Evaluating an iSAM-based SLAM System

Thuy Tu Ha
Department of Computer Science and Engineering
Ho Chi Minh City University of Technology
Ho Chi Minh City, Vietnam

Ngoc Minh Le
Department of Computer Science and Engineering
Ho Chi Minh City University of Technology
Ho Chi Minh City, Vietnam
minhle@cse.hcmut.edu.vn

***Abstract*—Many systems have been developed to solve the Simultaneous Localization and Mapping (SLAM) problem in practical applications. Particular systems employ different kinds of sensors; they produce localization as the main output or compute both localization and map (full SLAM). Most systems use filtering or smoothing approaches to update the measurements collected. Incremental Smoothing and Mapping (iSAM) is a method that employs the smoothing approach to solve the full SLAM problem. In this paper, we present an evaluation system for an iSAM-based SLAM system.**

Keywords-SLAM; iSAM; performance

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is the problem of building a map of a previously unknown environment and estimating robot locations at the same time. SLAM has been considered to be solved at a theoretical level using e.g. probabilistic methods [2]. However, substantial technical issues remain in practical systems. A practical system is typically constructed using a front-end and a back-end block. The function of the front-end block is to estimate spatial relations between robot poses as well as spatial relations among robot and landmarks with a certain amount of uncertainties. The function of the back-end block is to optimize these relations to generate new spatial relations with minimal uncertainties. These two blocks run iteratively one after another to build up the robot trajectory and the map.

Davison has developed monoSLAM, a monocular SLAM system that can run in real-time with several assumptions on the camera motion models [5,6]. In the front-end, assumptions on velocities of the camera motion are used to estimate spatial relations between camera poses; to estimate image feature positions, the "direct search approach" described in [9] is used. In the back-end, the extended Kalman filter (EKF) is used to optimize the poses of the camera. This combination constitutes the success of the monoSLAM system.

Kaess *et al.* have proposed incremental Smoothing and Mapping (iSAM) [4], a real-time variant of Square Root SAM [3]. iSAM can be used as a back-end in landmark-based SLAM or in pose-only systems. While monoSLAM concentrates on localization as the main output, SAM and iSAM give a full solution to the SLAM problem of determining both robot trajectory and map. iSAM has been employed as a core engine in some visual odometry and visual SLAM systems [8, 3]. High quality data collected with a laser rangefinder or a stereo camera have been used to experimentally evaluate these systems; results have shown their performance in accuracy and in computing speed, where the latter has been gained by comparing the output trajectories with those computed by other approaches. However, in his PhD thesis [9], Fakih has raised a question about the performance of iSAM: "It is not clear how this method would perform in the case of the highly non-linear and noisy visual measurements case." In this paper we evaluate an iSAM-based SLAM system using a variety of noisy synthetic data sets. The output trajectories determined by the system are compared to the ground truth trajectory to measure the system performance in each case.

Our system has been designed following the motion model and the observation model as described in the SLAM paper [2]. In the front-end, the assumptions of constant translational and angular velocities of the camera motion as in [5] are used to estimate spatial relations between consecutive camera poses. In addition, a highly non-linear visual measurement model based on motion field is established to estimate 3D positions of landmarks relative to the camera poses. In the back-end, iSAM module [10] is employed to smooth both the camera trajectory and the map. Additionally, constant covariance matrices are used for the measurement model to simplify the uncertainty model of 3D landmark measurements. The system can be extended with an independent component for visual feature detection and matching under the condition that this additional component provides a stream of 2D velocity pairs for each feature point (correspondences) in consecutive image frames. There are several ways to implement this component. Harris corner algorithm is a popular algorithm to detect image features. Scale-invariant feature transform (SIFT) and Speeded Up Robust Features (SURF) are robust techniques to match features between image frames. In this paper, we use a simple simulation of this component to generate noisy synthetic data to evaluate the system, refer to Figure 1.

The rest of this paper is organized as follows. Section II gives an introduction to our system. Section III describes the experimental results. Section IV discusses these experimental results. We conclude in Section V.

II. SYSTEM

Our system consists of three main components, depicted in Figure 1: Motion Prediction, Measurement Estimation, and iSAM-based Update. The Motion Prediction component implements the camera motion model that predicts the relative 3D translation and rotation of the camera with respect to the previous camera pose and feeds it into the Update component. The Measurement Estimation component implements the observation model that estimates the 3D landmark positions relative to the predicted camera pose and feeds them into the Update component. The iSAM-based Update component uses the iSAM approach to smooth the camera trajectory and the map. The updated pose of the camera and landmark positions will then be used by both Motion Prediction and Measurement Estimation components in the next step.

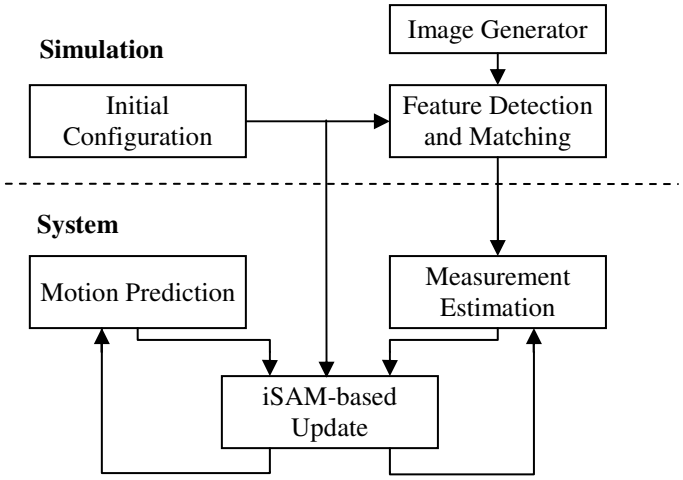

Figure 1. Block diagram of the system and simulation module.

A. Motion model

A pose can be specified by a translation vector $(x, y, z)^T$ and Euler angles $(\psi, \theta, \varphi)^T$. We denote by $pose^W$ the camera pose with respect to the world coordinates. A pose $pose^W = (x, y, z, \psi, \theta, \varphi)^T$ can be expressed as a homogeneous matrix of a translation vector and a rotation matrix,

$$M = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}, R = R(\psi)R(\theta)R(\varphi), T = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

where $R(\psi)$, $R(\theta)$, and $R(\varphi)$ are the matrices of rotation around the *Z*-, *Y*-, and *X*-axes, respectively.

Given a matrix M_1 specifying the current pose $pose_1^W = (x_1, y_1, z_1, \psi_1, \theta_1, \varphi_1)^T$ and a matrix M specifying a new camera pose $d^{pose_1} = (x, y, z, \psi, \theta, \varphi)^T$ relative to $pose_1^W$, then the matrix M_2 specifying the new camera pose with respect to the world frame is the product of M_1 and M.

$$M_2 = M_1 M = \begin{bmatrix} R_1 & T_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Hence, the parameters of the new pose $pose_2^W = (x_2, y_2, z_2, \psi_2, \theta_2, \varphi_2)^T$ can be written as, see [1]:

$$f = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ \psi_2 \\ \theta_2 \\ \varphi_2 \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \\ \arctan(r_{21}, r_{11}) \\ \arctan(-r_{31}, r_{11}c + r_{21}s) \\ \arctan(r_{13}s - r_{23}c, -r_{12}s + r_{22}c) \end{bmatrix} \quad (1)$$

where $c = \cos(\psi_2)$ and $s = \sin(\psi_2)$.

Using compounding operation, Equality (1) can be expressed as

$$pose_2^W = pose_1^W \text{ "+" } d^{pose_1}.$$

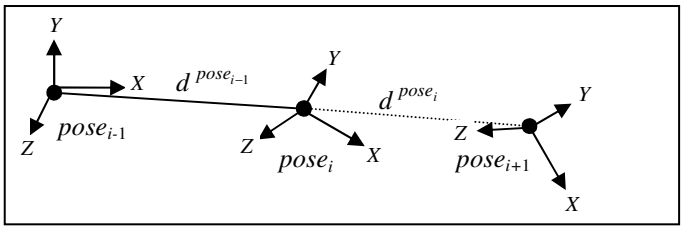

Figure 2. Motion model.

From the assumptions of constant translational and angular velocities we infer that the disparities between any two consecutive poses are approximately the same, i.e., from

$$d^{pose_{i-1}} = pose_i^W \text{ "–" } pose_{i-1}^W$$

$$d^{pose_i} = pose_{i+1}^W \text{ "–" } pose_i^W$$

we have

$$d^{pose_i} \approx d^{pose_{i-1}}, \text{ or}$$

$$d^{pose_i} = d^{pose_{i-1}} \text{ "+" } \varepsilon_i \quad (2)$$

where ε_i denotes the prediction error with zero mean and covariance matrix E_i.

We assume that the covariance E_i is a diagonal matrix and its sigma values depend on the maximal velocity $v_{\max}^W$, the maximal angular velocity $\omega_{\max}^C$, and the time step $\Delta\tau$. According to [7], we have

$$\mathrm{E}_i = \begin{bmatrix} V_{\max,\tau} & 0 \\ 0 & \Omega_{\max,\tau} \end{bmatrix}, \text{ where}$$

$$V_{\max,\tau} = (v_{\max}^W \Delta\tau)^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Omega_{\max,\tau} = (\omega_{\max}^C \Delta\tau)^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore, the covariance matrix of the prediction process can be estimated by the formula

$$\Lambda = J\ \mathrm{E}_i\ J^T \tag{3}$$

where J is the Jacobian matrix of f with respect to $(x, y, z, \psi, \theta, \varphi)^T$. For simplicity of presentation, the full 6×6 matrix J is omitted.

B. Measurement model

Given that the camera is at pose $pose_i^W$ and a 3D landmark $P = (X, Y, Z)^T$ is in the field of view of the camera, then the coordinates of the image feature p corresponding to the landmark will be $(u,v)^T$. Assuming that the camera motion is smooth and baselines are small, then the camera motion $C = (T, \Omega)^T$ relative to $pose_i^W$ is also small. Consequently, the motion field equation can be used to express the relationship between the 2D velocity of feature p and the relative camera motion,

$$\frac{d}{dt}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{1}{Z} & 0 & \frac{-u}{Z} & -uv & 1+u^2 & -v \\ 0 & \frac{1}{Z} & \frac{-v}{Z} & -1-v^2 & uv & u \end{bmatrix} \begin{bmatrix} T \\ \Omega \end{bmatrix}.$$

A pose is a tuple of six parameters. Hence, we can estimate the relative translation and rotation of the camera with respect to $pose_i^W$ by measuring the relative 2D velocities of at least 3 features. Given n image features detected in the image frame corresponding to the current pose $pose_i^W$, we have a vector of image feature coordinates $x = [u_1, v_1, ..., u_n, v_n]^T$ in this frame and a corresponding equation system

$$\frac{dx}{dt} = LC, \tag{4}$$

where L is called an interaction matrix. The dimension of L is $2n \times 6$.

An optimization method such as the method of least-squares can be used to estimate the new pose of the camera and hence the position of the landmark P with respect to the new pose

$$l_P^C = R^{CW}(\Omega)\ (l_P^{pose_i} - T),$$

where $R^{CW}(\Omega)$ is the rotation matrix of the world coordinates with respect to the new pose of the camera, and $l_P^{pose_i}$ is the landmark position with respect to pose $pose_i^W$.

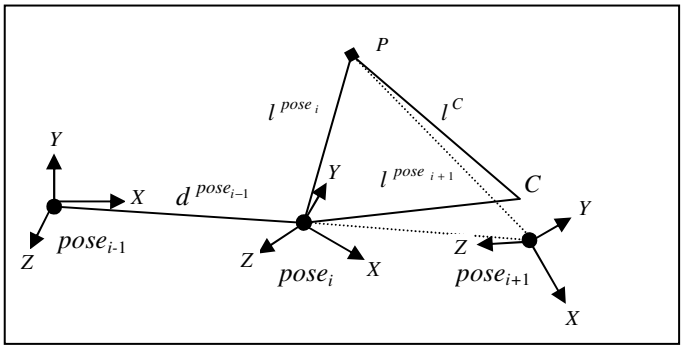

Figure 3. Measurement model.

Because of smooth motion and small baselines, the coordinates of landmark P with respect to the estimated pose $pose_{i+1}^W$ are approximately equal to its coordinates with respect to the pose $C = (T, \Omega)^T$:

$$l_P^{pose_{i+1}} \approx l_P^C, \text{ or}$$

$$l_P^{pose_{i+1}} = l_P^C + \xi_i$$

where ξ_i is the measurement error with zero mean and covariance matrix Ξ_i. This error consists of inherent uncertainties when estimating the parameters $(T, \Omega)^T$ in (4). In this paper, to simplify the uncertainty model, the covariance matrix Ξ_i is established using constant standard deviation values according to each experiment case.

III. EXPERIMENTAL RESULTS

Synthetic data have been used to evaluate the system; they include four landmark points that are always visible. While the simulated camera moves along the ground truth trajectory, the Image Generator component generates an image consisting of the projections on the image plane of the four fixed landmarks, and the Feature Detection and Matching component simulates the matching of these four projected feature points found in each pair of consecutive image frames.

Figure 4 shows the initial configuration for our experiments. The four landmarks are coplanar, lying in a plane parallel to the *XY*-plane. The distance from the plane to the world origin is 1. Initially, the camera is at the origin and its optical axis lies along the *Z*-direction. The four landmarks are at (0.2, 0.4, 1), (-0.2, 0.4, 1), (0.2, -0.4, 1), and (-0.2, -0.4, 1). Measurement units in all experiments are meter, radian and second. The time interval between each two consecutive image frames is 0.1 seconds. The maximum velocity $V_{\max,\tau}$ is assumed to be 0.3 m/s. Assuming that the camera does not rotate while moving, then the Jacobian matrix J becomes the identity matrix I. Therefore, the covariance matrix of the prediction process equals the covariance matrix of noise ε_i:

$$\Lambda = J\ \mathrm{E}_i\ J^T = \mathrm{E}_i\ .$$

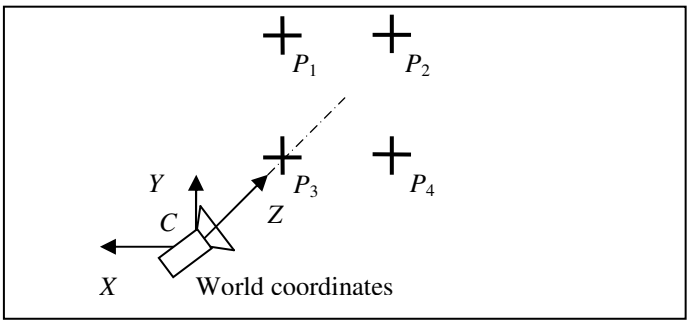

Figure 4. Initial configuration (right-handed coordinates). The four coplanar landmark points P_1 = (0.2, 0.4, 1), P_2 = (-0.2, 0.4, 1), P_3 = (0.2, -0.4, 1), P_4 = (-0.2, -0.4, 1) and the camera pose C = (0, 0, 0, 0, 0, 0).

A. *Experiment 1*

The Image Generator component generates images in an ideal virtual 2D image plane in which the coordinates of every point in the image plan are real numbers. The camera trajectory, as depicted in Figure 5, includes a line segment and a circle in the *XZ*-plane. The length of the line segment is 0.18 and the circle radius is 0.16. The camera moves along the line segment once and then along the circle 10 times without rotation. It moves one step in each time interval. The step size is 0.03 on the line segment and $2\pi/100$ on the circle. At each moving step, the ground truth data of the camera motion are generated. The measurement covariance matrix is assumed to be a diagonal matrix and all sigma values in the diagonal are 0.01 in Case a and 0.001 in Case b.

B. *Experiment 2*

This experiment is the same as Experiment 1, but with some noise added. Noise is added at 12 points in the ground truth data of the camera trajectory. All sigma values in the diagonal of the measurement covariance matrix are 0.01 in Case a and 0.001 in Case b.

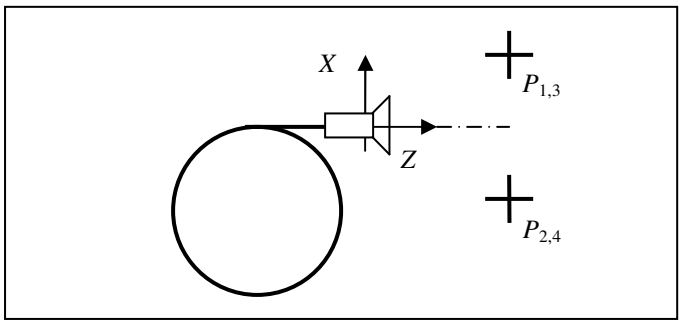

Figure 5. The geometry of the camera trajectory in all experiments. The camera moves along the line segment once and along the circle 10 times.

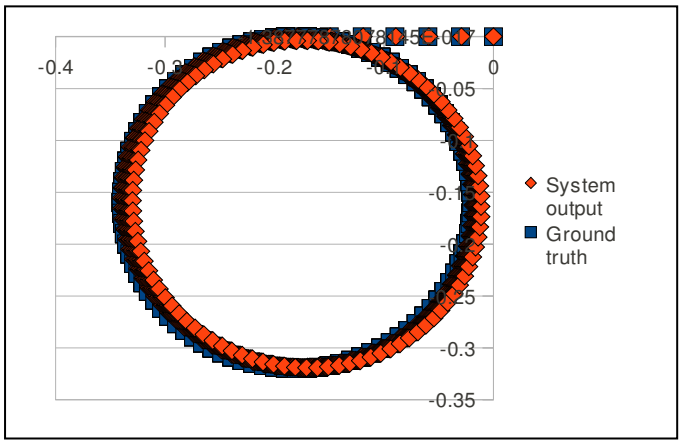

Figure 6. Camera trajectory determined in Experiment 1, Case a ($\sigma_z = \sigma_x = \sigma_y = 0.01$).

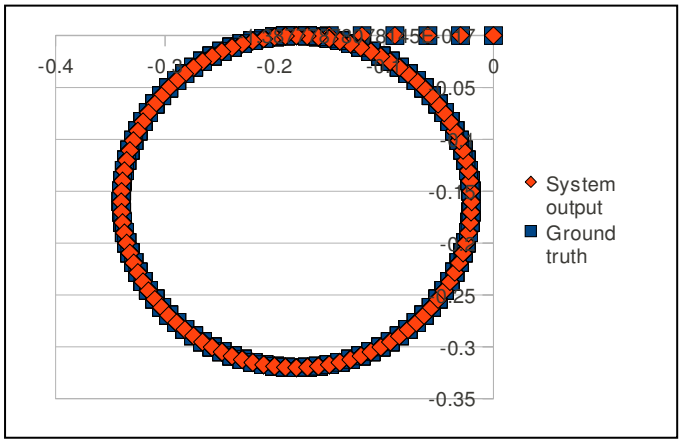

Figure 7. Camera trajectory determined in Experiment 1, Case b ($\sigma_z = \sigma_x = \sigma_y = 0.001$).

C. *Experiment 3*

In this experiment, the generated image in the virtual 2D image plane is digitalized in the form of pixels. The camera moves along the same ground truth trajectory as in that of Experiment 1. The step sizes are also the same. All sigma values in the diagonal of the measurement covariance matrix are 0.01 in Case a and 0.001 in Case b.

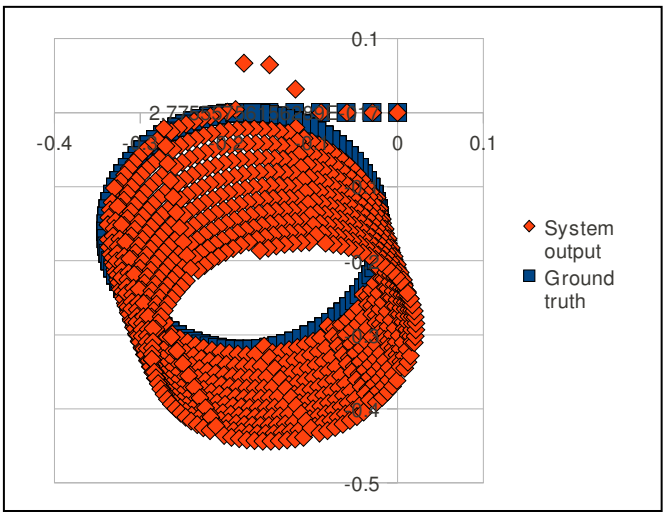

Figure 8. Camera trajectory determined in Experiment 2, Case a ($\sigma_z = \sigma_x = \sigma_y = 0.01$).

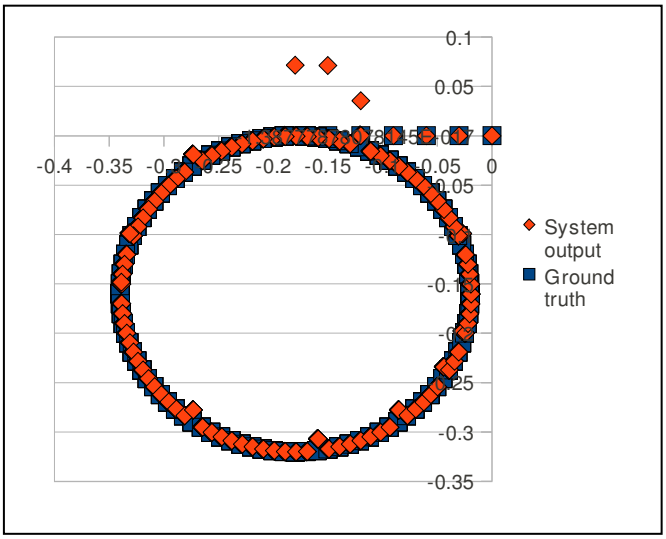

Figure 9. Camera trajectory determined in Experiment 2, Case b ($\sigma_z = \sigma_x = \sigma_y = 0.001$).

IV. Discussions

The purpose of Experiment 1 is to evaluate the measurement model of our system in an ideal environment with an ideal image plane and no noise. The output trajectory determined in Experiment 1b is much better than that in Experiment 1a. In Experiment 1b, the standard deviations of the determined trajectory with respect to the ground truth trajectory for all three axes are close to zero as shown in Table I. Additional experiments of this type have been done with various presumed measurement standard deviations and they show that the more precise the measurements of landmark positions (smaller sigma values) are, the closer to the ground truth trajectory the output is. This means that the more precise the measurements, the 'stronger' the influence of measurement model on the system output.

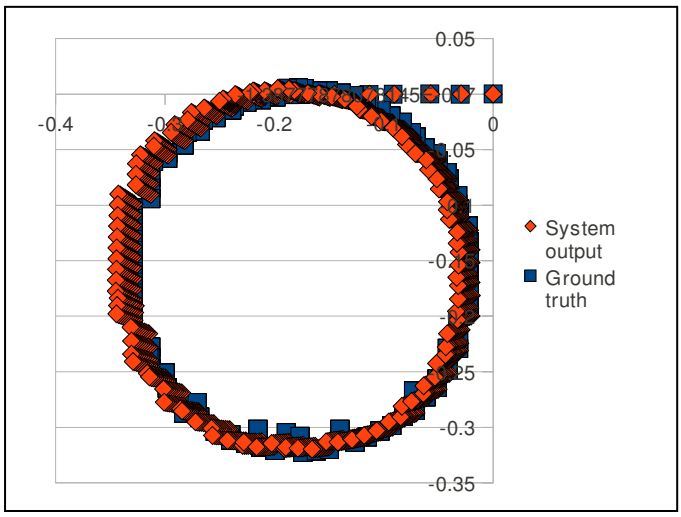

Figure 10. Camera trajectory determined in Experiment 3, Case a ($\sigma_z = \sigma_x = \sigma_y = 0.01$).

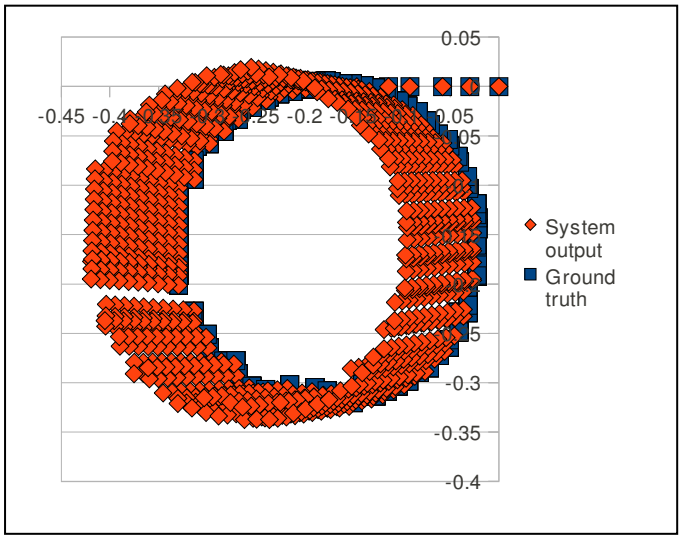

Figure 11. Camera trajectory determined in Experiment 3, Case b ($\sigma_z = \sigma_x = \sigma_y = 0.001$).

In Experiment 2, noise is added at 12 camera positions in the ground truth trajectory. We got the same result as in previous experiments: the output determined in Experiment 2b gets closer to the ground truth than that in Experiment 2a. Figure 9 also shows that the measurement model is so 'strong' that noise appears in the output trajectory as well.

If we compare the results of Experiments 1a and 2a, we observe that the outputs in both cases drift but the output in Experiment 2a drifts more severely than that in Experiment 1a even though the measurement standard deviations are the same in both cases. This is because the large sigma values 'weaken' the measurement model. Hence, the effect of the motion model, i.e. inertial motion, comes to be noticeable and the weighted effects of both models in both experiments make the outputs to be drifted. However, noise in Experiment 2a enforces the gradual drift, see Figure 8.

TABLE I. COMPARISONS OF STANDARD DEVIATIONS OF OUTPUTS.

Case	Maximum residual (in pixels)	Presumed measurement standard deviations $\sigma_{z,x,y}$ (in meters)	Standard deviations of the output trajectories with respect to the ground truth trajectory (in meters)		
			σ_z	σ_x	σ_y
1a	0.0013	0.010	0.0057	0.0026	0.0000
1b	0.0001	0.001	0.0000	0.0002	0.0000
2a	**0.0959**	**0.010**	**0.0288**	**0.0955**	**0.0000**
2b	0.0002	0.001	0.0007	0.0050	0.0000
3a	**0.8137**	**0.010**	**0.0067**	**0.0061**	**0.0017**
3b	**0.7999**	**0.001**	**0.0475**	**0.0099**	**0.0011**

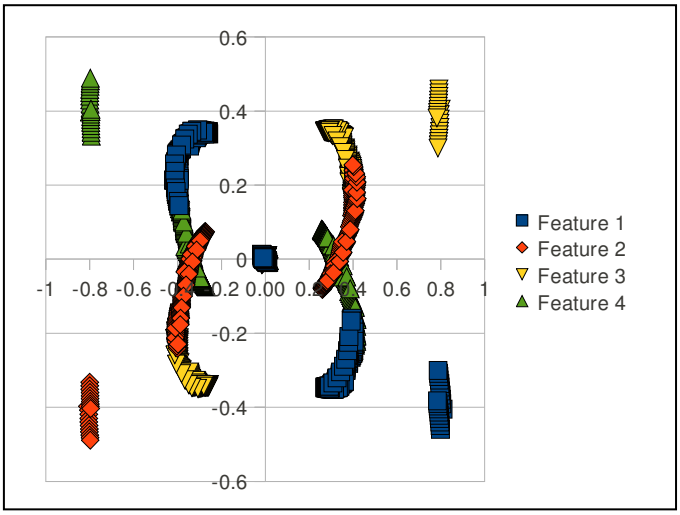

Figure 12. Residuals when optimizing system (4) of eight equations for the four landmarks in both Experiments 3a and 3b.

In Experiment 3, the Image Generator component digitalizes the synthetic image in the virtual image plane in the form of pixels, which makes landmark measurements become inaccurate significantly. Table I shows upper bounds on residuals obtained when solving System (4) of eight equations for the four landmarks using the least squares method. The residuals found in this experiment are so large that indeterminacies in estimating the pose parameters $(T,\Omega)^T$ cause large deviations of the estimated 3D landmark positions. In fact, the maximum residual can be up to 0.5 and 0.8137 in vertical and horizontal directions of the image respectively, see Figure 12. These large residuals cause the large actual deviations — up to 0.010 (i.e., 10 millimeters) in the *Z*-direction — of landmark position estimations, which are much greater than the presumed measurement standard deviations of 0.001 (i.e., 1 millimeter) in Experiment 3b. This results in the large drift in the determined camera trajectory depicted in Figure 11.

In Experiment 3a, the presumed measurement standard deviation is 0.01, which is close to the actual deviations of landmark measurements. Moreover, the standard deviations of the motion model calculated from (3) are always close to the pose prediction errors in all experiment cases. These correct values of the parameter sigma make the output trajectory look smooth (Figure 10).

V. CONCLUSIONS

Our experimental results, although using simulation, indicate the important role of measurement standard deviation values in determining the output trajectory accurately. They should be close to the actual measurement deviations. Values that are too large compared to the real deviations (Experiment 2a) or too small (Experiment 3b) will considerably impact the performance of iSAM module.

Our results also partly answer the question of Fakih that the performance of the iSAM module is quite sensitive to the measurement covariances whose correct sigma values are not easy to estimate in highly non-linear and noisy visual measurement cases.

ACKNOWLEDGMENT

We would like to thank Michael Kaess for his helpful support in hacking the iSAM module. We also thank the reviewer for his concise and valuable comments.

REFERENCES

[1] Randall Smith, Matthew Self, and Peter Cheeseman, "Estimating Uncertain Spatial Relationships in Robotics," in Autonomous Robot Vehicles, vol. 8 (1990) , pp. 167–193.

[2] Hugh Durrant-Whyte and Tim Bailey, "Simultaneous Localization and Mapping: Part I," in Robotics and Automation Magazine, June, 2006.

[3] Frank Dellaert and Michael Kaess, "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing," Intl. J. of Robotics Research, vol. 25, no. 12, Dec. 2006, pp. 1181–1204.

[4] Michael Kaess, Ananth Ranganathan, and Frank Dellaert, "iSAM: Incremental Smoothing and Mapping," IEEE Trans. on Robotics, vol. 24, no. 6, Dec. 2008, pp. 1365–1378.

[5] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse , "MonoSLAM: Real-Time Single Camera SLAM," IEEE Trans. PAMI 2007.

[6] Peter Gemeiner, Andrew J. Davison, and Markus Vincze, "Improving Localization Robustness in Monocular SLAM Using a High-Speed Camera," in Robotics Science and Systems, 2008.

[7] Sven Albrecht, "An Analysis of Visual Mono-SLAM," Master Thesis, 2009.

[8] Michael Kaess, "Incremental Smoothing and Mapping," Ph.D. Thesis, 2008.

[9] Adel H. Fakih, "Recursive Estimation of Structure and Motion from Monocular Images," Ph.D. Thesis, 2010.

[10] http://people.csail.mit.edu/kaess/isam.