International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:3, 2008

Novel Rao-Blackwellized Particle Filter for Mobile Robot SLAM Using Monocular Vision

Maohai Li, Bingrong Hong, Zesu Cai and Ronghua Luo

***Abstract*—**This paper presents the novel Rao-Blackwellised particle filter (RBPF) for mobile robot simultaneous localization and mapping (SLAM) using monocular vision. The particle filter is combined with unscented Kalman filter (UKF) to extending the path posterior by sampling new poses that integrate the current observation which drastically reduces the uncertainty about the robot pose. The landmark position estimation and update is also implemented through UKF. Furthermore, the number of resampling steps is determined adaptively, which seriously reduces the particle depletion problem, and introducing the evolution strategies (ES) for avoiding particle impoverishment. The 3D natural point landmarks are structured with matching Scale Invariant Feature Transform (SIFT) feature pairs. The matching for multi-dimension SIFT features is implemented with a KD-Tree in the time cost of $O(\log_2^N)$. Experiment results on real robot in our indoor environment show the advantages of our methods over previous approaches.

***Keywords*—**Mobile robot, simultaneous localization and mapping, Rao-Blackwellised particle filter, evolution strategies, scale invariant feature transform.

I. Introduction

A key prerequisite for a truly autonomous robot is that it can simultaneously localize itself and accurately map its surroundings [1]. The problem of achieving this is one of the most active areas in mobile robotics research, which is known as Simultaneous Localization and Mapping (SLAM). One of the popular successful attempts at the SLAM problem was the extended Kalman filter (EKF)[2,3]. One of the limitations of the EKF is their computational complexity [4]. The standard EKF approach requires time quadratic in the number of features in the map for each incremental update. The other is that it requires that features in the environment be uniquely identifiable, otherwise this can cause excessive data association difficulty [5]. Recently, particle filters have been at the core of solutions to higher dimensional robot problems such as SLAM, which, when phrased as a state estimation problem. Murphy adopted Rao-Blackwellized particle filters (RBPF) [6] as an effective way of representing alternative hypotheses on robot paths and associated maps. Montemerlo et al. [7] 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 used widely to build map [8,9,10]. Dailey describe the application of FastSLAM using a trinocular stereo camera [11]. Se et al. [12] demonstrate the use of Scale Invariant Feature Transform (SIFT) point features as landmarks for the SLAM problem using a trinocular stereo camera. Davison et al. [13] demonstrate a single-camera SLAM algorithm capable of learning a set of 3D point features. Most of these vision-based methods use the stereo camera to obtain straightly the 3D feature, and the association problem either between features in successive camera frames or between observed features and map features is solved ambiguously.

In this paper we present an investigation into the use of monocular vision for SLAM in indoor environment with 3D feature landmarks, which are structured from the SIFT feature matching pairs. These 2D SIFT features are used to structure 3D landmarks because they 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 [14]. This combination can result in many highly distinctive landmarks from environment, which simplifies the data association problem to only distinguishing unique landmarks. We presents a fast and efficient algorithm for matching features in a KD-Tree in the time cost of $O(\log_2^N)$ [15]. Following [6,7], our approach applies RBPF to estimate a posterior of the path of the robot, where each particle has associated with it an entire map, in which each landmark is estimated and updated by the unscented Kalman filter (UKF) [16], and UKF is used to sample new poses that integrate the current observation which drastically reduces the uncertainty about the robot pose. Furthermore, the number of resampling steps is determined adaptively [17], which seriously reduces the particle depletion problem, and introducing the Evolution strategies (ES) for avoiding particle impoverishment [18]. All of these specialties can make data association in this paper more robust than other methods, and the built precise map only need a small number of particles.

The paper is organized as follows: In the next section, the RBPF for SLAM problem is briefly reviewed, and then the novel RBPF method is described in detail, and section 3

Manuscript received March 12, 2006. This research is supported by the National Natural Science Foundation of China (69985002) and the National Hi-Tech Research and Development Program of China (2002AA735041).

Maohai Li is with the Department of Computer Science and Techonology, Harbin Institute of Techonology, CO 150001 China (e-mail: limaohai@ hit.edu.cn).

Bingrong Hong is with the Department of Computer Science and Techonology, Harbin Institute of Techonology, CO 150001 China.

Zesu Cai is with the Department of Computer Science and Techonology, Harbin Institute of Techonology, CO 150001 China.

Ronghua Luo is with the Department of Computer Science and Techonology, Harbin Institute of Techonology, CO 150001 China.

provides a detailed its implementation for monocular vision-based SLAM in unknown indoor environment. Experiment results and discussions are presented in section 4 with conclusion in section 5.

II. Novel Rao-Blackwellized Particle Filter for SLAM

Consider the case of a mobile robot moving through an unknown environment consisted of a set of landmarks. The landmark n is denoted θ_n. The robot moves according to a known probabilistic motion model $p(s_t|u_t,s_{t-1})$, 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 z_t of its environment through observation model $p(z_t|s_t,\theta,n_t)$, where θ is the set of all landmarks and n_t is the index of the particular landmark observed at time t. The SLAM problem is to recover the posterior distribution $p(s^t,\theta_1,\ldots,\theta_M|z^t,u^t,n^t)$, where M is the number of landmarks observed so far and the notation s^t denotes $s_1,\ldots,s_t$(and similarly for other variables). In [6], Murphy et al. provide an implementation of RBPF for SLAM:

$$p(s^t,\theta_1,\ldots,\theta_M \mid z^t,u^t,n^t) = p(s^t \mid z^t,u^t,n^t)\prod_{n=1}^{M} p(\theta_n \mid s^t,z^t,n^t). \quad (1)$$

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, and the amount of computation needed for each incremental update stays constant, regardless of the path length. Each map is constructed given z^t and the trajectory s^t represented by the corresponding particle. Each particle is of the form $S_t^{(i)}=\{s^{t,(i)},\mu_{1,t}^{(i)},\Sigma_{1,t}^{(i)},\ldots,\mu_{M,t}^{(i)},\Sigma_{M,t}^{(i)}\}$, where ($i$) indicates the index of the particle; $s^{t,(i)}$ is its path estimate, $\mu_{m,t}^{(i)}$ and $\Sigma_{m,t}^{(i)}$ are the mean and variance of the Gaussian representing the m-th landmark location. Our novel RBPF update is performed in the following steps:

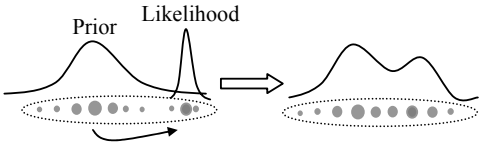


Fig. 1 Moving the samples in the prior to regions of high likelihood is important if the likelihood lies in one of the tails of the prior

A. Sampling New Poses Using UKF

Here we need to calculate the posterior over robot paths $p(s^t|u^t,z^t,n^t)$ 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 $p(s_t|u_t,s_{t-1})$ 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 showed 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.

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 the unscented transformation (UT) generated Gaussian approximation:

$$p(s_t \mid s^{t-1,(i)},u^t,z^t,n^t) \sim N(s_t;\tilde{s}_t^{(i)},P_t^{(i)}), \quad i=1,2,...,N. \quad (2)$$

UT can compute the mean and covariance up to the third order of the Taylor series expansion of the nonlinear observation function $g(\theta_{nt},s_t)$. Let L be the dimension of s_t, the UT computes mean and covariance as follows:

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

$$\chi_0 = \tilde{s}_t, \quad \chi_i = \tilde{s}_t + (\sqrt{(L+\lambda)P_{s_t}})_i, \quad i=1,...,L,$$
$$\chi_i = \tilde{s}_t - (\sqrt{(L+\lambda)P_{s_t}})_i, \quad i=L+1,...,2L. \quad (3)$$
$$W_0^m = \lambda/(L+\lambda), \quad W_0^c = W_0^m + (1-\alpha^2+\beta),$$
$$W_i^m = 1/(2\cdot(L+\lambda)), \quad i=1,...,2L, \quad \lambda=\alpha^2(L+\gamma)-L. \quad (4)$$

Where γ is a scaling parameter that controls the distance between the sigma points and the mean $\tilde{s}_t$, 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 0-th sigma point. α=0, β=0 and γ=2 are the optimal values for the scalar case. $(\sqrt{,(L+\lambda)P_{st}})_i$ is the i-th column of the matrix square root.

2) Propagate the sigma points through the nonlinear transformation:

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

3) Compute the mean and covariance of Z_i as follows:

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

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:

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

2) Using motion model to predict:

$$\chi^{*,(i)}_{t|t-1} = f(\chi^{(i)}_{t-1}, u^{(i)}_t), \quad \tilde{s}^{(i)}_{t|t-1} = \sum_{j=0}^{2L} W^{m,(i)}_j \chi^{*,(i)}_{j,t|t-1},$$
$$P^{(i)}_{t|t-1} = \sum_{j=0}^{2L} W^{c,(i)}_j [\chi^{*,(i)}_{j,t|t-1} - \tilde{s}^{(i)}_{t|t-1}][\chi^{*,(i)}_{j,t|t-1} - \tilde{s}^{(i)}_{t|t-1}]^T. \quad (8)$$

3) Incorporating new observation z_t:

$$Z^{*,(i)}_{t|t-1} = g(\chi^{*,(i)}_{t|t-1}, \theta_{n_t}), \quad \tilde{z}^{(i)}_{t|t-1} = \sum_{j=0}^{2L} W^{m,(i)}_j Z^{*(i)}_{j,t|t-1}. \quad (9)$$

$$P^{(i)}_{z_t z_t} = \sum_{j=0}^{2L} W^{c,(i)}_j [Z^{*,(i)}_{j,t|t-1} - \tilde{z}^{(i)}_{t|t-1}][Z^{*,(i)}_{j,t|t-1} - \tilde{z}^{(i)}_{t|t-1}]^T,$$
$$P^{(i)}_{s_t z_t} = \sum_{j=0}^{2L} W^{c,(i)}_j [\chi^{*,(i)}_{j,t|t-1} - \tilde{s}^{(i)}_{t|t-1}][Z^{*,(i)}_{j,t|t-1} - \tilde{z}^{(i)}_{t|t-1}]^T. \quad (10)$$

$$K^{(i)}_t = P^{(i)}_{s_t z_t} (P^{(i)}_{z_t z_t})^{-1}, \quad \tilde{s}^{(i)}_t = \tilde{s}^{(i)}_{t|t-1} + K^{(i)}_t (z^{(i)}_t - z^{(i)}_{t|t-1}),$$
$$P^{(i)}_t = P^{(i)}_{t|t-1} - K^{(i)}_t P^{(i)}_{z_t z_t} K^{(i)T}_t. \quad (11)$$

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

$$s^{(i)}_t \sim p(s_t \mid s^{t-1,(i)}, u^t, z^t) = N(s_t; \tilde{s}^{(i)}_t, P^{(i)}_t),$$
$$s^{t,(i)} = (s^{t-1,(i)}, s^{(i)}_t). \quad (12)$$

B. 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 $\tilde{\psi}_t$ 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: $\mu_{n,t}^{(i)} = \mu_{n,t-1}^{(i)}$, $\Sigma_{n,t}^{(i)} = \Sigma_{n,t-1}^{(i)}$. For the observed feature $n = n_t$, the update is specified through the following Equation:

$$p(\theta_{n_t} \mid s^{t,(i)}, n^t, z^t)$$
$$= \frac{p(z_t \mid \theta_{n_t}, s^{t,(i)}, n^t, z^{t-1}) p(\theta_{n_t} \mid s^{t,(i)}, n^t, z^{t-1})}{p(z_t \mid s^{t,(i)}, n^t, z^{t-1})}$$
$$= \eta \underbrace{p(z_t \mid \theta_{n_t}, s^{(i)}_t, n_t)}_{\sim N(z_t; g(\theta_{n_t}, s^{(i)}_t), R_t)} \underbrace{p(\theta_{n_t} \mid s^{t-1,(i)}, n^{t-1}, z^{t-1})}_{\sim N(\theta_{n_t}; \mu^{(i)}_{n_t,t-1}, \Sigma^{(i)}_{n_t,t-1})}. \quad (13)$$

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

1) Calculate the sigma points:

$$\xi^{(i)}_{n_t,t-1} = \{\mu^{(i)}_{n_t,t-1} \quad \mu^{(i)}_{n_t,t-1} \pm \sqrt{(L+\lambda)\Sigma^{(i)}_{n_t,t-1}}\}. \quad (14)$$

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

$$Z^{(i)}_{n_t,t} = g(\xi^{(i)}_{n_t,t-1}, s^{(i)}_t), \quad \bar{z}^{(i)}_{n_t,t} = \sum_{j=0}^{2L} W^{m,(i)}_j Z^{(i)}_{j,n_t,t},$$
$$P^{(i)}_{z_{n_t,t}} = \sum_{j=0}^{2L} W^{c,(i)}_j [Z^{(i)}_{j,n_t,t} - \bar{z}^{(i)}_{n_t,t}][Z^{(i)}_{j,n_t,t} - \bar{z}^{(i)}_{n_t,t}]^T. \quad (15)$$

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 measurement update:

$$K^{(i)}_t = \Sigma^{(i)}_{n_t,t-1} P^{(i)}_{z_{n_t,t}} (P^{(i)T}_{z_{n_t,t}} \Sigma^{(i)}_{n_t,t-1} P^{(i)}_{z_{n_t,t}} + R_t)^{-1},$$
$$\mu^{(i)}_{n_t,t} = \mu^{(i)}_{n_t,t-1} + K^{(i)}_t (z_t - \bar{z}^{(i)}_t)^T, \quad (16)$$
$$\Sigma^{(i)}_{n_t,t} = (I - K^{(i)}_t P^{(i)T}_{z_{n_t,t}}) \Sigma^{(i)}_{n_t,t-1}.$$

C. Adaptive Resampling

Next, we resample from temporary set of particles $\tilde{\psi}_t$, then form the new particle set ψ_t. Resampling is a common technique in particle filtering to correct for such mismatches, and avoiding particles degeneracy. By weighing particles in $\tilde{\psi}_t$, and resampling according to those weights, the resulting particle set indeed approximates the target distribution. After the resampling, all particle weights are then reset to $w_t^{(i)} = 1/N$. However, resampling can delete good particles from the sample set, in the worst case, the filter diverges. Accordingly, it is important to find a criterion when to perform a resampling step. Liu [19] introduced the so-called number of particles $N_{t,eff} = 1/\Sigma_{i=1}^{N} (w_t^{(i)})^2$ 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 $N_{t,eff}$. We resample each time $N_{t,eff}$ drops below a given threshold which was set to $0.6N$ 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 resampling operations are only performed when needed.

D. Introducing Evolution Strategy

The resampling step described before helps to avoid particle degeneracy, but also leads to an undesirable loss of particle diversity as resampling may result in multiple copies of only a few or, in the limit, only one particle. In this case, there is a severe depletion of samples. In order to introduce sample variety after resampling without affecting the validity of the approximation, we introduce the ES. Because the evolution operator can search for optimal particles, the sampling process is more efficient and the number of particles required to represent the posterior density can be reduced considerably. The two operators: crossover and mutation, work directly over the floating-points to avoid the trouble brought by binary coding and decoding. The crossover and mutation operator are defined as following:

Crossover: select two parent particles ($s_t^{(p1)}$, $w_t^{(p1)}$) and ($s_t^{(p2)}$, $w_t^{(p2)}$) randomly from population ψ_t, the crossover operator mates them by the following equation to generate two children particles:

$$\begin{cases} s_t^{(c1)} = \kappa s_t^{(p1)} + (1-\kappa) s_t^{(p2)} + \tau, & w_t^{(c1)} = p(z_t \mid s_t^{(c1)}) \\ s_t^{(c2)} = \kappa s_t^{(p2)} + (1-\kappa) s_t^{(p1)} + \tau, & w_t^{(c2)} = p(z_t \mid s_t^{(c2)}) \end{cases}. \quad (17)$$

Where $\kappa \sim U[0,1]$, $\tau \sim N(0,\Sigma)$, and $U[0,1]$ represents uniform distribution and $N(0,\Sigma)$ the normal distribution. Then, replace the parents $\{s_t^{(p1)}, s_t^{(p2)}\}$ by theirs children $\{s_t^{(c1)}, s_t^{(c2)}\}$ according to the following criterion: The child $s_t^{(c1)}$ would be accepted if $p(z_t|s_t^{(c1)}) > \max(p(z_t|s_t^{(p1)}), p(z_t|s_t^{(p2)}))$ value, else would be accepted with probability **Hata!Hata!**. In the similar form is accepted or rejected the child $s_t^{(c2)}$.

Mutation: select one parent particle $(s_t^{(p)}, w_t^{(p)})$, the mutation operator on it is defined as following:

$$s_t^{(c)} = s_t^{(p)} + \sigma, \quad w_t^{(c)} = p(z_t \mid s_t^{(c)}), \quad \sigma \sim N(0,\Sigma). \quad (18)$$

Then, the new particle $s_t^{(c)}$ is accepted if $p(z_t|s_t^{(c)}) > p(z_t|s_t^{(p)})$, else is accepted with probability $p(z_t|s_t^{(c)})/p(z_t|s_t^{(p)})$.

For more efficient, the crossover operator will perform adaptively with probability p_c and mutation operator will perform adaptively with probability p_m:

$$p_c = \begin{cases} p_{c1} - \dfrac{(p_{c1} - p_{c2})(f_c - f_{avg})}{f_{max} - f_{avg}}, & f_c \geq f_{avg} \\ p_{c1}, & f_c < f_{avg} \end{cases},$$

$$p_m = \begin{cases} p_{m1} - \dfrac{(p_{m1} - p_{m2})(f_{max} - f_m)}{f_{max} - f_{avg}}, & f_m \geq f_{avg} \\ p_{m1}, & f_m < f_{avg} \end{cases}. \quad (19)$$

Where f_{max} is the biggest fitness value in the population, and f_{avg} is the fitness average value, f_c is the bigger fitness value of two crossover individuals, f_m is the fitness value of mutation individual. In this paper, we set p_{c1}=0.85, p_{c2}=0.65, p_{m1}=0.1, p_{m2}=0.001.

III. Implementation Details of Using Monocular Vision

A. SIFT Feature Extraction

The Scale Invariant Feature Transform (SIFT) was proposed in [14] as a method of 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. 3) Orientation assignment. 4) Key-point descriptor. 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. The number of features generated is dependent on image size and content, as well as algorithm parameters. For a more detailed discussion see [14]. In this paper, we use the vectors with 128 elements as key-point descriptor. Fig. 2 shows an example of SIFT feature extraction.

B. KD-Tree Based Feature Matching

This section describes KD-tree algorithm for determining the matched SIFT feature pairs of successive images captured at relatively close positions along the robot's path by a monocular vision system. Given a SIFT key-points set E, and a target key-point vector d, then a nearest neighbor of d, d' is defined as:

$$\forall d'' \in E, | d \leftrightarrow d' | \leq | d \leftrightarrow d'' |, | d \leftrightarrow d' | = \sqrt{\sum_{i=1}^{k} (d_i \leftrightarrow d'_i)^2}. \quad (20)$$

Where d_i is the i-th component of d.

We implement the SIFT key-points matching algorithm which based on nearest neighbor search algorithm in a KD-tree,

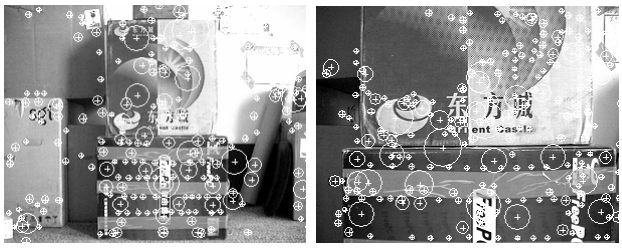


Fig. 2 Typical extracted SIFT features with their locations represented by '+'. The radius of the circle represents their scales: the 320×240 pixel test image taken at (a) 1618mm; (b) 756mm; and the result is (a) 278 key-points; (b) 267 key-points

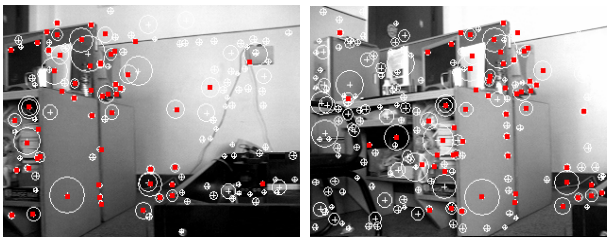


Fig. 3 The SIFT feature matches based on KD-tree, and the matching pairs are represented by red "·"

and the distance of the key-points is represented using the Euclidean distance between their according 128 dimensional descriptor vector (Equation 20), and we can use the following equation to judge the matching for two key-points:

$$| kp_1 \leftrightarrow kp | / | kp_2 \leftrightarrow kp | < \lambda. \quad (21)$$

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. Fig. 3 shows an example of SIFT feature matching for a pair image from the labor corner with different scale and direction, and we obtain 67 matched pairs which the matching accurate rate is higher than 80%.

C. 3D Structure

After the SIFT feature matching, we obtain the 2D SIFT image feature matching pairs along the robot's trajectory. In

this section, we use these feature pairs to structure the 3D spatial landmarks, which are in a single world model. Let $p_1(u_1,v_1)$ and $p_2(u_2,v_2)$ be the matching pair that observed from two different viewpoints, and p_1, p_2 associate the 3D spatial point landmark $P(X_w,Y_w,Z_w)$, as shown in Fig. 4, using the pinhole camera model:

$$z_{c1}\begin{bmatrix} u_1 & v_1 & 1 \end{bmatrix}^T = \mathrm{M}\begin{bmatrix} X_w & Y_w & Z_w & 1 \end{bmatrix}^T. \quad (22)$$

$$z_{c2}\begin{bmatrix} u_2 & v_2 & 1 \end{bmatrix}^T = \mathrm{M}\begin{bmatrix} X_w & Y_w & Z_w & 1 \end{bmatrix}^T. \quad (23)$$

The solution of three unknown variants X_w, Y_w and Z_w can be obtained through the least square method, and the projection matrix M:

$$\mathrm{M} = \begin{bmatrix} \alpha_x & 0 & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{R} & \mathrm{T} \\ 0^T & 1 \end{bmatrix}. \quad (24)$$

Where motion model provides extrinsic camera rotations R and translations T for each image. Offline calibration [23] yields the camera's intrinsic parameters α_x, α_y, u_0, v_0 as shown in Table 1.

D. Motion Model

The motion model $p(s_t|u_t,s_{t-1})$ predicts the movement and status over time of the robot. As shown in previous methods,

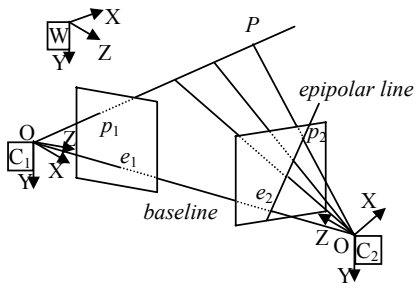


Fig. 4 Two viewpoints geometry and the epipolar constraint

TABLE I
THE INTRINSIC PARAMETERS AND EXTRINSIC PARAMETERS OF CAMERA

Intrinsic Parameters		Extrinsic Parameters	
α_x	368.82620	x_{cr}	2.1039 mm
α_y	369.90239	z_{cr}	100.1742 mm
u_0	159.67029	θ_{cr}	90°
v_0	121.54136		

when a control u, consisting of forward and angular velocity is applied to the robot:

$$\begin{aligned} p(s_t \mid u_t, s_{t-1}) &= f(u_t, s_{t-1}) + \varepsilon_t \\ &= \begin{bmatrix} x_{t+1}^i \\ y_{t+1}^i \\ \varphi_{t+1}^i \end{bmatrix} + \varepsilon_t = \begin{bmatrix} x_t^i \\ y_t^i \\ \varphi_t^i \end{bmatrix} + \begin{bmatrix} v\Delta T \cos(\varphi_t^i + \omega_t \Delta T) \\ v\Delta T \sin(\varphi_t^i + \omega_t \Delta T) \\ \varphi_t^i + \omega_t \Delta T \end{bmatrix} + \varepsilon_t . \end{aligned} \quad (25)$$

Where $(x_t^i, y_t^i, \varphi_t^i)$ is the robot's location and bearing at time t, for all particles $i=1,\ldots,N$, v_t is the line velocity, ω_t is the angular velocity at time t. ΔT is the time step and ε_t are noise in terms of a normal distribution $N(0,P_t)$.

E. Observation Model

Every time the robot is triggered, the CCD camera vision system captures the consecutive digital images and after SIFT feature extracting, matching current observed SIFT feature with the map database contained with 3D spatial natural landmarks through KD-tree based nearest neighbor search algorithm. Let $F_t=\{f_1,\ldots, f_k\}$ be the k SIFT feature key-points observed at time t, in which there are n key-points matching with the 3D landmarks in the map database: $n_t^l=\{f_1\sim L_{f1},\ldots, f_n\sim L_{fn}\}$, and there are m key-points matching the 2D SIFT feature key-points which observed at time t-1 and are not reconstructed and added to the map database: $n_t^v=\{f_{n+1}\sim V_{fn+1},\ldots,f_{n+m}\sim V_{fn+m}\}$. Then the likelihood of the observation z_t being obtained is:

$$p(z_t \mid s_t^{(i)}, \theta, n_t) = p(z_t^l \mid s_t^{(i)}, \theta, n_t^l)\, p(z_t^v \mid s_t^{(i)}, \theta, n_t^v) \quad (26)$$

Where z_t^l represents the observation $F_l=\{f_1,\ldots,f_n\}$, and z_t^v represents the observation $F_v=\{f_{n+1},\ldots,f_{n+m}\}$, $p(z_t^l|s_t^{(i)},n_t^l)$ represents the likelihood of the observation z_t^l given the matching relation n_t^l, and $p(z_t^l|s_t^{(i)},n_t^v)$ represents the likelihood of the observation z_t^v given the matching relation n_t^v, these two likelihood can be calculated separately as follows:

$$\ln p(z_t^l \mid s_t^{(i)}, \theta) = \sum_{j=1}^{n} \ln p(f_j \mid s_t^{(i)}, L_{fj}) \quad (27)$$

$$\ln p(z_t^v \mid s_t^{(i)}, \theta) = \sum_{j=n+1}^{n+m} \ln p(f_j \mid s_t^{(i)}, V_{fj}) \quad (28)$$

Where $p(f_j|s_t^{(i)},L_{fj})$ represents the likelihood of the observation being f_j when robot at pose $s_t^{(i)}$ observing the landmark L_{fj}, and $p(f_j|s_t^{(i)},V_{fj})$ represents the likelihood of the observation being f_j when robot at pose $s_t^{(i)}$ observing the SIFT feature V_{fj}. Let the 3D coordinates of the landmark L_{fj} be $(x_w^{(j)},y_w^{(j)},z_w^{(j)})$, then we can obtain $\ln p(f_j|s_t^{(i)},L_{fj})$ as follows:

$$\begin{aligned} &\ln p(f_j \mid s_t^{(i)}, L_{fj}) = -0.5 \min(T_l, (\hat{I}_j - I_j)^T S^{-1} (\hat{I}_j - I_j)), \\ &S = J(R_t G_{fj} R_t^T) J^T . \end{aligned} \quad (29)$$

Where J is the Jacobian matrix of the observation equation, G_{fj} is the covariance of L_{fj}. The maximum observation innovation T_l is constant (in our case, 3.0), which is selected so as to prevent outlier observations from significantly affecting the observation likelihood.

While the feature V_{fj} has no 3D spatial information, $\ln p(f_j|s_t^{(i)},V_{fj})$ is only calculated according to epipolar constraint:

$$\ln p(f_j \mid s_t^{(i)}, V_{fj}) = -0.5(dist(I_j, H_{fj}) + dist(I_{fj}, H_j)). \quad (30)$$

Where I_{fj} is the image coordinate of the feature V_{fj} , H_{fj} is the epipolar line on the image plane corresponding to V_{fj} 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.

After calculating the observation model $p(z_t|s_t,\theta,n_t)$, which can be used to evaluate the i-th particle weight $w_t^{(i)}$, and $w_t^{(i)}$ is taken as the fitness value in evolutionary process:

$$w_t^{(i)} = \frac{p(z_t \mid s_t^{(i)}, \theta, n_t)}{\sum_{j=1}^{N} p(z_t \mid s_t^{(i)}, \theta, n_t)}. \quad (31)$$

IV. Experimental Results and Discussion

The experiments are performed on a Pioneer 3-DX mobile robot incorporating an 800 MHz Intel Pentium processor as shown in Fig. 5.a. Motor control is performed on the on-board computer, while a 3 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 mounted at the front of the robot is used for detecting the landmarks. The test environment is a robot laboratory with limited space shown in Fig. 5.b.

For illustrating the advantages of our methods over previous approaches, we implement SLAM with our novel RBPF and previous method. The experiment is described as follows.

(a) (b)

Fig. 5 (a) Pioneer 3 mobile robot; (b) experimental environment

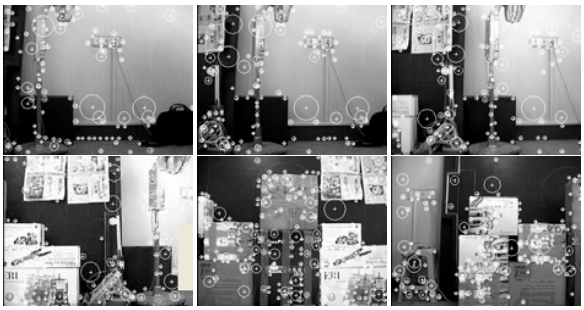
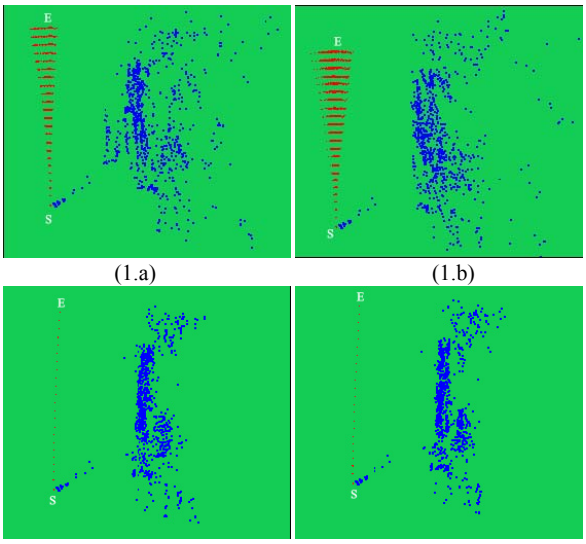


Fig. 6 The image sequence of the wall

Firstly, the robot is set at the distance of 2m from the lab wall, and the robot orientation is parallel with the wall, at the same time, let the CCD camera vision face with the wall. While the robot is moving ahead, the image frames are captured and processed, building the map of the wall. Fig. 6 shows some frames of size 320×240 (38 frames in total). At the end, a total of 1468 SIFT landmarks with 3D positions are gathered in the map, which are relative to the initial coordinates frame.



(1.a) (1.b)

(2.a) (2.b)

Fig. 7 Experiment results of map building based on (1) conventional RBPF: (a) 100 particles, (b) 500 particles; (2) novel RBPF: (a) 50 particles, (b) 100 particles

Fig. 7 shows the experiment results. In the map, 'S' represents the start point of robot path, and 'E' represents the end point of robot path, the red point represents the path particle, the 2D view of 3D landmarks in the map is represented with blue points. As shown in Fig. 7 (1), if we increase the number of particles, the performance of conventional RBPF will be improved largely, however, the storage requirement and calculation burden is severely aggravated, owning to each particle associated with a view of the map. Fig. 7 (2) shows the built map with the novel RBPF, which adopts separately 50 particles and 100 particles, and 8 evolutionary steps. For executing the evolution strategies, the most particles can be convergent to the region high weight, and approximate the posterior only with few particles. The performance of the novel RBPF changes a little with increasing the number of particles, specifically, we can build precise map only with few particles. The more detail comparison of performance with different numbers of particles is shown as Fig. 8, obviously, the robot pose and landmark estimation error is largely reduced, and we only need a few particles to reach remarkable results by means of incorporating current observation and thinking about evolution strategy and adaptive resampling, as well as the effective management structure based on Kd-tree. However, ES step can aggravate the computation burden, this negative impact can be largely reduced for less and less particles with the running process. The results are compared with previous methods indicate superior performance of presented method.

Another experiment was carried out in our single lab room, where the compact map is built with our method, and 186 image frames of size 320×240 are captured. Fig. 9 shows the bird's-eye view of the 3D spatial map.

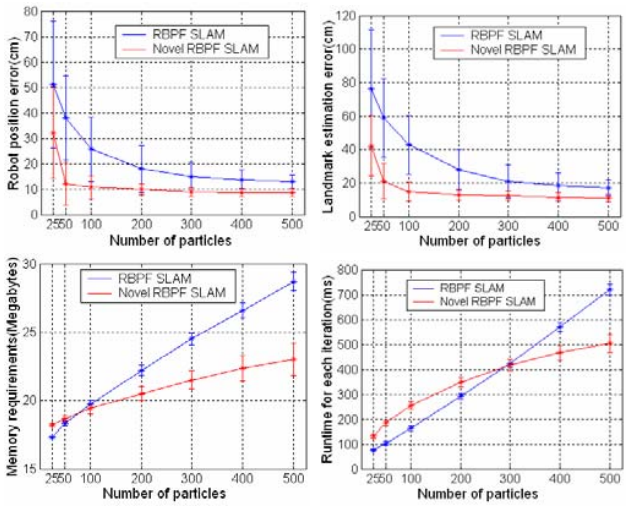


Fig. 8 Results of our novel RBPF SLAM algorithm compared with conventional RBPF

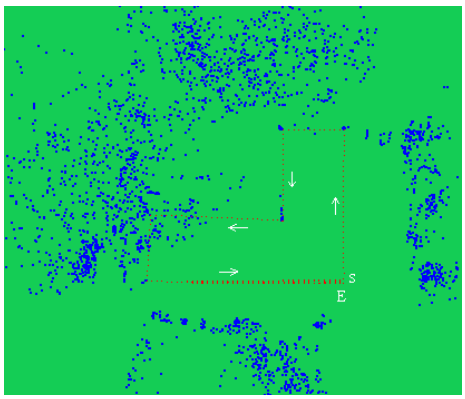


Fig. 9 Bird's-eye view of the SIFT landmarks in the map. 'S' indicates the initial robot position, 'E' indicates the path end, the dot line indicates the estimated robot path and '→' indicates the robot moving direction

V. Conclusion

This article described a novel algorithm for SLAM problem using monocular CCD camera. Like many previously published SLAM algorithms, our method calculates posterior probability distributions over 3D SIFT featured maps and robot locations. It does so recursively based on a key property of the SLAM problem: the conditional independence of feature estimates given the vehicle path. This conditional independence gives rise to a factored representation of the posterior using a combination of particle filters for estimating the robot path and UKF for estimating the map. Furthermore, the number of resampling steps is determined adaptively, which seriously reduces the particle depletion problem, and introducing ES step after the resampling for avoiding particle impoverishment. Experiment results on real robot in our indoor environment show the advantages of our methods over previous approaches.

References

[1] D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors, *AI-based Mobile Robots: Case studies of successful robot systems*, MIT Press, Cambridge, 1998, pp. 91–122.
[2] R. C. Smith, P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," *International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.
[3] J. Leonard, J. D. Tard´os, S. Thrun, and H. Choset, editors, Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots, in *Proc. IEEE Int. Conf. Robotics and Automation*, Washington, DC, 2002.
[4] J. E. Guivant, E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.
[5] A. J. Davison and D. W. Murray, "Simultaneous localization and map building using active vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865─880, 2002.
[6] K. Murphy and S. Russell, "Rao-blackwellized particle filtering for dynamic bayesian networks," in *Sequential monte carlo methods in practice*, Springer Verlag, 2001.
[7] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *Proc. IEEE Int. Conf. Robotics and Automation*, Taipei, 2003.
[8] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. of The Ninth Int. Conf. on Computer Vision ICCV'03*, Nice, France, 2003, pp. 1403─1410.
[9] C. Stachniss, G. Grisetti, and W. Burgard, "Recovering Particle Diversity in a Rao-Blackwellized Particle Filter for SLAM After Actively Closing Loops," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2005, pp. 667–672, Barcelona, Spain.
[10] R. Sim, P. Elinas, M. Griffin, and J. Little, "Vision-based SLAM using the Rao-Blackwellized Particle Filter," in *Workshop Reasoning with Uncertainty in Robotics*, Edinburgh, Scotland, 2005.
[11] M. N. Dailey and M. Parnichkun, "Landmark-based simultaneous localization and mapping with stereo vision," in *Proc. of the 2005 Asian Conf. on Industrial Automation and Robotics*, 2005.
[12] S. Se, D. Lowe, and J. Little, Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks, International Journal of Robotics Research, 21(8): 735–758, 2002.
[13] A. Davison, Y. Cid, and N. Kita, "Real-time 3D SLAM with wide-angle vision," in *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.
[14] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[15] A. W. Moore, "An introductory tutorial on kd-trees," Robotics Institute, Carnegie Mellon University, Pittsburgh, Technical Report No. 209, Computer Laboratory, University of Cambridge, 1991.
[16] R. Merwe, A. Doucet, N. Freitas, and E. Wan, "The Unscented Particle Filter," Technical Report CUED/FINFENG /TR 380, Cambridge University, Engineering Department, 2000.
[17] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Technical report, Signal Processing Group, Departement of Engeneering, University of Cambridge, 1998.
[18] T. Duckett, "A genetic algorithm for simultaneous localization and mapping," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2003, pp. 434–439.
[19] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamical systems," *J. Amer. Statist. Assoc.*, vol. 93, pp. 1032–1044, 1998.
[20] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *Proc. ICCV*, pp. 666–671, 1999.