

# Mobile Robot Navigation through Digital Landmarks

Jiali Shen and Huosheng Hu

Department of Computer Science, University of Essex, Colchester CO4 3SQ, U.K.

Email: [jshenh@essex.ac.uk](mailto:jshenh@essex.ac.uk), [hhu@essex.ac.uk](mailto:hhu@essex.ac.uk)

**Abstract:** Localisation, the ability to find out the position, is essential to a mobile robot. The most common localisation method is to use a Kalman filter to fuse the data from dead reckoning data with the observations of environment features. The different artificial landmarks have been introduced to correct the error of dead reckoning, including visual landmark based methods. Since multiple landmarks are needed in a complex or large environment, the problem on how to distinguish multiple landmarks comes. This paper introduces the digit landmarks for recognition of a large number of artificial digital landmarks in the robot environment. Experimental results show the feasibility, extendibility, and large capacity of the proposed method.

**Keywords:** Mobile Robots, Localisation, Visual Landmarks

## 1. Introduction

Localisation, the ability that the robots can locate themselves in the real world [2], is a basic and important aspect for a mobile robot to do something useful. The existing localisation algorithms can be divided into two categories: relative and absolute. Relative localisation relies on both odometry and robot start position, which however has accumulative errors. In contrast, absolute localisation relies on the observation of environmental features based on different sensor technologies. In practice, a combination of both methods is required to achieve robust localisation for a mobile robot, in which Kalman filters or extended Kalman Filter (EKF) is one of popular approaches [10].

Visual based methods use cameras to collect information, including 3 kinds of camera installation. Firstly, the cameras are fixed on the robot to be the eyes and the images are the robot's sight namely Self-Observing Algorithms (SOA). In different places, the robot sees different scenes; and the algorithms calculate the current position via what the robot is seeing. Another method is to install the panoramic cameras in the robot environment, e.g. in the ceiling of an indoor environment. They don't move with the robot. The camera works as a spectator. Algorithms abstract features from the captured images to obtain robot position information, which we name it as the spectator method. An example of this method can be found in [1]. Thirdly, stereo vision is also researched

widely, and the stereo analysis is the focus of these researches [15].

In a SOA system, the robot captures images from the cameras equipped, and extracts information for localisation. According to the information extraction, the SOA can be separated into two main categories. Some algorithms focus on landmarks, the easily detected features (either nature or artificial), and used the foreseen position of these landmarks to deduct where the robot is [7-9]. Other methods consider the captured images as a whole part, and match them to a set of pre taken images (sometimes panoramic images), find out the relationship between them and then implement the localisation [11-14]. Kim and Cho presented a method to locate the robot on a single landmark [7]. Several features are extracted from the landmark and the position of the observing camera can be deducted, using geometric calculation.

In visual landmark based algorithms, multiple landmarks are often required to provide the position information in a large or complex environment. These landmarks are certainly placed in different places so that the distinguishing of multi-landmarks is important. In some systems, the landmarks cannot be distinguished by themselves; the algorithms introduce the rough position information obtained from dead reckoning to identify the landmarks detected. This method doesn't work in the condition that the robot is initialised or kidnapped.

This paper presented a localisation algorithm which adopts digital landmarks. The digits can be used to distinguish the landmarks, and no other information is required to complete the localisation. This landmark also contains the features similar with that of Kim's, and can provided enough information by a single landmark. Least Square Estimator is used to reduce the locating errors. The dual-landmark method is also proposed to solve the problems in large observing angles.

The rest of this paper is organised as follows. Previous work on landmark based localisation methods is briefly reviewed in Section 2. Section 3 describes the digital landmarks adopted in this research. Section 4 discusses the localisation algorithm based on new landmarks. The new location method is proposed in Section 5. The experiment to test the proposed algorithm is presented in Section 6. Finally, the conclusion and the future work are presented in Section 7.

## 2. Previous Work

Landmarks are distinct features in the environment which can be easily separated from other objects or backgrounds. Landmarks are classified as either artificial landmarks or natural landmarks. Artificial landmarks can be geometrical shapes, e.g. rectangles, circles, corners, or bar-coded targets containing extra information. In contrast, natural landmarks are part of the real environment, and consist of vertical edges, lines and ceiling lights, etc.

In general, landmark based localisation methods are composed of two modules: landmark detection and position deduction, which play different roles in the robot positioning process.

### 2.1 Landmark Detection

In general, all landmarks have fixed and known positions and should be easy to be detected by the sensors relative to them. Among the various kinds of sensors that are used to detect the landmarks, cameras or video sensors are the most common ones.

For nature landmarks, Hough transform is a commonly used method to detect these features such as lines or corners. For instance, a SIFT (Scale Invariant Feature Transform) landmarks combined with horizontal and vertical lines are introduced in [3], in which, Hough and RANSAC approaches are adopted to detect SIFT. In a corridor environment, Vassallo used the crossing lines between walls and the floor as the features so that the Sobel gradient operator was used to detect them [4].

Detection is much easier with artificial landmarks, which are designed for optimal contrast. For instance, they are purposively placed, easily detected objects in the environment. They are often specially coloured objects [5][6], high contract strips [7][10] or simply white lines on the dark-coloured floor [8].

### 2.2 Position Deduction

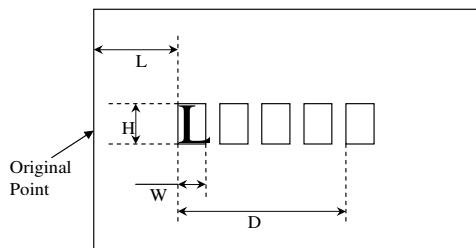
After extracting landmarks from the captured images, the problem of how to deduct the position of the robot comes. Triangulation is still a widely used method in visual based algorithms. In most of the application, the images are further processed besides beacon like usage. Basically, a camera coordinates and a global coordinates (or world coordinates) are required.

In [4], two lines (the two sides of corridor floor) are detected, and the vanishing point, crossing point of the two lines, is adopted to guide the robot to move in the middle of the corridors. Kim and Cho proposed three separate feature points that are extracted from one image [7]. The original arrangement of the features is known; therefore the robot can locate itself by calculating the feature positions in the single image.

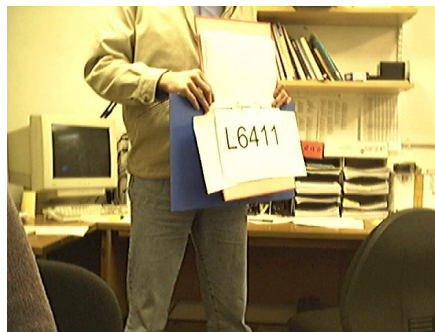
## 3. The proposed landmarks

We draw inspiration from wide applications of License Plate Recognition (LPR), and propose a new type of

landmarks with the embedded digit numbers. These landmarks are similar to the license plates used in transport. Figure 1(a) presents the format of the proposed landmark, and Figure 1(b) shows a real landmark hold by a person.



(a) The format of the landmark



(b) An example of the landmark

Figure 1 Proposed Landmark

Each landmark has the following features:

- Five characters, the letter L followed by 4 digits, are printed on the landmark.
- Each of the five characters has the same size, and the clearances between the characters are all the same. (H,W & D in Figure 1(a)).
- The positions of the characters are also known. (L in Figure 1(a))

Note that in the experimental system of this paper, we select the parameters are adopted:  $L=33$ ,  $D=200$ ,  $H=66$  and  $W=34$  (mm), which may be changed in different application environments.

The digits are the index of the landmark and the algorithm can identify the landmark with a digits recognition method. The standard size of the characters contains enough information for robot localisation [7]. Since the proposed landmark is similar to a license plate, many algorithms developed for license recognition can be used here directly, including the fuzzy-map method for locating the plate and the neural network for character recognition [16], and the fast plate location method based on vertical edges of the images [17].

## 4. Digits Recognition

We propose a new landmark recognition algorithm that has 3 main modules: region finding, digits finding and digits recognition, as shown in Figure 2.

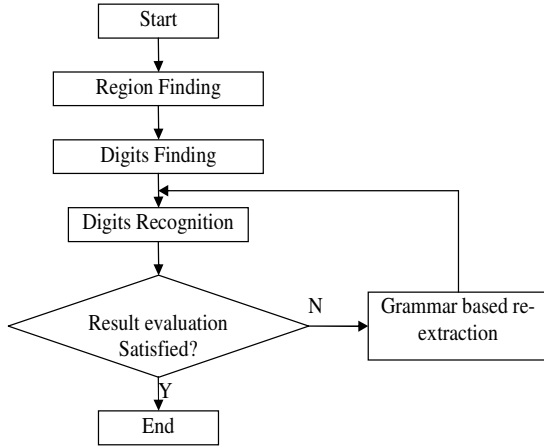


Figure 2 Landmark recognition algorithm

#### 4.1 Region Finding Module

This module is to find out all the probable regions that contains the landmark digits and exclude as more background as possible. Considering the features of the digits (sharply rising and falling edge in pairs in a horizontal scan line), we develop a simple region finding algorithm for extracting potential regions from images being captured. Figure 3 shows its flow chart.

While scanning the lines, the program will count the edge pairs (a pair is composed of a rising edge and a falling edge), and recorded the line sections that contains more than 4 edge pairs. In a scan line, the program may record more than one sections if the pairs are to further away from each other. The region extraction module analyses the line sections recorded and finds out the probable regions based on the following assumptions:

- The line sections will gather closely in the digits region.
- The numbers of line sections in the digits region will not be less than 10.
- The clearance between line sections in a digit region will be limited.

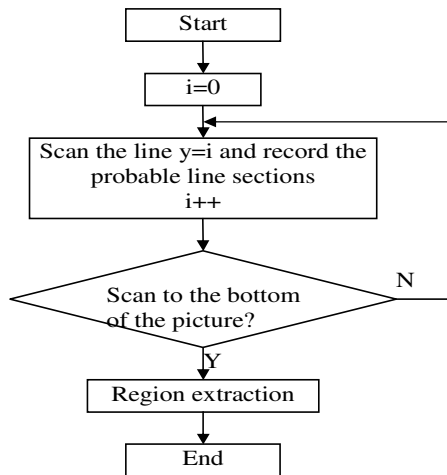


Figure 3 Region finding

#### 4.2 Digits Finding Module

The digits finding module is mainly based on an edge following algorithm. The steps can be described as follows:

- Scan the lines from top to bottom of a probable region, until finding an edge pixel of a probable digit.
- Follow the edge of the probable digits and record the parameters, e.g. position, width, height etc.
- Repeat step 1 and 2, until finding out all the probable digits in the region.

Figure 4 is the result of digits extraction. It should be noticed that two more fake digits are extracted in the right top of Figure 6. To remove these fault digits, we use the grammar based re-extraction module shown in Figure 2. Since the contrast of the characters is high and the edges are sharp, the digits finding module gives very accurate edge data, which is useful for the stage of position deduction.

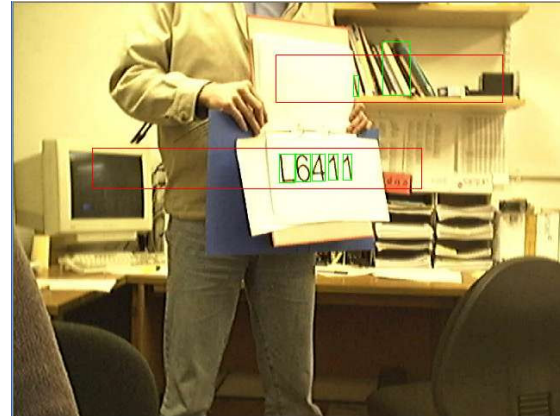


Figure 4 Digits finding module

#### 4.3 Digits Recognition Module

The digits recognition is composed of several steps as shown Figure 5.

**Normalization** binarizes the image areas decided by the  $i^{\circ}$  Digits Finding module and normalizes them to  $64*64$  arrays, no matter the original size of the digits. If some noisy areas were found with the digits, this step will normalize them as well, in order to avoid losing information. Figure 6 (a) is the result of Normalization of the digits detected.

**Thinning** is to extract out the skeleton (one-pixel-wide central line of a line). The skeleton is essential for texture analysis of a pattern. The end points, bifurcate points etc. can be extracted from the skeleton. The program adopts an updated Hilditch algorithm to implement the thinning operation. Figure 6(b) is the result of thinning. There often are some noises in the thinning image, e.g. some odd pixels which will generate fake endpoints and bifurcate points.

**Noise Remove** is to solve these problems. This step removes the pixels according to the following rules: (i) The isolate pixels are removed; (ii) Short lines (the length is less than 60 pixels) are removed; and (iii) Short odd lines are removed. An odd line is composed the pixels from an endpoint to a bifurcate point. The bifurcate point pixel is preserved while processing. Figure 6 (c) is an example of noise removed images.

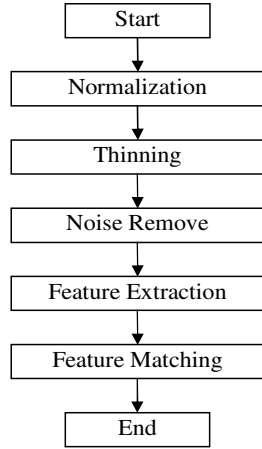


Figure 5 Flowchart for digits recognition

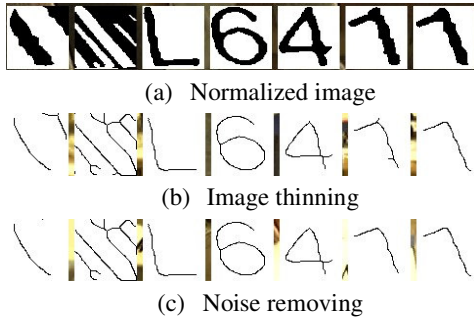


Figure 6 Normalization, thinning & noise removing

**Feature Extraction** extracts a grid based 9-element feature vector  $F=(f_1, f_2, \dots, f_9)^T$  for each of the **normalized probable digits (NPD)**. The 9 elements express the ratio of the number of black pixels in a sub-area. The following figure gives the serial number of the sub-areas in a NPD. The borderlines of sub-areas are the four lines shown in Figure 7, the coordinate value of a NPD is from 0-63 in both x and y axils.

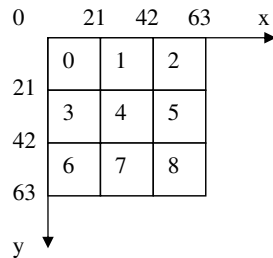


Figure 7 Sub-area numbers in a NPD

The elements are defined by the following equations:

$$f_i = \frac{N_i}{\sum_{i=0,8} N_i} \quad (1)$$

where  $N_i$  is the number of the black pixels in the  $i^{\text{th}}$  sub-area shown in Figure 7

For example, the feature vector for the letter 'L' in Figure 6 (c) is  $(0.2343, 0, 0, 0.246, 0, 0, 0.246, 0.168, 0.144)^T$ . We can find that in the NPD for 'L' no black pixel is in sub-area 1, 2, 4 and 5; and the black pixels in sub-area 7 & 8 are relatively smaller than those of the sub-area 0, 3 & 6.

Because the features are relative values in stead of absolute ones, the feature values are free from the different exposure level of the image, which causes the different width of the character strokes. The features are robust to the sloping digits, which are due to the sloping camera. This image in Figure 1 (b) is an example for sloping digits. We may found in the NPD that the distribution of black pixels in each sub-area of the NPD doesn't change due to the slope. The feature extracted from it also proves that.

**Features Matching** calculates the scalar products of the feature vector extracted in the above step and those from the features library; then it will gives out the result according to the minimum scalar product. This step also contains a simple judgement of the results if more than one probable region is found. The following conditions are adopted to do this, if

- 5 digits are found in a region.
- The first character of a region is recognized as 'L'.
- The minimum scalar products are very small.
- The region is more probable to be the right one.

Another function of this step is to connect the characters recognized into a string according to the right region judgment and positions of digits and then output it.

## 5. Position Deduction

This step is to locate the position of the robot, expressed by the vector  $p=(x,y,\theta)^T$ . Three coordinates is adopted to implement the function:

{W}: the global coordinates. The localisation is to find out  ${}^w P$ , the position vector in {W}.

{L}: the Landmark coordinates. It is fixed on the current landmark which is being seen. The original point is fixed on the position shown in Figure 1 (a).

{I}: the Image landmark. It is fixed on the image plane.

If a landmark is seen by the robot it is able to identify the landmark and get the position information of the landmark in {W} from the database, and therefore the transformation matrix  $C_L^w$  is known. If the position in {L},  ${}^L P$ , is deducted, the localisation can be done by the transformation

$${}^w P = C_L^w \cdot {}^L P \quad (2)$$

The problem now is to calculate the

$${}^L P = (x_c^L, y_c^L, \theta_c^L)^T \quad (3)$$

This paper presents two methods of localisation, triangulation and Least Square Estimation (LSE), in two cases, single landmark and double landmarks. In this section, the methods in different cases are introduced one after another.

### 5.1 Triangulation Method

Figure 8 is the sketch map of landmark imaging, using a pinhole model. P1 and P2 are two of the vertical edges of the five characters (10 edges all together). The position of P1 and P2 are known as  $(p_1^L, 0)$  and  $(p_2^L, 0)$ . The parameters of the landmark are shown in Figure 1(a).

According to the pinhole model, we get:

$$\frac{(y_2^L - y_1^L)r}{f} = \frac{H}{l_1} \quad (4)$$

and

$$\frac{(y_4^L - y_3^L)r}{f} = \frac{H}{l_2} \quad (5)$$

where,  $r$  is the resolution with unit of MPD (millimetres per dot), and  $(x_i^L, y_i^L)$  is the coordinate value of the  $i^{\text{th}}$  feature points, the both endpoints of the selected vertical edges.

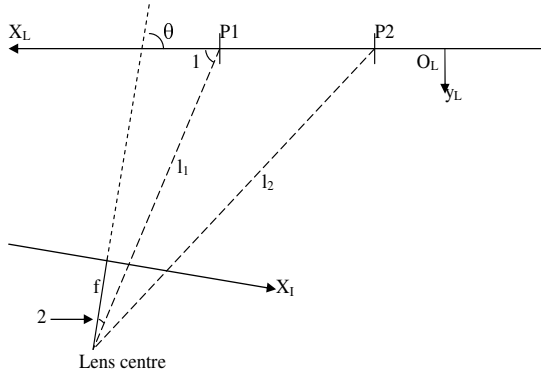


Figure 8 Landmark imaging

Assume that the position of the lens centre is  $(x_c^L, y_c^L)$ . According to Pythagoras theory, we have:

$$\begin{cases} (p_1^L - x_c^L)^2 + (y_c^L)^2 = l_1^2 \\ (p_2^L - x_c^L)^2 + (y_c^L)^2 = l_2^2 \end{cases} \quad (6)$$

Considering that  $y_c^L$  is always a positive value, we have:

$$\begin{cases} x_c^L = \frac{l_1^2 - l_2^2 + p_2^{L2} - p_1^{L2}}{2(p_2^L - p_1^L)} \\ y_c^L = \sqrt{l_1^2 - (x_c^L - p_1^L)^2} \end{cases} \quad (7)$$

In Figure 9, we can easily know that the direction is:

$$\theta = \angle 1 + \angle 2 \quad (8)$$

where

$$\angle 1 = \tan^{-1}\left(\frac{p_1^L - x_c^L}{y_c^L}\right) \quad (9)$$

and

$$\angle 2 = \tan^{-1}\left(\frac{x_1^L r}{f}\right) \quad (10)$$

Combined Equations (7) and (8) and substitute Equations (9) and (10) in, we have

$$\begin{cases} x_c^L = \frac{l_1^2 - l_2^2 + p_2^{L2} - p_1^{L2}}{2(p_2^L - p_1^L)} \\ y_c^L = \sqrt{l_1^2 - (x_c^L - p_1^L)^2} \\ \theta_c^L = \tan^{-1}\left(\frac{p_1^L - x_c^L}{y_c^L}\right) + \tan^{-1}\left(\frac{x_1^L r}{f}\right) \end{cases} \quad (11)$$

Here, we got the value  ${}^L P$ , and substitute it into Equation (1) the localisation is completed. Equation (11) is the result of localisation. The coordinates given by the program is in  $\{L\}$ .

The errors of this method are caused by the imprecise extraction of each character. Differentiate Equation (11):

$$\begin{cases} dx_c^L = \frac{1}{(p_2^L - p_1^L)}(l_1 dl_1 - l_2 dl_2) \\ dy_c^L = \frac{l_1}{\sqrt{l_1^2 - (x_c^L - p_1^L)^2}} dl_1 - \frac{x_c^L - p_1^L}{\sqrt{l_1^2 - (x_c^L - p_1^L)^2}} dx_c^L \\ d\theta_c^L = \frac{dx_1^L}{1 + \left(\frac{x_1^L r}{f}\right)^2} - \frac{2y_c^L(p_1^L - x_c^L)}{y_c^{L2} + (p_1^L - x_c^L)^2} (dx_c^L + \frac{dy_c^L}{y_c^L}) \end{cases} \quad (12)$$

Equation (12) shows the relationship between the errors of localisation and those of character extraction. It can be found that the error of locating is relative to  $l_1$  and  $l_2$ , the distances between the robot and the features, and the errors are supposed to be large when observing in large angles. In two-landmark case, the flowchart is the same like the one of single-landmark case. An updating is that the two features  $p_1$  and  $p_2$  can be selected from different landmarks, which can provide a wide arrangement of the features.

### 5.2 Least Square Estimator (LSE)

In a real application, the robot will continuously sample data using its onboard camera. Errors may be reduced by fusing the data of each sample. In this section, least square estimators are introduced in two different cases, single landmark case and dual landmarks case. The methods proposed in this paper are the extension of Boley's Least Square estimator [18].

#### Single-landmark LSE (SLSE)

The Single-landmark LSE algorithm is based on the coordinates shown in Figure 9. The origin point is placed at the landmark, and the x-axis is set parallel to the robot moving trajectories.

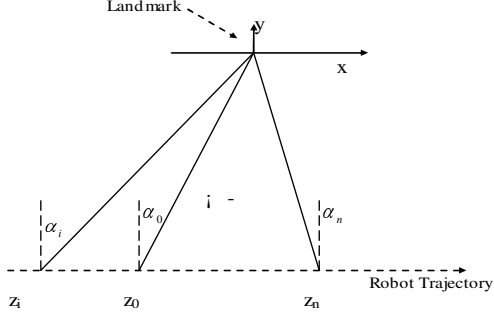


Figure 9 The measurement of the features

From each sampling point on a robot trajectory, the bearings to the landmark are measured as  $\alpha_i$   $i = 0, 1, \dots, n$ , by using the methods in Sec. 5.1. The position of each samples are noted in the vectors  $z_i = (x_i, y_i)^T$ , the distances between each positions and  $z_0$ , which can be obtained from the readings of odometer, are noted as  $d_i = z_i - z_0 = (x_i - x_0, y_0)^T$ . It is easy to know that:

$$\tan(\alpha_i) = \frac{x_0 + d_i}{y_0} \quad (13)$$

Rewrite Equation (13), we have:

$$x_0 \cos(\alpha_i) - y_0 \sin(\alpha_i) = -d_i \cos(\alpha_i) \quad (14)$$

Row-by-row collecting all the equations for  $i = 1, \dots, n$ , we have over determined equations which can be expressed as:

$$Az_0 = b \quad (15)$$

where

$$A = \begin{pmatrix} \cos(\alpha_1) & -\sin(\alpha_1) \\ \cos(\alpha_2) & -\sin(\alpha_2) \\ \dots & \dots \\ \cos(\alpha_n) & -\sin(\alpha_n) \end{pmatrix}, \quad z_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix},$$

and

$$b = \begin{pmatrix} -d_1 \cos(\alpha_1) \\ -d_2 \cos(\alpha_2) \\ \dots \\ -d_n \cos(\alpha_n) \end{pmatrix} \quad (16)$$

Using Least Square method [19], we can estimate the  $z$ :

$$z_0 = (A^T A)^{-1} A^T b \quad (17)$$

In this method, we adopt samples at the positions  $z_i$   $i = 1, \dots, n$  as the reference sample (RS) to estimate the wanted position  $z_0$ .

### Dual-landmark LSE (DLSE)

In this case, the coordinates is built on two landmarks, the original point is set to one landmark, and the x axle point to the other one. The coordinate values of two

landmarks are known as  $(0,0)$  and  $(D,0)$ , where  $D$  is the distance between two landmarks.

For each sampling point  $z_i$ , both landmarks are measured as  $(\alpha_{1i}, \alpha_{2i})^T$ , and two equations will be generated:

$$\tan(\alpha_i) = \frac{x_0 + d_{xi}}{y_0 + d_{yi}} \quad (18)$$

and

$$\tan(\alpha_i) = \frac{x_0 + d_{xi} - D}{y_0 + d_{yi}} \quad (19)$$

where,  $d_{xi}$  and  $d_{yi}$  are the displacement of the sampling points in two directions, which are obtained from the readings of odometer.

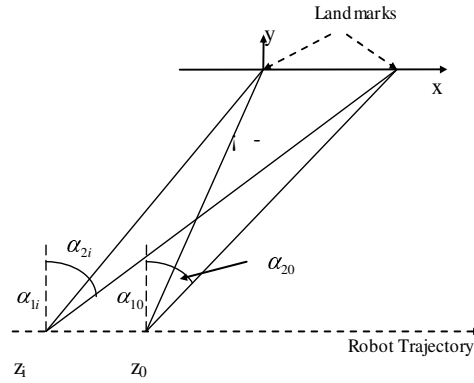


Figure 10 Measure the features from two landmarks

Rewrite Equation (18) and Equation (19), and collected the equations for each RS, we have:

$$Az_0 = b \quad (20)$$

where

$$A = \begin{pmatrix} \cos(\alpha_{11}) & -\sin(\alpha_{11}) \\ \dots & \dots \\ \cos(\alpha_{1n}) & -\sin(\alpha_{1n}) \\ \cos(\alpha_{21}) & -\sin(\alpha_{21}) \\ \dots & \dots \\ \cos(\alpha_{2n}) & -\sin(\alpha_{2n}) \end{pmatrix}, \quad z_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix},$$

and

$$b = \begin{pmatrix} \sin \alpha_{11} d_{y1} - d_{x1} \cos \alpha_{11} \\ \dots \\ \sin \alpha_{1n} d_{yn} - d_{xn} \cos \alpha_{1n} \\ \sin \alpha_{21} d_{y2} - (d_{x2} - D) \cos \alpha_{21} \\ \dots \\ \sin \alpha_{2n} d_{2n} - (d_{x2} - D) \cos \alpha_{2n} \end{pmatrix} \quad (21)$$

The position  $z_0$  can also be deducted out by using Equation (17).

### Different applications of LSE

In different applications, we may select different reference samples to estimate the required positions.

- Batch filter. The result is accurate but algorithm cannot be implemented in real time, because the estimator uses all the information when the sampling is completed.
- Recursive filter. It can be done in real time. However in the first a few samples, the locating error is big. But the estimated trajectory will soon converge with the more data samples.
- Moving windows (MW) filter. The computation cost is high. Since the previous samples have less inference value to estimate the current position, a good trade-off is to adopt finite samples, namely moving windows.

## 6. Experimental Result

The experiment is done with a  $i^{\circ}$  Logtech QuickCam<sub>i</sub>-web camera (1/4<sub>i</sub>-CCD sensor, 4.9mm lens). In the experiment, the landmark was fixed and the camera observed it and took pictures in different positions. The camera moved in several lines in different distances to landmarks as the straight lines shown in Figure 11. During the moving, the algorithm samples and estimates the current positions of the camera.

Figure 11 is the result of the triangulation method (see Section 5.1). Errors are big, especially in large observing angles. Figure 12 and Figure 13 present the result of LSE for single landmark and dual landmark cases, respectively. In the moving window filtering, we select 5 RSs for each sample in the experiment.

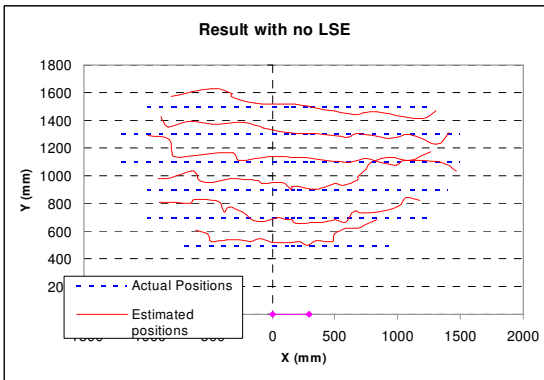
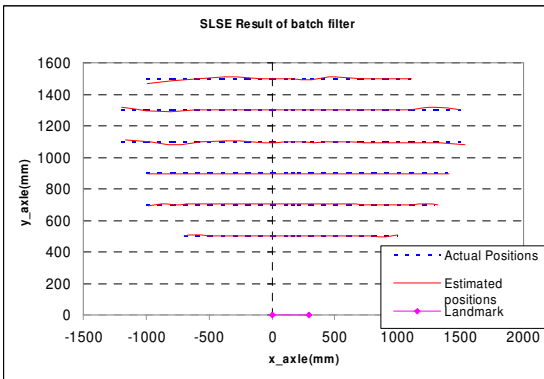
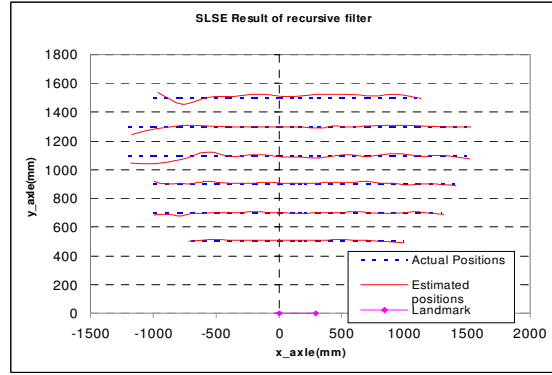


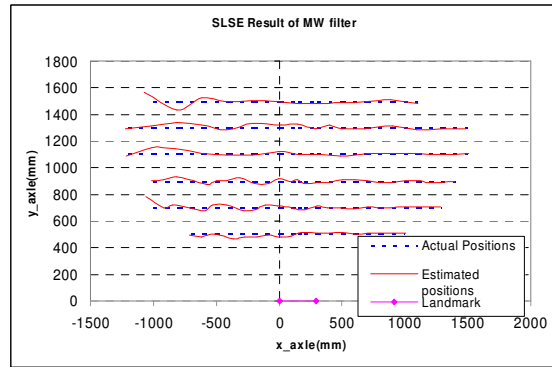
Figure 11 Triangulation Method without filtering



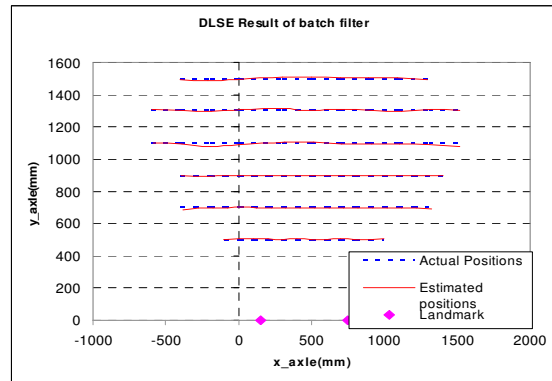
(a) SLSE result of batch filtering



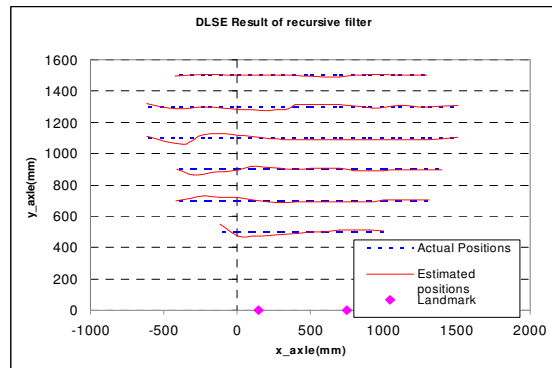
(b) SLSE result of recursive filtering



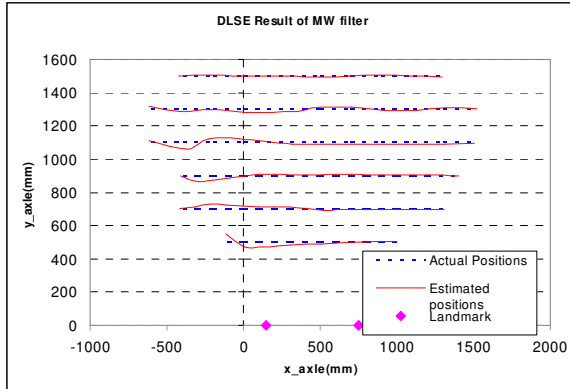
(c) SLSE result of move window filtering  
Figure 12 Single-landmark LSE



(a) DLSE result of batch filtering



(b) DLSE result of recursive filtering



(c) DLSE result of moving window filtering  
Figure 13 Dual-landmark LSE

The batch filter gave good localization results (Figure 12 (a) and Figure 13(a)), but it is not real time. Both the recursive and MW filters has a relatively large errors (the left side of each line), but the estimated result converge rapidly with more data collection, which can be used in a real time system and the MW algorithms have low computation costs.

## 7. Conclusions and Future Work

This paper presents a digital landmark based localisation algorithm for mobile robots, which uses a fast digits recognition method. The algorithm provides an easy solution on landmarks identification in complex environments, which is robust to slope images. Some advantages of the algorithm are:

- The system is easy to be extended by simply adopting different numbers in landmarks.
- The computation cost is relatively low by using the fast recognition method.
- The landmark identification is very reliable.

We are currently investigating the following four issues: (i) other information in the single landmark, e.g. the edges of middle characters, may be used; (ii) Other data fusion methods to use pre-known position information (from dead-reckoning or EKF; (iii) Multiply landmarks may be seen in some conditions, and triangulation methods may be used; and (iv) Localisation algorithms based on texture landmarks.

## References

- [1]. E. Kruse and F. Wahl, *Character-based Observation of Obstacle Motions to Derive Statistical Data for Mobile Robot Motion Planning*, Proceedings of IEEE Int. Conf. on Robotics and Automation, Leuven, Belgium, May 1998, pages 662-667
- [2]. L. Feng, J. Borenstein and H.R. Everett, *Where am I? Sensors and Methods for Mobile Robot Positioning*, University of Michigan, 1994
- [3]. S. Se, D. Lowe, and J. Little, *Global Localisation using Distinctive Visual Features*, Proc. of IEEE/RSJ Int. Symp. on Intelligent Robots & Systems, EPFL, Lausanne, Switzerland, October 2002, pages 226-231
- [4]. R. F. Vassallo, H. J. Schneebeli, and J. S. Victor, *Visual Servoing appearance for navigation*, Journal of Robotics and Autonomous Systems 31 (2000), pages 87-97
- [5]. H. Zhang and J.P. Ostrowski, *Visual Motion Planning for Mobile Robots*, IEEE Transactions on Robotics and Automation, Vol. 18, No. 2, 2002, pages 199-208
- [6]. D. Prasser and G. Wyeth, *Probabilistic Visual Recognition of Artificial Landmarks for Simultaneous Localisation and Mapping*, Proceedings of IEEE Int. Conference on Robotics & Automation, Taipei, Taiwan, September 14-29, 2003, pages 1291-1296
- [7]. J.J. Kim and H.S. Cho, *Real-time determination of a mobile robot's position by linear scanning of a landmark*, International Journal of Robotics, Vol. 10, 1992, pages 309-319
- [8]. H. Li, C. Xu, Q. Xiao, and X. Xu, *Visual Navigation of an Autonomous Robot Using White Line Recognition*, Proc. of the 2003 IEEE Int. Conference on Robotics & Automation, Taipei, Taiwan, September 14-29, 2003, pages 3923-3928
- [9]. L. Venturino, T. Gramegna, G. Cicirelli, and G. Attolico, *Autonomous Robot Positioning by Visual Servoing*, Proceedings of the 1<sup>st</sup> IASTED Int. Conference on Artificial Intelligence and Applications, Sept. 8-10, 2003, Benalmadena, Spain. Pages 40-44
- [10]. H. Hu and D Gu, *Landmark-based Navigation of Autonomous Robots in Industry*, Int Journal of Industrial Robot, Vol. 27, No. 6, 2000, Pages 458-467
- [11]. D. Huttenlocher, G. Klanderman, and W. Rucklidge, *Comparing images using the Hausdorff distance*, IEEE Trans. on Pattern Anal. Machine Intelligence. Vol. 15, pp. 850-863, 1993.
- [12]. A Adam, E. Rivlin, and I. Shimshoni, *Computing the Sensory Uncertainty Field of a Vision-Based Localisation Sensor*, IEEE Transactions on Robotics and Automation, Vol. 17, No. 3, June 2001, pages 258-267
- [13]. C. Zhou, Y. Wei, and T. Tan, *Mobile Robot Self-Localisation Based on Global Visual Appearance Features*, Proc of IEEE Int Conf. on Robotics and Automation, Taipei, Taiwan, 2003, pages 1271-1276
- [14]. H. Tagare, D. McDermott, and H. Xiao, *Visual Face Recognition for Autonomous Robots*, Proceedings of IEEE Int. Conference on Robotics & Automation, Leuven, Belgium, May 1998, pages 2530-2535
- [15]. T. Rabie and D. Terzopoulos, *Stereo Colour Analysis for Dynamic Obstacle Avoidance*, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 23-25 June 1998, pages 245 - 252
- [16]. S. Chage, L. Chen, Y. Chung, and S. Chen, *Automatic License Plate Recognition*, IEEE Transactions on Intelligent Transportation Systems, Vol. 5, No. 1 March 2004, pages 42-53
- [17]. H. Bai, J. Zhu, and C. Liu, *A Fast License Plate Extraction Method on Complex Background*, Intelligent Transportation Systems, Proceedings of IEEE, Vol. 2, 12-15 October 2003, Pages 985-987
- [18]. D. Boley, K. Sutherland, *A Rapidly Converging Recursive Method for Mobile Robot Localisation*, The international Journal of Robotics Research, Vol. 17, No. 10, October 1998, pages 1027-1039
- [19]. M. Betke, L. Gurvits, *Mobile Robot Localisation Using Landmarks*, IEEE Transactions on Robotics and Automation, Vol. 13, No. 2, 1997, pages 251-263