A MAP Approach for Vision-based Self-localization of Mobile Robot

WANG Ke[1] WANG Wei[1] ZHUANG Yan[1]

Abstract An on-the-fly, self-localization system is developed for mobile robot which is operative in a 3D environment with elaborative 3D landmarks. The robot estimates its pose recursively through a MAP estimator that incorporates the information collected from odometry and unidirectional camera. We build the nonlinear models for these two sensors and maintain that the uncertainty manipulation of robot motion and inaccurate sensor measurements should be embedded and tracked throughout our system. We describe the uncertainty framework in a probabilistic geometry viewpoint and use unscented transform to propagate the uncertainty, which undergoes the given nonlinear functions. Considering the processing power of our robot, image features are extracted in the vicinity of corresponding projected features. In addition, data associations are evaluated by statistical distance. Finally, a series of systematic experiments are conducted to prove the reliable and accurate performance of our system.

Key words Vision-based self-localization, MAP estimation, multi sensor fusion, unscented transformation, uncertainty propagation

In retrospect of self-localization, many vision-based self-localization systems have been developed for mobile robot[1]. One choice is based on geometric reasoning methods, [2−3] applied Hough transform to global localization. However, they are sensitive to sensor noise and may not provide an accurate pose estimate. Another framework is image retrieval system, and researchers have successfully localized the robot based on the appearance[4−6] of the environment or SVM method[7]. The drawback is that a large database has to be kept in robot memory for classification.

Experiments indicate that a reliable localization system should explicitly represent and maintain the uncertainty inherent in the available information[8−9]. Therefore, localization system could be designed in a probabilistic manner. [10−14] summarize some popular probabilistic methods in which Markov[15], Mont Carlo[16], and particle filter[17−18] hold the probability likelihood to represent the robot state belief. However, calculation of the Bayesian integration requires a huge computation, which may prevent their applications to robot vision. Another choice is extended Kalman filter[19−21]. This approach takes a linearization step with explicitly analytical Jacobian matrices. As known, in some cases, the matrices are difficult to obtain due to the complicated nonlinear transform. Moreover, the linearization might lead to the filter instability if the time step intervals are not sufficiently small[22].

In this paper, a vision-based self-localization system is developed based on our SmartROB2 mobile robot. Our design pattern focuses on the following issues:

Problem 1. How to design a vision-based self-localization system in a new viewpoint? This system can provide an accurate pose estimate with a more reasonable and efficient uncertainty manipulation.

Problem 2. Considering run-time coordinations among subprograms of the localization task, how to make use of the resource-constrained RTOS of an embedded robot platform?

The new standpoint we emphasis on is that we do not describe the design of self-localization method as a traditional state-estimate problem but as a statistical parameter estimate problem. That is, given a set of training classes previously constructed from 3D environmental information, there exists an unknown random vector (robot pose) evolved by a certain dynamic function with prior PDF. When the vector travels in the parameter space, some conditional events will generate from certain PDF of the given classes. Therefore, the robot could integrate these available events containing the external information to estimate the a posterior probability likelihood of such unknown vector. We choose MAP[23] (Maximum a posterior) method, which is computationally direct and has experimentally been shown to work well[24].

We consider the uncertainty inherent in the localization process and give explicitly analytical solution using unscented transform (UT)[25]. The UT applies the nonlinear function to a minimal set of sample points and yields more accurate statistical value than linear methods. Therefore, information and its uncertainty can be propagated through nonlinear transform without calculating Jacobian matrices.

Because of the limited resources, the system is developed based on concurrently hierarchical frame instead of inefficient serial processing mode. The former fulfills on-the-fly localization when there are several collaborative subprograms in localization task. Moreover, we carefully design an effective map using line segment with directional information to improve feature detection and data association.

This paper is organized as follows. We describe firstly the map modeling, self-localization algorithm and logic framework of our system. In Section 1, scene prediction is presented based on the camera model. Section 2 expatiates the manipulations of uncertainty related to our system, and how the uncertainty assists the image processing and data association. Finally, the experimental results are given.

1 Vision-based self-localization

1.1 Map modeling

As shown in Fig. 1, in the world coordinate the global map M_W consists of a set of 3D line segments that describe the line features in a whole soccer field. The primitive element, $m_W^i \in M_W$, is defined as a directional line segment, i.e.,

$$m_W^i = \{E_W^i(\boldsymbol{p}_W^s, \boldsymbol{p}_W^e), V_W^i(\boldsymbol{v}_W^i, \boldsymbol{n}_W^i)\}, i = 1, 2, \cdots, N \quad (1)$$

where $\boldsymbol{p}_W^s = [x_W^s, y_W^s, z_W^s]^{\mathrm{T}}$ and $\boldsymbol{p}_W^e$ represent two endpoints of m_W^i, $\boldsymbol{v}_W^i = [a_W^i, b_W^i, c_W^i]$ is the directional vector of m_W^i, $\boldsymbol{n}_W^i$ describes the physical color transitions and is orthogonal to $\boldsymbol{v}_W^i$.

Received May 24, 2006; in revised form September 24, 2007
Supported by National Natural Science Foundation of China (60605023, 60775048), Specialized Research Fund for the Doctoral Program of Higher Education (20060141006)
1. Research Center of Information and Control, Dalian University of Technology, Dalian 116024, P. R. China
DOI: 10.3724/SP.J.1004.2008.00159

Compared with the topological map[26] or the map that is defined only by 3D points[27], our map can provide abundant information and is more efficient in manipulation.

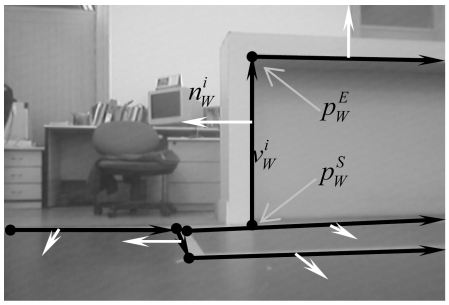


Fig. 1 Global 3D map with directional information

1.2 Self-localization algorithm

From a statistical viewpoint, the robot pose $\boldsymbol{\varphi} = [x, y, \theta]^{\mathrm{T}}$ is regarded as an unknown parameter. We suppose this parameter is driven by a Gaussian process

$$\boldsymbol{\varphi}_{k+1|k} = f(\boldsymbol{\varphi}_k, \boldsymbol{u}_{k+1}) + \boldsymbol{\nu}_{k+1} \tag{2}$$

where $f(\boldsymbol{\varphi}, \boldsymbol{u})$ is the dead-reckoning function[28], which is used to model the differential odometer of our robot; $\boldsymbol{u}$ is input and $\boldsymbol{\nu}$ is additive Gaussian white noise. The associated covariance $P_{k+1|k}$ of prior estimate $\boldsymbol{\varphi}_{k+1|k}$ is computed by unscented transform.

We assume $\boldsymbol{\varphi}$ obey the distribution of the a priori Gaussian PDF, which evaluates the statistical distance between $\boldsymbol{\varphi}$ and the prior estimate $\boldsymbol{\varphi}_{k+1|k}$, i.e.,

$$\boldsymbol{\varphi} \sim \mathcal{N}(\boldsymbol{\varphi}_{k+1|k}, P_{k+1|k}) \tag{3}$$

Estimation of $\boldsymbol{\varphi}$ also depends on the information $D_n = \{d_1, d_2, \cdots, d_n\}, n < N$ available during process (2). Sample d_i evaluates the Mahalanobis distance between measurement $\boldsymbol{Y}_j$ and its corresponding feature prediction $\boldsymbol{Y}_i$. $\boldsymbol{Y}_i$ is determined by the camera model, $\boldsymbol{\varphi}$ and map element m_W^i as follows

$$\boldsymbol{Y}_i = \hbar(\boldsymbol{\varphi}, m_w^i) + \boldsymbol{\xi} \tag{4}$$

where $\boldsymbol{\xi} \sim \mathcal{N}(0, R)$ is the additive noise. We will detail (4) later in Section 2.

Assume that each observation sample d_i generates independently from a joint parametric conditional PDF $p(D_n|\boldsymbol{\varphi})$

$$p(D_n|\boldsymbol{\varphi}) = \prod p(d_i|\boldsymbol{\varphi}) \tag{5}$$

The problem is then viewed as searching the optimal parameter $\hat{\boldsymbol{\varphi}}$, which maximizes the posterior probability $p(\boldsymbol{\varphi}|D_n)$ based on the MAP estimator, i.e.,

$$\begin{aligned} \hat{\boldsymbol{\varphi}} &= \arg\max_{\varphi}(p(\boldsymbol{\varphi}|D_n)) = \\ &\arg\max_{\varphi}(p(\boldsymbol{\varphi})p(D_n|\boldsymbol{\varphi})) \end{aligned} \tag{6}$$

Transform (6) into an equivalent optimization form

$$\hat{\boldsymbol{\varphi}} = \arg min_{\boldsymbol{\varphi}}(-2\ln(p(\boldsymbol{\varphi})) - 2\ln(p(D_n|\boldsymbol{\varphi}))) \tag{7}$$

where the first term $-2\ln(p(\boldsymbol{\varphi}))$ is

$$-2\ln(p(\boldsymbol{\varphi})) = (\boldsymbol{\varphi} - \boldsymbol{\varphi}_{k+1|k})^{\mathrm{T}} P_{k+1|k}^{-1} (\boldsymbol{\varphi} - \boldsymbol{\varphi}_{k+1|k}) + \mathcal{P} \tag{8}$$

where variable $\mathcal{P} = \ln|8\pi^3 P_{k+1|k}|$. After optimizing (7), the covariance matrix $\hat{P}$ of $\boldsymbol{\varphi}$ can be estimated according to the regression analysis proposed by [29].

$$\hat{P} = 2H^{-1} \tag{9}$$

where matrix H represents the Hessian of the objective function. The robot pose and covariance can be updated by fusing the information of odometry and visual observations.

1.3 System architecture

Here, we summarize the software framework of our self-localization system for SmartROB2 robot. The self-localization task is developed by Oberon system and implemented on native XO/2 RTOS[30]. The system can be logically divided into three layers, and each layer implements as a concurrent and collaborative subtask.

1) Perception layer (PL): In this underlying layer, a serving thread responds to detect of the sensors′ states, allocating sensor buffers, and online correcting the image distortion caused by the fish-eye camera. Another thread also keeps track of its communication between information processing layer (IPL) and receives sate feedback from data fusion layer (DFL).

2) Information processing layer: The received information from PL unfolds into image data, robot pose $\boldsymbol{\varphi}_{k+1|k}$, and its covariance $P_{k+1|k}$. Image processing unit (IPU) then use the initial scene estimated by $\boldsymbol{\varphi}_{k+1|k}$ and 3D map M_W to extract the corresponding image features.

3) Data Fusion Layer: As system kernel, it manipulates the information from each layer and uses MAP estimator to perform multi-sensor fusion for the a posterior pose and its covariance.

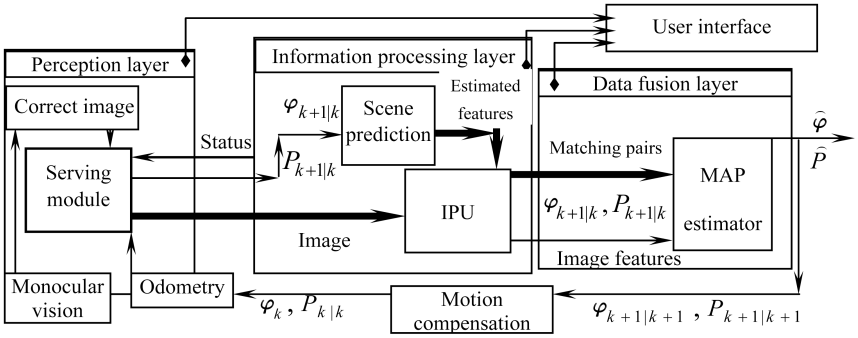


Fig. 2 System frame

Finally, our system adopts the motion compensation[31−32] procedure to compensate the robot movement for the system-processing delay.

2 Scene prediction

When camera parameters and $\boldsymbol{\varphi}$ are given, the robot can predict where in a captured image the map lines should be visible.

2.1 Generating model point

As shown in Fig. 3, we assume that camera optical center O_C and Z_R axes of robot are collinear, α is the pan angle, x_C axis is parallel to the ground. Function $proj_p$ projects a random point $\boldsymbol{p}_W^j = [x_W^j, y_W^j, z_W^j]^{\mathrm{T}}$ of m_W^i to the point $\boldsymbol{p}_I^j$ in the image plane:

$$\boldsymbol{p}_I^j = \mathrm{proj}_p(\boldsymbol{\varphi}, \boldsymbol{p}_W^j) \tag{10}$$

This function firstly transforms $\boldsymbol{p}_W^j$ to $\boldsymbol{p}_R^j$ in robot coordinates using $\boldsymbol{\varphi}$, then to $\boldsymbol{p}_C^j = [x_C^j, y_C^j, z_C^j]^{\mathrm{T}}$ in camera coordinates. If the parameters of the calibrated camera are given, the image point $\boldsymbol{p}_I^j$ can be computed. Notice that $\boldsymbol{p}_I^j$ may be defined by $\boldsymbol{p}_C^j$ if focal length f is provided, i.e.,

$$\boldsymbol{p}_{I/C}^j = [x_C^j f/z_C^j, y_C^j f/z_C^j, -f]^{\mathrm{T}} \tag{11}$$

where $\boldsymbol{p}_{I/C}^j$ denotes the point definition of camera coordinates for $\boldsymbol{p}_I^j$ in the image plane.

2.2 Generating model vector

Let line l_I^i be the projected image line of m_W^i. l_I^i is defined by $\boldsymbol{p}_I^j = [x_I^j, y_I^j, 0]^{\mathrm{T}}$ and vector $\boldsymbol{v}_I^i = [a_I^i, b_I^i, 0]^{\mathrm{T}}$, which can also be represented as $\boldsymbol{v}_{I/C}^i = [a_I^i, b_I^i, 0]^{\mathrm{T}}$ in camera coordinates. $\boldsymbol{v}_W^i \in m_W^i$ will be sequentially transformed to $\boldsymbol{v}_C^i = [a_C^i, b_C^i, c_C^i]^{\mathrm{T}}$, and then into $\boldsymbol{v}_I^i$. Because $\boldsymbol{v}_C^i$ lies in the interpretation plane[33] passing through O_C and l_I^i, the normal vector $\boldsymbol{N}_{I/C}^i$ of this plane is equal to the cross product of $\boldsymbol{v}_{I/C}^i$ and the vector defined by O_C and $\boldsymbol{p}_I^j$, i.e.,

$$\boldsymbol{N}_{I/C}^i = [b_I^i f, -a_I^i f,\ a_I^i y_I^j - b_I^i x_I^j] \tag{12}$$

The following equation verifies the orthogonality of $\boldsymbol{N}_{I/C}^i$ with respect to $\boldsymbol{v}_C^i$.

$$a_C^i b_I^i - b_C^i a_I^i + c_C^i (a_I^i y_I^j - b_I^i x_I^j)/f = 0 \tag{13}$$

Substituting (11) into (13), we obtain the formulation describing the relationship between $\boldsymbol{v}_I^i$ and $\boldsymbol{v}_C^i$ as follows.

$$a_I^i (b_C^i z_C^j - c_C^i y_C^j) = b_I^i (a_C^i z_C^j - c_C^i x_C^j) \tag{14}$$

After some functional decompositions and variable substitutions, the projected $\boldsymbol{v}_I^i$ can be formulated as

$$\begin{bmatrix} a_I^i \\ b_I^i \end{bmatrix} = \begin{bmatrix} (y_W^j - y)(a_W^i \cos\alpha + c_W^i \sin\alpha\cos\theta) - \\ (x_W^j - x)(b_W^i \cos\alpha + c_W^i \sin\alpha\sin\theta) + \\ \sin\alpha(H - z_W^j)(b_W^i \cos\theta - a_W^i \sin\theta) \\ \\ -c_W^i((x_W^j - x)\cos\theta + (y^j - y)\sin\theta)) - \\ (H - z_W^j)(a_W^i \cos\theta + b_W^i \sin\theta) \end{bmatrix} \tag{15}$$

where H is the distance between O_C and O_R. Similarly, we can obtain the vector $\boldsymbol{n}_I^i$ corresponding to $\boldsymbol{n}_W^i$.

2.3 Generating scene feature

The scene feature is represented in the Hough domain using the above results as

$$\hat{\boldsymbol{Y}}_i = [\rho, \vartheta]^{\mathrm{T}} = \mathcal{T}(\boldsymbol{p}_I^j, \boldsymbol{v}_I^i) \tag{16}$$

where function $\mathcal{T}$ transforms $\boldsymbol{p}_I^j$ and $\boldsymbol{v}_I^i$ into the line that is defined by ρ and ϑ.

3 Uncertainty manipulations

We describe here the uncertainty manipulation for robot motion, image processing, and data association. As shown in Fig. 3, the uncertainty of robot motion may inevitably disturb the projected feature of line. In this case, both the descriptions and coordinates of uncertainty are changed when the random vector is transformed by (2) and (4). We model these operations in a probability geometric manner[32] and hold the Gaussian assumption owing to the central limit theorem[33].

3.1 Uncertainty propagation

Instead of linearizing (2) and (4) with respect to $\boldsymbol{\varphi}$, we use unscented transform to capture the statistical properties of $\boldsymbol{\varphi} \in \mathbf{R}^{n\times 1}$. The uncertainty propagation related to (2) includes the following steps:

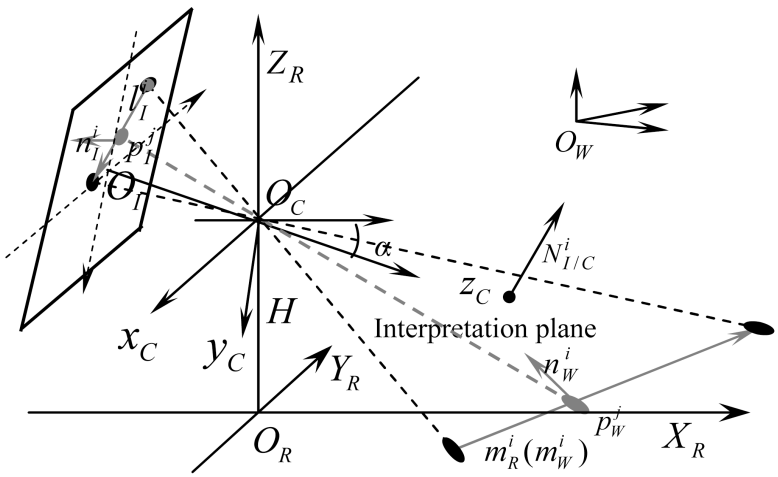


Fig. 3 Scene prediction

- Select sample $\boldsymbol{\sigma}_j$ from jth row or column of matrix $\pm\sqrt{nP_k}$.
- Construct the set of sigma points $X_k = [\boldsymbol{\varphi}_k, \boldsymbol{\varphi}_k + \boldsymbol{\sigma}_1, \cdots, \boldsymbol{\varphi}_k + \boldsymbol{\sigma}_{2n}]$.
- Use (2) to transform each sigma point $\boldsymbol{X}_k^j$ into $\boldsymbol{X}_{k+1}^j$.
- Let $W_j^{(c)}$ be the weighted scalar for jth point, and compute the mean $\boldsymbol{\varphi}_{k+1|k} = \sum_{j=0}^{2n} W_j^{(c)} \boldsymbol{X}_{k+1}^j$ and associated covariance $P_{k+1|k} = \sum_{j=0}^{2n} W_j^{(c)} \left[\boldsymbol{X}_{k+1}^j - \boldsymbol{\varphi}_{k+1|k}\right] \left[\boldsymbol{X}_{k+1}^j - \boldsymbol{\varphi}_{k+1|k}\right]^{\mathrm{T}}$.

For propagating the uncertainty through (4) with respect to $\boldsymbol{\varphi}$, we observe the following rules:

- Select sample $\boldsymbol{\kappa}_j$ from jth row or column of matrix $\pm\sqrt{nP_{k+1|k}}$.
- Construct the set of sigma points $\mathcal{X} = [\boldsymbol{\varphi}, \boldsymbol{\varphi} + \boldsymbol{\kappa}_1, \cdots, \boldsymbol{\varphi} + \boldsymbol{\kappa}_{2n}]$.
- Apply (4) to $\boldsymbol{X}^j$ for the transformed sample point $\hat{\boldsymbol{Y}}_i^j$ of map element m_W^i .
- Let $W_j^{(m)}$ be the weighted scalars for jth point, then predict the scene feature $\hat{\boldsymbol{Y}}_i = \sum_{j=0}^{2n} W_j^{(m)} \hat{\boldsymbol{Y}}_i^j$ and its associated covariance $R_i = \sum_{j=0}^{2n} W_j^{(m)} \left[\hat{\boldsymbol{Y}}_i^j - \hat{\boldsymbol{Y}}_i\right] \left[\hat{\boldsymbol{Y}}_i^j - \hat{\boldsymbol{Y}}_i\right]^{\mathrm{T}}$.

3.2 Image processing and data association

We just process the gray images to reduce the data transfer on bus and computation cost. In Fig. 4(a), the visible m_W^i is projected to its corresponding feature $\hat{\boldsymbol{Y}}_i$. Its covariance $\Sigma_{\rho\theta}$ with respect $\boldsymbol{\varphi}$ to is calculated by UT. We then use the search strategy within the area determined by $\boldsymbol{\sigma}_\rho$. Generally, there is a slight difference between $\hat{\boldsymbol{Y}}_i$ and the corresponding image edge $\boldsymbol{Y}_j$. The searching algorithms start from a random point on the given line, and then each pixel along direction $\boldsymbol{n}_I^i$ is convoluted to obtain the magnitude and direction of gradient. Once the magnitude is above a certain threshold and the gradient is opposite to vector $\boldsymbol{n}_I^i$, the pixel is regarded as a candidate edge pixel. Search algorithms then change to the gradient direction (DG) and terminate until the gradient magnitude is below the given threshold. Thus, real edge pixels are determined by selecting the one with the maximal magnitude from candidates. When two points are found, the corresponding image edge $\boldsymbol{Y}_j$ can be given. Fig. 4 (b) illustrates the data association results using our image processing method.

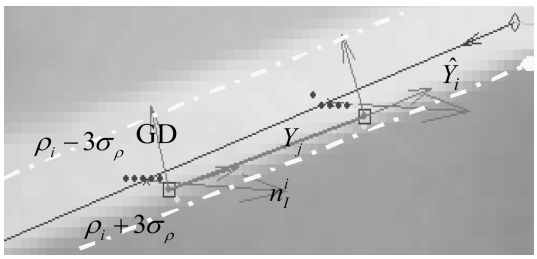
The direction of a detected edge is ϕ, a reasonable uncertainty description[34−35] for the edge pixel is given as follows

$$\Lambda = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma}_1^2 & 0 \\ 0 & \boldsymbol{\sigma}_2^2 \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \tag{17}$$

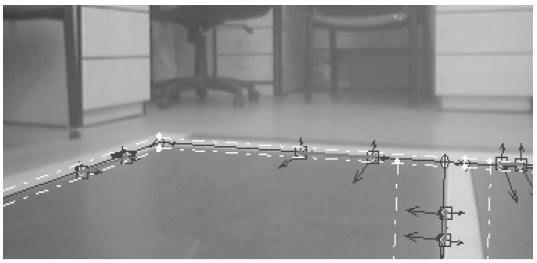
where $\boldsymbol{\sigma}_1 > \boldsymbol{\sigma}_2$, $\boldsymbol{\sigma}_1$ is the standard variance along ϕ. $\boldsymbol{\sigma}_2$ is the standard variance along the normal of this edge[36]. The uncertainty R_j for the extracted edge is computed by the transformation:

$$R_j \leftarrow \mathbb{R}(\Lambda_j^m, \Lambda_j^n) \tag{18}$$

where Λ_j^m and Λ_j^n correspond to uncertainties of the detected edge pixels.



(a) Feature detection



(b) Data association results

Fig. 4 Image processing and data association with uncertainty and local feature constraints

4 MAP estimation

Sample d_i is formulated as the statistical distance between $\hat{\boldsymbol{Y}}_i$ and $\boldsymbol{Y}_j$ as follows.

$$d_i = (\hat{\boldsymbol{Y}}_i - \boldsymbol{Y}_j)^{\mathrm{T}} (\boldsymbol{R}_i + \boldsymbol{R}_j)^{-1} (\hat{\boldsymbol{Y}}_i - \boldsymbol{Y}_j) \tag{19}$$

where $\boldsymbol{R}_i$ is covariance for $\hat{\boldsymbol{Y}}_i$ and $\boldsymbol{R}_j$ for the corresponding image edge $\boldsymbol{Y}_j$.

Let $C_{i,j} = \boldsymbol{R}_i + \boldsymbol{R}_j$. Then the likelihood of the independent information that is available to robot is

$$p(D_n|\boldsymbol{\varphi}) = \prod_{(i,j)} \frac{1}{2\pi\sqrt{|C_{i,j}|}} \exp(-\frac{1}{2} d_i) \tag{20}$$

The second term of (7) is

$$-2\ln(p(D_n|\boldsymbol{\varphi})) = \sum d_i + \sum \ln(|4\pi^3 C_{i,j}|) \tag{21}$$

MAP estimation leads to an optimization problem, which may be approached with various numerical techniques. In this paper, Newton method[37] is used to estimate the parameter $\boldsymbol{\varphi}$ by minimizing the objective function. As shown in Fig. 5, the algorithm is initialized by $\boldsymbol{\varphi}_{k+1|k}$ and covariance $P_{k+1|k}$. During the iteration, the estimated $\boldsymbol{\varphi}$ generates new projected lines (dash dotted lines), which tend to approach the corresponding edges. Because the deviations between prediction and observation features are generally small, the algorithm may perform in the vicinity of the optimal state and rapidly converge to the minimum of objective function. In a systematic viewpoint, our method folds each individual observation and fuses multisensor information in a batch manner. Compared with sequential EKF[38], this method might avoid the filter divergence caused by the integrating order of different accurate observations.

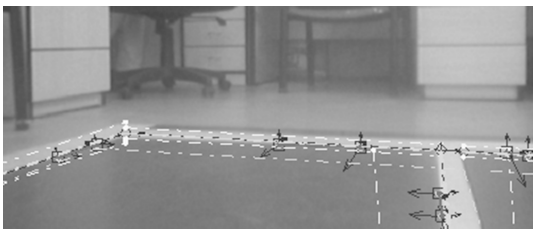


Fig. 5 MAP estimation using iterative optimization

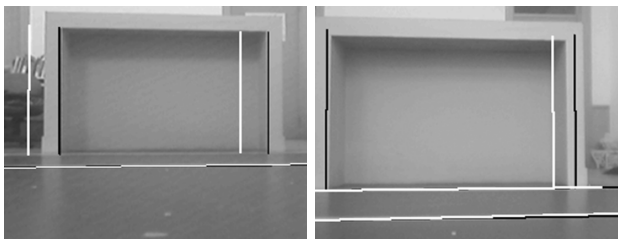
5 Experiments

5.1 Localization performance with single run

The following experiments investigate the accuracy performance of our method in a single run. The experimental analysis is based on a series of performance comparisons between SEKF and our method.

We set up the robot at three different positions. We initialize robot state to the value that is adjacent to the actual position. In addition, the initial state covariance is set to a very high level, which indicates that the belief of robot position is poor. Fig. 6 illustrates the projected model lines generated by SEKF (white) and MAP (black), respectively. In these cases, the resulting model lines estimated by SEKF do not approximate the targets. However, this estimation is substantially improved by MAP.

The accuracy of the estimated pose also depends on the number of features visible and their distances from the robot. The corresponding pose estimations of Fig. 6 are listed in Table 1. In Fig. 6 (a), the visible features are relatively far from the robot. In this case, the estimation of x coordinate is less accurate due to the camera resolution. However, the estimations of θ and y can be compensated by the vertical goalpost. In the areas where precise robot locations are required (Figs. 6 (b) and (c)), there are enough features available for estimation so that a quite accurate pose estimation is possible.



(a) (b)

(c)

Fig. 6 Accuracy comparisons between SEKF and MAP at different locations

Further data analysis on the above localization performance between SEKF and MAP can be seen from Table 1. The comparisons prove our method outweighs SEKF in most cases. Especially, in the case of estimation of θ, MAP can provide better approximate results. This fact indicates that our approach may cope with the rapid scene change caused by the heading uncertainty.

Table 1 Comparisons of localization performance between MAP and SEKF

Position	Method	Real pose (x, y, θ)	Estimated pose (x, y, θ)
(a)	MAP	(0.84,-0.06,-0.087)	(0.807,-0.068,-0.062)
	SEKF		(0.807,-0.092,-0.120)
(b)	MAP	(2.00, 0.00,-0.035)	(1.994,-0.019,-0.048)
	SEKF		(2.035,-0.038,-0.063)
(c)	MAP	(3.27,-0.10, 1.428)	(3.323,-0.125, 1.428)
	SEKF		(3.336,-0.120, 1.419)

5.2 On-the-fly performance

In this section, we conduct an experiment to evaluate on-the-fly localization performance. The principle is that using our method, the robot could have sense of its states and then could go to the given targets.

As shown in Fig. 7, the desired path consists of a set of given targets. We firstly put the robot at $\boldsymbol{\varphi} = [1.95, 0, 0]^{\mathrm{T}}$ while the system state is set to $\boldsymbol{\varphi}_0 = [2.0, 0, 0]^{\mathrm{T}}$. With our method, the robot tries to locate itself and goes to the targets. We illustrate parts of 2D trajectories generated by our method and dead reckoning. The state covariance is represented as error ellipse. Trajectory with black ellipses derives from our method, whereas the trajectory consisting of gray ellipses is exclusively obtained through dead reckoning. Without incorporation of visual information, the robot is unable to recover from the wrong positions where the gray ellipses assemble. In addition, the state covariance continuously diverges, which degrades the robot belief about its position. But, with our method, the robot may correct its position and may approximately reach the given targets.

Another experiment is conducted to evaluate the localization performance between SEKF and MAP. We firstly put robot at starting point $\boldsymbol{\varphi}_0 = [0.8, 0, 0]^{\mathrm{T}}$ and then make it move along a fixed path. At regular intervals, the robot pose and corresponding image are recorded. These data are regarded as reference values. For comparisons, robot starts at real position $\boldsymbol{\varphi}_0$ while the initial state is set to $\boldsymbol{\varphi}'_0 = [0.85, 0.05, 0]^{\mathrm{T}}$. Hence, when robot travels along the original path, the captured image will be the same as the one grabbed at corresponding reference location. Both localization algorithms are applied after 18 frames have been taken. In this case, the robot motion uncertainty has accumulated to a certain degree.

Fig. 8 shows the images captured from frames 19 to 24. The resulting projected lines will require several steps to approximate the corresponding edges. Apparently, the error of y component is not corrected properly by SEKF. The reason for this maybe the convergence rate when the observations are sequentially fused.

Fig. 9 demonstrates the localization results of our method. From frame 20, the resulting predicted features are much closer to its corresponding targets than those of SEKF.

Fig. 10 shows the derivations between reference and corresponding poses estimated by SEKF and MAP, respectively. The estimation of x by MAP approximates to that by SEKF. This may be due to the camera resolution on the depth of field. Map lines away from robot cannot provide accurate references. However, from the estimation of y and θ, we can tell that MAP could generate better estimations than SEKF.

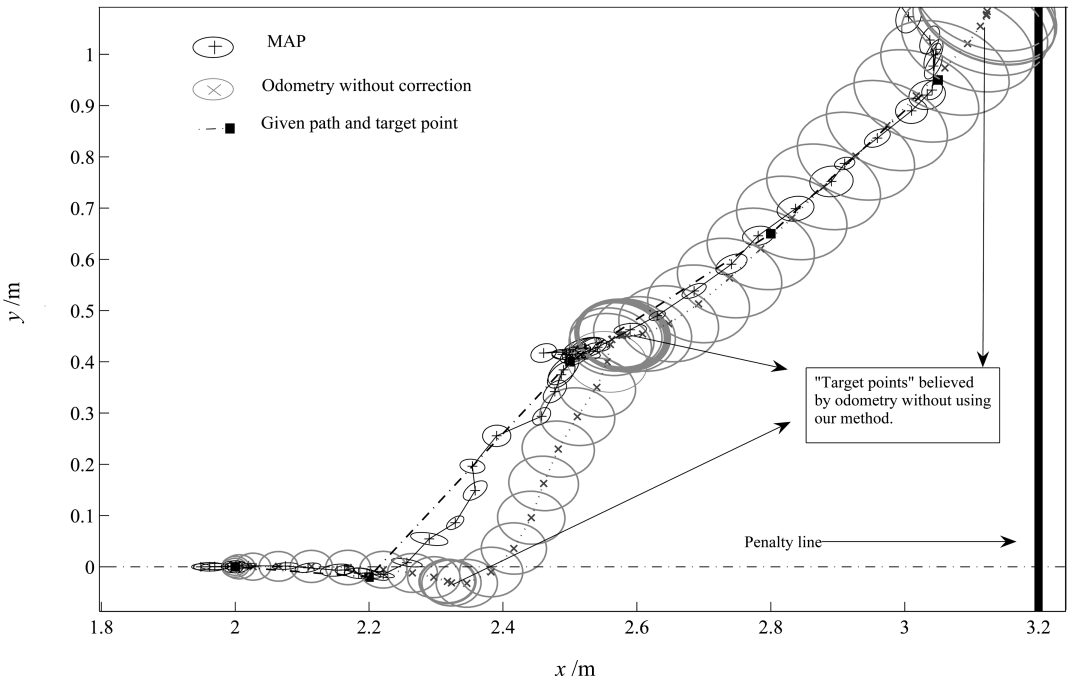
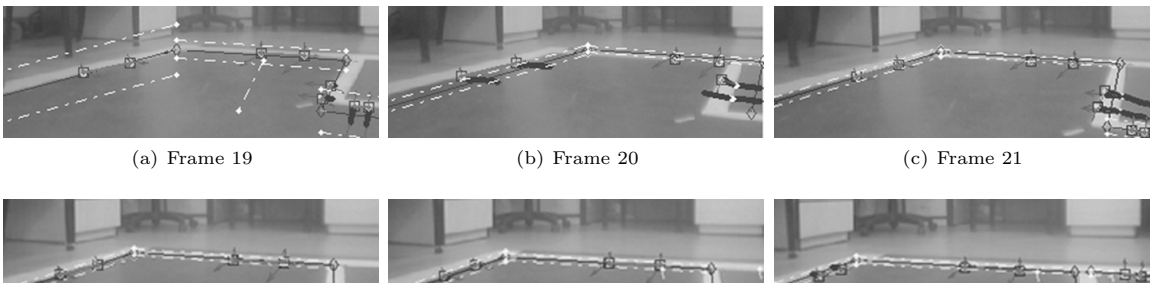


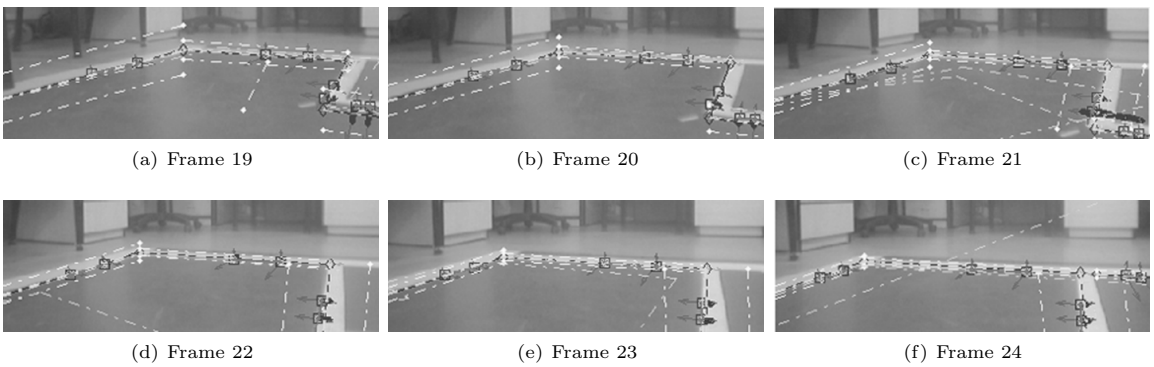
Fig. 7 On-the-fly self-localization, MAP vs. Dead reckoning



(a) Frame 19 (b) Frame 20 (c) Frame 21

(d) Frame 22 (e) Frame 23 (f) Frame 24

Fig. 8 On-the-fly self-localization using SEKF



(a) Frame 19 (b) Frame 20 (c) Frame 21

(d) Frame 22 (e) Frame 23 (f) Frame 24

Fig. 9 On-the-fly self-localization using MAP

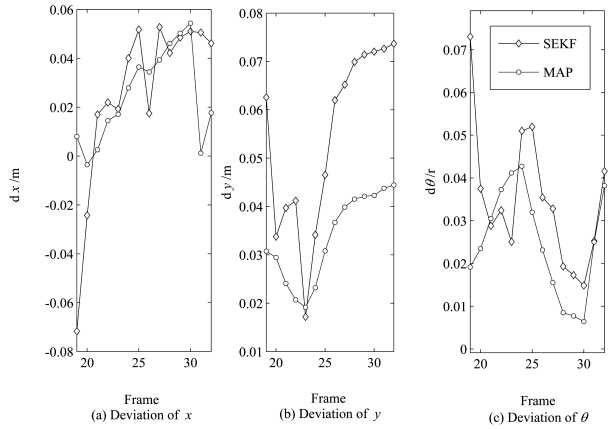


Fig. 10 On-the-fly self-localization performance, MAP vs. SEKF

5.3 Discussion

From above experimental results, we could now summarize the features of our system as follows.

Accuracy: This property originates from the MAP estimator, which uses an iterative scheme to get the solution. If the initial pose is close to the true solution and enough environmental features are available to the robot, our method can converge to the minimum quickly. In addition, the application of unscented enables robot to propagate the state accurately.

Robustness: This performance is due to the manipulation of uncertainty. We provide a reasonable error propagation scheme using unscented transform. All these properties improve the flexibility of our system when dealing with occasional errors. Thus, robot could recover from position disturbance to a certain degree.

6 Conclusion

In this paper, we develop a vision-based self-localization system for our mobile robot in a statistical sense. To fuse the information from camera and odometry, we use MAP estimator to generate relatively better estimate of system state. We developed hierarchical architecture to fulfill on-the-fly self-localization task. This enables us to profoundly understand how the multi-thread and real-time tasks work cooperatively on an embedded RTOS. We consider the uncertainty inherent in the sensor modeling, system state, and measurements. In contrast to the traditional linearization method, we manipulate the random error efficiently by using unscented transform. This generates the accurate mean and associated covariance without explicitly calculating the Jacobeans. Furthermore, we provide a fast-edge detection method and use the statistical Mahalanobis distance for data association. We plan to provide a robust feature extraction method and localization approach, which copes with the ambiguity of data association.

References

1 Guilherme N D, Avinash C K. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, **24**(2): 237−267

2 Iocchi L, Nardi D. Hough localization for mobile robots in polygonal environments. *Robotics and Autonomous Systems*, 2002, **40**(1): 43−58

3 Fang Fang, Ma Xu-Dong, Dai Xian-Zhong. Mobile robot global localization based on model matchingin hough space. *Robot*, 2005, **27**(1): 35−40 (in Chinese)

4 Jogan M, Leonardis A. Robust localization using an omnidirectional appearance-based subspace model of environment. *Robotics and Autonomous Systems*, 2003, **45**(1): 51−72

5 Emanuele M, Takeshi M, Hiroshi I. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 2004, **47**(4): 251−267

6 Linåker F, Ishikawa M. Real-time appearance-based Monte Carlo localization. *Robotics and Autonomous Systems*, 2006, **54**(3): 205−220

7 Li Gui-Zhi, An Cheng-Wan, Yang Guo-Sheng, Tan Min, Tu Xu-Yan. Scene recognition for mobile robot localization. *Robot*, 2005, **27**(2): 123−127 (in Chinese)

8 Thrun S, Beetz M, Bennewitz M, Cremers A B, Dellaert F, Fox D. Probabilistic algorithms and the interactive museum tour guide robot minerva. *International Journal of Robotics Research*, 2000, **19**(11): 972−999

9 Burgard W, Cremers A B, Fox D, Hähnel D, Lakemeyer G, Schulz D. Experiences with an interactive museum tour guide robot. *Artificial Intelligence*, 2000, **114**(1-2): 3−55

10 Li Mao-Hai, Hong Bing-Rong. Progress of probabilistic localization methods in mobile robots. *Robot*, 2005, **7**(4): 380−384 (in Chinese)

11 Li Qun-Ming, Xiong Rong, Chu Jian. Localization approaches for indoor autonomous mobile robot: a review. *Robot*, 2003, **25**(6): 560−567 (in Chinese)

12 Fang Zheng, Tong Guo-Feng, Xu Xin-He. Study of autonomous robot self-localization methods based on Bayesian filter theory. *Control and Decision*, 2006, **21**(8): 841−847 (in Chinese)

13 Durrant-Whyte H F, Bailey T. Simultaneous localization and mapping: Part I. *IEEE Robotics and Automation Magazine*, 2006, **13**(2): 99−110

14 Bailey T, Durrant-Whyte H F. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics and Automation Magazine*, 2006, **13**(3): 108−117

15 Adorni G, Cagnoni S, Enderle S, Kraetzschmar G K, Mordonini M, Michael P. Vision-based localization for mobile robots. *Robotics and Autonomous Systems*, 2001, **36**(2-3): 103−119

16 Fox D, Burgard W, Thrun S. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 1999, **11**: 391−427

17 Kwok C, Fox D, Meila M. Real-time particle filters. *Proceedings of IEEE*, 2004, **92**(3): 469−484

18 Kwok C, Fox D, Meila M. Adaptive real-time particle filters for robot localization. In: Proceedings of IEEE International Conference on Robotics and Automation. Taipei, Taiwan: IEEE, 2003. 2836−2841

19 Arras K O, Tomatis N, Jensen B T, Siegwart R. Multisensor on-the-fly localization: precision and reliability for applications. *Robotics and Autonomous Systems*, 2001, **34**(2-3): 131−143

20 Pan Liang-Chen, Chen Wei-Dong. Vision-based localization of indoor mobile robot. *Robot*, 2006, **28**(5): 504−509 (in Chinese)

21 Wang Jing-Chuan, Chen Wei-Dong, Cao Qi-Xin. Omnivision and odometer based self-localization for mobile robot. *Robot*, 2006, **27**(1): 41−45 (in Chinese)

22 Julier S J, Uhlmann J K, Durrant-Whyte H F. A new approach for filtering nonlinear systems. In: Proceedings of American Control Conference. Seattle, USA: IEEE, 1995. 1628−1632

23 Sergios T, Konstantinos K. *Pattern Recognition (Second Edition)*. Beijing: Publishing House of Electronics Industry, 2003. 31−34

24 Bing M, Lakshmanan S, Hero O A. Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion. *IEEE Transactions on Intelligent Transportation Systems*, 2000, **1**(3): 135−147

25 Julier S J, Uhlmann J K, Durrant-Whyet H F. A new method for the nonlinear transformation of means and covariances in filters and estimator. *IEEE Transactions on Automatic Control*, 2000, **45**(3): 477−482

26 Neto G, Costelha H, Lima P. Topological navigation in configuration space applied to soccer robots. *Lecture Notes in Computer Science*, 2004, **3020**: 551−558

27 Schmitt T, Hanek R, Beetz M, Buck S, Radig B. Cooperative probabilistic state estimation for vision-based autonomous mobile robots. *IEEE Transactions on Robotics and Automation*, 2002, **18**(5): 670−684

28 Diosi A, Kleeman L. Advanced sonar and laser range finder fusion for simultaneous localization and mapping. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems. Taipei, Taiwan: IEEE, 2004. 1854−1859

29 Bengtsson O, Baerveldt A J. Robot localization based on scan-matching-estimating the covariance matrix for the IDC algorithm. *Robotics and Autonomous Systems*, 2003, **44**(1): 29−40

30 Zhuang Yan, Wang Wei, Wang Ke, Xu Xiao-Dong. Mobile robot indoor simultaneous localization and mapping using laser range finder and monocular vision. *Acta Automatica Sinica*, 2005, **31**(6): 925−933 (in Chinese)

31 Nicola T. Hybrid, Metric-topological, Mobile Robot Navigation [Ph. D. dissertation], 2001

32 Borges G A, Aldon M B. Robustified estimation algorithms for mobile robot localization based on geometrical environment maps. *Robotics and Autonomous Systems*, 2003, **45**(3-4): 131−159

33 Dhome M, Richetir M, Lapreste J T. Determination of the attitude of 3D objects from a single perspective view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989, **11**(12): 1265−1278

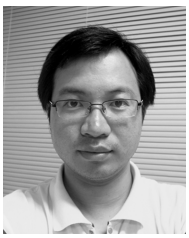
34 Durrant-Whyte H F. Uncertain geometry in robotics. In: Proceedings of International Conference on Robotics and Automation. Oxford, UK: IEEE, 1987. 851−856

35 Ayache N, Fauqeras O D. Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 1989, **5**(6): 804−819

36 Zhang Z Y, Faugeras O D. *3D Dynamic Scene Analysis: a Stereo Based Approach*. Berlin: Springer-Verlag, 1992. 43−54

37 Burden R L, Faires J D. *Numerical Analysis (Seventh Edition)*. Beijing: Higher Education Press, 2001. 611−628

38 Neira J, Tardód os J D, Horn J, Schmidt G. Fusing range and intensity images for mobile robot localization. *IEEE Transactions on Robotics and Automation*, 1999, **15**(1): 76−84



WANG Ke Ph. D. candidate in Research Center of Information and Control at Dalian University of Technology. His research interest covers computer vision, image processing, human machine interaction, and estimation theory and their applications in robotics. Corresponding author of this paper.
E-mail: qingshang00@163.com

WANG Wei Professor in Research Center of Information and Control at Dalian University of Technology. He received his Ph. D. degree in control theory and control engineering from Northeastern University in 1988. He was a post-doctor at Norwegian Science and Technology University (1990∼1992). His research interest covers predictive control, robotics, and intelligent control.
E-mail: wangwei@dlut.edu.cn

ZHUANG Yan Lecturer in Department of Automation at Dalian University of Technology. He received his Ph. D. degree in control theory and control engineering from Dalian University of Technology in 2004. His research interest covers robot localization, map building, and navigation.
E-mail: zhuang@dlut.edu.cn