REPORT DOCUMENTATION PAGE

Form Approved

OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
20-11-2014	Final	29 Aug 2012 - 28 Feb 2014

4. TITLE AND SUBTITLE

(124093) Inconsistency Correction and Re-localization for Robust Collaborative SLAM

5a. CONTRACT NUMBER

FA2386-12-1-4093

5b. GRANT NUMBER

Grant AOARD-124093

5c. PROGRAM ELEMENT NUMBER

61102F

6. AUTHOR(S)

Dr. Ping Tan

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National University of Singapore
4 Engineering Drive 3
Singapore 117576
Singapore

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AOARD
UNIT 45002
APO AP 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/AFOSR/IOA(AOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AOARD-124093

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Code A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this project, we solve two important problems in our CoSLAM system [1] – collaborative visual SLAM involving multiple cameras moving independently on different platforms. Firstly, we consider the correction of the inconsistency between 3D maps generated by different camera groups. This issue is generated when two groups of cameras were separated before and come back to have sufficient view overlap again. We adopt a graph-based approach to optimize the camera poses and individual maps together. Each camera pose is at a vertex in the graph and constrained linear least square problems are formulated and solved to obtain the optimized camera poses and a consistent 3D map. The other addressed issue is occasional failures of the SLAM system. Motion blur will be generated by fast motion of the UAV, and video frames might be lost due to problems of the Wi-Fi transmission. Both problems cause feature tracking failures and break the SLAM system. A re-localization mechanism is designed to make the CoSLAM system robust to these unexpected tracking failures, which register the current video frame with the previous cached key-frames.

15. SUBJECT TERMS

Navigation,Guidance, and Control, Vision

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Brian Sells, Lt Col, USAF
U	U	U	SAR	9	19b. TELEPHONE NUMBER *(Include area code)* +81-3-5410-4409

Final Report for AOARD Grant 124093

"**Inconsistency Correction and Re-localization for Robust Collaborative SLAM**"

Date: 2014-4-24

Name of Principal Investigators (PI and Co-PIs): TAN Ping
- e-mail address : eletp@nus.edu.sg
- Institution : National University of Singapore
- Mailing Address : Department of Electrical and Computer Engineering
- Phone : +65-6516-2130
- Fax : +65-6779-1103

Period of Performance: 08/29 /2012 –02/28/2014

Abstract: In this project, we solve two important problems in our CoSLAM system [1] – collaborative visual SLAM involving multiple cameras moving independently on different platforms. Firstly, we consider the correction of the inconsistency between 3D maps generated by different camera groups. This issue is generated when two groups of cameras were separated before and come back to have sufficient view overlap again. We adopt a graph-based approach to optimize the camera poses and individual maps together. Each camera pose is at a vertex in the graph and constrained linear least square problems are formulated and solved to obtain the optimized camera poses and a consistent 3D map. The other addressed issue is occasional failures of the SLAM system. Motion blur will be generated by fast motion of the UAV, and video frames might be lost due to problems of the Wi-Fi transmission. Both problems cause feature tracking failures and break the SLAM system. A re-localisation mechanism is designed to make the CoSLAM system robust to these unexpected tracking failures, which register the current video frame with the previous cached key-frames.

Introduction: Simultaneously localization and mapping (SLAM) is a critical component of autonomous robots. Visual SLAM aims at solving camera poses and building 3D map simultaneously based on the captured video frames. Traditionally, visual SLAM makes use of single camera or a pair of fixed stereo cameras mounted on a robot. These approaches are limited to static scenes and small workspaces. We developed a collaborative visual SLAM (CoSLAM) system [1] which allows collaboration between cameras mounted on independent platforms. Since the CoSLAM system is able to recover the 3D trajectories of dynamic objects, it can be used in more challenging scenes.

To further improve the robustness of the CoSLAM system and make it more applicable in practice, two important issues must be addressed. In our CoSLAM system, cameras are divided into groups according to their view overlap. Cameras with view overlaps are grouped together and collaborate with each other in map building and localization. This grouping could change dynamically over time. When two group of cameras (without view overlap previously) meet and start to have view overlap, they will be merged into a single group to facilitate their collaboration. However, the 3D maps built from these two groups could suffer from different drifting errors. Thus, these two 3D maps are inconsistent with each other and cannot be simply registered by a rigid transformation. This inconsistency of the 3D map needs to be corrected so that a unique global map can be reconstructed to guide the merged group of cameras. On the other hand, in real flights, the UAV often undergoes sudden fast motion or unstable communication link. Fast motion makes the video frame blurry, while communication errors generate delays in video transmission. Both problems will make frame-by-frame feature tracking algorithms fail. Thus, it is critical to develop an automatic recovery algorithm to allow the UAV to recover from such kind of failures. We build a re-localization

algorithm, which is activated once the feature tracking fails. This makes our CoSLAM algorithm robust to work in real UAV flights.

Experiment:

- Inconsistency correction

  Two camera groups will be merged if their cameras meet and have view overlap. To detect if cameras in different groups have view overlap, we project the map points generated from one camera onto the image planes of the cameras in the other group. If the number of visible points is large (> 30% of all map points from that camera in our implementation), and the area spanned by these points are large (> 70% of the image area), we consider the two cameras to have view overlap and will merge their camera groups.

  When cameras move away from each other, the mapping and localization are performed within each camera group independently. When the cameras meet again, due to drifting errors [2], the 3D maps reconstructed from different groups are inconsistent. For example, the same object could be reconstructed at different 3D positions in different groups. Hence, during group merging, we need to correct both the camera poses and map points to generate a single global consistent map. Suppose two camera groups are separated at the 1^{st} frame and are merged at the F^{th} frame. We will adjust all camera poses from frame 2 to F, and adjust the map points generated within these frames, which consists of two successive steps described in the following.

  We first estimate the correct relative poses between cameras at frame F. For this purpose, we detect and match SURF features between cameras in different groups, and then compute their relative poses (i.e. the essential matrices). We use these essential matrices to guide the matching of feature points (i.e. searching for correspondences in a narrow band within 3σ distance to the epipolar line). For each pair of matched feature points, we then merge their corresponding 3D map points by averaging their positions. In the next step, all the map points and their corresponding feature points in the F^{th} frame are put into bundle adjustment [3] to refine all camera poses.

  Now, we use the updated relative camera poses at the F^{th} frame as hard constraints to refine all camera poses. Figure 1 illustrates our problem formulation. We form an undirected graph where each camera pose is a vertex and each edge enforces a relative pose constraint. As shown in Figure 1, for each camera, its poses at neighboring frames are connected. For cameras in the same group, their poses at the same frame are connected if they are neighbors in the spanning tree. We fix camera poses in the 1^{st} frame. Except the relative poses at the F^{th} frame, we treat all the other relative poses as soft constraints. Hard and soft constraints are denoted by solid and dashed lines in Figure 1 respectively.

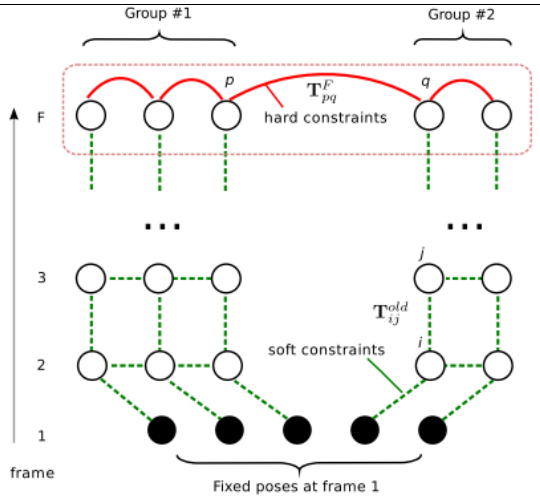

Fig. 1. Camera poses adjustment. Each vertex is a camera pose. Each edge represents a relative pose constraint, where solid and dash edges are hard and soft constraints respectively.

Let p = 1,…, P and q = 1,…, Q be cameras from different groups. We denote the pose of the camera p at the ith frame by T_p^i, and the relative pose between the camera p and q at the i[th] frame by T^i_{pq}, where

$$\mathbf{T}_p^i = \begin{pmatrix} \mathbf{R}_p^i & \mathbf{t}_p^i \\ \mathbf{0}^T & 1 \end{pmatrix}, \qquad \mathbf{T}_{pq}^i = \begin{pmatrix} \mathbf{R}_{pq}^i & \alpha \mathbf{t}_{pq}^i \\ \mathbf{0}^T & 1 \end{pmatrix}$$

R_p^i, R_{pq}^i and t_p^F, t_{pq}^F are rotation matrices and translation vectors. ® is used to account for the global scale difference between the two camera groups.

We treat the relative poses at the F[th] frame as hard constraints. Hence,

$$\mathbf{T}_q^F - \mathbf{T}_{pq}^F \mathbf{T}_p^F = \mathbf{0}_{4\times 4},$$

which is equivalent to

$$\begin{aligned} \mathbf{R}_q^F - \mathbf{R}_{pq}^F \mathbf{R}_p^F &= \mathbf{0}_{3\times 3} \\ \mathbf{t}_q^F - \mathbf{R}_{pq}^F \mathbf{t}_p - \alpha \mathbf{t}_{pq}^F &= \mathbf{0}_{3\times 1}. \end{aligned}$$

Although there are (P + Q) * (P + Q - 1) / 2 relative poses at the F[th] frame, we select only (P + Q - 1) of them, which either lie on the spanning trees of the camera groups or connect the two spanning trees, as illustrated by the solid lines in Figure 1. Putting all these constraints together, we get two linear systems with the following forms

$$Ur^F = 0 \quad \text{and} \quad Vt^F = 0,$$

Where r^F is a vector stacked with elements of all the rotation matrices at the F[th] frame, and t^F is a vector that consists of all the translation elements at the F[th] frame together with the scalar factor α.

The relative camera poses from the original SLAM process are used as soft constraints. For any cameras m and n connected by the dashed edge, we expect their relative pose to have small

change by the adjustment. Hence,

$$\mathbf{T}_m - \mathbf{T}_{mn}^{old}\mathbf{T}_n \approx 0.$$

Here, T_{mn}^{old} is the relative pose between m and n according to the SLAM process before merging. Putting all soft constraints together, we obtain two similar linear systems

$$Ar \approx a \neq 0 \quad \text{and} \quad Bt \approx b \neq 0,$$

where r and t are vectors stacked by all the rotation and translation elements of all frames. Notice that the right sides of the two linear systems are not equal to zero because the camera poses at the 1st frame are fixed.

Combining both the hard constraints and soft constraints, we obtain the updated cameras poses and the scale factor by solving two constrained linear least square problems

$$\arg\min_r \|Ar - a\|^2 \quad \text{s.t.} \quad \hat{U}r = 0$$

and

$$\arg\min_{\hat{t}} \|\hat{B}\hat{t} - \hat{b}\|^2 \quad \text{s.t.} \quad \hat{V}\hat{t} = 0,$$

where $\hat{t}$ is t appended with a scale factor α. $\hat{U}, \hat{V}, \hat{B}, \hat{b}$ are the augmented matrices and vectors by adding zero elements. Note that we do not impose orthonormality condition to the otation matrices in this formulation. Hence, once we obtain results from the above two equations, we further find the closest rotation matrices to the initial matrices by SVD (i.e. setting all the singular values to one).

The above optimization problem is converted to a set of sparse linear equations [4]. We use the CSparse [5] library to solve them in our system. After the camera poses have been updated, the 3D positions of map points are also updated by re-triangulating their corresponding feature points.

- Re-localization

  The SLAM system relies on feature tracking algorithm to propagate the map information from current frame to the next. If wrong or few feature correspondences are found, the camera pose estimation will fail and complete SLAM system could crash. The reasons for feature tracking failure between consecutive frames include motion blur, frame lost, and interference noise. In real applications such as UAV navigation, abrupt motion is inevitable and fast motion can cause blurry video frames. Also, when the SLAM system runs off-board on a ground station and receives video via a Wi-Fi link from the platform, there could be frame lost and noise interference during the transmission. All these issues can cause the camera tracking to fail and a method to recover the camera pose from tracking failure is proposed.

  Firstly, we design a heuristic method to cache some key frames. For each key frame, its downsized version and all its associated 2D features (including their locations and feature descriptors) are stored when the camera tracking is normal. We evaluate the tracking performance on every new image. The iterative Lucas-Kanade method with pyramids is used to find feature correspondences in current and last frames. If the correspondence cannot be found within 30 iterations or the search window moves by less than a threshold, we consider this feature a tracking outlier. Furthermore, we assume the feature movement between two

consecutive frames is small and reject those with a movement larger than certain pixels. Once the inlier ratio of the feature tracking algorithm falls below a critical threshold, the tracking is considered to be unreliable. To further enhance the robustness of tracking, we evaluate the camera pose estimation to determine a tracking failure. The reprojection error for each feature correspondence is computed after a new camera pose is solved. Feature correspondences with reprojection errors smaller than a threshold are considered inliers. If the ratio of inliers is smaller than a threshold, the tracking is treated as unreliable too.

The re-localization process is trigged right after the tracking is determined as unreliable. The re-localization algorithm will try to match the current frame directly with the cached key-frames to recover the camera pose. Since the nearest key-frame is most likely to be relevant, we always match the current frame with this nearest key-frame first. When it fails, we will search among all the other cached key-frames and identify the most similar key-frame to register the current frame. This search in key-frame is performed by evaluating the SSD (sum-of-squares) between the downsized thumbnails of the current frame and cached key frames, though more advanced visual search algorithms might be used such as the Vocabulary Tree algorithm [8].

Once the most similar key-frame is identified, we try to register the current frame with this key-frame. We adopt FAST feature detector [6] to find corners in current frame and compute the SIFT feature descriptor [7] for 2D feature locations in both frames. Then feature correspondences are found using Fast Approximation Nearest Neighbor Search Library (FLANN) implemented in OpenCV. The 3D locations of the matched feature points can be used to estimate the current camera pose by perspective-n-point algorithm. Once the current frame is registered, the system recovers from the tracking failure. So we further triangulate additional 3D map points to make the system more robust.

Results and Discussion: To verify the inconsistency correction, we examine the epipolar geometry before and after two camera groups are merged. Before the correction, the corresponding feature points in the images of group 2 do not lie on the epipolar lines generated by the ones of group 1. In comparison, after camera pose update and re-triangulation of map points, the corresponding feature points in all cameras lie on their respective epipolar lines. This experiment shows that the map inconsistency between two camera groups is corrected. Moreover, we also tested our re-localisation mechanism with some challenging videos.

Frame 508 Frame 509

Fig.2. Two consecutive images captured by a camera on a UAV. Some frames are lost obviously due to the wireless transmission and the view point is changed significantly, which causes feature tracking failure.

As shown in Figure 2, the view point between two consecutive frames in a video captured by a camera on a UAV could change a lot due to frame lost caused by Wi-Fi communication errors. When there is a large view change, most tracked feature points will be lost, which will cause localization failure.

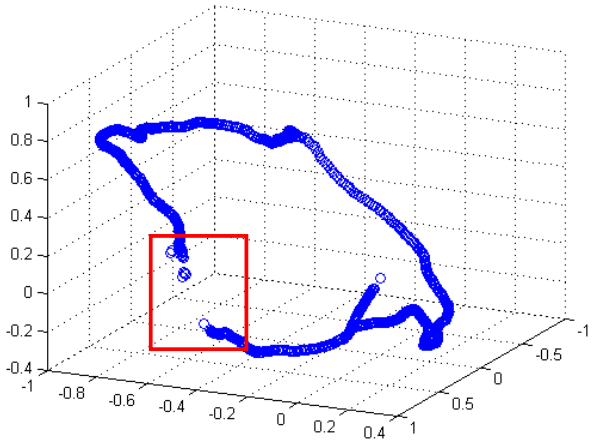

Figure 3. Blue curve shows the reconstructed camera trajectory (see text for more explanation).

Figure 3 shows the result of re-localization. The blue circles are the recovered camera poses at each frame of the input video. The red rectangle in Figure 3 highlights the effect of our re-localization component. There is a clear jump in the recovered camera poses. This is because of the frame loss due to poor Wi-Fi connection. This sudden large position change makes the feature tracking algorithm fail. Our re-localization algorithm successfully registered the camera pose despite the large position jump. The two frames with large position jumps are provided in Figure 2. The 508th frame is the most similar key-frame to the 509th frame, FAST feature is detected on both frames. The SIFT feature descriptors are evaluated at these features for matching. Thus, features in the 508th frame are re-tracked in the 509th frame and their 3D correspondences in the existing global map can be found. Then the camera pose of current frame is solved and new map points can be triangulated by newly detected 2D features.

We further verify the system with a video sequence affected by motion blur. Figure 4 shows a blurry frame due to fast camera motion, which also makes the feature tracking fail. Our CoSLAM system will enter re-localization and to find the most similar cached key frame. We compare the thumbnails of the current frame and all the cached key frames. If the SSD value is below a threshold, the current frame is then registered with the selected key frame. In our experiments, we set the threshold to 50 for 640 x 360 videos.

Figure 4. A blurred image caused by fast motion (Frame 113)

In Figure 5, our system identify a tracking failure at the 204th frame. It successfully finds the most similar key-frame, which is the 46th frame. We detect FAST feature and use the SIFT descriptor to match these frames. Our system identified about 215 features so that the camera pose of the current frame can be estimated accordingly.

Figure 5. The left is the found key frame and the right is the current image. 215 feature correspondences are found to re-localize the camera.

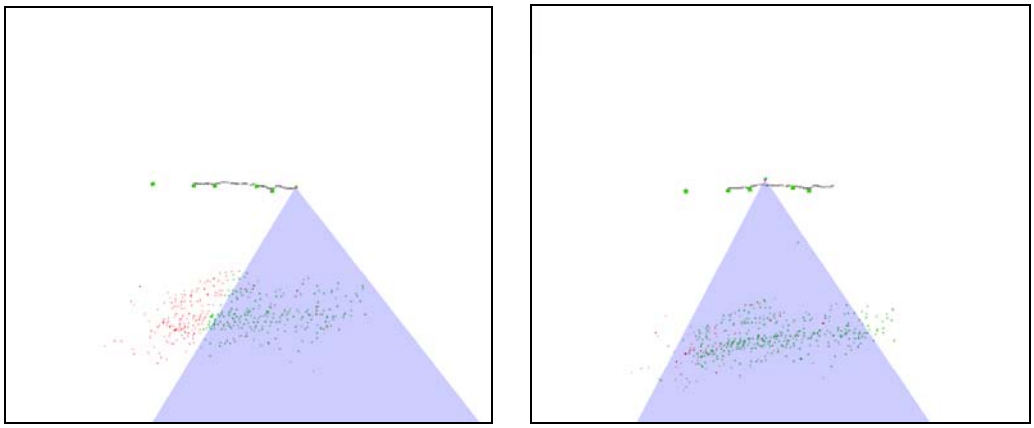

Figure 6. The left shows the camera position right before the camera tracking failure. The right shows the camera position after re-localization. Light blue area indicates the FOV of the camera. Sparse 3D map is represented by the gray dots and reconstructed camera trajectory is in blue.

Fig.6 shows the camera is successfully registered to the global map after the feature matching is performed with cached the key frame. The new camera pose is solved by the 2D-3D correspondences.

In future, we plan to fuse the results of visual SLAM and measurements from the inertial sensors for

better pose estimation. Also, the system is expected to be used in autonomous UAV navigation.

Reference

[1] D. Zou and P. Tan. CoSLAM: Collaborative Visual SLAM in Dynamic Scenes, IEEE Trans. on Pattern Analysis and Machine Intelligence, 35 (2), 354—366, 2013.
[2] K. Cornelis, F. Verbiest, and L. Van Gool. Drift detection and removal for sequential structure from motion algorithms. IEEE Trans. on Pattern Analysis and Machine Intelligence, 26(10):1249–1259, 2004.
[3] C. Bibby and I. Reid. Simultaneous localisation and mapping in dynamic environments (slamide) with reversible data association. In Proc. of Robotics: Science and Systems, 2007
[4] G. Golub. Numerical methods for solving linear least squares problems. Numerische Mathematik, 7(3):206–216, 1965.
[5] T. Davis. Direct Methods for Sparse Linear Systems. SIAM, 2006.
[6]. E. Rosten, and T. Drummond. "Machine learning for high-speed corner detection." European Conference on Computer Vision (ECCV), 2006. 430-443.
[7]. Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.
[8] D. Nister, Scalable Recognition with a Vocabulary Tree, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

List of Publications and Significant Collaborations that resulted from your AOARD supported project:

No publication yet

Attachments: Publications a), b) and c) listed above if possible.

DD882: As a separate document, please complete and sign the inventions disclosure form.

Important Note: If the work has been adequately described in refereed publications, submit an abstract as described above and refer the reader to your above List of Publications for details. If a full report needs to be written, then submission of a final report that is very similar to a full length journal article will be sufficient in most cases. This document may be as long or as short as needed to give a fair account of the work performed during the period of performance. There will be variations depending on the scope of the work. As such, there is no length or formatting constraints for the final report. Keep in mind the amount of funding you received relative to the amount of effort you put into the report. For example, do not submit a $300k report for $50k worth of funding; likewise, do not submit a $50k report for $300k worth of funding. Include as many charts and figures as required to explain the work.