AN ITERATED EXTENDED KALMAN FILTER FOR 3D MAPPING VIA KINECT CAMERA

Li Ling, Eva Cheng, and Ian S. Burnett

School of Electrical and Computer Engineering
RMIT University, Melbourne, Australia
{li.ling, eva.cheng, ian.burnett}@rmit.edu.au

ABSTRACT

This paper proposes the use of the Iterated Extended Kalman Filter (IEKF) in a real-time 3D mapping framework applied to Microsoft Kinect RGB-D data. Standard EKF techniques typically used for 3D mapping are susceptible to errors introduced during the state prediction linearization and measurement prediction. When models are highly nonlinear due to measurement errors e.g., outliers, occlusions and feature initialization errors, the errors propagate and directly result in divergence and estimation inconsistencies. To prevent linearized error propagation, this paper proposes repetitive linearization of the nonlinear measurement model to provide a running estimate of camera motion. The effects of iterated-EKF are experimentally simulated with synthetic map and landmark data on a range and bearing camera model. It was shown that the IEKF measurement update outperforms the EKF update when the state causes nonlinearities in the measurement function. In the real indoor environment 3D mapping experiment, more robust convergence behavior for the IEKF was demonstrated, whilst the EKF updates failed to converge.

Index Terms— Iterated Extended Kalman Filter, 3D Reconstruction, Kinect

1. INTRODUCTION

Over the last few decades, the Extended Kalman Filter (EKF) algorithm for real-time camera motion estimating is one of the key technologies for Simultaneous Localization and Mapping (SLAM) in robotic automation and computer vision [1~6]. In the EKF-SLAM process, the performance and convergence of the estimates are mainly influenced by two error effects. First, the feature initialization uncertainty is a problem critical to the EKF that precipitates immediate and substantial estimation inconsistency. Second, the EKF linearization of the non-linear state and measurement model has the potential to introduce errors that will be propagated during the evolution.

MonoSLAM [1] presents promising results by using a standard extended Kalman filter to perform real-time motion and structure estimation from a single camera. As the depth is unknown, MonoSLAM evaluates the depth of an observed feature using a particle filter; however, this approach causes a delay of a few frames. Civera et al. [2] and Montiel et al. [3] proposed an inverse depth parametrization for single camera SLAM that permits efficient representation of Gaussian linearity of 3D feature points during initialization and within the EKF. Assuming that the image measurement process is nearly linear, inverse depth parametrization can process features from nearby to infinity.

Recent RGB-D cameras, such as the Microsoft Kinect, provide synchronized color and per-pixel depth information in real-time. By using a depth sensor, the Kinect avoids the complexity of robust visual correspondence computation for depth estimation from stereo matching. The depth sensor consists of one IR projector and one IR camera, and the relative geometry between the IR image and the projector pattern can be easily measured. The depth data output by the Kinect for each frame is the 'true' 3D information that addresses the long-standing problem of real time feature initialization in EKF processing for camera motion and 3D structure estimation.

In the Kinect SLAM 3D mapping approaches, Huang et al. [4] proposed an autonomous flight system for visual odometry and mapping using the EKF-SLAM on Kinect cameras. Henry et al. [5] proposed RGB-D mapping framework that built dense 3D maps of indoor environments through TORO, an optimization tool developed for SLAM. Hervier et al. [6] built an accurate 3D map based on the estimated covariance of the Iterative Closest Point (ICP) and the data fusion by EKF with a Kinect sensor and a three-axis gyroscope. In the approaches of [4-6], the EKF linearizes the measurement prediction and all unknown transformations in the state prediction using series expansion, substituting Jacobian matrices for linear transformations in the Kalman filter. However, such linearizations assume that the prediction errors can be well approximated by a linear function. If this condition cannot be satisfied, the errors will propagate and directly result in divergence of the estimations.

To overcome the disadvantages of existing EKF-SLAM mapping systems, this paper proposes the use of an IEKF-based SLAM algorithm to reconstruct 3D scene geometry in real-time using the RGB-D data from a Kinect camera. The proposed approach re-linearizes the measurement equation by iterating an approximate maximum a posteriori (MAP) estimate around the updated state, rather than relying on the predicted state. This paper follows the standard Kalman Filter processing steps including initialization, prediction and update. In addition, the paper proposes the management of feature points to control the number of points in the map by dynamically adding visible features with reliable 3D information or removing the occluded features during the evolution. To alleviate feature outlier errors, the robust estimation algorithm RANSAC [8] is employed to remove the effect of mismatched points. Efficient rendering of complex geometric objects is then performed using surfel (surface element) [9] representations of the depth data. To evaluate the proposed IEKF approach, a simulation based on a synthetic map and landmark data [7] compares the performance and consistency of the standard EKF and proposed IEKF algorithms under variant image measurement noise. The IEKF is then applied to the 3D mapping of a real indoor environment to compare the convergence

behavior of the IEKF compared to the EKF in realistic environmental conditions.

In the following, the proposed IEKF approach is presented in Section 2. Section 3 then presents and discusses the simulation and real 3D space mapping experiments and results, whilst Section 4 concludes the paper.

2. METHOD

2.1 State Vector Definition

The general problem of parameter estimation from discrete nonlinear measurements can be described as:

$$s_k = f_k s_{k-1} + w_{k-1} \quad (1)$$

$$z_k = h(s_k) + v_k \quad (2)$$

where the dynamic model in Eq.(1) describes how the state vector s evolves in each time step k by the state transition function f and the process noise w. In Eq. (2), the measurements z are expressed as a function h of the unknown state s, plus measurement noise v. The process and image measurement noise are both assumed to be zero mean multivariate Gaussian noise with covariance matrices Q and B. The state vector is accompanied by a single covariance matrix L and in this paper, state vector $\hat{s}$ is composed of the camera motion information and the 3D point locations:

$$\hat{s} = \left(\hat{s}_v \; \hat{M}_1 \; \hat{M}_2 \cdots\right)^T \quad (3)$$

The dynamic model applied to the camera motion information is:

$$\hat{s}_v = \left(t \; \theta \; \dot{t} \; \dot{\theta}\right)^T \quad (4)$$

where the camera's state vector $\hat{s}_v$ comprises a metric 3D translation vector t, camera rotation/orientation represented by a 3D Euler angle representation $\theta = [\theta_x \; \theta_y \; \theta_z]^T$, translational velocity vector $\dot{t}$ and rotational velocity vector $\dot{\theta}$. This paper uses the 'hat' in $\hat{s}_v$ to indicate an estimate of s_v. The 3D feature points locations are stored as: $\hat{M}_i = (X_i \; Y_i \; Z_i)^T$.

2.2 Feature Initialization

The feature points from images are detect use the Scale Invariant Feature Transform (SIFT) [10], where this paper utilizes the SIFTGPU package [11]. The points with valid Kinect point depth are selected as key points; in particular, the visibilities of the feature points are constrained by the back-projections of the corresponding 3D points within the range of the image. The feature points on the two consecutive images are matched with the Zero-mean Normalized Cross-Correlation (ZNCC) algorithm. Typically, these matched feature points contain a fair number of outliers. In this paper, the robust estimation algorithm RANSAC [8] is employed to remove the effect of mismatched points (outliers). The classification is done by employing a cost function together with a threshold which depends on the expected measurement noise. This threshold is directly correlated with the number of feature points. To ensure real-time processing without trading off on accuracy, the number of inliers N is bounded around 50 for each time step in practice.

The 3D feature point in a world coordinate system is represented by the homogeneous vector $M=[X,Y,Z,1]^T$, and the feature point in an image is represented by $m=[x,y,1]^T$. Camera and image coordinates are related by the perspective projection equations:

$$\frac{x - x_0}{f_x} = \frac{x_C}{d}; \frac{y - y_0}{f_y} = \frac{y_C}{d}; \quad (5)$$

where f_x and f_y are the distances from the centre of projection to the image plane, $[x_0,y_0]^T$ is the coordinate of the camera centre, d is the depth of the image point $m=[x,y,1]^T$ and $m_C = [x_C, y_C, 1]^T$ is the camera coordinate of m. Thus, 3D point M can be estimated from:

$$M = R^{-1} m_C + t \quad (6)$$

where R is the rotation matrix representation of the Euler angle rotation $\theta = [\theta_x \; \theta_y \; \theta_z]^T$, and t is the translation vector; both t and R are estimated by the proposed IEKF approach for each time step. Upon system initialization, the first image centre is assumed as the origin point, with $\theta = [0\ 0\ 0]^T$ and $t=[0\ 0\ 0]^T$. The initial 3D feature points can then be obtained from Eq. (6).

2.3 Feature Points Management

In this paper, feature points can be dynamically added to the map when new landmarks are required (if $N<50$), and can also be deleted if features are invisible. For example, using three images to illustrate the proposed feature management: in the first two images, a process matching the SIFT feature points sets up a matched flag for each matched point (x_{i1} and x_{i2}). For each new time step (adding a new image) in the system, the second/last image and the third/new image feature points are matched first. Then, the matched points are checked in the second/last image, if the match flag for one certain point is already true, that means this point x_{i3} in the third image is in the same track of x_{i1} and x_{i2}, corresponding to the same 3D point X_i. Thus, the 3D feature point and corresponding covariance matrix in the last time step are assumed as initial values of the point in the third image. A feature is deleted (3D feature point and the covariance matrix are deleted) from the system if a match point cannot be found in the new image.

New features are only added into the system if the number of features in the current time step is less than the threshold N. With a perspective camera, the position at which the feature is expected to be found in the image has the form:

$$h = (x,y)^T \quad (7)$$

The state covariance after initialization is:

$$P_{k|k}^{new} = G \begin{bmatrix} P_{k|k} & 0 \\ 0 & B \end{bmatrix} G^T \quad (8)$$

The state covariance is initialized as:

$$P_{k|k} = diag\left[0_{diag^3} \;\; 0_{diag^3} \;\; (\delta\dot{t})^2_{diag^3} (\delta\dot{\theta})^2_{diag^3}\right] \quad (9)$$

where $diag^3$ means the 3×3 diagonal matrix, $\delta\dot{t}$ and $\delta\dot{\theta}$ are standard errors of $\dot{t}$ and $\dot{\theta}$ which are predefined. G can be calculated from:

$$G = \begin{bmatrix} I & \cdots & 0 \\ \dfrac{\partial M}{\partial t} & \dfrac{\partial M}{\partial \theta} & 0 \cdots 0 \;\; \dfrac{\partial M}{\partial h} \end{bmatrix} \quad (10)$$

In Eq. (10), the image measurement noise covariance B is taken to be diagonal with the magnitude determined by the measurement noise.

2.4 Prediction

In the prediction stage, the motion model follows the standard EKF [1]. The predicted state and covariance estimates have the form:

$$\hat{s}_{k|k-1} = f(\hat{s}_{k-1|k-1}) \quad (11)$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \quad (12)$$

The new state estimate is:

$$f(\hat{s}_{k-1|k-1}) = \begin{pmatrix} t_k \\ q(\theta_k) \\ \dot{t}_k \\ \dot{\theta}_k \end{pmatrix} = \begin{pmatrix} t_{k-1} + (\dot{t}_{k-1} + \delta\dot{t}\Delta T)\Delta T \\ q(\theta_{k-1}) \times q(\dot{\theta}_{k-1} + \delta\dot{\theta}\Delta T)\Delta T \\ \dot{t}_{k-1} + \delta\dot{t}\Delta T \\ \dot{\theta}_{k-1} + \delta\dot{\theta}\Delta T \end{pmatrix} \quad (13)$$

$$F_{k-1} = \frac{\partial f(\hat{s}_{k-1|k-1})}{\partial \hat{s}_{k-1|k-1}} \quad (14)$$

ΔT is time step, $q(\theta_k)$ means the quaternion representation of θ_k, and $\delta\dot{t}$ and $\delta\dot{\theta}$ are zero-mean Gaussian distributed noise coming from an impulse of acceleration and angular acceleration, respectively.

The process noise covariance Q has the form:

$$Q_{k-1} = \frac{\partial f_{k-1}}{\partial w} P_n \frac{\partial f_{k-1}}{\partial w}^T \quad (15)$$

where $w = (\delta\dot{t}\Delta T \;\; \delta\dot{\theta}\Delta T)^T$ is the process noise vector.

P_n is the covariance of noise vector w, and it has the form:

$$P_n = diag\left[(\delta\dot{t}\Delta T)^2 \;\; (\delta\dot{\theta}\Delta T)^2\right] \quad (16)$$

The image position of each feature is initialized by:

$$h_C = (R_k)(M - t_k) \quad (17)$$

Then the estimate of the feature on image has the form:

$$h = \begin{pmatrix} \hat{x} & \hat{y} \end{pmatrix}^T = \begin{pmatrix} x_0 - f\frac{h_{Cx}}{h_{Cz}} & y_0 - f\frac{h_{Cy}}{h_{Cz}} \end{pmatrix}^T \quad (18)$$

where the Jacobian of h is: $H = \frac{\partial h}{\partial \hat{s}}$.

2.5 Update

The IEKF update is based on the Gauss Newton algorithm. The update optimally minimizes the cost function of EKF in a second order Taylor series about the i-th iterated value of the estimate of s_k, denoted as s_k^i. The update of IEKF is:

$$\hat{s}_k^{i+1} = \hat{s}_k^i + P_{k|k}^i (H_k^i)^T (R_k^{noise})^{-1}(z_k - h_k^i) - P_{k|k}^i P_{k|k-1}^{-1}(\hat{s}_{k|k}^i - \hat{s}_{k|k-1}) \quad (19)$$

$$P_{k|k}^i = P_{k|k-1} - P_{k|k-1}(H_k^i)^T (H_k^i P_{k|k-1}(H_k^i)^T + R_k)]^{-1} H_k^i P_{k|k-1} \quad (20)$$

As a result, the IEKF repeatedly calculates an intermediate posterior state $\hat{s}_k^i$, where i is the iteration number. The IEKF starts from the prior state, where $\hat{s}_k^0 = \hat{s}_{k|k-1}$, $H_k^0 = H_k$, $P_{k|k}^0 = P_{k|k-1}$. At each iteration, the estimate and covariance matrix of the previous iteration are used as the new *a priori* information. When the consecutive values differ by less than a preselected threshold or after a certain number of iterations, the iterations are stopped. In this paper, trading off between estimation accuracy and computational cost, the maximum number of iterations is set as 30 in practice.

2.6 Surface Merging

Motivated by [9], surface merging based on surfel representation is performed using two operations: surfel update and surfel addition. A surfel is updated when the new surfel satisfies three conditions: (1) if the depth of the surfel is valid, the re-projection of this surfel is in the range of the current image; (2) if the normal angle between the new and old camera principal axes is less than the pre-defined maximum angle of 60º, the surfel is considered as visible in the new image. (3) if two different surfels correspond to the same object, comparing the depth value of existing and new surfels, the one closer to the camera is considered as visible. Otherwise, if condition (1) cannot be satisfied, this surfel is omitted. If conditions (2) or (3) cannot be satisfied, this surfel is removed from the surface queue. After all existing surfels have been updated, surfels are added in the regions where the new depth map is not covered by existing surfels.

3. EXPERIMENTS AND RESULTS

Two experiments are conducted to validate the improvement of consistency and robustness through repetitive linearization of the nonlinear observation model in EKF-SLAM algorithm. In the first experiment, a simulator with the synthetic map and landmarks is used to keep the system as simple as possible to demonstrate the influence of system noise on the estimation consistency. Then, a real room environment 3D mapping application presents that the IEKF can increase the robustness of convergence against error propagation.

3.1 Simulation

A range and bearing camera model [12] with synthetic map and landmark data was used to compare the consistency between EKF and IEKF under variant image noise. The camera's true trajectory is known as a circle of centre (0, 20)m and 20m radius. The landmarks that are visible in the semi-circular field of view of the camera are selected in each time step. The experimental parameters are set with $\delta\dot{t} = 1.0 m/s^2$, $\delta\dot{\theta} = 3.0 rad/s^2$, δv=*1.0* pixels and ΔT=0.1s where the measurement noise is assumed to be 3 pixels (δv=*3.0* pixels).

Assuming the z axis as zero, where the camera only moves on the x-y coordinate system, the camera state at time step k is simplified into the form:

$$s_k = [t_x \; t_y \; \theta_k \; \dot{t}_k \; \dot{\theta}_k \; x_1 \; y_1 \cdots x_N \; y_N]^T = \begin{bmatrix} s_{vk} \\ M_{1\ldots N} \end{bmatrix} \quad (21)$$

The Normalised Estimation Error Squared (NEES) method [7] is used to characterize the filter performance:

$$\varepsilon_k = (s_k - \hat{s}_{k|k})^T P_{k|k}^{-1}(s_k - \hat{s}_{k|k}) \quad (22)$$

where s_k is the 'true' state vector from the synthetic data and $\hat{s}_{k|k}$ is the estimated value. Each filter was run for two loops of the trajectory and each simulation was repeated 50 times. Figs. 1 and 2 show the NEES results of EKF and IEKF averaged over the 50 simulation repetitions, where x and y axes indicate time step and NEES error, respectively (the total number of time steps is k = 277). In Fig. 1, the NEES of EKF has a quick increase from k = [50,150], maintaining this high error level NEES until the simulation finishes. When k = [100, 150], the camera closes the first loop. In the second loop, the measurement error is accumulated, and EKF maintains this error into the second loop. In Fig. 2, the NEES of IEKF peaks when the camera closes the first loop then quickly converges; thus, it can be seen that the IEKF observations are more accurate than EKF since NEES mean value is 33.1 lower than the EKF.

In the second experiment, measurement noise of δv=*3.0* pixels is added, whilst all other parameters are kept unchanged. Figs. 3

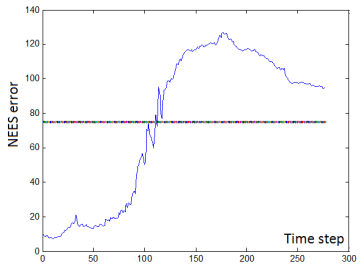

Fig. 1. The State NEES of EKF (δv=*1.0* pixels)

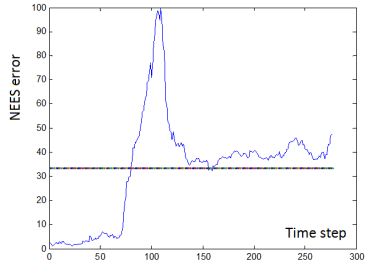

Fig. 2. The State NEES of IEKF (δv=*1.0* pixels)

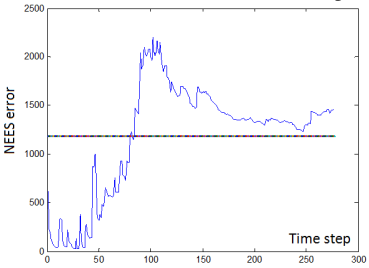

Fig. 3. The State NEES of EKF (δv=*3.0* pixels)

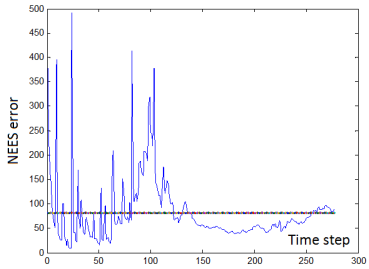

Fig.4. The State NEES of IEKF (δv=*3.0* pixels)

and 4 are the NEES of EKF and IEKF, respectively. In comparison to Figs. 1 and 2, the mean NEES values of both algorithms show an increase: the EKF increases from 78.87 to 1289.63, whilst the IEKF increases from 33.1 to 79.23. IEKF shows a better consistency against increased measurement noise compared to EKF.

3.2 3D Mapping

To evaluate the proposed IEKF approach, this paper performed 3D reconstruction of a real room environment using a Microsoft Kinect; the movement is estimated in real-time by EKF and IEKF respectively. The Kinect is fixed on a trolley whose wheels can only move back and forth. In the experiment, the trolley moves in a circle of radius 0.6m with the left wheel of the trolley along the

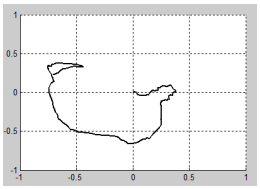
Fig. 5 EKF trajectory (m)

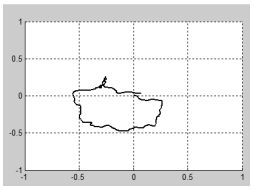
Fig.6 IEKF trajectory (m)

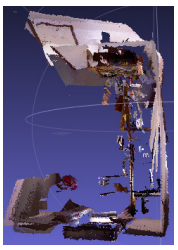
Fig.7 EKF Top View

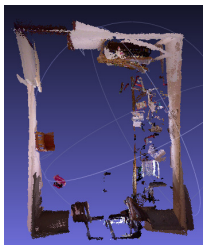
Fig.8 IEKF Top View

edge of circle. This paper uses a free moving camera model with constant velocity $\dot{t} = 0 m/s$ and constant angular velocity $\dot{\theta} = 0 rad/s$. The acceleration $\delta\ddot{t} = 0.007\, m/s^2$ and angular acceleration $\delta\ddot{\theta} = 0.001\, rad/s^2$ and the image noise is 1 pixel and $\Delta T = 1s$.

Figs. 5~8 show the Kinect trajectory estimate with an unknown scale and 3D mapping results of the EKF and IEKF. EKF assumes that the system noise is Gaussian, and represents the state uncertainty by approximate mean and variance. However, this approximation is not an adequate description, and the EKF linearization of the non-linear model cannot accurately match the true state following initialization, as shown in Fig.5. The errors are propagated with increasing time steps, which leads to a discontinuity failure in the left half circle. The camera motion measurement error thus directly impacts the 3D reconstruction result, as shown in the EKF result of Fig. 7. In Fig. 7, the 3D map shows reconstruction inconsistencies at the beginning and fails in the left side of the room when errors accumulated. In the IEKF results Fig. 6 and 8, the IEKF benefits from the re-linearization of the measurement model. The reconstruction in Fig. 6 is complete and the rectangle shape of the room in Fig. 8 is better than EKF result in Fig. 7. In the experiment practice, we find the eventual inconsistency of both algorithms is inevitable as time progresses; however, the proposed iterated version of the EKF shows increased convergence robustness against error propagation.

4. CONCLUSION

This paper proposed the use of an IEKF-based SLAM algorithm for real-time reconstruction of 3D scene geometry recorded with RGB-D data from a Kinect camera. Simulation results show improved estimation consistency and more accurate state estimation of IEKF over the EKF algorithm when the image measurement noise increases. Further, results obtained from 3D mapping experiments of a real indoor environment validate the improvement of state estimate consistency by comparing the proposed iterated EKF with the standard EKF technique, where more robust convergence behavior for the IEKF was demonstrated, whilst the EKF updates failed to converge.

5. REFERENCES

[1] A. J. Davison and I. D. Reid, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, p. 15, 2007.

[2] J. Civera and A. J.Davison, "Inverse depth parametrization fro monocular SLAM," *IEEE Trans. on Robotics,* vol. 24, p. 13, 2008.

[3] J. M. M. Montiel and J. Civera, "Unified Inverse Depth Parametrization for Monocular SLAM," in *In Proceedings of Robotics: Science and Systems*, 2006.

[4] A. S.Huang and A. Bachrach, "Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera," presented at the Int. Symposium on Robotics Research, Flagstaff,Arizona,USA, 2011.

[5] P. Henry and M. Krainin, "RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments," in *Proc. of International Symposium on Experimental Robotics*, 2010.

[6] T. Hervier and Silv`ere Bonnabel, "Accurate 3D maps from depth images and motion sensors via nonlinear Kalman filtering," presented at the IEEE International Conference on Intelligent Robots and Systems 2012.

[7] T. Bailey, "Consistency of the EKF-SLAM Algorithm " presented at the international Conference on Intelligent Robots and Systems, , 2006.

[8] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. of the ACM,* vol. 24, 1981.

[9] T. Weise, "In-hand Scanning with Online Loop Closure," presented at the international conference on Computer Vision Workshps, 2009.

[10] D. G. Lowe, "Object Recognition from Local Scale Invariant Features " presented at the Proceedings of the International Conference on Computer Vision, 1999.

[11] W. changchang, *http://cs.unc.edu/~ccwu/siftgpu/*.

[12] S. J.Julier, "A Counter Example to the Theory of Simultaneous Localization and Map Building," presented at the A Counter Example to the Theory of Simultaneous Localization and Map Building, Seoul, Korea, 2001.