SLAM WITH KLT POINT FEATURES

Yoichi Nakaguro,[1] *Matthew Dailey,*[2] *Stanislav Makhanov*[1]

[1]Sirindhorn International Institute of Technology, Thammasat University
P.O. Box 22, Thammasat-Rangsit Post Office, Pathumthani 12121 Thailand

[2]Computer Science and Information Management, Asian Institute of Technology
P.O. Box 4, Klong Luang, Pathumthani 12120 Thailand

E-mail: ynaka96@yahoo.com, mdailey@ait.ac.th, makhanov@siit.tu.ac.th

ABSTRACT

In the simultaneous localization and mapping (SLAM) problem, a mobile robot must localize itself in an unknown environment using its sensors and at the same time construct a map of that environment. While SLAM utilizing costly (expensive, heavy and slow) laser range finders as a sensor has been very successful in both indoor and outdoor environments, large-scale SLAM with cost-effective vision-based sensors has yet to be realized. In this paper, we evaluate the effectiveness of one possible low-cost vision-based approach to SLAM. We take 3D points constructed from Kanade-Lucas-Tomasi (KLT) image feature points in trinocular camera images as the basic landmarks for SLAM. We demonstrate the feasibility of KLT-based SLAM by conducting an experiment in a real indoor environment.

1. INTRODUCTION

Simultaneous localization and mapping (SLAM) is one of the fundamental problems in robotics. The problem is for a mobile robot, while moving around in some unknown environment, to use its sensors to construct a map of that unknown environment. SLAM is difficult mainly because the robot cannot determine its position precisely. It might have access to some positioning sensors such as wheel encoders, GPS, or a compass, but still, some kind of environmental feedback will always be necessary to help correct the error that inevitably exists in these sensor readings. The main sensors used in SLAM for this kind of feedback are laser range finders and video cameras.

We are interested in SLAM for constructing metric maps of large scale environments such as office buildings and mine fields. Laser range finders have been particularly successful sensors for these kinds of environments — see, for example, [14] — because lasers are extremely accurate. On the other hand, they are also heavy, expensive, and slow. In our work, we focus on the use of cameras as sensors due to their high speed, small size, and low cost.

Vision-based SLAM is an actively developing research area, but thus far most of the existing systems construct either occupancy grids or topological maps (see [5] for a survey), and these approaches are inappropriate for large scale metric mapping. For large scale metric maps, the simplest approach is to represent the world with a sparse collection of *landmarks*. These landmarks could be distinctive-looking 3D points or more complex objects such as lines, curves, corners, and so on. There have been several SLAM systems based on visual landmarks that work in small environments [1, 3, 4, 8, 15, 16], but thus far, there has been no successful robust, large-scale demonstration of vision-based SLAM.

Towards the goal of achieving large-scale metric vision-based SLAM, there has been some recent work on applying the efficient Rao-Blackwellised particle filter (RBPF) [9] as the underlying estimation algorithm and a stereo vision head as the sensor [2,12]. Both of these systems use Thrun et al.'s FastSLAM algorithm [14] for the RBPF to create sparse landmark maps organized by k-D trees for efficient search and modification.

We are particularly interested in combining multiple information sources, for example, line segments and distinctive points, to achieve robust large-scale vision-based SLAM at minimal cost. For point features, however, SIFT is computationally expensive; it requires construction of a scale space representation of each image, multiple convolutions, and extraction of a rich descriptor of the local image statistics around each point of interest. Combined with the computational complexity of maintaining many robot path estimates in the RBPF, systems based on SIFT and FastSLAM are going to be expensive or slow for several years to come.

In this paper, we explore the use of the KLT interest point detector [11] with trinocular stereo vision and the FastSLAM algorithm. On the one hand, KLT feature locations can be sensitive to noise, but on the other hand, they are quite lightweight in comparison with SIFT. We find that with the help of rather strict epipolar line constraints on the images obtained by a trinocular camera system, it is possible to choose only reliable points from a set of KLT points in an image set and use them to reconstruct 3D geometric point landmarks in an environment. We ran the FastSLAM algorithm with the 3D landmark point observations and verified the consistency of the result by comparing the performances with different number of particles used in the particle filter.

2. KLT-BASED FASTSLAM

Here we describe the application of FastSLAM [14] to the problem of vision-based SLAM with KLT point features as observations.

2.1. The FastSLAM algorithm

FastSLAM [14] is an elegant solution to the SLAM problem that maintains a full posterior over possible robot paths (as opposed to a maximum a posteriori estimate) using the RBPF [9]. The posterior distribution over possible robot paths is represented by a set of samples or particles, where each particle at time t represents one possible robot path up to time t, one possible series of data association assumptions for the sensor measurements up to time t, and a stochastic landmark map [13] based on those assumptions. Since each particle represents a particular robot path and a particular series of data association decisions up to time t, the observed landmarks are conditionally independent, so the posterior over landmark positions can be represented simply as a list of landmark estimates with associated uncertainties. The assumption of a particular robot path and particular series of data associations allows a representation of the map that is linear in the number of landmarks (the classical stochastic map is quadratic in the number of landmarks due to the correlations introduced by uncertain robot positions).

In this paper we adapt Thrun et al.'s "FastSLAM 1.0" [14] algorithm to the vision-based SLAM problem. At each time t, we seek a recursive estimate of

$$p(s_{0:t}, \Theta_t \mid u_{1:t}, z_{1:t}) \tag{1}$$

where $s_{0:t}$ is the robot's path from time 0 to time t, Θ_t is a map containing a set of landmarks, $u_{1:t}$ is a set of robot actions, and $z_{1:t}$ is a set of sensor observations. Each element s_i of $s_{0:t}$ is a vector describing the robot's pose at time i; we use a six degree of freedom representation for s_i.

The idea of the RBPF is to represent the posterior (Eq. 1) with a discrete set of M_t samples or particles

$$\left\{\left\langle s_{0:t}^{[m]}, \Theta_t^{[m]}\right\rangle, \text{where each index } m \in 1, \ldots, M_t\right\}. \tag{2}$$

$s_{0:t}^{[m]}$ is a specific robot path associated with particle m, and $\Theta_t^{[m]}$ is the stochastic landmark map associated with particle m, derived from $s_{0:t}^{[m]}$, the robot actions $u_{1:t}$, and the observations $z_{1:t}$.

FastSLAM 1.0 uses sequential importance resampling, also known in the computer vision literature as the "condensation" algorithm [7]. At each time t, for each particle m, we sample from the *proposal distribution*

$$p(s_t \mid s_{t-1}^{[m]}, u_t)$$

to obtain a temporary set of particles for time t. Then, for each temporary particle m, we compute the *importance weight*

$$w^{[m]} \propto p(z_t \mid s_t^{[m]}, \Theta_{t-1}^{[m]})$$

and update the particle's map with z_t assuming $s_t^{[m]}$ to get $\Theta_t^{[m]}$. The importance weights are normalized to sum to 1, then we sample M_t particles, with replacement, from the temporary particle set according to the normalized weights. The result is a new set of particles (Eq. 2) that represents the posterior (Eq. 1) at time t.

Our approach is identical to planar FastSLAM 1.0 [14] except that we use a six degree of freedom motion model $p(s_t \mid s_{t-1}, u_t)$ and a 3D point sensor model $p(z_t \mid s_t, \Theta_{t-1})$ using landmarks derived from KLT features on a trinocular stereo vision rig. We now describe the trinocular stereo sensor in detail.

2.2. Trinocular KLT as a sensor model for FastSLAM

In FastSLAM, the sensor model is fully described by the conditional probability $p(z_t \mid s_t, \Theta_{t-1}, n_t)$, explicitly conditioning on n_t, the set of correspondences between observations z_t and landmarks stored in Θ_{t-1}. The distribution is assumed to be a deterministic measurement function $f(\Theta_{t-1}, s_t, n_t)$ corrupted by Gaussian noise.

In our case, the observations are sets of 3D points in robot-relative coordinates, estimated by triangulation with a trinocular stereo vision rig. Our 3D point extraction procedure begins by obtaining 2D KLT (Kanade-Lucas-Tomasi) corner features [11] from each of three calibrated images simultaneously captured by the trinocular camera rig. We then find sets of corresponding features across the three images and triangulate to obtain an estimate of the putative feature's 3D position relative to the robot.

The basic idea of using KLT as a 2D feature detector is to find points with a complex local gradient field. Complexity of the gradient field is measured by the smaller eigenvalue of the matrix

$$Z = \begin{pmatrix} g_x^2 & g_{xy} \\ g_{xy} & g_y^2 \end{pmatrix}$$

in which the quantities are integrals of the squared gradient (in the case of g_x^2 and g_y^2) or the integral of the product of x and y gradients (g_{xy}) in a neighborhood around the point of interest. A point is selected as a KLT feature if the smaller eigenvalue λ_2 of Z is a local maximum and above some threshold λ. The motivation is that image points meeting the criterion have a local gradient structure that cannot be described by a single eigenvector (as would be the case for a simple edge), but have a more complex corner-like structure that should be easy to detect under various imaging conditions.

After extracting a set of KLT feature points from each of the three images acquired at time t, we attempt to find triples of corresponding points as a necessary step prior to triangulation. For each KLT point $p_{1,i}$ detected in image 1, we search image 2 for potentially corresponding points. For each point $p_{2,j}$ in image 2 close enough to the epipolar line corresponding to $p_{1,i}$, we triangulate using the calibrated intrinsic and extrinsic parameters of the camera rig to predict the putative object feature's appearance in image 3. If a suitable KLT point $p_{3,k}$ exists in image 3, we consider the triple

$(p_{1,i}, p_{2,j}, p_{3,k})$ a candidate match and continue searching for other possible matches for $p_{1,i}$. If no consistent triples or more than one consistent triple is found for $p_{1,i}$, we throw it out. On the completion of this simple correspondence algorithm, we have a set of corresponding triples of 2D points that can then used for 3D estimation. Typically we begin with about 200 KLT points in each image and end up with about 20 corresponding triples.

The last step of obtaining a sensor measurement is to estimate a 3D landmark in robot-relative coordinates given each triple of corresponding 2D KLT points. For each correspondence (p_1, p_2, p_3), we obtain an initial estimate of the 3D position P by triangulating from p_1 and p_2, then we use the Levenburg-Marquardt nonlinear least squares optimization algorithm [10] to find the 3D position P maximizing the likelihood of the 2D observations (p_1, p_2, p_3) assuming spherical Gaussian error in the measured image coordinates. We also obtain an estimate of confidence in the 3D point landmark position P by propagating the assumed measurement error through the maximum likelihood estimation procedure using the standard first-order approximation [6].

After 2D feature detection, correspondence estimation, and triangulation, we obtain a set of 3D point landmark observations with associated error covariance matrices. The set of landmarks with covariances makes up z_t, the robot's observation at time t, which is input to FastSLAM. From this point on, our system is identical to Thrun et al.'s FastSLAM 1.0 algorithm [14].

3. EXPERIMENTAL METHODS

To test KLT-based FastSLAM, we performed an experiment in the Image and Vision Computing Laboratory at SIIT. The room is a typical laboratory with desks, bookshelves and computers. Figure 1 shows an image set captured in the lab with the 10cm-baseline trinocular camera rig that was used in the experiment.

In this experiment, rather than mounting the rig on a robot, we simulated robot motions by manually moving a camera tripod. The simulated robot's position s_t in world coordinates at time t is defined as a vector with six degrees of freedom $s_t = (x, y, z, \phi, \theta, \psi)^T$. Here the x and y axes span a plane parallel to the floor of the lab, and z is the vertical distance of the reference camera's origin from the ground plane. The remaining three variables represent the robot's orientation. ϕ, θ and ψ stand for pitch, roll and yaw of the camera rig, respectively. During the experiment, due to the flat floor, z, pitch, and roll was always equal to zero throughout the experiment.

The camera rig cannot move itself, so in the experiment we roughly pushed or rotated the rig by hand from its original position to the next destination position in order to emulate a real robot move. Since each move of the rig is not perfect, the rig normally reaches a position slightly away (in terms of x, y and yaw, we do not measure z, pitch and roll since they are assumed to be zero in the experiment) from its destination position. So we treated the difference of the original position and the desired destination position as robot odometry, and the difference of the original position and the actually reached position as a true move. To make the experiment simpler, we composed camera rig odometry so that each odometric move involves only translation or only rotation. More specifically, odometry is of the form $(x, y, 0, 0, 0, 0)^T$ for translation, and $(0, 0, 0, 0, 0, \psi)^T$ for rotation.

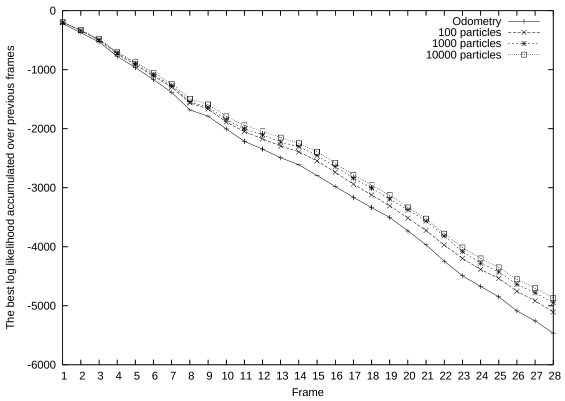


Fig. 2. Log likelihood of observation sequence given the model.

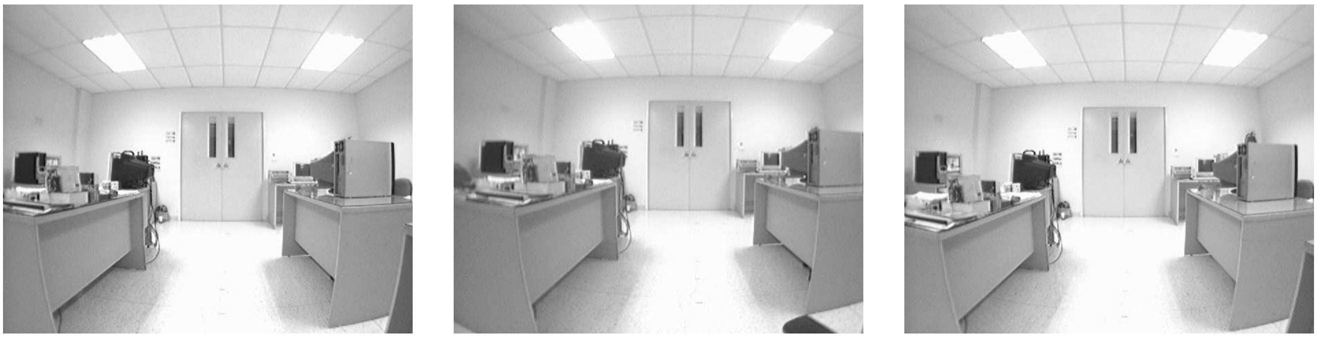
The actual path of the camera rig consisted of 29 positions $D_0, D_1, \ldots, D_{28}$ marked on the floor of the lab. At first the rig was positioned at D_0, which we defined to be the origin of the world coordinate system. The rig was then moved to each destination. Along the way, at each position, we measured the true position $T_1, T_2, \ldots, T_{28}$ of the rig and captured a trinocular image set. The simulated odometry measurements $O_1, O_2, \ldots, O_{28}$ were computed as $O_i = D_i - T_{i-1}$.

In this indoor experiment the robot's path was approximately composed of a 4 meter forward translation from O_1 to O_{10} (roughly 0.4 meters per move), a 180 degree rotation from O_{11} to O_{18} (roughly 22.5 degrees per move), and finally a 4 meter forward translation from O_{19} to O_{28} (roughly 0.4 meters per move).

Image sets (29 frames including the initial state) and odometry (28 six dimensional vectors) were collected in the lab. They were used as the input for KLT-Based FastSLAM to estimate the path of the camera rig and generate a 3D metric map of the lab. We ran the algorithm with 100, 1000 and 10000 particles. In order to compare the algorithm's performance against a baseline, we also ran the same mapping algorithm purely using odometry as the estimate of the camera position.

4. RESULTS

Log likelihood is a measure of accuracy of the current landmark observation given the previous observation. It is given



(a) (b) (c)

Fig. 1. Trinocular image set captured in the lab. (a) Reference image. (b) Horizontally aligned image. (c) Vertically aligned image.

by

$$\begin{aligned} & \mathit{ln}\left(p(z_t \mid s_t^{[m]}, \Theta_{t-1}^{[m]})\right) \\ \sim \quad & -\frac{1}{2}\mathit{ln}\left|2\pi Q_t^{[m]}\right| - \frac{1}{2}(z_t - \hat{z}_t^{[m]})^T Q_t^{[m]-1}(z_t - \hat{z}_t^{[m]}) \end{aligned}$$

with the covariance

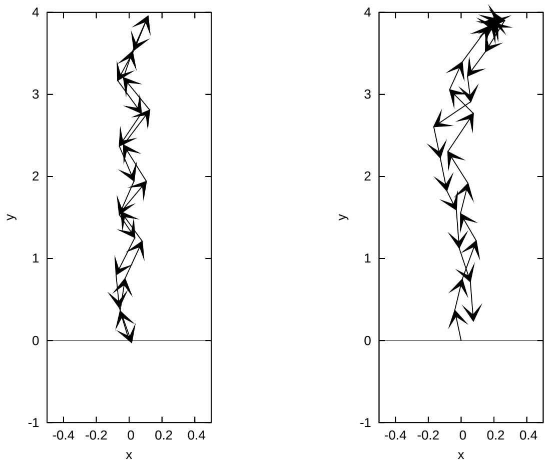
$$Q_t^{[m]} = G_t^{[m]T}\Sigma_{t-1}^{[m]}G_t^{[m]} + R_t$$

, where $\hat{z}_t$ is an estimation of the new observation z_t, Σ_{t-1} is the covariance of the landmark before the new observation is made, G_t is the Jacobian of the sensor model with respect to the landmark, and R_t is the covariance of the Gaussian noise of the new observation [14].

For each particle of each sequence of observation, we calculated the accumulated log likelihood, which is an addition of log likelihood over all the past sequences. It tells the degree of consistency of the map recorded in a particle. For each sequence of observation, we chose the particle that has the best (largest) value of accumulated log likelihood. The result is shown in Figure 2. As the number of the particle used in the FastSLAM algorithm increases, the accumulated log likelihood becomes better. The result tells that the particle filter is working properly in the experiment, i.e. with more particles, the better localization of the camera rig and estimate of landmark positions for each observation sequence is achieved.

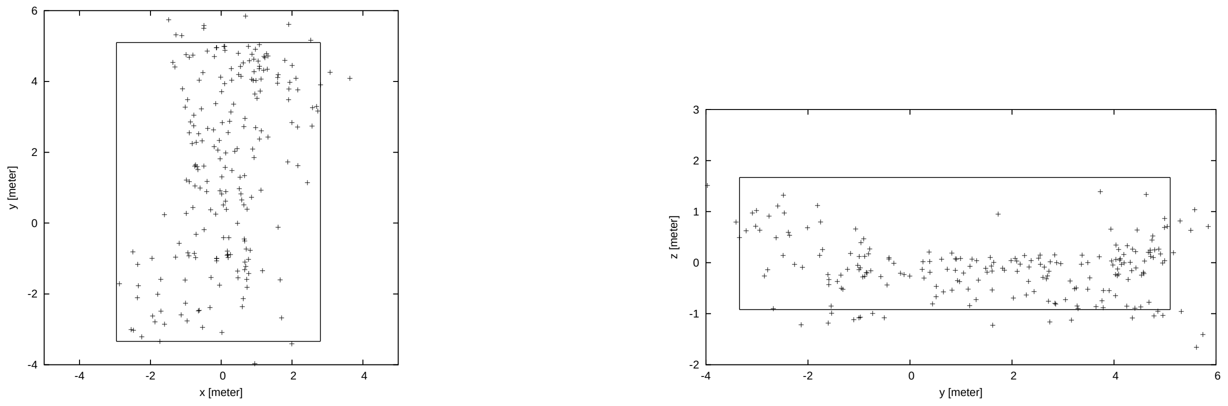
Figure 3 is 2D projections of the generated 3D map of the lab using 1000 particles. Only KLT point landmarks that were observed more than twice over all the observation sequences are plotted since landmarks observed only once tend to be noisy observations. Point landmarks in the map captured the actual distribution of edges and corners of objects seen in the lab.

In Figure 4, estimated path of the camera rig is shown.



(a) (b)

Fig. 4. Path of the camera rig projected onto the $x - y$ plane. Each move is represented as a vector. The rig was put at $(x, y) = (0, 0)$ initially and was moved 28 times while taking an image set after each move. (a) True path of the camera rig. (b) Estimated path of the camera rig using 1000 particles.



(a) (b)

Fig. 3. Projection of the 3D metric map into 2D planes. The boundary of the lab is shown as a rectangle in the figures. (a) KLT point landmarks projected into $x - y$ plane, the top view of landmarks. (b) KLT point landmarks projected into $y - z$ plane, the side view of landmarks.

5. CONCLUSION

In this paper, we have demonstrated the feasibility of KLT-based FastSLAM on a data set collected in a real indoor environment. We confirmed the positive effect of increasing the number of particles by looking at the accumulated log likelihood of particles per each sequence of observation. The distribution of 3D KLT point landmarks in the generated map globally represented the real distribution of edge and corner points of objects seen the lab.

However, there are noticeably many noisy cluttering landmark points in the 3D map which will hamper the navigation task based on the map. This happened mainly due to the fact that the calibration of the cameras on the rig was not ideally done in the time of the experiment. Poorly calibrated camera parameters give noisy estimation of the 3D landmark position derived from the 2D pixel coordinates of the landmark in each image of an image set.

The estimated path did not show any significant improvement against the path based on odometry nor on true measurement. One possible reason is that the odometry used in the experiment was so close to the truth that it was beyond the capability of the estimation algorithm to get the better estimate of the path. To verify it, we need more experiments with varying degrees of odometry error against the true measurement.

In the future work, we plan to analyze to what extent the calibration of the camera rig is affecting the resultant accuracy of estimation of landmark positions. We also seek to see the better estimation of camera rig positions over odometry by testing with various degrees of erroneous odometry. Finally, we plan a direct comparison of KLT and SIFT features as landmarks in SLAM.

6. ACKNOWLEDGMENTS

We thank Nico Hochgeschwender for help with software development. This research was supported by Thailand Research Fund grant MRG4780209 to MND.

7. REFERENCES

[1] N. Ayache and 0.D. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, 1989.

[2] M. Dailey and M. Parnichkun. Simultaneous localization and mapping with stereo vision. In *Proceedings of the IEEE International Conference on Automation, Robotics, and Computer Vision (ICARCV)*, 2006. To appear.

[3] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1403–1410, 2003.

[4] A. Davison, Y. Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.

[5] G.N. DeSouza and A.C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. University Press, Cambridge, UK, 2000.

[7] M. Isard and A. Blake. Condensation — conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[8] D.J. Kriegman, F. Triendl, and T.O. Binford. Stereo vision and navigation in buildings for mobile robots. *IEEE Transactions on Robotics and Automation*, 5(6):792–803, 1989.

[9] K. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems (NIPS)*, 1999.

[10] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 1988.

[11] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Patter Recognition (CVPR '94)*, pages 593–600, 1994.

[12] R. Sim, P. Elinas, M. Griffin, and J.J. Little. Vision-based SLAM using the Rao-Blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR)*, 2005.

[13] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer Verlag, 1990.

[14] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 2004. To appear.

[15] Y. Yagi, Y. Nishizawa, and M. Yachida. Map-based navigation for a mobile robot with omnidirectional image sensor copis. *IEEE Transactions on Robotics and Automation*, 11(5):634–648, 1995.

[16] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis*. Springer-Verlag, 1992.