2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Monocular SLAM Using a Rao-Blackwellised Particle Filter with Exhaustive Pose Space Search

Masahiro Tomono

Abstract— **This paper presents a method of 3-D SLAM using a single camera. We utilize a Rao-Blackwellised particle filter (RBPF) to deal with a large number of landmarks. A difficulty in monocular SLAM is robustness to outliers and noise, which may cause false estimates especially under short baseline conditions. We propose an exhaustive pose-space search that finds all the plausible hypotheses efficiently using epipolar geometry. The obtained pose hypotheses are refined by the RBPF. Simulations and experiments show that the proposed method successfully performed 3-D SLAM with a small number of particles.**

Index Terms— **Object modeling, 3-D maps, 3-D reconstruction, Structure from motion, Dense reconstruction**

I. Introduction

3-D Simultaneous Localization and Mapping (SLAM) is a challenge in mobile robotics. 6-DOF localization in a 3-D map is crucial in order for a robot to navigate in a complex environment and to perform a complicated task such as object carrying. Vision-based SLAM is a promising approach to this problem. Especially, monocular SLAM is attractive because its hardware configuration is simple. Furthermore, monocular SLAM can reconstruct distant objects in large environments since its baseline distance is variable. We consider in this paper a system which utilizes a single camera only. Motion sensors such as gyro are not necessary but can be used to enhance accuracy and efficiency.

Monocular SLAM estimates camera motions and landmark locations in 3-D space using features extracted from images captured by a moving camera. Since a single image has no depth information, the system must reconstruct the depth of each feature from two or more images simultaneously with reconstructing the camera motion. This is a well-known problem referred to as Structure-from-Motion (SFM) in the computer vision community. This problem is especially crucial at the initialization phase, where the system has no 3-D reference points (landmarks) yet. In SFM, the stability of the system heavily depends on outliers and noise in the feature positions in the images. Even small noise will affect the estimates significantly when the camera motion is small. Thus, robustness against outliers and noise is crucial.

This paper presents a monocular SLAM scheme focusing on this problem. To increase robustness, the system searches all the camera motion hypotheses exhaustively. If the extracted features are noisy, many hypotheses can be generated. When the camera motion is small, it is difficult to determine which hypothesis is correct. Thus, we find all the plausible hypotheses. A key point is an efficient search by the reduction of the search space dimension from 5-D to 3-D using epipolar geometry. Another key point is that we employ a multiple hypothesis tracking scheme, in which the system tracks all the plausible hypotheses using the Rao-Blackwellised particle filter (RBPF) [13], [11]. The RBPF filters out false hypotheses and finds the correct one based on successive measurements. The RBPF is also suitable for vision-based SLAM since it can handle a large number of landmarks.

M. Tomono is with the Department of System Robotics, Toyo University, Kawagoe, SA, Japan tomono@eng.toyo.ac.jp

II. Related Work

A monocular SLAM system was firstly developed by Davison [1]. His system employs the Extended Kalman Filter (EKF) and particle filters for landmark initialization. Eade et al. developed a monocular SLAM system using an RBPF for scalable SLAM [3]. Elinas et al. proposed $\sigma - SLAM$ using binocular stereo and an RBPF with SIFT features to build indoor maps robustly [4]. These systems do not use motion sensors.

Monocular SLAM is regarded as a kind of bearing-only SLAM. In bearing-only SLAM, the landmark location is estimated using EKF with observations from two or more robot poses. When the distance between the robot poses is short, the gaussianity of the obtained estimation is too poor to employ EKF. Several approaches to this problem have been proposed including multiple hypothesis filter [9], federated information sharing [16], and inverse depth scheme [3], [12].

Monocular SLAM is also related with the Structure-from-Motion (SFM) that has been studied in the computer vision community. SFM reconstructs camera motions and object shapes simultaneously based on epipolar geometry with an optimization scheme [8]. A number of methods have been developed including the eight point method [7], the factorization method [18], the trifocal tensor [6], bundle adjustment, and so on. Nistér developed visual odometry based on the SFM scheme [14]. Most of these systems assume that feature correspondences are given by a feature tracker, and employ a robust estimation technique such as RANSAC [5] in order to eliminate outliers.

SFM has the same structure as bearing-only SLAM. A difference between them is that bearing-only SLAM is an estimation problem with a motion model, for which motion sensors such as odometry and gyro are used in many cases. On the other hand, most SFM systems have no motion models. Our method is based on the SLAM scheme with a motion model, which predicts the camera motion using monocular images, not motion sensors.

Some systems in SFM need no feature correspondences. Dellaert et al. proposed a SFM method without correspondence based on the Expectation-Maximization scheme [2]. Makadia et al. proposed a SFM method without correspondence using a Radon transform [10]. The latter is based on a kind of voting scheme, and our approach is conceptually similar. The difference is that our approach searches the camera pose space directly by reducing the dimension based on the fact that the camera translation is not independent of the camera rotation under epipolar geometry.

III. Basic Framework

A. SLAM using a Rao-Blackwellised Particle Filter

The SLAM considered here estimates the joint probability density $p(x_{1:t}, m|z_{1:t}, u_{1:t}, c_{1:t})$ of robot poses $x_{1:t}$ and map m [17]. Here, $z_{1:t}$ is the features observed from time step 1 to t, and $u_{1:t}$ is a sequence of motion commands. The map m is a set of landmarks m_i, and $c_{1:t}$ is correspondences between landmarks and observed features. In this paper, as other vision-based SLAM, a feature is a 2-D point extracted from a captured image, and a landmark is a 3-D point which corresponds to a feature. For simplicity, we equate robot pose with camera pose.

The RBPF-based SLAM factors $p(x_{1:t}, m|z_{1:t}, u_{1:t}, c_{1:t})$ as follows by exploiting the conditional independence between robot poses and landmark locations [13], [11].

$$\begin{aligned} &p(x_{1:t}, m|z_{1:t}, u_{1:t}, c_{1:t}) \\ &= p(x_{1:t}|z_{1:t}, u_{1:t}, c_{1:t}) \prod_{i}^{n} p(m_i|x_{1:t}, z_{1:t}, u_{1:t}, c_{1:t}) \end{aligned} \quad (1)$$

The joint distribution is decomposed into low-dimensional probabilities, which are much more tractable than the original one. The probability density of robot poses $p(x_{1:t}|z_{1:t}, u_{1:t}, c_{1:t})$ is represented using a particle filter. The probability density of a landmark location $p(m_i|x_{1:t}, z_{1:t}, u_{1:t}, c_{1:t})$ is represented with a Gaussian distribution which can be computed using an EKF.

In implementation, the i-th particle ν_t^i at time t is represented in the following fashion.

$$\nu_t^i =< x_t^i, (\mu_{1,t}^i, \Sigma_{1,t}^i), \ldots, (\mu_{N,t}^i, \Sigma_{N,t}^i) >$$

x_t^i is the robot pose estimate and $\mu_{j,t}^i, \Sigma_{j,t}^i$ are the j-th landmark location estimate and its covariance matrix.

$p(x_{1:t}|z_{1:t}, u_{1:t}, c_{1:t})$ is estimated using a particle filter based on a motion model and a measurement model. Intuitively, the probability density of $x_{1:t+1}$ is predicted based on the motion model and the probability density of $x_{1:t}$, and then the importance weight of each particle is calculated using the likelihood of the observed features based on the measurement model. By resampling particles according to the importance weights, the probability density of $x_{1:t+1}$ is obtained. The details of this procedure in our system is presented in Section V.

B. Our Approach to Monocular SLAM

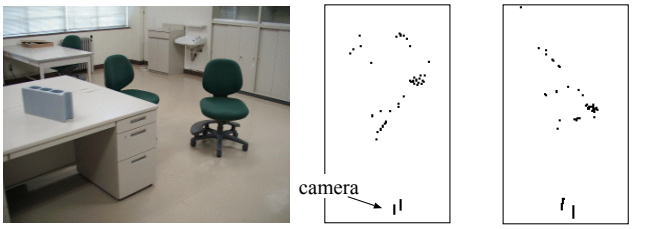
As mentioned in Section I, outliers in feature correspondences and/or noise in the feature positions cause false estimates especially when the robot motion is small. This is crucial at the initialization phase, where the system has no 3-D landmarks yet. The essential point here is that many subsets of features could generate a different hypothesis of the camera motion when there are outliers and/or noise. (Note that any subset of features would generate the same estimate without outliers nor noise.) The RANSAC is a useful scheme to find a good hypothesis, but unfortunately the hypothesis having the best score is not necessarily the correct estimate. Fig. 1 shows examples of hypotheses generated by SFM with RANSAC. (b) is the correct estimate and (c) is the false one, but the score of (b) is smaller than that of (c).

To cope with this problem, our system searches all the plausible hypotheses over the camera pose space, and filters out false hypotheses using an RBPF in order to find the correct one. In this process, we utilize the fact that the camera translation is determined linearly based on epipolar geometry when a camera rotation angle is given. This enables us to search the camera pose (rotation and translation) space exhaustively only by traversing the rotation space. Furthermore, this implies that the robot pose has virtually 3-DOFs for rotation only, and that the number of particles could be reduced.

More concretely, we discretize the camera rotation space, and find the most plausible camera translation for each discretized rotation angle. Given two point correspondences and a rotation angle, the camera translation is exactly determined up to scale based on epipolar geometry as mentioned later. To find the most plausible translation from a set of point correspondences, we employ a voting scheme. For each pair of feature correspondences, we calculate the camera translation and vote into the corresponding bin in the translation space. Then, the bin with the highest score is selected as the most plausible translation at the rotation angle. By repeating this process for all the discretized rotation angles, we have the score distribution over the rotation space. Now, we choose the rotation angles having a high score as good hypotheses. Outliers can be eliminated through the voting process.

Our approach can find all the feasible hypotheses over the pose space exhaustively. Since the RBPF can filter out false hypotheses efficiently, the key point is whether the obtained hypotheses include the true one or not. The RANSAC can generate feasible hypotheses, but it is not realistic to examine all the hypotheses over the pose space exhaustively since the RANSAC searches hypotheses over the correspondence space. The exhaustiveness over the pose space is the major advantage of our approach.

Another advantage is that our approach is quite suitable for a motion sensor such as gyro. Odometry is not applicable to the robots that move with 6-DOF in 3-D space. Although a gyro measures merely rotation angles, it will be sufficient for our scheme. The measurements from a gyro can narrow the search region in the rotation space significantly, and it will increase the accuracy and efficiency of our approach.



(a) Scene (b) Correct hypothesis (score = 40) (c) False hypothesis (score = 42)

Fig. 1. Examples of reconstruction hypotheses

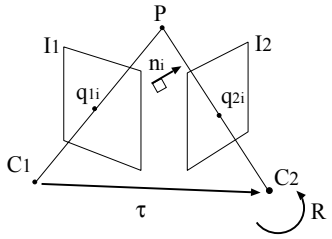


Fig. 2. Epipolar geometry

IV. Motion Estimation by Exhaustive Search

A. Scoring Function over Pose Space

Let I_1 and I_2 be images captured from a moving camera, and Q_1 and Q_2 be the feature sets extracted from I_1 and I_2 respectively. The problem considered here is to estimate the camera motion $r = \langle \psi, \tau \rangle$ from I_1 to I_2 given Q_1 and Q_2. Here, ψ is rotation angles (roll, pitch, yaw), and τ is a translation vector. Note that we assume the camera intrinsic parameters are known.

We propose a method that searches the camera pose space exhaustively. First, we define the scoring function $G(r)$ for camera motion r.

$$G(r) = \sum_{q_{1i} \in Q_1} \sum_{q_{2i} \in Q_2} g(q_{1i}, q_{2i}) D(r, q_{1i}, q_{2i}) \tag{2}$$

$g(q_{1i}, q_{2i})$ is the matching score of image feature points q_{1i} and q_{2i}. $D(r, q_{1i}, q_{2i})$ represents the score related with errors in the epipolar constraint, to be mentioned later. By calculating $G(r)$ for each r, we have a score distribution over the camera pose space. The camera poses having a high score in this distribution are regarded as a good hypothesis. However, it is not realistic to search directly all the point r in the camera pose space since the dimension of r is essentially five (the scale cannot be obtained from images only). Makadia et al. proposed a method of reducing computational complexity using spherical harmonic analysis [10]. We propose a method of calculating Eq.(2) more directly in the next subsection.

In the general framework, q_{1i} and q_{2i} cover all the points in the images, and no explicit correspondences between them are necessary. In this paper, however, for simple implementation, we assume the explicit one-to-one correspondences between Q_1 and Q_2 using a feature tracker such as the KLT tracker [15]. Thus, $g(q_{1i}, q_{2i})$ is defined as follows. This restriction will be removed in the near future.

$$g(q_{1i}, q_{2i}) = \begin{cases} 1, & q_{1i} \text{ and } q_{2i} \text{ are matched} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

B. Translation Estimation by Epipolar Geometry

Eq.(2) can be calculated efficiently by traversing the camera rotation space only. The basic idea is to calculate the camera translation from two point correspondences using epipolar geometry given a camera rotation angle.

Let q_{1i} and q_{2i} be a feature point in image I_1 and I_2 respectively as shown in Fig. 2. It is assumed that q_{1i} and q_{2i} are matched by a feature tracker. Then, the well-known epipolar constraint holds as follows.

$$(q_{1i} \times R q_{2i})^T \tau = 0 \tag{4}$$

Here, R and τ are the rotation matrix and the translation vector of r respectively. $q_{1i} \times R q_{2i}$ is the normal vector of the epipolar plane. We denote it by n_i.

If the rotation matrix R is constant, Eq.(4) will be a linear equation with respect to τ. Given two point correspondences, we can easily obtain τ by computing the cross product of the normal vectors n_i and n_j of the two epipolar planes which are determined by the point correspondences (q_{1i}, q_{2i}) and (q_{1j}, q_{2j}) $(i \neq j)$.

$$\tau = n_i \times n_j \tag{5}$$

We assume $||\tau|| = 1$ since the real scale cannot be obtained from images.

We calculate $D(r, q_{1i}, q_{2i})$ in Eq.(2) as follows.

$$\begin{aligned} D(r, q_{1i}, q_{2i}) &= \sum_{q_{1j} \in Q_1} \sum_{q_{2j} \in Q_2} g(q_{1j}, q_{2j}) \\ &\times D_0(r, q_{1i}, q_{2i}) D_0(r, q_{1j}, q_{2j}) \\ D_0(r, q_1, q_2) &= e^{-\alpha |(q_1 \times R q_2)^T \tau|^2} \end{aligned} \tag{6}$$

$D_0(r, q_1, q_2)$ represents the score related with errors in the epipolar constraint. α is a given constant.

C. Voting into Translation Space

We compute the scoring function $G(r)$ using a voting scheme.

(1) Discretization of the camera rotation space
We define a region which will cover all the possible rotation angles between I_1 and I_2, and discretize the region. We denote a discretized angle by ψ_n. This region is expected to be small in the case of monocular SLAM, which is a sequential process in usual.

(2) Discretization of the camera translation space
We create a voting table by discretizing the translation space. Since τ is a unit vector, τ is represented by two angles in a polar coordinate system.

(3) Estimation of translation for a rotation angle
Given a discretized rotation angle ψ_n, we calculate the translation vector τ using Eq.(5) for each pair of feature points in $Q_1 \times Q_2$. In this paper, we approximate D_0 in Eq.(6) simply as a delta function, and vote into the bin corresponding to τ in the translation voting table. Then, we find the bin τ_m having the maximal

score. Now, we define $G(\langle\psi_n, \tau_m\rangle)$ as the maximal score.

(4) Estimation of rotation angle
By repeating step (3) for all the discretized rotation angles, we have the score distribution over the rotation space. Note this is an approximation of $G(r)$.

(5) Selection of pose hypotheses
We employ as pose hypothesis each r at which $G(r)$ exceeds a given threshold th_1.

The hypotheses obtained at step (5) have insufficient accuracy because of the discretization of the pose space. Thus, we refine each hypothesis using a non-linear optimization method that minimizes the reprojection errors, which is a well-known technique in computer vision.

The computational complexity of this procedure is $O(KN^2)$ when we assume the feature correspondence is one-to-one as Eq. (3). K is the number of discretized angles in the rotation space, and N is the number of feature points.

D. Elimination of Outliers

The voting process eliminates outliers in the camera motion estimation. If feature correspondences include outliers, the votes calculated from the outliers will be distributed randomly over the translation space. Thus, outliers will not affect the score distribution as long as the outlier rate is not significantly large (see Section VI-A).

Once a camera pose r is obtained, we can eliminate outliers with respect to r using epipolar geometry. If q_1 and/or q_2 are outliers with respect to r, $(q_1 \times Rq_2)^T\tau$ will be large. Thus, we eliminate the features which make the value larger than a given threshold.

V. SLAM Formalization

A. Motion Model

The motion model $p(x_t|x_{t-1}, u_t)$ is the probability density that the robot moves from x_{t-1} to x_t given motion command u_t. Without motion sensors, we define the motion model using a Gaussian mixture which consists of camera pose hypotheses estimated by the abovementioned method. Each pose hypothesis is represented by $N(x_t^i, \Sigma_{x_t^i})$. x_t^i is calculated as $x_t^i = r^i + x_{t-1}$, where r^i is the i-th pose hypothesis obtained by the voting scheme. The covariance is calculated as $\Sigma_{x_t^i} = (J_{x_t^i}^T \Sigma_{z_t}^{-1} J_{x_t^i})^{-1}$, where $J_{x_t^i}$ is the Jacobian of perspective projection function $z_t = h(x_t, m_{c_t})$ with respect to the camera pose at x_t^i. Σ_{z_t} is the covariance of the feature noise.

If we have a motion sensor, we can reduce the number of possible hypotheses significantly. The motion model based on the velocity and acceleration estimated from the past camera trajectory is also useful to filter out the hypotheses. This is important from a practical point of view, but we do not discuss it in this paper.

B. Measurement Model

The measurement model $p(z_t|x_t, m_{c_t}, c_t)$ is the probability density that landmark m_{c_t} is projected onto feature point z_t when the camera pose is x_t. c_t represents the correspondence between m and z_t.

We approximate this probability density with a Gaussian distribution. Based on the perspective projection model, the j-th feature point $z_{j,t}$ is a function of the camera pose x_t and the corresponding landmark m_j, that is, $z_{j,t} = h(x_t, m_j)$. By linearizing this function using Taylor expansion with respect to m_j, we have the following equation.

$$z_{j,t} \quad = \quad \hat{z}_{j,t} + J_{m_{j,t-1}}(m_j - \bar{m}_{j,t-1}) + v_j$$

Here, $\hat{z}_{j,t} = h(\hat{x}_t, \bar{m}_{j,t-1})$. $\hat{x}_t$ is the prediction of x_t by the motion model. $J_{m_{j,t-1}}$ is the Jacobian of $h(x_t, m_j)$ with respect to $\bar{m}_{j,t-1}$. v_j is measurement noise in a 2-D feature point, which is represented by $N(0, R)$. Then, $z_{j,t}$ is represented as a Gaussian $N(\hat{z}_{j,t} + J_{m_{j,t-1}}(m_j - \bar{m}_{j,t-1}), R)$.

C. Importance Weight

We calculate the importance weight of each particle according to FastSLAM1.0 [17]. The proposal distribution is as follows.

$$p(x_{1:t}|z_{1:t-1}, u_{1:t}, c_{1:t-1}) = p(x_t|x_{t-1}, u_t)p(x_{1:t-1}|z_{1:t-1}, u_{1:t-1}, c_{1:t-1})$$

Importance weight w_t^i is calculated as follows.

$$\begin{aligned} w_t^i \quad &= \quad \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(x_{1:t}^i|z_{1:t}, u_{1:t}, c_{1:t})}{p(x_{1:t}^i|z_{1:t-1}, u_{1:t}, c_{1:t-1})} \\ &= \quad \eta \int p(z_t|m_t, x_t^i, c_t)p(m_t|x_{1:t-1}^i, z_{1:t-1}, c_{1:t-1})dm_t \end{aligned}$$

This is a convolution of $N(\hat{z}_{j,t}+J_{m_{j,t-1}}(m_j-\bar{m}_{j,t-1}), R)$ and $N(\bar{m}_{j,t-1}, \Sigma_{m_{j,t-1}})$. We have the importance weight as follows.

$$w_t^i \propto \prod_j N(\hat{z}_{j,t}^i,\ R + J_{m_{j,t-1}}^T \Sigma_{m_{j,t-1}} J_{m_{j,t-1}}) \qquad (7)$$

D. Landmark Update

The probability density of landmark location is updated as follows. In the RBPF-SLAM, this is calculated using EKF.

$$p(m_{c_t}|x_{1:t}, z_{1:t}, c_{1:t}) = \eta p(z_t|x_t, m_{c_t}, c_t)p(m_{c_t}|x_{1:t-1}, z_{1:t-1}, c_{1:t-1})$$

In this paper, however, we estimate landmark locations simply using the triangulation from feature points on two images. When the baseline distance is short, the errors in the location of a landmark reconstructed from images would be too large to represent by a Gaussian distribution because of the non-linearity of perspective projection. Thus, we estimate the landmark location using the triangulation at every frame, and select the most accurate estimation based on the covariance matrix of the estimated landmark location. This is the landmark initialization problem well-known in monocular SLAM, and we will improve the process by employing EKF with the inverse depth scheme [12] in the future.

The covariance of a landmark location is calculated as follows [8]. $\Sigma_{m_{j,t}}$ is computed using SVD.

$$\Sigma_{m_{j,t}} = (J_{m_{j,t}}^T \Sigma_{z_{j,t}}^{-1} J_{m_{j,t}})^{-1}$$

E. Procedure

Our method is performed in the following procedure.

(a) Initialization ($t = 1$ to k)

Since there are no landmarks at the initialization step, the system estimates the camera motion and landmarks simultaneously only from images without motion sensors. To ensure sufficient baseline distance, we use k images. Currently, k is given by human.

(1) Camera pose estimation
We compute the score distribution from images I_1 and I_k using the method in Section IV, and create particles for the hypotheses having a high score.
(2) Landmark initialization
For each particle, we eliminate outliers and reconstruct landmarks by the triangulation using I_1 and I_k.

(b) Sequential reconstruction ($t > k$)

(1) Camera pose prediction
We compute the score distribution using the method in Section IV, and select the hypotheses having a high score. For each hypothesis, we eliminate outliers and estimate the camera pose.
Then, we create new particles by pairing each hypothesis at time t and each particle at time $t-1$. The number of particles increases in this process.
(2) Importance weight and resampling
We calculate the importance weight of each particle based on Eq.(7), and resampling particles according to the normalized importance weights. The number of particles is reduced to the original one.
(3) Landmark update
For each resampled particle, we eliminate outliers and reconstruct landmarks using the triangulation. If the landmark is new, we just reconstruct it from the first two images in which the landmark appears. If the landmark is already registered, we update it when the covariance of the new reconstruction is smaller than the old one.

The real scale cannot be obtained only from images. The scale of the generated 3-D map is proportional to τ obtained at the initialization step. Note that we assume $||\tau|| = 1$ as mentioned above. At the sequential reconstruction step, we estimate the scale factor using the 3-D map built so far. This is performed by minimizing the reprojection errors of the landmarks in the 3-D map onto the images using a non-linear optimization method.

VI. Experiments

A. Simulation

We carried out a simulation to evaluate the performance of our method by comparison with a RANSAC-based method. Fig.3 shows the success rates of camera pose estimation by the two methods. In this simulation, 50 landmarks are randomly generated in 3-D space, and are projected onto two images at different camera poses. Varying feature noise level σ (Gaussian) and outlier rate, the relative camera pose is reconstructed from the two images. The 8-point method

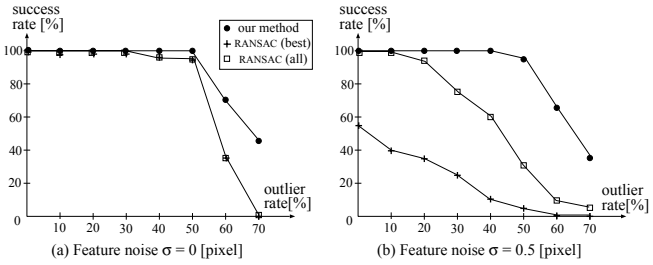


Fig. 3. Success rate of camera pose estimation

[7] is used for reconstruction in the RANSAC-based method. The number of samples in RANSAC is 1000.

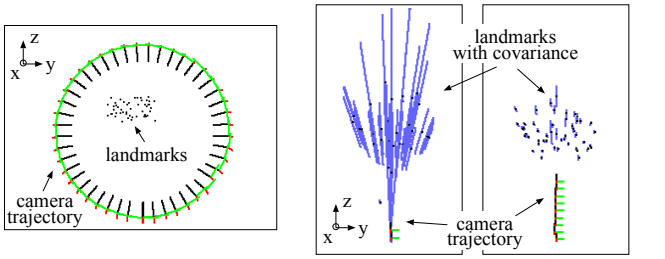
In this simulation, we judged a hypothesis is passed if its error in each camera rotation angle is within 1.0 [deg]. For our method, "success" means that at least one of the hypotheses selected at step (5) in Section IV-C is passed. The threshold th_1 was set to 70% of the maximal votes. For the RANSAC-based method, we employed two criteria. One is that it is successful when at least one of the 1000 samples is passed. The other is that it is successful when the sample having the best score is passed.

Theoretically, in the case of using the 8-point method, 1177 samples will provide 99% success rate at 50% outlier rate [8] when $\sigma = 0$. Fig.3 (a) supports it. Fig.3 (b) shows that the success rate of the RANSAC-based method is degraded more than that of our method when feature noise of $\sigma = 0.5$ [pixel] is added. From this result, we found that our method outperforms RANSAC in finding good hypotheses. We also found that the best hypothesis can be false. Multiple hypothesis tracking by RBPF addresses this problem.

Fig.4 shows the simulations of monocular SLAM by our method. 50 landmarks are randomly generated in 3-D space, and the camera moves along the predefined trajectories: a circle with a radius of 700 [cm] and a straight line of 1600 [cm]. Feature noise of $\sigma = 0.5$ [pixel] is added to each feature on the images, and outlier rate is 20 % in each image. The number of particles in RBPF is 20. Fig.4 shows the camera trajectory of the best of the 20 particles. In (a), the standard deviation of the camera poses in the best particle is $\sigma_x = 7.4$ [cm], $\sigma_y = 55.0$ [cm], $\sigma_z = 95.2$ [cm], $\sigma_{roll} = 0.36$ [deg], $\sigma_{pitch} = 0.40$ [deg], $\sigma_{yaw} = 0.27$ [deg]. In (b), the standard deviation is $\sigma_x = 12.0$ [cm], $\sigma_y = 8.2$ [cm], $\sigma_z = 21.9$ [cm], $\sigma_{roll} = 0.37$ [deg], $\sigma_{pitch} = 1.02$ [deg], $\sigma_{yaw} = 0.08$ [deg].

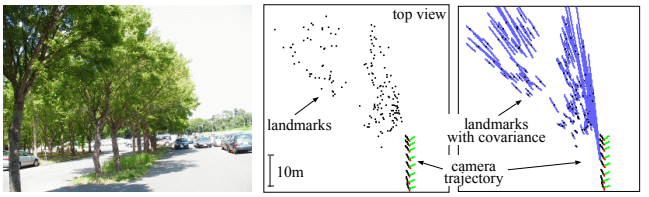
B. Experiments in Real Environments

We conducted experiments in indoor and outdoor environments. Images were captured by human with a digital camera. The image size was 320 by 240 pixels. The number of particles is 20. The correspondences between feature points were obtained using the KLT tracker [15]. The number of feature point in an image was 50. The experiments were done off-line. The maps were reconstructed using key frames, each of which was extracted over every n frames (n =3 to 8). n was given by human, which was constant in one experiment.



(a) Circular trajectory (b) Straight trajectory (left: step2, right: step9)

Fig. 4. Simulation results



(a) Snapshot of the environment (b) Result (c) Cov. of the landmarks

Fig. 5. Result of an experiment in outdoors

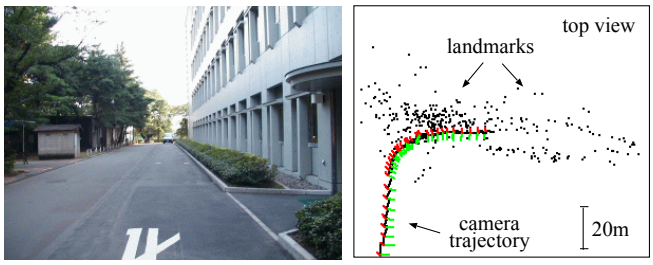


Fig. 6. Result of another experiment in outdoors

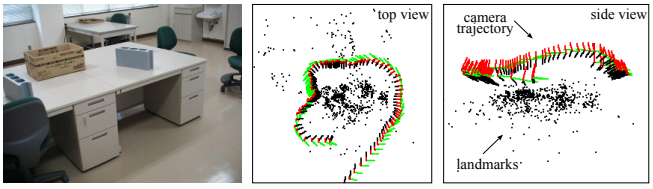


Fig. 7. Result of an experiment in indoors

Fig.5 shows the result of an experiment in outdoors. The camera moved about 15[m] and captured 30 images. The total number of landmarks is 173. Although outdoor environments have both near and far landmarks, they were reconstructed well as shown in (b). The covariance of each landmark is shown in (c). In this experiment, many hypotheses were generated at the initialization step. It is difficult to find which one is correct from a small number of measurements. The RBPF filtered out false hypotheses to find the correct one based on the measurements obtained time after time.

Fig.6 shows the result of another experiment in outdoors. The camera moved about 80[m] and captured 180 images. The total number of landmarks is 368. There are many landmarks at distant locations. Fig.7 shows the result of an experiment in indoors. The camera moved in a 10[m] $\times$ 10[m] room and captured 180 images. The total number of landmarks is 773. This is a good example of the 6-DOF camera motion in 3-D space.

In these experiments, the rotation space was discretized from -10 [deg] to 10 [deg] by 1 [deg] interval for each angle. The computation time is currently 3 to 10 seconds per key frame. The computation time will be reduced by program customization and parallel processing.

VII. Conclusions

The paper has presented a monocular SLAM scheme using a Rao-Blackwellised particle filter. Our contribution is an exhaustive pose space search, in which all the plausible hypotheses are found efficiently using epipolar geometry and a voting scheme. By tracking and refining multiple hypotheses using the RBPF, 3-D SLAM is performed robustly. Future work includes error analysis and more efficient implementation of the system.

References

[1] A. J. Davison: "Real-time simultaneous localization and mapping with a single camera," *Proc. of CVPR'03*, 2003.
[2] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun: "Structure from motion without correspondences," *Proc. of CVPR2000*, 2000.
[3] E. Eade and T. Drummond: "Scalable Monocular SLAM," *Proc. of CVPR'06*, 2006.
[4] P. Elinas, R. Sim, and J. J. Little: "σ *SLAM*: Stereo Vision SLAM Using the Rao-Blackwellised Particle Filter and a Novel Mixture Proposal Distribution," *Proc. of ICRA2006*, pp. 1564–1570, 2006.
[5] M. Fischler and R. Bolles: "Random Sample Consensus: a Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography", *Communications ACM*, 24:381-395, 1981.
[6] A. W. Fitzgibbon and A. Zisserman: "Automatic Camera Recovery for Closed or Open Image Sequences," *Proc. of ECCV'98*, 1998.
[7] R. Hartley: "In defense of the eight-point algorithm," *IEEE Trans. PAMI*, Vol. 19, No. 6, pp. 580–593, 1997.
[8] R. Hartley and A. Zisserman: "Multiple View Geometry in Computer Vision," Cambridge University Press, 2000.
[9] N. M. Kwok and G. Dissanayake: "An Efficient Multiple Hypothesis Filter for Bearing-Only SLAM," *Proc of IROS2004*, 2004.
[10] A. Makadia, C. Geyer, and K. Daniilidis: "Radon-based Structure from Motion Without Correspondences," *Proc of CVPR'05*, 2005.
[11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit: "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," *Proc of AAAI2002*, 2002.
[12] J. M. M. Montiel, J. Civera, and A. J. Davison:"Unified Inverse Depth Parametrization for Monocular SLAM," *Proc. of RSS2006*, 2006.
[13] K. Murphy and S. Russell: "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," in *A. Doucet ed. : Sequential Monte Carlo Methods in Practice*, Springer, 2001.
[14] D. Nistér, O. Naroditsky, and J. Bergen: "Visual Odometry," *Proc. of CVPR'04*, 2004.
[15] J. Shi and C. Tomasi: "Good Features to Track," *Proc. of CVPR'94*, pp. 593-600, 1994.
[16] J. Sola, A. Monin, M. Devy, and T. Lemaire: "Undelayed Initialization in Bearing Only SLAM," *Proc. of IROS2005*, pp. 2751-2756, 2005.
[17] S. Thrun, W. Burgard, and D. Fox: "Probabilistic Robotics," the MIT Press, 2005.
[18] C. Tomasi and T. Kanade: "Shape and Motion from Image Streams under Orthography: A Factorization Approach," *Int. J. of Computer Vision*, 9(2):137-154.