# Novel Mobile Robot Simultaneous Loclization and Mapping Using Rao-Blackwellised Particle Filter

**Li Maohai[1]; Hong Bingrong[1] & Luo Ronghua[2]**
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[2]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China
limaohai@163.com

***Abstract:*** *This paper presents the novel method of mobile robot simultaneous localization and mapping (SLAM), which is implemented by using the Rao-Blackwellised particle filter (RBPF) for monocular vision-based autonomous robot in unknown indoor environment. The particle filter is combined with unscented Kalman filter (UKF) to extending the path posterior by sampling new poses that integrate the current observation. The landmark position estimation and update is implemented through the unscented transform (UT). Furthermore, the number of resampling steps is determined adaptively, which seriously reduces the particle depletion problem. Monocular CCD camera mounted on the robot tracks the 3D natural point landmarks, which are 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 in the time cost of O(log2$^N$). Experiments on the robot Pioneer3 in our real indoor environment show that our method is of high precision and stability.*

***Keywords:*** *mobile robot, simultaneous localization and mapping, Rao-Blackwellised particle filter, vision, Scale Invariant Feature Transform*

## 1. 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). Particle filters provide an attractive approach for updating distributions of data (Doucet, 1998). Early successes of particle filters can be found in the area of robot localization (Dellaert et al, 1999). 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, involves a variable number of dimensions. Murphy adopted Rao-Blackwellized particle filters (RBPF) ( Murphy, 2001) as an effective way of representing alternative hypotheses on robot paths and associated maps. Montemerlo et al. (Montemerlo & 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.

The difficulty of the SLAM depends on the robot's environment, its sensors, and the representation of map. The environment could be relatively benign indoors with flat floors. But it could also be quite subversive such as aircraft and submarines. The most common sensors in use are sonar sensors, laser range finders and video cameras. Sonar readings are susceptible to high degrees of uncertainty especially due to angular and radial errors. Lasers are accurate while they are heavy, expensive. Sonar and lasers are primarily used for 2D map. On the other hand, cameras are light, cheap, and can provide abundant environmental information, but are difficult to work with. Popular choices for the map representation include grid-based (Schultz & Adams, 1998), topological (Choset & Nagatani, 2001) and feature based models (Chong & Kleeman, 1999). Grid-based models are easy to build and maintain while implies high data requirements and induces high computational costs. Topological maps usually have the advantage of being compact, and more tolerant to errors in the robot location. Feature based representations have been difficult to build while being significantly less complex.

We primarily focus on investigating real-time, monocular vision based SLAM for indoor environments, and constructing 3D feature map from video data. Scale invariant features are extracted through Scale Invariant Feature Transform (SIFT) (Lowe, 2004), which are used to structure 3D landmarks because they are invariant to image scale, rotation and translation as well as partially invariant to illumination changes. We presents a fast and efficient algorithm for matching features in a KD-Tree in the time cost of *O*(log2$^N$) (Moore, 1991). RBPF is used 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 transform (UT) (Merwe et al, 2000), and unscented Kalman filter (UKF) is used to sample new poses that integrate the current observation. Furthermore, the number of resampling steps is determined adaptively, which seriously reduces the particle depletion problem. All of these specialties can make data association in this paper more robust than other methods. Experiment results are compared with those of the EKF methods applied to the same robot in the same environment and indicate superior performance.

## 2. Background

Consider the case of a mobile robot moving through an unknown environment consisting of a set of landmarks $\theta$. 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 $s^t$ denotes $s_t$,…$s_t$(and similarly for other variables). In (Doucet *et al.*, 2000) Doucet et al. provide an implementation of RBPF for SLAM:

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

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. 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.
A successful instance of the RBPF SLAM is FastSLAM, which offers many improvements over the traditional EKF-based SLAM framework: it has excellent time complexity; it does not need to linearize the robot's motion model; especially it can maintain several data association hypotheses. However, FastSLAM also has drawbacks: each particle has a different view of the map, integrating these views to obtain a single map is nontrivial, and more importantly, data association must be performed for each particle independently, which introduces a significant computational burden; FastSLAM is prone to diverge in regions where its measurements are not very informative, either due to high noise or the sparseness of landmarks.

## 3. Novel Rao-Blackwellized Particle Filter for 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 $S_t^{(i)}$={ $s^{t,(i)}$, $\mu_{1,t}^{(i)}$, $\sum_{1,t}^{(i)}$,…, $\mu_{M,t}^{(i)}$, $\sum_{M,t}^{(i)}$}, where ($i$) indicates the index of the particle; $s^{t,(i)}$ is its path estimate, and $\mu_{m,t}^{(i)}$ and $\sum_{m,t}^{(i)}$ are the mean and variance of the Gaussian representing the $m$-th landmark location. 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:

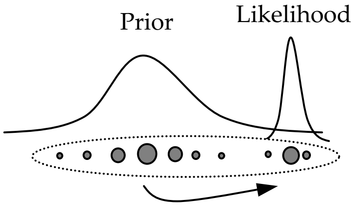


Fig. 1. Moving the samples in the prior to regions of high likelihood is important if the likelihood lies in one of the tails of the prior.

### *3.1. 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.
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:

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

EKF approximates the distribution through the first-order Taylor-series expansion of the nonlinear observation function $z_t=g(\theta_{nt}, s_t)$ 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 $\underline{z}_t=g(\theta_{nt}, \underline{s}_t)$, $P_{zt}=g'(\theta_{nt}, \underline{s}_t)^T P_{st}\, g'(\theta_{nt}, \underline{s}_t)$ 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 $g(\theta_{nt}, s_t)$ (Merwe et al, 2000). Let $L$ be the dimension of $s_t$, the UT computes mean and covariance as follows:

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

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

where $\gamma$ is a scaling parameter that controls the distance between the sigma points and the mean $\underline{s}_t$, $\alpha$ is a positive scaling parameter that controls the higher order effects resulted from the non-linear function g, $\beta$ is a parameter that controls the weighting of the 0-th sigma point $\alpha$=0, $\beta$=0 and $\gamma$=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. Note that the 0-th sigma point's weight is different for calculating mean and covariance.

2) Propagate the sigma points through the nonlinear transformation:

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

3) Compute the mean and covariance as follows:

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

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

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

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

2) Using motion model to predict:

$$\chi_{t|t\text{-}1}^{*,(i)} = f(\chi_{t-1}^{(i)}, u_t^{(i)}) \qquad \underline{s}_{t|t-1}^{(i)} = \sum_{j=0}^{2L} W_j^{m,(i)} \chi_{j,t|t\text{-}1}^{*,(i)}$$
$$P_{t|t-1}^{(i)} = \sum_{j=0}^{2L} W_j^{c,(i)} [\chi_{j,t|t\text{-}1}^{*,(i)} \text{-} \underline{s}_{t|t-1}^{(i)}][\chi_{j,t|t\text{-}1}^{*,(i)} \text{-} \underline{s}_{t|t-1}^{(i)}]^T \tag{8}$$

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

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

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

$$s_t^{(i)} \sim p(s_t | s^{t-1,(i)}, u^t, z^t) = N(s_t; \underline{s}_t^{(i)}, P_t^{(i)})$$
$$s^{t,(i)} = (s^{t-1,(i)}, s_t^{(i)}) \tag{10}$$

### 3.2. *Updating the observed landmark estimate*

In this step, we update the posterior over the landmark estimates represented by the mean $\mu_{n,t\text{-}1}^{(i)}$, and the covariance $\sum_{n,t\text{-}1}^{(i)}$. The updated values $\mu_{n,t}^{(i)}$ and $\sum_{n,t}^{(i)}$ 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:

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

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

1) Calculate the sigma points:

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

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

$$Z_{n_t,t}^{(i)} = g(\xi_{n_t,t-1}^{(i)}, s_t^{(i)}) \qquad \underline{z}_{n_t,t}^{(i)} = \sum_{j=0}^{2L} W_j^{m,(i)} Z_{j,n_t,t}^{(i)}$$
$$P_{z_{n_t,t}}^{(i)} = \sum_{j=0}^{2L} W_j^{c,(i)} [Z_{j,n_t,t}^{(i)} \text{-} \underline{z}_{n_t,t}^{(i)}][Z_{j,n_t,t}^{(i)} \text{-} \underline{z}_{n_t,t}^{(i)}]^T \tag{13}$$

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:

$$K_t^{(i)} = \Sigma_{n_t,t-1}^{(i)} P_{z_{n_t,t}}^{(i)} (P_{z_{n_t,t}}^{(i)\,T} \Sigma_{n_t,t-1}^{(i)} P_{z_{n_t,t}}^{(i)} + R_t)^{-1}$$
$$\mu_{n_t,t}^{(i)} = \mu_{n_t,t-1}^{(i)} + K_t^{(i)} (z_t - \underline{z}_t^{(i)})^T \tag{14}$$
$$\Sigma_{n_t,t}^{(i)} = (I - K_t^{(i)} P_{z_{n_t,t}}^{(i)\,T}) \Sigma_{n_t,t-1}^{(i)}$$

### 3.3. *Selective resampling*

Next, we resample from temporary particles set $\check{S}_t$, then form the new particle set $S_t$. The necessity to resample arises from the fact that the particles in $\check{S}_t$ do not yet match the desired posterior. Resampling can avoid particles degeneracy. By weighing particles in $\check{S}_t$, and resampling according to those weights, the resulting particle set indeed approximates the target distribution. To determine importance weight of each particle, it will prove useful to calculate the actual proposal distribution of the path particles in $\check{S}_t$. Under the assumption that the set of path particles in $S_{t\text{-}1}$ is distributed according to $p(s^{t\text{-}1} | z^{t\text{-}1}, u^{t\text{-}1}, n^{t\text{-}1})$, path particles in $\check{S}_t$ are distributed as:

$$p(s^{t,(i)} | z^{t-1}, u^t, n^{t-1})$$
$$= p(s^{t,(i)} | s_{t-1}^{(i)}, u^t) p(s^{t-1,(i)} | z^{t-1}, u^{t-1}, n^{t-1}) \tag{15}$$

Target distribution $p(s^{t,(i)} | z^t, u^t, n^t)$ takes into account the measurement $z_t$ along with the correspondence $n_t$. The importance weight of resampling process accounts for the difference of the target and the proposal distribution, which is given by the quotient of the target and the proposal distribution, applying Bayes rule and Markov assumption and omitting the irrelevant variables:

$$w_t^{(i)} = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(s^{t,(i)} \mid z^t, u^t, n^t)}{p(s^{t,(i)} \mid z^{t-1}, u^t, n^{t-1})} \qquad (16)$$
$$= \eta \int p(z_t \mid s_t^{(i)}, \theta_{n_t}, n^t) p(\theta_{n_t} \mid s^{t-1,(i)}, z^{t-1}, n^{t-1}) d\theta_{n_t}$$

To calculate $w_t^{(i)}$ in closed form, we employ the very same approximation used in the measurement update. In particular, the weight is given by

$$w_t^{(i)} \approx \eta \left| 2\pi L_t^{(i)} \right|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}(z_t - \underline{z}_t^{(i)})^T L_t^{(i)-1} (z_t - \underline{z}_t^{(i)}) \right\} \qquad (17)$$
$$L_t^{(i)} = G_t^{(i)T} \Sigma_{n,t-1}^{(i)} G_t^{(i)} + R_t$$

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 (Liu and Chen, 1998) introduced the so-called number of particles $N_{t,eff}$ $=1/\sum^{N,i=1}(w_t^{(i)})^2$ 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 0.5$N$ 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.

## 4. Implementation Details for Monocular Vision

### *4.1. SIFT Feature Extraction*

The Scale Invariant Feature Transform (SIFT) was proposed in (Lowe, 2004) 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. The first stage finds scale-space extrema located in Difference of Gaussians (DOG) function $D(x,y,\theta)$, which can be computed from the difference of two nearby scaled images separated by a multiplicative factor $k$:

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \qquad (18)$$
$$= L(x,y,k\sigma) - L(x,y,\sigma)$$

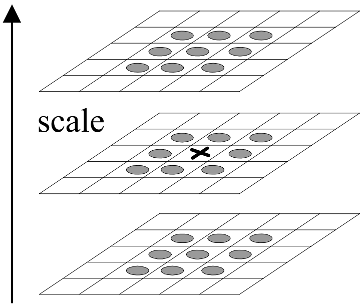


Fig. 2. Maxima and minima are detected by comparing a pixel (marked with X) to its 26 neighbors in 3× 3 regions at the current and adjacent scales (marked with circles).

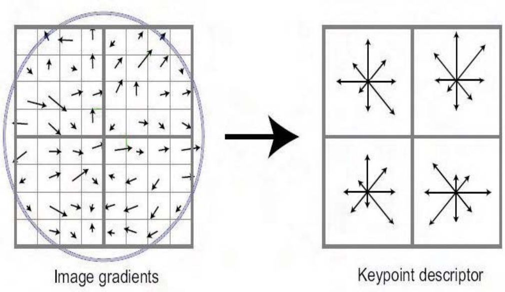


Fig. 3. A key-point descriptor.

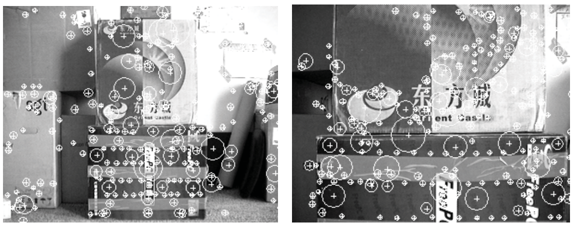


Fig. 4. Typical extracted SIFT features with their locations represented by '+'. The radius of the circle represents their scales: the 320×240 pixel test image taken at (a) 1618mm; (b) 756mm; and the result is (a) 278 key-points; (b) 267 key-points.

where $L(x,y,\sigma)$ is the scale space of an image, built by convolving the image $I(x,y)$ with the Gaussian kernel $G(x,y,\sigma)$. 2) Key-point localization. The location and scale of each candidate point is determined and key-points are selected based on measures of stability(Fig. 2). 3) Orientation assignment. One or more orientations are assigned to each key-point based on local image gradients. For each image sample $L(x,y)$ at this scale, the gradient magnitude $m(x,y)$ and orientation $\theta(x,y)$ is computed using pixel differences :

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$
$$\theta(x,y) = \tan^{-1}((L(x+1,y) - L(x-1,y)) / (L(x,y+1) - L(x,y-1))) \qquad (19)$$

4) Key-point descriptor. Typical key-point descriptors use 16 orientation histograms aligned in a 4× 4 grid. Each histogram has 8 orientation bins each created over a support window of 4× 4 pixels (Fig. 3). The resulting feature vectors are 128 elements with a total support window of 16× 16 scaled pixels. For a more detailed discussion see (Lowe, 2004). The number of features generated is dependent on image size and content, as well as algorithm parameters. In this paper, we use the vectors with 128 elements as key-point descriptor. Fig. 4 shows an example of SIFT feature extraction for a cluttered and occluded image of size 320×240 pixels.

### *4.2. KD-Tree based Feature Matching*

This section describes KD-tree algorithm for determining the matching SIFT features pairs of successive images captured at relatively close positions along the robot's path 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:

$$\forall d'' \in E, | d \Leftrightarrow d' | \le | d \Leftrightarrow d'' |$$
$$| d \Leftrightarrow d' | = \left(\sum_{i=1}^{k} (d_i \Leftrightarrow d'_i)^2\right)^{1/2} \qquad (20)$$

where $d_i$ is the *i*-th component of *d*. The KD-tree based SIFT feature matching algorithm is described as following: Constructing a KD-tree. The pivot-choosing procedure chooses a good vector from *E* to use as the tree's root, which is desirable for the tree to be reasonably balanced, and also for the shapes of the hyper-regions corresponding to leaf nodes to be fairly equally proportioned. Our pivoting strategy is to pick the splitting dimension *split* firstly: for each *i* dimension, compute the maximal value $max_i$ and the minimal value $min_i$ from the *i*-th dimension element of every key-point in set *E*, and choose the according dimension which has the most spread $max_i$-$min_i$, its median value $med_i$. Secondly, we pick the key-point provided with the minimal value between the *split*-th dimension and $med_i$ as tree's root.

After constructing the KD-tree, the nearest neighbor search algorithm which is depth first is used to search the child node which contains the target. The space occupied by set *E* is represented by a hyper-rectangle composed of two arrays: one of its minimum coordinates, the other of its maximum coordinates. To cut the hyper-rectangle, so that one of its edges is moved closer to its centre, the appropriate array component is altered. To check to see if a hyper-rectangle *hr* intersects with a hyper-sphere radius *r* centered a point *target*, we find the point *p* in *hr* which is closest to *target*. Write $hr_i^{min}$ as the minimum extreme of *hr* in the *i*-th dimension and $hr_i^{max}$ as the maximum extreme $p_i$, the *i*-th component of this closest point is computed thus:

$$p_i = \begin{cases} hr_i^{min} & \text{if } target_i \le hr_i^{min} \\ target_i & \text{if } hr_i^{min} < target_i < hr_i^{max} \\ hr_i^{max} & \text{if } target_i \ge hr_i^{max} \end{cases} \qquad (21)$$

The object intersect only if the distance between *p* and *target* is less than or equal to *r*.

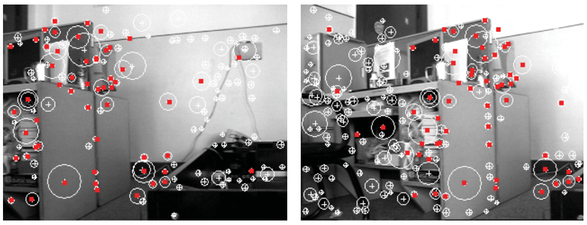


Fig. 5. The SIFT feature matches based on KD-tree, and the matching pairs are represented by red "·".

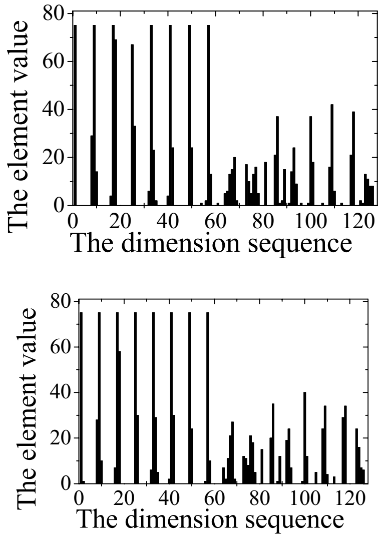


Fig. 6. The key-point descriptor histograms of the matching key-point at different scale and direction.

We implement the SIFT key-points matching algorithm based on nearest neighbor algorithm in a KD-tree, and the distance of the key-points is represented using the Euclidean distance between their according 128 dimensional descriptor vector. The basic process for matching is 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 $|kp_1 \equiv kp| / |kp_2 \equiv kp|$ 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:

$$| kp_1 \Leftrightarrow kp | / | kp_2 \Leftrightarrow kp | < \lambda \qquad (22)$$

where $\lambda$ is constant, and $0<\lambda<1$ (in this paper $\lambda$ is evaluated as 0.7), if this equation is satisfied, then the matching is successful, and simultaneously eliminates the false matching. Fig. 5 shows an example of SIFT feature matching, and the matched accurate rate is higher than 80%. Fig. 6 shows the key-point descriptor histograms of one matching pair at different scale and direction, which proves the robust matching algorithm.

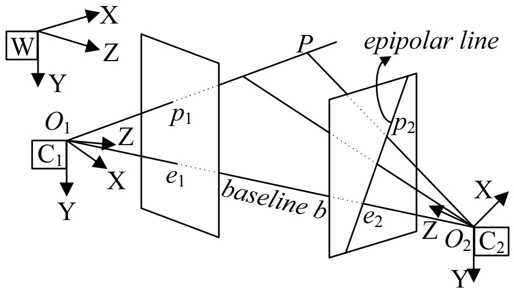


Fig. 7. Two viewpoints geometry and the epipolar constraint.

### *4.3. 3D Structure*

After the SIFT feature matching, we obtain the 2D SIFT image feature matching pairs used to structure the 3D spatial landmarks, which are in a single world model. As seen from Fig. 7, According to the epipolar constraint, all the entities $P$, $O_1$, $O_2$, $p_1$, $p_2$, $e_1$, $e_2$, $b$ should be coplanar. Through epipolar constraint, the matches with large error are eliminated. Let $f$ be the focus of the CCD camera. The relationship between a 3D point $P(X_w,Y_w,Z_w)$ and the image coordinates $p(u,v)$ where it is projected is given by the pinhole camera model (Ma & Zhang, 1998):

$$Z_c\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}\alpha_x&0&u_0&0\\0&\alpha_y&v_0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}\mathrm{R}&\mathrm{T}\\0^T&1\end{bmatrix}\begin{bmatrix}X_w\\Y_w\\Z_w\\1\end{bmatrix}=\mathrm{M}\begin{bmatrix}X_w\\Y_w\\Z_w\\1\end{bmatrix} \qquad (22)$$

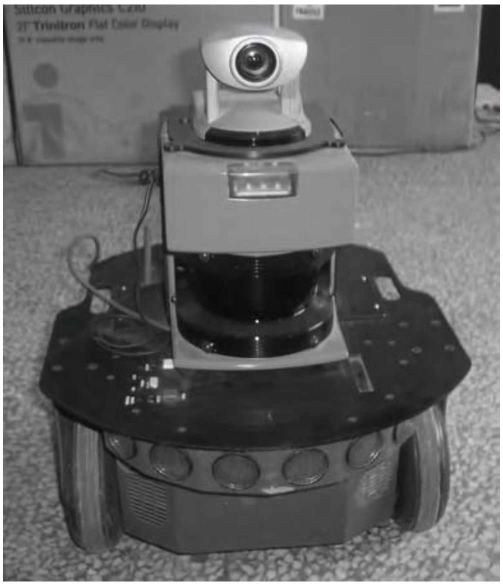
where robot motion provides extrinsic camera rotations R and translations T for each image. Offline calibration yields the camera's intrinsic parameters $\alpha_x$, $\alpha_y$, $u_0$, $v_0$. For any pair of matching points $p_1(u_1,v_1,1)$ and $p_2(u_2,v_2,1)$ corresponding to a 3D point $P(X_w,Y_w,Z_w)$, using the pinhole camera model:

$$z_{c1}\left[u_1\ v_1\ 1\right]^T=\mathrm{M}\left[X_w\ Y_w\ Z_w\ 1\right]^T,\ z_{c2}\left[u_2\ v_2\ 1\right]^T=\mathrm{M}\left[X_w\ Y_w\ Z_w\ 1\right]^T \qquad (23)$$

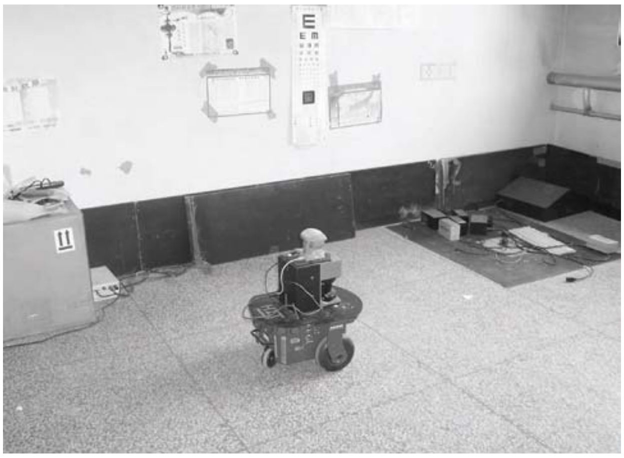
The solution of three unknown variants $X_w$, $Y_w$ and $Z_w$ can be obtained through the least square method.

## 5. 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. 8(a). Motor control is performed on the on-board computer, while a 2.6GHz 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. The test environment is a robot laboratory with limited space as shown in Fig. 8(b).

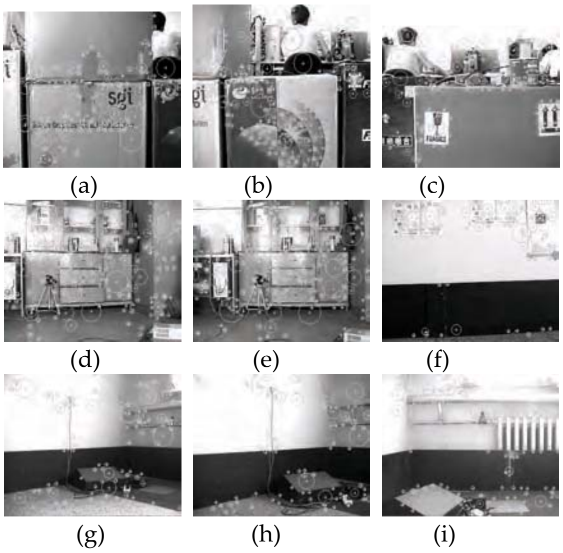


(a)



(b)

Fig. 8. (a) Pioneer 3 robot ;(b) experiment enviroment.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 9. Frames of an image sequence with SIFT features marked: (a) 2th frame; (b) 9th frame; (c) 19th frame; (d) 70th frame; (e) 79th frame; (f) 100th frame; (g) 150th frame; (h) 163th frame; (i) 172th frame.

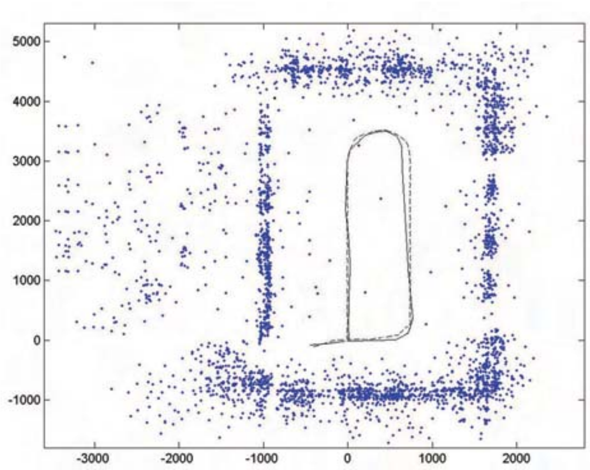
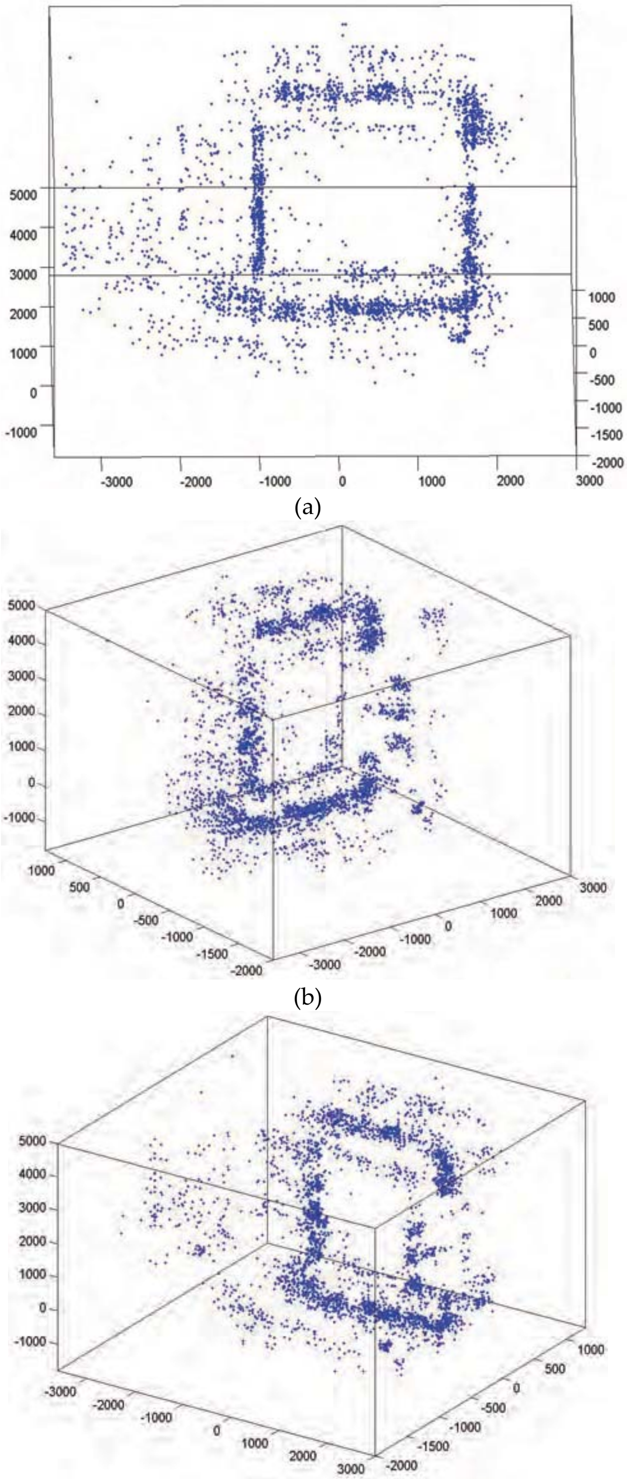


Fig. 10. Bird's-eye view of the SIFT landmarks in the map. the dashed line indicates the estimated robot path and the solid line indicates the real robot path.



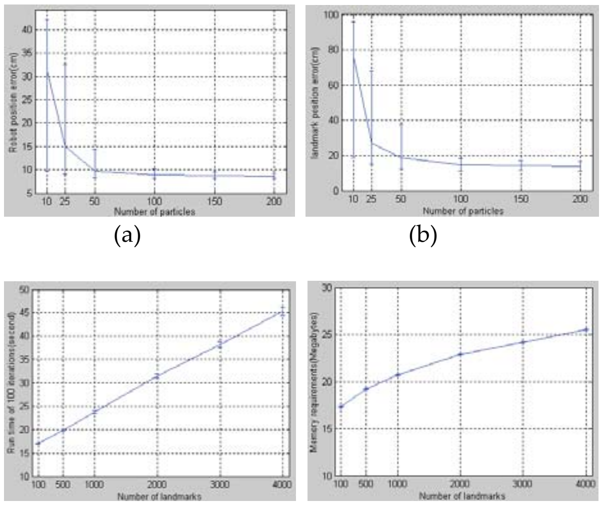
(a)

(b)

(c)

Fig. 11. The 3D SIFT landmark database map viewed from different angles. Each landmark has appeared consistently in every view: (a) from top; (b) from left; (c) from right.

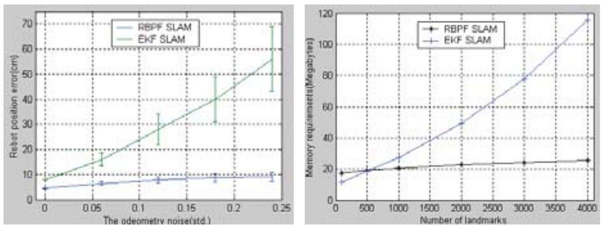
The images are captured and processed, the map is kept and updated on the fly while the robot is moving around. The robot goes around in the laboratory for one loop and to come back. Fig. 9 shows some frames of the 320× 240 image sequence (189 frames in total) captured while the robot is moving around. A total of 4068 SIFT landmarks with 3D positions are gathered in the map. The runtime of our RBPF SLAM algorithm with different numbers landmarks is shown in Fig. 10. Other performance of our SLAM algorithm with different numbers of particles and landmarks is also shown in Fig. 12. Fig. 10 shows the bird's-eye view of all these landmarks. Fig. 11 shows three views of the 3D SIFT landmark map from different angles. Finally, we compare our method with traditional EKF method, and our method shows superior performance as shown in Fig. 13.



(a) (b)

(c) (d)

Fig. 12. Performance of our RBPF SLAM algorithm: (a) robot position error and (b) landmark error with different numbers of particles; (c) runtime and (d) memory requirement with different numbers of landmarks.



(a) (b)

Fig. 13. Comparison of our RBPF SLAM algorithm and EKF for (a) robot position error and (b) memory requirement.

## 6. Conclusion

Novel RBPF is presented to implement monocular vision-based mobile robot SLAM in indoor environment. The particle filter is combined with UKF to sampling new poses integrating the current observation. The landmark position estimation and update is implemented through the UT and EKF respectively. For solving the particle depletion problem, the number of resampling steps is selected adaptively. Single camera tracks the 3D natural point landmarks, which are structured with matching feature pairs extracted through SIFT. The matching for highly distinctive SIFT features descripted with multi-dimension vector is implemented with a KD-Tree in the time cost of $O(\log_2^N)$. Experiment results show superior performance for our method.

## 7. References

Chong, K. S. & Kleeman, L. (1999). Feature-based mapping in real large scale environments using an ultrasonic array, International Journal of Robotics Research, Vol. 18, No. 1, pp. 3-19, ISSN 0278-3649

Choset, H. & Nagatani, K. (2001). Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization, IEEE Trans. Robot. Automat., Vol.17, No. 2, pp. 125-137, May 21-26, Seoul, Korea

Dellaert, F., Fox, D., Burgard, W. & Thrun, S. (1999). Monte Carlo localization for mobile robots, in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), Vol. 2, pp. 10-15, Detroit, MI, USA

Doucet, A. (1998). On sequential simulation-based methods for Bayesian filtering, Technical report, Signal Processing Group, Departement of Engeneering, University of Cambridge

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.,Vol. 93, pp. 1032-1044

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints, Int. J. of Computer Vision, Vol. 60, No. 2, pp. 91-110, ISSN 0920-5691

Ma S. D. & Zhang ZH. Y. (1998). Computer vision-computational theories and algorithm, Science Press, Beijing, pp.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, Engineering Department

Montemerlo, M. & Thrun, S. (2003). Simultaneous localization and mapping with unknown data association using FastSLAM, in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), Taipei, China

Moore, A.W. (1991). An introductory tutorial on kd-trees, Robotics Institute, Carnegie Mellon University, Pittsburgh, Technical Report No. 209, Computer Laboratory, University of Cambridge

Murphy, K. & Russell, S. (2001). Rao-blackwellized particle filtering for dynamic bayesian networks, in Sequential monte carlo methods in practice, Springer Verlag

Schultz, A. C. & Adams, W. (1998). Continuous localization using evidence grids, in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), pp.2833-2839, May 16-20, Leuven, Belgium