8

Vision Based SLAM for Mobile Robot Navigation Using Distributed Filters

Young Jae Lee and Sankyung Sung
Konkuk University
Seoul, Korea

1. Introduction

Simultaneous Localization and Mapping (SLAM) has been developed for mobile robot applications to obtain navigation information. SLAM is a strategy for building up maps in unknown environments and figuring out its position without any prior map information. To implement SLAM, various kinds of sensors such as laser range finders, ultrasonic sensors, vision sensors, etc. (Sheng Fu et al., 2007) can be considered. Each configuration often depends upon the required performance and cost condition.

In this chapter, we suggest a vision-based SLAM method to improve the navigation performance of mobile robot. It consists of a set of vision sensors and encoders which are equipped on mobile robot's wheels (Leonard & Durrant-Whyte, 1991). The heading and position of a mobile robot can be independently estimated with only 2 encoders. However the result often yields large inaccuracies due to errors and the noise of sensors (i.e. a vision sensor and encoders).

On the other hand, as the vision sensor is incorporated, a feature point tracking algorithm must be used (it's not mentioned in this chapter, because it's out of focus). In the integrated system, the relative position between feature points and the mobile robot is continuously estimated using information (i.e. feature points), so the position and heading errors are compensated using the filter estimation results (Hugh Durrant-Whyte & Tim Bailey, 2006).

The Extended Kalman Filter (EKF) has been widely used for SLAM (Jing Wu & Hong Zhang, 2007). Additionally, a particle filter is occasionally used with landmark or geometric constraints to enhance performance (Sukhan Lee & Seongsoo Lee, 2006), because the measurement model of the vision sensor is not linear. In comparison with previous works, this chapter takes a distributed particle filter approach without landmark information and geometric constraints. The distributed particle filter is known to have powerful performance under nonlinear, multi-modal models and features points in varying conditions (Ristic et al., 2004).

In addition to a vision sensor, range sensors are generally used to compute the 3-D position of feature points, because it is difficult to estimate the position in case of a single vision sensor system. This chapter uses a delayed initialization method instead of range sensors to compute the feature point position using single vision sensor system (Philip J. Schneider & David H. Eberly, 2003).

Computer simulations are conducted to demonstrate the performance of the suggested algorithm. The navigation performance is enhanced compared to the encoder only system during simulation. It depends upon the number of particles, feature points and the angle of vision sensor. Furthermore, using landmarks, which are from a previous known position, also improve the performance.

2. SLAM

Many vision sensor applications have been developed in recent years due to low costs, low power consumption and the development of high speed processors. Vision sensors can provide continuous image data. If there are some points that are not changed in the whole images during the image acquisition time, they can be continuously distinguishable from other objects (background or moving object) for some periods of time. Then the distinguishable points in the image data are referred to as feature points. Through tracking feature points, the amount of vision sensor's moving can be estimated.
There are many kinds of image types (gray scale, RGB, etc). In this chapter, 8 bits gray scale image is used, which expresses an image with 256 step intensity (Rafael C & Gonzalez, 2004). On image plan, intensity of pixel is changed by location (x, y) and acquisition time (t). Thus, the image (I) can be denoted by the following expression (Carlo Tomasi & Takeo Kanade, 1991).

$$I(x, y, t) \tag{1}$$

Let's assume that there are displacements ($d = (\xi , \eta)$) and the time difference (τ) is small on continuous 2 images, then the relation can be expressed as below.

$$I(x, y, t+\tau) = I(x-\xi, \quad y-\eta, \quad t) \tag{2}$$

It is observed that displacements are proportional to the movement of the vision sensor. If the displacements of a feature point are estimated, the movement of vision sensor can be obtained. Usually, feature point can be selected by using Harris Corner Detection (Konstantinos G. Derpanis, 2004), SUZAN (S. M. Smith & J. M. Brady, 1997) or Fast Corner Detection (Edward Rosten and Tom Drummond, 2005), which can be further used for the continuous tracking of feature point positions on the whole image.

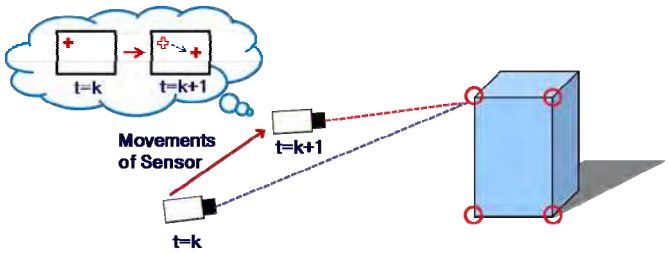

Fig. 1. Concpt of vision based SLAM (vSLAM)

Conventional SLAM algorithms using the vision sensor are adapted for the system integration. Using the vision sensor, feature points are selected and tracked on continuous frames. The vision sensor also provides bearing, elevation or range information from the feature point, which is deduced from the feature points tracking data. Other sensor output goes to SLAM block with vision sensor data. Navigation information (position, velocity, attitude, etc.) and errors in sensors are estimated by integrating information from vision and other sensor. Assuming feature points are fixed and not movable in the local coordinate frame, navigation errors come mainly from sensor outputs. Thus, by compensating estimated errors from sensor output, navigation data can be precisely calculated.

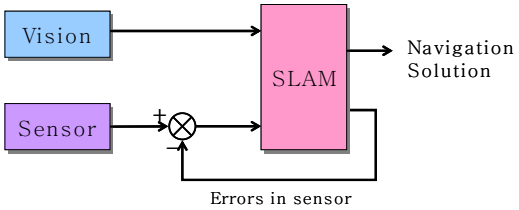

Fig. 2. Basic algorithm of a vision based SLAM

3. System Modeling

Figure 3 shows a simple mobile robot platform having only planar dynamics. It has a vision sensor and 2 encoders. A vision sensor acquires image continuously, then feature points are selected and tracked. The encoders are equipped on wheels and provide wheel rotation data. Given the information about wheel radius, the distance between the 2 wheels and pulses per rotation of encoder, range and heading information can be numerically computed. There are, however, various kinds of errors; wheel radius error, wheel distance error, slips error, conversion factor error, etc. The overall effect from the above mentioned sources resulted in accumulated errors and degraded navigation performance, which necessitated error compensation using aided sensors and filters.

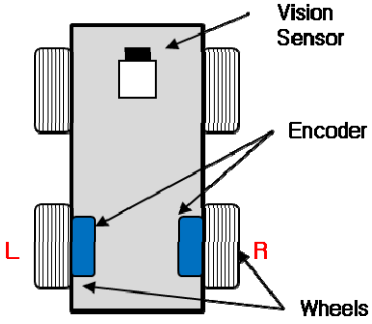

Fig. 3. Mobile robot equippd with encoders and vision sensor

3.1 Dynamic Model

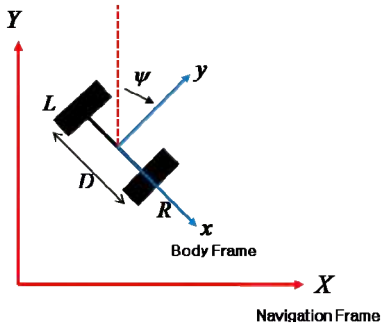

Fig. 4. Coordinate system of mobile robot

In Figure 4, X-Y denotes navigation frame and x-y denotes body frame on Figure 4. Position and heading of mobile robot can be expressed using the following equations.

$$\begin{aligned} X_{k+1} &= X_k + inc \times \sin\psi_k \\ Y_{k+1} &= Y_k + inc \times \cos\psi_k \\ \psi_{k+1} &= \psi_k + \frac{L-R}{D} \\ inc &= \frac{R+L}{2} \end{aligned} \tag{3}$$

L and R are movements of the left and right wheel, respectively. X, Y denotes position in each axis and ψ is heading. D is the distance between two wheels. S_R, S_L are scale factor errors of the encoder, so it is a kind of bias. We consider $S_R, S_L, \delta D$ as a random constant.

$$\begin{aligned} S_{R_{K+1}} &= S_{R_K} \\ S_{L_{K+1}} &= S_{L_K} \\ \delta D_{K+1} &= \delta D_K \end{aligned} \tag{4}$$

To estimate navigation error, the system model was changed into the following form using perturbation.

$$\begin{aligned} \hat{X}_{k+1} &= \hat{X}_k + \sin\hat{\psi}_k \times \left(\frac{R(1+S_{R_k}) + L(1+S_{L_k})}{2} \right) \\ \hat{Y}_{k+1} &= \hat{Y}_k + \cos\hat{\psi}_k \times \left(\frac{R(1+S_{R_k}) + L(1+S_{L_k})}{2} \right) \\ \hat{\psi}_{k+1} &= \hat{\psi}_k + \left(\frac{L(1+S_{L_k}) - R(1+S_{R_k})}{D+\delta D_k} \right) \\ \hat{X}_k &= X_k + \delta X_k,\ \hat{Y}_k = Y_k + \delta Y_k,\ \hat{\psi}_k = \psi_k + \delta\psi_k \end{aligned} \tag{5}$$

Applying a perturbed model and small angle assumption, (3) can be changed into (4). State vector consists of position error (X, Y), heading error, scale factor error (R, L) and wheel distance error.
The following state space equation is an error model of the system.

$$\mathbf{X} = [\delta Y \;\; \delta X \;\; \delta\psi \;\; S_R \;\; S_L \;\; \delta D]^T$$

$$\mathbf{X}_{k+1} = \mathbf{\Phi}_k \mathbf{X}_k + \mathbf{w}_k$$

$$\mathbf{\Phi}_k = \begin{bmatrix} 1 & 0 & -\frac{R+L}{2}\sin\psi_k & \frac{R}{2}\cos\psi_k & \frac{L}{2}\cos\psi_k & 0 \\ 0 & 1 & \frac{R+L}{2}\cos\psi_k & \frac{R}{2}\sin\psi_k & \frac{L}{2}\sin\psi_k & 0 \\ 0 & 0 & 1 & -\frac{R}{D} & \frac{L}{D} & \frac{R-L}{D^2} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

3.2 Measurement Model

A vision sensor provides sequential images. Feature points are selected and tracked through these images. The feature point position is not changed on the navigation frame. So a measured feature point position is used for reference, when the mobile robot position is estimated.

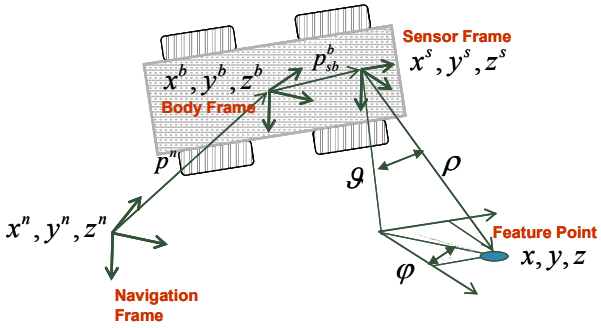

Fig. 5. Coordinate of vision based SLAM

Figure 5 shows the relation between the feature point and frames. The reference frame is a navigation frame; however, the feature point is measured on the sensor frame, so a coordinate transform is needed.
Equation (7) denotes a feature point position on the sensor frame. Using equation (6) and (7), the feature point position can be predicted.

$$x_{FP}^{s} = C_{b}^{s}\left(C_{n}^{b}\left(x_{FP}^{n} - p_{sb}^{b}\right) - p^{n}\right) \tag{7}$$

C_{b}^{s} : Coordinate transform (body to sensor frame)

C_{n}^{b} : Coordinate transform (navigation to body frame)

p_{sb}^{b} : Lever arm (between sensor and body frame)

p^{n} : Lever arm (between body and navigation frame)

Equation (8) is the measurement model. Measurement vector contains bearing and elevation angle, which are readily calculated from the feature point position information.

$$z = \begin{bmatrix} \tan^{-1}\left(\dfrac{y}{x}\right) \\ \tan^{-1}\left(\dfrac{z}{\sqrt{x^2 + y^2}}\right) \end{bmatrix}$$
$$x_{fi}^{s} = \begin{bmatrix} x \\ y \\ z \end{bmatrix},\ z = \begin{bmatrix} \phi \\ \theta \end{bmatrix} \qquad \begin{pmatrix} \phi : \text{Feature Point Bearing} \\ \theta : \text{Feature Point Elevation} \end{pmatrix} \tag{8}$$

4. Integration Filter

4.1 Particle Filter

In the filter system implementation, the measurement model of a vision sensor is observed to be nonlinear, which requires for a nonlinear filter approach. Extended Kalman Filter, Unscented Kalman Filter (UKF) and Gaussian Sum Filter are common nonlinear filter (Ristic et al., 2004). In this chapter, a particle filter (PF) is used for state estimation, which is a technique for implementing a recursive Bayesian filter by Monte Carlo simulation. The key point of PF is to represent the posterior density function by a set of random samples with its associated weight and to compute estimates based on these samples and weights.

It can be shown that, at the time instant k, the particles $\left\{x_{k|k-1}(i) : i = 1, 2 \ldots, N\right\}$ and $\left\{x_{k}(i) : i = 1, 2 \ldots, N\right\}$ can be recursively obtained by the following algorithm (Ristic et al., 2004):

1. Assume that there is a set of random samples (i.e. particles) $\{x_{k-1}(i) : i = 1, 2, \ldots, N\}$ from the pdf $p(x_{k-1} \mid \mathrm{Y}_{k-1})$.
2. Prediction: Sample N values $\{w_{k-1}(i) : i = 1, 2, \ldots, N\}$ from the pdf of system noise w_{k-1}. Use these samples to generate new swarm of points $\{x_{k|k-1}(i) : i = 1, 2, \ldots, N\}$ which approximates the predicted pdf $p(x_k \mid \mathrm{Y}_{k-1})$ where
$$x_{k|k-1}(i) = f_{k-1}(x_{k-1}(i), w_{k-1}(i))$$
3. Update: Assign each $x_{k|k-1}(i)$ a weight $\mathrm{w}_k(i)$ for $i = 1, 2, \ldots, N$, after measurement y_k is received. The weights are given by
$$\mathrm{w}_k(i) = \frac{p(\mathrm{y}_k \mid x_{k|k-1}(i))}{\sum_{\mathrm{j}=1}^{N} p(\mathrm{y}_k \mid x_{k|k-1}(j))}$$
This defines a discrete distribution over $\{x_{k|k-1}(i) : i = 1, 2, \ldots, N\}$, which assigns probability mass $\mathrm{w}_k(i)$ to the element $x_{k|k-1}(i)$ and results in the posterior pdf $p(x_k \mid \mathrm{Y}_k)$ being represented in terms of weighted samples (i.e. particles).
4. Resample: Resample independently N times from the above discrete distribution. The resulting particles $\{x_k(i) : i = 1, 2, \ldots, N\}$ which satisfies
$$\Pr\{x_k(i) = x_{k|k-1}(j)\} = \mathrm{w}_k(j)$$
form an appropriate sample (with equal weight to each element) from the posterior pdf $p(x_k \mid \mathrm{Y}_k)$.
5. The prediction, update and resample step form a single iteration and is recursively applied each time k.

4.2 Integration Filter

The following figure shows the structure of a vision based SLAM. An indirect filter structure is used to reduce computation time and power.

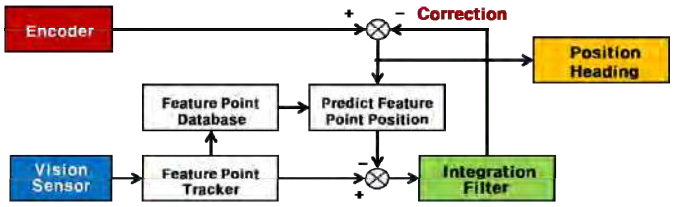

Fig. 6. Structure of vision based SLAM

In the encoder block, the position is locally calculated using large sensor errors. In the vision sensor block, the image data is acquired and feature points are selected. Then, the feature points are tracked via feature point tracker block, where the tracking information is stored in the database and sent to integration filter block simultaneously. Using encoder measurements and feature point database, the next feature point position can be predicted on the image frame. In the integration filter block, the measured feature point position is compared with the predicted position, and navigation errors are estimated. Estimated errors go to the encoder block, and finally compensate the navigation data.
Figure 7 shows the structure of the distributed filter which has several local filters. The distributed filter has good performance with respect to computation time and power, fault detection, isolation, variation in measurements number, etc. The number of local filters is varying in proportion to the number of feature points, which depends on visible environment. The estimated errors from each local filter are combined in the master filter for data integration. Finally, the estimated errors are fed into an encoder to compensate for the position data. In the chapter, error estimate and covariance of master filter is not fed into the local filters for simplicity.

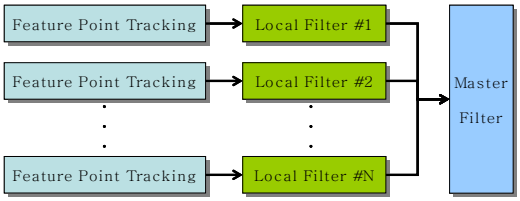

Fig. 7. Structure of distributed filter

5. Simulation Results

Computer simulation is done to demonstrate the performance of the suggested algorithm. For simulation, 10m x 10m square track and 10 feature points are used. The time step for each epoch is set to 0.01sec with total epoch number of 4790. A detailed description is shown in Table 1.

Filter	Particle filter (number of particle: 200)
Wheel radius	50 ± 1 mm
Encoder error	± 6.3 mm/rotation
Velocity	2 m/sec
Track	10m x 10m
Feature points	200 in 40m x 40m
Simulation frequency	Positioning: 100 Hz Error Compensation: 10 Hz

Table 1. Simulation paraeters

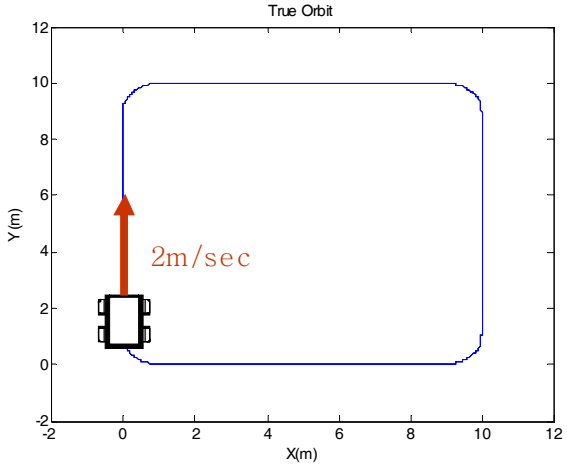

Fig. 8. Simulation track

The navigation performance of a vision based SLAM integrated system is compared together with that of a standalone encoder system. First, Figure 9 is the simulation result showing the navigation performance of an encoder only system. The red line denotes the true trajectory and the blue line denotes the estimated trajectory. As time increases, cumulative errors are increased, which results in an unbounded deviation from the true track. Error characteristics illustrates that the dominant error source of the encoder is scale factor and the distance between two wheels.

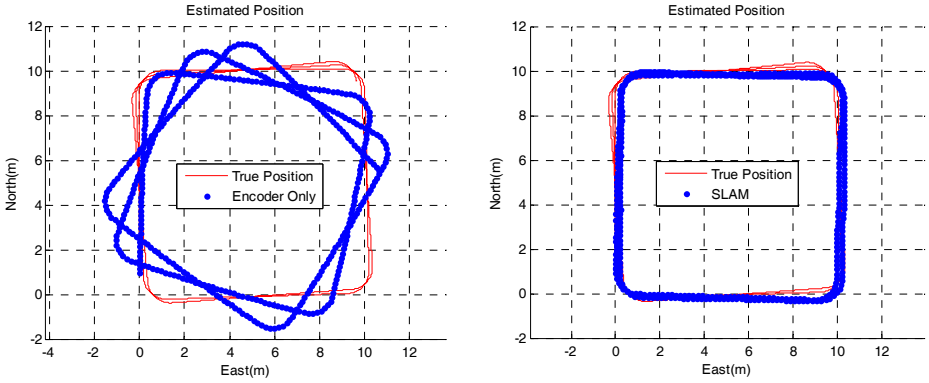

Fig. 9. Estimated position of encoder only navigation (left) and vision based SLAM (right)

In the vision based SLAM case, the navigation performance is greatly enhanced. The estimated position has a bounded error from the true trajectory, which does not diverge as time increases. Throughout the trajectory, properly deployed feature points compensate error accumulation from the encoder and the initial heading bias produced a slightly deviated square trajectory.

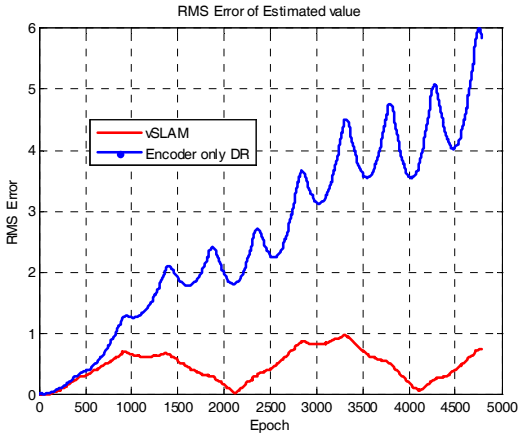

Fig. 10. Position error comparison between encoder only navigation and vision based SLAM

Figure 10 shows the position error of the two methods. In case of the encoder, the error diverges with sinusoid-like fluctuation. The fluctuation comes from the encoder output scale factor error at every turn. The estimation error of the SLAM aided integration system is error bound by about 1m.

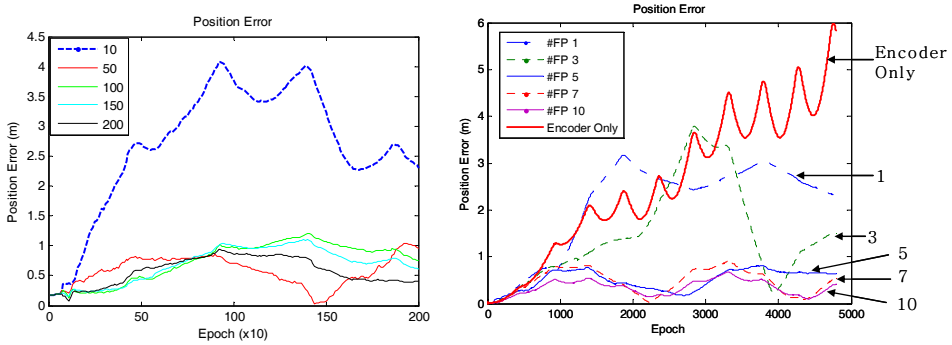

Fig. 11. Position error through changing the number of particles (left) and feature points (right)

In implementing the navigation computer, the computing time highly depends on the particle number and the feature point number, which constrains the particle size in each particle filter. Figure 11 shows position error when the numbers of particles on filter changes. The number of feature points is fixed as 10. When there are particles of more than 50, the error is below 1.5m during operation times. The number of feature points is changed from 1 to 10, and the number of particles is fixed at 100. In the figure, even though #FP 1 and 3 cases have large error boundaries, the position error does not diverge and shows better estimation performance than the encoder only system. The maximum position error is observed to be smaller than 1m when there are more than 5 feature points. Table 2 summarizes the data at the epoch 1000, 2000, 3000 and 4000 in Figure 11.

Epoch / #FP	1000	2000	3000	4000	Mean	Maximum Error
Encoder Only	1.26	1.98	3.14	3.56	2.58	5.99
1	0.81	3.00	2.49	2.89	2.12	3.17
3	0.82	1.41	3.50	0.31	1.49	3.79
5	0.71	0.38	0.40	0.68	0.50	0.81
7	0.77	0.27	0.73	0.33	0.48	0.90
10	0.47	0.28	0.48	0.30	0.34	0.67

Table 2. Position error through changing the number of feature points

In #FP 1 and 3 cases, the position error is larger than the other cases at early stage, thus these large errors affect all over the stage, so estimated trajectory become inaccurate. In other cases, the error is decreased and has boundary. Compared with the mean error of the encoder only case at 4000 epoch, the #FP 7 case has 82% smaller error and in the #FP 10 case has an 87 % smaller error than the encoder only case. In conclusion, when the number of particles is set to 100 for low computational burden, the number of particles over 5 is required for better performance.

For further verification of suggested algorithm, the simulation condition is changed. A difference in simulation is an angle of the vision sensor. Figure 12 (left) illustrates attached angle of vision sensor. The angle is changed from 0 to 180 degree.

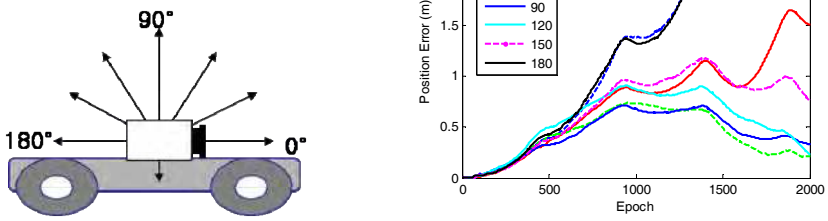

Fig. 12. Attached angle of a vision sensor and position error through the changing of attached angle

Figure 12 (right) shows the position error through the changing of the angle. The error is below 1m when the angle is changed from 60 to 120, whereas the other cases have larger errors.

In other words, if the vision sensor looks toward the top direction (90 degree), the feature point position of the horizontal direction is detected and measured more precisely. So the quality of measurement is better than the other cases. For this reason, the horizontal error can be decreased. Since the angle is far from 90 degree, the quality of measurements on horizontal direction becomes poor. That affects the position and heading error.

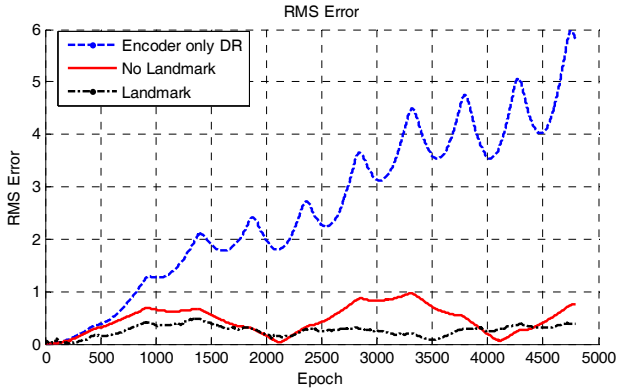

Fig. 13. Position error in case of using the landmark

If there are some points which are known positions with previous information, the navigation performance can be increased. In this chapter, such a point is called the 'Landmark'. The landmark has precise position information, so it is used for accurate measurements when it is observed. In the integration filter, feature point estimation step will be skipped because the position is already known. In Figure 13, landmark aids position estimation performance to be increased compare with general case (without landmark). The landmark aiding case has 0.2567m RMS error (mean), which is smaller than 0.4656m in general cases. This means that precise measurements (landmark) prevent a divergence of estimation error sand improve the estimation performance in mobile robot navigation.

6. Conclusion

In this chapter, a vision based SLAM and encoder integrated system is presented for mobile robot navigation. By considering the nonlinear measurement model and feature point availability around the trajectory, a distributed particle filter approach is applied. Simulation results demonstrate the performance of the implemented mobile robot. Further results confirm that the estimation performance largely depends on the number of feature points and particles, which will be mutually associated while implementing the embedded navigation computer. It also depends on the attached angle of a vision sensor and the landmark.

7. References

Carlo Tomasi and Takeo Kanade, “Detection and Tracking of Point Features”, *Technical Report CMU-CS-91-132*, April 1991

Edward Rosten and Tom Drummond, "Fusing Points and Lines for High Performance Tracking", *IEEE Int. Conf. Computer Visio*n, October 2005, pp.1508-1511.

Hugh Durrant-Whyte and Tim Bailey, “Simultaneous Localization and Mapping”, *IEEE Robotics & Automation Magazine*, June 2006, pp.99-117

J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and Localization for an autonomous mobile robot," *IEEE Int. Wkshp. Intell. Robots Syst.*, Osaka, Japan, vol. 3, November 1991, pp. 1442-1447.

Jing Wu and Hong Zhang, "Camera Sensor Model for Visual SLAM", *IEEE. Conf, Computer and Robot Vision*, May 2007, pp.149-156

Konstantinos G. Derpanis, " The Harris Corner Detector", October 2004.

Philip J. Schneider and David H. Eberly, *Geometric Tools for Computer Graphics*, Morgan Kaufmann Publishers, 2003, pp.400-433

Rafael C. Gonzalez, *Digital Image Processing using MATALB*, Prentice Hall, 2004, pp.26-27

Ristic, Arulman and Gordon, *Beyond the Kalnman Filter*, Artech House, 2004, pp.35-48

S. M. Smith and J. M. Brady, "SUSAN - a new approach to low level image processing," *International Journal of Computer Vision*, May 1997.pp.45-78.

Sheng Fu, Hui-ying Liu, Lu-fang Gao and Yu-xian Gai, "SLAM for Mobile Robots Using Laser Range Finder and Monocular Vision", *IEEE Int, Conf. Mechatronics and Machine Vision in Practice*, December 2007, pp.91-96

Sukhan Lee and Seongsoo Lee, "Recursive Particle Filter with geometric Constraints for SLAM", *IEEE Int. Conf, Multisensor Fusion and Integration for Intelligent Systems*, 2006, pp.359-401

Mobile Robots Navigation

Edited by Alejandra Barrera

ISBN 978-953-307-076-6
Hard cover, 666 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

Mobile robots navigation includes different interrelated activities: (i) perception, as obtaining and interpreting sensory information; (ii) exploration, as the strategy that guides the robot to select the next direction to go; (iii) mapping, involving the construction of a spatial representation by using the sensory information perceived; (iv) localization, as the strategy to estimate the robot position within the spatial map; (v) path planning, as the strategy to find a path towards a goal location being optimal or not; and (vi) path execution, where motor actions are determined and adapted to environmental changes. The book addresses those activities by integrating results from the research work of several authors all over the world. Research cases are documented in 32 chapters organized within 7 categories next described.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821