Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Stereo vision based SLAM using Rao-Blackwellised particle filter*

Er-yong WU[†1], Gong-yan LI[2], Zhi-yu XIANG[†‡1], Ji-lin LIU[1]

([1]*Department of Information Science and Electrical Engineering, Zhejiang University, Hangzhou 310027, China*)

([2]*National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing 100080, China*)

[†]E-mail: wueryong343@sohu.com; xiangzy@zju.edu.cn

Received July 4, 2007; revision accepted Nov. 5, 2007

Abstract: We present an algorithm which can realize 3D stereo vision simultaneous localization and mapping (SLAM) for mobile robot in unknown outdoor environments, which means the 6-DOF motion and a sparse but persistent map of natural landmarks be constructed online only with a stereo camera. In mobile robotics research, we extend FastSLAM 2.0 like stereo vision SLAM with "pure vision" domain to outdoor environments. Unlike popular stochastic motion model used in conventional monocular vision SLAM, we utilize the ideas of structure from motion (SFM) for initial motion estimation, which is more suitable for the robot moving in large-scale outdoor, and textured environments. SIFT features are used as natural landmarks, and its 3D positions are constructed directly through triangulation. Considering the computational complexity and memory consumption, Bkd-tree and Best-Bin-First (BBF) search strategy are utilized for SIFT feature descriptor matching. Results show high accuracy of our algorithm, even in the circumstance of large translation and large rotation movements.

Key words: Robot, Vision based SLAM, SIFT, Feature management
doi:10.1631/jzus.A071361 **Document code:** A **CLC number:** TP391.7

INTRODUCTION

Localization and mapping are the most important two issues for mobile robot navigation. The process of estimating both ego-motion and environment structure simultaneously is called SLAM, on which many research papers have been published in the past 15 years. Most of the researches concentrate on the indoor or semi-constructed environments, and sonar or laser range finder is adopted as the main sensor. More recently, extensive work has been done to solve SLAM problem using computer vision technologies (Davison, 2003; Davison *et al*., 2007; Sim *et al*., 2005; Elinas *et al*., 2006; Eade and Drummond, 2006). This work is mostly close to structure from motion (SFM), but prior to the numerical approximation methods, and pursuing the stochastic filter process. The main advantages of vision sensor are its low cost and nearness to human being visual effects. Unfortunately the projective transform of cameras makes the 3D information recovering a difficult job to be handled, although it is the most important for solving SLAM problem. Davison (2003) firstly put forward vision SLAM with a single camera, and established the well-suited extended Kalman filter (EKF) estimation framework. His system tracks few corners like features and estimates feature depth using a one-dimensional particle filter. Although his system is accurate and robust, it cannot be used in large-scale environments. Its full state EKF maintains N^2 covariance matrix for N landmarks, and updating each landmark needs $O(N^2)$ computation cost. Meanwhile EKF lacks self-rehabilitation ability and is sensitive to data association error. Particle filter (Arulampalam *et al*., 2002) is one kind of Monte Carlo application. As a new non-linear filter, it uses a set of discrete weighted samples to simulate the posterior probability of the estimated state, and carries out through state predicting, state updating, weight updating and resampling. Particle filter is not subject to system's linear hypothesis or sensor's Gaussian noise

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 60534070 and 60505017), and the Science Planning Project of Zhejiang Province (No. 2005C14008), China

hypothesis, and can deal with non-linear and non-Gaussian system effectively. But in SLAM, particle filter cannot deal with high dimensional estimation. Murphy (1999) introduced a Rao-Blackwellised particle filter for factorizing the full state posterior of SLAM, and Montemerlo *et al*.(2003) proposed the FastSLAM algorithm which puts forward Murphy's work to practical applications. FastSLAM makes the computational complexity down to $O(M\log N)$ with M particles and N landmarks. Meanwhile Montemerlo *et al*.(2003) introduced a new proposal function, which reduces the number of particles needed to the level of hundreds, which is critical for the Rao-Blackwellised particle filter. Sim *et al*.(2005) firstly presented stereo vision based SLAM using the FastSLAM algorithm, but his global SIFT feature matching influences the processing velocity seriously. Eade and Drummond (2006) proposed the scheme for monocular vision SLAM using the FastSLAM algorithm. Their patch-based feature searching and few landmarks maintained make the filter slowly converged and not suitable for large-scale environments. SIFT (Lowe, 2004) feature will be used as the natural landmark, although its extraction process is relatively time consuming. Robust SIFT feature will make accurate estimate of frame-to-frame motion easier, and can deal with camera's large translation and rotating movements effectively (Lowe, 2004). We will not use conventional Gaussian stochastic motion model which means little frame-to-frame motion, but rather recover motion from multiple view geometry. Well-established motion estimation will be due to SIFT's robustness and RANSAC (Hartley and Zisserman, 2003) iteration method. Then in FastSLAM framework, the motion and those 3D points are filtered sequentially, which makes the error reduced totally. In our application, there may be hundreds of thousands of features accumulated, whose matching and management become more urgent for practical application. So an efficient matching and memory management will be utilized, and one approach combining Bkd-tree (Procopiuc *et al*., 2002) with Best-Bin-First (BBF) search strategy (Beis and Lowe, 1997) is developed for fast SIFT matching.

The rest of the paper is organized as follows. Section 2 presents the framework of our implementation. Section 3 gives the naive motion model and observation model, introduces the initial motion estimation and 3D points reconstruction approaches, and puts forward the recursive estimation process similar to FastSLAM algorithm and the new feature management method. Experiments are given in Section 4 to verify our algorithm's effectiveness and robustness. Finally Section 5 concludes the paper.

FRAMEWORK OF STEREO VISION SLAM

We use one stereo camera with known calibration parameters to realize vision SLAM. The stereo vision SLAM framework is shown in Fig.1. The synchronously captured left and right images through distortion rectification are prepared for SIFT feature extracting. Each feature can be described as $\boldsymbol{f}$=(loc, s, o, des) which denotes location, scale, orientation and descriptor, respectively. Extracted features of the left image matched with those of the right image successfully will be defined as landmarks, and the 3D positions of landmarks will be determined uniquely by triangulation with known calibration parameters. So at

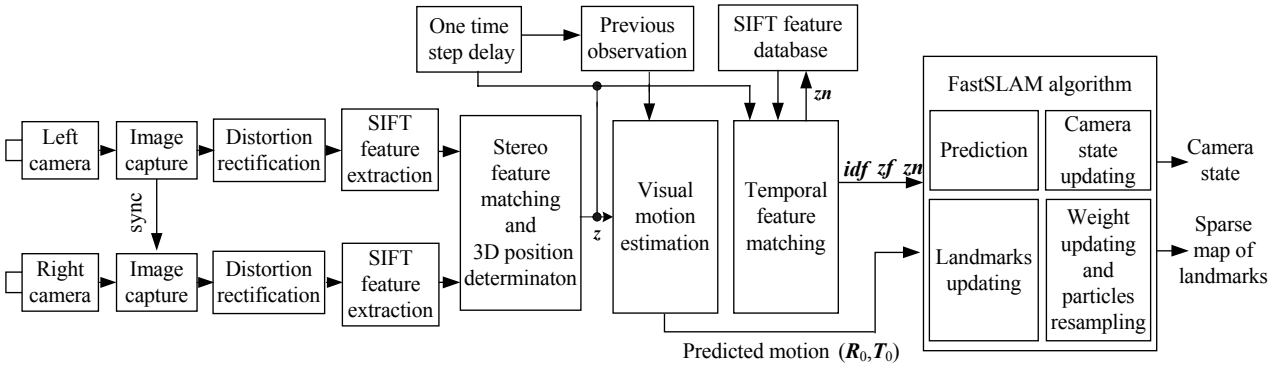

Fig.1 Framework of stereo vision simultaneous localization and mapping (SLAM)

time t, we define the current observation as a set of landmarks with known SIFT feature characteristics and its 3D positions refer to the left camera, which can be denoted as $\boldsymbol{z}_t=\{\boldsymbol{f}_t^i,\boldsymbol{P}_t^i,\boldsymbol{C}_t^i\}_{i=1}^N$, where $\boldsymbol{P}$ and $\boldsymbol{C}$ denote the point's 3D location and covariance matrix respectively, and N is the number of landmarks observed. At time t, the left camera's state will be defined as $\boldsymbol{s}_t=(x, y, z, heading, pitch, bank)^{\mathrm{T}}$, which denotes the camera's location and orientation. From those matched landmarks between previous observation and current observation, the initial motion between $\boldsymbol{s}_{t-1}$ and $\boldsymbol{s}_t$ can be determined uniquely in a robust way (Vergauwen *et al*., 2003; Nister *et al.*, 2004). In SIFT feature database, all newly observed SIFT features are saved for future matching, which is organized as $\boldsymbol{\Theta}_t=\{\boldsymbol{zn}_k\}_{k=1}^t$, and $\boldsymbol{zn}_k$ denotes the newly observed SIFT features at the kth time step. Current observation $\boldsymbol{z}_t$ is matched with $\boldsymbol{\Theta}_t$ to distinguish the already observed feature ***zf*** from the newly observed features ***zn***. ***idf*** denotes the correspondence indices between ***zf*** and the SIFT feature database $\boldsymbol{\Theta}_t$.

Then in FastSLAM 2.0 framework (Montemerlo *et al*., 2003), one Rao-Blackwellised particle filter maintains a set of particles like $\{\boldsymbol{s}_t^i,w_t^i,\boldsymbol{map}_t^i\}_{i=1}^M$ to estimate the camera's state $\boldsymbol{s}_t$ and every particle's sparse landmark map $\boldsymbol{map}_t^i$ which includes all observed landmarks' 3D positions and their covariances. With the initial motion parameter ($\boldsymbol{R}_0$, $\boldsymbol{T}_0$), camera's state $\boldsymbol{s}_t$ will be predicted by motion model firstly. Then for every associated landmark in ***zf*** and associated landmark in ***map***, an EKF is used to update $\boldsymbol{s}_t$ iteratively. With this more accurate camera state, every associated landmark in ***map*** will be updated by an EKF as well. Meanwhile, the newly observed landmarks in ***zn*** will be added to particle's ***map*** and feature database, respectively. Finally, according to the likelihood between the true and predicted observations, the particle's weight is renewed, and particles are resampled by a generic resampling algorithm.

SYSTEM MODEL

Unlike the conventional constant velocity motion model for each camera state hypothesis (Davison, 2003; Eade and Drummond, 2006), we prefer to calculate the frame-to-frame motion directly from multiple view geometry. With enough reliable matched landmarks between two consecutive observations, their motion can be determined effectively (Nister *et al*., 2004). Once this motion is obtained, the only thing left is to correct the small estimation error accumulated.

Robust initial motion estimation

Given two sets of matched three 3D points, their relative motion can be determined uniquely (Horn, 1987). In order to preclude the contamination of outliers, a robust estimation scheme, called random sampling consensus (RANSAC), is used to estimate camera-to-camera rotation and translation. The stereo robust visual motion estimation operates as follows:

(1) Match feature points between previous observation and current observation. Two sets of corresponding 3D points are obtained.

(2) Select three randomly associated points as hypothesis generator and compute the motion with RANSAC followed by iterative refinement. The scoring and iterative refinements are based on log-likelihood between the two subset points.

(3) Repeat (2) for a certain number of times.

The computed motion ($\boldsymbol{R}$, $\boldsymbol{T}$) are used as initialization for a nonlinear Leverberg-Marquardt minimization, which finds back the values of ($\boldsymbol{R}$, $\boldsymbol{T}$) that minimize the sum of distance between the predicted image location and true observation. This result is more accurate for the relative transformation between two cameras.

Motion model

Camera's state evolution is a process of rigid Euclidean transformation. For each particle, given the initial motion estimate $\boldsymbol{u}_t$=($\boldsymbol{R}$, $\boldsymbol{T}$) and previous camera state $\boldsymbol{s}_{t-1}$, the new camera's state $\boldsymbol{s}_t$ can be written as

$$\begin{cases}\boldsymbol{s}_t=f(\boldsymbol{s}_{t-1},\boldsymbol{u}_t)+\boldsymbol{\varepsilon}_t,\\ \overline{\boldsymbol{s}}_t\equiv(\boldsymbol{T}_t,\boldsymbol{R}_t),\\ \boldsymbol{R}_t=\boldsymbol{R}_{t-1}\boldsymbol{R},\ \boldsymbol{T}_t=\boldsymbol{R}_{t-1}\boldsymbol{T}+\boldsymbol{T}_{t-1},\\ \boldsymbol{s}_t=\overline{\boldsymbol{s}}_t+\boldsymbol{\varepsilon}_t,\end{cases}\tag{1}$$

where we assume the rotation matrix and Euler angle can be converted mutually so $\boldsymbol{s}_t$ can be represented as ($\boldsymbol{T}_t$, $\boldsymbol{R}_t$). An additive Gaussian noise with mean $\boldsymbol{\varepsilon}_t$ and covariance Q_t is added to the motion model.

Observation model

Suppose a landmark $\boldsymbol{\theta}$ with known 3D position $\boldsymbol{P}$ and covariance matrix $\boldsymbol{C}$, and camera lies at ($\boldsymbol{T}$, $\boldsymbol{R}$), then its projective image position (observed image point) $\boldsymbol{z}$ holds

$$\begin{cases} \boldsymbol{z} = h(\boldsymbol{s}_t, \boldsymbol{\theta}) + \boldsymbol{\upsilon}_t, \\ \boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \boldsymbol{K} \begin{pmatrix} \boldsymbol{R}^{\mathrm{T}} & -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{T} \\ \boldsymbol{R}_{\mathrm{c}}\boldsymbol{R}^{\mathrm{T}} & -\boldsymbol{R}_{\mathrm{c}}\boldsymbol{R}^{\mathrm{T}}\boldsymbol{T} + \boldsymbol{t}_{\mathrm{c}} \end{pmatrix} \boldsymbol{P} + \boldsymbol{\upsilon}_t, \end{cases} \tag{2}$$

where $\boldsymbol{z}=(z_1, z_2)^{\mathrm{T}}$, the homogeneous image points z_1 and z_2 denote left and right image positions respectively, and $\boldsymbol{K}$ is camera's intrinsic parameter matrix, ($\boldsymbol{R}$, $\boldsymbol{T}$) is current camera location, ($\boldsymbol{R}_{\mathrm{c}}$, $\boldsymbol{t}_{\mathrm{c}}$) is camera's extrinsic parameter. $\boldsymbol{\upsilon}_t$ is the observation Gaussian noise, and its covariance matrix is $\boldsymbol{O}_t$.

Inversely, given the corresponding image point pair and stereo camera's intrinsic and extrinsic parameters, landmark's 3D position can be obtained by triangulation, that is

$$\begin{cases} \boldsymbol{M} = \boldsymbol{K}(\boldsymbol{R}_{\mathrm{c}} \ \ \boldsymbol{t}_{\mathrm{c}}), \\ \boldsymbol{A} = \begin{pmatrix} K_{11} & 0 & u_1 - K_{13} \\ 0 & K_{22} & v_1 - K_{23} \\ u_2M_{31} - M_{11} & u_2M_{32} - M_{12} & u_2M_{33} - M_{13} \\ v_2M_{31} - M_{21} & v_2M_{32} - M_{22} & v_2M_{33} - M_{23} \end{pmatrix}, \\ \boldsymbol{b} = \begin{pmatrix} 0 & 0 & M_{14} - u_2M_{34} & M_{24} - v_2M_{34} \end{pmatrix}^{\mathrm{T}}, \\ \boldsymbol{P} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}, \\ \boldsymbol{J} = \begin{pmatrix} \dfrac{\partial \boldsymbol{P}}{\partial u_1} & \dfrac{\partial \boldsymbol{P}}{\partial v_1} & \dfrac{\partial \boldsymbol{P}}{\partial u_2} & \dfrac{\partial \boldsymbol{P}}{\partial v_2} \end{pmatrix}, \\ \boldsymbol{C}_m = \mathrm{diag}(\sigma_u^2, \sigma_v^2, \sigma_u^2, \sigma_v^2), \ \ \boldsymbol{C} = \boldsymbol{J}\boldsymbol{C}_m\boldsymbol{C}^{\mathrm{T}}, \end{cases} \tag{3}$$

where K_{ij} and M_{ij} are the ith row and jth column components of the camera's intrinsic parameter $\boldsymbol{K}$ and projective matrix $\boldsymbol{M}$, respectively; the first homogeneous image point $\boldsymbol{m}_1=(u_1, v_1, 1)^{\mathrm{T}}$, and the second homogeneous image point $\boldsymbol{m}_2=(u_2, v_2, 1)^{\mathrm{T}}$; σ_u and σ_v are covariances of image point at u and v direction, respectively; ($\boldsymbol{P}$, $\boldsymbol{C}$) are reconstructed 3D position and covariance relative to the left camera.

Recursive estimation

Our estimation algorithm is based on FastSLAM 2.0 algorithm (Montemerlo *et al*., 2003). Some modifications are necessary to make the algorithm compatible with our application. The biggest difference is that we use only visual information to complete SLAM. Some changes are made for proposal function of particle filter after considering that partial observation has been used for initial motion estimation. Another modification is that we expand the whole algorithm to multiple simultaneously observed landmarks. Finally, we replace all Jacobian matrix's computation with the Unscented Transform (Julier and Uhlmann, 1996) (replacing the linearization step), which is more suitable for rotation related motion model and observation model. Owing to space constraints, we do not discuss this unscented transform.

We seek to estimate the camera's trajectory and the global positions of SIFT landmarks simultaneously. Our goal is to estimate the posterior density

$$p(\boldsymbol{s}^t, \boldsymbol{\Theta}_t \mid \boldsymbol{z}^t, \boldsymbol{u}^t, \boldsymbol{n}^t) = p(\boldsymbol{s}^t \mid \boldsymbol{z}^t, \boldsymbol{u}^t, \boldsymbol{n}^t) \prod_{k=1}^{N} p(\boldsymbol{\theta}_k \mid \boldsymbol{s}^t, \boldsymbol{z}^t, \boldsymbol{u}^t, \boldsymbol{n}^t). \tag{4}$$

As shown in Eq.(4), the joint posterior density is factorized, which is firstly introduced by Murphy (1999), and this factorization process is called Rao-Blackwellised. The camera's trajectory, which denotes camera's state from time 1 to time t, is described as $\boldsymbol{s}^t$. The visual landmarks $\boldsymbol{\Theta}_t = \{\boldsymbol{\theta}_k\}_{k=1}^{N}$ means the landmarks accumulated up to time t. The observations which correspond to SIFT features extracted from captured images from time 1 to time t are described as $\boldsymbol{z}^t$. $\boldsymbol{u}^t$ denotes the estimated initial motion from time 1 to time t. The data association result $\boldsymbol{n}^t$, up to time t, denotes the association between SIFT feature descriptor of the current observation and SIFT feature descriptors already observed.

As described in (Montemerlo *et al*., 2003), Rao-Blackwellised particle filter uses a set of particles to represent the uncertainty of camera's trajectory. And within each particle, individual landmark map is maintained. For every observed landmark $\boldsymbol{\theta}$, a Gaussian state ($\boldsymbol{x}$, $\boldsymbol{C}$) is maintained if it is observed at time t, then its state will be updated by one independent EKF.

Camera trajectory updating

As described in (Montemerlo *et al*., 2003), let the posterior density of the camera's trajectory be represented by a set of particles like $\boldsymbol{s}^{(m),t}$, where m=1, 2, ..., M, then

$$\begin{cases} p(\boldsymbol{s}^t \mid \boldsymbol{z}^t, \boldsymbol{u}^t, \boldsymbol{n}^t) = \sum_{m=1}^{M} w_t^{(m)} \delta(\boldsymbol{s}^t - \boldsymbol{s}^{t,(m)}), \\ w_t^{(m)} \sim p(\boldsymbol{z}_t \mid \boldsymbol{s}^{t-1,(m)}, \boldsymbol{z}^{t-1}, \boldsymbol{u}^t, \boldsymbol{n}^t), \\ \boldsymbol{s}^{(m),t} \sim p(\boldsymbol{s}_t \mid \boldsymbol{s}^{t-1,(m)}, \boldsymbol{z}^{t-1}, \boldsymbol{u}^t, \boldsymbol{n}^t), \end{cases} \tag{5}$$

where δ is Dirac function, $w_t^{(m)}$ is the weight for the mth particle at time t.

With "pure" vision SLAM, initial motion estimation depends on the most recent two observations as discussed before. So here, the proposal density will be conditioned only by those landmarks associated with observation $\boldsymbol{z}^{t-2}$. Assuming that the first L observation elements are associated to $\boldsymbol{z}^{t-2}$, then

$$\begin{aligned} & p(\boldsymbol{s}_t \mid \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^t, \boldsymbol{n}^t) \\ & \overset{\text{Bayes}}{=} \eta p(\boldsymbol{s}_t \mid \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-1}, \boldsymbol{n}^t) p(\boldsymbol{z}_t \mid \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-1}, \boldsymbol{n}^t) \\ & \overset{\text{Markov}}{=} \eta p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t) p(\boldsymbol{z}_t \mid \boldsymbol{s}_t, \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-1}, \boldsymbol{n}^t). \end{aligned} \tag{6}$$

With known initial estimated $\boldsymbol{u}_t$, assume that $\boldsymbol{z}_{t-1}$ do not provide any information about $\boldsymbol{z}_t$, then Eq.(6) becomes

$$\triangleq \eta p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t) p(\boldsymbol{z}_t \mid \boldsymbol{s}_t, \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-2}, \boldsymbol{n}^t). \tag{7}$$

The Theorem of Total Probability is used to condition the second term of product on the currently observed landmarks $\boldsymbol{\theta}_{\boldsymbol{n}_t}$, then Eq.(7) becomes

$$\begin{aligned} & = \eta \int \Big[p(\boldsymbol{z}_t \mid \boldsymbol{\theta}_{\boldsymbol{n}_t}, \boldsymbol{s}_t, \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-2}, \boldsymbol{n}^t) \cdot \\ & \quad p(\boldsymbol{\theta}_{\boldsymbol{n}_t} \mid \boldsymbol{s}_t, \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-2}, \boldsymbol{n}^t) \Big] \mathrm{d}\boldsymbol{\theta}_{\boldsymbol{n}_t} \, p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t). \end{aligned} \tag{8}$$

The first term of the integrand is simply the observation model $p(\boldsymbol{z}_t \mid \boldsymbol{\theta}_{\boldsymbol{n}_t}, \boldsymbol{s}_t, \boldsymbol{n}_t)$. The second term of the integrand can also be simplified because $\boldsymbol{s}_t$, $\boldsymbol{n}_t$ and $\boldsymbol{u}_t$ do not provide any information about $\boldsymbol{\theta}_{\boldsymbol{n}_t}$ without $\boldsymbol{z}_t$, then Eq.(8) becomes

$$\begin{aligned} & \overset{\text{Markov}}{=} \eta \int \underbrace{\Big[p(\boldsymbol{z}_t \mid \boldsymbol{\theta}_{\boldsymbol{n}_t}, \boldsymbol{s}_t, \boldsymbol{n}_t) \Big]}_{\sim N(\boldsymbol{z}_t; h(\boldsymbol{s}_t, \boldsymbol{\theta}_{\boldsymbol{n}_t}), \boldsymbol{O}_t)} \underbrace{p(\boldsymbol{\theta}_{\boldsymbol{n}_t} \mid \boldsymbol{s}_t, \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-2}, \boldsymbol{n}^t)}_{\sim N(\boldsymbol{\theta}_{\boldsymbol{n}_t}; \boldsymbol{x}_{\boldsymbol{n}_t,t-1}^{(m)}, \boldsymbol{C}_{\boldsymbol{n}_t,t-1}^{(m)})} \Big] \mathrm{d}\boldsymbol{\theta}_{\boldsymbol{n}_t} \\ & \quad \cdot \underbrace{p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t)}_{\sim N(\boldsymbol{s}_t; f(\boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t), \boldsymbol{Q}_t)}. \end{aligned} \tag{9}$$

Assuming that there are $\{\boldsymbol{\theta}_{\boldsymbol{n}_t^k}\}_{k=1}^{L}$ landmarks associated to $\boldsymbol{z}_{t-2}$. Within Rao-Blackwellised particle filter framework, all landmarks are independent, then Eq.(9) becomes

$$\begin{aligned} & = \eta \underbrace{p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t)}_{\sim N(\boldsymbol{s}_t; f(\boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t), \boldsymbol{Q}_t)} \cdot \\ & \prod_{k=1}^{L} \int \underbrace{p(\boldsymbol{z}_t^k \mid \boldsymbol{\theta}_{\boldsymbol{n}_t^k}, \boldsymbol{s}_t, \boldsymbol{n}_t)}_{\sim N(\boldsymbol{z}_t^k; h(\boldsymbol{s}_t, \boldsymbol{\theta}_{\boldsymbol{n}_t^k}), \boldsymbol{O}_t)} \underbrace{p(\boldsymbol{\theta}_{\boldsymbol{n}_t^k} \mid \boldsymbol{s}_t, \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-2}, \boldsymbol{n}^t)}_{\sim N(\boldsymbol{\theta}_{\boldsymbol{n}_t^k}; \boldsymbol{x}_{\boldsymbol{n}_t^k,t-1}^{(m)}, \boldsymbol{C}_{\boldsymbol{n}_t^k,t-1}^{(m)})} \mathrm{d}\boldsymbol{\theta}_{\boldsymbol{n}_t^k}. \end{aligned} \tag{10}$$

This expression shows that the sampling distribution is a product of two Gaussian convolutions, multiplied by the third. With linearized motion function f and observation function h, obviously the expression can be calculated iteratively by convolution. Over index k, the proposal density can be updated effectively:

$$\begin{aligned} & \tilde{\boldsymbol{s}}_t^0 = f(\boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t) + \boldsymbol{\varepsilon}_t, \\ & \tilde{\boldsymbol{\Sigma}}_t^0 = \boldsymbol{Q}_t, \\ & \text{for} \quad k = 1 : L, \\ & \quad \left. \begin{aligned} & \tilde{\boldsymbol{z}}_t^k = h(\tilde{\boldsymbol{s}}_t^{k-1}, \boldsymbol{\theta}_{\boldsymbol{n}_t^k}) \big|_{\boldsymbol{\theta}_{\boldsymbol{n}_t^k} = \boldsymbol{x}_{\boldsymbol{n}_t^k,t-1}^{(m)}}, \\ & \boldsymbol{Z}_t^k = \boldsymbol{H}_\theta \boldsymbol{C}_{\boldsymbol{n}_t^k,t-1}^{(m)} \boldsymbol{H}_\theta^{\mathrm{T}} + \boldsymbol{H}_s \tilde{\boldsymbol{\Sigma}}_t^{k-1} \boldsymbol{H}_s^{\mathrm{T}} + \boldsymbol{O}_t, \\ & \tilde{\boldsymbol{\Sigma}}_t^k = \{\boldsymbol{H}_s \{\boldsymbol{Z}_t^k\}^{-1} \boldsymbol{H}_s^{\mathrm{T}} + \{\tilde{\boldsymbol{\Sigma}}_t^{k-1}\}^{-1}\}^{-1}, \\ & \tilde{\boldsymbol{s}}_t^k = \tilde{\boldsymbol{s}}_t^{k-1} + \tilde{\boldsymbol{\Sigma}}_t^k \boldsymbol{H}_s^{\mathrm{T}} \{\boldsymbol{Z}_t^k\}^{-1} (\boldsymbol{z}_t^k - \tilde{\boldsymbol{z}}_t^k). \end{aligned} \right\} \\ & \text{end} \end{aligned} \tag{11}$$

The Jacobian matrices are

$$\begin{cases} \boldsymbol{H}_s = \nabla_{\boldsymbol{s}_t} h(\boldsymbol{s}_t, \boldsymbol{\theta}_{\boldsymbol{n}_t^k}) \big|_{\boldsymbol{s}_t = \tilde{\boldsymbol{s}}_t^{k-1}, \boldsymbol{\theta}_{\boldsymbol{n}_t^k} = \boldsymbol{x}_{\boldsymbol{n}_t^k,t-1}^{(m)}}, \\ \boldsymbol{H}_\theta = \nabla_{\boldsymbol{\theta}_{\boldsymbol{n}_t^k}} h(\boldsymbol{s}_t, \boldsymbol{\theta}_{\boldsymbol{n}_t^k}) \big|_{\boldsymbol{s}_t = \tilde{\boldsymbol{s}}_t^{k-1}, \boldsymbol{\theta}_{\boldsymbol{n}_t^k} = \boldsymbol{x}_{\boldsymbol{n}_t^k,t-1}^{(m)}}. \end{cases} \tag{12}$$

Then, a sample is drawn for $\boldsymbol{s}_t^{(m)}$ which obeys Gaussian distribution $N(\tilde{\boldsymbol{s}}_t^L, \tilde{\boldsymbol{\Sigma}}_t^L)$. Every particle draws one distinctive sample from its own proposal density. Then all samples constitute the newly predicted particles clouds.

Landmark position updating

With particles extracted from the proposal density, the next step is to update the states of those

landmarks which are associated to current observation. For the $\boldsymbol{n}_t^k$ independent landmarks maintained by the mth particle, we expand its posterior using Bayes Rule.

$$\begin{aligned} & p(\boldsymbol{\theta}_{\boldsymbol{n}_t^k} \mid \boldsymbol{s}^{t,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^t, \boldsymbol{n}^t) \\ & \overset{\text{Bayes}}{=} \eta p(\boldsymbol{z}_t \mid \boldsymbol{\theta}_{\boldsymbol{n}_t^k}, \boldsymbol{s}^{t,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-1}, \boldsymbol{n}^t) p(\boldsymbol{\theta}_{\boldsymbol{n}_t^k} \mid \boldsymbol{s}^{t,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-1}, \boldsymbol{n}^t) \\ & \overset{\text{Markov}}{=} \eta \underbrace{p(\boldsymbol{z}_t \mid \boldsymbol{\theta}_{\boldsymbol{n}_t^k}, \boldsymbol{s}_t^{(m)})}_{\sim N(\boldsymbol{z}_t^k; h(\boldsymbol{s}_t^{(m)}, \boldsymbol{\theta}_{\boldsymbol{n}_t^k}), \boldsymbol{O}_t)} \underbrace{p(\boldsymbol{\theta}_{\boldsymbol{n}_t^k} \mid \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^{t-1}, \boldsymbol{z}^{t-1}, \boldsymbol{n}^{t-1})}_{\sim N(\boldsymbol{\theta}_{\boldsymbol{n}_t^k}; \boldsymbol{x}_{\boldsymbol{n}_t^k,t-1}^{(m)}, \boldsymbol{C}_{\boldsymbol{n}_t^k,t-1}^{(m)})}. \end{aligned} \quad (13)$$

For those elements of observation not associated to landmark $\boldsymbol{\theta}_{\boldsymbol{n}_t^k}$, we assume that they do not provide any information for the landmark's state updating. With linearized observation function h, the first term in Eq.(13) can be expressed as Gaussian. The product of two Gaussians can be obtained using the standard EKF updating equation:

$$\begin{cases} \tilde{\boldsymbol{z}}_t^k = h(\boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{\theta}_{\boldsymbol{n}_t^k}), \ \boldsymbol{H}_\theta = \nabla_{\boldsymbol{\theta}_{\boldsymbol{n}_t^k}} h(\boldsymbol{s}_t, \boldsymbol{\theta}_{\boldsymbol{n}_t^k})\big|_{\boldsymbol{s}_t = \tilde{\boldsymbol{s}}_t^{k-1}, \boldsymbol{\theta}_{\boldsymbol{n}_t^k} = \boldsymbol{x}_{\boldsymbol{n}_t^k,t-1}^{(m)}}, \\ \boldsymbol{Z}_t^k = \boldsymbol{H}_\theta \boldsymbol{C}_{\boldsymbol{n}_t^k,t-1}^{(m)} \boldsymbol{H}_\theta^{\mathrm{T}} + \boldsymbol{O}_t, \ \boldsymbol{K}_t = \boldsymbol{C}_{\boldsymbol{n}_t^k,t-1}^{(m)} \boldsymbol{H}_\theta \{\boldsymbol{Z}_t^k\}^{-1}, \\ \boldsymbol{x}_{\boldsymbol{n}_t^k,t}^{(m)} = \boldsymbol{x}_{\boldsymbol{n}_t^k,t-1}^{(m)} + \boldsymbol{K}_t(\boldsymbol{z}_t^k - \tilde{\boldsymbol{z}}_t^k), \ \boldsymbol{C}_{\boldsymbol{n}_t^k,t}^{(m)} = (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_\theta) \boldsymbol{C}_{\boldsymbol{n}_t^k,t-1}^{(m)}. \end{cases} \quad (14)$$

If the landmark is simply not observed at time t, its state will not be altered. Those newly observed landmarks are added to particle's individual landmarks map after having been converted to a global coordinate system.

Weight calculating

After camera's trajectory state and every associated landmark's position having been updated, each particle's importance weight must be updated subsequently. Let the importance weight $w_t^{(m)}$ be conditioned twice on $\boldsymbol{s}_t$ and $\boldsymbol{\theta}_{\boldsymbol{n}_t}$ with the Theorem of Total Probability, then

$$\begin{aligned} & w_t^{(m)} \sim p(\boldsymbol{z}_t \mid \boldsymbol{s}^{t-1,(m)}, \boldsymbol{z}^{t-1}, \boldsymbol{u}^t, \boldsymbol{n}^t) \\ & = \iint \Big[p(\boldsymbol{z}_t \mid \boldsymbol{\theta}_{\boldsymbol{n}_t}, \boldsymbol{s}_t, \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-1}, \boldsymbol{n}^t) \cdot \\ & p(\boldsymbol{\theta}_{\boldsymbol{n}_t} \mid \boldsymbol{s}_t, \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t, \boldsymbol{z}^{t-1}, \boldsymbol{n}^t) \Big] \mathrm{d}\boldsymbol{\theta}_{\boldsymbol{n}_t} p(\boldsymbol{s}_t \mid \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^t) \mathrm{d}\boldsymbol{s}_t \\ & \overset{\text{Markov}}{=} \prod_{k=1}^{L} \iint \underbrace{\Big[p(\boldsymbol{z}_t^k \mid \boldsymbol{\theta}_{\boldsymbol{n}_t^k}, \boldsymbol{s}_t) \cdot}_{\sim N(\boldsymbol{z}_t^k; h(\boldsymbol{s}_t, \boldsymbol{\theta}_{\boldsymbol{n}_t^k}), \boldsymbol{O}_t)} \\ & \underbrace{p(\boldsymbol{\theta}_{\boldsymbol{n}_t^k} \mid \boldsymbol{s}_t, \boldsymbol{s}^{t-1,(m)}, \boldsymbol{u}^{t-1}, \boldsymbol{z}^{t-1}, \boldsymbol{n}^{t-1}) \Big]}_{\sim N(\boldsymbol{\theta}_{\boldsymbol{n}_t^k}; \boldsymbol{x}_{\boldsymbol{n}_t^k,t-1}^{(m)}, \boldsymbol{C}_{\boldsymbol{n}_t^k,t-1}^{(m)})} \mathrm{d}\boldsymbol{\theta}_{\boldsymbol{n}_t^k} \underbrace{p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t)}_{\sim N(\boldsymbol{s}_t; f(\boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t), \boldsymbol{Q}_t)} \mathrm{d}\boldsymbol{s}_t. \end{aligned} \quad (15)$$

With linearized motion function f and observation function h, the three terms in this expression are all Gaussians, corresponding to the observation model, the landmark estimate at time $t-1$, and the motion model, respectively. Two applications of the convolution theorem yield

$$\begin{cases} \boldsymbol{L}_t^k = \boldsymbol{H}_s^k \boldsymbol{Q}_t (\boldsymbol{H}_s^k)^{\mathrm{T}} + \boldsymbol{H}_\theta^k \boldsymbol{C}_{\boldsymbol{n}_t^k,t-1}^{(m)} (\boldsymbol{H}_\theta^k)^{\mathrm{T}} + \boldsymbol{O}_t, \\ w_t^{(m)} \sim \prod_{k=1}^{L} N(\boldsymbol{z}_t^k; \tilde{\boldsymbol{z}}_t^k, \boldsymbol{L}_t^k), \\ w_t^{(m)} = \prod_{k=1}^{L} \Big\{ \exp\Big(-\frac{1}{2} (\boldsymbol{z}_t^k - \tilde{\boldsymbol{z}}_t^k)^{\mathrm{T}} \{\boldsymbol{L}_t^k\}^{-1} (\boldsymbol{z}_t^k - \tilde{\boldsymbol{z}}_t^k) \Big) \\ \qquad \cdot |2\pi \boldsymbol{L}_t^k|^{-1/2} \Big\}. \end{cases} \quad (16)$$

With a large number of feature observations, special consideration must be taken when computing $w_t^{(m)}$. The product of many Mahalanobis distance likelihoods may lead to numerical overflow or underflow. So we use log-likelihood to compute the importance weight. Meanwhile, in order to preclude outlier correspondences from significantly affecting the likelihood, the maximum observation innovation threshold T_l is set:

$$\begin{cases} S_k = \min\Big(T_l, \ (\boldsymbol{z}_t^k - \tilde{\boldsymbol{z}}_t^k)^{\mathrm{T}} \{\boldsymbol{L}_t^k\}^{-1} (\boldsymbol{z}_t^k - \tilde{\boldsymbol{z}}_t^k)\Big), \\ \log(w_t^{(m)}) = \sum_{k=1}^{L} \Big(-\frac{1}{2} \Big(\log|2\pi \boldsymbol{L}_t^k| + S_k \Big) \Big). \end{cases} \quad (17)$$

Feature management

With robot moving further, more features are accumulated, and one-to-all matching strategy becomes more time consuming. So some researchers use kd-tree to construct SIFT features descriptors, and a new searching strategy named BBF is used for real-time matching (Beis and Lowe, 1997; Sim *et al*., 2005; Jensfelt *et al*., 2006). But with continuously added feature descriptors, kd-tree becomes larger and larger, and unbalanced, which will lead to searching efficiency's rapid decrease. Meanwhile, for a 128 dimensional SIFT feature descriptor (we use only 36

dimensions), every feature will consume about 0.5K bytes memory, and space utilization must be considered for large scale application or textured environment. Rather than kd-tree, we adopt Bkd-tree (Procopiuc *et al*., 2002) to construct SIFT feature descriptors, which is one kind of improved rebalanced K-D-B-Tree (Robinson, 1981), and has high space utilization and dynamic updating ability. Its paging mechanism makes most data be stored in external memory, and its efficient bulk loading algorithm makes Bkd-tree have high I/O throughput capacity. This will ease tree balance problem and enhance internal memory utilization efficiency.

For Nearest-Neighbor (NN) search, primitive strategy of searching all the bins near the query point is not suitable for high dimensional SIFT descriptor matching. This is due to so much nearby bins existing, which will reduce the search efficiency rapidly. BBF is an approximate NN search strategy in high-dimensional spaces (Beis and Lowe, 1997). It maintains one ascendingly arranged priority queue according to the distance between the query point and its nearby bin's border. Only a fixed number of bins associated with forefront elements of priority queue are examined, which will reduce the searching time tremendously. So we immigrate BBF search strategy to Bkd-tree to realize SIFT descriptor matching. This will be more suitable for vision SLAM with a large quantity of image features.

Another issue for feature management is about landmarks arrangement. Because every particle maintains one landmark cloud, every landmark maintains one 3D position and their 3×3 covariance matrix. For our application, their memory consumption needs to be considered with hundreds of thousands of landmarks. In order to reduce the computational complexity of FastSLAM algorithm to $O(M\log N)$ from $O(N^2)$, where M is particle number and N is landmark number, as described in (Montemerlo *et al*., 2003), kd-tree is used to organize the landmarks. With the strategy of sharing subtree, unnecessary copies in resampling stage of Rao-Blackwellised particle filter are reduced. We also immigrate this kd-tree to Bkd-tree with subtree extraction strategy, one for memory utilization, one for computational efficiency. Although this will make the updating procedure more complicated to implement, it results in a tremendous savings of both memory and computation.

EXPERIMENTAL RESULTS

The experiment was carried out on a Pioneer 3 robot platform from ActiveMedia Company. The camera used in the experiment is STH-MDCS2 of Videre Design Company. The image is grabbed at 1 Hz, and the image size is 640×480. In order to validate our algorithm's insensitivity to big translation and rotation, the images were discarded if the camera has not moved more than 30 cm or 5.0°.

At the beginning, the STH-MDCS2 stereo camera was calibrated for getting intrinsic and distortion parameters. Then we manually drove the robot by joystick without mechanic odometer data collected. We made the robot moving around the scene, going back to the start point after a loop is finished. Totally the robot traveled about 35.1 m, while only 76 images were used for experiments. Fig.2 displays a subset of images (every 8th image).

Fig.2 Sample images from the extracted frames (every 8th image)

All programs were developed based on iLab Neuromorphic Vision C++ Toolkit (Laurent, 2006). Within iNVT library, SIFT feature was extracted similar to (Lowe, 2004), but only a 36-dimensional descriptor was used. In FastSLAM, we use 100 particles, and set the maximum observation innovation threshold T_l to 4.0.

Fig.3 depicts the map constructed for maximum likelihood particle at the end of exploration. This map is not post-processed to remove noise or perform any global optimization. The circular trajectory indicates the trajectory of the best sample and the blue points are sparse map of landmarks. Let the camera's initial position be (0, 0, 0), while the final position stays at (−0.16, 0.10, −0.06) m, the total accumulation error is

0.20 m. The initial orientation (Euler angle) of the camera is at (0, 0, 0), the final orientation stays at (2.9°, −1.2°, −0.6°), and the maximum error for the heading angle is about 2.9°. We conclude that the accumulated loop error was less than 1.6% for our algorithm.

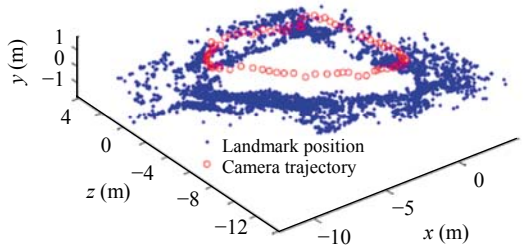

Fig.3 Camera trajectory and landmark's positions

Fig.4 shows the localization comparison of dead-recking results based on the initial estimation only, our stereo vision SLAM output and GPS (NovAtel OEM4 GPS Differential System, the precision is 0.02 m) output. The final accumulated error with only initial motion estimation is about 1.31 m, and heading angle error accumulates to about 24.6°. So we conclude that the error accumulated by the initial motion estimation is much bigger than SLAM approach. This is due to the effective adjustments of both the camera trajectory and landmarks position in SLAM process.

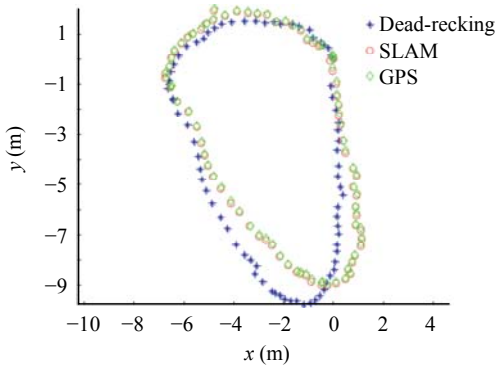

Fig.4 Localization comparison of only visual motion estimation, stereo vision SLAM output and high precision GPS output

Fig.5 shows the resulted DEM (digital elevation map) for 3D reconstructed landmarks. We set the grid size as 0.1 m×0.1 m. The DEM reflects the true terrain although the points are sparse.

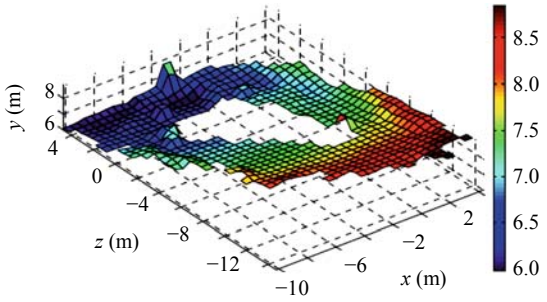

Fig.5 Constructed DEM of the sparse landmarks map

Our algorithm is implemented on a computer with 2.8 G Pentium IV CPU and 512 M memory. Fig.6 shows the engrowing process of SIFT feature accumulated number (landmark number) in SIFT feature database with time step. On average 250 newly observed features are found for a frame. At last, 19161 features are accumulated. With such fast and large accumulated features, naive implementation of matching or memory management precludes our algorithm to practical applications.

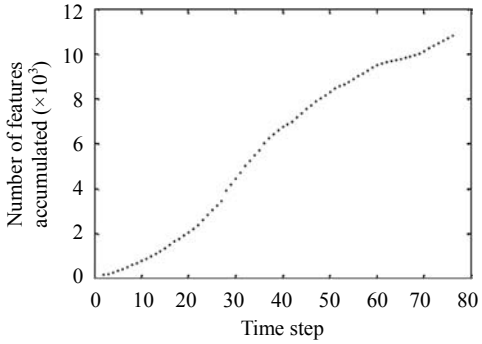

Fig.6 Number of features accumulated

Fig.7a shows the matching time of our Bkd-tree and BBF combined searching method, and Fig.7b shows the matching time of the original one-to-all matching scheme. From Fig.7, the matching is accelerated greatly (more than 100 times). The average matching time for our approach is 0.0356 s, while the one-to-all global method needs 3.9 s.

Each particle maintains its own corresponding landmark map. For M particles and N landmarks, it will consume $48MN$ bytes memory. In our application it is about 87 Mbytes. The continuingly accumulated landmarks may bring memory allocation problems to the operating system (Windows XP for us). In our

scheme, we set page size to 16 kB as suggested by Procopiuc *et al*.(2002), and primary memory buffer 64 Mbytes. Due to these major contributions, our approach opens the door to large-scale vision SLAM, without worrying about the memory limitation and matching speed un-prediction.

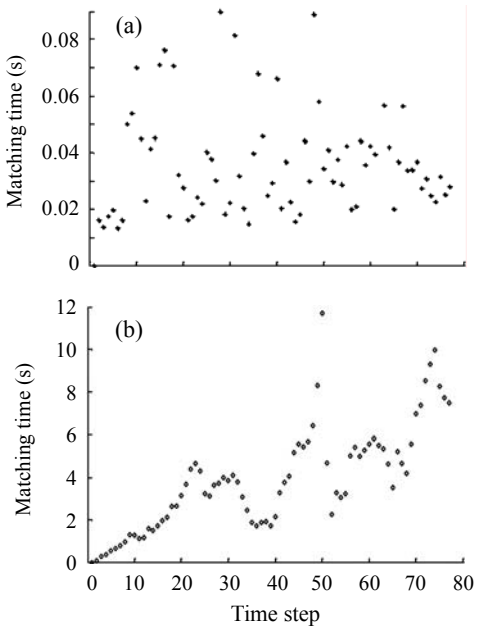

Fig.7 (a) Matching time with combined Bkd-tree and BBF; (b) Matching time with one-to-all approach

CONCLUSION

In this paper, a stereo "pure" vision SLAM is presented. SIFT is used as image feature, and with the multiple view geometry of cameras, the frame-to-frame rotation and translation are estimated in a robust way. Then in FastSLAM framework, naive motion model and observation model are proposed, and the landmark's 3D position is obtained directly from triangulation. Within camera trajectory updating, two frame delay observations are used for filtering, which mostly reduces the computational complexity without losing much accuracy. Meanwhile, efficient landmark state updating and weight computation approaches are advanced. For computation saving, we utilize one approach combined with Bkd-tree and BBF searching for matching between current observation and SIFT feature database. In order to save landmark copies during the particle resampling stage, the sharing sub-tree approach is used with more effective Bkd-tree and higher memory utilization. Experiments in outdoor textured environment show that our algorithm not only keeps the accumulation error at a level of less than 1.7%, but also enhances the memory space utilization and improves the computational efficiency.

References

Arulampalam, M.S., Maskell, S., Gordon, N., 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. on Signal Processing*, **50**(2):174-188. [doi:10.1109/78.978374]

Beis, J.S., Lowe, D.G., 1997. Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.1000-1006. [doi:10.1109/CVPR.1997.609451]

Davison, A.J., 2003. Real-Time Simultaneous Localisation and Mapping with a Single Camera. Proc. Ninth IEEE Int. Conf. on Computer Vision, p.1403-1410. [doi:10.1109/ICCV.2003.1238654]

Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O., 2007. MonoSLAM: real-time single camera SLAM. *IEEE Trans. on Pattern Anal. Machine Intell.*, **29**(6):1052-1067. [doi:10.1109/TPAMI.2007.1049]

Eade, E., Drummond, T., 2006. Scalable Monocular SLAM. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, p.469-476.

Elinas, P., Sim, R., Little, J.J., 2006. σSLAM: Stereo Vision Slam Using the Rao-Blackwellised Particle Filter and a Novel Mixture Proposal Distribution. Proc. IEEE Int. Conf. on Robotics and Automation, p.1564-1570.

Hartley, R., Zisserman, A., 2003. Mutiple View Geometry in Computer Vision. Cambridge University Press.

Horn, B., 1987. Closed-form solution of absolute orientation using unit quaternion. *J. Opt. Soc. Am.*, **4**:629-642.

Jensfelt, P., Kragic, D., Folkesson, J., Bjorkman, M., 2006. A Framework for Vision Based Bearing Only 3D SLAM. Proc. IEEE Int. Conf. on Robotics and Automation, p.1944-1950.

Julier, S., Uhlmann, J., 1996. A General Method for Approximating Nonlinear Transformations of Probability Distributions. Technical Report, Department of Engineering Science, University of Oxford. Http://www.robots.ox.ac.uk/siju/work/publications/letter size/Unscented.zip

Laurent, I., 2006. The iLab Neuromorphic Vision C++ Toolkit. University of Southern California. Http://ilab.usc.edu/toolkit/home.shtml

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision*, **60**(2):91-110. [doi:10.1023/B:VISI.0000029664.99615.94]

Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., 2003. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. Proc. Int. Joint Conf. on Artificial Intelligence,

p.1151-1156.

Murphy, K., 1999. Bayesian Map Learning in Dynamic Environments. *In*: Advances in Neural Information Processing Systems. MIT Press, Denver, USA, p.1015-1021.

Nister, D., Naroditsky, O., Bergen, J., 2004. Visual Odometry. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, p.652-659.

Procopiuc, O., Agarwal, P., Arge, L., Vitter, J.S., 2002. Bkd-tree: A Dynamic Scalable Kd-tree. *In*: Advances in Spatial and Temporal Databases (SSTD). Springer Press, Berlin/Heidelberg, **2750**:46-65. [doi:10.1007/b11839]

Robinson, J.T., 1981. The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes. Proc. Int. Conf. on Management of Data, p.10-18.

Sim, R., Elinas, P., Griffin, M., Little, J.J., 2005. Vision-based SLAM Using the Rao-Blackwellised Particle Filter. Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics, p.9-16.

Vergauwen, M., Pollefeys, M., Goll, L.V., 2003. A stereo-vision system for support of planetary surface exploration. *Machine Vision and Applications*, **14**(1):5-14. [doi:10.1007/s00138-002-0097-7]