Large-Scale Monocular SLAM by Local Bundle Adjustment and Map Joining

Liang Zhao[1,2], Shoudong Huang[1], Lei Yan[2], Jack Jianguo Wang[1], Gibson Hu[1], Gamini Dissanayake[1]
1: ARC Centre of Excellence for Autonomous Systems, Faculty of Engineering and Information Technology
University of Technology, Sydney, PO Box 123 Broadway, Ultimo, NSW 2007, Australia,
Email: {liang.zhao,sdhuang,jwang, gibson.hu,gdissa}@eng.uts.edu.au
2: Spatial Information Integration and Its Applications Beijing Key Laboratory, Institute of Remote Sensing and GIS
School of Earth and Space Science, Peking University, 5 Yiheyuan Road, Haidian District, Beijing China, 100871,
Email: LYan@pku.edu.cn

***Abstract*—This paper first demonstrates an interesting property of bundle adjustment (BA), "scale drift correction" property. Here "scale drift correction" means that BA can converge to the correct solution (up to a scale) even if the initial values of the camera pose translations and point feature positions are calculated using different scale factors. This property together with other properties of BA makes BA the best approach for monocular SLAM when no camera motion information is available, although the computational cost of BA is an issue.**

This naturally leads to the idea of using local BA and map joining to solve large-scale monocular SLAM problem, which is proposed in this paper. The local maps are built through Scale-Invariant Transform Feature (SIFT) detector and matching, random sample consensus paradigm (RANSAC) at different levels for robust outlier removal, and BA for optimization. To reduce the computational cost of the large-scale map building, the features in each local map are properly selected and then the local maps are combined using a recently developed 3D map joining algorithm. The proposed large-scale monocular SLAM algorithm is evaluated using a publicly available dataset. It is shown that the camera poses estimate is very accurate as compared with the ground truth provided.

***Keywords—Visual SLAM*, map joining, bundle adjustment**

I. Introduction

Simultaneous localization and mapping (SLAM) is the problem where a mobile robot needs to build a map of its environments and simultaneously use the map to locate itself. When the only sensor equipped onboard the robot is a single camera, the problem is called monocular SLAM. Monocular SLAM problem is very challenging due to the lack of information on camera motion, the unavailability of depth information from the single image, as well as the unobservable scale factor [1][2].

Recently, a number of monocular SLAM algorithms have been developed. Typically, Extended Kalman Filter (EKF) is used to estimate the camera pose as well as 3D feature positions. The inverse-depth parameterization has been shown to be useful to avoid the lack of depth information in monocular SLAM [1], especially for far-away features. However, the EKF prediction step requires a relatively accurate motion model of camera poses due to the linearization process involved. Thus a constant velocity camera motion model is normally used [1][2]. Although it was claimed that "by choosing appropriate values for the initial velocities and the covariance of the process noise, the EKF-SLAM is able to obtain an approximate scale for the map" [2], some problems may happen when the camera motion is irregular [3]. Very recently, interacting multiple model monocular SLAM is proposed [3] where different motion model can be applied at different situations. However, some kinds of motion models still need to be assumed and the switching between the models is non-trivial.

Moreover, the potential estimation inconsistency of EKF SLAM algorithms has been demonstrated in 2D case where the major cause of the inconsistency is from the robot orientation error [4][5]. The fundamental reason for the potential inconsistency is due to the fact that the Jacobian of observation/odometry functions with respect to a feature/pose gets evaluated at different feature/pose location estimates, resulting in the flow of incorrect information to the estimation process [6]. It can be imagined that the potential inconsistency involved in 3D EKF SLAM is stronger than that of 2D EKF SLAM since three orientation angles instead of one are involved in the robot/camera poses.

Bundle adjustment (BA) from multi-view geometry of computer vision [7] completely avoids the use of a camera motion model. This makes it suitable for structure from motion problem when an arbitrary sequence of images is used. Moreover, it is well known that BA can provide the optimal solution by performing a least squares optimization, which also avoids the cause of potential estimation inconsistency. The only problem of BA is the computational cost that prevents the real-time application for large-scale problems.

Local submap joining is an efficient strategy for solving large-scale SLAM problems [2][8][9]. The idea is to build local maps using local information and then combine the local maps into a global map. Thanks to the sparseness nature of the map joining problem [2][9], the map joining process can now be made very efficient. Local map joining has been applied to monocular SLAM in [8], where conditionally independent local maps are built and then carefully combined together to avoid information reuse. However, the local maps are still built by EKF and a constant velocity camera motion model is assumed.

Since BA can provide optimal solution without the need of a camera motion model, why not use BA for small-scale local map building (where computational cost is not a problem) and them combine them using map joining strategy? This motivates the research work in this paper.

In this paper, we further investigate the BA algorithm through simulations and real experiments and find an interesting property, namely, the "scale drift correction" property. That is, BA can converge to the correct solution (up to a scale) even if the initial values of the camera pose translations and point feature positions are decided by different scale factors.

Furthermore, we demonstrate how local maps built by BA can be joined together using our recently developed 3D map joining algorithm, Iterated Sparse Local Map Joining Filter (I-SLSJF). We use the Málaga 2009 Robotic Dataset [10] to test the proposed large-scale monocular SLAM algorithm and show that the estimated camera poses are very close to the ground truth provided by the dataset.

The paper is organized as follows. Section II states the large-scale monocular SLAM problem considered in this paper and outlines the proposed approach. Section III discusses the pros and cons of BA algorithm. Section IV details the process of local map building and Section V explains the map joining process. In Section VI, some simulation and experimental results are provided. Finally Section VII concludes the paper.

II. Large-Scale Monocular SLAM

This section explains the large-scale monocular SLAM problem considered in this paper and outlines the proposed approach.

A. *Momocular SLAM problem*

The monocular SLAM problem considered in this paper is to use a sequence of images to estimate the camera poses as well as the 3D position of extracted point features. All the camera poses and feature positions are with respect to the coordinate system decided by the first camera pose. The translation of the camera poses and the position of point features are up to a scale.

We assume the camera is moving freely in 6D and there is no information on the camera motion available. However, we do assume that there is enough overlap between two adjacent images such that the relative camera poses can be determined (up to a scale). We also assume that the camera is calibrated and the calibration parameters are available.

B. *Proposed Approach*

In this paper, we propose to use SIFT to find the features and match them in the images, then use RANSAC for outlier removal. We then use BA to build local maps, each local map uses a small number of images. Finally, the local maps will be joined together using 3D I-SLSJF.

The main reason that we use BA instead of EKF for the local map building is that BA provides the optimal solution and does not need a camera motion model. Since the local maps are small, the computational cost is not a problem.

In the next section, we will examine some details of the properties of BA.

III. Pros and Cons of Bundle Adjustment

Once the point features are selected and matched, the optimal solution can be achieved by performing BA, an optimization process to find the best camera poses and the feature positions by minimizing the re-projection errors.

Bundle adjustment constitutes a large, nonlinear least-squares problem that is often solved as the last step of feature-based structure and motion estimation in computer vision algorithms to obtain optimal estimates. Due to the very large number of parameters involved, a general purpose least squares algorithm incurs high computational and memory storage costs when applied to BA. Fortunately, the lack of interaction among certain subgroups of parameters results in the corresponding Jacobian being sparse, a fact that can be exploited to achieve considerable computational savings.

An initial value of the camera poses and 3D feature positions, together with the feature positions 2D location in each images and the camera calibration parameters need to be provided as the input of BA. The output of BA is the optimized camera poses and feature positions.

A. *Cons of BA*

The only disadvantage of BA is the computational cost. However, due to the increased computer power, running BA with tens of frames in real-time is now achievable [11]. Thus BA can be used in local map building with no problem.

B. *Pros of BA*

There are a number of advantages using BA. The first is that BA can provide the optimal solution based on the information available. Moreover, BA is more robust to outliers [11]. Furthermore, as an optimization algorithm, BA avoids the potential estimate inconsistency as compared with filter based SLAM [6][12].

C. *Scale drift correction property of BA*

Apart from the well-known advantages of BA, we also notice another key advantage of BA algorithm, called 'scale drift correction" in this paper, which is crucial for monocular SLAM problem. When converges, BA will make the translations of all the poses up to one scale. Because if the scales is not the same for the translations of different poses, the 3D positions of the same common features will not be the same during triangulation from different pairs of poses. On the other hand, if we use the poses with different translation scales, the projective positions points in the images of the same features will not be the same. The BA will adjust this error during the Levenberg-Marquardt algorithm. This property is demonstrated using both simulation and real images in the results in Section VI-A and Section VI-D.

Since scale drift is a major issue in monocular SLAM, this scale drift correction property makes BA the best candidate for solving the monocular SLAM problem.

IV. Local map building

The local map building process involves feature selection and matching, outlier removal, relative pose computation, feature position calculation, and bundle adjustment.

A. Feature selection and matching

SIFT descriptor has been used to select and match the features in the images. The SIFT detector extracts from an image a collection of frames or keypoints [13]. These are oriented disks attached to blob-alike structures of the image. As the image translates, rotates and scales, the frames track these blobs and thus the deformation. By canonization, i.e. by mapping the frames to a reference (a canonical disk), the effect of such deformation on the feature appearance is removed.

The SIFT descriptor is a coarse description of the edge found in the frame. Due to canonization, descriptors are invariant to translations, rotations and scaling and are designed to be robust to residual small distortions.

Once frames and descriptors of two images have been computed, we can estimate the pairs of matching features by using Lowe's method to discard ambiguous matches [13].

B. Multi-level RANSAC for outlier removal

An effective robust algorithm for processing noisy data with outliers is the random sample consensus paradigm (RANSAC) [14]. Given that a large proportion the data may be useless, RANSAC is the opposite approach of conventional smoothing techniques. Rather than using as much data as possible to obtain an initial solution and then attempting to identify outliers, as small a subset of the data as is feasible to estimate the parameters used (e.g. two point subsets for a line, seven correspondences for a fundamental matrix), and this process is repeated enough times on different subsets to ensure that there is a 95% chance that one of the subsets will contain only good data points. The best solution is that which maximizes the number of points whose residual is below a threshold. Once outliers are removed the set of points identified as non-outliers may be combined to give a final solution.

Use of the RANSAC method to estimate the epipolar geometry was first reported in Torr and Murray [15]. A brief summary of random sampling algorithm are as follows:

1. Repeat for m samplings:

(a) Select a random sample of the minimum number of data points to make a parameter estimate the fundamental matrix (F).

(b) Calculate the distance from each feature to the epipolar lines of F.

(c) In the case of the RANSAC estimator calculate the number of inliers consistent with F. In the case of LMS calculate the median error.

2. Select the best solution i.e. the biggest consistent data set. In the case of ties select the solution which has the lowest standard deviation of inlying residuals.

3. Re-estimate the parameters using all the data that has been identified as consistent, a different more computationally expensive estimator may be used at this point e.g. Powell's method.

We always compute the Essential Matrix from the inliers after RANSAC with two different singular values using no restriction algorithm just like the eight point algorithm which is described in the next. It is mainly because of the errors of the matching features, not only the mismatch, but also the location of features in the images. To solve the problem we have to make the threshold of the distance of features to the epipolar lines of F much smaller to make sure the inlier matching features are not mismatched and have less location errors. But small threshold will bring the wrong result because RANSAC is a random sample algorithm. So a multi-level RANSAC has been used in this paper to insure the right result and high precision of the features in location. Run RANSAC with thresholds as 2, 0.5, 0.1, 0.05 by steps and it can remove almost all the outlier even such as the features on the moving leaves and cars which is much worse to estimate the Essential Matrix and relative pose. And the two singular valves of he Essential Matrix is nearly the same with 0.01% difference. And because of removing bad features by steps, the number of samples can be reduced and it also gets very good result, the computation cost will not be more than one step RANSAC.

The outcomes of the RANSAC are the correct matching features of each pair of images and the fundamental matrix F.

C. Relative pose computation

Here we do not use the fundamental matrix obtained from RANSAC, but only the inlier features selected by RANSAC, because the least square solution is more accuracy. The two singular values are always not the same using 8 point algorithm to compute the Essential Matrix. In this paper we prefer 8 point algorithm because it is much easier and less computation cost than the 5 point algorithm and after RANSAC by steps, the point matches are precise enough to get two singular values of the Essential Matrix nearly the same with 0.01% difference. This will not bring error to the relative pose.

The eight point algorithm [7] is used for computing the relative pose between two camera poses. Below are the details.

The fundamental matrix is defined by the equation

$$x_i^{'} F x_i = 0 \qquad (1)$$

for any pair of matching points $x_i \leftrightarrow x_i^{'}$ in two images. Given sufficiently many point matches $x_i \leftrightarrow x_i^{'}$ (at least 7), equation (1) can be used to compute the unknown matrix F. In particular, writing $x = (x, y, 1)^T$ and $x' = (x', y', 1)^T$ each point match gives rise to one linear equation in the unknown entries of F. The coefficients of this equation are easily written in terms of the known coordinates x and x'. Specifically, the equation corresponding to a pair of points $(x, y, 1)$ and $(x', y', 1)$ is

$$(x'xf_{11} + x'yf_{12} + x'f_{13} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33}) = 0 \quad (2)$$

Denote by f the 9-vector made up of the entries of F in row-major order. Then (2) can be expressed as a vector inner product

$$(x'x + x'y, x', y'y, y', x, y, 1)\, f = 0 \qquad (3)$$

From a set of n point matches, we obtain a set of linear equations of the form

$$Af = \begin{bmatrix} x_1^{'}x_1 & x_1^{'}y_1 & x_1^{'} & y_1^{'}x_1 & y_1^{'}y_1 & y_1^{'} & x_1 & y_1 & 1 \\ & & \vdots & & & \vdots & & & \\ x_n^{'}x_n & x_n^{'}y_n & x_n^{'} & y_n^{'}x_n & y_n^{'}y_n & y_n^{'} & x_n & y_n & 1 \end{bmatrix} f = 0 \qquad (4)$$

Then the normalized 8-point algorithm [7] is described as follows:

1. Normalization: Transform the image coordinates according to $\hat{x}_i = Tx_i$ and $\hat{x}'_i = T'x'_i$, where T and T' are normalizing transformations consisting of a translation and scaling.

2. Find the fundamental matrix $\hat{F}'$ corresponding to the matches $\hat{x}_i \leftrightarrow \hat{x}_i^{'}$ by

(a) Linear solution: Determine $\hat{F}$ from the singular vector corresponding to the smallest singular value of $\hat{A}$, where $\hat{A}$ is composed from the matches $\hat{x}_i \leftrightarrow \hat{x}_i^{'}$ as defined in (4).

(b) Constraint enforcement: Replace $\hat{F}$ by $\hat{F}'$ such that $\det \hat{F}' = 0$ using the SVD

3. Demoralization: Set $F = T'^{T}\, \hat{F}'T$. Matrix F is the fundamental matrix corresponding to the original data $x_i \leftrightarrow x_i^{'}$.

The relationship between the fundamental and essential matrices is

$$E = K^T F K \qquad (5)$$

For a given essential matrix $E = U diag(1,1,0) V^T$, and first camera matrix $P = [I \mid 0]$, there are four possible choices for the second camera matrix P', namely
$P' = [UWV^T \mid +u_3]$ or $[UWV^T \mid -u_3]$ or $[UW^TV^T \mid +u_3]$ or $[UW^TV^T \mid -u_3]$,

where $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ (6)

Then we can get the relative poses of each pair of images from

$$P' = R[I \mid T] \qquad (7)$$

D. Feature position calculation

In this paper we use the triangulation method [7] to compute the position of a point in 3D-space given its image in two views and the camera matrices of those views. In each image we have a measurement x = PX, x' = P'X, and these equations can be combined into a form AX = 0, which is an equation linear in X.

First the homogeneous scale factor is eliminated by a cross product to give three equations for each image point, of which two are linearly independent. For example for the first image, $x \times (PX) = 0$ and writing this out gives

$$\begin{aligned} x(p^{3T}X) - (p^{1T}X) &= 0 \\ y(p^{3T}X) - (p^{2T}X) &= 0 \\ x(p^{2T}X) - y(p^{1T}X) &= 0 \end{aligned} \qquad (8)$$

where p^{iT} are the rows of P. These equations are linear in the components of X.

An equation of the form AX = 0 can then be composed, with

$$A = \begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p'^{3T} - p'^{1T} \\ yp'^{3T} - p'^{2T} \end{bmatrix} \qquad (9)$$

where two equations have been included from each image, giving a total of four equations in four homogeneous unknowns. This is a redundant set of equations, since the solution is determined only up to scale. One way of solving the set of equations of the form AX = 0 is to find the solution as the unit singular vector corresponding to the smallest singular value of A.

Obtain the SVD of A. The unit singular vector corresponding to the smallest singular value is the solution X. Specifically, if A = UDV^T with D diagonal with positive diagonal entries, arranged in descending order down the diagonal, then X is the last column of V.

E. Bundle adjustment

In this paper we use SBA (Sparse Bundle Adjustment)[18], a publicly available C/C++ software package for realizing generic bundle adjustment with high efficiency and flexibility regarding parameterization [10].

When building local maps, we using BA not only make the translations of all the poses up to one scale, but also optimize the local maps. Now every local map is up to one scale, and it will not cost too much computation because there are limited number of poses and features in each local map.

F. Information matrix computaion

To join the local maps together using 3D I-SLSJF [17], the information matrix of the local map is needed.

The SBA software package does not provide the information matrix of the local map estimate. However, the information matrix can be easily computed once the BA converges. We use the standard least square approach to compute the information matrix. The Jacobians are computed using the result from BA.

V. Map joining by 3D I-SLSJF

This section explains how to join the local maps built by BA to get the global map. The map joining algorithm we used is the 3D I-SLSJF [17].

A. 3D I-SLSJF algorithm

The 3D I-SLSJF algorithm is an extension of the 2D I-SLSJF (the MATLAB source code of 2D I-SLSJF is available on OpenSLAM website). The algorithm uses extended Information Filter (EIF) to fuse the local maps in sequence and performs a linearized least squares to improve the quality of the map whenever necessary. This approach is computationally more efficient than the typical maximum likelihood method and also shows better accuracy compared with 3D EKF [17]. Because the algorithm exploit the exact sparseness of the map joining process, it is computationally efficient. Since the algorithm itself is incremental in nature, it can be easily implemented to a real time system.

B. Local map feature selection

We divide all the images into different local maps, and then use sparse bundle adjustment to make local maps optimization. Because of less poses and features in each local map, the SBA will converge very fast with good result.

How to select features from each local map is very important because we cannot use them all due to the computational complexity. In this paper we use two ways to select the feature in each local map:

1. In the 3D I-SLSJF algorithms, the common features in different local maps are very important to join the local maps and optimize. But the features only in one local map are useless because the local map has been optimized by the sparse bundle adjustment.

This will delete nearly about more than 90% of the features, which will reduce the computation cost greatly.

2. In multiple view geometry and computer vision, when camera moves towards straightly, the features near the principle point are not good and with large uncertainty in the depth direction because the angle of the two projective lines of the feature is too small and will bring large error of the 3D position when doing triangulation.

This will delete about 5% of the features. This is not very useful to the computational cost but will make the result much more accurate.

C. Relative scales estimation between local maps

Although every local map is up to one scale as shown in Section III.C, the scales between different local maps might be different. In this paper we use the common features to estimate relative scales between local maps.

There are a lot of common features between local maps. And using the distance between two common features in different local maps, we can easily estimate the relative scales between local maps. A random two-point selector has been used to make the estimation more accuracy. Because scales are multiple factors, we use their log to get the arithmetical mean. Compare with less common features only in two pairs of poses, there are more common features and they are from different poses. So this method reduces most of the scale drift and makes the relative scales estimation much more accurate at the same time.

VI. Result

Some simulation and experimental results are presented in this section to support the claims in this paper.

A. Scale drift correction property of BA

During our work, we use the simulation data first to test if BA can adjust the scales. We simulate 23 poses trajectory as a circle with 392 features. The simulation environment is shown in Figure 1. The red points are the 3D features and the yellow triangles are the poses of the robot. First we make the relative translation and 3D feature positions of each pose multiply a scale factor. The scale factors of the two simulations are randomly between 0.5 to 1.5 and 0 to 2 shown in table 1. The trajectory is shown in Figure 2 as the black lines with circles. Then we do the BA and the trajectory has return back to circle as the red lines with stars. After BA, all the translations and 3D feature positions of the poses are up to one scale.

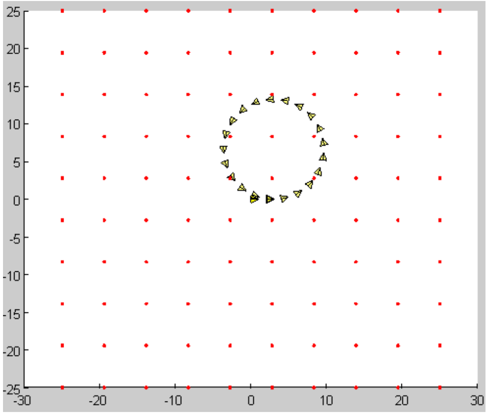
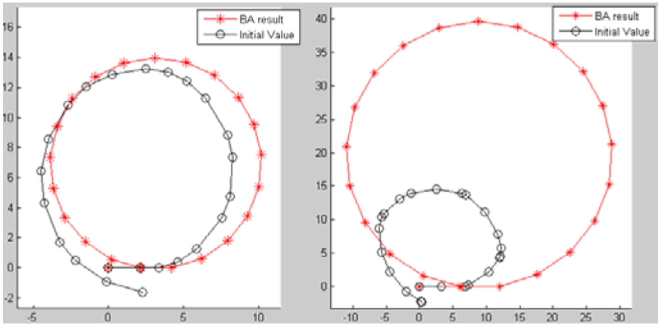


Figure 1. Simulation environment: Features and poses in XY coordinate

Table 1 Random scale factors

Scale factors (0.5-1.5)							
1	2	3	4	5	6	7	8
1.047	0.638	0.649	0.757	1.340	0.754	1.314	0.743
9	10	11	12	13	14	15	16
1.429	0.849	0.696	0.751	1.116	0.973	0.851	1.330
17	18	19	20	21	22		
1.085	1.049	1.417	0.785	1.257	1.253		
Scale factors (0-2)							
1	2	3	4	5	6	7	8
1.629	1.811	0.253	1.826	1.264	0.195	0.556	1.093
9	10	11	12	13	14	15	16
1.915	1.929	0.315	1.941	1.914	0.970	1.600	0.283
17	18	19	20	21	22		
0.843	1.831	1.584	1.918	1.311	0.071		



(a) Scale factors (0.5-1.5) (b) Scale factors (0-2)

Figure 2. The simulation result of BA Scale drift correction

B. *Dataset*

In this paper we use the Málaga 2009 Robotic Dataset Collection PARKING-6L [10]. This dataset was collected at the parking of the Computer Science building of the University of Málaga (Spain) using an electric car equipped with 3 SICK and 2 Hokuyo laser scanners, 2 Firewire color cameras, one xSens IMU, three RTK GPS receivers and one consumer-grade USB GPS receiver. As described in the paper, a centimeter-level ground truth is robustly computed from the RTK data, thus making the dataset an ideal testbed for SLAM or localization techniques.

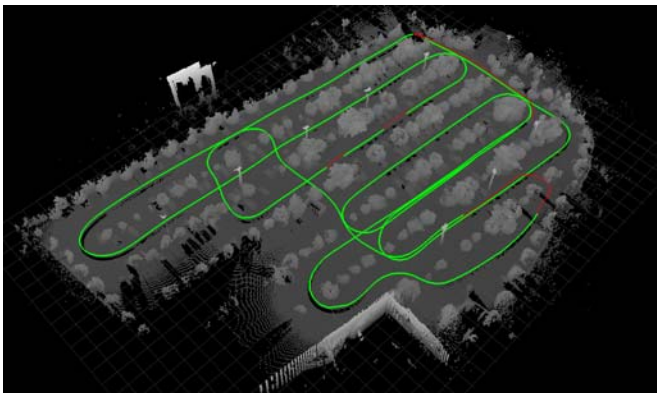


Figure 3. Overview of the 3D point-cloud

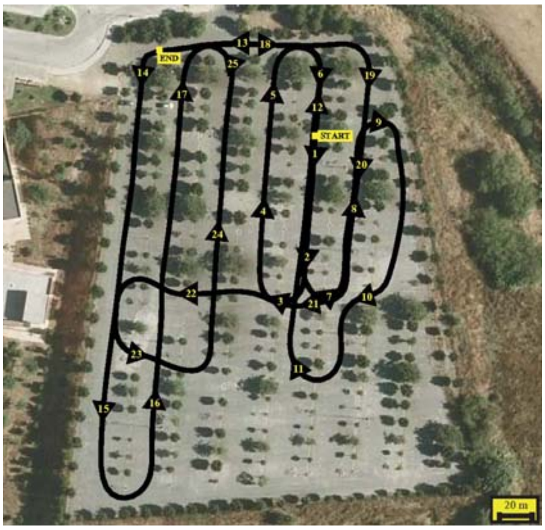
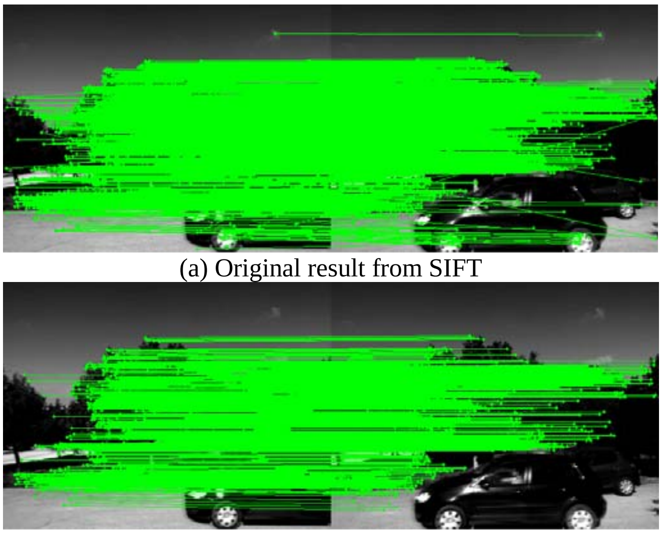


Figure 4. Vehicle trajectory

We use one close loop images grabbed by the right camera during the whole dataset duration, rectified to compensate the camera distortion. The original framerate is 7.5Hz. In this paper we use 2.5 Hz as the framerate to select 170 images for the monocular SLAM. The estimated camera calibration parameters can be found in the dataset. The path ground truth for the right camera, i.e. the vehicle poses plus the right camera location on the vehicle using the appropriate 6D pose composition also has been used in this paper to check the accuracy of the SLAM result.

C. *Result of SIFT and Multi-level RANSAC*

The feature matching result is shown in Figure 5. The upper one is the original result from SIFT, and the lower one is the result after RANSAC with thresholds as 2, 0.5, 0.1, 0.05 by steps. It is obvious that the mismatch features and features on the moving cloud and cars have been removed after Multi-level RANSAC.



(a) Original result from SIFT

(B) Result of Multi-level RANSAC

Figure 5. Feature matching result using SIFT and Multi-level RANSAC

And the estimation of Essential Matrix with these matching features using the 8 point algorithm also get the good result with nearly the same two singular values with 0.01% difference, which is not bad compared with the 5 point algorithm but much more easier and less computation cost. Some singular values of the Essential Matrices for different pair of images are shown in Table 2.

Table 2. Singular values of some Essential Matrices

	1	2	3	4
SV1	20.4492	19.2341	10.8472	17.1941
SV2	20.4438	19.2273	10.8239	17.1828

D. *Result of scale estimation*

Then we use the 170 images real dataset to check the scale estimation result. The visual odometry result without scale factors is shown in Figure 6 as the green line. The result of BA using the whole poses and features without scale factor is shown as the blue line. The result is very good and it is also approved that the BA can make all the translations and 3D feature positions of the poses up to one scale.

At last we divided the images into ten groups and build ten local maps, using the method of scale estimation described in Section V-C, joining the 10 local maps without map joining algorithm, the result is shown as the purple line. The result is also very good with at most 1.8m error.

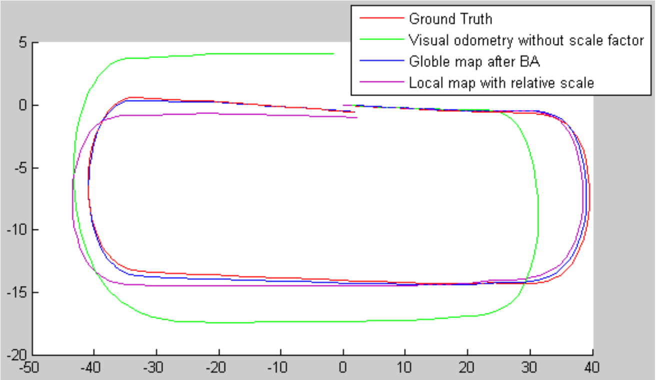


Figure 6. Scale estimation result of the real dataset

E. *Comparison of the 3D I-SLSJF results with the ground truth*

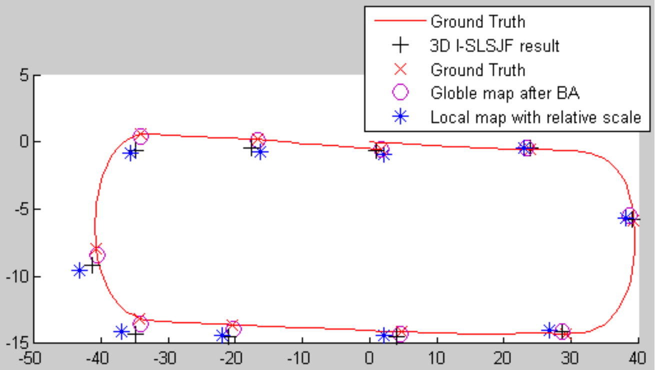


Figure 7. The 3D I-SLSJF result compared with ground truth

The 3D I-SLSJF result is shown in Figure 7 as the black cross. It is better than the one only joining the 10 local maps without map joining algorithm and nearly the same compared with the ground truth with at most 0.8m error. Although the result of BA using the whole poses and features is best, it cost about more than 4 hours to make the BA converge. And using the method described in this paper, the BA of each local map will cost about 100s because of less poses and features; and the 3D I-SLSJF algorithm will take about 628s to join all the local maps together due to the local map feature selection. It can reduce the computation cost greatly and also have very good result.

VII. CONCLUSIONS AND DISCUSSIONS

This paper proposed a map joining algorithm for solving large-scale monocular SLAM problem. The algorithm uses bundle adjustment (BA) to build the local maps and then use 3D Iterated Sparse Local Submap Joining Filter (I-SLSJF) to join the local maps together. The map joining results using the Málaga 2009 Robotic Dataset Collection PARKING-6L dataset demonstrate the accuracy and efficiency of the proposed approach.

One of the key properties of BA is its ability of correcting the scale drift. As far as we know, this paper is the first that point out this interesting property. The “scale drift correction” property is demonstrated in this paper using both simulation and experimental results. Since scale drift is an important issue for monocular SLAM, and BA can provide the optimal and reliable solution without the need of a camera motion model, we believe that BA is the ideal approach for local map building.

The approach in this paper has some similarity with the local bundle adjustment proposed in [17]. The major difference is that our map joining performs a high level optimization on top of the local maps, while the method in [17] does not have theis step. The approach is also similar to FrameSLAM developed by Konolige and Agrawal [8]. The major difference is that in FrameSLAM, only camera poses of key frames are kept for the high level optimization where we also keep some point features to improve the quality of the pose estimation. Furthermore, the results on FrameSLAM were only provided with a sequence of stereo images [8] and whether the approach can be applied to monocular SLAM or not is not very clear. Local map joining strategy is also used in [2] but the local maps were built by EKF and the local maps are not completely independent.

We are in the process of improving the reliability and efficiency of the proposed algorithm, testing it using more datasets, and comparing its performance with other approaches for monocular SLAM. We are also planning to investigate more on the “scale drift correction” property of BA and try to provide a theoretical proof on it.

In the current work, we have not paid too much attention to the selection of images, in the results shown in Section VI, we simply select one out of every three frames. In the future, we will address this key frame selection issue [8][17] and improve the reliability of the proposed algorithm. The map joining algorithm used in this paper is 3D I-SLSJF which is a generic map joining approach. Future work also includes the development of map joining algorithms particularly suitable for monocular SLAM.

REFERENCES

[1] J. Civera, A.J. Davison, J. Montiel, “Inverse Depth Parametrization for Monocular SLAM,” IEEE Transactions on Robitics, vol. 24, Iss. 5, pp. 932-945, October 2008.

[2] P.Pinies, J.D. Tardos, “Large-Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision,” IEEE Transactions on Robitics, vol. 24, Iss. 5, pp. 1094-1106, October 2008.

[3] J. Civera, A.J. Davison, J. M. M. Montiel, “Interacting Multiple Model Monocular SLAM,” IEEE International Conference on Robotics and Automation, ICRA 2008, pp. 3704- 3709, 2008.

[4] T. Bailey, J. Nieto, J. Guivant, M. Stevens, E. Nebot. “Consistency of the EKF-SLAM Algorithm,” Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China. pp. 3562-3568, October 2006.

[5] S. Huang and G. Dissanayake, “Convergence and consistency analysis for Extended Kalman Filter based SLAM," IEEE Transactions on Robotics, vol. 23, Iss. 5, pp. 1036-1049, 2007.

[6] S. Huang, Z. Wang, G. Dissanayake, U. Frese, “Iterated D-SLAM Map Joining: Evaluating its performance in terms of consistency, accuracy and efficiency,” Autonomous Robots, vol. 27, pp. 409-429, 2009.

[7] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed., Cambridge University Press, 2003, pp. 237-323.

[8] K. Konolige, M. Agrawal, “FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping,” IEEE Transactions on Robitics, vol. 24, Iss. 5, pp. 1066-1077, October 2008.

[9] S. Huang, Z. Wang, G. Dissanayake, “Sparse Local Submap Joining Filter for Building Large-Scale Maps,” IEEE Transactions on Robitics, vol. 24, Iss. 5, pp. 1121-1130, October 2008.

[10] J.L. Blanco, F.A. Moreno, J. Gonzalez, “A collection of outdoor robotic datasets with centimeter-accuracy ground truth,” Auton Robot, vol. 27, pp. 327–351, August 2009.

[11] C. Engels, H. Stewnius, D. Nister, “Bundle adjustment rules,” Photogrammetric Computer Vision, September 2006.

[12] F. Dellaert and M. Kaess, “Square root SAM: Simultaneous localization and mapping via square root information smoothing," International Journal of Robotics Research, vol. 25, no. 12, pp. 1181-1203, December 2006.

[13] D. G. Lowe, “Distinctive image features from scaleinvariant keypoints,” International Journal of Computer Vision, vol. 60(2), pp. 91-110, 2004.

[14] M. Pischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography,” Common Assoc. Comp. Mach., vol. 24, pp. 381-95, 1981.

[15] P H. S. Torr and D.W Murray, “Outlier detection and motion segmentation. In P.S. Schenker, editor,” Sensor Fusion Vl, vol. 2059, pp. 432-443, 1993.

[16] Hu, G., Shoudong Huang, Dissanayake, G., “3D I-SLSJF: A Consistent Sparse Local Submap Joining Algorithm for Building Large-Scale 3D Maps,” 48th IEEE Conference on Decision and Control, Shanghai, China, pp. 6040 – 6045, 2009.

[17] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, “Generic and real-time structure from motion using local bundle adjustment,” Image and Vision Computing, vol. 27, Iss. 8, pp. 1178-1193, July 2009.

[18] M. I. A. Lourakis, and A. A. Argyros, SBA: A software package for generic sparse bundle adjustment. ACM Trans. Math. Softw. 36, 1, Article 2 (March 2009), 30 pages. DOI =10.1145/1486525.1486527 http://doi.acm.org/10.1145/1486525.1486527, 2009.

[19] J. Sola, A. Monin, M. Devy, T. Vidal-Calleja, “Fusing Monocular Information in Multicamera SLAM,”, IEEE Transactions on Robitics, vol. 24, Iss. 5, pp. 958-968, October 2008.

[20] L.M. Paz, P. Pinies, J.D. Tardos, J. Neira, “Large-Scale 6-DOF SLAM With Stereo-in-Hand,” IEEE Transactions on Robitics, vol. 24, Iss. 5, pp. 946-957, October 2008.

[21] E. Eade, T. Drummond, “Edge landmarks in monocular SLAM,” Image and Vision Computing, vol. 27, pp. 588–596, 2009.

[22] E. Eade, T. Drummond, “Monocular SLAM as a Graph of Coalesced Observations,” IEEE 11th International Conference on Computer Vision, ICCV 2007, pp. 1-8, 2007.

[23] R. Munguia, A. Grau, “Monocular SLAM for Visual Odometry,” IEEE International Symposium on Intelligent Signal Processing, WISP 2007, pp. 1- 6, 2007.

[24] T. Lemaire, S. Lacroix, “Monocular-vision based SLAM using Line Segments,” IEEE International Conference on Robotics and Automation, pp 2791-2796, 2007.

[25] E. Eade, T. Drummond, “Scalable Monocular SLAM,”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition vol. 1 , pp. 469- 476, 2006.

[26] E. Eade and T. Drummond, “Unified loop closing and recovery for real time monocular slam,” In BMVC, 2008.

[27] D. Nister, “An efficient solution to the five-point relative pose problem,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, Iss. 6, pp. 756- 770, 2004.

[28] H. Li, R. Hartley, “Five-Point Motion Estimation Made Easy,” 18th International Conference on Pattern Recognition, ICPR 2006. vol. 1, pp. 630-633, 2006.

[29] E. Royer, M. Lhuillier, M. Dhome, T. Chateau, “Localization in urban environments: monocular vision compared to a differential GPS sensor,” Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2, pp. 114- 121, 2005.