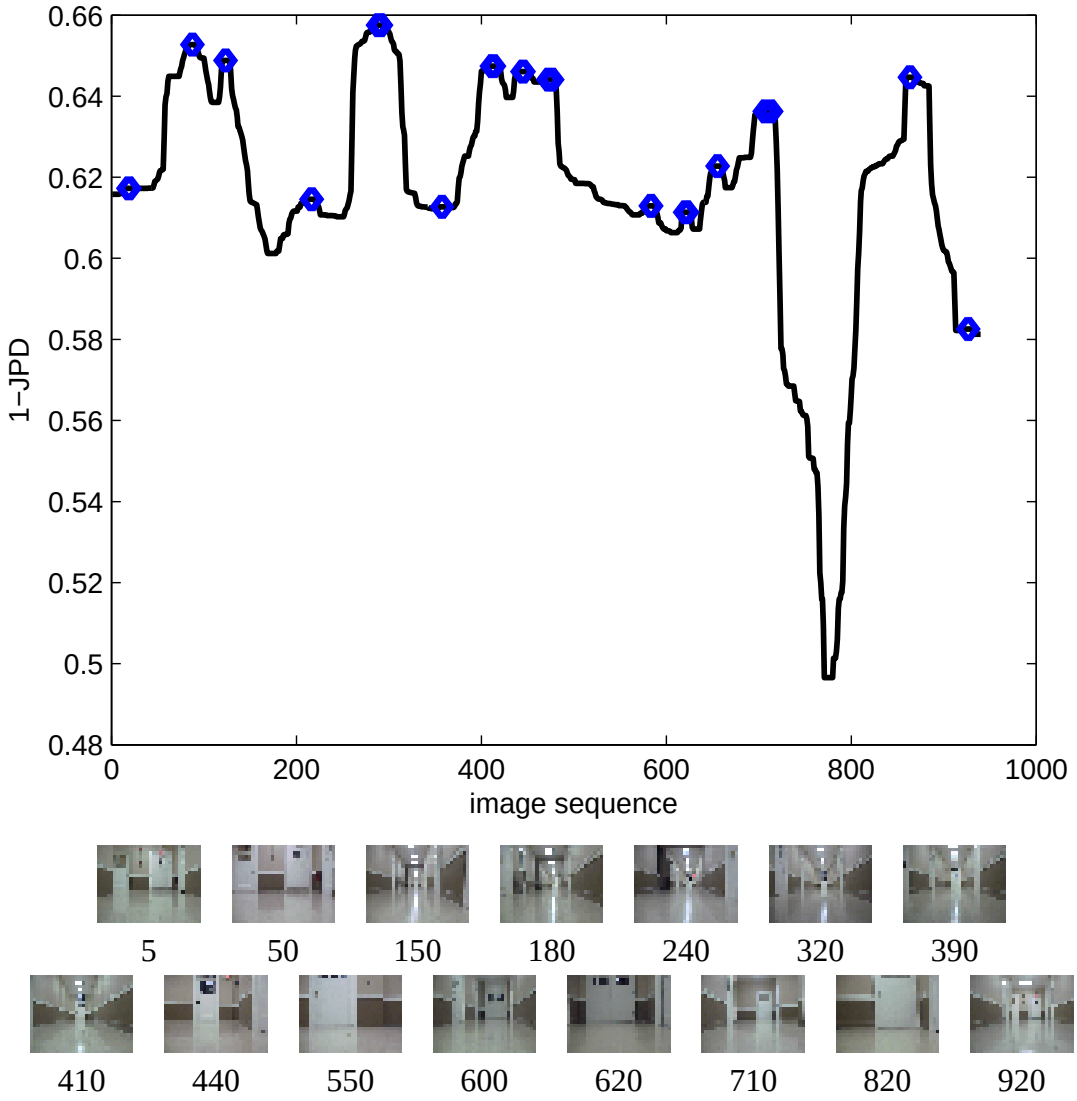

Figure 3.15: TOP: JPD for the right side of the corridor: Each peak corresponding to the local maxima defines a region of high saliency — a landmark. BOTTOM: Images corresponding to the peaks shown in sequence with respect to the peaks from left to right.

Because the relative entropy between two images is independent of the absolute entropy of either one, X and Y can be considered as independent variables and the distribution is plotted over time. Local maxima on the JPD give locations/images that represent landmarks (see Figures 3.14 and 3.15). It can be seen from the results in Figure 4.3 that even in images of low-resolution (where traditional point features are hard to detect/track) the simple measures

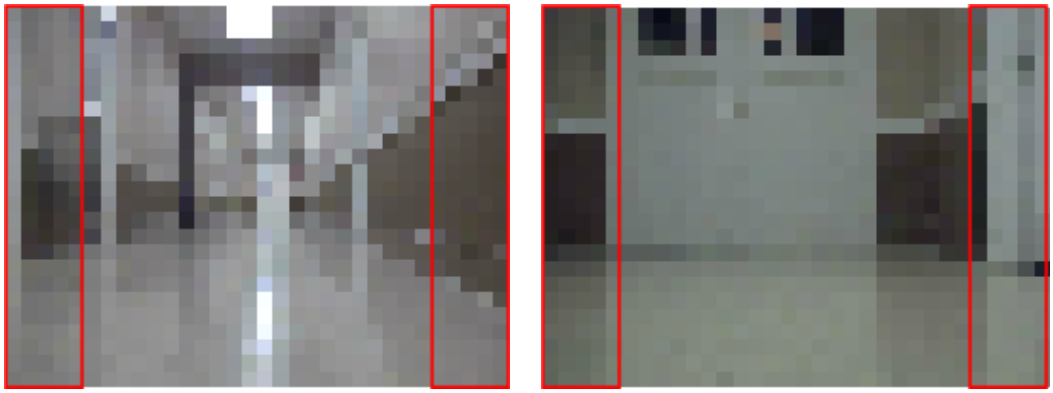

Figure 3.16: LEFT: Water fountain in the left one-sixth of the image. RIGHT: Door in the right one-sixth of the image

indicated above give a clear indication of a landmark. The algorithm does not represent each landmark uniquely (which would be difficult in a typical indoor environment consisting of corridors with identical doors) but instead represents locally the presence of a landmark.

3.8 Geometric correction of odometry

Since the robot's odometry is prone to drift over large distances, these plots include an effective method to reduce the drift using the motor commands given by the vision module. It is important to note that this correction was done mainly for display purposes and for mapping and does not play any role in navigation. Inspired by the work of Crowley [15], which combines the measured position and the expected position based on motor commands using a Kalman filter and a retroactive odometric correction using sensor fusion [32], we simply use the motor commands issued by the vision module to incrementally update the odometry. It is important to note that this correction was done only for the driving mode (straight line navigation using ceiling lights and homing) and only for updating the heading of the robot. During the turning mode there is no vision control till the ceiling lights and high entropy are seen.

Therefore drift error at turns persists in rotation. But this incremental method is sufficient for the general purpose of this initiative and for a display plot (see Figure 3.17).

Let t_{module} be the time taken for one iteration of any vision module that controls the robot and let ω_v be the desired rotational velocity command sent to the robot in one iteration. Then the estimated angle by which the robot turned in one iteration is given by

$$\theta_r = \omega_v t_{module} \tag{3.7}$$

where ω_v is used to control the rotational velocity of the robot. The robot's heading is updated using the value θ_r calculated as above in the centering and homing modes. This reduces the drift error considerably as the ceiling lights enable a fairly stable straight line navigation. The robot's odometry is not updated in the turning mode.

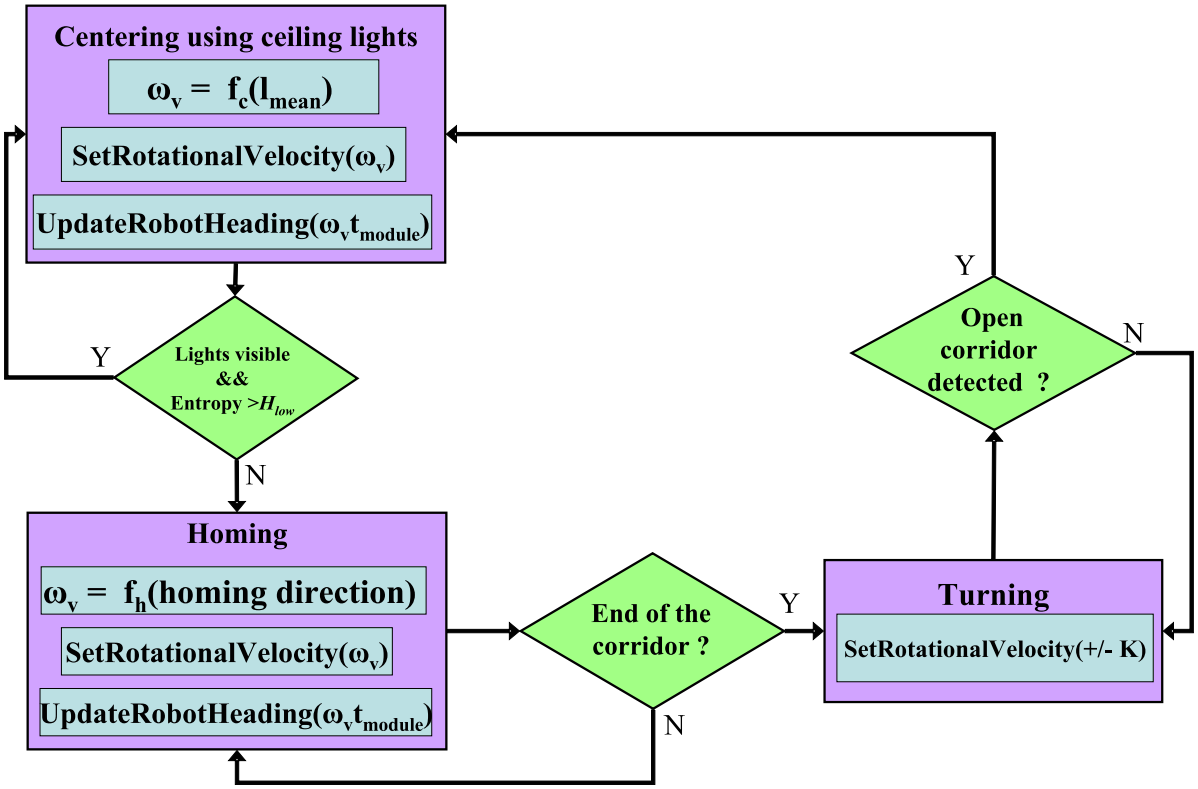

Figure 3.17: A flow diagram explaining the odometry correction. Odometry heading is corrected using visual control output during centering using ceiling lights and homing. t_{module} is the time taken for one iteration of either of these visual modules. In turning mode, the rotational velocity is a constant +/- K and the robot's odometry is trusted.

Chapter 4

Experimental results

4.1 Platform and primary test environment

Figure 4.1: The Pioneer 3 robot used for all the experiments. More that 50 hours of experiments were run on this setup.

The algorithm was tested on an all-terrain robotic platform Pioneer III manufactured by ActivMedia Robotics. We mounted a forward-facing Logitech Quickcam Pro4000 webcam on it as shown in Figure 4.1. ARIA is the interface used for interfacing the portable PC/laptop with the microcontrollers on board the robot which are responsible for the low level motor controls and odometry. The command center is the portable PC and the application was built using Visual C++. The algorithm was tested in all the floors (basement and floors 1, 2, and 3) of Riggs Hall on our campus. For historical reasons, the three floors do not share the same appearance in terms of the color of the walls, the placement of the lights, the locations of the doors, the locations of nearby corridors, and so on. In particular, the corridors have entirely different lighting conditions, ranging from a single row of fluorescent lamps to sodium vapor lamps or lights on either sides of the corridor ceiling (see Figure 4.2). The texture (information content) in the corridors is also different, with the basement having textureless walls and floors of uniform color (see Figure 4.2). Only the grayscale information from the 32×24 downsampled images from the camera was used.

4.2 Navigation

On all four floors the robot autonomously navigated the corridors, turning at the end of each corridor using the algorithm described. At the end of a corridor, the robot turned left arbitrarily by ninety degrees and then right, searching for lights and high entropy; otherwise the robot turned in the open direction based on high entropy. Figures 4.3, 4.4, 4.5, and 4.6 show the path followed by the robot on all of the floors, overlaid on a hand-constructed map of the environment to provide context for interpreting the results. In the basement the robot turned right, then turned left (arbitrarily), navigated to the end of the corridor, then turned around 180 degrees and headed back down the last short corridor in the opposite direction. On the third floor the robot turned left twice at the end of each corridor, and repeated the process in

Figure 4.2: Example experimental sites shown in high-resolution to reveal the difference in texture and lighting. TOP LEFT: Riggs basement, TOP RIGHT: Riggs floor 1, BOTTOM LEFT: Riggs floor 2, BOTTOM RIGHT: Riggs floor 3.

the short wing of the corridor. In the first floor the robot turned right twice at the end of each corridor, and in the second floor, the robot turned left twice at the end of each corridor. In all cases the robot remained close to the center of the corridor, avoiding collision with the walls or static obstacles.

4.3 Mapping

Figures 4.3, 4.4, 4.5 and 4.6 also show the generated Voronoi-based map overlaid. Most of the important landmarks have been captured. The nodes represented in the Figure 4.6 represent distinctive regions along the corridor of the third floor. With odometry combined

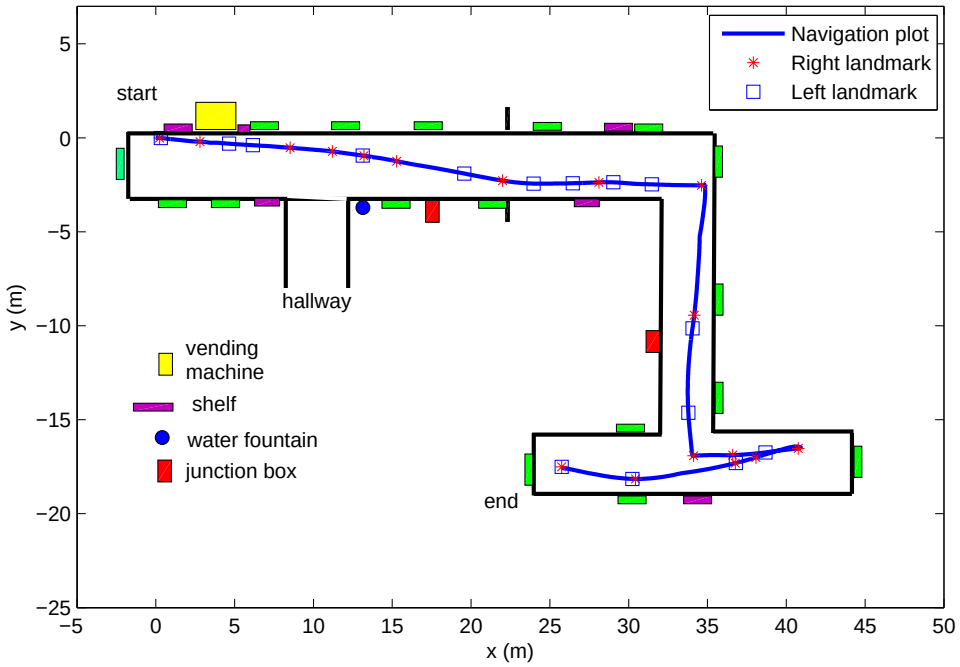

Figure 4.3: Automatically generated Voronoi map of the basement of the building.

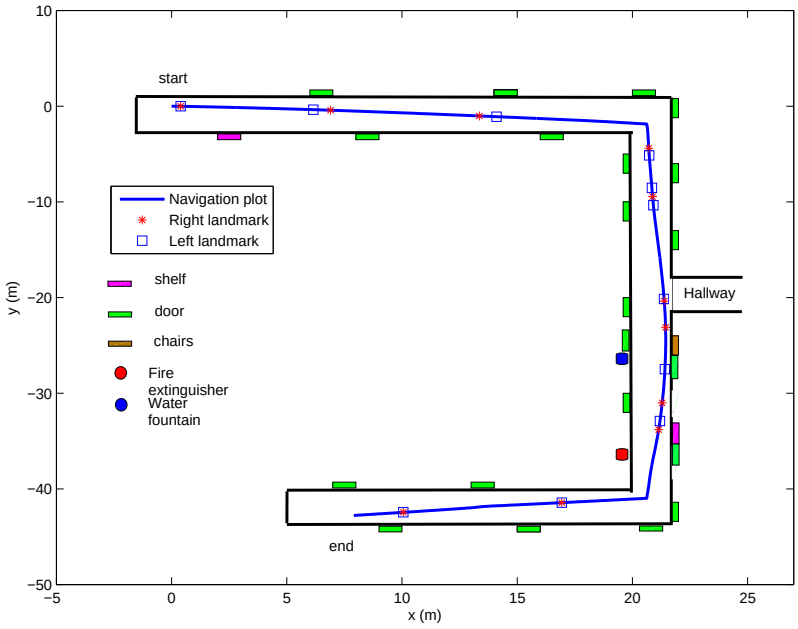

Figure 4.4: Automatically generated Voronoi map of the first floor of the building.

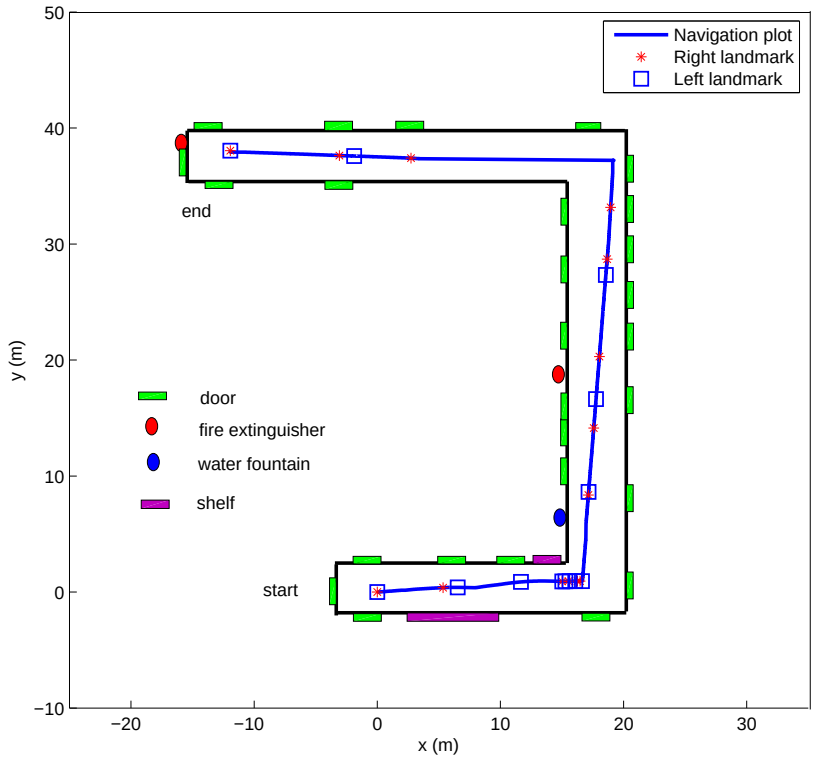

Figure 4.5: Automatically generated Voronoi map of the second floor of the building.

Location(Riggs)	N_L	N_D	F	M	% success
Basement	15,14	15,13	1,2	1,1	93.3, 78.5
Floor 1	12,11	10,10	2,0	2,1	66.67, 90.9
Floor 2	13,15	14,11	4,1	2,5	76.92, 66.6
Floor 3	12,13	14,13	4,3	2,3	83.3, 76.9

Table 4.1: Quantitative landmark detection results. From left to right: the number of landmarks N_L, the total number detected by the algorithm N_D, the number of false landmarks detected F, the number of landmarks missed by the algorithm M, and the percentage of success in detecting landmarks. Each cell in the table contains the number for left and right, separated by a comma.

it can be described as a topo-geometric map similar to the description in [7] as it combines real distances with the skeleton. The landmarks seen to the left of the robot are represented by a square, and the landmarks seen on the right are represented by an asterisk. At corridor

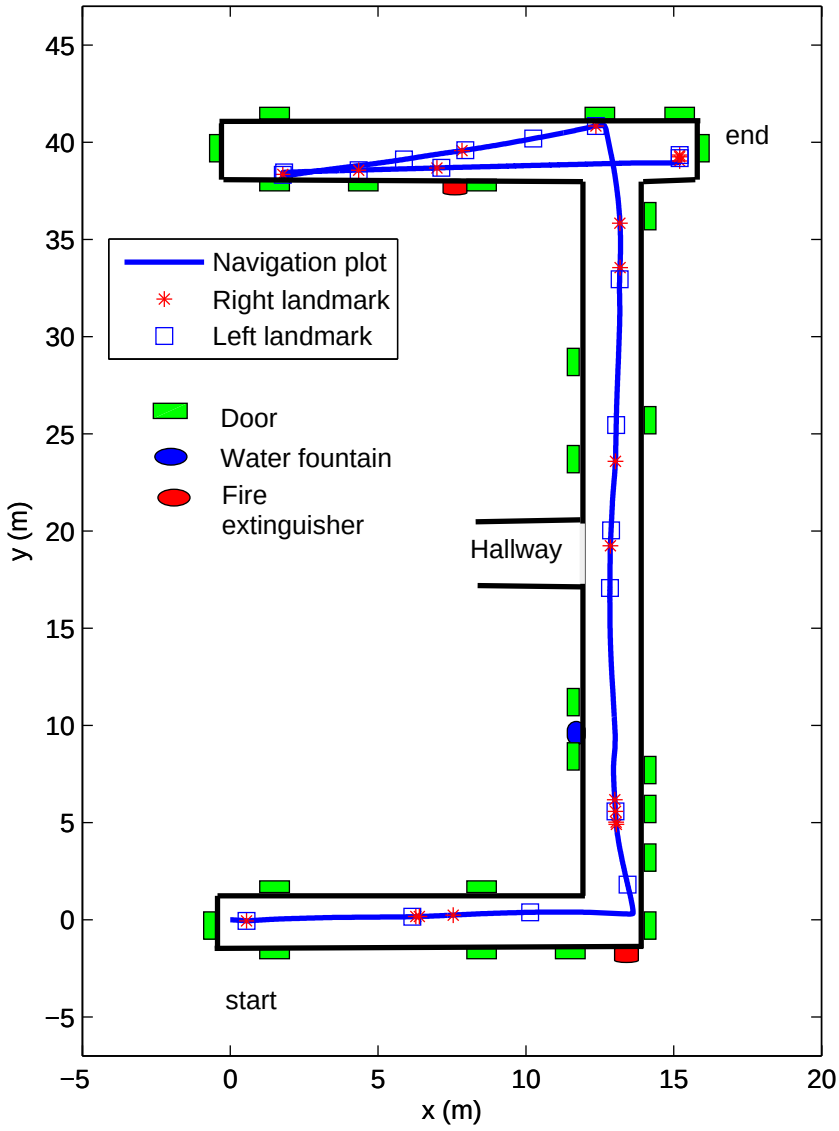

Figure 4.6: Automatically computed Voronoi-based map with nodes representing the approximate distinctive landmarks on the third floor of the building. It can be seen that the landmarks have been verified by the returning robot in the top wing of the corridor.

junctions it can be seen that left and right landmarks overlap. This is because the robot turns at junctions to search for lights. Furthermore, the multiple doors at junctions are recognized as one landmark because they are all captured during the rotation of the robot at junctions. It is interesting to observe the top wing of the corridor in Figure 4.6. The left and right landmarks

Figure 4.7: Landmark images containing a landmark on the left side of the image.

Figure 4.8: Landmark images containing a landmark on the right side of the image.

validate each other because the robot returns along the same path in the opposite direction. Some example enlarged images with left and right landmarks are shown in Figures 4.7 and 4.8.

Table 4.1 shows the analysis of the results. The landmarks are counted in the order of the robot's navigation path, while the returning landmarks are not counted. Also note that in some cases two entities that are immediately next to each other are detected as one distinct region/landmark (e.g., a door with an adjoining shelf on the wall). This detection process is a simple probabilistic estimation and not efficient enough because not all the landmarks are successfully detected. More work needs to be done to define a robust measure using the saliency metrics.

4.4 Odometry drift correction

Good improvement in odometry is seen after drift correction. But since correction is not done in the turning mode, and only the heading is corrected, drift errors are not completely eliminated. The correction procedure is detailed in section 3.8. However, the efficiency of the

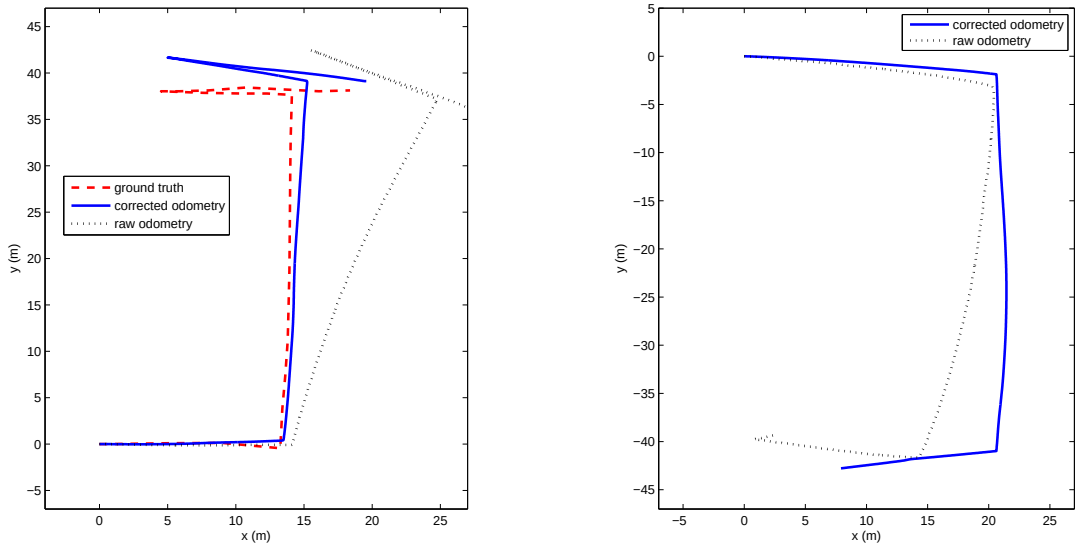

Figure 4.9: LEFT: Ground truth versus corrected odometry and raw odometry for navigation in Riggs floor 3: observe that drift is reduced in the corrected odometry. RIGHT: Raw versus corrected odometry for navigation in Riggs floor 1: Note the improvement. NOTE: The ground truth was determined by placing markers manually along the path traveled by the robot and then taking measurements of the markers in the corridor.

navigation route should not be tested by odometry alone as sufficient errors persist over long distances. In order to estimate the extent of error, an experiment was conducted by manually measuring the navigation path of the robot by placing markers at regular intervals. The ground truth as determined by the markers and the navigation as determined by the corrected odometry and the raw odometry are overlaid in Figure 4.9, for Riggs floor 3. A plot of corrected versus raw odometry is also shown for Riggs floor 1 and in both cases, the improvement seen by the odometry correction is considerable.

Location	No. of successful trials / No. of total trials	Percentage success
Basement	$2/2$	100%
Floor 1	$2/2$	100%
Floor 2	$2/2$	100%
Floor 3	$8/8$	100%

Table 4.2: Repeatability: The number of successful trials of repeated runs in each of the floors.

4.5 Analysis of results

4.5.1 Experimental trials

The robot was tested successfully on all the four floors in Riggs multiple times. Success in all of these cases were measured by whether the robot was able to start off at one end of the corridor and reach the other end without any manual intervention (see Table 4.2). The analysis of these results tend to be more of a qualitative nature than quantitative because of the goal of this initiative. This is true of most systems interacting with a complex environment where the factors controlling the success of the system are too many to be elaborately estimated in a quantitative manner.

Also these results are run on a real indoor environment with emphasis on the navigation aspects. For this reason, the system was not tested on any arbitrary dataset. Most datasets available are suited for SLAM like environments that use high-resolution based processes with very few frames/images of a given environment.

On each floor the robot was able to navigate successfully at least twice and has been tested successfully in the third floor for eight trials. An overlay of four of those trials is shown in Figure 4.10.

To test the robustness of the system, several long trials were conducted. In the last wing of the third floor corridor, the robot successfully continued navigating for approximately **45 minutes** navigating more that **850 meters** autonomously as shown in figure 4.11. It would

Image Conversions and display	$0.04ms$
Jeffrey Divergence	$0.02ms$
Time-to-collision	$0.26ms$
Homing	$0.08ms$
Get the mean of ceiling lights	$0.06ms$
Entropy	$0.06ms$
Downsampling	$0.18ms$
Total	$0.70ms$

Table 4.3: Performance: Time taken by the different vision modules.

have continued running, but it had to be stopped manually because the laptop battery had completely discharged. In the basement, the robot ran successfully for **20 minutes** in the main wing (The side door was closed for this experiment, blocking the connecting corridors). Although it successfully drove past a brightly lit vending machine at one end several times, eventually it mistook it for ceiling lights and had to be manually stopped to prevent it from crashing into it.

The robot showed successful navigation in different situations like when it was started facing a nearby wall as shown in Figure 4.12 and when it was started very close to a wall as seen in Figure 4.13. In both cases it is seen that the robot recovers quickly from the difficult initial conditions and proceeds navigating in the expected manner. In both these trials it is seen that acceptable changes in actual starting position, orientation and location do not affect the navigation of the robot. The measures of entropy helps the robot to recover from walls and dead ends and continues navigation by searching for lights and high entropy as described earlier.

4.5.2 Computational efficiency

The algorithm is efficient, capable of running at over 1000 frames per second (see Table 4.3). Therefore with a standard 30 Hz camera, the algorithm consumes approximately 3% of the CPU, thus freeing the processor for other concurrent tasks. In our experiments, the robot was

run indoors at a moderate speed of 0.4 m/s for two reasons. One reason was to avoid contingencies in the indoor environment because the system does not currently support dynamic obstacle avoidance. The other reason was that near the end of the corridor, the robot has to stop and turn. It needs to maintain a constant speed throughout because of TTC calculation. When it stops, the momentum allows the robot to move a few inches beyond that desired. This needed to be controlled by a reasonable speed. Future work will involve dynamic speed control at the end.

4.6 Other environments and failure modes

[h] Though the algorithm is not specialized for a particular environment, it currently works well only for a certain class of environments, namely those with ceiling lights symmetrically distributed in the corridor, no specular surfaces or structures near the actual lights and there is not much graffiti on the lower ends of walls/doors. The structure also assumes that corridors are placed at right angles to each other because in the turning mode the robot searches for corridors from -90° to $+90^{\circ}$. The robot failed in Fluor Daniel building (EIB), because of the complex structure of the reception hallway. Ceiling lights here were not visible from a forward facing camera, and one side of the hallway was enclosed by glass looking outdoor. The system also failed in Lowry hall because of glass panels situated on either sides of walls that provided a specular distraction. Simple experiments run for TTC and Jeffrey divergence at corridor ends in different environments yielded successful results (see Figure 3.11).

In another trial in the basement, it had to be stopped at the very end because of a double glass door present at the end (see Figure 4.14). The basement provides a challenging environment for the robot because of the vending machines and highly reflective walls and floors. The problem with glass doors is two-fold. Firstly they reflect light and confuse the robot. Secondly, when the robot approaches a glass door, it sees the objects behind it and leads to erroneous

estimation of TTC and Jeffrey divergence. The main weakness of this algorithm is the use of ceiling lights. A continuous measure of ceiling symmetry rather than ceiling lights might overcome the problems with reflections and also allow navigation in different environments.

Landmark detection in floor 2 was poor because of the specular reflections from the white laminated posters on the walls. Landmark detection is also affected by the navigation path of the robot. If the robot navigates very close to a wall for a period of time, then during that time several landmarks are missed or wrongly detected. These factors also contribute to the large number of false positives in the detected landmarks.

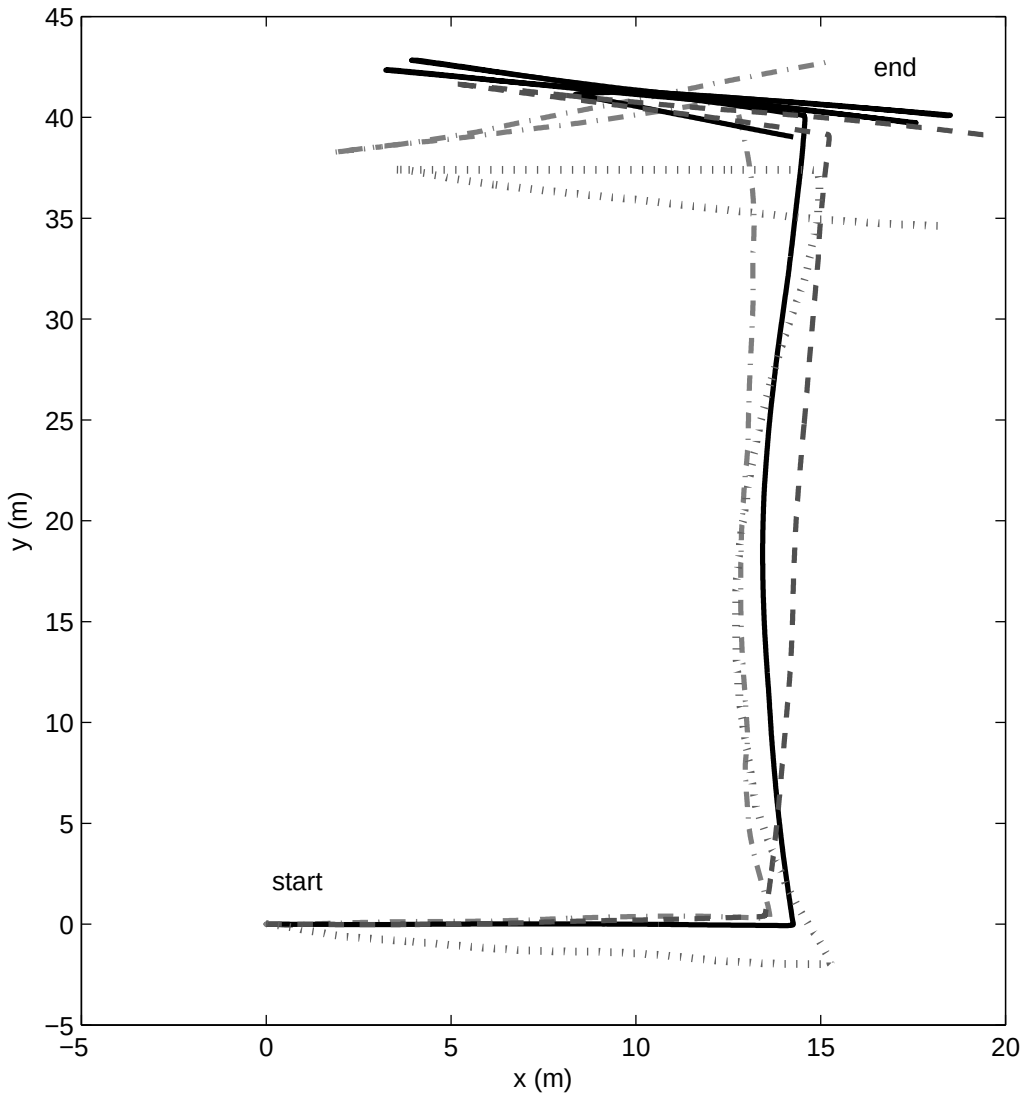

Figure 4.10: Four trial runs on the third floor of Riggs. It can be seen that the route has been successfully re-traced. The error seen at the last wing is due to accumulation of odometric drift. Though this was corrected using the vision module motor commands, some drift persists due to variation in processing time and image capture delay.

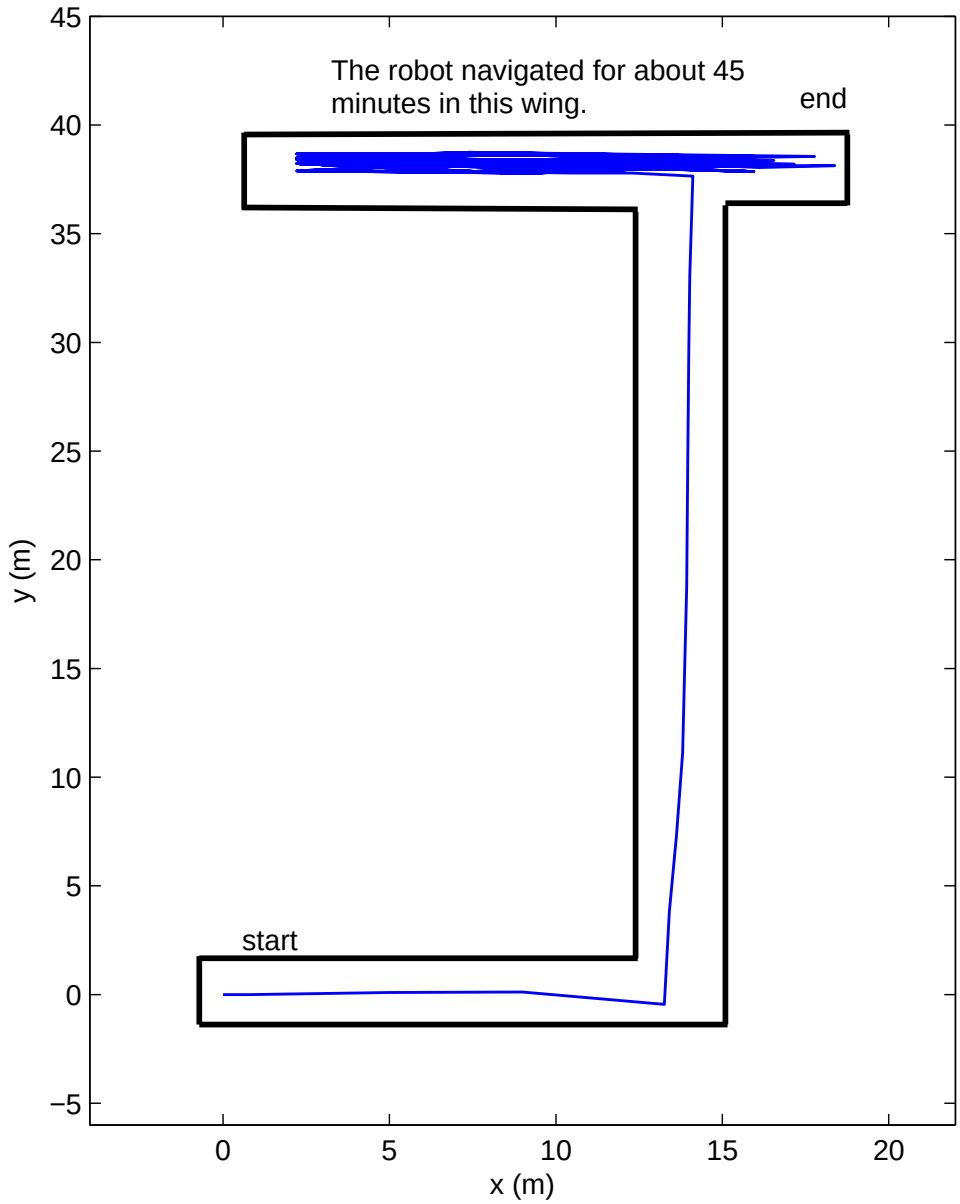

Figure 4.11: Robot navigated Riggs floor 3 and runs for about 45 minutes in the last wing of the corridor autonomously. The navigation path was measured manually using markers at regular intervals (ground truth). The robot navigated a distance greater that 850 meters in this trial.

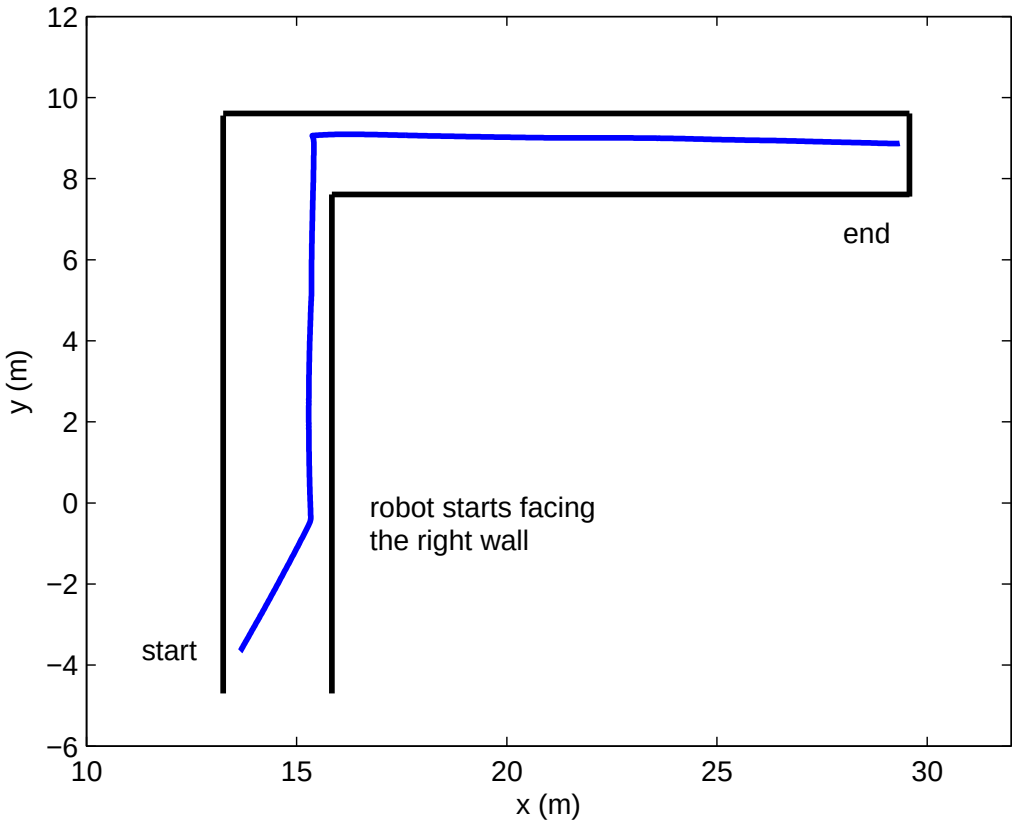

Figure 4.12: Robot starts facing the right wall in Riggs floor 3, recovers before its reaches the wall, turns and continues navigation.

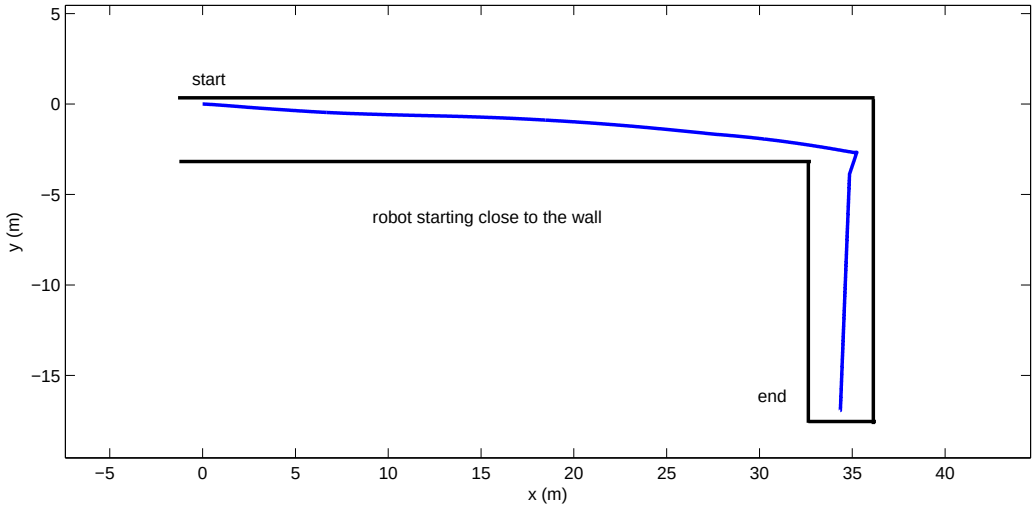

Figure 4.13: Robot starts very close to a wall in Riggs floor 3, corrects its path using ceiling lights (correcting orientation and horizontal position) and continues navigation.

Figure 4.14: Three examples of navigation failure. LEFT: Lowry Hall, glass panel on top right, CENTER: EIB: Glass structure on one side and ceiling lights not effective, RIGHT: Riggs basement: Double glass door.

Chapter 5

Conclusion and future work

The navigational behavior of a mobile robot is modeled by a set of visual percepts that work in conjunction to correct its path in an indoor environment based on different metrics. Special emphasis is placed on using low-resolution images for computational efficiency and metrics that capture information content and variety that cannot be represented using traditional point features and methods. The resultant algorithm enables end-to-end navigation in indoor environments with self-directed decision making at corridor ends, without the use of any prior information or map. The system forms the basis of an autonomous mapping system that is built using the same low-resolution metrics to present a Voronoi-based topo-geometric map that can be used for robot localization.

The primary contribution of this work is the implementation using low-resolution images that yields a high computational efficiency without sacrificing robustness. This work also takes a small step towards combining perceptual navigation with mapping and localization. Though the navigational competencies are simple and fairly specialized, they have proved to work well in a class of indoor environments and most importantly continued navigation has been achieved by the decision making at corridor ends. All of these built into a system with a

simple mapping capability added, work well in a given typical building with stable navigation seen across all the floors (which have different appearances).

Future work may involve several activities that make the existing algorithm more robust, making it environment independent, achieving localization with the given mapping algorithm, or using machine intelligence to train the system in an indoor environment to learn typical information content along the corridor. Another goal associated with the mapping could be the development of a layered approach where higher resolution image processing will augment the system to handle complex requirements like landmark matching. The Joint Probability Distribution can be made more robust using multiple temporal derivatives and smoothing. The ultimate goal is to achieve local-global localization by an autonomous navigating robot in an environment-independent manner.

APPENDIX

Time-to-contact relative to a planar surface: derivation from first principles

Horn *et al.* [24] have described the calculation of time-to-contact using spatial and temporal image derivatives and can be computed using just two frames in a sequence. The system does not use any tracking or calibration.

If the distance from the approaching surface is Z, then the time-to-contact is defined as

$$T = \frac{-Z}{\frac{dZ}{dt}}$$

where $\frac{dZ}{dt}$ is the derivative of the distance with respect to time. According to perspective projection equations, if S is the length of the cross-section of the approaching planar object and s is the size of its image, then, $\frac{s}{f} = \frac{S}{Z}$ where f is the focal length of the camera, which implies that $S\frac{dZ}{dt} + Z\frac{ds}{dt} = 0$ (see Figure 5.1).

Then $S\frac{dZ}{dt} = -Z\frac{ds}{dt}$ which implies that

$$\frac{-Z}{\frac{dZ}{dt}} = \tau_{TTC} = \frac{S}{\frac{ds}{dt}} \tag{A-1}$$

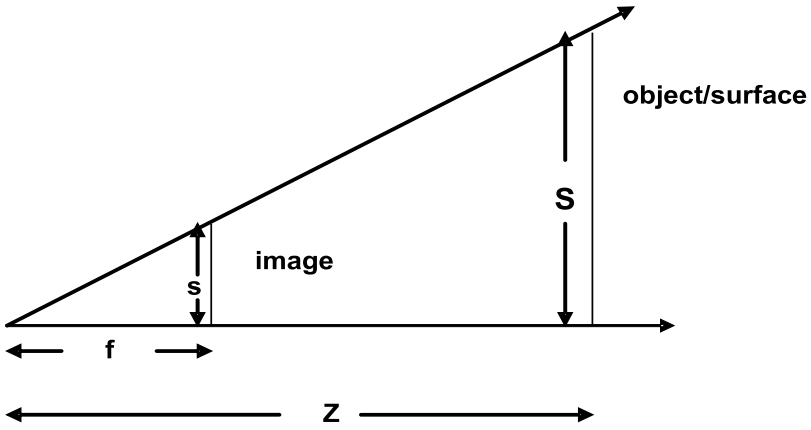

Figure 5.1: Perspective projection.

Considering the brightness constancy assumption of an image E, (the brightness of a pixel corresponding to a point on the object does not change with time), we have

$$E\left(x+\Delta x, y+\Delta y, t+\Delta t\right) = E(x,y,t)$$

Assuming small motion between successive frames (small Δx and Δy), the above can be expanded using first order Taylor's series to get

$$E_x\frac{dx}{dt} + E_y\frac{dy}{dt} + E_t = 0$$

or

$$uE_x + vE_y + E_t = 0 \qquad \text{(A-2)}$$

where $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$ respectively , $E_x = \frac{\partial E}{\partial x}$ and $E_y = \frac{\partial E}{\partial y}$ are spatial image brightness derivatives and $E_t = \frac{\partial E}{\partial t}$ is the temporal brightness derivative.

Once again obtaining perspective projection equations of the camera

$$\frac{x}{f} = \frac{X}{Z} \qquad \text{and} \qquad \frac{y}{f} = \frac{Y}{Z} \qquad \text{(A-3)}$$

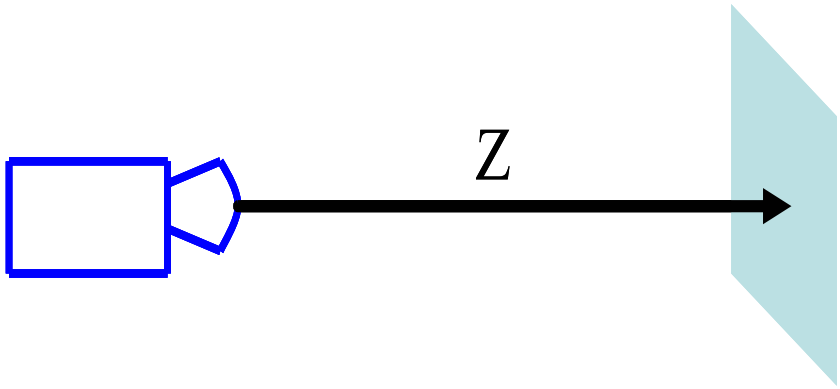

Figure 5.2: Camera moving such that optical axis is perpendicular to the approaching surface.

where X, Y, and Z are coordinates of a point in space and x, y are corresponding image coordinates. Differentiating equation (A-3) with respect to time, we get

$$\frac{u}{f} = \frac{U}{Z} - \frac{X}{Z}\frac{W}{Z} \qquad \text{and} \qquad \frac{v}{f} = \frac{V}{Z} - \frac{Y}{Z}\frac{V}{Z} \tag{A-4}$$

where U, V, W are temporal derivatives of X, Y, Z respectively and represent velocity of the point on the object relative to the camera. u, v are temporal derivatives of x and y (the motion field in the image). Substituting equation (A-3) in equation (A-4), we get

$$\frac{u}{f} = \frac{U}{Z} - \frac{x}{f}\frac{W}{Z}, \qquad \text{and} \qquad \frac{v}{f} = \frac{V}{Z} - \frac{y}{f}\frac{V}{Z} \tag{A-5}$$

which leads to

$$u = \frac{1}{Z}\left(fU - xW\right) \qquad \text{and} \qquad v = \frac{1}{Z}\left(fV - yW\right) \tag{A-6}$$

Considering the simple case where the translation is perpendicular to the optical axis (see Figure 5.2), U and V can be set to 0 in equation (A-6).

$$u = -x\frac{W}{Z} \qquad \text{and} \qquad v = -y\frac{V}{Z} \tag{A-7}$$

Substituting equation (A-7) in equation (A-2), we get

$$-\frac{W}{Z}\left(xE_x + yE_y\right) + E_t = 0 \tag{A-8}$$

or

$$CG + E_t = 0 \tag{A-9}$$

where $C = -\frac{W}{Z}$ from equation (A-1) and is the inverse of TTC, and $G = xE_x + yE_y$.

Formulating a least squares method to minimize $\sum \left(CG + E_t\right)^2$ where the sum is over all pixels of interest, which could be the whole image, we get

$$\sum \left(CG + E_t\right) G = 0$$

$$C = \frac{\sum G\left(x, y\right) E_t}{-\sum (G\left(x, y\right))^2} \tag{A-10}$$

It is evident that when C increases, TTC decreases. Intuitively this explains that as the camera approaches the surface being viewed, the temporal change in brightness values increases rapidly and the spatial change decreases (because when the object/surface grows bigger, the sum of spatial gradients is lower), and therefore the TTC decreases as the object/surface looms closer to the camera.

Bibliography

[1] N. Ancona and T. Poggio. Optical flow from 1D correlation: Application to a simple time-to-crash detector. In *Proceedings of the 4th International Conference on Computer Vision*, 1993.

[2] Jeffrey Andre, D. Alfred Owens, and Lewis O. Harvey, Jr., editors. *Visual perception : The influence of H. W. Leibowitz*. Washington, DC: American Psychological Association, 2003.

[3] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[4] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *9th European Conference on Computer Vision*, Graz Austria, May 2006.

[5] G. Blanc, Y. Mezouar, and P. Martinet. Indoor navigation of a wheeled mobile robot along visual routes. In *Proceedings of the International Conference on Robotics and Automation*, pages 3354–3359, 2005.

[6] J. L. Blanco, J.A. Fernández-Madrigal, and J. Gonzalez. A new approach for large-scale localization and mapping: Hybrid metric-topological SLAM. In *Proceedings of the International Conference on Robotics and Automation*, 2007.

[7] B. L. Boada, D. Blanco, and L. Moreno. Symbolic place recognition in Voronoi-based maps by using hidden Markov models. *Journal of Intelligent and Robotic Systems*, 39(2):173–197, 2004.

[8] 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.

[9] Joachim M. Buhmann, Wolfram Burgard, Armin B. Cremers, Dieter Fox, Thomas Hofmann, Frank E. Schneider, Jiannis Strikos, and Sebastian Thrun. The mobile robot RHINO. *AI Magazine*, 16(2):31–38, 1995.

[10] Zhichao Chen and Stanley T. Birchfield. Qualitative vision-based mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2686–2692, May 2006.

[11] K. Choi, S. Bae, Y. Lee, and C. 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*, volume 3, 2003.

[12] Howie Choset, Ilhan Konukseven, and Alfred Rizzi. Sensor based planning: A control law for generating the generalized Voronoi graph. In *Proceedings of the IEEE International Conference on Advanced Robotics*, 1997.

[13] D. Coombs, M. Herman, T. Hong, and M. Nashman. Real-time obstacle avoidance using central flow divergence and peripheral flow. In *Proceedings of the 5th International Conference on Computer Vision*, July 1995.

[14] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.

[15] J. L. Crowley. Asynchronous control of orientation and displacement in a robot vehicle. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1277–1282, 1989.

[16] A. J. Davison and N. Kita. 3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 384–391, 2001.

[17] A. J. Davison and N. Kita. Sequential localization and map-building for real-time computer vision and robotics. *Robotics and Autonomous Systems*, 36(4):171–183, 2001.

[18] A. J. Davison and D. W. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, July 2002.

[19] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.

[20] Guilherme N. DeSouza and Avinash C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002.

[21] 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.

[22] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[23] M. Hjek and H. Benoit-Cattin. *Texture Analysis for Magnetic Resonance Imaging*. Med4publishing, 2006.

[24] B. K. P. Horn, Y. Fang, and I. Masaki. Time to contact relative to a planar surface. *IEEE Intelligent Vehicles Symposium*, pages 68–74, June 2007.

[25] Ian Horswill. Specialization of perceptual processes. Technical Report AITR-1511, MIT-AI, 1994.

[26] Stephen D. Jones, Claus S. Andersen, and James L. Crowley. Appearance based processes for visual navigation. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 551–557, 1997.

[27] Niklas Karlsson, Enrico Di Bernard, Jim Ostrowski, Luis Goncalves, Paolo Pirjanian, and Mario E. Munich. The vSLAM algorithm for robust localization and mapping. In *Proceedings of the International Conference on Robotics and Automation*, pages 24–29, 2005.

[28] J. Klippenstein and H. Zhang. Quantitative evaluation of feature extractors for visual SLAM. In *CRV '07: Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, pages 157–164, Washington, DC, USA, 2007. IEEE Computer Society.

[29] Akio Kosaka and Avinash C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP: Image Understanding*, 56(3):271–329, 1992.

[30] S. Kullback. *Information Theory and Statistics*. John Wiley and Sons, Inc., 1959.

[31] F. Launay, A. Ohya, and S. Yuta. Image processing for visual navigation of mobile robot using fluorescent tubes. In *Proceedings of the International Conference on Circuits/Systems, Computers and Communications*, volume C5-1, pages 664–667, July 2001.

[32] F. Launay, A. Ohya, and S. Yuta. A corridors lights based navigation system including path definition using a topologically corrected map for indoor mobile robots. In *Proceedings of the International Conference on Robotics and Automation*, volume 4, pages 3918–3923, 2002.

[33] H. W. Leibowitz, C. S. Rodemer, and J. Dichgans. The independence of dynamic spatial orientation from luminance and refractive error. *Perception & Psychophysics*, 25(2):75–79, February 1979.

[34] Y. Matsumoto, K. Ikeda, M. Inaba, and H. Inoue. Exploration and navigation in corridor environment based on omni-view sequence. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 2, pages 1505–1510, 2000.

[35] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 83–88, 1996.

[36] Y. Matsumoto, K. Sakai, M. Inaba, and H. Inoue. View-based approach to robot navigation. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 545–550, 2000.

[37] M. Meng and A.C. Kak. NEURO-NAV: a neural network based architecture for vision-guided mobile robot navigation using non-metrical models of the environment. In *Proceedings of the IEEE International Conference on Robotics and Automation,*, volume 2, pages 750–757, 1993.

[38] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[39] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.

[40] H. Moravec. Locomotion, vision and intelligence. In Michael Brady and Richard Paul, editors, *Robotics Research: The First International Symposium*, pages 215–244. Cambridge, Massachusetts: The MIT Press, August 1984.

[41] V. Murali and S. T. Birchfield. Autonomous navigation and mapping using monocular low-resolution grayscale vision. In *IEEE Computer Society Workshop on Visual Localization for Mobile Platforms (in association with CVPR), Anchorage, Alaska*, 2008.

[42] R. C. Nelson. *Visual navigation*. PhD thesis, University of Maryland at College Park, College Park, MD, USA, 1988.

[43] R. C. Nelson and J. Aloimonos. Using flow field divergence for obstacle avoidance towards qualitative vision. In *Proceedings of the 2nd International Conference on Computer Vision*, pages 188–196, 1988.

[44] Antonia Papandreou-Suppappola. *Applications in Time-Frequency Signal Processing*. CRC Press, 2002.

[45] F. T. Ramos, J. I. Nieto, and H. F. Durrant-Whyte. Recognising and modelling landmarks to close loops in outdoor SLAM. In *In Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, pages 2036–2041, April 2007.

[46] A. Ranganathan, E. Menegatti, and F. Dellaert. Bayesian inference in the space of topological maps. *IEEE Transactions on Robotics*, pages 92–107, 2006.

[47] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision*, volume 2, pages 1508–1511, October 2005.

[48] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443, May 2006.

[49] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings of the International Conference on Robotics and Automation*, volume 2, 2001.

[50] S. Se, D. Lowe, and J. J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, pages 364–375, 2005.

[51] Shishir Shah and J. K. Aggarwal. Mobile robot navigation and scene modeling using stereo fish-eye lens system. *Machine Vision and Applications*, 10(4):159–173, 1997.

[52] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

[53] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[54] Team ARobAS. Advanced robotics and autonomous systems: 2007 Activity Report. Technical Report RA2007, INRIA, 2007.

[55] C. E. Thorpe. *Vision and Navigation, the Carnegie Mellon NAVLAB.* Kluwer, 1990.

[56] A. Torralba, R. Fergus, and W. T. Freeman. Tiny images. Technical Report MIT-CSAIL-TR-2007-024, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, 2007.

[57] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of the International Conference on Robotics and Automation*, pages 1023–1029, April 2000.