

Journal of Zhejiang University SCIENCE A
 ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
 www.zju.edu.cn/jzus; www.springerlink.com
 E-mail: jzus@zju.edu.cn



A novel method for mobile robot simultaneous localization and mapping*

LI Mao-hai[†], HONG Bing-rong, LUO Rong-hua, WEI Zhen-hua
 (School of Computer Science, Harbin Institute of Technology, Harbin 150001, China)

[†]E-mail: limaochai@163.com

Received Oct. 25, 2005; revision accepted Jan. 18, 2006

Abstract: A novel mobile robot simultaneous localization and mapping (SLAM) method is implemented by using the Rao-Blackwellized particle filter (RBPF) for monocular vision-based autonomous robot in unknown indoor environment. The particle filter combined with unscented Kalman filter (UKF) for extending the path posterior by sampling new poses integrating the current observation. Landmark position estimation and update is implemented through UKF. Furthermore, the number of resampling steps is determined adaptively, which greatly reduces the particle depletion problem. Monocular CCD camera mounted on the robot tracks the 3D natural point landmarks structured with matching image feature pairs extracted through Scale Invariant Feature Transform (SIFT). The matching for multi-dimension SIFT features which are highly distinctive due to a special descriptor is implemented with a KD-Tree. Experiments on the robot Pioneer3 showed that our method is very precise and stable.

Key words: Mobile robot, Rao-Blackwellized particle filter (RBPF), Monocular vision, Simultaneous localization and mapping (SLAM)

doi:10.1631/jzus.2006.A0937

Document code: A

CLC number: TP24

INTRODUCTION

A key prerequisite for a truly autonomous robot is that it can simultaneously localize itself and accurately map its surroundings (Kortenkamp *et al.*, 1998), which is known as Simultaneous Localization and Mapping (SLAM), which, when phrased as a state estimation problem, involves a variable number of dimensions. Murphy and Russell (2001) adopted Rao-Blackwellized particle filters (RBPFs) as an effective way for representing alternative hypotheses on robot paths and associated maps. Montemerlo and Thrun (2003) extended this method to efficient landmark-based SLAM using Gaussian representations of the landmarks and were the first to successfully implement it on real robots. More recently, RBPF is widely used to build map (Sim *et al.*, 2005; Davison, 2003; Stachniss *et al.*, 2005).

We focus mainly on investigating real-time monocular vision based SLAM for indoor environments, and constructing map with 3D feature landmarks structured from the Scale Invariant Feature Transform (SIFT) feature matching pairs. These SIFT features are invariant to image scale, rotation and translation as well as partially invariant to illumination changes and affine or 3D projection, and their description is implemented with multi-dimensional vector (Lowe, 2004). This combination can result in many highly distinctive landmarks from environment, which simplifies the data association problem to only distinguishing unique landmarks. We present a fast and efficient algorithm for matching features in a KD-Tree at time cost of $O(\log 2^N)$ (Moore, 1991); our approach applies RBPF to estimate a posterior of the path of the robot, and each landmark is estimated and updated by the unscented Kalman filter (UKF) (Merwe *et al.*, 2000), and UKF is used to sample new poses integrating the current observation. Furthermore, the number of resampling steps is determined adaptively,

* Project (No. 2002AA735041) supported by the Hi-Tech Research and Development Program (863) of China

which greatly reduces the particle depletion problem. Experimental results indicated superior performance.

RAO-BLACKWELLIZED MAPPING

Consider the case of a mobile robot moving through an unknown environment consisting of a set of landmarks θ . The robot moves according to a known motion model $p(s_t|s_{t-1}, u_t)$, where s_t denotes the robot state at time t , and the control input u_t carried out in the time interval $[t-1, t]$. As the robot moves around, it takes measurements of its environment. A measurement z_t is related to the position of a landmark through observation model $p(s_t|u_t, \theta, s_{t-1})$.

The SLAM problem is that of simultaneously inferring the location of all landmarks and the path followed by the robot based on a set of measurements and inputs. Ideally, one would like to recover the posterior distribution $p(s^t, \theta | z^t, u^t, n^t)$, where the notation $s^t = s_1, s_2, \dots, s_t$ (and similarly for other variables). Doucet et al.(2000) provided an implementation of RBPF for SLAM:

$$p(s^t, \theta | z^t, u^t, n^t) = p(s^t | z^t, u^t, n^t) \prod_{n=1}^M p(\theta_n | s^t, z^t, n^t). \tag{1}$$

This can be done efficiently, since the factorization decouples the SLAM problem into a path estimation problem and individual conditional landmark location problems, and the quantity $p(\theta_n | s^t, z^t, n^t)$ can be computed analytically once s^t and z^t are known. The posterior $p(s^t | z^t, u^t, n^t)$ over the potential robot trajectories uses a particle filter in which an individual map is associated to each particle. Each map is constructed given the observations z^t and the trajectory s^t represented by the corresponding particle.

RBPF FOR MONOCULAR VISION BASED SLAM

RBPF calculates the posterior over robot paths $p(s^t | z^t, u^t, n^t)$ by a particle filter. The remaining M posteriors over landmark locations $p(\theta_n | s^t, n^t, z^t, u^t)$ are calculated and updated with UKF. Each UKF conditioned on robot paths estimates a single landmark pose. Each particle is of the form $S_t^{(i)} = \{s^{t,(i)}, \mu_{1,t}^{(i)},$

$\Sigma_{1,t}^{(i)}, \dots, \mu_{m,t}^{(i)}, \Sigma_{m,t}^{(i)}\}$, where (i) indicates the index of the particle; $s^{t,(i)}$ is its path estimate, and $\mu_{m,t}^{(i)}$ and $\Sigma_{m,t}^{(i)}$ are the mean and variance of the Gaussian representing the m th landmark location respectively. Together, all these quantities form the i th particle $S_t^{(i)}$, of which there is a total of N in the posterior. Our RBPF update is performed in the following steps:

1. Sampling new poses using UKF

Here we need to calculate the posterior over robot paths $p(s^t | z^t, u^t, n^t)$ approximated by a particle filter. Each particle in the filter represents one possible robot path s^t from time 0 to time t . Since the map landmark estimates $p(\theta_n | s^t, z^t, n^t)$ depend on the robot path, the particles sampling step is very important. However, most methods use the state transition prior $p(s_t | s_{t-1}, u_t)$ to draw particles. Because the state transition does not take into account the most recent observation z_t , especially when the likelihood happens to lie in one of the tails of the prior distribution or if it is too narrow, as shown in Fig.1. If an insufficient number of particles are employed, there may be a lack of particles in the vicinity of the correct state, leading to divergence of the filter. This is known as the particles depletion problem.

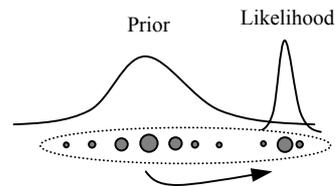


Fig.1 Moving the samples in the prior to regions of high likelihood is important if the likelihood lies in one of the tails of the prior

In our methods, the i th new pose $s_t^{(i)}$ is drawn from the posterior $p(s_t | s^{t-1,(i)}, u^t, z^t, n^t)$, which takes the measurement z_t into consideration, along with the landmark n_t , and $s^{t-1,(i)}$ is the path up to time $t-1$ of the i th particle. An effective approach to accomplish this is to use an EKF generated Gaussian approximation:

$$p(s_t | s^{t-1,(i)}, u^t, z^t, n^t) \sim N(s_t; \bar{s}_t^{(i)}, P_t^{(i)}), \quad i = 1, 2, \dots, N \tag{2}$$

EKF approximates the distribution through the

first-order Taylor-series expansion of the nonlinear observation function $g(\theta_{n_t}, s_t)$ around the mean \underline{s}_t :

$$z_t = g(\theta_{n_t}, s_t) \sim g(\theta_{n_t}, \underline{s}_t) + \Delta_{s_t} g'(\theta_{n_t}, \underline{s}_t). \quad (3)$$

The first-order mean and covariance used in the EKF is given by $\underline{z}_t = g(\theta_{n_t}, \underline{s}_t)$, $P_{z_t} = g'(\theta_{n_t}, \underline{s}_t)^T P_{s_t} g'(\theta_{n_t}, \underline{s}_t)$ which often introduces large errors. However, the unscented transformation (UT) is an elegant way to accurately compute the mean and covariance up to the third order of the Taylor series expansion of $g(\theta_{n_t}, s_t)$ (Merwe *et al.*, 2000). Let L be the dimension of s_t , the UT computes mean and covariance as follows:

(1) Deterministically generate $2L+1$ sigma points $S_t = \{\chi_i, W_i\}$:

$$\begin{aligned} \chi_0 &= \underline{s}_t, \chi_i = \underline{s}_t + (\sqrt{(L+\lambda)P_{s_t}})_i, i = 1, \dots, L \\ \chi_i &= \underline{s}_t - (\sqrt{(L+\lambda)P_{s_t}})_i, i = L+1, \dots, 2L \\ W_0^m &= \lambda/(L+\lambda), W_0^c = W_0^m + (1-\alpha^2 + \beta), \\ W_i^m &= 1/[2 \cdot (L+\lambda)], i = 1, \dots, 2L \\ \lambda &= \alpha^2(L+\gamma) - L, \end{aligned} \quad (4)$$

where γ is a scaling parameter that controls the distance between the sigma points and the mean \underline{s}_t , α is a positive scaling parameter that controls the higher order effects resulted from the non-linear function g , β is a parameter that controls the weighting of the 0th sigma point $\alpha=0, \beta=0$ and $\gamma=2$ are the optimal values for the scalar case (Merwe *et al.*, 2000). $(\sqrt{(L+\lambda)P_{s_t}})_i$ is the i th column of the matrix square root. Note that the 0th sigma point's weight is different for calculating mean and covariance.

(2) Propagate the sigma points through the nonlinear transformation:

$$Z_i = g(\theta_{n_t}, \chi_i), i = 0, \dots, 2L. \quad (5)$$

(3) Compute the mean and covariance as follows:

$$\begin{aligned} \underline{z}_t &= \sum_{i=0}^{2L} W_i^m Z_i, \\ P_{z_t} &= \sum_{i=0}^{2L} W_i^c (Z_i - \underline{z}_t)(Z_i - \underline{z}_t)^T. \end{aligned} \quad (6)$$

Now we follow UKF algorithm to extend the path $s^{t(i)}$ by sampling the new poses $s_t^{(i)}$ from the posterior $p(s_t | s^{t-1(i)}, u^t, z^t, n^t)$:

(1) Calculate the sigma points according to Eq.(4):

$$\chi_{t-1}^{(i)} = \{\underline{s}_{t-1}^{(i)}, \underline{s}_{t-1}^{(i)} \pm \sqrt{(L+\lambda)P_{t-1}^{(i)}}\}. \quad (7)$$

(2) Using motion model to predict:

$$\begin{aligned} \chi_{t|t-1}^{*(i)} &= f(\chi_{t-1}^{(i)}, u_t^{(i)}), \underline{s}_{t|t-1}^{(i)} = \sum_{j=0}^{2L} W_j^{m,(i)} \chi_{j,t|t-1}^{*(i)}, \\ P_{t|t-1}^{(i)} &= \sum_{j=0}^{2L} W_j^{c,(i)} [\chi_{j,t|t-1}^{*(i)} - \underline{s}_{t|t-1}^{(i)}][\chi_{j,t|t-1}^{*(i)} - \underline{s}_{t|t-1}^{(i)}]^T. \end{aligned} \quad (8)$$

(3) Incorporating new observation z_t , along with the landmark n_t :

$$\begin{aligned} Z_{t|t-1}^{*(i)} &= g(\chi_{t|t-1}^{*(i)}, \theta_{n_t}), \underline{z}_{t|t-1}^{(i)} = \sum_{j=0}^{2L} W_j^{m,(i)} Z_{j,t|t-1}^{*(i)}, \\ P_{z_t z_t}^{(i)} &= \sum_{j=0}^{2L} W_j^{c,(i)} [Z_{j,t|t-1}^{*(i)} - \underline{z}_{t|t-1}^{(i)}][Z_{j,t|t-1}^{*(i)} - \underline{z}_{t|t-1}^{(i)}]^T, \\ P_{s_t z_t}^{(i)} &= \sum_{j=0}^{2L} W_j^{c,(i)} [\chi_{j,t|t-1}^{*(i)} - \underline{s}_{t|t-1}^{(i)}][Z_{j,t|t-1}^{*(i)} - \underline{z}_{t|t-1}^{(i)}]^T, \\ K_t^{(i)} &= P_{s_t z_t}^{(i)} (P_{z_t z_t}^{(i)})^{-1}, \underline{s}_t^{(i)} = \underline{s}_{t|t-1}^{(i)} + K_t^{(i)} (z_t - \underline{z}_{t|t-1}^{(i)}), \\ P_t^{(i)} &= P_{t|t-1}^{(i)} - K_t^{(i)} P_{z_t z_t}^{(i)} (K_t^{(i)})^T. \end{aligned} \quad (9)$$

(4) Sampling new pose $s_t^{(i)}$ and extending the path $s^{t(i)}$:

$$\begin{aligned} s_t^{(i)} &\sim p(s_t | s^{t-1(i)}, u^t, z^t) = N(s_t; \underline{s}_t^{(i)}, P_t^{(i)}), \\ s^{t(i)} &= (s^{t-1(i)}, s_t^{(i)}). \end{aligned} \quad (10)$$

2. Updating the observed landmark estimate

In this step, we update the posterior over the landmark estimates represented by the mean $\mu_{n,t-1}^{(i)}$, and the covariance $\Sigma_{n,t-1}^{(i)}$. The updated values $\mu_{n,t}^{(i)}$ and $\Sigma_{n,t}^{(i)}$ are then added to the temporary particle set \hat{S}_t along with the new sampling pose $s_t^{(i)}$. The update depends on whether or not a landmark n was observed at time t . For $n \neq n_t$, the posterior over the landmark remains unchanged. For the observed feature $n = n_t$, the update is specified as follows:

$$\begin{aligned} &p(\theta_{n_t} | s^{t(i)}, n^t, z^t) \\ &= \frac{p(z_t | \theta_{n_t}, s^{t(i)}, n^t, z^{t-1}) p(\theta_{n_t} | s^{t(i)}, n^t, z^{t-1})}{p(z_t | s^{t(i)}, n^t, z^{t-1})} \\ &= \underbrace{\eta p(z_t | \theta_{n_t}, s_t^{(i)}, n_t)}_{\sim N(z_t; g(\theta_{n_t}, s_t^{(i)}), R_t)} \underbrace{p(\theta_{n_t} | s^{t-1(i)}, n^{t-1}, z^{t-1})}_{\sim N(\theta_{n_t}; \mu_{n,t-1}^{(i)}, \Sigma_{n,t-1}^{(i)})}. \end{aligned} \quad (11)$$

The probability $p(\theta_{n_t} | s^{t-1(i)}, z^{t-1}, n^{t-1})$ at time $t-1$ is represented by a Gaussian with mean $\mu_{n_t, t-1}^{(i)}$, and the covariance $\Sigma_{n_t, t-1}^{(i)}$. For the new estimate at time t to also be Gaussian, we need to generate Gaussian approximation for the perceptual model $p(z_t | s_t^{(i)}, \theta_{n_t}, n_t)$. Our methods also use UT to approximate the non-linear measurement function $g(\theta_{n_t}, s_t^{(i)})$. This can accurately compute the mean and covariance up to the third order of the Taylor series expansion:

(1) Calculate the sigma points according to Eq.(4):

$$\xi_{n_t, t-1}^{(i)} = \left\{ \mu_{n_t, t-1}^{(i)}, \mu_{n_t, t-1}^{(i)} \pm \sqrt{(L + \lambda) \Sigma_{n_t, t-1}^{(i)}} \right\}. \quad (12)$$

(2) Using observation model to compute the mean and covariance of the observation as follows:

$$\begin{aligned} Z_{n_t, t}^{(i)} &= g(\xi_{n_t, t-1}^{(i)}, s_t^{(i)}), \quad \underline{z}_{n_t, t}^{(i)} = \sum_{j=0}^{2L} W_j^{m, (i)} Z_{j, n_t, t}^{(i)}, \\ P_{z_{n_t, t}}^{(i)} &= \sum_{j=0}^{2L} W_j^{c, (i)} [Z_{j, n_t, t}^{(i)} - \underline{z}_{n_t, t}^{(i)}][Z_{j, n_t, t}^{(i)} - \underline{z}_{n_t, t}^{(i)}]^T. \end{aligned} \quad (13)$$

(3) Under this approximation, the posterior for the location of landmark n_t is indeed Gaussian. The new mean and covariance are obtained using the following update:

$$\begin{aligned} K_t^{(i)} &= \Sigma_{n_t, t-1}^{(i)} P_{z_{n_t, t}}^{(i)} [(P_{z_{n_t, t}}^{(i)})^T \Sigma_{n_t, t-1}^{(i)} P_{z_{n_t, t}}^{(i)} + R_t]^{-1}, \\ \mu_{n_t, t}^{(i)} &= \mu_{n_t, t-1}^{(i)} + K_t^{(i)} (z_t - \underline{z}_t^{(i)})^T, \\ \Sigma_{n_t, t}^{(i)} &= [I - K_t^{(i)} (P_{z_{n_t, t}}^{(i)})^T] \Sigma_{n_t, t-1}^{(i)}. \end{aligned} \quad (14)$$

3. Selective resampling

Next, we resample from temporary set of particles \tilde{S}_t , then form the new particle set S_t . The necessity to resample arises from the fact that the particles in the temporary set \tilde{S}_t do not yet match the desired posterior. Resampling is a common technique in particle filtering to correct for such mismatches, and avoid particles degeneracy. By weighing particles in \tilde{S}_t , and resampling according to those weights, the resulting particle set indeed approximates the target distribution. To determine the importance weight of each particle, let $F_t = \{f_1, \dots, f_k\}$ be the k SIFT feature key-points observed at time t , in which there are n key-points matching with the landmarks in the map database: $Match_t = \{f_1 \sim L_{f_1}, \dots, f_n \sim L_{f_n}\}$; and there are

m key-points matching the SIFT key-points which observed at time $t-1$ and are not added to the map database: $Match_v = \{f_{n+1} \sim V_{f_{n+1}}, \dots, f_{n+m} \sim V_{f_{n+m}}\}$. Then the log-likelihood of the observation z_t being obtained is:

$$\log p(z_t | m_t^{(i)}) = \log p(F_t | Match_t) + \log p(F_v | Match_v), \quad (15)$$

where $F_t = \{f_1, \dots, f_n\}$, $F_v = \{f_{n+1}, \dots, f_{n+m}\}$. The log-likelihood $\log p(F_t | Match_t)$ and $\log p(F_v | Match_v)$ are given respectively by:

$$\log p(F_t | Match_t) = \sum_{j=1}^n \log p(f_j | L_{f_j}), \quad (16)$$

$$\log p(F_v | Match_v) = \sum_{j=n+1}^{n+m} \log p(f_j | V_{f_j}), \quad (17)$$

where $\log(p_{f_j} | L_{f_j})$ indicates the log-likelihood of the match between the current feature f_j and the landmark L_{f_j} , $\log(p_{f_j} | L_{f_j})$ indicates the log-likelihood of the match between the current feature f_j and the feature V_{f_j} at prior time $t-1$. The $\log(p_{f_j} | L_{f_j})$ is represented as follows:

$$\begin{aligned} \log p(f_j | L_{f_j}) &= -0.5 \min \left(p_t, (\hat{I}_j - I_j)^T S^{-1} (\hat{I}_j - I_j) \right), \\ S &= J(R_t C_{f_j} R_t^T) J^T, \end{aligned} \quad (18)$$

where C_{f_j} is the 3D landmark covariance, J is the Jacobian matrix of the projection equation. The observation innovation p_t is constant (in our case, $p_t=3.0$), which is selected so as to prevent outlier observations from significantly affecting the observation likelihood. In the same way, $\log(p_{f_j} | L_{f_j})$ is also represented as:

$$\begin{aligned} \log p(f_j | V_{f_j}) &= -0.5 \min \left(p_t, \text{dist}(I_j, H_{f_j}) \right. \\ &\quad \left. + \text{dist}(I_j, H_j) \right), \end{aligned} \quad (19)$$

where I_{f_j} is the image coordinate of the feature V_{f_j} , H_{f_j} is the epipolar line on the image plane corresponding to V_{f_j} at time t , and H_j is the epipolar line on the image plane corresponding to the feature f_j at time

$t-1$, $dist(\cdot)$ is the function of the distance between point and line. Then $p(z_t | m_t^{(i)})$ can be used to evaluate the i th particle weight:

$$w_t^{(i)} = \frac{p(z_t | m_t^{(i)})}{\sum_{j=1}^N p(z_t | m_t^{(j)})}. \quad (20)$$

After the resampling, all particle weights are then reset to $w_t^{(i)}=1/N$. However, resampling can delete good samples from the sample set, in the worst case, the filter diverges. Accordingly, it is important to find a criterion when implementing a resampling step. Liu and Chen (1998) introduced the so-called number of particles $N_{t,\text{eff}} = 1 / \sum_{i=1}^N (w_t^{(i)})^2$ to estimate how well the current particle set represents the true posterior. Our approach determines whether or not a resampling should be carried out according to $N_{t,\text{eff}}$. We resample each time $N_{t,\text{eff}}$ drops below a given threshold which was set to $0.5N$ where N is the number of particles. In our experiments we found that this technique drastically reduces the risk of replacing good particles, because the number of resampling operations is reduced and resampling operations are only performed when needed.

IMPLEMENTATION DETAILS

Motion model

The motion model $p(s_t | u_t, s_{t-1})$ predicts the movement and status over time of the robot. When a control u , consisting of forward and angular velocity is applied to the robot, we employ Eq.(2) to predict the robot moves:

$$p(s_t | u_t, s_{t-1}) = f(u_t, s_{t-1}) + \varepsilon_t, \\ f(u_t, s_{t-1}) = \begin{bmatrix} x_{t+1}^i \\ y_{t+1}^i \\ \varphi_{t+1}^i \end{bmatrix} = \begin{bmatrix} x_t^i \\ y_t^i \\ \varphi_t^i \end{bmatrix} + \begin{bmatrix} v\Delta T \cos(\varphi_t^i + \omega_t \Delta T) \\ v\Delta T \sin(\varphi_t^i + \omega_t \Delta T) \\ \varphi_t^i + \omega_t \Delta T \end{bmatrix}, \quad (21)$$

where $(x_t^i, y_t^i, \varphi_t^i)$ is the robot's location and bearing at time t , for all particles $i=1, \dots, N$, v_t is the line velocity, ω_t is the angular velocity at time t , ΔT is the time step

and ε_t is noise in terms of a normal distribution $\mathcal{N}(0, P_t)$.

SIFT feature extraction

SIFT was proposed in (Lowe, 2004) as a method for extracting and describing key-points, which are robustly invariant to common image transforms. The SIFT algorithm has four major stages:

(1) Scale-space extrema detection. The first stage searches over scale space using a Difference of Gaussian function to identify potential interest points.

(2) Key-point localization. The location and scale of each candidate point is determined and key-points are selected based on measures of stability.

(3) Orientation assignment. One or more orientations are assigned to each key-point based on local image gradients.

(4) Key-point descriptor. A descriptor is generated for each key-point from local image gradients information at the scale found in Stage 2. An important aspect of the algorithm is that it generates a large number of highly distinctive features over a broad range of scales and locations.

KD-Tree based feature matching

This section describes KD-Tree algorithm for determining the matching SIFT features pairs of successive images captured by a monocular vision system mounted on the robot. Every time the CCD camera vision system is triggered, it captures the consecutive digital images of pixels and after SIFT feature extracting, generates SIFT feature match pairs in adjacent images through KD-Tree based feature matching algorithm. The match pairs are used for the landmarks' 3D structure. Given a SIFT key-points set E , and a target key-point vector d , then a nearest neighbor of d , d' is defined as:

$$\forall d'' \in E, |d \leftrightarrow d'| \leq |d \leftrightarrow d''|, \\ |d \leftrightarrow d'| = \left[\sum_{i=1}^k (d_i \leftrightarrow d'_i)^2 \right]^{1/2}, \quad (22)$$

where d_i is the i th component of d . The KD-Tree based SIFT feature matching algorithm is described as follows: A KD-Tree is constructed using all key-points of the image I_t . For each key-point kp in the next image I_{t+1} , finding the two most nearest neighbors kp_1 and kp_2 based on nearest neighbor algorithm in a KD-Tree. As proved in our experiment,

if $|kp_1 - kp|/|kp_2 - kp|$ is bigger, then the matching quality between kp and kp_1 is much higher, otherwise the matching quality is lower. So we can use the following equation to judge the matching for two key-points:

$$|kp_1 - kp|/|kp_2 - kp| < \lambda, \quad (23)$$

where λ is constant, and $0 < \lambda < 1$ (in this paper λ is evaluated as 0.7), if this equation is satisfied, then the matching is successful, and simultaneously eliminates the false matching.

3D structure

After the SIFT feature matching, we obtain the feature matching pairs. For any pair of matching SIFT feature points $p_1(u_1, v_1, 1)$ and $p_2(u_2, v_2, 1)$, using the pinhole camera model (Ma and Zhang, 1998) to construct the landmarks' 3D world coordinates $P(X_w, Y_w, Z_w)$ of corresponding SIFT feature matches, and all the landmarks are in a single world model.

Observation model

In this section we consider the observation model $p(z_i | s_i, \theta, n_i)$ of the landmark θ_{n_i} which is of the following form:

$$p(z_i | s_i, \theta, n_i) = g(\theta_{n_i}, s_i) + \delta_i, \quad (24)$$

where $\delta = [\delta_x, \delta_y, \delta_z]^T$ is noise in terms of a normal distribution $N(0, R_i)$.

As shown in Fig.2, the estimate of the robot position through motion model, and the computation of landmark's 3D spatial position $P(X_w, Y_w, Z_w)$ through the structure for the matching SIFT image feature pairs $p_1(u_1, v_1)$ and $p_2(u_2, v_2)$, which allow the landmark's measurement to be predicted as $p'(u', v')$ according to measurement function $g(\theta_{n_i}, s_i^{(i)})$, and

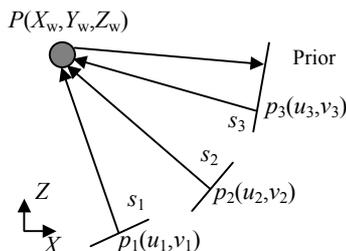


Fig.2 Observation model

$p'(u', v')$ is the projection of 3D spatial point $P(X_w, Y_w, Z_w)$ on the image plane. The uncertainty in this prediction, represented in the covariance matrix Σ , gives the shape of the Gaussian probability distribution over image coordinates and choosing a number of standard deviations defines an elliptical window within which the feature should lie with high probability. For further analysis, we can obtain the Jacobian G_{θ, n_i} for updating the observed landmark estimation:

$$G_{\theta, n_i} = \begin{bmatrix} \frac{\partial u_i}{\partial X_{wi}} & \frac{\partial u_i}{\partial Y_{wi}} & \frac{\partial u_i}{\partial Z_{wi}} \\ \frac{\partial v_i}{\partial X_{wi}} & \frac{\partial v_i}{\partial Y_{wi}} & \frac{\partial v_i}{\partial Z_{wi}} \end{bmatrix} (G_{\theta, n_i})^T \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} \Delta X_{wi} \\ \Delta Y_{wi} \\ \Delta Z_{wi} \end{bmatrix}. \quad (25)$$

EXPERIMENTAL RESULTS

Experiments were performed on a Pioneer 3-DX mobile robot incorporating an 800 MHz Intel Pentium processor as shown in Fig.3. Motor control is achieved by the on-board computer, while a 2.6 GHz PC connected to the robot by a wireless link provides the main processing power for vision processing and the SLAM software. A monocular color CCD camera with effective field of view of about 50° is used for detecting the landmarks. The test environment was a robot laboratory with limited space as shown in Fig.3.



Fig.3 Pioneer 3 is running in the test environment

The following experiment was carried out online, i.e., the continuous images are captured and processed, and the map is kept and updated on the fly while the robot is moving around. The robot goes around in the

laboratory for one loop and comes back. At each frame, it keeps track of the landmarks in the map, adds new ones and updates existing ones if matched. Fig.4 shows some frames of the 320×240 image sequence (180 frames in total) captured while the robot is moving around. At the end, a total of 4068 SIFT landmarks with 3D positions are gathered in the map, which are relative to the initial coordinates frame. The typical time required for each iteration is around 0.35~0.45 s, in which most of time is spent on the SIFT

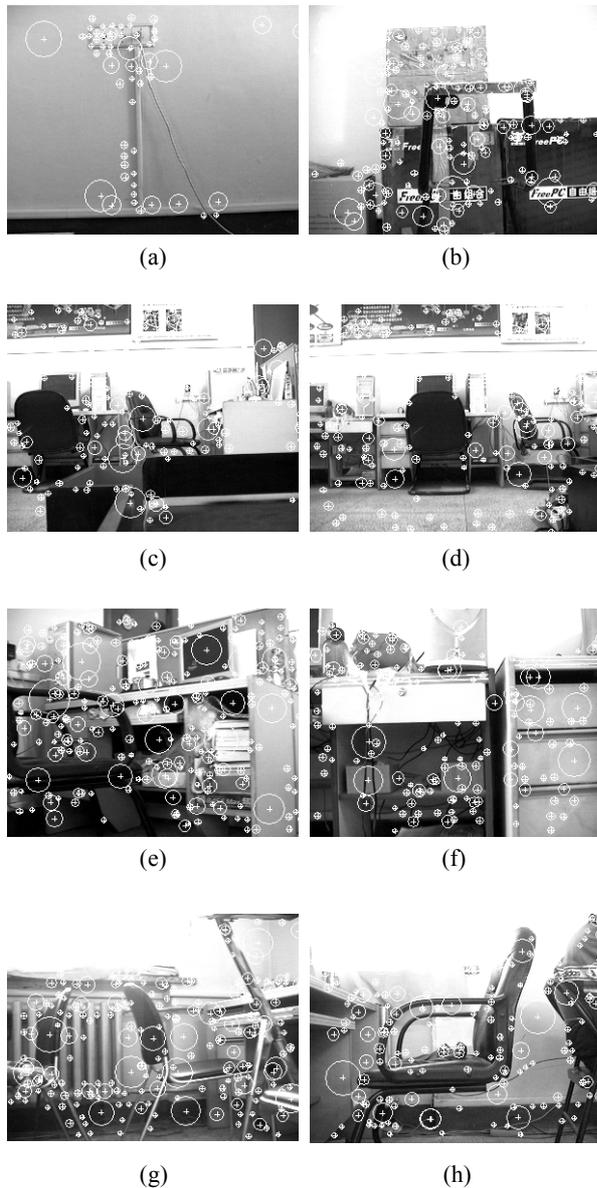


Fig.4 Frames of an image sequence with SIFT features marked: (a) 4th frame; (b) 19th frame; (c) 70th frame; (d) 79th frame; (e) 110th frame; (f) 124th frame; (g) 142nd frame; (h) 169th frame

feature extraction stage. The runtime of our RBPf SLAM algorithm with different numbers landmarks is shown in Fig.5. Other performance of our SLAM algorithm with different numbers of particles is also shown in Fig.5. Fig.6 shows the bird's-eye view of all

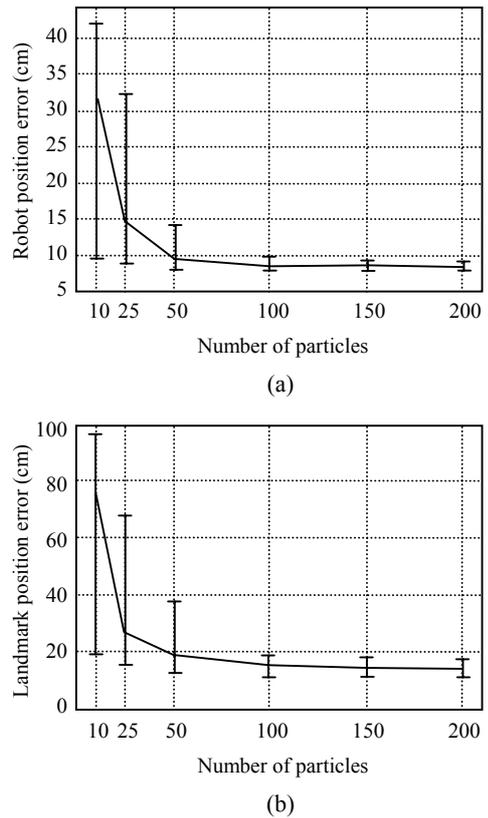


Fig.5 Performance of our RBPf SLAM algorithm with different numbers of particles. (a) Robot position error affected by the number of particles; (b) Landmark position error affected by the number of particles

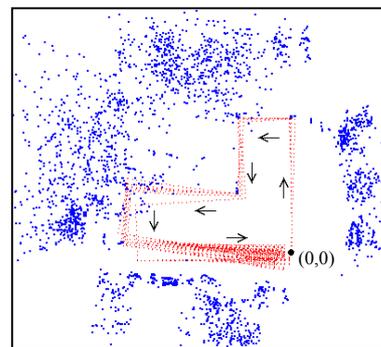


Fig.6 Bird's-eye view of the SIFT landmarks (blue dot) in the map. The coordinate (0,0) indicates the initial robot position, the dashed line indicates the estimated robot path. The arrow indicates the direction of robot's movement

these landmarks. Consistent clusters are observed corresponding to objects such as chairs, shelves, cartons and posters in the scene.

At last we compare our method with traditional EKF method, with our method showing superior performance as shown in Fig.7.

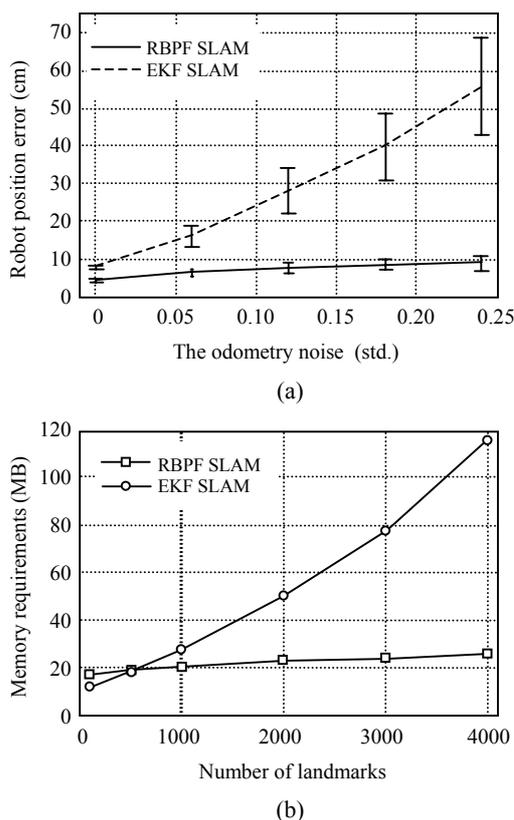


Fig.7 Comparison of our RBPF SLAM algorithm and EKF for error and memory requirement. (a) Robot position error affected by odometry noise; (b) Robot position error affected by the number of landmarks

CONCLUSION

This article described a novel algorithm for SLAM problem using monocular CCD camera to extract SIFT features. Being invariant to image scale, rotation and translation as well as partially invariant to illumination changes and affine or 3D projection, and their description is implemented with multi-dimensional vector, highly distinctive SIFT features

are good natural visual landmarks for tracking over a long period of time from different views. These tracked landmarks are used for concurrent robot pose estimation and 3D map building with promising results shown. Further experiments in larger environments are planned to evaluate the scalability of our approach.

References

- Davison, A.J., 2003. Real-Time Simultaneous Localisation and Mapping with a Single Camera. Proc. of the Ninth Int. Conf. on Computer Vision ICCV'03. Nice, France, p.1403-1410.
- Doucet, A., de Freitas, J., Murphy, K., Russel, S., 2000. Rao-Blackwellized Particle Filtering for Dynamic Bayesian Networks. Proc. of Conf. on Uncertainty in Artificial Intelligence (UAI). California, USA.
- Kortenkamp, D., Bonasso, R.P., Murphy, R., 1998. AI-based Mobile Robots: Case Studies of Successful Robot Systems. MIT Press, Cambridge.
- Liu, J.S., Chen, R., 1998. Sequential Monte Carlo methods for dynamical systems. *J. Amer. Statist. Assoc.*, **93**(443): 1032-1044. [doi:10.2307/2669847]
- Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. of Computer Vision*, **60**(2):91-110. [doi:10.1023/B:VISI.0000029664.99615.94]
- Ma, S.D., Zhang, Z.Y., 1998. Computer Vision—Computational Theories and Algorithm. Science Press, Beijing, p.52-79 (in Chinese).
- Merwe, R., Doucet, A., Freitas N., Wan, E., 2000. The Unscented Particle Filter. Technical Report CUED/FINFENG/TR 380. Cambridge University.
- Montemerlo, M., Thrun, S., 2003. Simultaneous Localization and Mapping with Unknown DATA Association Using FastSLAM. Proc. IEEE Int. Conf. Robotics and Automation (ICRA). Taipei, China, p.1985-1991.
- Moore, A.W., 1991. An Introductory Tutorial on KD-Trees. Technical Report No. 209. Computer Laboratory, Carnegie Mellon University, Pittsburgh, Cambridge.
- Murphy, K., Russell, S., 2001. Rao-Blackwellized Particle Filtering for Dynamic Bayesian Networks. In: Doucet, A., Freitas, N., Gordon, N. (Eds.), Sequential Monte Carlo Methods in Practice. Springer-Verlag, p.499-515.
- Sim, R., Elinas, P., Griffin, M., Little, J., 2005. Vision-Based SLAM Using the Rao-Blackwellized Particle Filter. IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR). Edinburgh, Scotland.
- Stachniss, C., Grisetti, G., Burgard, W., 2005. Recovering Particle Diversity in a Rao-Blackwellized Particle Filter for SLAM after Actively Closing Loops. Proc. the IEEE Int. Conf. on Robotics and Automation (ICRA). Barcelona, Spain, p.667-672.