Agricultural Journal 14 (5): 109-117, 2019
ISSN: 1816-9155

A Linearly Optimization Method for Slam Estimation Based on Stereo Features

[1]Mohamed H. Mahmoud, [2]Nashaat Hissen and [3]Elsayed Hemayed
[1]Department of Communications Engineering, MSA University, Egypt
[2]Department of Communications Engineering, Fayoum University, Cairo, Egypt
[3]Department of Computer Science Engineering, Cairo University, Giza, Egypt
[3]Zewail City for Science and Technology, Giza, Egypt

Abstract: Recently graph based SLAM has become a popular representation for solving SLAM problem but it is affected by its inherent nonlinearity due to noisy measurements and reprojection errors which lead to non-consistent map. The proposed system provides the entire robot trajectory at lower cost using stereo camera, generates a globally consistent 3D map and provides linear graph optimization. We show that the validation of our system based on stereo camera which utilizes a linear optimization method which provides only one accurate solution in contrast to the well-known nonlinear global optimization like BA of ORB-SLAM which provides multiple non-accurate solutions for optimization. Furthermore, our system runs in real time and is 3X faster than the current state-of-the-art SLAM systems for stereo SLAM. Our system gives comparable accuracy compared with the state of the art. We tested our system using KITTI data sets. We show that our system can be an alternative to bundle adjustment approaches with a better accuracy, robustness and more efficient than some well-known systems.

Key words: SLAM (Simultaneous Localization and Mapping), robotic systems, stereo camera, tracking, loop closure

INTRODUCTION

SLAM is an active research topic for more than 30 years with overwhelming literature in the robotics community (e.g.,). Mathematically, SLAM is modeled as a high-dimensional nonlinear estimation problem whose aim is to build a map for unknown environments in obscurity of referencing systems like GPS and simultaneously utilize the built map to find the "optimal" estimate for robot poses (location and orientation) and correct errors of previous estimations using noisy measurements and uncertain priors. The measurement function is generally a nonlinear objective function in its arguments especially in case of camera sensor. Since, the distance and angle of robot w.r.t locations of features are computed using trigonometric functions that are non-linear in the coordinates of the camera and features.

Generally, a graph-based SLAM system can be divided into three sequential modules (Bailey and Durrant-Whyte, 2006) frontend, backend and map representation. The frontend is used for measurement of motion which processes sensor data to extract geometric motion and spatial constraints (data association), e.g., between the key frame and map points at different points in time. The backend (optimizer) is used to estimate and correct the poses of the keyframes (maximum a posterior) to obtain a consistent map of the environment given the constraints from front-end. For this target, back-end solutions have developed gradually from lter based methods (Davison *et al*., 2007; Civera *et al*., 2007; Li and Mourikis, 2013; Hesch *et al*., 2012; Lynen *et al*., 2013; Jones and Soatto, 2011) to graph optimization methods (Klein and Murray, 2007; Engel *et al*., 2013; Mur-Artal *et al*., 2015; Mur-Artal and Tardos, 2017).

In summary, we propose a real-time visual SLAM system with the following properties: Our system keeps the separable structure of SLAM, since we estimate the linear variables using nonlinear measurements corrupted by Gaussian noise. Thus, the investigated system has a linearity property which is a significant improvement compared to other systems.

Our system does not only maximize the accuracy of the robot trajectory but also impedes drift-buildup that minimizes the number of failures of the tracking. Thus, achieving accurate and globally consistent framework compared to that of the state-of-the-art techniques.

Our implementation provides a faster linear SLAM. Furthermore, our proposed system can detect and correct the erroneous loops in the trajectory of robot using stereo camera.

Corresponding Author: Mohamed H. Mahmoud, Department of Communications Engineering, MSA University, 6th of October, Egypt

MATERIALS AND METHODS

System preliminaries and problem formulation

Graph based SLAM preliminaries: Let $P = \{p_1, ..., p_N\}$ be a set of N nodes representing position of a mobile robot at consecutive time instants (frame poses) and $\xi = \{e_1, ..., e_M\}$ be a set of M edges represent relative translation vector between every two nodes e_{ij}. The objective of pose graph optimization is to compute an estimate of the nodes con guration that maximizes the likelihood of the measurements by using mathematical model that is solved by linear method. Since, relative pose measurements are affected by noise, the measured quantities are in the form $\varepsilon_{ij} = \varepsilon_{ij} + s_{ij}$ where $s_{ij} \in R3$ is a zero-mean Gaussian noise, i.e., $s_{ij} \approx N(0, c_{ij})$ being c_{ij} a 3×3 covariance matrix. The uncertainty of each edge measurement (covariance matrix) $C_{e_{ij}}$ is obtained easily from the uncertainties of the two frames locations as given by Eq. 2 as follows:

$$e_{ij} = p_j - p_i \quad (1)$$

$$C_{e_{ij}} = C_{pi} + C_{pj} \quad (2)$$

Problem definition: The role of localization thread of SLAM system is to compute positions of the robot (frames) in the world coordinate frame: $p_t = \{x, y, z\}$ where x, y, z are translation coordinates.

For each frame we need to find a set of 2D image points U such that u = proj(X, t) where $X \in R3$ and proj is the projection function that, projects 3D triangulated points X from current observation (X, t) in the map of SLAM to the 2D image plane.

Figure 1 shows the projection model of stereo camera. To keep the model as close to the ground-truth as possible, we maximize the likelihood of the measurements by using mathematical model that is solved by linear method and minimize reprojection error. The variables to be optimized are p_t. It should be mentioned that the projection function proj is strongly nonlinear, hence, the optimization problem has many minima, hence, we should have good initial solutions for p_t. Fortunately, the previous stage of RANSAC and triangulation provide such good initial solutions (Fig. 2).

In a summary our target in this research is the detection of loops and graph optimization. We can distinguish the two different threads as described in Fig. 2, although, they are closely linked. The first thread consists of the identification of a previously visited place (loop). When a loop is detected, a new relation is added to the graph that relates the current pose with the pose in the past where we visited the same place. The second thread tries to reduce the accumulated errors from the pose estimation based on pairwise alignment and reprojection errors. This will get more consistent maps, especially when a place is revisited after a long period of time.

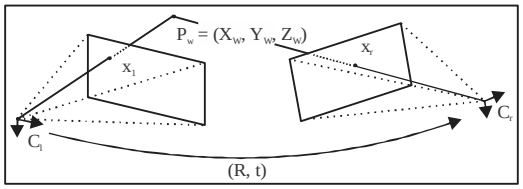

Fig. 1: Stereo camera projection model

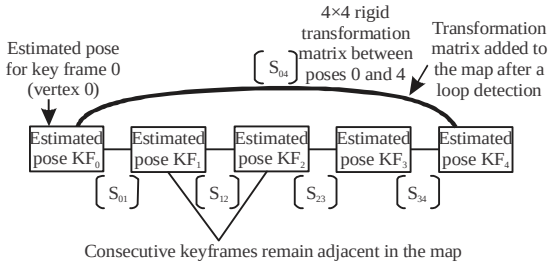

Fig. 2: Keyframe poses optimization and loop detection

General structure of proposed V-SLAM system: The general structure of the proposed vSLAM system is depicted in Fig. 3 and its main modules are described in the following sections. Our system generates a global, scalable and reliable 3D map based on pose-graph of keyframes and computing relative poses of frames. Our proposal SLAM system has three main modules; tracking, loop closure and optimization modules. Our SLAM system can be divided into two parts front-end and back-end. The front end is used to determine camera pose (translation and rotation) and detect loop closure. The back-end part is used to optimize the trajectory of robot by reducing accumulated errors using our new investigated method. This ecient distribution allows for a continuous tracking of the tracking module while the global optimization and the loop closure ones are processed in the background only when a new keyframe is added. System takes a set of pairwise relative poses between cameras and rotations as input and outputs the position of all keyframes in global coordinate. The camera poses are computed through motion averaging linear algorithm.

Tracking module: Most visual SLAM systems used stereo camera for tracking and mapping. A common configuration is a stereo (two) cameras that are fixed relative to each other with the fixed transformation between them known at prior. A stereo-camera is able to recover ambiguous scale by exploiting the parallax (difference) between the two captured images. Unfortunately the parallax diminishes as the imaged scene is very far from the camera (the two images become essentially the same). We use a general stereo camera model in which the camera mapping function is termed perspective projection and the parameters for each frame

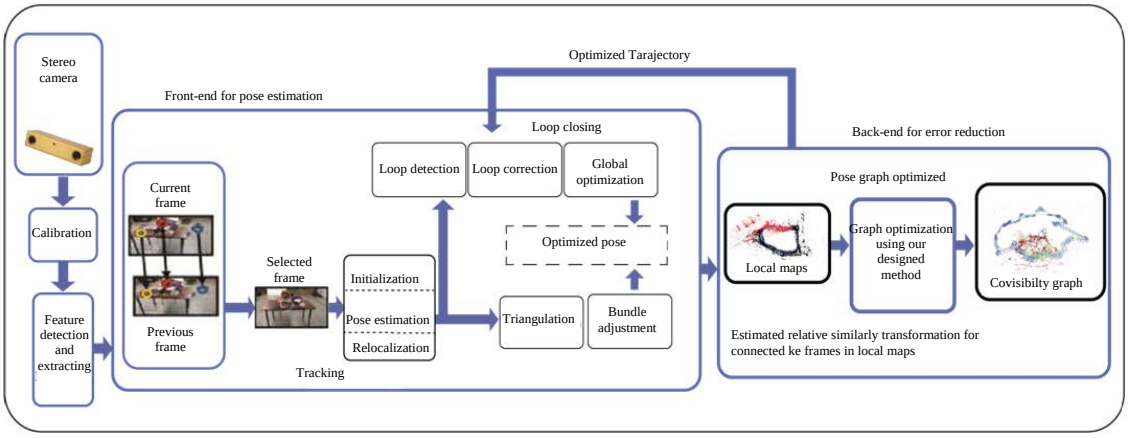

Fig. 3: Overview of the global vSLAM pipeline that is implemented in our designed system

the 3×3 rotation matrix, the 3×1 translation vector as illustrated in Fig. 1, the focal length of every camera, the baseline, the left intrinsic camera matrix, the right intrinsic camera matrix and the two radial distortion parameters and. The relations for projecting a 3D point into image plan is:

$$x_l = K_l X \quad (3)$$

(this relation to project 3D point X to image plane that converts from world to left camera coordinates):

$$x_r = K_r RX+t \quad (4)$$

(this relation to project 3D point X to image plane that converts from world to right camera coordinates). In order to use a stereo camera, we can estimate the intrinsic parameters (focal lengths, skew and the principal point) and extrinsic parameters (rotation and translation describe the pose of the right camera with respect to the left camera) using camera calibration tool used in (Mur-Artal and Tardos, 2017). We set the world frame coordinates to coincide with the left camera frame coordinate.

Our SLAM system initially loads the frames, then detects and extracts the features, then processes them by the tracking subsystem. Once the keyframe selection is determined, an initial map is generated by triangulating feature points in the initialization thread. Our global tracking module pipeline is illustrated in block diagram of our system. The pipeline of tracking based vision system includes processes to detect features from each frame, extract features from each frame, match the current frame features to the previously frame features, compute relative pose of the current frame with respect to the previous frame and optimize relative pose computations using Bundle Adjustment (BA) algorithm. The key solution of global localization can be classified according to whether the features correspondence is specified in 2D or 3D as follows:

2D-2D feature correspondence: This type is used for vision based localization and the relative pose between two frames is determined by decomposing of essential matrix using a set of correspondence points from the two frames. But the scale factor is unknown and we can compute it independently.

3D-3D feature correspondence: This type is used for stereo vision based localization and the relative pose between two frames is represented by rigid transformation and can determined by point clouds of the two frames using ICP algorithm (Iterative Closest Point).

3D-2D feature correspondence: This type is used for projection from world coordinates (environment or space) to frame plane using 3×4 camera matrix (intrinsic and extrinsic parameters) and the pose estimation is computed, if there are three or more 3D-2D correspondences are obtained, the pose estimation is a problem called Perspective-n-Point (PnP) for perspective camera.

The feature is a specific 2D structure in the frame that can be descripted like point and corner. Detection of image features is a key-operation for any visual system to detect repeated desirable points then compute a unique descriptor vector for feature used for feature matching. Oriented FAST and Rotated BRIEF feature (ORB) (Rublee *et al*., 2011) is one of the most powerful invariant rotation and fast feature descriptors that describes features by binary representation, reduces runtime performance of system and works as follows:

- The image is filtered using median filter, to eliminate the noise
- FAST corner detection (for ORB feature detection), a point P is considering a FAST corner when this point has enough gray value pixels in different gray area around the point

- Determine the directions of these points in intensity centroid
- Extracting BRIEF binary descriptor
- Determine low pixels blocks in greedy algorithm

Finally, the descriptor of each feature is a vector of 256-bit binary array descriptors that represents the normalized histogram of image gradient at the feature location. The next step is to track the image features into the consecutive image frames, this necessitates matching the features of every new frame to the old ones. Two features are considered matched when their descriptors are similar to a sufficient extent. A matching function is used to compare the two feature descriptors yielding a matching value that will be compared to a threshold to decide if there is a match. The matching function for ORB descriptors is the minimum distance between the two descriptor vectors and can be calculated using Hamming distance which given by this function:

$$\mathrm{Ham}(x;y) = \sum_{i=1}^{n} \mathrm{weight}_i \left(b_i(x) \otimes b_i(y)\right) \quad (5)$$

Where:

$b_i(x) \in \{0; 1\}$: Binary descriptor i of the current frame
$b_i(y) \in \{0; 1\}$: Binary descriptor i of the previous frame
$\otimes$: Stands for bitwise XOR operation

Matching between frames is used to find correspondence of keypoints over a sequence of frames to keep only key frames to maintain robot trajectory which whose topology differs according type of sensor used. The outliers of matches are filtered depending on constraint of epipolar. The RANSAC algorithm is applied to get fundamental matrix by discarding the matches whose distance from frame points to their corresponding epipolar line are >2 pixels.

Loop closure: This section illustrates the work of detecting loops and correcting the erroneous of loops in the pose graph of optimization. Our implementation is based on visual feature matching to determine if an area has been visited or not. For each new keyframe created by the tracking subsystem, feature matching is performed. We use a RANSAC algorithm with 30 iterations to check for 3D-correspondences. If the number of resulting inliers is above a certain threshold (40 matches), we consider that the current frame and the keyframe to which pairwise alignment is performed, a good matching. And hence, an optimization with all matches are performed and a similarity transformation SIM3 matrix is added from the current keyframe to each loop keyframe (4×4 transformation matrix is created). Only these keyframes which have a higher similarity score than a threshold and are not directly attached to the current keyframe are considered as loop keyframes. This thread is illustrated in Fig. 2. Once a list of loop keyframes is found, this gives a measure on the accumulated error.

The following steps are performed to increase the robustness of our loop detection. First, the map features detected in the loop keyframe and its non-directly connected keyframes are reprojected using the SIM3 matrix to get more correspondence matches. If the total matches were enough (60 matches), this keyframe is accepted and considered as a trusted loop keyframe and SIM3 transformation matrix is a loop keyframe and its pose is computed using our linearized method nearby with the locations of its directly connected keyframes in the covisibility graph. A new edge in the covisibility graph is generated between the two keyframes of the loop. Furthermore, a graph optimization is provided to get better accuracy of the map on localization stage.

Graph optimization (error reduction subsystem): This module represents back-end subsystem and has modules: local maps, covisibility graph and optimization algorithm, we will describe work of optimization and mathematical form in the following section.

Optimization algorithm: For each new frame, we compute camera poses and their related uncertainty represented by information matrices. The 3D keypoints are then projected to the new keyframe pose and the reprojection errors are minimized to obtain both the camera pose and the information associated to such estimation. Graph optimization algorithm (covisbilty graph optimization) is the thread of computing keyframes (or points) positions and information matrix with known edge vectors. There is an offset ambiguity in the edge map. To resolve this ambiguity, we add a special reference keyframe to the surrounding environment. The robot should be able to identify this reference keyframe and evaluate the map offset such that the position of the reference keyframe is always fixed. For our later discussions we will assume that the last keyframe pose p_{Nl} is the reference keyframe (or frame).

There is an inconsistency in the map because of noisy measurements. To reach the optimized trajectory of robot, we provide an objective function that minimizes the inconsistency as possible (reduce errors part) by minimizing re-projection geometric error (difference measure between the predicted and the measured reprojections), corresponding to edge position observations model which is described in Fig. 4. We define the inconsistency error err_{ij} that connects p_i to p_j as follows:

$$err_{ij}|_{trans} = p_j - p_i - e_{ij} \quad (6)$$

The orientation is computed with sufficient accuracy using the methods proposed by (Kneip *et al*., 2012 and

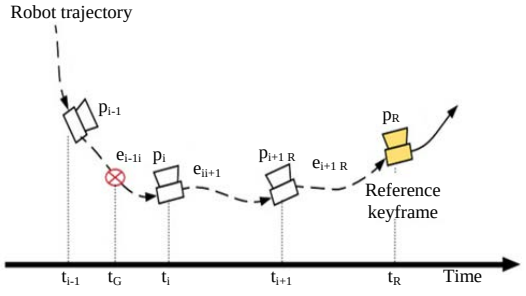

Fig. 4: Robot poses for optimization algorithm

Kneip and Lynen, 2013). We exclude orientation computation from our estimation model to make the estimation problem a linear estimation model. The edge variable is an additive white Gaussian measurement noise with covariance Σ_{ij} is considered 3-D random vector with Gaussian distribution with $\bar{e}_{ij}$ mean and covariance matrix Σ_{ij}. Uncertainty in edge vector represented by its covariance matrix Σ_{ij} varies from edge to another. This inconsistency can be formulated as a non-linear least squares problem (edge vector normalized error $nErr_{ij}$) as follow:

$$nerr_{ij}|_{trans} = err_{ij}^{T}\sum_{ij}^{-1} err_{ij} = \left(p_j - p_i - e_{ij}\right)^{T}\sum_{ij}^{-1}\left(p_j - p_i - e_{ij}\right) \quad (7)$$

For simplicity we set $S_{ij} = \Sigma_{ij}^{-1}$. The matrix S_{ij} represents the amount of information or certainty in the edge vector between two nodes i and j. Now, we can write $nerr_{ij}$ as:

$$nerr_{ij_{trans}} = \left(p_j - p_i - e_{ij}\right)^{T} S_{ij}\left(p_j - p_i - e_{ij}\right) \quad (8)$$

Since, the probabilities densities are Gaussians, we show that maximizing measurement likelihood of edge position observation model is equivalent to minimizing the sum of the weighted residual errors $nerr_{ij}$. Finally, we are now able to define the total graph incosistent error F as the objective function which is the double summation of all inconsistent errors due to all edges vectors of graph as follows:

$$F = \frac{1}{2}\sum_{i=1}^{N_l}\sum_{j=1,\,j\neq 1}^{N_l}\left(p_j - p_i - e_{ij}\right)^{T} S_{ij}\left(p_j - p_i - e_{ij}\right) \quad (9)$$

By using the mathematical methods for optimization (Barooah and Hespanha, 2007; Merzban *et al*., 2013; Yin *et al*., 2014), we can derive Eq. 7. The objective is toassociate an absolute pose to each node in the graph. We will obtain matrix equation form for linearization system which is the output of the optimization algorithm as follows:

$$AP = B \quad (10)$$

Table 1: Dimensions of terms appeared in graph-based SLAM

Terms in graph-optimization algorithm. of SLAM	Symbol	Dimension
Relative displacement vector between two keyframes i and j	e_{ij}	3*1
Covariance matrix of edge	Σ_{ij}	3*3
State vector (linearized solution)	P	3N*1
System matrix	A	3N*3N
Coefficient vector	B	3N*1

N: Number of keyframes (vertices)

where, A (system matrix), P(state vector), B(coefficient vector) are defined by:

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \cdots \\ p_{N_l} \end{bmatrix} \quad (11)$$

$$A = \begin{bmatrix} \left(\sum_{i=1,i\neq 1}^{i=N_l} S_{1i}\right) & -S_{12} & \cdots & -S_{1N_l} \\ -S_{21} & \left(\sum_{i=1,i\neq 2}^{i=N_l} S_{2i}\right) & \cdots & -S_{2N_l} \\ \cdots & & & \\ -S_{N_l 1} & -S_{N_l 2} & \cdots & \left(\sum_{i=1,i\neq N_l}^{i=N_l} S_{N_l i}\right) \end{bmatrix} \quad (12)$$

$$B = \begin{bmatrix} \sum_{i=1,i\neq 1}^{i=N_l} S_{1i}e_{i1} \\ \sum_{i=1,i\neq 2}^{i=N_l} S_{2i}e_{i2} \\ \cdots \\ \sum_{i=1,i\neq N_l}^{i=N_l} S_{N_l i}e_{iN_l} \end{bmatrix} \quad (13)$$

Equation 11-13 are the solution to the optimization problem expressed in Eq. 10. The dimensions of every vector and matrix is given in Table 1. These results show that graph optimization is equivalent to solving a linear system of equations. So, the proposed system is truly linear without approximation and has the following properties:

Matrix A is symmetric: this is due to the fact that S_{ij} is a symmetric matrix, hence, the off diagonal elements of A will be the same.

Matrix A is singular with nullity = 3. This is true because any sub matrix row in matrix A is the negative sum of all other sub matrix rows.

Matrix A is positive semidefinite which can be derived. Because matrix A is symmetric and positive definite, it satisfies the conditions of covariance matrix. We will highlight later the fact that A matrix is the information matrix of the map. In Table 1, we provide the dimensions of vectors and matrices used in our implemenation. Equation 10 will compute the mean estimation of keyframes poses. In next section we provide formula to compute the covariance matrix Σ_P of the poses of keyframes. In our system we assume that the reference pose of the robot to be the last global frame, i.e. $P_{Nl} = 0_3$.

We can use graph optimization for point map optimization. The edges are independent from nodes and from each other. This process enables us to estimate each edge vector alone which extremely simplifies the map update. Instead of updating the whole map every time, we add measurements to the map, we only need to update the optimization problem and then solve it to yield keyframes (or frames) positions which represent full optimized trajectory.

Computing the information matrix forthe translation vector: To compute the unknown 3D keypoints locations and keyframes poses from the observations, we minimize their total prediction error. Our optimization is the model refinement. Hence, it is essentially a matter of optimizing a nonlinear objective function represents the total reprojection error over nonlinear parameters (the features and camera parameters). The inverse Hessian H (second order derivative) at the minimum is a good estimate of the covariance matrix (uncertainty amount) for these parameters. The objective function defined as the mean square error of reprojection errors as illustrated (defined according to the method of least squares):

$$E = \sum_{j=1}^{N} \left(u_{1j} - u_{ij}^{*}\right)^{T} \left(u_{1j} - u_{1j}^{*}\right) + \sum_{j=1}^{N} \left(u_{2j} - u_{2j}^{*}\right)^{T} \left(u_{2j} - u_{2j}^{*}\right) \quad (14)$$

Where:

$$u_{ij}^{*} = \text{proj}(X`, \text{image}1)$$
$$u_{2j}^{*} = \text{proj}(X, \text{image}2)$$

where, u_{1j}, u_{2j} are the projection of keypoint j onto frames 1, 2, respectively. For each frame we need to find a set of image points u_{1j}^{*}, u_{2j}^{*} where $X \in R3$ is the 3D environment points and proj is the projection function that projects 3D triangulated points from current observation (X, t) in the map of SLAM to the 2D frame plane 1 and 2 as illustrated in Fig. 1. The pose P_i is the pose of frame i. It includes both orientation and translation of frame i. In this optimization problem, u_{1j}, u_{2j} are considered to be the measurements that have uncertainty given by covariance matrix Σ_{uij}. Both u_{1j} and u_{2j} are 2-D vector that represents the location of a detected keypoint (image feature) inside the image plane and it has the units of pixels. Its 2×2 uncertainty matrix Σ_{uij} is taken as unity matrix which means that the uncertainty is taken as 1 pixel as given in the literature. We differentitate Eq. 14 twice with respect to camera pose (translation vector and rotation matrix), we get:

$$\frac{\partial^2 E}{\partial T \partial T} \cong 2\sum_{j=1}^{N} \left(\frac{\partial u_{1j}^{*}}{\partial T}\right)^{T} \left(\frac{\partial u_{1j}^{*}}{\partial T}\right) + 2\sum_{j=1}^{N} \left(\frac{\partial u_{2j}^{*}}{\partial T}\right)^{T} \left(\frac{\partial u_{2j}^{*}}{\partial T}\right) \quad (15)$$

We can use:

$$\sum_{T}^{-1} = H \cong H2J_1^{T}J_1 + 2J_2^{T}J_2$$

where, J_1, J_2 are the Jacobins of u_{1j}^{*}, u_{2j}^{*} w.r.t camera pose. Thus, computing the error vector of reprojection in camera of point p and its Jacobians with respect to the camera pose results in a matrix of dimensions 2×6.

RESULTS AND DISCUSSION

Experimental simulations and results evaluation: In this section we evaluate the accuracy and speed of our system, by the comparison our designed system with the stereo version of ORB-SLAM and LSD SLAM system. For different datasets like KITTI dataset, TUM-RGBD dataset (Sturm *et al*., 2012) and EuroC dataset (Burri *et al*., 2016). In Fig. 5, we show output of our system which consists of robot trajectory and point map. There are evaluation tools to compute error metrics of the robot estimated trajectory, to evaluate the quality and accuracy of our SLAM system. The error metric doesn't assess the map accuracy but can assess localization accuracy. One of the most important types of errors was mentioned in the literature for evaluating the SLAM problem is Relative Pose Error (RPE) (Kummerle *et al*., 2009) that is helpful for graph-based SLAM. We evaluate it in the TUM-RGBD dataset and EuroC dataset. A more intuitive direction is to estimate the Absolute Trajectory Error (ATE) (Sturm *et al*., 2012) after mapping the two trajectories: the ground truth and the estimated path. We evaluate it in the KITTI dataset.

The Relative Pose Error (RPE) which is estimated by calculating the translational and rotational difference between estimated robot poses $P = \{p_1, p_2, p_3, ..., p_N\}$ and ground truth robot poses $\hat{P} = \{\hat{p}_1, \hat{p}_2, \hat{p}_3, ..., \hat{p}_N\}$ as follows:

$$RPE(\hat{P}; P) = \frac{1}{N}\Sigma_i \text{trans}(\hat{p}_i - p_i)^2 + \text{rot}(\hat{p}_i - p_i)^2 \quad (16)$$

where, N is the number of estimated relative poses. The unit of RPE is m^2, rad^2. The Absolute Trajectory Error (ATE) which is estimated by calculating the translational difference between estimated robot poses $P = \{p_1, p_2, p_3, ..., p_N\}$ and $\hat{P} = \{\hat{p}_1, \hat{p}_2, \hat{p}_3, ..., \hat{p}_N\}$ ground truth robot poses which defined as the squared Euclidean distance between corresponding poses as follows:

$$ATE(\hat{P}; P) = \frac{1}{N}\Sigma_i \text{trans}(\hat{p}_i - p_i)^2 \quad (17)$$

where, N is the number of estimated relative poses and the unit of ATE is m^2. To compute RMSE, we take the squared root of ATE error as follows:

$$RMSE(\hat{P}; P) = \sqrt{ATE} \quad (18)$$

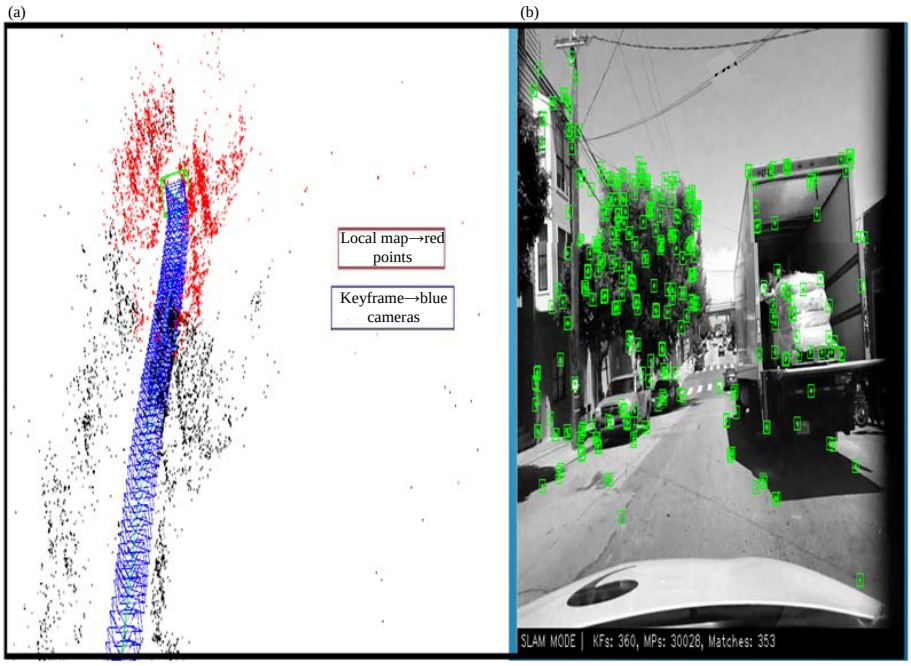

Fig. 5 (a-b): (a) Local maps (red points) and keyframes (blue cameras) mapping for our system and (b) Robot tracking (poses) and feature matching (green rectangles)

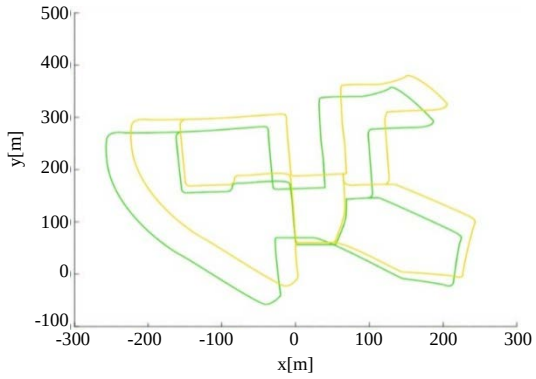

Fig. 6: Robot trajectory for our system (orange line) and ground truth (green line) in case of dataset KITTI_0

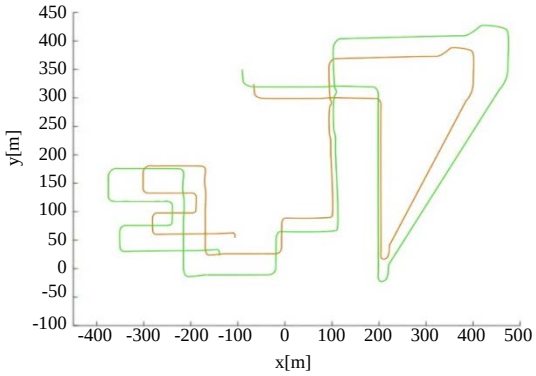

Fig. 7: Robot trajectory for our system (green line) and ground truth (red line) in case of dataset KITTI_01 (150 frames only)

The unit of ATE is m. We provide results of the trajectories and maps estimated by our proposal and ORB-SLAM system for KITTI dataset. First, we have simulated our vSLAM system on the KITTI dataset (Geiger *et al*., 2012) which is recorded by a stereo camera at 20 fps and a resolution 512×382 saved in lossless png format. We use the 11 sequences that is recorded for a car driven around a residential area with accurate ground truth from GPS and a Velodyne laser scanner. The ground truth/estimated trajectory of the robot is shown in Fig. 6-8 for a robot moving in sequence 00 and sequence 01 (150 frames only and sequence 08 of KITTI database (which

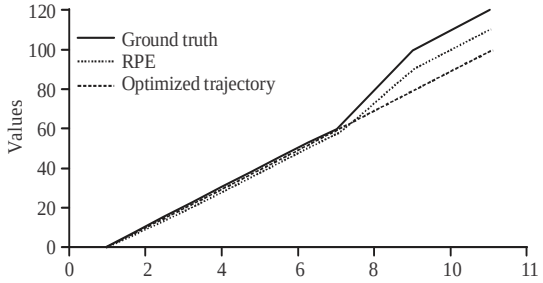

Fig. 8: Robot trajectory for our system (orange line), ground truth (blue line) and RPE error in case of dataset KITTI_01 (150 frames only)

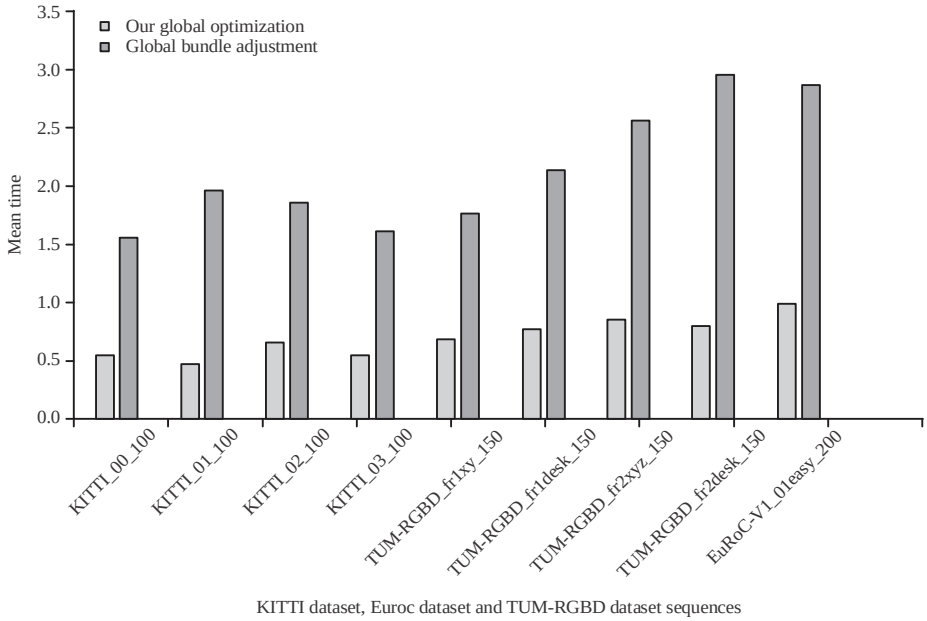

Fig. 9: Synthetic evaluation of speed performance of our system (blue lines) compared to the well-known BA algorithm (orange lines).

show the loop closure). We show that our system can detect loops in the previous figures. The RMSE was estimated for KITTI dataset for all sequences. As shown in the figures, the proposed vSLAM produced an accutate camera tarjectory which is roughly aligned with the trajectory of ground truth.We estimate the performance of our proposed graph optimization algorithm of SLAM system w.r.t well-known nonlinear optimization algorithm which is called global Bundle Adjustment (BA) used in ORB SLAM system (Mur-Artal and Tardos, 2017). We note that the speed and efficiency of our pose graph optimization is higher than that of global BA of ORB SLAM system because it runs in real time faster (3X) than global BA as shown in Fig. 9.

CONCLUSION

Our graph-based SLAM can compute the translation vector, scale and rotation matrix individually. We show that our approach optimization method can provide reliable and promoted results based on stereo camera compared to the newly state-of-the-art SLAM systems specially in accuracy and in dealing with incorrect loops for any general scene planar or non-planar. We utilize a stereo camera which minimizes scale drifts, initialization failures and integrate it with a linear optimization algorithm can provide one accurate solution in contrast to the well-knownnonlinear global optimization BA of stereo ORB-SLAM which provides multiple non-accurate solutions for optimization. Furthermore, our stereo system runs in real time faster (3 X) than the well-known systems in the literature.

RECOMMENDATIONS

In future work, we can enhance tracking performance by using minimization of the photometric residual corresponding to photometric intensity observation model and including uncertainty of this error in our optimization. We conclude that, we can enhance SLAM accuracy in another future work using IMU sensor device, the IMU will aid in the recovery of ambiguous scale provided that the motion dynamics is rich enough. The IMU can also enhance the short-time estimates for pose, since, it usually has a higher sampling rate than the camera (upto 1000 sample/s compared to the common 30 frame/s of the camera). There is no constraints on the combination the sensors used IMU, laser range finder and GPS all together.

ACKNOWLEDGEMENT

We thank Dr. Mohamed Hamdy for a lot of helps and discussions. The work was supported by National Research Centre and Siemens Research Cooperation.

REFERENCES

Bailey, T. and H. Durrant-Whyte, 2006. Simultaneous localisation and mapping (SLAM): Part II. State of the art. Robot. Autom. Magaz., 13: 108-117.

Barooah, P. and J.P. Hespanha, 2007. Estimation on graphs from relative measurements. IEEE. Control Syst., 27: 57-74.

Burri, M., J. Nikolic, P. Gohl, T. Schneider and J. Rehder *et al*., 2016. The EuRoC micro aerial vehicle datasets. Intl. J. Rob. Res., 35: 1157-1163.

Civera, J., A.J. Davison and J.M.M. Montiel, 2007. Inverse depth to depth conversion for monocular SLAM. Proceedings of the 2007 IEEE International Conference on Robotics and Automation, April 10-14, 2007, IEEE, Roma, Italy, ISBN:1-4244-0601-3, pp: 2778-2783.

Davison, A.J., I.D. Reid, N.D. Molton and O. Stasse, 2007. Mono SLAM: Real-time single camera SLAM. Patt. Anal. Mach. Intell. IEEE Trans., 29: 1052-1067.

Engel, J., J. Sturm and D. Cremers, 2013. Semi-dense visual odometry for a monocular camera. Proceedings of the IEEE International Conference on Computer Vision, December 1-8, 2013, Sydney, Australia, pp: 1449-1456.

Geiger, A., P. Lenz and R. Urtasun, 2012. Are we ready for autonomous driving? the KITTI vision benchmark suite. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 16-21, 2012, IEEE, Providence, Rhode Island, USA., ISBN:978-1-4673-1226-4, pp: 3354-3361.

Hesch, J.A., D.G. Kottas, S.L. Bowman and S.I. Roumeliotis, 2012. Observability-constrained vision-aided inertial navigation. Master Thesis, University of Minnesota, Minneapolis, Minnesota, USA.

Jones, E.S. and S. Soatto, 2011. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. Intl. J. Rob. Res., 30: 407-430.

Klein, G. and D. Murray, 2007. Parallel tracking and mapping for small AR workspaces. Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2007, November 13-16, 2007, IEEE, Nara, Japan, ISBN:978-1-4244-1749-0, pp: 225-234.

Kneip, L. and S. Lynen, 2013. Direct optimization of frame-to-frame rotation. Proceedings of the IEEE International Conference on Computer Vision, December 1-8, 2013, IEEE, Sydney, Australia, pp: 2352-2359.

Kneip, L., R. Siegwart and M. Pollefeys, 2012. Finding the exact rotation between two images independently of the translation. Proceedings of the European Conference on Computer Vision, October 8-16, 2016, Springer, Berlin, Germany, ISBN:978-3-642-33782-6, pp: 696-709.

Kummerle, R., B. Steder, C. Dornhege, M. Ruhnke and G. Grisetti *et al*., 2009. On measuring the accuracy of SLAM algorithms. Auton. Rob., 27: 387-407.

Li, M. and A.I. Mourikis, 2013. High-precision, consistent EKF-based visual-inertial odometry. Intl. J. Rob. Res., 32: 690-711.

Lynen, S., M.W. Achtelik, S. Weiss, M. Chli and R. Siegwart, 2013. A robust and modular multi-sensor fusion approach applied to MAV navigation. Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), November 3-7, 2013, IEEE, Tokyo, Japan, pp: 3923-3929.

Merzban, M.H., M. Abdellatif, H. Abbas and S. Sessa, 2013. Toward multi-stage decoupled visual SLAM system. Proceedings of the 2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE), October 21-23, 2013, IEEE, Washington, USA., pp: 172-177.

Mur-Artal, R. and J.D. Tardos, 2017. ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE. Trans. Rob., 33: 1255-1262.

Mur-Artal, R., J.M.M. Montiel and J.D. Tardos, 2015. ORB-SLAM: A versatile and accurate monocular SLAM system. IEEE. Trans. Rob., 31: 1147-1163.

Rublee, E., V. Rabaud, K. Konolige and B. Gary, 2011. ORB: An efficient alternative to SIFT or SURF. Proceedings of the IEEE International Conference on Computer Vision (ICCV), November 6-13, 2011, IEEE, Barcelona, Spain, ISBN:978-1-4577-1101-5, pp: 2564-2571.

Sturm, J., N. Engelhard, F. Endres, W. Burgard and D. Cremers, 2012. A benchmark for the evaluation of RGB-D SLAM systems. Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 7-12, 2012, IEEE, Vilamoura, Portugal, ISBN:978-1-4673-1737-5, pp: 573-580.

Yin, J., L. Carlone, S. Rosa and B. Bona, 2014. Graph-based robust localization and mapping for autonomous mobile robotic navigation. Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation (ICMA), August 3-6, 2014, IEEE, Tianjin, China, ISBN:978-1-4799-3978-7, pp: 1680-1685.