Toward Multi-Stage Decoupled Visual SLAM System

Mohamed H. Merzban*, Mohamed Abdellatif*, Hossam Abbas*§ and Salvatore Sessa*¶
*Mechatronics and Robotics Eng. Dept.
Egypt-Japan University of Science and Technology, Alexandria, Egypt
Email:{mohamed.merzban, mohamed.abdellatif, hossam.abbas}@ejust.edu.eg
§Electrical Eng. Dept., Assiut University, Assiut, Egypt
¶Graduate School of Creative Science and Engineering, Tokyo, Japan
Email:s.sessa@aoni.waseda.jp

***Abstract*—SLAM is defined as simultaneous estimation of mobile robot pose and structure of the surrounding environment. Currently, there is a much interest in Visual SLAM, SLAM with a camera as main sensor, because the camera is an ubiquitous and affordable sensor. Camera measurements formed by perspective projection is highly nonlinear with respect to estimated states, leading to complicated nonlinear estimation problem. In this paper, a novel system is proposed that divides the problem into two parts: local and global motion estimation. This division leads to a simple linear estimation system. In the first stage, local motion parameters (acceleration, velocity, angular acceleration and orientation) are estimated in robot local frame. Robot position and the scene map are then estimated in the second stage in global frame as global motion parameters. Map is updated at each camera frame and is represented in a relative way to decouple robot pose from map structure estimation. The new system simplified the map correction to a linear optimization problem. Simulation results showed that the proposed system converges and yields accurate results.**

Keywords*—*Visual SLAM, Sensor Fusion, Inertial Sensors, Robot Localization, Relative Map, Graph Theory

I. Introduction

Localizing a robot is a basic need for several autonomous mobile robot applications. SLAM techniques aim to solve this problem by sensing the surrounding environment around the robot then building a map. Afterwards, the map is used to localize the robot in its environment. In most cases, robot sensors measure partial information about surrounding environment, for example landmark positions relative to the robot in case of laser range finder or perspective projections of landmarks on the image plane in case of camera. This fact implies that map structure should be estimated by fusing all partial measurements through a map correction/optimization process. This process is highly nonlinear in Visual SLAM systems [1] [2].

Extensive research was performed on SLAM problem. From our point of view, there were two challenges that motivated researchers. First, localization accuracy enhancement. This could be achieved by enhancing sensor accuracy or by improving computational aspect of the estimation process. Currently, there is no single sensor that gives accurate measurements under all conditions and in all environments. Therefore, sensing is usually enhanced by utilizing several sources of information and merging them using the probabilistic framework. Sensors like laser range finder, inertial sensors, GPS and camera were usually employed in localization process [3] [4] [5]. Considering computational aspects, localization accuracy may be enhanced by elaborating on map representation, either absolute or relative [6] [7] [8], or by elaborating on computational approach either Filtering or Optimization techniques [9] [2].

Second challenge is the computational cost of SLAM technique especially in large environments. As the map size increases, two operations scale up as well: feature matching and map optimization. Fortunately, research revealed that feature matching can be highly accelerated by utilizing appearance based methods [10] [11] [12]. These methods make advantage of text-retrieval techniques by decomposing scenes into visual words and searching for them in a vocabulary tree. On the other side, map optimization is still a computational issue. Current successful systems either summarize the map in a special way [3] or perform sub map optimization [13].

In this paper, we address two aspects of SLAM problem, accuracy and computational cost. A novel system for solving SLAM problem is introduced which exploits both vision and inertial sensors. The processing is divided into two stages, in the first stage, acceleration, velocity, angular acceleration, and orientation are estimated using EKF, while in the second stage, robot position and absolute landmark positions are estimated. The estimation system in the second stage is linear which leads to optimal and computationally efficient estimation. Furthermore, map structure is decoupled from position estimation by utilizing graph-based relative map. The map is updated whenever new camera measurement is received. Linearisation of the second stage with relative map representation which take all measurements into account helps to achieve accurate robot localization with computationally affordable map optimization. The novel two stage pipeline leads to linearisation of the costly map optimization, hence its solution is simpler than nonlinear optimization methods used in the literature [13] [3].

This paper is organized as follows: Section I introduces the current state of SLAM and proposed system. System structure and the algorithm description are presented in section II. Estimation of local motion is presented in section III, while section IV describes the global motion estimation procedure. Simulation results are presented in section V and conclusions

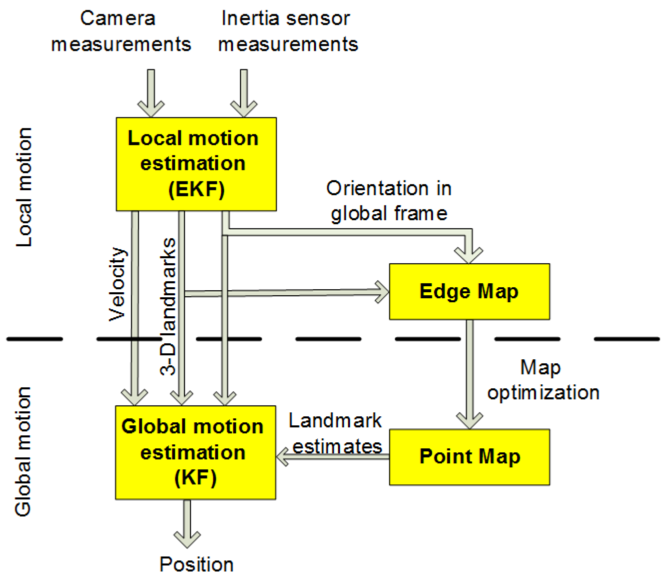


Fig. 1. Computational system of the proposed SLAM System.

are finally given in section VI.

Algorithm 1 Pseudo code of proposed SLAM algorithm

```
function ROBOT MOTION AND EDGE MAP ESTIMATION PROCESS
    for every new frame do
        Start local motion estimation.
        Add newly appearing landmarks to state.
        Remove disappeared landmarks from state.
        Local motion EKF correction step.
        Local motion EKF prediction step.
        Output estimated velocity and relative landmark positions.
        Start global motion estimation.
        Correct seen map edges using results of the first stage.
        New robot position EKF prediction step.
        Robot position EKF correction step.
    end for
end function
function MAP CORRECTION PROCESS
    for every frame in N frames do
        Use edge map as input
        Calculate solution for landmark positions
        Calculate covariance matrix for landmark positions
    end for
end function
```

II. System Overview

The proposed system is shown in Fig. 1. The relative map representation is utilized, in which the map is represented as a graph with landmarks as vertices and displacement vectors between different landmarks as edges. In this graph, the edges are used as state variables that are estimated and considered independent from the robot pose and from each other. This enables separate estimation of each edge vector, and hence simplifies the map update. The edge map is used for the map update, then for localization it is converted into landmark map. Pseudo-code for this algorithm is shown in algorithm 1. The processing can be divided into four steps: **1) Local motion estimation.** In which a vision sensor is employed besides inertia sensors including an accelerometer, gyroscope, and magnetometer to estimate acceleration, velocity, angular velocity ,orientation and map landmarks 3-D positions relative to local frame using EKF. **2) Update of edges in map graph.** The results of local motion estimation are used to update edges of the map graph. Edge measurements in the global frame are calculated from estimated 3-D relative positions of landmarks plus estimated orientation. The map edges are updated whenever any new measurement is received, hence there is no information loss, thanks to relative map representation. **3) Map correction/optimization.** An optimization criterion is used to reduce the map inconsistency. Using this criteria the edge map is converted to point map where each point is the position of a landmark. Map correction is only performed infrequently due to its computational load. It will be shown later that this process is equivalent to the solution of a linear system. **4) Global motion estimation.** Given estimated robot velocity as input, 3-D estimated positions of landmarks as measurements and optimized point map, the robot position is estimated through a simple linear process.

III. Local motion estimation

In this stage, vision and inertial sensors are employed to acquire information about the robot motion and the surrounding environment. It is assumed that the camera and inertial sensors frames of reference are coincident with robot local frame. The acceleration measured using an accelerometer includes gravity acceleration vector in addition to motion acceleration, to obtain motion acceleration only gravity should be subtracted vectorially from accelerometer reading. A very good estimation of orientation should be maintained to avoid interference of gravity. The angular velocity is measured using 3-axis gyroscope. Orientation of the robot is obtained by combining gyroscope, accelerometer, and magnetometer readings. Usage of magnetometer may be problematic because of magnetic field interference caused by magnetic field sources like electric motors. here we neglect interference. Usage of inertial sensors together with a camera requires pre-calibration to determine the relative transformation between the camera and inertial sensors. The work done by [14] illustrates how to achieve this calibration. Inertia sensors reading usually contains some slowly growing bias. It is assumed that the sensors are initially calibrated and hence there is zero initial bias. The slowly changing biases are modelled as being states of the system.

A. Estimation Model

The goal of the first stage is to estimate local motion parameters, namely acceleration a, velocity v, angular velocity ω, orientation quaternion q, landmark positions in local frame H_1 H_2 ... H_n and sensor biases a_b, $\omega_b$and M_b

Instead of Cartesian parametrization for landmarks, another representation which we call H-parametrization is utilized. It is related to Cartesian parametrization by the following relations:

$$h1 = \frac{1}{x} \qquad h2 = \frac{y}{x} \qquad h3 = \frac{z}{x} \tag{1}$$

The H-parametrization has been employed to enhance estimation accuracy. Using inverse depth for feature parametrization of features enables it to be better represented as probabilistic Gaussian random variable as discussed in [15].

The robot local motion parameters and local-frame landmark positions evolve according to the following prediction model:

$$a(k) = a(k-1) - \omega \times a(k-1) + n_a \tag{2}$$
$$v(k) = v(k-1) + a(k-1) * \delta t - \omega \times v(k-1) * \delta t + n_v \tag{3}$$
$$\omega(k) = \omega(k-1) + n_\omega \tag{4}$$
$$q(k) = q(k-1) * q_\epsilon * n_q \tag{5}$$
$$X_i(k-1) = T_{xyz}(H_i(k-1)) \tag{6}$$
$$X_i(k) = X_i(k-1) - v(k-1) * \delta t - \omega \times X_i(k-1) * \delta t + n_x \tag{7}$$
$$H_i(k) = T_H(X_i(k)) \text{ , i=1, 2, ..., N} \tag{8}$$

where T_{xyz} is the transformation from H parametrizaton to xyz parametrization and T_H is the transformation from xyz parametrization to H parametrization, X_i is xyz-parametrization of each landmark, H_i is H-parametrization of each landmark, q_ϵ is Rotation quaternion corresponding to the rotation vector $\omega\delta t$, it is the amount of rotation in δt time and R_θ is Rotation matrix that represents total orientation of the robot. Equation 7 represents the change in landmark positions from frame k-1 to frame k. Although the landmarks are stationary by assumption, it is needed to transform them from local frame k-1 to local frame k. This is done by observing that, if the robot moves with velocity v and rotates with angular velocity ω, The landmarks will look for an observer attached to the robot as if it moves with velocity $-v$ and angular velocity $-\omega$. Since H-parametrization is employed for the landmarks, they should be transformed to xyz-parametrization then evolved using estimated v and ω then transformed back to H-parametrization.

The inertial and vision sensors measurements are modelled by the following measurement equations.

$$a_m(k) = a(k) + R_\theta * g + a_b + v_a \tag{9}$$
$$\omega_m(k) = \omega(k) + \omega_b + v_\omega \tag{10}$$
$$M_m(k) = R_\theta * M_o + M_b + v_M \tag{11}$$
$$y_i = CameraProj(X(k)) \tag{12}$$

where a_m, ω_m, and M_m are accelerometer, gyroscope and magnetometer measurements respectively, a_b, ω_b, and M_b are Accelerometer, gyroscope and magnetometer biases respectively, M_o is Earth's magnetic field, y_i for i=1, 2, ... , n are 2-D vectors that represent measured image coordinates of landmarks, n_a, n_v, n_ω, n_q and n_x are process noise, v_a, v_ω and v_M are Measurement noise. Equation 12 is the camera measurement equation, where CameraProj is the nonlinear projection function. All estimated parameters are expressed in local frame except orientation which is expressed in global frame. Given the camera intrinsic parameters matrix K, This function may be written as

$$y = \frac{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * K * X}{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} * K * X} \tag{13}$$

EKF is utilized for the estimation and combining various sensor measurements.

B. Landmark initialization

The landmarks are initialized such that their uncertainty region is almost in the depth direction. For each new landmark, its projection on image plane [u v]^T of the camera and measurement uncertainty covariance matrix Q_{uv} are known. Its initial mean H_{init} and covariance Q_{init} are assigned as follows.

$$X_{init} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{14}$$

$$h_{init1} = h_{1initial} \quad h_{init2} = \frac{y_{init}}{x_{init}} \quad h_{init3} = \frac{z_{init}}{x_{init}} \tag{15}$$

$$H_{init} = \begin{bmatrix} h_{init1} \\ h_{init2} \\ h_{init3} \end{bmatrix} \tag{16}$$

$$K^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & s22 & s23 \\ 0 & s32 & s33 \end{bmatrix} \tag{17}$$

$$= \begin{bmatrix} 1 & 0_{1x2} \\ 0_{2x1} & S \end{bmatrix} \tag{18}$$

$$Q_{init} = \begin{bmatrix} q_{11initial} & 0_{1x2} \\ 0_{2x1} & SQ_{uv}S^T \end{bmatrix} \tag{19}$$

where K is intrinsic camera parameters matrix. There are two parameters $h_{1initial}$ and $q_{11initial}$ which represent mean and variance of landmark depth respectively. These values could be chosen arbitrarily to any reasonable value. A reasonable range for $h_{1initial}$ is from 0.5 to 2 and for $q_{11initial}$ is from 0.5 to 1. As the robot moves the true depth is resolved and uncertainty decreases. The depth direction is X-direction.

IV. Global motion estimation

Given the estimation of orientation from the first stage, then robot position and map can be computed in the second stage. Excluding orientation from our estimation model, it turns into a linear estimation problem. Two tasks are performed in global motion problem:

1) Estimation of robot absolute position from velocity input and measured landmark positions given the map.
2) Correction of map structure from landmark measurements.

In several SLAM systems the two problems were solved together [9] because there is a dependency between absolute landmark position and robot pose. To alleviate this problem, the map is represented as a connected graph of landmarks in which the landmarks are the vertices and the relative

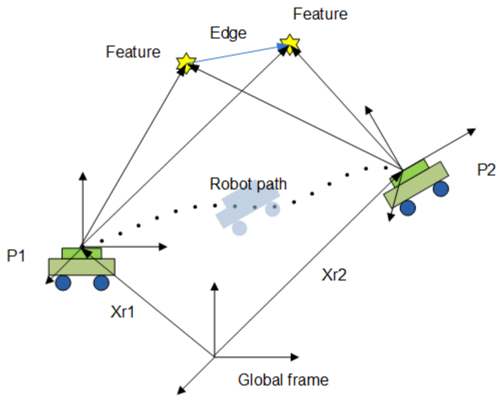


Fig. 2. An edge viewed from two robot poses.

displacement vectors between landmarks are the edges as shown in Fig.2. Since orientation is estimated in the first stage, it is possible to measure the edge vector in absolute frame directly and the edge vectors measurement are independent from the robot position. Instead of correcting the whole map whenever measurements are added to the map, it is only needed to update independent edges in the scene. But absolute landmark positions are still needed, The landmark absolute positions are calculated from map graph edges. This process will be formulated and solved as an optimization problem. The last problem to be considered is the absolute referencing. The map conversion process will yield only relative position of landmarks. To disambiguate the absolute position problem, it is required to have reference landmarks which may be selected from the first scene. A good algorithm for the determination of all other absolute positions can be found in [7].

The estimation system consists of the following states: **1)** Robot position. **2)** Map edge vectors. **3)** Landmarks absolute positions. The first two states represent independent set of states. On the other side, landmark absolute positions are dependent on edge vectors. Robot position is computed by simple integration of velocity. Edge measurements y^e_{ij} can be calculated from relative landmark positions $y_i(k+1)$. The system can be described by the following equations:

$$Xr(k+1) = Xr(k) + v(k) * \delta t \tag{20}$$

$$y_i(k+1) = R^T_\theta (L_i(k+1) - Xr(k+1)) \tag{21}$$

$$\begin{aligned} y^e_{ij}(k+1) &= y_j(k+1) - y_i(k+1) \\ &= R^T_\theta (L_j(k+1) - L_i(k+1)) \\ &= R^T_\theta e_{ij}(k) \end{aligned} \tag{22}$$

where: $Xr(k)$ is robot position, $e_{ij}(k)$ is edge vector connecting landmark i to landmark j, $L_i(k)$ is Landmark i position. where i=1, 2, ... , N_l, $v(k)$ is velocity input from first stage, $y_i(k)$ are landmark measurements from first stage, $y^e_{ij}(k)$ are edge measurement of $e_{ij}(k)$, R_θ is robot rotation matrix.

Equations 20 and 21 are used to estimate robot position given the landmarks absolute positions L_i, i=1, 2, ..., N_l. Kalman Filter (KF) is employed in state estimation. Since the system is linear the application of KF will be straightforward. It is assumed in this section that mean X_L and covariance matrix P_L of map landmarks are known. Of course they need to be calculated from map graph. This calculation will be presented in section IV-1. The map is updated at each new landmark measurement. Landmark measurements are converted to edge measurements as shown in Equation 22.

1) Map optimization: Map optimization is the process of calculating landmark positions and covariance matrices given a map graph with known edge vectors.

Due to measurement noise the edge graph will generally be inconsistent. The edge vectors will disagree on the location of a point that has several incident edges on it. To extract the landmark positions, an optimization criteria that minimizes the inconsistency error is needed. The total map inconsistency error E, is defined as the sum of all inconsistency errors for all edges.

$$E = \frac{1}{2}\Sigma_{i=1}^{N_l}\Sigma_{j=1,j\neq i}^{N_l}(L_j - L_i - e_{ij})^T S_{ij}(L_j - L_i - e_{ij}) \tag{23}$$

where we define $S_{ij} = P^{e\,-1}_{ij}$.

Once edge vectors e_{ij} are known, the next step is to optimize with respect to landmark variables L_i, i=1, 2, ..., N_l. The error E is differentiated with respect to L_k where k=1, 2, ..., N_l, and the result is set equal to zero. This leads to a matrix equation which is the result of optimization process.

$$AL = B \tag{24}$$

Where A, L, and B are defined by:

$$L = \left[L_1^T L_2^T ... L_{N_l}^T\right]^T \tag{25}$$

$$A = \begin{bmatrix} (\Sigma_{i=1,i\neq 1}^{i=N_l} S_{1i}) & -S_{12} & ... & -S_{1N_l} \\ -S_{21} & (\Sigma_{i=1,i\neq 2}^{i=N_l} S_{2i}) & ... & -S_{2N_l} \\ ... & & & \\ -S_{N_l 1} & -S_{N_l 2} & ... & (\Sigma_{i=1,i\neq N_l}^{i=N_l} S_{N_l i}) \end{bmatrix} \tag{26}$$

$$B = \begin{bmatrix} \Sigma_{i=1,i\neq 1}^{i=N_l} S_{1i} e_{i1} \\ \Sigma_{i=1,i\neq 2}^{i=N_l} S_{2i} e_{i2} \\ ... \\ \Sigma_{i=1,i\neq N_l}^{i=N_l} S_{N_l i} e_{iN_l} \end{bmatrix} \tag{27}$$

Equations 24 to 27 show that map optimization is equivalent to solving a linear system of equations, But it is noted that the matrix A is singular with nullity of three, this represents three translation degrees of freedom expected for relative map representation.

2) Solution of map optimization problem: Since matrix A is singular, one more constraint is needed to reach a unique solution. A reference point constraint is added, which means that it is assumed that there is a point in the map whose position is perfectly known. Last landmark position L_{N_l} is chosen as the reference landmark. We add this constraint as: $L_{N_l} = L_{ref}$. The last row in matrix A represents redundant equation. This redundant equation is replaced with constraint equation, then the new system will be:

$$A_d L = B_d \tag{28}$$

Where A_d and B_d are modified A and B matrices. The resulting system of equations is deterministic linear system which can be solved for L_1, L_2, ..., L_{N_l-1}.

3) Determination of covariance matrix: Equation 28 will yield the mean estimation of landmark positions. But, it is also needed to determine the covariance matrix Q_L of the landmarks. This is needed in robot position estimation.

For the determination of Q_L, the landmark positions L_1, L_2, ...,L_{N_l} are treated as Gaussian random variables with mean $\overline{L_1}$, $\overline{L_2}$, ..., $\overline{L_{N_l}}$. The edges e_{ij} are treated as Gaussian random variables with mean $\overline{e_{ij}}$.

The covariance Q_L is defined as:

$$Q_L = E[(L - \overline{L})(L - \overline{L})^T] \tag{29}$$

But we have

$$L = {A_d}^{-1} B_d$$

Substituting this value of L in equation 29 and after some mathematical manipulation, it can be proved that:

$$Q_L = \begin{bmatrix} A_{11}^{-1} & 0_{(3N_l-3)x3} \\ 0_{3x(3N_l-3)} & 0_{3x3} \end{bmatrix} \tag{30}$$

Having landmark positions L and landmark covariance matrix Q_L, all information required for robot position estimation is available.

V. Simulation Results

A synthetic environment was prepared to test the algorithm, where the true robot trajectory is generated as smooth piece-wise cubic polynomial curves. Landmarks were abstracted as randomly distributed points in a square area of 40x40 m area and 4 m height. Hypothetical camera and inertial sensors were attached to the robot. It was assumed that the camera provides 30 frames per second and delivers consecutive stream of images. Perfect feature matching is assumed. Attached accelerometer, gyroscope, and magnetometer supply Gaussian modelled noisy measurements at the same rate of the camera. System parameters are summarized in Table I.

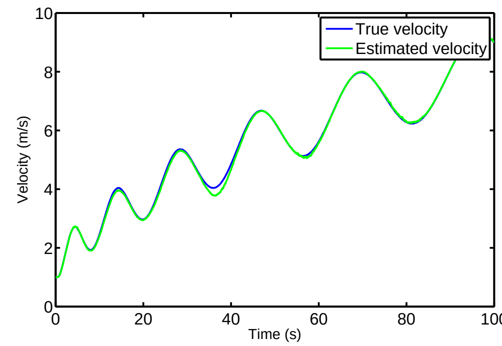
TABLE I. Simulation Parameters

Parameter	Value
Camera resolution	1024x1024 pixel
Camera viewing angle	120 degree
Std. of point landmark projection in image plane:	1 pixel
Std. of accelerometer noise:	0.05 m/s^2
Std. of gyroscope noise:	0.2 deg/s
Std. of magnetometer noise:	0.5 mgauss

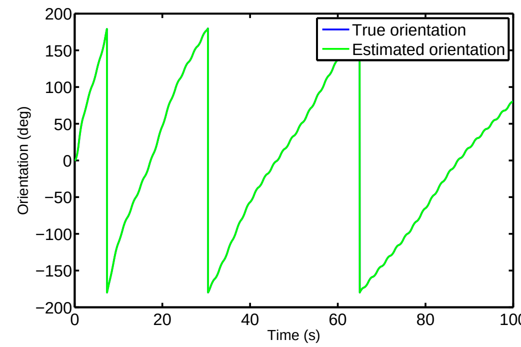
The robot estimates its 3-D motion variables while following the path. True and estimated motion variables are recorded during simulation for later accuracy evaluation.

The error metric at each time sample is defined by:

$$Error = \|x_{true} - x_{estimated}\|^2 \tag{31}$$



(a) velocity



(b) orientation

Fig. 3. True versus estimated velocity and orientation during first motion scenario.

The root mean square error (RMSE) during the whole trajectory is defined by:

$$RMSE = \sqrt{\frac{1}{N}\Sigma_i \|x^i_{true} - x^i_{estimated}\|^2} \tag{32}$$

Fig.3(a) and Fig.3(b) present the true versus estimated velocity (x-direction) and orientation (yaw angle), respectively. The RMSE error of each motion variable is listed in Table II.

TABLE II. RMSE error of local motion variables

motion variable	RMSE
acceleration	0.0868 m/s^2
velocity	0.0441 m/s
angular velocity	0.3405 deg/s
orientation	0.0585 deg

The ground truth/estimated path of the robot is shown in Fig.4 for a robot moving on circular path with constant velocity. Initially, the reference point is not visible to the robot, so it estimates position by only integrating velocity input. Once the reference point becomes visible, the map correction process is executed every fixed number of camera frames, in this motion scenario, the position RMSE error was 0.2445 m.

The ground truth/estimated path of the robot for a highly dynamic path with many rotations and in wide area is shown in Fig.5. Position error depends on map correction rate, therefore results at different map correction rates are presented in Table III, the RMSE position error for cases of 5, 20, 50 and 200 correction rate is presented.

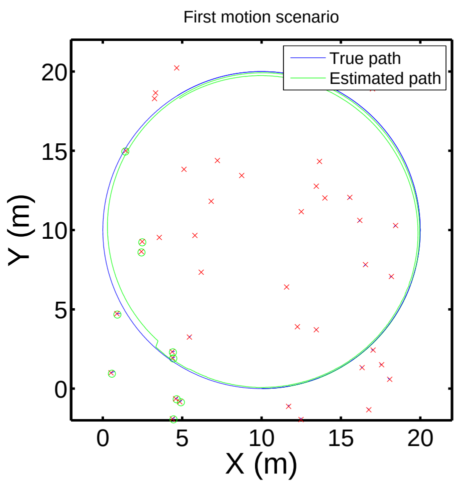


Fig. 4. First motion scenario.

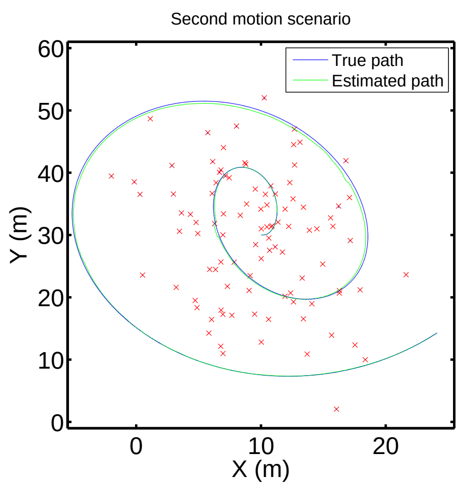


Fig. 5. Second motion scenario.

TABLE III. RMSE POSITION ERROR

Map optimization frequency	RMSE
every 5 frames	0.1368 m
every 20 frames	0.1883 m
every 50 frames	0.2095 m
every 200 frames	0.2030 m

VI. CONCLUSIONS

In this paper, a novel system for solving SLAM problem was presented. It was shown how to break the problem into two simpler problems, local motion estimation and global motion estimation. It had been shown that the two problems of robot position estimation and map structure estimation can be decoupled. The main advantage of this architecture is that the computationally expensive process of correcting the map is now linear and that there is no need to skip any camera frame or any other sensory information. It is possible to perform map correction over long periods without affecting long-term map accuracy.

Estimation of local motion is a complex nonlinear estimation process but is independent from the mapping process, so its computational cost will depend only on the number of features detected in the scene, hence nearly constant. The most computationally expensive process is map optimization. For full information matrix A, the optimization will require $O(N^3)$ time , where N is the total number of landmarks stored in the map. On the average, due to its structure, matrix A will be sparse with M number of entries in each row where M is the average number of landmarks appearing in single camera image. By using sparse matrix solvers, the time cost can be reduced to $O(NM^2)$. Our future work is to enhance the map optimization procedure by making it incremental instead of global optimization to the whole map that is computationally expensive.

ACKNOWLEDGMENT

The first author is supported by a scholarship from the Ministry of Higher Education, Government of Egypt which is gratefully acknowledged.

REFERENCES

[1] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM," *IEEE Intelligent Transportation Systems Magazine*, pp. 31–43, 2010.

[2] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, 2007.

[3] K. Konolige and M. Agrawal, "FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping," *Robotics, IEEE Transactions on*, no. 5, pp. 1066–1077, 2008.

[4] B. Clipp, J. Lim, J.-m. Frahm, and M. Pollefeys, "Parallel , Real-Time Visual SLAM," *Intelligent Robots and Systems IROS*, 2010.

[5] C. Roussillon, A. Gonzalez, J. Solà, J.-M. Codol, N. Mansard, S. Lacroix, and M. Devy, "RT-SLAM: A Generic and Real-Time Visual SLAM Implementation," 2012, cite arxiv:1201.5450Comment: 10 pages. [Online]. Available: http://arxiv.org/abs/1201.5450

[6] Z. Wang, S. Huang, and G. Dissanayake, "D-slam: Decoupled localization and mapping for autonomous robots," in *Proceedings of the 12th International Symposium of Robotics Research*, 2005.

[7] A. Martinelli and N. Tomatis, "Open Challenges in SLAM: An Optimal Solution Based on Shift and Rotation Invariants," in *In Proceedings of the International Conference on Robotics and Automation - ICRA*, 2004.

[8] V. Nguyen, A. Martinelli, and R. Siegwart, "Improving the Consistency of Relative Map," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 3556–3561.

[9] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 1052–1067, 2007.

[10] C. Mei, G. Sibley, and P. Newman, "Closing loops without places," *Intelligent Robots and Systems IROS 2010 IEEERSJ International Conference on*, pp. 3738–3744, 2010.

[11] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, no. 6, pp. 647–665, 2008.

[12] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 1470–1477.

[13] H. Strasdat, A. Davison, J. M. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual SLAM," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2352–2359.

[14] T.-C. Dong-Si and A. Mourikis, "Estimator initialization in vision-aided inertial navigation with unknown camera-IMU calibration," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 1064–1071.

[15] J. Civera, A. Davison, and J. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *Robotics, IEEE Transactions on*, no. 5, pp. 932–945, 2008.