Indoor Mobile Robot Navigation with Continuous Localization

By

Lam Chin Hung

Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF PHILOSOPHY

In
The Department of Mechanical and Automation Engineering
The Chinese University of Hong Kong
Shatin, N.T.
Hong Kong

Sep 20, 1999

Acknowledgments

I am indebted to many persons for their help in the two years' study of my research work. Acknowledgment is hereby given to them and also to the Department of Mechanical and Automation Engineering, which provides me good facility for the research project. Many individuals deserve special mention here for their help and suggestions in preparing thesis.

Special thanks should be given to my supervisor, Prof. Y. H. Liu for his guidance and support in this two years. I would like to thanks Prof Y. S. Xu, Prof. Ronald Chung and Prof. Alex Zelinsky for their comments and suggestions in revising the thesis. I am also grateful for Prof. Ronald Chung who gave me all the knowledge of computer vision.

In addition, I would like to thank Mr. W. K. Fung who has been keeping the B21 robot in a very good condition and gave me tremendous help in experiments. Dr. F. Dornaika and Mr. Y. H. Chow gave me many helpful suggestions in implementing the camera calibration algorithm and other vision algorithms.

Contents

List of Figures

List of Tables

Abstract

Localization is one of the fundamental problems in autonomous mobile robot navigation. High level tasks like path planning and object grasping would fail without precise information about robot's position and orientation.

In this thesis, we study how to localize position and orientation of a mobile robot with respect to the world coordinate frame. This thesis presents a localization approach using natural landmarks in the environment. Specifically, the algorithm uses vertical lines in the environment to determine the robot position and uses vanishing point to determine the robot orientation. A stereo vision system is used to extract features in the environment. To improve the efficiency and eliminate noises, edges only in three prominent 3D orientations are extracted. 3D information of the features are obtained from a pair of stereo images used to construct a local map. A sequence of local maps are maintained and integrated together using odometry information of the robot to form a mature local map in order to improve accuracy of localization. Maps are represented as the projection of all the 3D edges in the environment to a horizontal plane parallel to the floor of the environment. Orientation of the robot is computed using a vision algorithm that relies on vanishing point. A single image is enough for the extraction of vanishing point and estimates the orientation accurately. Robot position is found by matching the global map and the mature local map by an iterated-hill climbing algorithm. A retroactive pose correction method is implemented to continuously localize the

robot's position and orientation. The proposed method is implemented on a RWI B21 robot with a stereo vision system and verified by several experiments. The experiments illustrated that the proposed approach can localize the robot's position and orientation in high accuracy without any artificial landmark and can work well in a corridor environment where people are moving around.

摘要

在移動機械人自動化導航的領域中，機械人能夠自動定位是一個非常基本的問題。如果機械人不能夠確定自己的精確位置，一些比較高層次的工作例如路徑規劃或在工作環境中拿起一些物件等工作就不能須利完成。

在這遍論文中，定位意思是指機械人能夠於一個工作環境的坐標系中認知自己所在的位置和方向。本論文將會詳述一種依靠自然環境來作地標的定位系統。具體來說，此系統利用工作環境中的垂直線來確定自己位置，而水平線則用作方位的確定。一部裝在機械人上的立體視覺系統會選擇和探測一些在環境中有代表性〔有特定方向〕的直線，計算及記錄他們在基於機械人坐標系的三維信息，然後繪製成一個細小局部的地圖，再由數個局部地圖結合成一個較大的地圖。這些地圖的表達方式是二維的，他們是由那些在環境中的三維邊緣線投射到地平面而得出的。最終，機械人的位置是由一種稱爲〝爬山搜尋法〞的算法演算得出。這種方法的思想核心是利用一幅記錄著整個工作環境的地圖和一幅由立體相機拍攝的較小地圖作比較，然後逐一搜尋匹配，得到一個位置，使他們的差別最爲接近，而這個位置就是機械人目前的位置。這個較大的地圖是預先製成的，它記錄著工作環境中的所有細節。而機械人的方向是由影像的沒影點來推算的，單幅影像就能夠準確地計算出機械人相對於工作環境的方向。進一步，我們將先前所提及的方法，與一種有追溯性連續的位置更新算法結合，使機械人能夠連續更新它的位置和方向。

最後，我們利用論文中所提及的新方法，在由RWI公司開發的B21型機械人上做了大量的實驗，實驗結果的証明，這個系統能夠在無人工地標且有行人的走廊上，準確地計算出機械人的位置和方向。

Chapter 1

Introduction

In recent years, there has been increasing interest in investigating the use of autonomous service robot in indoor environment such as hospital, office, factory plants and nuclear reactor plants to perform mundane tasks. In a real indoor dynamic environment, the robot can never follow a predefined path to navigate due to motion uncertainty (e.g. wheel slipping), obstacle uncertainty and so forth. As long as the robot moves in an ambiguous location, it needs to determine its location in order to re-plan the path to goal position. Robot self-localization is the ability of the robot to determine its position and orientation within the working environment. Hence, robot localization is one of the fundamental but important problems of autonomous robot navigation. Odometer readings provide a real time checking of robot position but its error will accumulate over time. As a result, we have to find an alternative that gives more precise results and works efficiently.

There are mainly three kinds of sensors currently used for mobile robot navigation — sonar, laser range finder and camera. Sonar sensors are very effective to detect empty space. However, they can only provide coarse information due to their wide field nature. Crosstalk and specular reflection bedevils the sonar to give accurate result. Laser range finder has higher resolution

but the sensing area is restricted at one level only. Its 2D nature prevents it to sense objects not inside the structure light plane.

Stereo vision provides rich information about the environment that can determine object distance with good precision. Unlike laser range finder, it can sense the whole volume it facing. However, it needs much computational efforts. Depth information is lost due to image projection. There are two major approaches to recover the 3D information. The first one uses visual motion to recover depth. It takes a dense image sequence with a relative motion between images to recover 3D structure. The advantage is the correspondence problem is easy to solve. One drawback of this approach is that it needs a very long image sequence, up to hundreds of frame [26] to accurately recover depth. Whenever there has some moving objects appear in the frames, the depth recovery problem become difficult. Moreover, significant time is needed to process the long sequence of images. The second approach is stereo vision, which recovers depth by a pair of images. In stereo vision, one of the key issues is feature correspondence. Depth is recovered by first establishing feature correspondence between the two images, then depth extraction is only a simple triangulation problem. In contrast, the correspondence of stereo vision is not easy to solve. It was because if we want to recover depth accurately, the baseline between the two cameras should farther apart and hence, for each feature in one image, the search distance for the corresponding feature in another image is large. Much effort is still needed to match features across views although there has the spatial relationship between the two cameras that reduce the search space to 1D along the epipolar line. [44] describes a useful and fast stereo vision system. To overcome the two approaches' disadvantages, we combine the

stereo vision and visual motion together to yield better results. In our implementation, we only establish stereo correspondence for vertical edges. So a simple and fast stereo matching method can be used to reduce the processing time. To have better result, several 3D information obtained by stereo vision is integrated together to minimize the effect of moving object in the scene. The integration is based on the motion predicted by wheel encoders of the robot between stereo image pairs.

Tremendous research efforts have been make to robot localization in last two decades. A good survey on robot localization using different sensors and technique can be found in [1] and [2]. Existing approaches can be grouped into those using artificial landmarks and those without using artificial landmarks.

The first approach [3] refers to localization using artificial landmarks. Usually, these landmarks have specific patterns or forms that can be easily recognized from others features and have known location in the working environment. As long as the landmarks are detected, localization is automatically complete. The advantage is the algorithm is simple and can be executed very fast. However, when there have some temporary obstacles like humans accidentally hide the landmarks, the whole localization algorithm may fail. In [38], the landmark used is a kind of sound source, the robot is localized when it "hear" the sound source, locate it and estimate the robot's position and orientation with respect to that sound source. Tashiro K. et al [40] used signboard system as landmark in which each signboard has 4 LEDs and has known position in the environment. The robot is localized when it determined its relative pose from the landmark. Use of laser reflectors and retro-reflective strips as artificial landmarks are also reported in

[41, 42].

The second approach has more flexibility than the first approach. It mainly has two branches. The first branch uses computer vision for localization. A 3D CAD model of the environment is compared with the image taken at a particular instance to determine robot pose [8, 9]. This approach also called Model-based mobile robot localization [31]. Akio Kosaka and Juiyao Pan [10] use a CAD model to generate an expectation view of the environment, match with the image taken by the robot and use Kalman filter to yield the robot position vector. The advantage is that only a single image is needed to localization the robot. But it cannot handle the dynamic environment. Moreover, localization will only start when the position error is larger than a particular threshold, so resources are used for tracking the position uncertainty and more time is needed since the search distance is longer. In [22, 43] the robot has the ability of avoiding collision of obstacle while localizing itself. [46] is another example of method of collision avoidance using vision. In [39], the localization algorithm uses a stereo vision system to detect and recover the 3D information of corners in the environment to localize the robot. The different between the use of features in [39] and the one we are using is that, we only recover the 3D information of vertical edges only. This kind of edges is very common in all human environments and the number of them is dominating the other kinds of feature and so, they can easily be detected.

The second branch usually collects proximity sensor information together with, for instance, Bayesian techniques [32, 34] or Dempster-Shafer techniques [33] to construct a small local map in real-time fashion. Then the small local map is register with a stored global map to determine robot position and orientation. The

disadvantage of using proximity sensors is that they can only provide limited information. For instances, low resolution and radial error of sonar limits the potential of the localization algorithm. Although laser range finder can provide higher resolution, but it can sense the environment at a particular level only. The approach presented in [5] has proven to be robust with noisy sensors and inaccurate maps. But it still has problems working in dynamic environments. Alan and Adams use a continuous localization approach [6], successfully eliminate accumulated odometry errors with a constant error of five inches. In [7], map is built and updated by Frontier-Based exploration with continuous localization. Zelinsky describes a good mobile robot exploration algorithm [36], which makes the robot works in a partial known or completely unknown environment. The method in [45] use maximum likelihood estimation to correct the error when building map using wheel encoders.

Many methods have been published for mobile robot localization. However, owing to their sensor disadvantages or computational time constrains, most of them are not suitable to use in an environment with human movement. This thesis tries to introduce a system that can improve their shortcomings while keeping the system reliable and efficiency working in such environment.

Our implementation did not rely on any kind of artificial landmark. It uses a simple and fast vision algorithm to extract significant long edges in prominent 3D orientation in the working environment to eliminate noises, especially in a dynamic environment before further process. Advantage is those edges are very common in man-made environments. If some are hidden, there still have many to use. Moreover, this stereo algorithm can filter out most of the dynamic noises in the

image. Since we use vision to extract features, so the sensing area is not restricted to only a particular level like sonar or laser range finder. In fact, our map representation is suitable to integrate their readings to enrich the environmental representation. The stereo correspondence problem of features is simplified to match only the vertical edges to fasten the time for image processing. Vertical edges are used for creating a local map that determines the robot position, while horizontal edges are used to estimate the robot orientation. The robot is continuously localized by a retroactive updated method. By re-localizing the robot often, the pose error is known to be small and a fast-updating method can be employed. The registration between global and local maps includes a hill-climbing search within a predefine search space to yield a set of "offsets" to update the robot current position and orientation.

The system is implemented and tested in a RWI B21 robot in an indoor environment to show its effectiveness. The experiments show that the system is capable of working in an environment with people moving around. Numerical results are presented in chapter 5. As discussed in this chapter, a simple extension of the algorithm can make it has the ability to update the global map. In chapter 2, we first describe the problem and outline the proposed algorithm and assumptions, and then chapter 3 describes the stereo algorithm and selection of useful features for the construction of a mature local map. In chapter 4, the localization system is presented in detail. The implementation on a RWI B21 robot and the experiment results are presented in chapter 5. Finally, we make conclusions in Chapter 6.

Chapter 2

Algorithm Outline

This chapter outlines the vision based robot localization system we developed. The proposed localization algorithm is capable of working in a corridor environment with normal movement of people. First, several assumptions are stated to simplify the problem. Then a brief description of the algorithm is presented.

2.1 Assumptions

The problem we are addressing is how to localize the position and orientation of a mobile robot in the working environment. To clearly defined and simplify the problem, we assume,

- The robot is working in indoor environment where people are walking around. In other words, the robot is working in a dynamic indoor environment.
- A world model of the working environment is given in advance. The model provides all the knowledge for the robot to localize within the working

environment.

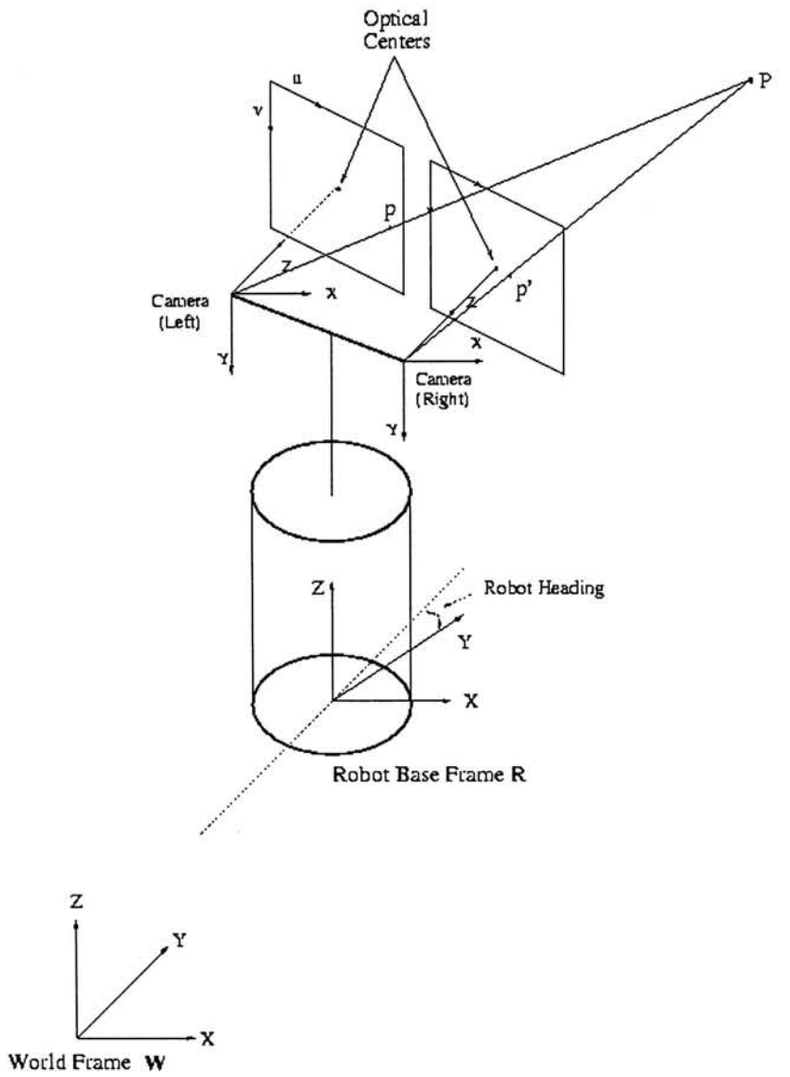


Figure 1. Coordinate System.

2.2 Robot Localization

Figure 1 shows the world and robot coordinate systems used in this thesis. The robot has a mobile frame **R** is navigating in the environment with the fixed

world frame **W**. Since we assume that the floor of the environment is flat enough that the roll and pitch angle of the robot can be neglected, so the z-axes of the robot and world frames are both vertical and parallel together. But their x and y axis are not parallel in general as the robot's orientation changes. The robot-heading angle or the robot orientation angle is the angle between the y-axis of the mobile frame **R** and the y-axis of the fixed frame **W**. Robot position is the relative translation of the robot with respect to the world frame **W**. A stereo vision system is mounted on top of the robot. The axis of the camera frame is z along the principle axis and x and y parallel to the image plane; x along the horizontal (+ve right) and y vertical (+ve down). The cameras have a fixed relation with the robot base frame **R** and this relation can be found by eye/wheel calibration. Cameras are well calibrated in which all their intrinsic and extrinsic parameters are known.

The transformation from the world frame to the camera frame is given by:

$$T_{WP} = T_{CP}\, T_{RC}\, T_{WR}$$

T_{CP} is the perspective projection from the camera frame to image plane and is given by,

$$\begin{bmatrix} \alpha_u f & 0 & u_0 & 0 \\ 0 & \alpha_v f & v_0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where α_u and α_v are the pixels per unit length in u and v direction, u_0 and v_0

are the coordinates of the optical center of the camera (pixels) and f is the focal length of camera in mm. All these parameters can be obtained from camera calibration (Appendix A1), and so, it is a constant matrix.

T_{RC} is the transformation matrix from the robot frame to the camera frame and is a constant matrix determined by eye/wheel calibration (Appendix A3).

T_{WR} is the homogeneous coordinate transformation matrix from the world frame to the robot frame. In contrast, is not a constant matrix, it changes from time to time as the robot moves. As the roll and pitch angle of the robot can be neglected therefore,

$$T_{WR} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & -x \\ -\sin\theta & \cos\theta & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where θ represents the orientation angle of the robot, x and y are the position of the robot with respect to world coordinate frame **W**.

Hence, the goal of the localization is to estimate the transformation matrix T_{WR} or more specifically, the position and orientation of the robot (x, y, θ) with respected to world coordinate frame **W**.

2.3 Algorithm Outline

The robot navigates in a corridor environment with a stereo vision system. The stereo system extracts edges in 3D prominent orientation with length longer than a predefined threshold in the image to minimize the effects of noise. 3D prominent orientation means the vertical edges, and horizontal edges that perpendicular to each other in the environment. With this feature selection, most of the unwanted edges are filtered out and the remaining edges are used for localization. Figure 2 illustrates the use of these features. Vertical edges in the images are used to extract their 3D information for the construction of local map by a simple stereo algorithm. Horizontal edges that parallel or nearly parallel with the optical axis of the cameras are used to determine the vanishing point. Hence their orientation appear in the image is shown in the middle of figure 2 (second kind of horizontal edges), the dot near the center is the vanishing point. Only a single image is enough to determine the location of the vanishing point and this point is used to estimate the orientation of the robot. Hence, we can use the same stereo image pair for both constructing a local map and estimating the orientation of the robot.

Both global and local maps are represented as 2D planes that give a top view of the environment. Global map gives a top view of the whole environment while local map gives a top view of its immediate environment. Hence, 3D vertical edges detected by the stereo vision system are projected on a 2D local map as points. As the robot moves, it keeps taking pictures so different local maps are obtained.

Several local maps are maintained in memory and combined together to form a *mature local map*. A local map is said to be *mature* if it contain enough information for registration provided that the accumulated positional error is small. In this thesis, local map refers to the map construct by 3D information of vertical edges in a single pair of stereo images. A mature local map is the integration of several local maps.

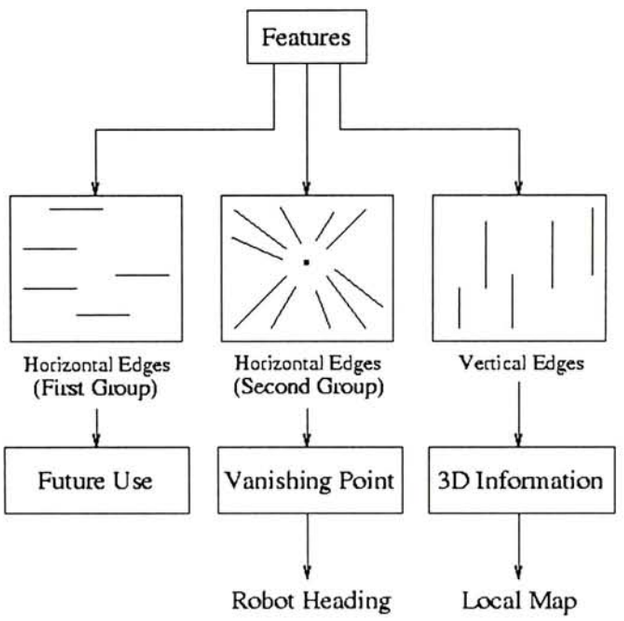


Figure 2. Features selection and their functions.

When the mature local map is ready, a registration process is carried out. It involves a search in a predefined search space gives position that maximizing the match score between the mature local map and the global map. The registration process involves an iterated hill-climbing search within a particular configuration space. It climbs from an initial position feedback by wheel encoders until reaching a maximum score. The correlation based matching score measures the "similarity"

between the mature local map and the global map. As the vanishing point determined the robot orientation, the three-dimensional search in the configuration space becomes a two-dimensional search at a particular level of cross section in the configuration space (Figure 3), that is, only search for robot's x and y positions with respect to the world frame.

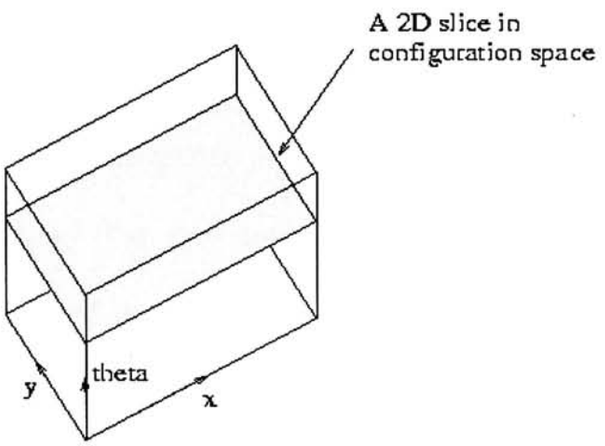


Figure 3. The two-dimensional search space.

Once the position and orientation are updated, the old coordinate is adjusted to follow the new system by a retroactively update approach. This makes the robot to re-localize while navigating in the environment without stop. Localization process will repeat continuously making small correction to further restrict the 2D search space as small as possible.

To summarize, the localization system has the following properties:

- It works in a normal indoor environment. Using the fast line detector [11], most of the edges not useful for localization are filtered out, including those contributed by human.

- The localization process is repeated regularly as the robot navigating. Duration is the runtime parameter of the system. This guarantee the search space is small and fast localization algorithm can be used.
- Fast processing time. Since only vertical lines are needed to establish the stereo correspondence, processing time for stereo matching is reduced. In addition, the extracted vanishing point can be used to obtain robot heading by simple calculation. The position search is only a 2D search in the configuration space with restricted area.
- The algorithm do not has the ability to update or build the global map. However, a simple modification can make it has the ability to locate any new edge in the environment (Chapter 5).

Chapter 3

Global and Local Maps

We use stereo vision to construct the local map. Stereo vision can recover depth information. If the baseline between two cameras is large enough and the two images have significant overlaps, stereo vision can give accurate result. However, stereo vision is limited by its fundamental problem – the *correspondence problem.* Although the stereo algorithm developed recently improves the computational complexity, but it is unavoidably to take a significant processing time. Instead of processing all the observable features, we can pick up the most useful information from huge amount of raw data and discard the others. This saves the time for processing the useless data. Moreover, accuracy is improved by a kind of local map integration that uses wheel encoder information.

In our representation, local map is a 2D map that gives a top view of the robot immediately environment (Refers to the coordinate system in figure 1). Since only 3D information of vertical edges are added into the local map, so they are appear as "dots" in the map. The only information we needed is the depth or the distance of the edges from the robot and their x positions from the robot. Their y positions are useless in this map representation. Once depth information is recovered, x position is automatically obtained.

Figure 4(a)

Figure 4(b)

Figure 4(c)

Figure 4(d)

Figure 4(e)

Figure 4. (a) A corridor scene with a human standing in the middle. (b) Edges extracted in the scene. Edges extracted after filtering (c), (d) and (e). Note that most of the edges contributed by human are effectively filtered out.

3.1 Feature Selection

Since we use vision as sensor to detect the environment, it can provide rich information. However, some of the information may not be useful for the localization process. In fact, the exceeding information including noises may cause the unnecessary workload and disturbance to the system. To avoid the unnecessary process and increase the efficiency, we select the most useful features in the environment and try to minimize the noises. Only line segments in three prominent 3D orientations (one vertical and two horizontal perpendicular to each other) are extracted using a fast line detector [11]. The detector uses a priori knowledge of the orientation of edges to be extracted. Figure 4 shows edges extracted by the line detector. The advantage of choosing these features is that they are very common in all human environments. For example, doors, bookcases, desks, corridors, even temporary storage store boxes are all particular parallel lines in world coordinate system. Moreover, image processing time is reduced by ignoring the useless and noisy edges. With this selection of feature, we can see in figure 4 that most of the dynamic noises are filtered out before further process.

Among these three groups of line segments (One vertical and two horizontal perpendicular to each other), we only pick up two groups: the vertical ones (Figure 4c) and the set that nearly parallel with the optical center of the cameras in the environment. So they are appeared in the image as shown in figure 4d. The remaining group is leave for future use. As mentioned earlier, localization only uses the first group of line segments to recover depth information for the

construction of local map, so the correspondence problem can be simplified. Vanishing point can easily be extracted by locating the intersection of second group of segments (Figure 4d), and is used to estimate the orientation of the robot. Hence we only need to establish correspondence relationship between the vertical lines so that the computation time is significantly reduced. It should be noted that the vanishing point can be computed in real time.

3.2 Line Correspondence

After we filtered the useless edges, we can further process the image data. Since the features for stereo matching are simple and small in number, we can employ a basic stereo algorithm to recover the 3D information.

Image data are first smoothed by convoluting a gaussian profile prior to the gradient calculation. Edge pixels are extracted by applying the canny edge detector [47]. Line segments are then fitted into the edgels by orthogonal regression. In order to minimize the effect of noise, those edges shorter than a predefined threshold are discarded. One should notice that edges not in the desired orientation are filtered out before the line fitting process.

Stereo correspondence is established for vertical lines only. Based on the epipolar geometry (Appendix A4), the two end points of a segment in left image produce two epipolar lines, which enclose a region in right image. So the corresponding segment should find in that region. This simplifies in terms of computational efficiency, the subsequent stereo matching problem. Then the

matchability between lines is determined by three factors: similar orientation, similar contrast value and same contrast sign. The first two factors are thersholded and the third factor is a necessary condition. Every segment in the left image is forced to match exactly one segment in the right. Then depth is recovered by simple triangulation. All the features further than 4 meters from the camera are discarded to ensure the accuracy of position estimation. To guarantee significant overlaps of the stereo pair of images, the robot avoid taking image of an object near that 1-meter.

The stereo algorithm may fail to find the correspondence and mismatch features across the left and right images when there have several similar edges that locate closed together in the environment. This result in producing some wrong edges in the local map. However, those kind of wrong edges generally have strange 3D information that is outside the range for our feature selection. Moreover, in case those edges are added into the local map, the quality of the localization result will not degrade since other correct information are dominating.

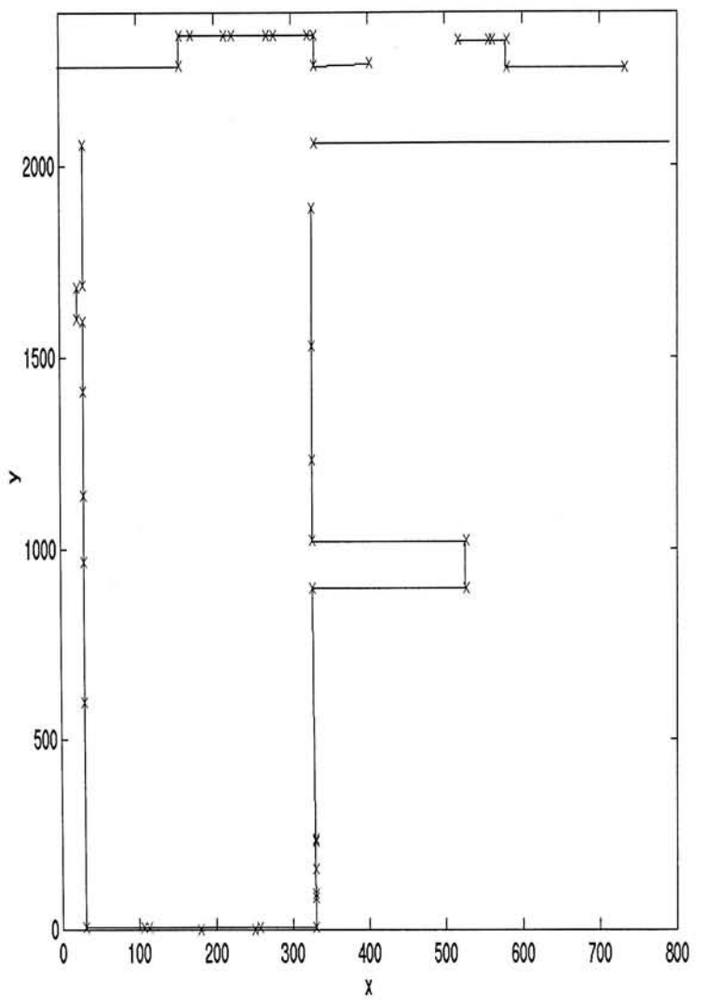


Figure 5. Global map at the fourth floor of MMW building.

3.3 Map Representation

After we obtained the 3D information of the edges, this section describes how we use these information to model the 3D global and local environment into a 2D map. Maps are represented as the projection of edges into x-y plane in the environment, so all vertical edges appear as points on the map. All other edges

(two groups of horizontal edges) are represented as straight lines on the map. As these straight lines representing walls in the environment, so they provide a physical constraint that the robot cannot pass through these lines in the map. The entire map is divided into a discrete space as a number of 2D grids. Each grid contains a value range from 0 to 10 representing the probability that the corresponding location is being occupied by an edge; the larger the value, the more likely an edge is at that grid. The probability of a grid is modeled by Gaussian distribution with the exact edge location (either by measuring the environment when constructing the global map or by results from stereo vision when constructing the local map) as the center. Therefore the farther from the center, the less probability an edge is at that location.

3.3.1 Global map

The global map is the grid representation of the entire environment in which the robot navigating. The location of edges in the map is obtained by the building draft of the environment and by direct measurement. The map is produced in advance; it contains all the significant edge information in the environment. Figure 5 shows a global map of a corridor environment. Crosses are the vertical edges in the environment and lines are the horizontal edges representing walls in the environment. Equation (1) gives a particular grid score of global map at position (x_G, y_G).

$$G_G = (x_G, y_G, v_{Gjx}, v_{Gjy}) = 10 \cdot \exp(-\frac{(x_G, - v_{Gjx})^2 + (y_G - v_{Gjy})^2}{2\sigma^2}) \quad \ldots(1)$$

Where (V_{Gjx}, V_{Gjy}) is the coordinates of jth global edge. σ is a constant defined as follow: When the map is crowded with edges, we choose a small σ, otherwise set it large enough to make the search in the right direction.

The construction of global map is based on the building draft and direct measurement of the environment. The locations of all the significant vertical edges in the environment are measured, hence the ith vertical edge in the global environment will has a Gaussian score with center at (v_{gix}, v_{giy}) as stated in equation (1). Horizontal edges are also appeared in global map. However, we do not add all the horizontal edges found in the environment in the map but we will add those representing walls in the environment. Obviously, this kind of edges in the global map provides a physical constraint that the robot cannot pass through them.

3.3.2 Local Map

Local map represents the immediate environment of the robot as probability grids. It is constructed according to the 3D information of the detected edges, which only vertical edges are existed. The local map gives position (x_L, y_L) of vertical edges in the local environment with respected to the robot frame, then cell score in the local map G_L is determined by equation (2),

$$G_L=(x_L, y_L, v_{Lix}, v_{Liy})=10\cdot\exp(-\frac{(x_L,-v_{Lix})^2+(y_L-v_{Liy})^2}{2\sigma^2}) \quad \ldots(2)$$

where (V_{Lix}, V_{Liy}) is the location of ith local edge, σ is a constant same as equation (1).

In perfect case, a local map is a subset of global map, but in real situation, owing to the error in sensing the environment and the disturbance of obstacles, there are some missing edges or additional edges appear in the local map, but this will not degrade the localization process. Figure 6 shows a local map in grid representation in the lower center of the corridor in figure 5. White pixels show the locations have the probability that edges are presented. The brighter the pixel, the higher the probability an edge is at that location. We can see as compared with figure 6, figure 5 has some missing edges.

Figure 6. A grid representation of local map.

3.4 Integration of Multiple Local 2D Maps

As mentioned earlier, we will integrate several local maps together to form a bigger map called a "mature local map" to register with the global map. This section describes how we merge the local maps together. The merging process uses the information from odometer readings. Robot odometry error usually accumulates over time, but within a short travel, odometer still is reliable. We make use of the odometer reading to estimate the motion between robot different locations in a short distance. Infer motion using long sequence of images is not feasible here since it generally needs long time to process and require relatively long motion distance. Local map here is a 2D map obtained from the projection of the 3D vertical edges from stereo. A local map model is constructed by integrating two or more local maps.

Figure 7 shows the integration process. If there is no local map model $\boldsymbol{M}$ in memory, integrate two successive local maps to form a new one. If yes, combine the local map model and the new local map as follow: Suppose that at a particular instance t_i, we have a local map model $\boldsymbol{M_{i-1}}$ obtained at t_{i-1} and a local map $\boldsymbol{V_i}$ at t_i to be integrated with $\boldsymbol{M_{i-1}}$. From the wheel encoder reading we can estimate the motion from t_{i-1} to t_i, then define a transformation matrix according to that motion, and transform those features in $\boldsymbol{M_{i-1}}$ to the coordinate system as $\boldsymbol{V_i}$. After establishing their correspondence, we can integrate them and form a new local map model $\boldsymbol{M_i}$ at t_i. The integration process should be finished within the time interval between constructing two local maps. The number of local map for integration is

limited by the positional uncertainty. More positional error will be accumulated when the sequence of images is long. In our implementation, three pairs of stereo images are taken, with approximately 0.5m of travel, so three local maps are integrated for the construction of the mature local map.

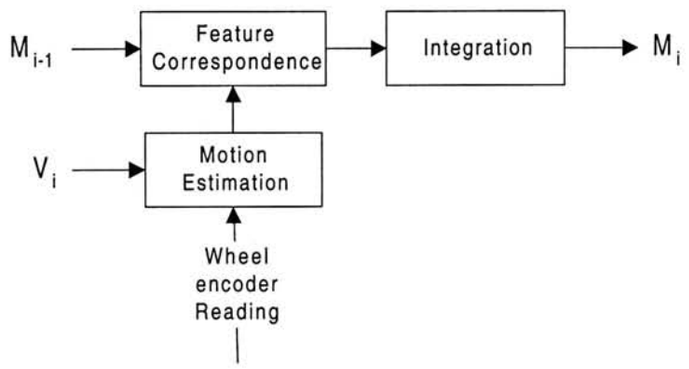


Figure 7. Local 2D maps integration.

The feature correspondence problem to merge edges did not keep track of how far a feature in one frame moved in others. In fact, the motion cue is provided by the robot's encoder readings. In figure 8, $\boldsymbol{R}$ and $\boldsymbol{T}$ are the rotation and translation estimated by wheel encoder. Each edge $\boldsymbol{P_i}$ in the new local map has a 2D position with respect to its frame $\boldsymbol{V_i}$. We define a small uncertainty region around the location of each edge in frame $\boldsymbol{V_i}$, so edge $\boldsymbol{P_{i-1}}$ in the frame $\boldsymbol{M_{i-1}}$ falls into that region is considered as "feature pair" ($\boldsymbol{P_{i-1}}$ and $\boldsymbol{P_i}$) for merge. The dotted ellipse in the figure illustrates the uncertainty region. After established the correspondence, $\boldsymbol{P}$ is obtained by averaging the feature positions and a new local map mode $\boldsymbol{M_i}$ is obtained. Any edges in the frame $\boldsymbol{V_i}$ (or $\boldsymbol{M_{i-1}}$) that cannot form "feature pair" are considered as new (or old) features and added to $\boldsymbol{M_i}$ as well in this stage. A global

check of all the features in ***M*** (the final model) is carried out after all the integration procedures finished and right before the construction of mature local map to remove those features that are not appear in successive frames. This minimizes the effect of noisy edges as well as some temporally edges in the scene. To summarize this section, we have developed a simple, fast and effective motion stereo scheme to construct a 2D local map model. Stereo vision and motion analyze are combined together to make the model construction more accurate. Such a fast image processing technique benefits the localization process.

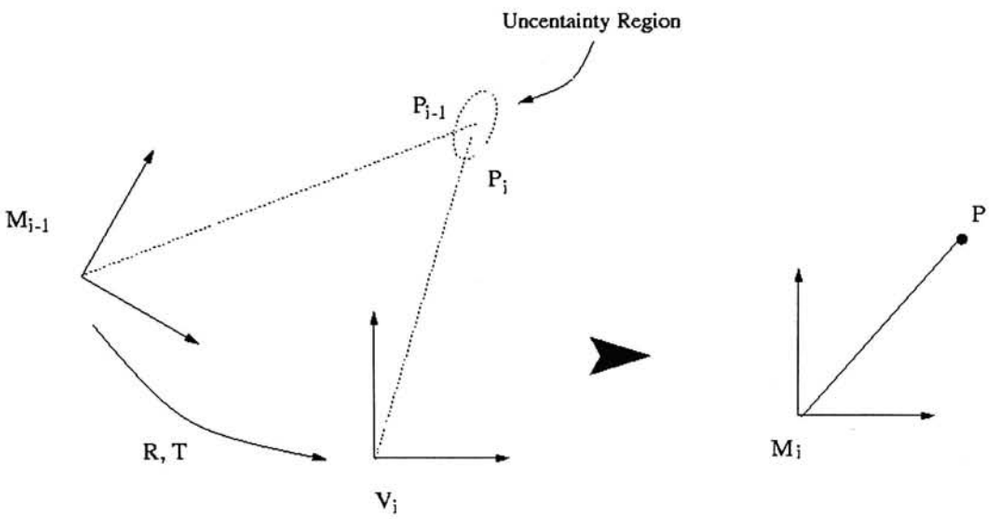


Figure 8. Merging edges from different local maps

Chapter 4

Localization Algorithm

The previous chapter describes how to construct and represent maps, this section mention how to use the mature local map and the global map to compute the position and orientation of the robot. Figure 9 shows an overview of the localization algorithm. A stereo camera system is used to detect features in the environment, those features include the three groups of edges we discussed before. Then the algorithm splits into two parts, the first part estimates the robot orientation while the second part evaluates the robot position through a map registration process.

Horizontal edges are used to locate the vanishing point. Then the robot orientation is estimated based on this vanishing point. 3D information of the vertical edges is extracted using a simply stereo algorithm and add to the local map. Several local maps are merged together to form a bigger map called "*mature local map*". Then a registration process is carried out to register the mature local map with the global map together with the odometer readings to have the robot position.

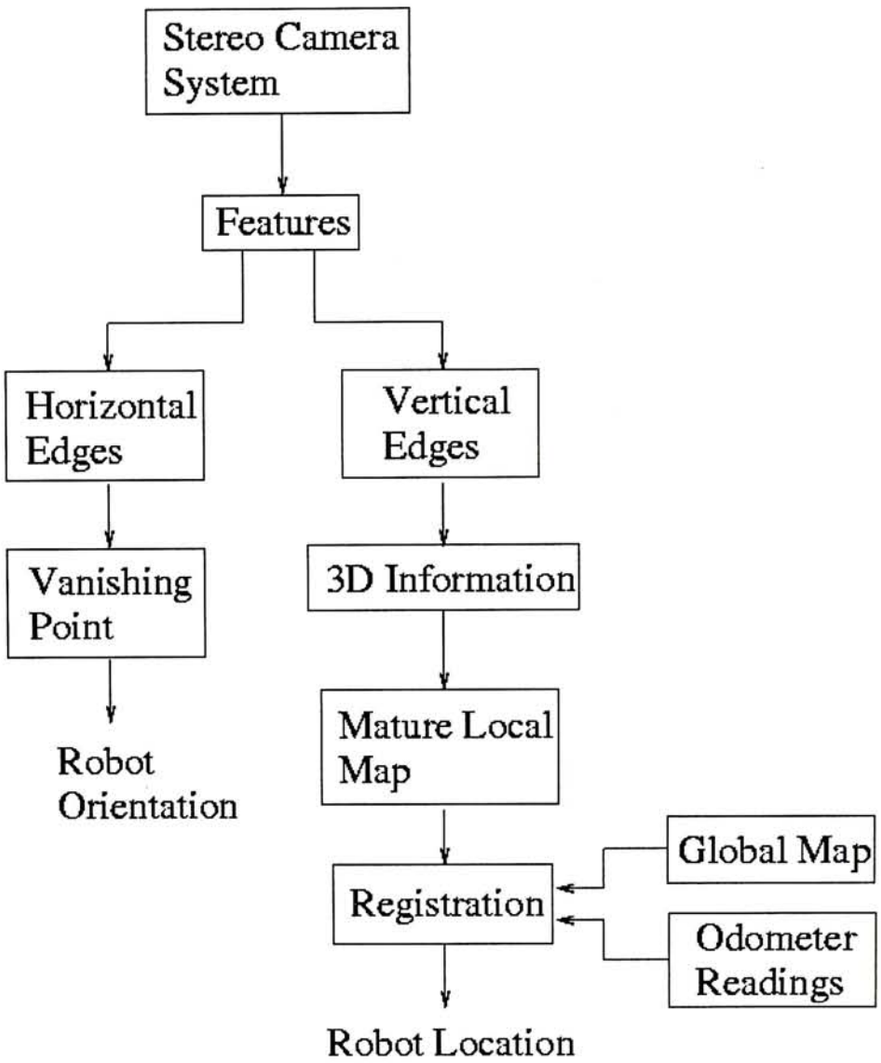


Figure 9. Overview of the localization algorithm.

4.1 Robot Orientation

As mentioned before, vanishing point (u_{vp}, v_{vp}) can easily be extracted by using a single image. To have better precision, robot roll and heading angle should be computed with a vanishing point that varies most with them. In our

implementation, we use the vanishing point near the center of the left camera image. Let θ be the orientation angle of robot, therefore θ is the arctangent of,

$$-(u_0 - u_{vp})/(\alpha_u f) \qquad \ldots(3)$$

Where u_0 and v_0 are the coordinates of the optical center of the camera (pixels), α_u and α_v are the pixels per unit length in u and v direction (Appendix A1). Although the robot roll angle is not used in localization algorithm, but it still can be found for future use,

$$Roll\ Angle = \arctan\{-(v_0 - v_{vp})/(\alpha_v f)\} \qquad \ldots(4)$$

Hence, the orientation of the robot can be obtained using (3) and the search space will be reduced to only two-dimensional. Searching neighborhoods per iteration are reduced to 8 ($3^2 - 1$) pose cells and this will fasten the searching time. The accuracy of the estimation of the orientation angle is depended on the accuracy of the vanishing point extraction and the precision of camera calibration. Chapter 5 illustrates the quality of the estimation of the robot heading.

4.2 Robot Position

The robot position is estimated by a registration process that measures the similarity between the mature local map and the global map. The registration process involves a search in the space of offset in translation that uses the current

pose of the robot from the wheel encoder to start the search until reached the state that the matching score is maximum. Then the solution state is directly applied to the robot odometry and became the current robot pose.

4.2.1 Match Function

The match function is defined as the feature similarity between the mature local map and the global map. Suppose the mature local map size is M x N, one well know similarity measurement is the sum of square error:

$$E(i,j)=\sum_{m}^{M}\sum_{n}^{N}[\sum G_L(m,n,v_{Lx},v_{Ly})-\sum G_G(i+m+o_x,j+n+o_y,v_{Gx}^{'},v_{Gy}^{'})]^2 \quad ...(5)$$

Where G_L is the local cell score at location (m, n) and G_G is the global cell score at $(i+m, j+n)$. $(V_{Gx}^{'}, V_{Gy}^{'})$ is the global edge location express as the coordinate system of the mature local map. o_x and o_y are the initial offsets between the mature local map and the global map. This measurement $E(i, j)$ gives the dissimilarity between two map, so our goal is to find i and j that minimize E. Equation (5) can be further simplified by expanding and eliminating constant terms -- is equivalent to maximizing,

$$E^{'}(i,j)=\sum_{m}^{M}\sum_{n}^{N}[\sum G_L(m,n,v_{Lx},v_{Ly})\cdot\sum G_G(i+m+o_x,j+n+o_y,v_{Gx}^{'},v_{Gy}^{'})] \quad ...(6)$$

Hence the match function defined is the *product match* between the global and the mature local map cell scores and the matching score is obtained by aligning the mature local map with the global map then multiply the cell score between them, summation over the area of the mature local map.

4.2.2 Search Algorithm

The search space, with the robot orientation determined, become an area of a two-dimensional space with x and y axis. Where x and y are translation offsets from the current robot pose. As the robot performs self-localization often, the search space is restricted to a small area. Our implementation defines the space as all the possible poses within ± 30cm in translations of robot's current pose. Current pose can be predicted by wheel encoder readings.

The search method uses is iterated hill-climbing search. It uses an initial pose as the search center and starts to climb. The search divides the search space into pose cells by an initial resolution. Then computes the match scores between the mature local map and the global map at the current pose and the 8 ($3^2 - 1$) immediately neighboring pose cells. If a neighbor cell has a higher score, then repeat the search using that cell as center. If no better score, the search repeats with a higher resolution. The search stops when the predefined resolution is reached. Our implementation defines the initial resolution as 8cm in translations, while the final resolution is 1cm.

If the robot is put in an unknown location initially, it has to find out where it

currently is from the knowledge of global map. In this case, the search space will becomes 3D (with x, y, and θ axis) and a global search is performed to look for every possible position and orientation. This kind of search is very time consuming. In our case, it takes about 2.6 minus to locate itself in a 5.4m by 15m area.

4.3 Continuous Localization with Retroactive Pose Update

The idea of continuous localization is first described in [6] for continuously re-localizing the robot using evidence grids. A Bayesian map updated technique is used to construct the global map and local map using both sonar and laser range finder readings. In our case, the global map is constructed in advance and we use stereo vision to detect the 3D position of vertical edges to construct the mature local map.

Our goal is to make the robot navigates smoothly along a corridor. However, the localization process needs significant time for processing and this results in the robot halting until all vision and localization processes have finished. What we really need is the robot to continue to navigate while self-localizing. That is, offsets of translation from localization module are retroactively combined with odometer readings to update robot's current location. This is core idea of the proposed retroactive updating method.

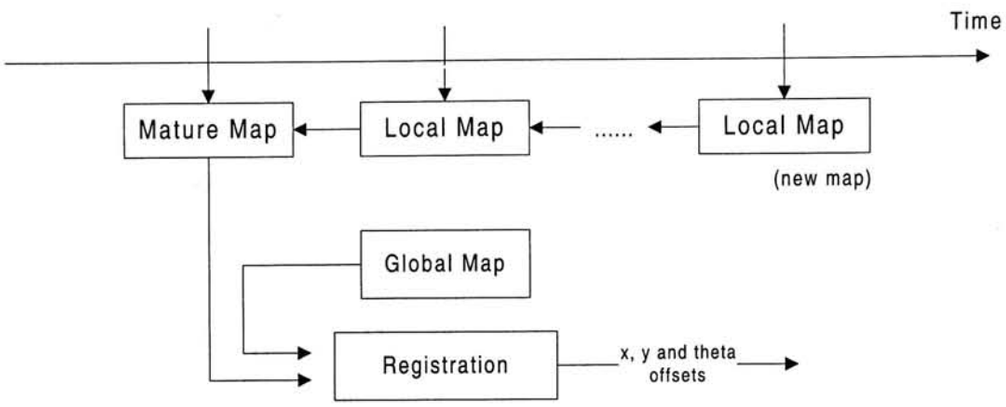


Figure 10. One localization cycle.

Our implementation is a slight modification of the method reported in [12]. It contains two parts: Localization process and retroactive pose update process. Figure 10 shows one cycle of the localization process. Local maps are constructed as the robot navigates, several are maintained in memory. They are integrated with a new local map using encoder information as described in section 3.4. After sufficient local maps are integrated together to form a mature local map, the registration process is carried out to update robot position in the environment. In our implementation, three local maps are stored in memory and merged together. A new localization process begins immediate after the previous the old process has finished, so the whole process is called continuous localization.

Time is needed for image processing and searching pose. Instead of making the robot stop to wait until localization result comes out, the robot continues to travel and uses the retroactive update method. Figure 11 shows the retroactive pose updating process.

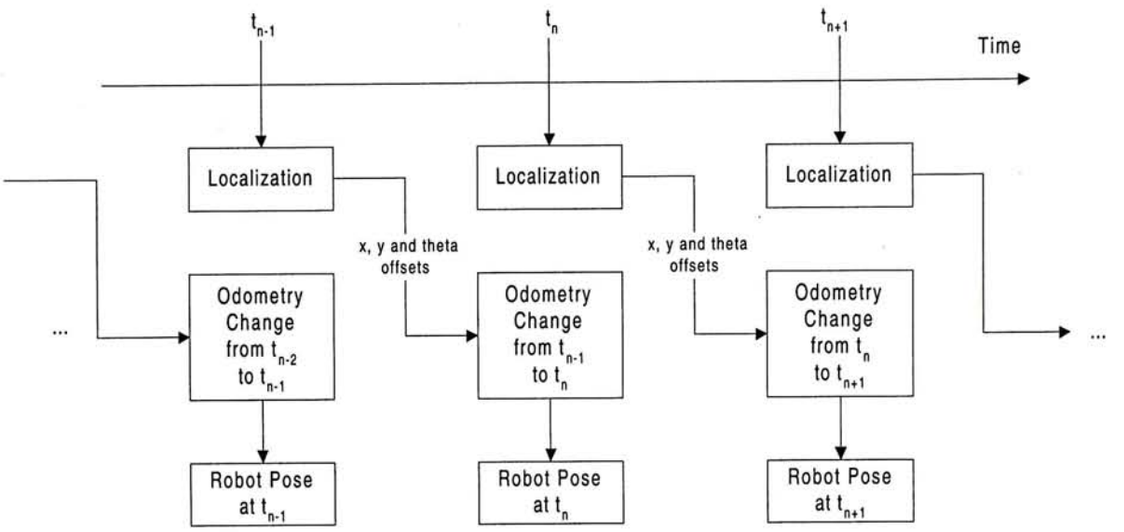


Figure 11. Continuous localization with retroactive pose update.

Suppose the localization process start at time t_{n-1} is finished at t_n. The x, y, and orientation offsets results are the pose correction at time t_{n-1}. They combined with the odometry changes from t_{n-1} to t_n gives robot's current pose. This pose also contains some error due to the odometry error from t_{n-1} to t_n, but is corrected at time t_{n+1}. Moreover, this odometry error will not accumulate as the previous error is corrected. With this retroactive pose correction, the robot can navigate without stopping for self-localization.

Chapter 5

Implementation and Experiments

Several experiments were performed to determine the effectiveness of the proposed localization algorithm. The algorithm is implemented in C language in a RWI B21 robot (Figure 12). It is controlled by an onboard Pentium processor running Linux. Although there have other sensors to use, including sonar and infrared sensors, we only use a stereo vision system in experiments. All the experiments assume that the cameras are well calibrated. That is, the spatial relationship between cameras and the relation from the camera frame to the robot frame are known. Calibration information can be found in Appendix. We will show the results of experiment in dynamic environments with moving objects.

Figure 12. The RWI B21 Robot.

5.1 Computing Robot Orientation

Robot orientation is one of the three localization parameters we have to determine. It is a very important parameter since a very little orientation error may result in a very large translation error after long distance travel. The accuracy of the computation of robot orientation is depended on the accuracy of camera calibration and the extraction of position of the vanishing point. Vanishing point can be located easily and accurately in a corridor environment.

In these experiments, the robot is commanded to rotate in increment of 5 degree start from –10 degree to 10 degree. We will add some disturbances in the environment to check the quality of the estimation. Each result is obtained from a single image from the left camera. Table 1 shows the results of the estimation of robot orientation in a clear corridor while table 2 is the results with human walking along the corridor.

Figure 13. A corridor without moving obstacle (Corridor 1).

Angle (Degree)	Corridor 1		Corridor 2	
	Estimated (Degree)	ABS Error (Degree)	Estimated (Degree)	ABS Error (Degree)
10	10.776933	0.776933	10.138600	0.138600
5	4.874607	0.125393	4.868680	0.131320
0	0.568738	0.568738	0.580160	0.580160
-5	-4.802665*	0.197335	-5.165905	0.165900
-10	-9.511093	0.488907	-10.289980*	0.289980

Table 1. Estimation of heading angle in a clear corridor. Data marked with (*) is the result from figures 13 and 14 respectively.

Figure 14. A scene in corridor 2.

Figure 15. Same corridor as figure 13 with a human walking.

Angle (Degree)	Corridor 1	
	Estimated	ABS Error
10	11.100676	1.100676
5	5.352859*	0.352859
0	0.763003	0.763003
-5	-4.521051	0.478949
-10	-9.807420	0.192580

Table 2. Estimation of heading angle in a corridor with human movement.

Noteworthily, the computation of orientation angles still works in a dynamic environment. Table 3 illustrates the results from figure 16. Figure 16 is a classical dynamic environment. A bookshelf, a chair and a moving human that does not appear in the global map are added in the environment.

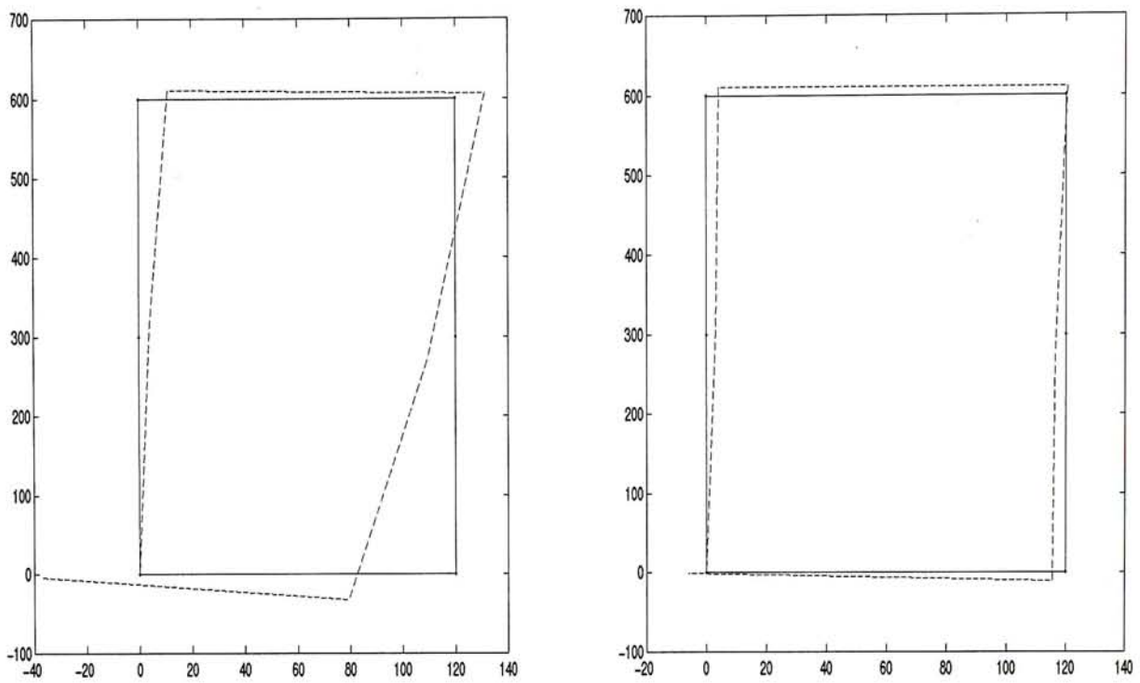
The results show that the errors are not depended on how the environment changes; the location of vanishing point determines the accuracy. Only use few edges can locate the vanishing accurately. Among all the estimation, the maximum error is 1.364582 degree and the average error is 0.46107955 degree.

Figure 16. A new bookshelf, chair and human appear in the environment.

Angle (Degree)	Corridor 1	
	Estimated	ABS Error
10	11.364582	1.364582
5	5.317281	0.317281
0	0.601916	0.601916
-5	-5.548377	0.548377
-10	-10.038102*	0.038102

Table 3. Estimation of heading angle in a dynamic corridor.

To illustrate the usefulness of the robot orientation computation, the robot is commanded to follow a rectangular trajectory in a corridor with total travel distance 14.5 meters and with presence of human. Only orientation correction is performed in this experiment. The broken line in figure 17(a) is the actual path which the robot is using odometry correction. The odometry error is small at the beginning, then accumulate as the robot travels. As the robot orientation error accumulates, the final translation error is large after a long distance travel. Figure 17(b) shows the path of the robot with correction by vision; it keeps the robot to move parallel to the corridor. Correction is carried out every time the detected heading error is larger than 2 degree. During the experiment, a human is wandering around to disturb the robot. Although there still has error in position, however the result shows that the robot orientation angle correction effectively eliminates the accumulated odometry errors and the final translation error is small. Moreover, the moving human does not affect the accuracy of results.



(a) (b)

Figure 17. (a) Path of the robot using odometer. (b) With orientation correction.

We can notice that the estimation of robot orientation is not affected by people moving around nor affected by the dynamical environment. However, the estimation will degrade if large portion of the field of view is occluded by the people as shown in figure 18(a), as the significant 3D edges cannot be detected. Figure 18(b) is another example showing not enough significant 3D edges for the determination of robot orientation. In this case, since we still have vertical edges in the mature local map, the robot orientation can be found together with the robot position. That is, all of the three localization unknowns (x, y, θ) are determined by the iterated hill-climbing search with a 3D search space (With x, y, and θ axis). Since the search space become 3D, the search will take more time to finish as the immediate neighborhood pose cells become $27(3^3 - 1)$ instead of 8. (Figure 19).

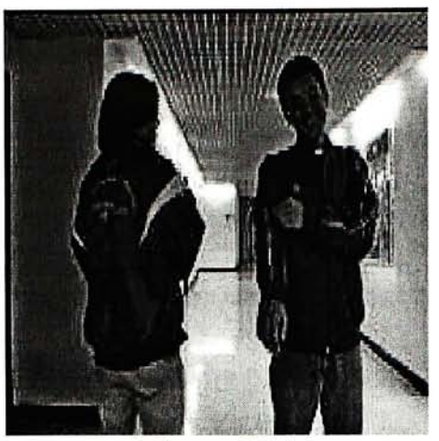


Figure 18 (a)

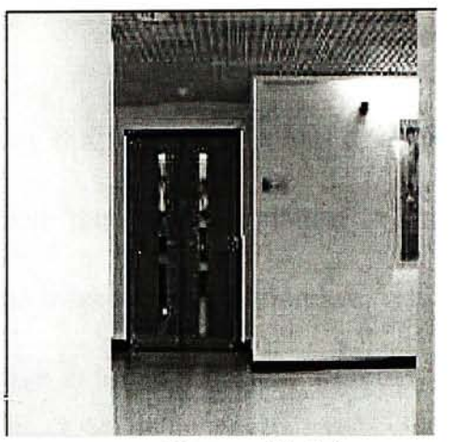
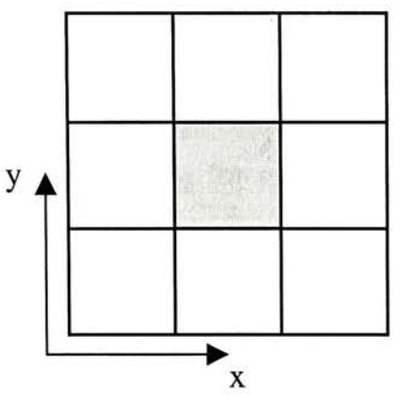
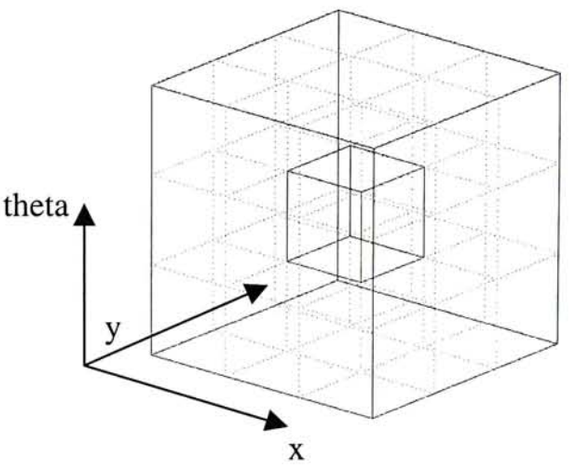


Figure 18 (b)

Figure 18. An example of scene that makes the orientation accuracy degrades. (a) Significant 3D edges are occluded by human. (b) Significant edges cannot be detected.



A 2D search space with 8 neighborhoods



A 3D search space with 27 neighborhoods

Figure 19. Relaxation of search space.

5.2 Robot Position by Map Registration

In the previous experiment, we have shown that with the robot orientation angle corrected, translation error is small when the travel distance is small. However, translation error still accumulates as the travel distance increase. Hence, in this experiment, the robot is commanded to move in a hallway with total distance of approximately 23 meters while localizing 12 times in the environment and evaluating all the localization parameters, i.e. the position and orientation of the robot with respected to the world coordinate frame.

The robot is commanded to move with average speed of 15 cm/s in the hallway at 4^{th} floor of MMW building with human moving around using the tele-operation interface we developed (Appendix A.5). The black line in figure 20 is the planed path, dotted lines are 1 meter on each side. For each localization, three local maps are merged together to form a mature local map and register with the global map. Time for processing each pair of stereo images is approximately 3 seconds, so it takes nearly 9 seconds to merge the three local maps together (each mature local map contains the environmental information gathered in the most recent 52cm of travel). Time for the iterated hill-search is another 2 seconds, hence the localization cycle repeats approximately every 12 seconds. Crosses in the figure show the actual path of the robot in which localization are finished and robot position are measured at these points. The odometry error we define is the odometer readings minus the robot actual position, and the localization error is the result from map registration minus the actual position.

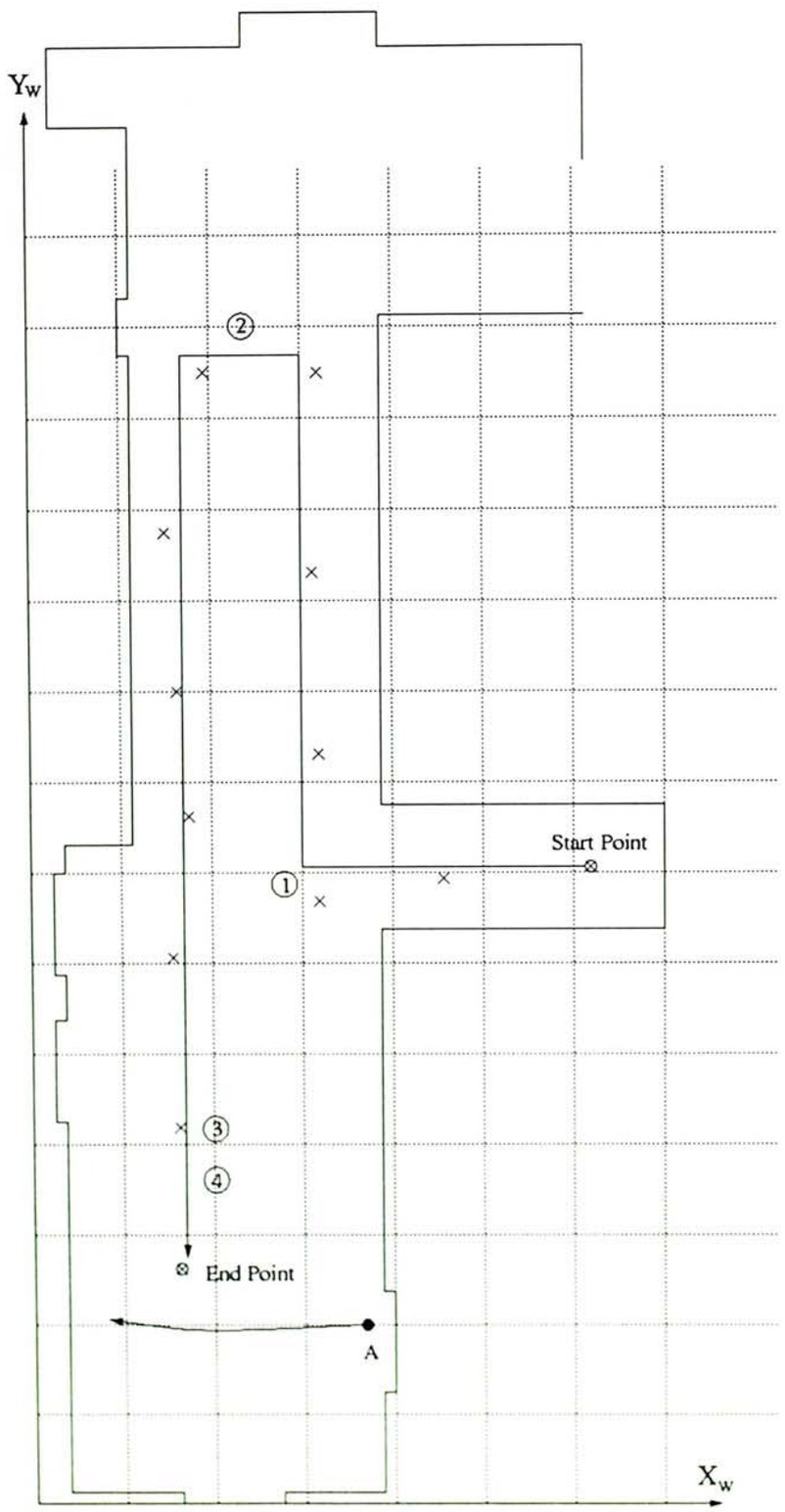


Figure 20. Robot trajectory in the experiment. X's are the actual robot trajectory. A moving human A is crossing the corridor while the robot localizing itself.

Left Image

Right Image

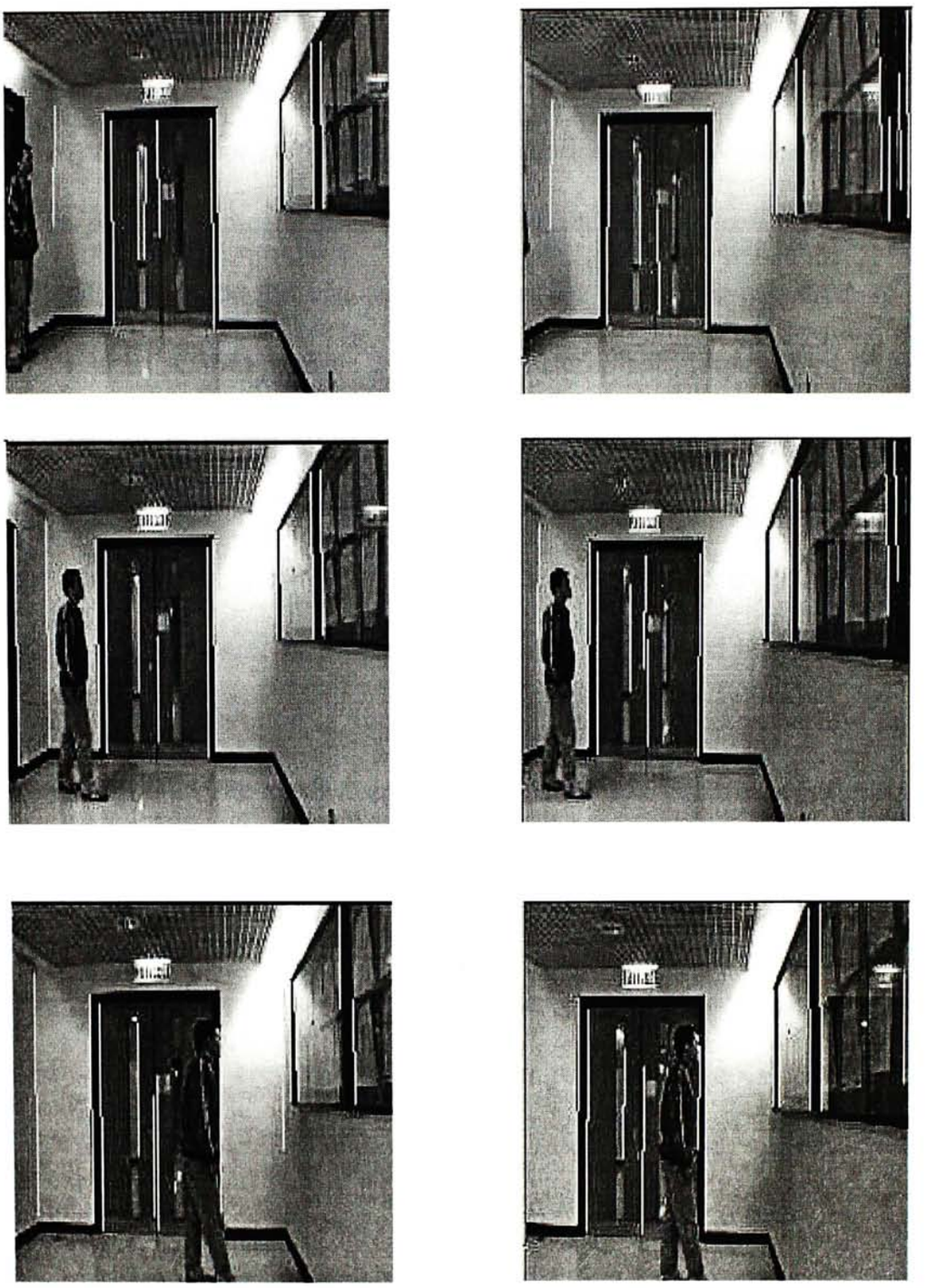


Figure 21. Image sequence for constructing a mature local map for the last localization cycle. Note that the moving human contribute no edge in local maps. This proves the effectiveness of our dynamic noise-filtering scheme.

At the beginning of the navigation, since there has not enough edges for determining the robot orientation, the robot drifts from the original path. At position 1, the robot rotate 90-degree and facing a long corridor as in figure 13, orientation of the robot is corrected and it continue to move while keeping its orientation parallel to the corridor. The robot makes a 'U' turns in position 2 and keeps travel forward to the end point. Figure 21 shows the sequence of stereo image pair taken at from position 3 to 4. Each stereo pair outputs a local map and three of them are merged to form a mature local map. During taking these images, a human A is crossing the corridor and appeared in the image sequence. Note that no edge contribute by the moving human is added into the local map. For the experiment reported here, the initial resolution of the iterated hill-climb search is 8cm and the final resolution is 1cm. Figure 22 is the error comparison between odometry error and the error using the proposed localization algorithm. The distance between the robot's odometer position and the actual position is measured at the cross-positions as marked in figure 20. We can observe that the robot's odometry error accumulates as the robot travel. The localization algorithm proposed successfully keeps the position error to an average value about 10cm. This error includes the orientation error since orientation error reflects error as a translation error.

We can notice that at the first two localization cycles, the positional error is very closed to the odometry error. It was because the vision algorithm cannot determine the robot's orientation due to not enough edges for computation as in figure 18(b). One way to solve this is to release the search space to a 3D search (search for position and orientation). That is, instead of only finding the robot

position, the iterated hill-climbing search evaluates the robot orientation also (Figure 19). The drawback is the searching time increases to 8 seconds. Look at the table 4, after releasing the search space to 3D, the positional errors are reduced which makes the overall average positional error equals to 9.6cm. The error plot after released the search space is in figure 23 and we can see the position error is reduced. The numerical data can be found in table 4.

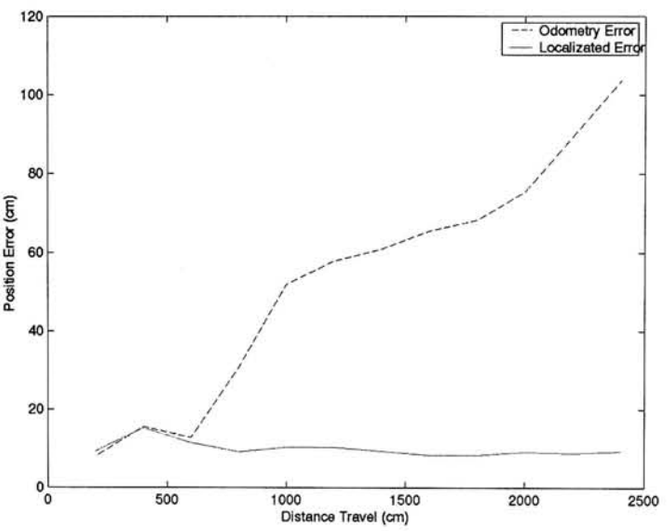


Figure 22. Odometry error V.S. localized error.

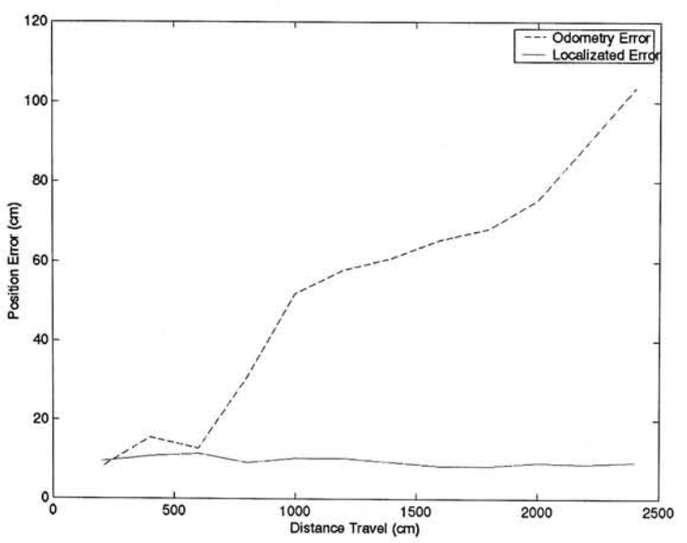


Figure 23. Odometry error V.S. localized error: With relaxed search space.

Distance Traveled (cm)	Odometry Error (cm)	Localized Error using a 2D search space (cm)	Localized Error using a 2D search space (cm)
200	8.0623	9.3238	→ 9.4523 cm
400	15.5563	15.2438	→ 10.8732 cm
600	12.7059	11.4802	
800	30.8423	9.2089	
1000	51.9278	10.3744	
1200	57.7690	10.3847	
1400	60.8908	9.3347	
1600	65.5433	8.3348	
1800	68.3008	8.2884	
2000	75.4345	9.2486	
2200	89.4530	8.8398	
2400	103.5672	9.3844	

Table 4. Numerical results.

Since the localization algorithm cannot determine the robot orientation at the first two localization cycles only, so the relaxed search is carried out for those two cycles. For the rest of the localization process, the search is remained as a 2D search to fasten the localization process.

5.2.1 Error Analysis

The constant average error in this experiment can be explained by the following source of errors:

- Error in sensing the environment. We use stereo vision to sense the environment. Recovery of 3D information using stereo has two sorts of errors:

error in stereo correspondence of feature points in left and right images and depth error due to limited camera resolution. First kind of error refers to the mismatch of features in the stereo pair of images. When we recover depth information using the wrong pair of features in left and right images, the triangulation process estimates the wrong 3D information. Second kind of error occurs because of the limited resolution of cameras. Image position error causes error in depth. Moreover, the resolution of the cameras are limited, error is larger when object far away from the cameras.

- Odometry error. As we can see in figure 11, the robot pose is updated by a retroactive update method, the robot current pose is obtained by combining the localization result from the previous localization process and the odometer readings, hence error is introduced by the odometer readings. It is a constant error as the previous localization process localizes the robot and this small error will not accumulate.

- Error in the iterated hill climbing search. Due to the time constrain problem, it is not feasible to have a global search in the configuration space to obtain a global solution. Iterated hill climbing search is a steepest-ascent search method that use an evaluation function to check how close a given state is to a goal state. The evaluation function we use is the matching function described in section 4.2.1. It is much faster than the global search, however, it may be trapped in local maxima in which the global goal state may not be found. So there has error if the local goal state is used as the solution in the search.

5.3 Discussions

The localization approach presented in this thesis only relies on the vertical edges to determine the robot position, and only one group of horizontal edges is used to estimate the robot orientation. The remaining group of edges may also be added to the mature local map to provide more information for determining the robot position. This group of edges is those horizontal edges that appear nearly perpendicular with the cameras optical axis. So they appear as horizontal edges in the image. Hence we can add them as lines in the mature local map in which only their depth or distance from the robot is useful as shown in figure 24. Crosses are the vertical edge and lines in the local map are the horizontal edges detected. These kind of horizontal edges are very useful to determine the robot's y position and orientation in the environment. However, adding the horizontal edges increases the image processing time since the stereo correspondence is also needed.

The map representation introduced in this thesis is suitable to integrate with the sonar or laser range finder information with the camera information. Proximity sensors usually update the local map using the evidence grid representation [32,33]. Evidence grids [35] are Cartesian grids, it divides the environment into cells, which stores the probability that the cell is being occupied. Evidence grids are similar to the map representation uses in this thesis. Since both Bayesian and Gaussian methods used in this thesis are probability representation, only a small modification will allow the fusion of proximity sensor readings with those obtained from cameras to enhance the information.

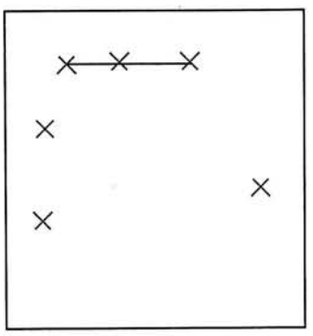


Figure 24. Horizontal edges are added in the local map.

In order to make the robot work in dynamic environments, the global map should change from time to time, for instance, if a new bookshelf is added or something in the environment have been moved away, all these action will make the global map change. So the robot should have the ability to update the environment information. Detection of new environment changes can be done by comparing the mature local map and global map after map registration. We exploit the fact that the two maps should be in nearly perfect registration immediately after the robot has localized itself in the environment. So the discrepancy between the global and the mature local map is most probably caused by unexpected obstacle that should be saved in memory. It should be added to the global map only when it was detected several times as the robot navigates to avoid wrong updating. From a local point of view, the immediate mismatched edges should be the new obstacle information and may give the information to the robot to avoid collision. It should be reminded that our features used are those edges in 3D prominent orientation in the environment (one vertical and two horizontals, perpendicular to each other). Most likely, moving objects such as people do not have these kinds of edges. Only

stationary objects like bookshelf, doors have those kind of edges. Hence, the particular noteworthy to our selection of features is that they can be used for both localization and obstacle detection that trace any changes in the global map. Once the robot has the ability to update the global map, it can work in a partially known or totally unknown environment and build the map incrementally.

Chapter 6

Conclusion

In this thesis, we have presented a vision based mobile robot localization system. A simple vision-filtering scheme is constructed to extract natural landmarks in the environment while minimizing the effect of dynamical noises. The system uses vertical edges to determine robot position and uses horizontal edges to determine robot orientation. In our representation, the local map only contains the 3D information of vertical edges in the environment and several of them are integrated together to form a mature local map. This map contains enough information for the registration with the global map. The registration involves an iterated hill-climbing search that determines the position of the robot in which the matching score is maximum. The matching score is a measurement of the "similarity" between global map and the mature local map. To improve the accuracy of the localization and keeping the search space to be small, a continuous localization approach is employed which repeats the localization processing continuously. A retroactive pose updating approach is employed keeping the robot navigates smoothly while localizing. The most important contribution in this work is we only uses vertical and horizontal edges to localize the robot continuously in the environment.

The proposed method is implemented in a RWI B21 robot and experiments show that the proposed algorithm can work in the environment where has human moving around, in other words, in a dynamic environment. Typically, the algorithm can eliminate the accumulated odometry error to constant of 10cm. The average processing time for extracting 3D-information from each pair of stereo image is 3 seconds while time for robot pose searching is 2 seconds. This is an advantage, which can allow the robot to travel at a maximum speed of 15cm/s without get lost in the environment.

Future works include combining the localization system with others modules like obstacle avoidance and path planning to make a autonomous mobile robot system and add the ability for updating the global map as described in chapter 5.

Appendix

A.1 Intrinsic and Extrinsic Parameters

Ref: [16,20]

Intrinsic parameters include focal length f, pixels per unit length (α_u and α_v), coordinates of the optical center of the camera (u_0 and v_0). Extrinsic parameters include the translate and rotation difference between camera and the world frame.

Camera manufacturer provides two of the intrinsic parameters we needed – pixels per unit length in u and v direction (α_u and α_v). Other parameters can be found as follows.

The relationship **M** (3x4 matrix) between the world coordinate (X, Y, Z) and image coordinate (u, v) assuming no lens distortion is:

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\text{with} \qquad M = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} R_T & t_T \\ 0 & 1 \end{pmatrix}$$

Where α_u and α_v are conversion factors (pixels per unit length). f is the effective focal length defined in pinhole camera model. u_0 and v_0 are the coordinates of the optical center of the camera (pixels). R_T and t_T are the rotation and translation matrixes from world to camera frame.

Matrix **M** is estimated by using least-squares error method that establishing all the point correspondences between the world and image frame to determine the entities of **M** by a set of over-determined system of equations. Point correspondences of features in image and world frame is established manually by first face the camera to the calibration pattern, usually are dots or squares painted on a paper, then save their image position and the corresponding world coordinates.

A.2 Relation Between Cameras (Stereo Camera Calibration)

Ref: [20,28]

The translation and rotation relationship between two cameras can be found by using the extrinsic parameters. The loop in figure A2 shows the estimation of this relationship. Two cameras is facing at the same calibration pattern, then relations A_1 and A_2, which are the extrinsic parameter of the cameras, forming a loop with the calibration frame. Hence, both the translation and rotation relationship can easily be found by closing the loop: $A = A_2 A_1^{-1}$.

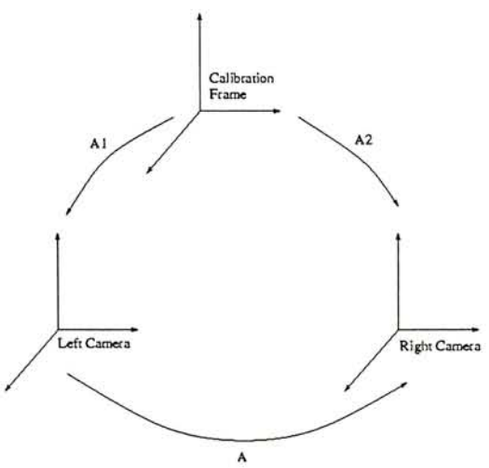


Figure A2. Relation between cameras and the calibration pattern.

A.3 Wheel-Eyes Calibration

Ref: [18,29]

The idea is same as stereo camera calibration but we have a larger loop this time. From the center loop in figure A3, an equation relating A, B and X can be immediately established:

$$\mathbf{AX=XB}$$

C1 and C2, P1 and P2 are two different locations of the left camera frame and the pan-tile base frame respectively. When the pan-tilt system moves to any new position, different A's and B's will come out. Let n be the number of different positions of the pan-tilt system with respected to a fixed calibration frame. From equation 2, a set of n-1 matrix equations is formed:

$$\mathbf{A_1X = XB_1}$$
$$\mathbf{A_2X = XB_2}$$
$$\ldots$$
$$\mathbf{A_iX = XB_i}$$
$$\ldots$$
$$\mathbf{A_{n-1}X = XB_{n-1}}$$

$\mathbf{A_i}$ = Transformation between position i and position i+1 of the camera frame.

$\mathbf{B_i}$ = Transformation between position i and position i+1 of the pan-tilt frame.

Matrix **X** is estimated by solving the above set of n-1 matrix equations. $\mathbf{A_i}$s can be found from A1, the pan-tilt angles and the dimension of links of the pan-tilt system define $\mathbf{B_i}$s.

There are many solution methods in the literature for this problem. The one we use can be found in [18]. It first decomposing equation AX = XB in two equation with R and T separated, then predict matrix X by a linear error

minimization method. Detail results can be found in a MAE 5020 term project report, named "Wheel-Eye Calibration", in 1998, 1st term. T is a matrix from the dimension specifications of the robot from RWI. Hence, with X known, the wheel-eyes calibration is finished.

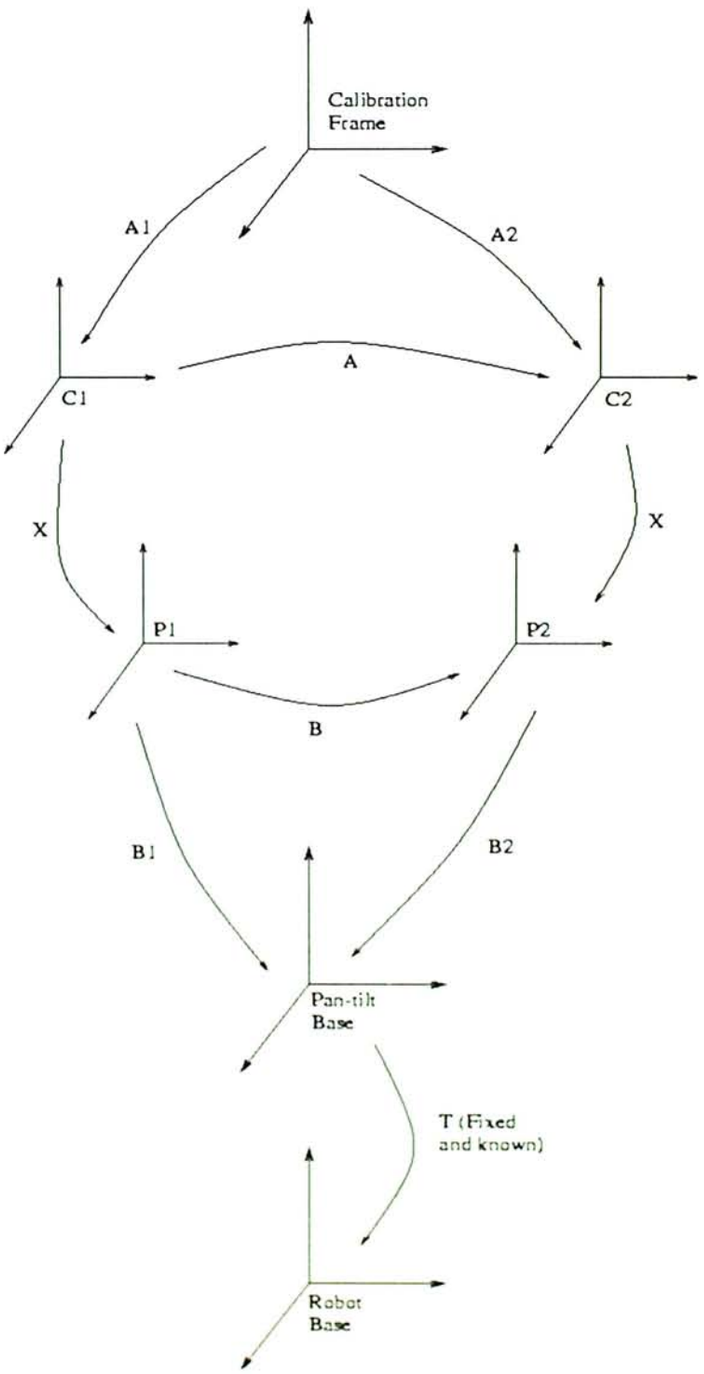


Figure A3. A view showing two different positions of the pan-tilt device.

A.4 Epipolar Geometry

Epipolar constraint is very useful in stereo vision, it reduce the 2D-search space to 1D space.

In figure A4, a projection point p in the left image is the projection of a world point P, which lie on the line passing through left camera center C and p. The plane containing the two camera center C and C' and p defines the epipolar plane. This plane intersects the right image as a line called *epipolar line*. The projection of the world point P on the right image must lie on this epipolar line.

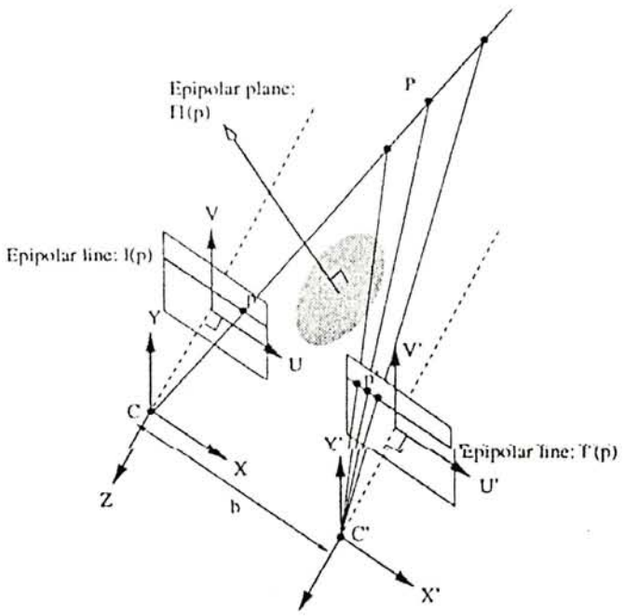


Figure A4. A parallel axis stereo camera system. Note that in general, the two camera axis may not be paralleled.

A.5 The Tele-operate Interface

This section describes the tele-operation interface we developed and used in this project. The main function of the interface is to control the movement of the robot with a computer under human control. Instead of using the joystick connected to the robot, the robot can be commanded to navigate safely under the control of human through the interface. During the navigation, the all needed sensory data are recorded simultaneously. Figure A5 shows the tele-operation interface. You can command the robot to move under different speed and different acceleration (Translation and Rotation) and adjust the pan and tilt angle of the stereo system. The latest sonar readings are shown graphically in the Sonar Map window and images taken by camera are also displayed.

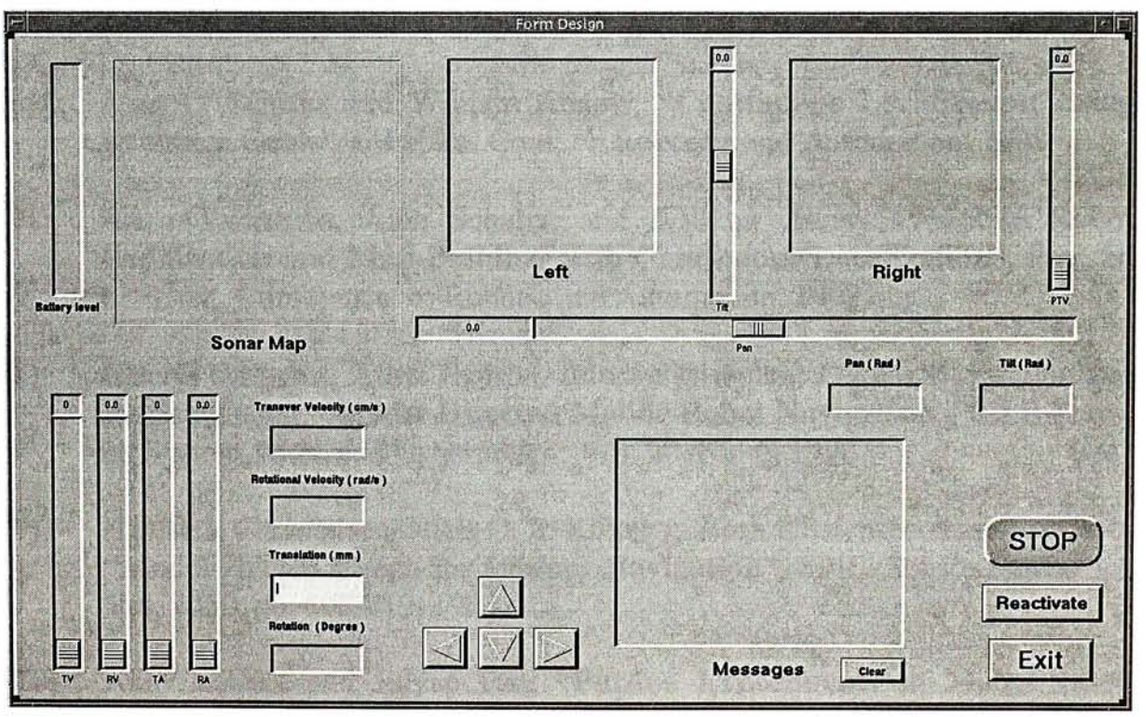


Figure A5. The tele-operation interface.

References

[1] J. Borenstein, H. R. Evertt, L. Feng, and D. Wehe, **"Mobile Robot Positioning-Sensors and Techniques"**, Invited paper in Journal of Robotic System. Special Issue on Mobile Robots. Vol. 14 No. 4, pp. 231-249.

[2] J. Borenstein, H. R. Evertt, L. Feng, **"Where am I? Sensors and Methods for Mobile Robot Positioning"**, Navigating Mobile Robots—Systems and techniques.

[3] M. Oussalah, H. Maaref, C. Barret, **"Positioning of a Mobile Robot With Landmark-Based Method"**, IEEE/RSJ Proc. of Inter. Conf. on Intelligent Robots and Systems, Sep 1997, (IROS 97). Vol 2, pp.865-871.

[4] Giuseppe Oriolo, Giovanni Ulivi, and Marilena Vendittelli, **"Real-Time Map Building and Navigation for Autonomous Robots in Unknown Environments"**, IEEE Tran. on Systems, Man, And Cybernetics—Part B. Vol. 28, No. 3 June 1998.

[5] Wolfram Burgard, Dieter Fox, Daniel Hennig and Timo Schmidt, **"Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids"**, Proc. of the Fourteenth National Conference on Artificial Intelligence.

[6] Alan C. Schultz and William Adams, **"Continuous Localization Using Evidence Grids"**, IEEE Int. Conf. on Robotics and Automation, 1998.

[7] Brian Yamauchi, Alan Schultz, and William Adams, **"Mobile Robot Exploration and Map-Building with Continuous Localization"**, Proc. of IEEE Int. Conference. of Robotic and Automation, 1998.

[8] Claude Fennema, Allen Hanson, Edward Riseman, J. Ross Beveridge, and Rakesh Kumar, **"Model-Directed Mobile Robot Navigation"**, IEEE Trans. on System, Man, and Cybernetics. Vol. 20, No. 6, 1990.

[9] Henrik I. Christensen, Niels O. S. Kirkeby, Steen Kristensen, Lars Knudsen, **"Model-Driven Vision for In-door Navigation"**, SPIE, Sensor Fusion VI, Vol. 2059, pp. 408-419, 1993.

[10] Akio Kosaka and Juiyap Pan, **"Purdue Experiments in Model-Based Vision for Hallway Navigation"**, Proc. of Workshop on Vision for Robots in IROS 95.

[11] X. Lebegue and J. K. Aggarwal, **"Detecting 3-D Parallel Lines for Perceptual Organization"**, Proc. 2nd Euro. Conference on Computer Vision., 1992, pp. 720-724.

[12] A. Kosaka, M. Meng and A. C. Kak, **"Vision-Guided Mobile Robot Navigation Using Retroactive Updating of Position Uncertainty"**, Proc. of IEEE Int. Conf. On Robotics and Automation, 2, pp. 1-7, 1993.

[13] Cordelia Schmid and Andrew Zisserman, **"Automatic Line Matching across Views"**, CVPR 97, pp.666-671.

[14] Olivier Faugeras and Bernard Mourrain, **"On the geometry and algebra of the point and line correspondences between N images"**, Proc. of ICCV95, pp. 951-956.

[15] Dinesh Nair and J. K Aggarwal, "**Moving Obstacle Detection from a Navigating Robot**", IEEE Transactions on Robotics and Automation. Vol 14, no. 3 pp. 404-416 , June 1998.

[16] Roger Y. Tsai, **"A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the Shelf TV Cameras and lenses"**, IEEE Journal of Robotics And Automation, Vol. RA-3 (4): 323-344.

[17] X. Lebegue and J. K. Aggarwal, **"A Mobile Robot for Visual Measurements in Architectural Applications"**, in Proc. IAPR Workshop Machine Vision Application. Tokyo, Japan, 1992, pp. 195-198.

[18] Radu Horaud and Fadi Dornaika, **"Hand-Eye Calibration"**, Inter. J. of Robotics Research, vol. 14(3): 195-210.

[19] O.D. Faugeras and M. Hebert, **"The Representation, Recognition and Locating of 3D Objects"**, 1996 Int. J. of Robotics Research, 5(3):27-52.

[20] O.D. Faugeras, and G Toscani, **"The calibration problem for stereo"**, Proc. of Computer Vision and Pattern Recognition, pp. 15-20, 1986.

[21] R. I. Hartley, **"Euclidian Reconstruction from Uncalibrated Views"**, Applications of Invariance in Computer Vision, p.237-256, Springer Verlag, Berlin Heidelberg, 1994.

[22] Akihisa Ohya, Akio Kosaka, and Avi Kak, **"Vision-Based Navigation of Mobile Robot with Obstacle Avoidance by Single Camera Vision and Ultrasonic Sensing"**, Proc. of Int. Conf. on Intelligient Robots and Systems, Sep 1997, (IROS 97), Vol 2, pp.704-711.

[23] Zhengyou Zhang and Olivier Faugeras, **"Building a 3D World Model with a Mobile Robot: 3D Line Segment Representation and Integration"**, IEEE, 1990.

[24] In. Hyuk Moon, Jun Miura, Yoshiharu Yanagi, and Yoshiaki Shirai, **"Planning of Vision-Based Navigation for a Mobile Robot under Uncertainty"**, Proc. of 1997 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Sep 1997, (IROS 97), Vol2, pp. 1202-1207.

[25] Xavier Lebegue, and Aggarwal J. K, **"Significant Line Segments for an Indoor Mobile Robot"**, IEEE Trans. on Robotics and Automation. Vol 9. No. 6. pp. 801-815.

[26] C. Tomasi and T. Kanada, **"Shape and Motion from image streams under orthography: A Factorization Method"**, Inter. J. of Computer Vision, 9(2): 137-154, 1992.

[27] Dieter Fox, Wolfram Burgard, and Sebastian Thrun, **"The Dynamic Window Approach to Collision Avoidance"**, IEEE Robotics & Automation Magazine, p23 - p33, March 1997.

[28] R. Chung and S.-K. Wong, **"Stereo Calibration from Correspondences of OTV Projections"**, IEEE Proc. Vision Image Signal Process., Vol. 142, no. 5, October, 1995.

[29] Xavier Lebegue and J. K. Aggarwal, **"A Mobile Robot for Visual Measurements in Architectural Applications"**, IAPR Workshop on Machine Vision Applications, 1992.

[30] **"Wheel-Eye Calibration"**, MAE 5020 term project report. Department of Mechanical and Automation Engineering, CUHK, 1998, 1st term.

[31] A. Kosaka and A. C. Kak, **"Fast Vision-Guided Mobile Robot Navigation Using Model-Based Resoning and Prediction of Uncertainties"**, CVGIP – Image Understanding, **56**(3), pp. 271-329, 1992.

[32] Elfes, A. E., **"Multi-source Spatial Data Fusion using Bayesian Reasoning"**, Data Fusion in Robotics and Machine Intelligence, Academic Press, pp 137-163, 1992.

[33] K. Hughes and R. R. Murphy, **"Ultrasonic Robot Localization using Dempster-Shafer Theory"**, SPIE Stochastic Methods in Signal Processing, Image Processing, and Computer Vision, invited session on Applications for Vision and Robots, San Diego, CA, July 19-24, 1992.

[34] H. P. Moravec, **"Sensor Fusion in Evidence Grids for Mobile Robots"**, AI Magazine, pp. 61-74, 1988.

[35] Hans Moravec and Albert Elfes, **"High Resolution Maps from Wide Angle Sonar"**, Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 116-121, 1985.

[36] Alexander Zelinsky, **"A Mobile Robot Exploration Algorithm"**, IEEE Trans on Robotics and Automation, Vol. 8, No. 6, Dec, 1992.

[37] Alexander Zelinsky, Jarvis R. A, Byrne J. C and Yuta S, **"Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot"**, Int. Conf. On Advanced Robotics (ICAR'93), Nov, 1993.

[38] H. Jie, T. Tadawute, I. Terakura, N. Ohnishi and N. Sugie, **"Mobile Robot and Sound Localization"**, Proc. of Int. Conf. on Intelligient Robots and Systems, Sep 1997, (IROS 97), Vol 2, pp.683-689.

[39] Jamea J. Little, Jiping Lu and Don R. Murray, **"Selecting Stable Image Features for Robot Localization Using Stereo"**, Proc. of Int. Conf. on Intelligient Robots and Systems, Oct 1998, (IROS 98), Vol 2, pp.1072-1077.

[40] K. Tashiro, J. Ota, Y. C. Lin and T. Arai, **"Design of the Optimal Arrangement of Artificial Landmarks"**, Proc. of 1995 IEEE Int. Conf. on Robotics and Automation, May, 1995, pp.407-414.

[41] J. Horn and G. Schmidt, **"Continuous Localization for Long-Range Indoor Navigation of Mobile Robots"**, Proc. of 1995 IEEE Int. Conf. on Robotics and Automation, May, 1995, pp.387-394.

[42] T. Nishizawa, A. Ohya and S. Yuta, **"An Implementation of On-Board Position Estimation for a Mobile Robot"**, Proc. of 1995 IEEE Int. Conf. on Robotics and Automation, May, 1995, pp.395-400.

[43] A. Ohya, K. Akio and K. Avinash, **"Vision-Based Navigation by a Mobile Robot with Obstacle Avoidance Using Single-Camera Vision and Ultrasonic Sensing"**, IEEE Trans on Robotics and Automation, Dec 1998, Vol.14 No. 6, pp. 969-978.

[44] Y. Matsumoto, T. Shibata, K Sakai, M. Inaba and H. Inoue, **"Real-Time Color Stereo Vision System for a Mobile Robot based on Field Multiplexing"**, Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 1934-1939, 1997.

[45] S. Thrun, D. Fox and W. Burgard, **"Probabilistic Mapping of an Environment by a Mobile Robot"**, Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 1546-1551, 1998.

[46] I. Horswill, **"Visual Collision Avoidance by Segmentation"**, Proc. of Int. Conf. on Intelligient Robots and Systems, May 1994, (IROS 94), Vol 2, pp.902-909.

[47] R. Jain, R. Kasturi, B. G. Schunck, **"Machine Vision"**, McGRAW-HILL, Inc, International Editions, 1995.