A Simple Framework for Indoor Monocular SLAM

Xuan-Dao Nguyen, Bum-Jae You, and Sang-Rok Oh

Abstract: Vision-based simultaneous localization and map building using a single camera, while compelling in theory, have not until recently been considered extensive in the practical realm of the real world. In this paper, we propose a simple framework for the monocular SLAM of an indoor mobile robot using natural line features. Our focus in this paper is on presenting a novel approach for modeling the landmark before integration in monocular SLAM. We also discuss data association improvement in a particle filter approach by using the feature management scheme. In addition, we take constraints between features in the environment into account for reducing estimated errors and thereby improve performance. Our experimental results demonstrate the feasibility of the proposed SLAM algorithm in real-time.

Keywords: Bearing only SLAM, localization, monocular.

1. INTRODUCTION

A map is a spatial model of the environment, which is essential for a mobile robot to perform its tasks. Therefore, the ability for a robot to autonomously build its own map while localizing itself (SLAM) has received much attention recently. Regarding uncertainty in positions and observed features as well as data association, a probabilistic framework is necessary for combining and optimizing those things over time.

Conventionally, most work on SLAM has focused predominantly on active range sensors such as lasers for data association. Laser sensors have high depth resolution providing accurate measurements of landmark positions but are expensive, heavy, high in power consumption, and suffer from the perceptual aliasing problem. On the other hand, cost effective cameras, which are easy to embed on robots, seem very attractive. Though suffering from computational costliness and difficulty in implementation, such cameras are becoming more and more popular in the SLAM community because good real-time vision tracking algorithms emerge as well as the increasing power in computer technology. Moreover, cameras with unique interest feature regions might lead to more reliable data association than is possible with a laser scanner.

Most vision based approaches use binocular or trinocular stereos that are similar to range finding devices [1-3] in the sense that they concentrate on solving the pose estimation problem from discrepancy image features. Stereo geometry requires the corresponding pattern to be visible in both cameras, therefore, cameras need to undergo synchronous calibration and be placed relatively close together. However, the accuracy of the pose estimate depends on the spacing between the cameras and the synchronous calibration since a small camera spacing results in a low depth resolution and asynchronous calibration degrades the accuracy. Conversely, single camera restrictions are relaxed allowing the camera placement to be more optimal with respect to occlusion and accuracy. In a reduced dimension, some researchers have pointed single cameras toward the floor or ceiling [4]; Folkesson *et al.* [5] similarly exploits the visual and geometric salience of the walls between those two extremes. The use of general monocular vision has only recently drawn some attention [6-10] due to several problems of a single camera. First, since at least two observations of the same landmark from two sufficiently spaced locations are necessary to initialize its position in the state vector, monocular SLAM aggravates the typical challenge of SLAM, namely the inter-dependence between mapping and localization. Second, the landmark observation consistencies between two positions depend on the stability of the tracking algorithm as well as the camera location uncertainty. Third, the high uncertainty landmark model computed from the aforementioned initialization may lead to the divergence of the SLAM algorithm as well as imprecise data association. However, these constraints can be manageable with proper initialization algorithms. Probably, the most important problem is the representation and maintenance of these initialized

Manuscript received February 23, 2007; revised September 4, 2007; accepted October 11, 2007; recommended by Editorial Board member Sooyong Lee under the direction of Editor Jae-Bok Song.

Xuan-Dao Nguyen, Bum-Jae You, and Sang-Rok Oh are with the Korea Institute of Science and Technology, 39-1 Hawolgok-dong, Deongbuk-gu, Seoul 136-791, Korea (e-mails: {nxdao, ybj, sroh}@kist.re.kr).

uncertainties for monocular SLAM. Recent approaches into two categories can be divided as follows: bottom-up and top-down approaches. They are natural inheritances of Structure from Motion (SFM) approaches of the vision community and Kalman Filter (KF) approaches of the control community. Bottom-up approaches use a batch nonlinear bundle adjustment or batch EKF procedure to initialize the landmarks from several past observations. Measurements to not-yet-initialized features are stored in the state vector, together with the corresponding robot positions. At a later stage, when the probability density of the landmark position has become sufficiently Gaussian, it is initialized in a batch update. Several bottom-up approaches use a large database of feature descriptors, into which features from novel views match up to localize the robot [11-13]. On the other hand, top-down approaches [14-16] acknowledge the fact that feature depth uncertainty during initialization is not well-modeled by a standard Gaussian distribution in Euclidean space. Hence, a new feature based on maintaining several depth hypotheses as Gaussian volumes for each initialized feature spread in a geometric sum was created. In 'delayed' initialization style, observations of features were pruned using observation evidences until the most likely single Gaussian remained which is then added to the filter to update the camera pose estimate. In un-delayed initialization, multi-hypotheses distributions are explicitly imposed on the state. However, the convergence of the filter when updating a multi-Gaussian feature and real time performance is not proved.

Along these SLAM real-time performance remains unproven in the literatures, the main contribution of our method is a general framework for monocular SLAM using line features. We combine the advantage of inverse depth concept [17] for good point modeling as well as Plucker constraint for line modeling. The approach inherits the advantage of location independence of the top-down approach. Moreover, our research deploys particle filter to take advantage of the well handling dimensionality and complexity. It has some commonalities with the researches in vision-based approaches held in the SLAM community recently [18,19]. These approaches however have generally been very different from ours because the applications are aimed at different initialization strategies and different incorporation filter techniques, and suffer some limitations from their assumptions. Meanwhile, we provide an alternative approach of the hybrid particle-unscented filter to accommodate a non-linear observation function. Our framework does have some good features from a practical viewpoint. It does not require a delay initialization. It uses the feature management process to improve the uncertainty. It can also impose indoor environment constraints to improve the map. The remaining organization of this paper is as follows. Section 2 of this paper introduces a top-down framework of the monocular-SLAM problem, emphasizing a modeling feature. Section 3 then briefs on our SLAM map representation, observation, and motion models in the context of hybrid unscented particle filter. Section 4 exhibits how the feature management improves the convergence of features through log likelihood and the constraints algorithm. Section 5 provides both simulation and real-world demonstration in an indoor environment to support the validity of our approach. Section 6 concludes and discusses some remaining problems.

2. GENERAL MONOCULAR-SLAM APPROACH

In accordance with popular SLAM literature, let s_t denote the robot's pose at time t. The map contains n features denoted $\Theta = \theta_1 \ldots \theta_n$. Also, let u_t denote robot's motion from time t-1 to time t with z_t as the current observation. The set of observations and motions from time 0 to t are denoted by z^t and u^t respectively. The data associations of particular observations to particular landmarks, up to time t are denoted n^t.

The vehicle is given a probabilistic motion model $p(s_t \mid u_t, s_{t-1})$ and a probabilistic observation model, denoted $p(z_t \mid s_t, \Theta, n_t)$, describing how measurements evolve from state. In general, both models are nonlinear functions with independent white noises P_t and R_t:

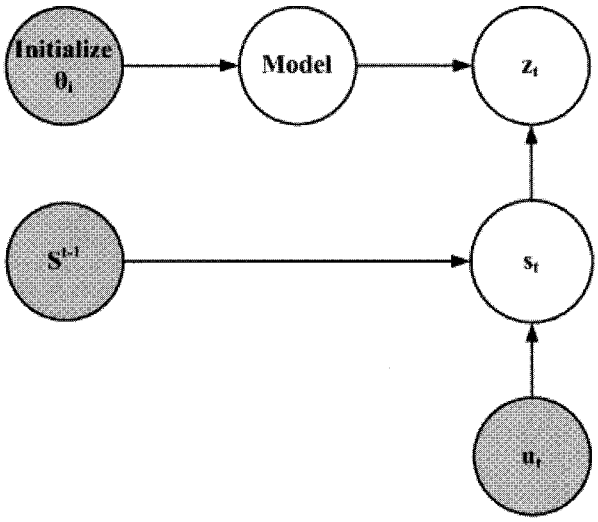


Fig. 1. A Bayes network showing common monocular-slam model. (Inputs are shaded nodes and outputs are unshaded nodes, association is not included for clarity.)

$$p(s_t \mid u_t, s_{t-1}) \approx h(u_t, s_{t-1}) + P_t, \quad (1)$$

$$p(z_t \mid s_t, \Theta, n_t) \approx g(s_t, \theta_{n_t}) + R_t. \quad (2)$$

SLAM algorithms, which recursively calculate distribution $p(s^t, \Theta | z^t, u^t, n^t)$, need to approximate both h and g as linear Gaussian distributions. Particle filter is a good approach for handling non-linearity of h. For g, however, it is more difficult as it is both non-linear and partial in monocular SLAM.

2.1. Initialization

In this paper, the landmark model is a half line considering the following reasons. Firstly, they consist of a point and line that combine strong characteristics of relatively easy-localize point correspondences with robust-illuminant line correspondences between frames. Furthermore, they are rich and easy to find in any indoor environment. Landmark features result from image segmentation into contours, which correspond to physical elements within the indoor environment, such as edges constituted from intersections between flat surfaces. Therefore, the proposed landmark model is easy to extract from environments and their characterization, by means of polygonal approximation or edge detection, reliable even in the presence of noise. Secondly, it is concise and possible to represent uncertainty in order to predict what is expected to be seen from a predicted camera pose. Moreover, it is relatively easy to track them in approximately linear motion since their shape does not change abruptly. In addition, partial occlusion due to either the view angle or the presence of non-modeled objects, does not affect line representation parameters. Finally, we proposed a modeling method based on these features correspondences.

First, Canny edge detection with adaptive threshold value are used to detect some of the most prominent half-lines. We generate a set of N samples point equally distributed on each detected feature. The Lucas-Kanade (KL) algorithm, as introduced in [20], is adopted to track these sample points. Let $W(x;p)$ denote the parameterized set of an allowed warping. Consider the set of affined warping described by the following equation.

$$W(x;p) = \begin{pmatrix} (1+p_1) & p_3 & p_5 \\ p_2 & (1+p_4) & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (3)$$

where six parameters in $p=(p_1, p_2, p_3, p_4, p_5, p_6)^T$ are used to represent an affined warping. We minimize the sum of squared errors between the template image T and the image I warped back onto the coordinate frame of the template by iteratively minimizing it with respect to Δp:

$$\sum_x [I(W(x, p+\Delta x)) - T(x)]^2. \quad (4)$$

The parameters undergo update by Δp iteratively until p converges. We use this gradient descent scheme for real-time tracking of each sample point on a detected line. The sampled template images, which change abruptly, experience removal as an outlier, with the rest fitting to the tracked line. KL tracking is typically applied between consecutive image frames due to the fact that linear approximation depends on a small motion assumption. Hence, we will *initialize new feature* during smooth motion for efficiency. When features are *fully initialized*, each feature is matched in subsequent frames using a more robust algorithm, namely, normalized sum-of-squared difference correlation. That is applying the 3D model-based line tracking algorithm to take advantage of the robustness to temporarily occlude the erratic motion while normalizing to deal with illumination changes. Observations are made by actively searching new frames through the current estimates of camera pose and landmark locations, and the uncertainty in these estimates are described in more detail in the observation section. Note that a landmark model is a half-line with one distinct end-point. If two neighboring landmark models happen to be co-aligned, these distinct endpoints will either separate the landmark models or merge them into one segment depending on the specific application. In our case, we separate them for simple computation.

2.2. Model

Probably, the most important problem is the representation and maintenance of uncertainty for monocular SLAM. Several approaches to solve the landmark initialization problem are based on approximating the non-Gaussian landmark state with a sum of Gaussians (GSF) (or particles, which are degenerated Gaussians). Then, subsequent observations update this PDF, making it converge to a single Gaussian. Inspired by the Federated Filter (FF), [14] proposes the FIS initialization method, which could be seen as a shortcut of the more proper GSF-SLAM. A more efficient solution is given by considering inverse depth instead, for which a Gaussian distribution is acceptable since inverse depth varies linearly with disparity [6,9], and as a result the EKF can still be used. This paper, however, propose a different approach using Scale Unscented Filter (SUF) to accommodate a highly non-linear observation function of a line model. We first describe the landmark model representation in 3D space in the context of SLAM. We then present the initialization steps of SUF for a vectorised line feature model in the normal case as well as in the ill-conditioned case.

We use Exponential Map representation of rotation [21] for the following reasons. First, the exponential

map requires only three parameters to describe a rotation (minimal representation). Second, it does not suffer from the gimbal lock problem and its singularities occur in a region of the parameter space that are easy to avoid. Let $\vec{w}=[w_x\ w_y\ w_z]^T$ be a 3D vector and $\theta=\|\vec{w}\|$ be its norm. An angular rotation θ around an axis of direction $\vec{w}$ can be represented:

$$R(\Omega)=I+\sin\theta\hat{\Omega}+(1-\cos\theta)\hat{\Omega}^2, \tag{5}$$

where $\hat{\Omega}$ is the skew-symmetric matrix corresponding to the unit vector $\frac{\vec{w}}{\|\vec{w}\|}$:

$$\hat{\Omega}=\begin{bmatrix} 0 & \frac{-w_z}{\theta} & \frac{w_y}{\theta} \\ \frac{w_z}{\theta} & 0 & \frac{-w_x}{\theta} \\ \frac{-w_y}{\theta} & \frac{w_x}{\theta} & 0 \end{bmatrix}. \tag{6}$$

It could be seen that $R(\Omega)=\exp(\Omega)=I+\Omega+\frac{1}{2!}\Omega^2+..$, where Ω is the skew-symmetric matrix of $\vec{w}$. $R(\Omega)$ is not singular when θ goes to zero as replacing $\frac{\sin\theta}{\theta}$ and $\frac{1-\cos\theta}{\theta^2}$ by the first two terms of their Taylor expansions.

The camera state x_v is composed of location: C^G camera optical center, $R(\Omega^G)$ rotation; linear and angular velocity $V^G=(T^G,A^G)$

$$x_v=\begin{pmatrix} C^G \\ \Omega^G \\ V^G \end{pmatrix}. \tag{7}$$

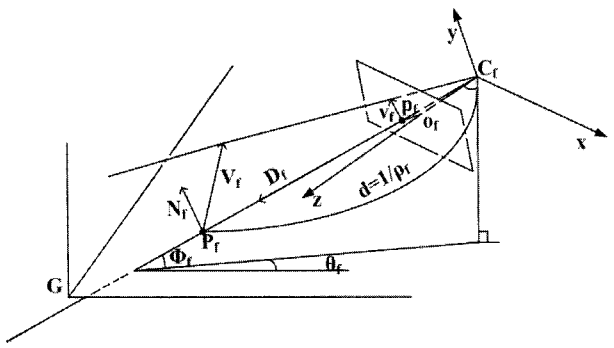


Fig. 2. Feature parametrization.

A half line feature is defined by the dimension 9 state vector (see Fig. 2): $x_f=\left(C_f^G\ \ D_f^G\ \ \rho_f\ \ V_f^G\right)^T$, which models a half- line feature $\left(\vec{P}_f^G,\vec{V}_f^G\right)$, where $\vec{P}_f^G$ is a 3D point located at $C_f^G+\frac{1}{\rho_f}D_f^G$, C_f^G is camera position, D_f^G is ray directional vector coded in the global reference as (θ_f,ϕ_f) azimuth and elevation, depth along the ray is coded by its inverse depth ρ_f, and $\vec{V}_f^G$ is direction vector of line feature x_f.

Camera calibration was computed beforehand to acquire camera intrinsic parameters. For a known camera calibration matrix K, camera focal length f_c we have: $\begin{pmatrix} \vec{p}_f^C \\ \vec{v}_f^C \end{pmatrix}=K\left(R^{-1}(\Omega^G)\begin{pmatrix}\vec{P}_f \\ \vec{V}_f\end{pmatrix}-\begin{pmatrix}C_f^G \\ C_f^G\end{pmatrix}\right)$,

where $R^{-1}(\Omega^G)$ is the inverse rotation matrix, $\vec{p}_f^C=\begin{pmatrix}p_x\\p_y\\f_c\end{pmatrix}$ is image feature point, $\vec{v}_f^C=\begin{pmatrix}v_x\\v_y\\0\end{pmatrix}$ is image feature vector in camera coordinate, while $\vec{P}_f^G$ and $\vec{V}_f^G$ are feature point and feature vector of x_f in global coordinate respectively. The state codes as: $\begin{bmatrix}x_v\\x_{f_1}\\x_f\end{bmatrix}$, where x_{f_1} is the first observation of x_f and considered as constant along the estimation. For equation simplicity, we could assume that each feature has already been computed and compensated for distortion using calibration matrix K.

Hereafter, a feature means an undistorted one. The

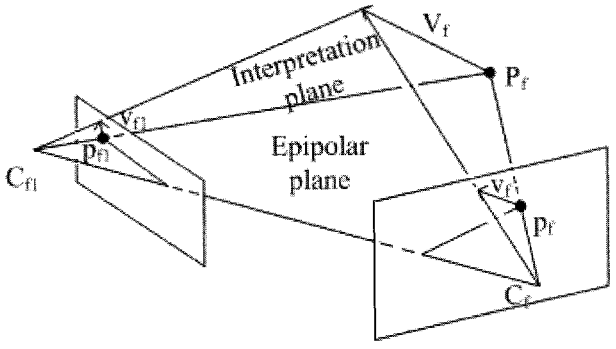


Fig. 3. Triangulation at 2 positions.

feature is then calculated by triangulation:

$$x_f = g\left(\left(C_{f1}^G \quad D_{f1}^G \quad \vec{p}_{f1}^C \quad \vec{v}_{f1}^C\right), \left(C_f^G \quad D_f^G \quad \vec{p}_f^C \quad \vec{v}_f^C\right)\right), \tag{8}$$

We have:

$$\begin{cases} \vec{n}_{f1}^C = \vec{v}_{f1}^C \times \vec{p}_{f1}^C \\ \vec{n}_f^C = \vec{v}_f^C \times \vec{p}_f^C \\ \vec{P}_f^C = k_1 \vec{p}_{f1}^C \\ \vec{P}_f^C = k \vec{p}_f^C, \end{cases} \tag{9}$$

where $\vec{n}_f^C$ and $\vec{n}_{f1}^C$ are normal vector of interpretation planes that go through camera positions and features half-line x_f expressed in camera coordinates, k and k_1 are unknown scale factors. Hence:

$$\vec{V}_f^G = (R(\Omega_{f1}^G)(\vec{v}_{f1}^C \times \vec{p}_{f1}^C)) \times (R(\Omega_f^G)(\vec{v}_f^C \times \vec{p}_f^C)), \tag{10}$$

$$\begin{aligned} \vec{P}_f^G &= (R(\Omega_{f1}^G)k_1 \vec{p}_{f1}^C + C_{f1}^G) \\ &= (R(\Omega_f^G)k \vec{p}_f^C + C_f^G), \end{aligned} \tag{11}$$

or $A_i k_1 + B_i k = C_i$ for $i = 1,2$. (see Appendix A) (12)

If $\det\begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \end{bmatrix} \neq 0$ then $k_1 = \dfrac{\det\begin{bmatrix} C_1 & B_1 \\ C_2 & B_2 \end{bmatrix}}{\det\begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \end{bmatrix}}$ and

$$\vec{P}_f^G = (R(\Omega_{f1}^G)k_1 \vec{p}_{f1}^C + C_{f1}^G). \tag{13}$$

Function $x_f = g(x)$, where $x = \left(C_f^G \quad D_f^G \quad \vec{p}_f^C \quad \vec{v}_f^C\right)$ can be approximated to Gaussian PDF by using SUF [22].

Assume that x has mean $\bar{x}$ and covariance P_x, a set of $2n_x + 1$ (n_x is the dimension of x) weighted samples $S_i = \{W_i, \chi_i\}$ are chosen as follows:

$$\lambda = \alpha^2(n_x + \kappa) - n_x, \tag{14}$$

$$\chi_0 = \bar{x}, \tag{15}$$

$$\chi_i = \bar{x} + \sqrt{(n_x + \lambda)P_x} \quad i = 1,...,n_x, \tag{16}$$

$$\chi_i = \bar{x} - \sqrt{(n_x + \lambda)P_x} \quad i = n_x + 1,...,2n_x, \tag{17}$$

$$W_{0(m)} = \frac{\lambda}{n_x + \lambda}, \tag{18}$$

$$W_{0(c)} = \frac{\lambda}{n_x + \lambda} + (1 - \alpha^2 + \beta), \tag{19}$$

$$W_{i(m)} = W_{i(c)} = \frac{1}{2(n_x + \lambda)} \quad i = 1,...,2n_x, \tag{20}$$

where κ is a scaling parameter, α is a positive scaling parameter that could be made as small as possible to minimize higher order effects, and β is a parameter that minimizes the effects from high order terms. Then each sample propagates through the nonlinear function g

$$x_{f_i} = g(\chi_i) \quad i = 0,1,...,2n_x. \tag{21}$$

The estimated mean and covariance of x_f are:

$$\bar{x}_f = \sum_{i=0}^{2n_x} W_{i(m)} x_{f_i}, \tag{22}$$

$$P_{x_f} = \sum_{i=0}^{2n_x} W_{i(c)}(x_{f_i} - \bar{x}_f)(x_{f_i} - \bar{x}_f)^T. \tag{23}$$

Note that n_x=9 in our case. Only 19 samples were generated and the distribution of y is accurate to the second order of a Taylor series expansion.

However, in case of ill-conditioned $\det\begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \end{bmatrix} = 0$ uch as zero base-line or features at infinitives, the initial location for the observed feature is defined as:

$$\lim_{\substack{C_f^G \to C_{f1}^G \\ \Omega_f^G \to \Omega_{f1}^G}} g\left(C_f^G \quad D_f^G \quad \vec{p}_f^C \quad \vec{v}_f^C\right). \tag{24}$$

We express as follows:

$$\begin{aligned} &\hat{x}_f = \left(\hat{C}_f^G \quad \hat{D}_f^G \quad \hat{\rho}_f \quad \hat{V}_f^G\right)^T, \\ &\hat{C}_f^G = C_{f1}^G. \end{aligned} \tag{25}$$

Let $\hat{D}_f^G = D_{f1}^G = \begin{pmatrix} D_x \\ D_y \\ D_z \end{pmatrix}$ then

$$\begin{pmatrix} \hat{\theta}_i \\ \hat{\phi}_i \end{pmatrix} = \begin{pmatrix} \tan^{-1}(-D_y, \sqrt{D_x^2 + D_z^2}) \\ \tan^{-1}(D_x, D_z) \end{pmatrix}. \tag{26}$$

The initial value for $\hat{\rho}_f$ is derived heuristically to cover in its 95% acceptance region a working space from infinity to a predefined close distance, d_{min} expressed as inverse depth: $\left[\frac{1}{d_{\min}}, 0\right]$, so:

$$\hat{\rho}_f = \frac{\rho_{\min}}{2} \quad \sigma_\rho = \frac{\rho_{\min}}{4} \quad \rho_{\min} = \frac{1}{d_{\min}}. \tag{27}$$

Similarly, $\hat{V}_f = \hat{\alpha} R(\Omega_{f_1}^G)(\vec{v}_{f_1}^C \times \vec{p}_{f_1}^C)$ where

$$\alpha \in [0,\pi], \hat{\alpha} = \frac{\pi}{2}, \sigma_\alpha = \frac{\pi}{4}. \quad (28)$$

3. PARTICLE FILTER SLAM

In this section, we provide descriptions of applying a hybrid particle-unscented filter to a general standard SLAM framework. Particle Filter (PF) has some advantages over KF, such as the abilities to deal with highly nonlinear sensor and robot motion as well as non-Gaussian noise. PF derives from Sequential Monte Carlo (SMC) and Bayes rule. It implements likelihood distribution, and uses particles instead of a determined point for state estimation. Moreover, the observation distribution used SUF to generate particles, resulting in a hybrid UPF to take advantage of well handling dimensionality and non-linearity observation. It is based on the fact that the posterior

$$\begin{aligned} &p(s^t, \Theta \mid z^t, u^t, n^t) \\ &= p(s^t \mid z^t, u^t, n^t) \prod_{m=1}^{N} p(\theta_m \mid s^t, z^t, u^t, n^t) \end{aligned} \quad (29)$$

can be factored into n landmark state-estimators and one vehicle trajectory estimator. Each particle p_m, with associated likelihood weight w_m corresponds to a complete pose and map hypothesis consisting of a robot pose $s_t^{[m]}$ and estimates of all landmarks $\{\theta_i\}$. Each particle has attached its own map and an independent Kalman filter is implemented for each landmark in the map $S_t^{[m]} = \langle s^{t,[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, ..., \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \rangle$, where $s^{t,[m]}$ is the m^{th} particle path estimate, and $\mu_{n,t}^{[m]}$ and $\Sigma_{n,t}^{[m]}$ are the mean and covariance of the Gaussian representing the n^{th} feature location conditioned on the path $s^{t,[m]}$.

Before addition to the filter, features are passed through the initialization and modeling process as mentioned above. At each time step t, the following steps take place.

3.1. Prediction

The prediction stage turns the sampled representation of pose into a Gaussian mixture representation of pose. We consider measurement z_t. Before sampling, we update particles $s^{t,[m]}, m = 1..M$ ith UKF as follows.

1) Calculate the sigma points as in modeling:

$$X_{t-1}^a = \left[\bar{x}_{t-1}^a, \bar{x}_{t-1}^a \pm \sqrt{(n_x + \lambda) P_{t-1}^a} \right]. \quad (30)$$

2) Time update:

$$X_{t|t-1}^x = h\left(X_{t-1}^x, X_{t-1}^V\right), \ \bar{x}_{t|t-1}^x = \sum_{i=0}^{2n_x} W_{i(m)} X_{i,t|t-1}^x, \quad (31)$$

$$Y_{t|t-1} = g\left(X_{t|t-1}^x, X_{t-1}^n\right), \ \bar{y}_{t|t-1} = \sum_{i=0}^{2n_x} W_{i(m)} Y_{i,t|t-1}^x, \quad (32)$$

$$P_{t|t-1} = \sum_{i=0}^{2n_x} W_{i(c)} \left[X_{i,t|t-1}^x - \bar{x}_{t|t-1} \right] \left[X_{i,t|t-1}^x - \bar{x}_{t|t-1} \right]^T. \quad (33)$$

3) Measurement update:

$$P_{y_t y_t} = \sum_{i=0}^{2n_x} W_{i(c)} \left[Y_{i,t|t-1}^x - \bar{y}_{t|t-1} \right] \left[Y_{i,t|t-1}^x - \bar{y}_{t|t-1} \right]^T, \quad (34)$$

$$P_{x_t y_t} = \sum_{i=0}^{2n_x} W_{i(c)} \left[X_{i,t|t-1}^x - \bar{x}_{t|t-1} \right] \left[Y_{i,t|t-1}^x - \bar{y}_{t|t-1} \right]^T, \quad (35)$$

$$K_t = P_{x_t y_t} P_{y_t y_t}^{-1}, \quad (36)$$

$$\bar{x}_t = \bar{x}_{t|t-1} + K_t \left(y_t - \bar{y}_{t|t-1} \right), \quad (37)$$

$$P_t = P_{t|t-1} - K_t P_{y_t y_t} K_t^T. \quad (38)$$

We obtain $\bar{s}_t^{[m]}$, $P_t^{[m]}$ and then we sample:

$$s_t^{[m]} \sim p(s_t \mid s^{t-1,[m]}, u^t, z^t, n^t) = N\left(\bar{s}_t^{[m]}, P_t^{[m]}\right). \quad (39)$$

3.2. Observation

Using this predicted pose distribution, and the associated landmark estimates for each particle, landmark observations are extracted from the new frames. For each landmark to be observed, the Gaussian estimate of the landmark under each particle is projected into the image by taking the weighted mean and covariance. This yields a single Gaussian estimate of landmark location in the image. The corresponding 3σ ellipse in the image is then searched to locate the landmark. The landmark's patch is warped by an affine homography A computed from the mode camera pose estimate and the initial camera pose from which the landmark's patch was captured. The location inside the ellipse yielding maximal normalized cross correlation (NCC) with the warped patch is taken as an observation of the landmark if the NCC score is above a certain threshold. If no such match is found, the landmark measurement is considered a failure.

3.3. Update stage

The update stage then computes the posterior distribution by incorporating these observations. Update observed landmark θ_{n_t} by normalized product of state distribution $p(s_t \mid s_{t-1}^{[m]}, u_t)$, and the

probability of measurement z_t.

$$p(s_t \mid s^{t-1,[m]}, u^t, z^t, n^t) = \eta^{[m]} \int \underbrace{p(z_t \mid \theta_{n_t}, s_t, n_t)}_{\sim N(z_t; g(\theta_{n_t}, s_t), R_t)} \underbrace{p(\theta_{n_t} \mid s^{t-1,[m]}, z^{t-1}, n^{t-1})}_{\sim N(\theta_{n_t}; \mu^{[m]}_{n_t,t-1}, \Sigma^{[m]}_{n_t,t-1})} d\theta_{n_t} \underbrace{p(s_t \mid, s^{[m]}_{t-1}, u_t)}_{\sim N(s_t; h(s^{[m]}_{t-1}, u_t), P_t)} \tag{40}$$

Here η is a constant, $\hat{z}^{[m]}_t = g(\hat{\theta}^{[m]}_{n_t}, \hat{s}^{[m]}_t)$ enotes the predicted measurement, $\hat{s}^{[m]}_t = h(s^{[m]}_{t-1}, u_t)$ he predicted robot pose and $\hat{\theta}^{[m]}_n = \mu^{[m]}_{n,t-1}$ the predicted landmark location.

3.4. Updating the observed landmark estimate.

The probability for the m^{th} particle to be sampled (with replacement) is given by the following variable:

$$w^{[m]}_t \propto p(z_t \mid s^{t-1,[m]}, u^t, z^{t-1}, n^t) = \left(\int\int \underbrace{p(z_t \mid \theta_{n_t}, s_t, n_t)}_{\sim N(z_t; g(\theta_{n_t}, s_t), R_t)} \underbrace{p(\theta_{n_t} \mid s^{t-1,[m]}, u^{t-1}, z^{t-1}, n^{t-1})}_{\sim N(\theta_{n_t}; \mu^{[m]}_{n_t,t-1}, \Sigma^{[m]}_{n_t,t-1})} d\theta_{n_t} \underbrace{p(s_t \mid, s^{[m]}_{t-1}, u_t)}_{\sim N(s_t; \hat{s}^{[m]}_{t-1}, P_t)} ds_t. \right. \tag{41}$$

This expression can once again be approximated as a Gaussian with mean $\hat{z}_t$ and covariance $Q^{[m]}_t$.

4. DATA ASSOCIATIONS AND FEATURE MANAGEMENT

In this section, we show a feature management scheme for data association improvement in particle filter approach. In addition, constraints between features in the environment are taken into account for reducing estimated errors and improving performance. We select the data association n_t that maximizes the probability of the measurement z_t for the m^{th} particle:

$$\hat{n}^{[m]}_t = \arg\max_{n_t} p(z_t \mid n_t, \hat{n}^{t-1,[m]}, s^{t,[m]}, z^{t-1}, u^t). \tag{42}$$

Data association is calculated as follows.

$$p(z_t \mid n_t, \hat{n}^{t-1,[m]}, s^{t,[m]}, z^{t-1}, u^t) = \int \underbrace{p(z_t \mid \theta_{n_t}, n_t, s^m_t)}_{\sim N(z_t; g(\theta_{n_t}, s^{[m]}_t), R_t)} \underbrace{p(\theta_{n_t} \mid \hat{n}^{t-1,[m]}, s^{t-1,[m]}, z^{t-1})}_{\sim N(\theta_{n_t}; \mu^{[m]}_{n_t,t-1}, \Sigma^{[m]}_{n_t,t-1})} d\theta_{n_t} \tag{43}$$

g is a Gaussian over z_t with mean $g(\mu^{[m]}_{n_t,t-1}, s^{[m]}_t)$ and covariance $Q^{[m]}_t$. Each particle makes its own local data association, which renders significantly more robust to noise than EKF-style algorithms.

Following [23], our approach removes such unlikely landmarks by keeping track of their posterior probability of existence. Our mechanism analyzes measurements for the presence and absence of features. Observing a landmark provides positive evidence for its existence, whereas not observing it when $\mu^{[m]}_n$ alls within the robot's perceptual range provides negative evidence. The posterior probability of landmark existence is accumulated by the following Bayes filter, whose log-odds form is:

$$\tau^{[m]}_n = \sum_t \ln \frac{p(i^{[m]}_n \mid s^{[m]}_t, z_t, \hat{n}^{[m]}_t)}{1 - p(i^{[m]}_n \mid s^{[m]}_t, z_t, \hat{n}^{[m]}_t)}. \tag{44}$$

Here $\tau^{[m]}_n$ re the log-odds of the physical existence of landmark $\theta^{[m]}_n$ in map and $p(i^{[m]}_n \mid s^{[m]}_t, z_t, \hat{n}^{[m]}_t)$ is the probabilistic evidence provided by a measurement. Under appropriate definition of the latter, this rule provides for a simple evidence-counting rule. If the log-odds drop below a predefined threshold, the corresponding landmark is removed from the map. This mechanism enables particles to free themselves of unlikely landmarks. Because the landmark estimates within each particle are independent, each landmark update can be computed in constant time. Thus, at the end of each time step, the particle cloud is a set of samples drawn from the posterior distribution of poses and landmarks given by all observations up to the current time. The total cost of updating landmark estimates and optimizing the proposal over M particles given k observations is O(Mk), independent of the number of landmarks N. In contrast, the EKF with full covariance requires O(N)^2 time to formulate observation updates, which makes large numbers of landmarks impracticable.

Most slam algorithms make few assumptions about the environment; thus, slam does not take advantage of prior information when the environment is known to have specific structural characteristics. In this paper, we take into account the fact that indoors environments can often be assumed to be "mostly" rectilinear. A few previous works have enforced constraints on maps represented using an extended Kalman filter (EKF) [24,25]. In this paper, we exploit the Rao-Blackwellized constraint filter. The major difficulty is that RBPF SLAM relies on the conditional independence of landmark estimates given a robot's pose history, but relative constraints introduce correlation between landmarks. Our approach exploits a property similar to that used in [26] for standard Rao-Blackwellization slam: conditioned on values of constrained state variables, unconstrained state variables are independent. We use this fact to incorporate per-particle constraint enforcement into the filter. The algorithm is brief as follows:

Algorithm 1: Constraints enforcement initialization and update

a) Initialize algorithm

1. Initialize backup state: $\beta_{n+1} \leftarrow \theta_{n+1}; \Lambda_{n+1} = P_{n+1}$
2. Initialize measurement accumulator: $z_{n+1} \leftarrow [0]; Q_{n+1} \leftarrow [\infty]$
3. Initialize constraint set: $R \leftarrow \{\ \}$
4. **For** all previously constrained groups $L_i \in \Theta$ **do**

   Draw constraint parameters $c_{n+1,j} \sim p(c_{n+1,j}), \forall \theta_j \in L_i$

   If constrained: $\exists \theta_j \in L_i$ such that $c_{n+1,j} \neq *$

   For all $\theta_j \in L_i$ do

   Add x_j to constraint set: $R \leftarrow R \cup \{\theta_j\}$

   End for

   End if

   Remove old super landmark: $\Theta \leftarrow \Theta \setminus L_i$

   End for
5. **If** no constraints on x_{n+1} return: $R = \varnothing$ **end if**
6. Add new landmark to constraint set: $R \leftarrow R \cup \{\theta_{n+1}\}$
7. Add new super landmark: $\Theta \leftarrow \Theta \cup \{R\}$
8. **For** all constrained landmarks $\theta_j \in R$ **do**

   Compute unconstrained state estimate $\hat{\theta}_j \leftarrow \beta_j + \Lambda_j Q_j^{-1}(Z_j - \beta_j)$

   Compute unconstrained covariance $\hat{P}_j \leftarrow \Lambda_j - \Lambda_j Q_j^{-1} \Lambda_j^T$

   End for
9. ML estimate covariance of ρ: $P_{\hat{\rho}} \leftarrow \left(\sum_{\theta_j \in R} P_{j,\rho}^{-1}\right)^{-1}$
10. ML estimate of ρ: $\hat{\rho} \leftarrow P_{\hat{\rho}}^{-1}\left(\sum_{\theta_j \in R} h_j\left(c_{n+1,j}; \theta_{j,\rho}\right) P_{j,p}^{-1}\right)$
11. **For** all constrained landmarks $\theta_j \in R$ **do**

    Rewind state to pre-particalized version: $\theta_j \leftarrow \beta_j; P_j = \Lambda_j$

    Conditional mean given ρ: $\theta_{j,\bar{\rho}} \leftarrow \theta_{j,\bar{\rho}} + P_{j,\bar{\rho}\rho} P_{j,\rho}^{-1}\left(g_j\left(c_{n+1,j}; \hat{\rho}\right) - \theta_{j,\rho}\right)$

    Conditional covariance: $P_{j,\bar{\rho}} \leftarrow P_{j,\bar{\rho}} - P_{j,\bar{\rho}\rho} P_{j,\rho}^{-1} P_{j,\bar{\rho}\rho}^T$

    Fix constrained variables: $\theta_{j,\rho} = g_j\left(c_{n+1,j}; \hat{\rho}\right); P_{j,\rho} \leftarrow [0]; P_{j,\bar{\rho}\rho} \leftarrow [0]$

    Replay since particalization $\theta_j \leftarrow \theta_j + P_j Q_j^{-1}\left(Z_j - \theta_j\right); P_j \leftarrow P_j - P_j Q_j^{-1} P_j^T$

    End for

b) Update algorithm

1. Update state: $\theta_j \leftarrow \theta_j + P_j\left(P_j + R\right)^{-1}\left(z - \theta_j\right)$
2. Update covariance $P_j = P_j - P_j\left(P_j + R\right)^{-1} P_j^T$
3. **If** x_j constrained: $\exists L \in \Theta, \theta_k \in L$ such that $\theta_j \in L$ and $\theta_j \neq \theta_k$

   Update measurement accumulator: $Z_j \leftarrow Z_j + Q_j\left(Q_j + R\right)^{-1}\left(z - Z_j\right)$

   Update accumulator covariance: $Q_j \leftarrow Q_j - Q_j\left(Q_j + R\right)^{-1} Q_j^T$

   else

   Update backup state, covariance: $\beta_j \leftarrow \theta_j; \Lambda_j = P_j$

   end if

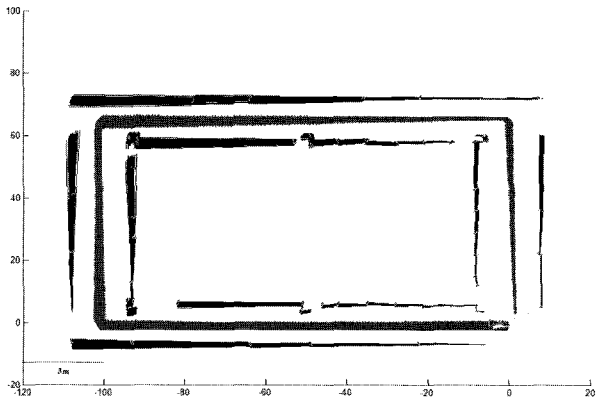
5. EXPERIMENT RESULT

This section shows both simulation and real world experiments to prove that integrating topological alignment with Rao-Blackwellized particle filter will result in a more accurate global map upon loop closure.

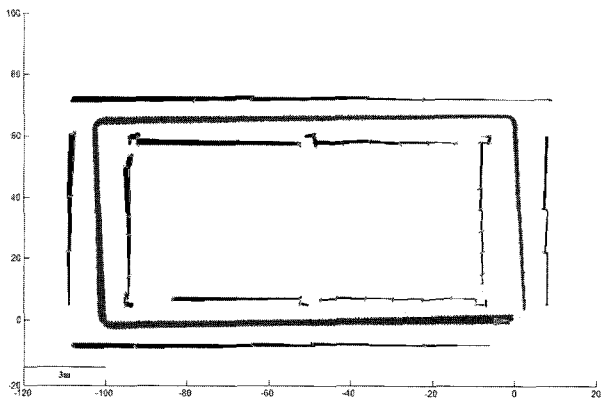
5.1. Simulation experiment

In this simulation part, we use the adaptive sampling approach described by Grisetti, Stachniss, and Burgard [27]. Noise was introduced by perturbing measurements and motions in proportion to their magnitude. Line segment features estimated covariance using the nearer end-point and a direction vector rather than the full data. Data association are assumed corrected.

The first set of experiments were designed to validate the advantages of using SUF initialization over normal initialization. We compute over 50 Monte Carlo trials running the mapping algorithm with and without SUF initialization conditioned on error models setup as described above. It is immediately apparent from Fig. 4 that the algorithm map with SUF initialization permits more accurate results than the uninitialized map in terms of real errors; the particles have smaller spread and are nearer the true state.



(a) Standard rbpf slam.



(b) Using SUF for initializing rbpf slam.

Fig. 4. Simulation in a normal environment.

As the robot goes further away from the origin, measurement noise is increased, but the estimation errors of the initialized algorithm is about 1.6 times better than that of uninitialized algorithms. Fig. 5 shows normalized estimation error squared (nees) of the robot's estimated pose with respect to the ground truth. It is worth noting that in a real environment, an algorithm with SUF initialization is much better in terms of correct data association and resampling. These issues will be further discussed in the real experiment section.

In the second set of experiments, different types of buildings with different motion error models are used to test the outperformance of enforcing constraints over relaxing constraints in mapping. The tests also determine the convergence characteristics of the algorithm when running in various conditions in respect to type of environment, areas, time, and error model. Additionally, we restricted the maximum range to at most 5 m. Figs. 8, 10, and 12 present the simulation results of our SLAM algorithm on different types of environments. In a typical square building (Fig. 7), the results illustrate the behavior of our system using a standard proposal distribution. The robot closed the loop properly in its line-based landmark map. In a circle corridor (Fig. 9), a polygonal approximation can be obtained. Finally, in a slight curved corridor (Fig. 11), a high threshold value

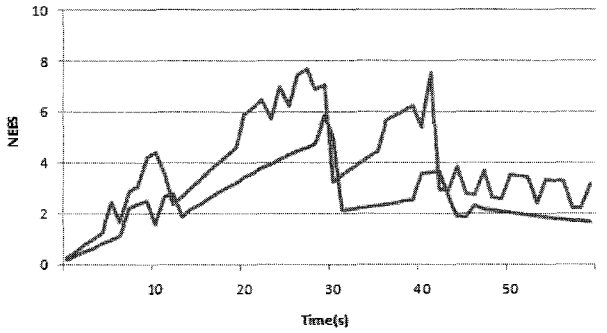


Fig. 5. The blue plot is the error for standard rbpf slam. The red plot is the error for our algorithm with SUF initialization.

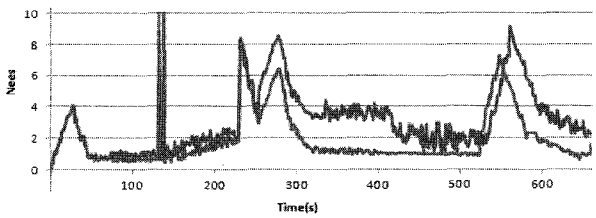


Fig. 6. The red plot is the error for standard (unconstrained) rbpf slam. The blue plot is the error for our algorithm with rectilinearity constraints.

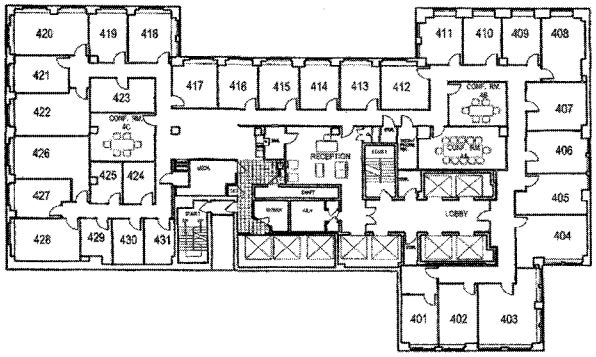


Fig. 7. Typical square building.

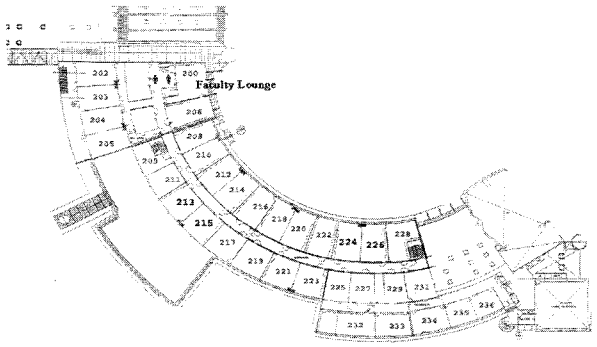


Fig. 11. A slight curved corridor.

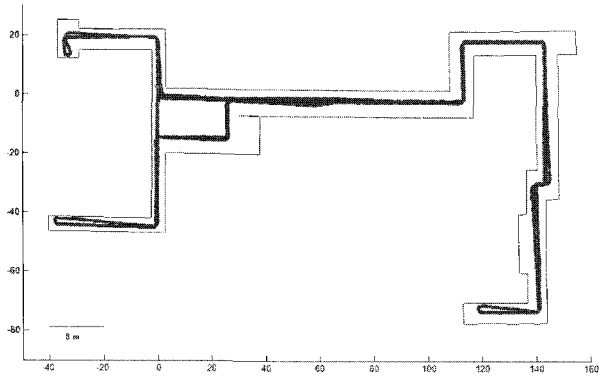


Fig. 8. Simulation result with threshold angle equal to 10 degrees.

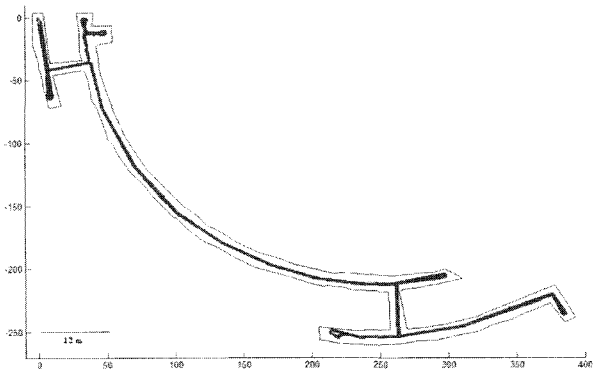


Fig. 12. Simulation result with threshold angle equal to 3 degrees but taking a long time.

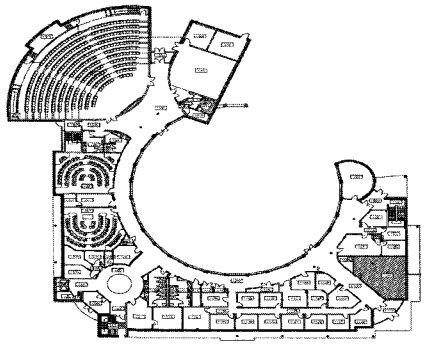


Fig. 9. A circle corridor.

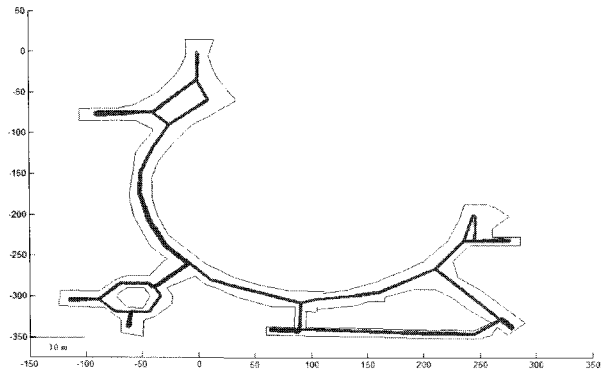


Fig. 10. Simulation result with threshold angle equal to 10 degrees.

for rectilinear set results in the uncorrected data, exhibiting significant error, particularly with respect to the robot's orientation. At slight curve line, the robot mistakenly decides constraint groups of features and produces a wrong map. To cover this problem, the threshold for the rectilinear model and the error model are set to low and we increase the number of constraint groups. However, this will slow down the performance. Note that due to large uncertainty, there exist spurious landmarks in the results; however, the environment's structure and the robot's trajectory were properly recovered. These results also indicate that fewer particles are required than in the unconstrained approach even in the case of high uncertainty parameters during initial pose initialization. All error models, environment complexity, and robot behaviors could affect SLAM result. Relaxed approximations can lead to poor data association or even divergence as in the last case. It is worthwhile noting that the filter can have real-time performance for 50 particles. Moreover, the rate of growth in the number of landmarks is approximately constant over the entire run. The built maps are correct and most of the time the robot returns to the map origin (with errors < 30cm). Fig. 6 shows normalized estimation error squared (nees) of the robot's estimated pose with respect to the ground truth, computed over 50 Monte Carlo trials for a typical

square environment. We can conclude that incorporating and enforcing constraints leads to a significant improvement in the resulting maps and a reduction in estimation error.

5.2. Real world experiment

For further evaluating the performance of SLAM, we perform the real experiments in a large environment. Two characteristics of our proposed algorithm are analyzed: robust data association and consistent convergence. Testing of our SLAM system was performed on a home service robot ISSAC shown in Fig. 13. Only odometry sensors and a USB video camera are equipped on the robot. Vision processing is based on OpenCV library while Bayes++ library is used for particle filter SLAM algorithm and scale unscented transform. All of this was implemented on a notebook computer (1.6GHz). The flowchart of the implemented algorithm is shown in Fig. 14. The testing environment was the floor of our L1 building laboratory. In order to perform these experiments autonomously, the robot was given a behavior-based exploration strategy. The behaviors include wall following and a look-back behavior. It was triggered by either needing to find a new landmark or turning at waypoints. The purpose of this behavior was to view the same feature from different directions and compute the topological relationship of new and old landmarks. The statistics of the experiments are provided in Table 1.

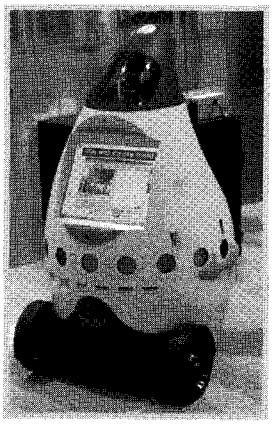


Fig. 13. ISSAC.

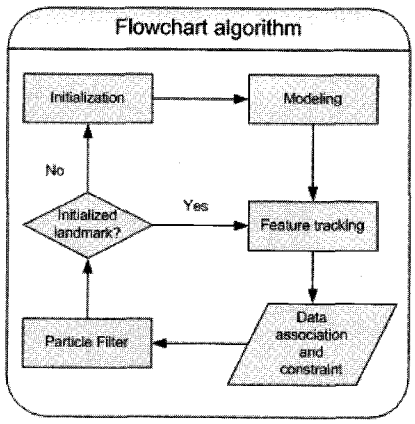


Fig. 14. Overall flowchart.

First, we test the data association performance of our algorithm. In SLAM, it is well known that, applying an angular velocity to a robot will results in a non-linear pose distribution. Without managing heading error, the filter will diverge rapidly. In addition, the non-linearity of the observation model is even aggravated through compounding with the relative distance model. Therefore, the nearest neighbor test for data association quickly fails in our experiment for normal particle filter approaches with the same number of particles. However, with SUF initialization, the robot can still associate data successfully. Moreover, high uncertainty accompanied by non-linear rotation effect is further reduced by pairing the relationship of features. Line equations that are near co-alignment and have distance between them smaller than a threshold are considered the same. Similarly, we apply the rule to the parallel and orthogonal lines relation. These constraints will be used in updating the probability of the robot position and in keeping the uncertainty manageable. Finally, inheriting the robustness of particle filter for model line tracking, in 96% of our experiments, the robot can successfully recover from "lost" due to infrequently erratic motions or temporary occlusions. Fig. 15 indicates the floor-plan. Some typical images captured from the robot camera during an actual experiment are revealed in Fig. 16.

Second, we vary the number of particles and the "effective sample size" to test the performance of sampling/resampling factors and convergence of our algorithm. One important factor, which greatly affects the consistent performance of the particle filter, is the sampling/resampling process. Incorporating constraints enables consistent mapping even with fewer particles-this leads to significant computational performance increases and the map can be created in real-time.

Table 1. Statistics of the environment.

	A square building
Dimension	20x40m
Particles (constrained)	30
Particles (unconstrained)	200
Average runtime (constrained 50 runs)	650s
Average runtime (unconstrained 50 runs)	1400s
Success rate	96%
Sensing range	5 m
Path length	120m
Number of landmarks	80
Constrained group	4

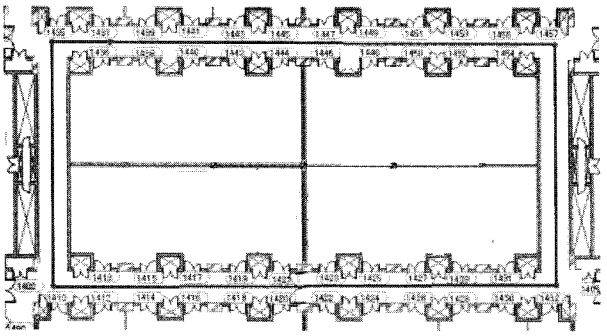


Fig. 15. Real map of environment.

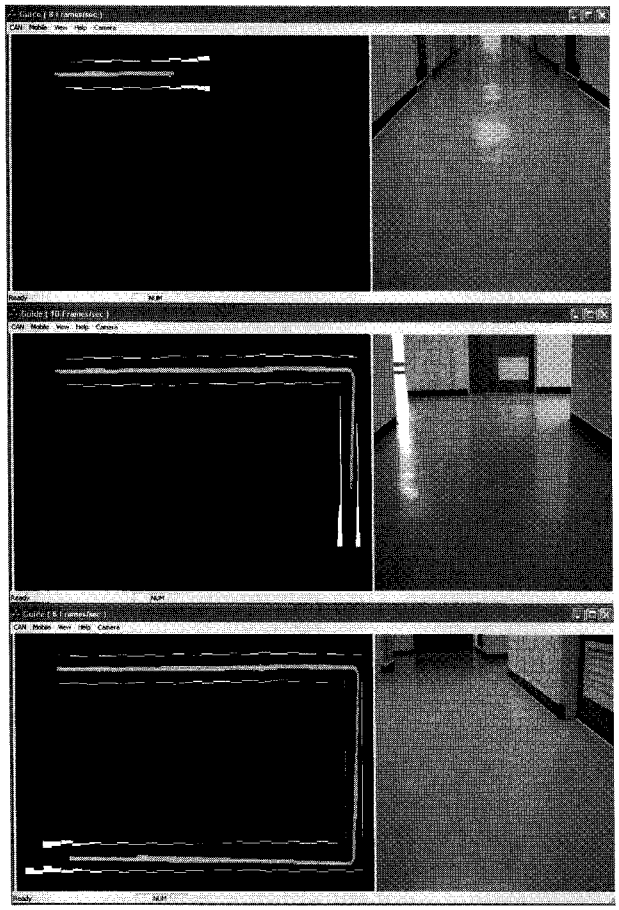


Fig. 16. Map of predicted landmarks and moving path.

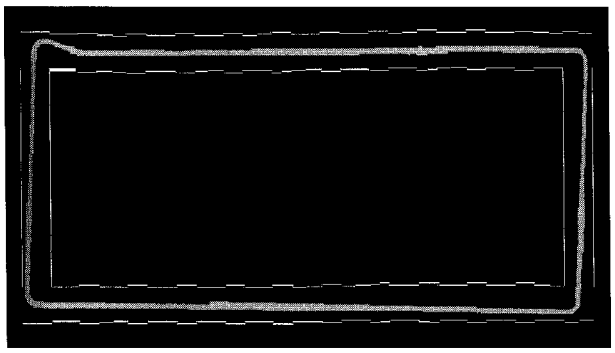


Fig. 17. Result of our algorithm.

Further, high effective sample size setting reduces the impact of resampling. The fact is that to reduce the uncertainty, we need to incorporate additional uncorrelated information to the vehicle position and relative indoor structures are considered a reasonably good choice. Hence, similar to simulation, the real experiments of our algorithm always yield a consistent map of the environment because uncertainty is well managed under SUF initialization and the integration of topological alignments. Fig. 17 shows a map created using 50 particles. In fact, our algorithm could reliably converge with 30 samples indicating an enormous improvement.

6. CONCLUSION

We have proposed an approach to use a single camera with natural landmark models for the SLAM algorithm of indoor mobile robots. The use of SUF to approximate highly non-linear observations makes the algorithm applicable in a general 3D bearing-only SLAM case. Based on assumptions that the indoor environment is almost structure, constraints are imposed on the computed landmarks to improve both the consistent and computational performance. Integration of these within a Rao-Blackwellized particle filter shows the possibility of the proposed algorithm for real-time application with the approximate rate of 10 Hz. The algorithm has been applied for a mobile robot, dubbed 'ISSAC', successfully. Our future work will include the solution to autonomously build the large, complicated map with a fully explored indoor environment.

APPENDIX A

We have: $A_i k_1 + B_i k = C_i$ for $i = 1,2$ or

$$\begin{bmatrix} 0 & \frac{-w_{z_{f_1}}}{\theta_{f_1}} & \frac{w_{y_{f_1}}}{\theta_{f_1}} \\ \frac{w_{z_{f_1}}}{\theta_{f_1}} & 0 & \frac{-w_{x_{f_1}}}{\theta_{f_1}} \\ \frac{-w_{y_{f_1}}}{\theta_{f_1}} & \frac{w_{x_{f_1}}}{\theta_{f_1}} & 0 \end{bmatrix} \begin{bmatrix} k_1 p_{x_1} \\ k_1 p_{y_1} \\ k_1 f \end{bmatrix} + \begin{bmatrix} C_{x_{f_1}} \\ C_{y_{f_1}} \\ C_{z_{f_1}} \end{bmatrix} = \begin{bmatrix} 0 & \frac{-w_{z_f}}{\theta_f} & \frac{w_{y_f}}{\theta_f} \\ \frac{w_{z_f}}{\theta_f} & 0 & \frac{-w_{x_f}}{\theta_f} \\ \frac{-w_{y_f}}{\theta_f} & \frac{w_{x_f}}{\theta_f} & 0 \end{bmatrix} \begin{bmatrix} kp_x \\ kp_y \\ kf \end{bmatrix} + \begin{bmatrix} C_{x_f} \\ C_{y_f} \\ C_{z_f} \end{bmatrix}, \tag{45}$$

where

$$A_1 = (\frac{-w_{z_{f_1}}}{\theta_{f_1}} p_{y_1} + \frac{w_{y_{f_1}}}{\theta_{f_1}} f),$$

$$B_1 = (\frac{w_{z_f}}{\theta_f} p_y + \frac{w_{y_f}}{\theta_f} f),$$

$$C_1 = C_{x_f} - C_{x_{f_1}},$$

$$A_2 = (\frac{w_{z_{f_1}}}{\theta_{f_1}} p_{x_1} + \frac{-w_{x_{f_1}}}{\theta_{f_1}} f),$$

$$B_2 = (\frac{-w_{z_f}}{\theta_f} p_x + \frac{w_{x_f}}{\theta_f} f),$$

$$C_2 = C_{y_f} - C_{y_{f_1}}.$$

REFERENCES

[1] D. O. Gorodnichy and W. W. Armstrong, "Single camera stereo for mobile robot world exploration," *Proc. of Vision Interface Conf. VI*, 1999.

[2] S. Thrun, A. Buecken, W. Burgard, D. Fox, T. Froehlinghaus, D. Henning, T. Hofmann, M. Krell, and T. Schmidt, "Map learning and high-speed navigation in RHINO," *AI-based Mobile Robots: Case Studies of Successful Robot Systems*, D. Kortenkamp, R. P. Bonasso, and R. Murphy, ed., MIT Press, 1998.

[3] D. Murray and J. Little, "Using real-time stereo vision for mobile robot navigation," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1998.

[4] W. Y. Jeong and K. M. Lee, "Visual SLAM with line and corner features," *Proc. of the IEEE/RSJ International Conference on Intelligent Robotics and System*, Beijing, China, October 2006.

[5] J. Folkesson, P. Jensfelt, and H. I. Christensen, "Vision SLAM in the measurement subspace," *Proc. of the IEEE International Conference on Robotics and Automation*, 2005.

[6] J. M. M. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," *Proc. of Robotics: Science and Systems, Philadelphia*, 2006.

[7] E. Eade and T. Drummond, "Edge landmarks in monocular slam," *Proc. of British Machine Vision Conference*, 2006.

[8] P. Smith, I. Reid, and A. Davison, "Real-time monocular slam with straight lines," *Proc. of British Machine Vision Conference*, 2006.

[9] E. Eade and T. Drummond, "Scalable monocular slam," *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[10] T. Lemaire and S. Lacroix. "Monocular-Vision based SLAM using line segments," *Proc of Robotic 3D Environment Cognition, Workshop at the International Conference Spatial Cognition*, 2006.

[11] P. Elinas, R. Sim, and J. J. Little, "σSLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution," *Proc. of the IEEE International Conference on Robotics and Automation*, 2006.

[12] R. Sim, P. Elinas, M. Griffin, A. Shyr, and J. J. Little, "Design and analysis of a framework for real-time vision-based SLAM using Rao-Blackwellised particle filters," *Proc. of the IEEE International Conference on Robotics and Automation*, Orlando, Florida, May 2006.

[13] M. Li, B. Hong, Z. Cai, and R. Luo, "Novel Rao-Blackwellized particle filter for mobile robot SLAM using monocular vision," *International Journal of Intelligent Technology*, vol. 1, no. 1, 2006.

[14] J. Sola, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," *Proc. of the IEEE International Conference on Robotics and Automation*, 2005.

[15] A. J. Davison, "Real-time simultaneous localization and mapping with a single camera," *Proc. of International Conference on Computer Vision*, 2003.

[16] T. Lemaire and S. Lacroix, "Monocular-vision based SLAM using line segments," *Proc. of the IEEE International Conference on Robotics and Automation*, 2007.

[17] E. Eade and T. Drummond, "Scalable monocular SLAM," *Proc. of Conference on Computer Vision and Pattern Recognition*, New York, USA, pp. 469-468, 2006.

[18] N. M. Kwok and G. Dissanayake, "Bearing-only SLAM in indoor environments using a modified particle filter," *Proc. of the Australasian Conference on Robotics & Automation*, 2003.

[19] M. Pupilli and A. Calway, "Real-time camera tracking using a particle filter," *Proc. of British Machine Vision Conference*, 2005.

[20] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework: Part 1," http://citeseer.nj.nec.com/531560.html, 2002.

[21] V. Lepetit and P. Fua, "Monocular model-based 3d tracking of rigid objects: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, 2005.

[22] S. J. Julier, "The scaled unscented transformation," http://www.cs.unc.edu/welch/kalman/media/pdf/ACC02-IEEE1357.PDF

[23] M. Montemerlo, S. Thrun, D. Koller, and B.Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," *Proc. of the AAAI National Conf. on Artificial Intelligence*, Edmonton, Canada,

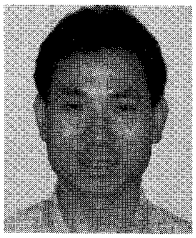
AAAI, 2002.

[24] W. Wen and H. Durrant-Whyte, "Model-based multi-sensor data fusion," *Proc. of IEEE Intl. Conf. on Robotics and Automation*, pp. 1720-1726, 1992.

[25] D. Rodriguez-Losada, F. Matia, A. Jimenez, and R. Galan, "Consistency improvement for SLAM -EKF for indoor environments," *Proc. of IEEE Intl. Conf. on Robotics and Automation*, pp. 418-423, 2006.

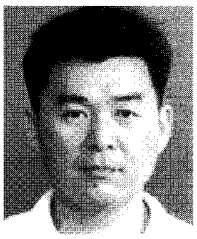
[26] K. R. Beevers, W. H. Huang, "Inferring and enforcing relative constraints in SLAM," *Proc. of the 7th Intl. Workshop on the Algorithmic Foundations of Robotics*, New York, USA, July 16-18, 2006.

[27] G. Grisetti, C. Stachniss, and W. Burgard, "Improving gridbased SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," *Proc. of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005.



Xuan-Dao Nguyen is currently a Ph.D. student at the Korea Institute of Science and Technology (KIST). He received his B.Sc. (1997) at Wollongong University, NSW, Australia. From 2002 to the present, he has studied at KIST. His research interests include SLAM, machine vision, and localization for mobile robots.

Bum-Jae You received the B.S. degree in Control and Instrumentation Engineering from Seoul National University, Seoul, Korea, in 1985 and the M.S. and Ph.D. degrees in Electrical and Electronic Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, in 1987 and 1991, respectively. He worked as Head of the Robotics Division at Turbo Tek Co., Ltd., Korea, from 1991 to 1994, conducting the development of a high-speed image processor and motion controller for robots. He joined the Korea Institute of Science and Technology (KIST), Seoul, Korea, in 1994, and is now the Center Director at the Intelligent Robotics Research Center. He was a Visiting Fellow at the Department of Computer Science, Yale University, New Heaven, for three months in 1997, conducting three-dimensional visual tracking of polyhedral objects. His research interests include vision-based robotics, vision-based control, real-time computer vision, intelligent service robots, as well as humanoid and digital signal processor applications. Dr. You is a Member of IEEE, the Korea Institute of Electrical Engineering, and Institute of Control, Robotics and Systems.



Sang-Rok Oh received the B.S. degree in Electronic Engineering from Seoul National University, Seoul, Korea, in 1980, and the M.S. and Ph.D. degrees in Electrical and Electronic Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daeduk, Korea, in 1982 and 1987, respectively. He worked as a Research Associate at the Systems Control Laboratory, KAIST, for ten months in 1987, conducting the design and implementation of multiprocessor-based automatic assembly machines for micro electronic components and robotic control systems for multilegged locomotion. In 1988, he joined the Korea Institute of Science and Technology (KIST), where he has been working as a Principal Research Engineer at the Intelligent System Control Research Center (ISCRC). During 2000-2003, he was a Head of ISCRC, KIST. In 1999, he also became Director of the Bio-mimetic Control National Research Laboratory, designated by the Ministry of Science and Technology, Korea. He was a Visiting Scientist at the T. J. Watson Research Center, IBM, Yorktown Heights, NY, from 1991 to 1992, conducting precision assembly using the magnetically levitated robot twist. He also worked as a Visiting Scientist at the Mechanical Engineering Laboratory, Tsukuba, Japan, for three months in 1995, investigating the area of mobile manipulation. Dr. Oh is a Member of IEEE, the Korea Institute of Electrical Engineering, the Korea Fuzzy Logic and Intelligent System Society, and Institute of Control, Robotics and Systems.