Rough Terrain Visual Odometry

Motilal Agrawal
Artificial Intelligence Center
SRI International
Menlo Park, CA, USA
Email: agrawal@ai.sri.com

Kurt Konolige
Artificial Intelligence Center
SRI International
Menlo Park, CA, USA
Email: konolige@ai.sri.com

Abstract— **We present an integrated system to localize a mobile robot in rough outdoor terrain using visual odometry. Our previous work [1] presented a visual odometry solution that estimates frame-to-frame motion from stereo cameras and integrated this incremental motion with a low cost GPS. We extend that work through the use of bundle adjustment over multiple frames. Bundle adjustment helps to reduce the error significantly, thereby making our system more robust and accurate while still operating in real-time. Our new system can keep the robot well localized over several hundreds of meters to within 1% error. We present experimental results for our system over a 300 meters run in a challenging environment and compare it with ground truth Real Time Kinematic (RTK) GPS.**

I. Introduction

Visual Odometry (VO) [1], [2] is a relatively new technology for localizing mobile robots. VO works by finding interest points in the images and matching them between successive images. Robust methods are then used to estimate the camera motion from these matched interest points. Since cameras are inexpensive and provide high information bandwidth, they can serve as cheap and precise localization sensor. In addition, processing power has increased significantly to a point that such visual motion estimation can now be performed in real time on standard processors.

VO for outdoor terrains is in some ways more challenging than indoor environments. Firstly, outdoor environments are unstructured, and simpler features such as corners, planes and lines that are abundant in indoor environment rarely exist in natural environments. Secondly, outdoor terrains can be rough and undulating. Thus planar motion assumptions that are so common in indoor envoronments are not applicable. Succesful navigation over such rough terrains requires a euclidean motion model that utilizes the full six degrees of freedom. Another consequence of the rough terrain is that bumps in the ground can cause motion abruptness thereby making visual registration harder. Lighting changes and shadows also make it harder to match images.

Although motion estimation from video has been a widely researched topic in computer vision, real-time systems utilizing vision for localization of robots have been very few. One of the first of these systems was presented by Nister [3] for a monocular camera. Davison [4] also presented a real time monocular system using the extended kalman filter. However, their approach is best suited for indoor environments because of algorithmic complexity and growing uncertainty in feature locations. When compared to monocular video, motion estimation from stereo images is relatively easy and tends to be more stable and well behaved. Another major advantage of using stereo cameras is that one need not worry about the scale ambiguity present in monocular camera case. VO for stereo cameras was presented by Nister et al; [2] and Agrawal et al; [5], [1].

In our earlier work [5], [1], we presented a real time VO system and integrated it with an inexpensive GPS system to provide localization with ≈ 5% error over 100 meters or so. However, the motion estimation algorithm was sub optimal since it estimated the incremental motion between two consecutive frames. This work enhances our previous VO system through the use of bundle adjustment over multiple frames. In this work, features are tracked for as many frames as possible, and a sliding window of frames and features (the 'bundle') are adjusted in a non-linear optimization, to give the best motion estimate.

Bundle adjustment [6] is a process that iteratively adjusts the camera pose and the 3D location of the interest points in order to minimize the reprojection error of the interest points in all the camera frames. Global bundle adjustment is a non linear, compute intensive process and is usually used in its sparse form [7], [8] and invoked whenever a new frame is added to the system. We approximate each iteration of the bundle adjustment by alternating between estimation of the camera motion and 3D scene reconstruction. We initialize our approximate bundle adjustment with the camera pose obtained from our incremental VO algorithm [1]. Since we have stereo images, we initialize the 3D location of the interest points through stereo triangulation. Bundle adjustment helps to reduce the error significantly, thereby making our system more robust and accurate while still operating in real-time.

Our system is most similar to the recent work of Mouragnon et al. [9] with a few major differences. Firstly, they use a monocular camera while ours is a stereo algorithm. Secondly, our bundle adjustment process for stereo alternates between pose computations and 3D leading to faster convergence and run time. Therefore we can use all the frames of the sequence without discarding the non key frames. We integrate our VO system with a cheap IMU/GPS system to maintain global pose consistency. Finally we present results of our robot navigating over rough terrain whereas [9] presents results over smooth roads.

Figure 1 shows our robot in action on a typical terrain. A pair of stereo cameras are located on the sensor mast of the robot looking outward. The cameras are pointed forward at a slight angle. The baseline is 12 cm and the height above ground is about 0.5 m. This arrangement presents a challenging situation: short baseline and a small offset from the ground plane makes it difficult to track points over longer distances. A pair of wheel encoders and an Xsens IMU are used to complement the visual pose system. A Garmin GPS sensor is located on top of the sensor mast.

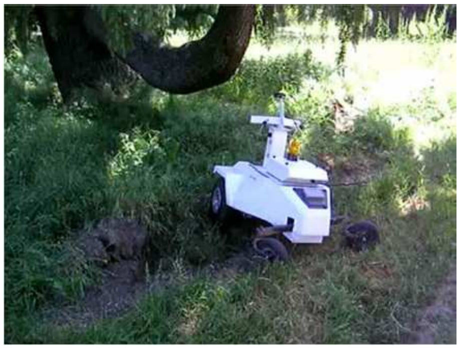


Fig. 1. Our robot in typical terrain. Robot is part of a DARPA project, Learning Applied to Ground Robots(LAGR). Two stereo systems are on the upper crossbar.

The rest of the paper is organized as follows. First we present our incremental visual odometry algorithm in Section II. Our bundle adjustment process is explained in Section III. Fusion with GPS and IMU sensors is described in Section IV. Results on long robot runs are presented and compared to ground truth RTK GPS in Section V and finally Section VI concludes this presentation and discusses ongoing and future work.

II. Incremental VO

Our visual odometry system [5], [1] uses feature tracks to estimate the relative incremental motion between two frames that are close in time. Interest points which are Harris corners are detected in the left image of each stereo pair and tracked across consecutive frames. These interest points are then triangulated at each frame based on stereo correspondences. Three of these points are used to estimate the motion using absolute orientation. This motion is then scored using the pixel reprojection errors in both the cameras. We use the disparity space homography [10] to evaluate the inliers for the motion. In the end, the hypothesis with the best score (maximum number of inliers) is used as the starting point for a nonlinear minimization problem that minimizes the pixel reprojection errors in both the cameras simultaneously, resulting in a relative motion estimate between the two frames. The IMU and the wheel encoders are also used to fill in the relative poses when visual odometry fails. This happens due to sudden lighting changes, fast turns of the robot or lack of good features in the scene (e.g. blank wall). Thus it complements the visual pose system.

We have found that the approach outlined above is very efficient and works remarkably well, even for stereo rigs with a small baseline. The fact that we are triangulating the feature points for each frame, builds a firewall for error propagation. However, this also means that there will be a drift when the rig is stationary. In order to avoid this drift, we update the reference frame (the frame with reference to which the motion of the next frame is computed) only when the robot has moved some minimum distance (taken to be 5 cm in our implementation). The fundamental reason that our approach gives reliable motion estimates, even in small-baseline situations, is our use of image-based quantities, and treating both the left and right images symmetrically.

III. Bundle adjusted VO

Bundle adjustment [6] iteratively adjusts the camera pose and the 3D location of the interest points in order to minimize the reprojection error of the interest points in all the camera frames. For bundle adjustment to be effective, interest points need to be tracked over multiple frames. Our incremental VO was computed frame-to-frame, that is, feature points were extracted for frame 1 and frame 2, the motion between the frames was computed, and then the features were discarded. In the bundle adjusted VO, features are tracked for as many frames as possible, and a sliding window of frames and features are adjusted in a non-linear optimization, to give the best motion estimate. This optimization step reduces the error resulting in very accurate poses.

Our two-frame matching algorithm [1] can be used to link the interest points over the multiple frames. In addition, since we get the motion from our incremental VO algorithm and we have stereo, we can predict the location of the interest point in the current frame. This drastically cuts down the search space for feature matching and helps find matches for additional interest points which did not match previously due to ambiguity caused by large regions in which to search for the match. Furthermore, we use only those interest points which are inliers to the computed motion from the two frame algorithm and also can be tracked over at least m frames. This ensures that only good tracks are used for bundle adjustment.

Bundle adjustment is a non linear, compute intensive process and is usually used in its sparse incremental form [7], [8], [9] and invoked whenever a new frame is added to the system. Each bundle adjustment is performed over a sliding window of N frames, n of whose motion is already known ($1 \leq n < N$). Bundle adjustment solves for the $N-n$ motion parameters and also the 3D positions of the tracked interest points. Our bundle adjustment process is summarized below.

- Initialize pose of the latest frame i from incremental VO
- Retain inlier feature tracks which are at least m frames long
- Initialize the pose of all the other frames from previous bundle adjustment
- Iterate till convergence
    1) *Structure Computation:* For each inlier feature, compute the 3D location from the image tracks and the current poses of the frames
    2) *Motion Computation:* For each of the $N-n$ frames, compute its pose from the 3D locations of the feature points and its corresponding image locations
    3) Compute the reprojection errors for convergence test

Our bundle adjustment interleaves structure and motion computation in the main loop. Motion of the most recent frame is initialized from the results of incremental VO. Motion of all the other frames is initialized from the results of previous bundle adjustment. Given the motion of each stereo frame, we can reconstruct each feature point and get its 3D location. This structure computation is then followed by refining the motion using the structure to bootstrap. These two steps are described briefly next. More details can be found in [11].

A. Structure Computation

Denote by x^i_j the coordinates of the $j-th$ point as seen by the $i-th$ camera. Let P^i denote the projection matrix of the $i-th$ camera and let X^j be the 3D location of the point. Therefore we have $x^i_j = P_i X^j$. Given x^i_j and P_i, X_j can be recovered easily through Direct Linear Transform (DLT).

B. Motion Computation

Recovering P_i from x^i_j and X^j is nonlinear process and can be accomplished using Levenberg-Marquardt minimization. The quantity to be minimized in this case is the geometric distance between the measured and the projected point

$$\min_{P_i} \sum_{ij} d(P^i X_j, x^i_j) \tag{1}$$

where $d(x, y)$ is the geometric image distance between x and y

The above interleaving of structure and motion computations minimizes the same cost functions as bundle adjustment and it should result in the same solution as the full blown bundle adjustment, provided it converges. Since we have a very good initial estimate of the motion from the incremental VO, it only takes a few iterations for the interleaving to converge. Interleaving has the added advantage that it is computationally very efficient since at each step we only are dealing with either motion or structure computations.

IV. Sensor fusion

In order to maintain a long term accurate global pose, we performed two types of filtering on the bundle adjusted VO output. These filters provide the necessary corrections to keep errors from growing without bounds.

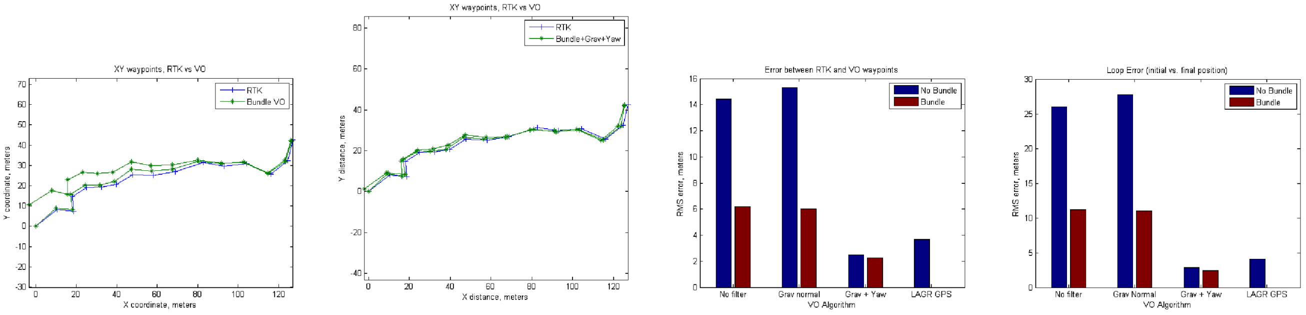
1) *Gravity Normal:* The IMU records tilt and roll based on gravity normal, calculated from the three accelerometers. This measurement is corrupted by robot motion, and is moderately noisy.
2) *GPS Yaw:* The IMU yaw data is very bad, and cannot be used for filtering (for example, over the 150 m run, it can be off by 60 degrees). Instead, we used the yaw estimate available from the LAGR GPS. These yaw estimates are comparable to a good-quality IMU. Over a very long run, the GPS yaw does not have an unbounded error, as would an IMU, since it is globally corrected;

Our filters for the gravity normal and yaw are very simple linear filters that essentially nudge the robot's pose towards global consistency through a very small gain factor. GPS yaw filtering is done when the GPS receiver has at least a 3D position fix and the vehicle is travelling 0.5 m/s or faster, to limit the effect of velocity noise from GPS on the heading estimate. In addition, filtering is performed only if the robot has travelled a certain distance from the last filtering. This limits the effect of short term noise in the fused pose and also makes sure that the robot's pose does not change when the vehicle is stationary.

Though the filter is very simple, it is effective and improves the accuracy of the computed motion over long term. A better way would be to perhaps incorporate the yaw and grav estimates directly into the bundle adjustment, using their covariances. Unlike our previous work [1], we have turned off any position filtering based on GPS i.e; we completely ignore position estimates from the GPS. The bundle adjusted VO does a very good job of estimating the distance travelled. As long as the yaw is accurate, the robot will stay well localized.

V. Results

We surveyed a course using an accurate RTK GPS receiver. The 'Canopy Course' was under tree cover, but the RTK GPS and the LAGR robot GPS functioned well. Sixteen waypoints were surveyed, all of which were within 10 cm error according to the RTK readout. The total length of the course was about 150 meters. Subsequently, our robot was joysticked over the course, stopping at the surveyed points. The robot was run forward over the course, and then turned around and sent backwards to the original starting position. The course itself was flat, with many small bushes, cacti,



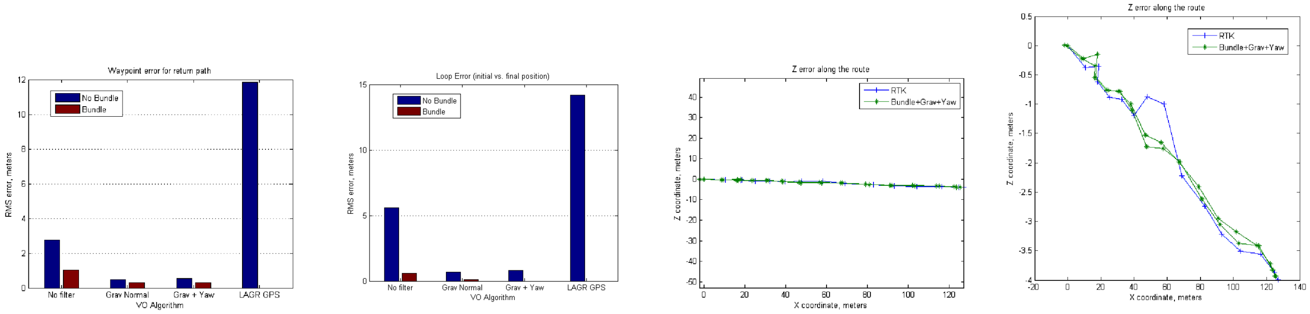
(a) Bundle adjusted VO

(b) Bundle adjusted VO with gravity normal and yaw filtering

(c) RMS error

(d) Loop closure error

Fig. 2. XY localization error for the canopy sequence



(a) Z wayoint RMS error

(b) Z loop closure error

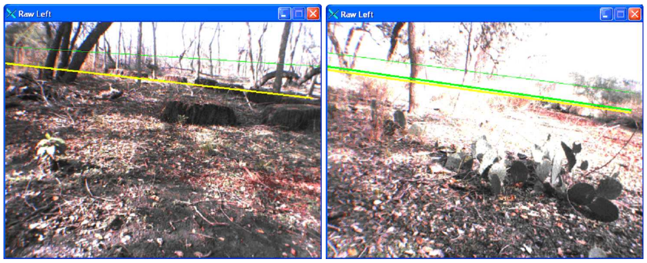
(c) Z error along the route

(d) Z error magnified

Fig. 3. Z error for the canopy sequence

downed tree branches, and other small obstacles. Notable for VO was the sun angle, which was low and somewhat direct into the cameras on several portions of the course. Figure 4 shows a good scene from the LAGR robot in the shadow of the trees, and a poor image where the sun washes out a large percentage of the scene. (The lines in the images are horizon lines taken from VO and from ground plane analysis). The uneven image quality makes it a good test of the ability of VO under challenging realistic conditions. The joystick control was moderate, with no sharp turns or quick accelerations and average speed of the robot about 1m/s. Our VO algorithm was able to match interest points along the whole route, even though the average data rate was only about 5 Hz. Although there were some big jumps in the images (one of over 0.5 meters), the feature tracker was able to track enough features to successfully match the frames. Bundle adjustment was performed over five stereo frames by matching all the features that could be tracked over those frames. Results for varying the number of frames are presented in Section V-C.

Since we do not know the exact initial heading of the robot in UTM coordinates, we used an alignment strategy that assumes there is an initial alignment error, and corrects it by rotating the VO path rigidly to align the endpoint as best as possible. This strategy minimizes VO errors on the outward path, and may underestimate them. However, for the return path, the errors will be caused only by VO, and can be taken as a more accurate estimate of the error. Since the robot was moved in a loop, the difference between the initial and final pose can also be used as a measure of error, and this measure is not corrupted by the initial heading problem.



(a) A good scene as viewed from the left camera

(b) Scene washed out by the sun

Fig. 4. Sample images from the LAGR robot for the canopy course

We analyzed the following errors:

1) XY distance error between the VO poses and the RTK poses.
2) Z distance error between the VO poses and the RTK

poses. Z error is less accurate on the RTK poses, and is also globally correctable by gravity normal for VO.

3) Initial vs. final pose for VO, in XY and Z.

A. XY Error

Figure 2 compares the XY locations of the waypoints with ground truth RTK GPS. Figure 2(a) shows the best result obtained using our bundle-adjusted VO with gravity normal and GPS yaw filtering. As shown, the errors between waypoints is very small, amounting to less than 1% of distance traveled. Without filtering, the results are worse(Figure 2(b)), amounting to about 3% of distance traveled. At some points in the middle of the return trip, the VO angle starts to drift, and at the end of the backward trip there is about a 10m gap. Note that this effect is almost entirely caused by the error in the yaw angle, which can be corrected by a good IMU.

Figure 2(c) shows the RMS error between VO (with different filters) and the RTK waypoints, on the return path. As noted above, the forward VO path of the robot has been aligned with the RTK path. Without yaw filtering, the bundle adjustment does much better than the non-bundle adjusted VO. It is marginally better with yaw filtering. Both versions of VO beat the LAGR GPS over the 150 meters of the return path.

We can also examine the loop closure error (Figure 2(d)), looking at the difference between start and end positions. As noted, this measure is not influenced by the initial angle. The loop errors, without filtering, are quite large for non-bundled VO. The bundled VO does a respectable job of keeping the error low without filtering, amounting to about 3% of distance traveled (300m). With yaw filtering, the errors are comparable to the waypoint errors, and less than 1% of distance traveled. Again, both filtered VO algorithms beat LAGR GPS.

B. Z Error

Figure 3(a) shows the RMS error between VO (with different filters) and the RTK waypoints, on the return path. For the Z direction, the forward-path alignment also aligns the IMU gravity normal to the camera frame. Normally, the IMU orientation with respect to the camera frame is indirect: camera to vehicle, and then vehicle to IMU. The latter is affected by things like tire pressure. For our gravity-normal filtering, we do a direct transformation from the IMU gravity normal to camera coordinates, based on alignment from the outward path.

Without gravity-normal filtering, the errors are modest, less than XY errors. Bundle adjustment gives two times improvement, with and without filtering. Both versions of VO beat the notoriously unreliable LAGR GPS over the 150 meters of the return path. We also examine the loop closure error in Figure 3(b). The loop errors, without filtering, are modest for non-bundled VO. The bundled VO does a great job of keeping the error low without filtering, amounting to $<$ 0.3% of distance traveled (300m). With gravity normal filtering, the errors are much less than the waypoint errors, and almost vanish for bundled VO.

Given the low error for loop closure, it is interesting to ask if the waypoint errors are caused by error in the RTK readings. Figures 3(c) and 3(d) plots RTK vs VO readings along the course. The first plot uses equal axis scales, and shows a gradual declination of about a 1.5% grade along the route. We used the outward trip to align the IMU and the camera frames; then, the backward trip shows this alignment works very well, to adjust the VO readings. Most of the Z error comes in two readings in the middle of the course; we expanded the vertical scale in the next figure. The course was relatively uniform the robot did not traverse any major ditches or hills. So the RTK readings in the middle of the course are probably anomalous. The RTK Z error is nominally 40 cm, and may have been worse under the tree canopy. So in this case, we can take the bundle-adjusted VO as the ground truth, and measure RTK Z errors.

C. Effect of number of frames

It is obvious that as we increase the number of frames, N, over which we perform bundle adjustment, the visual odometry results will become more accurate. Figure 5 plots the loop closure error for different number of frames. In the absence of any bundle adjustment (the first bar), the loop closure error is maximum and about 23m. Performing bundle adjustment even over two frames brings the error down to 21m. This error keeps on decreasing with the number of frames until it reaches a minimum of 11m for 5 frames. Further increasing the number of frames does not change the error much. This is due to the fact that most features can only be tracked for about five frames.

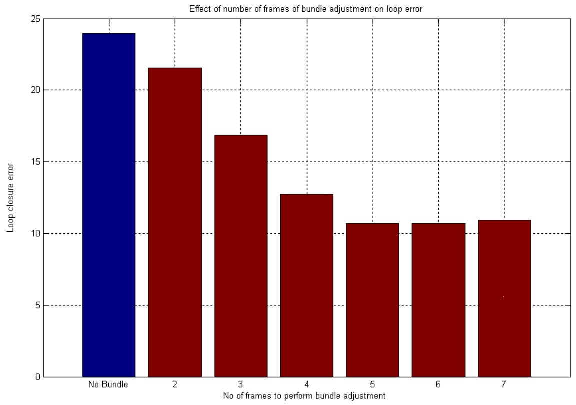


Fig. 5. Effect of the number of frames for bundle adjustment on the loop closure error

VI. Conclusion

We have presented a complete system for localization of a robot in unstructured rough terrain, using stereo vision as the primary sensor. The system presented here enhances our previous system through the use of Bundle adjustment over multiple frames. This helps to keep the drift error down. Bundle-adjusted VO has the potential to be an accurate substitute for RTK GPS, over distances on the order of a kilometer. With good IMU readings for yaw, and noisy gravity normal readings, it is possible to get $< 1\%$ error over 300m.

Our localization system has been tested in varied outdoor terrain, including under tree cover where GPS does not work very well and sandy terrain which causes lot of wheel slippage and wheel based odometer to fail. Our system is very robust - we can typically give it a goal position several hundred meters away, and expect it to get there within a meter or two. We are currently in the process of porting our system on a larger robot that can travel upto 5m/s. Finally, we would also like to augment our VO system with visual landmarks to further reduce the drift error and recognize places seen before, thereby allowing us to do loop closures.

Acknowledgement

The work reported in this paper was supported in part by a contract with DARPA under the Learning Applied to Ground Robots program.

References

[1] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," in *Proc. International Conference on Pattern Recognition*, August 2006.

[2] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2004.

[3] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 26, no. 6, pp. 756–770, June 2004.

[4] A. Davison, "Real-time simultaneaous localisation and mapping with a single camera," in *Proc. International Conference on Computer Vision (ICCV)*, 2003, pp. 1403–1410.

[5] M. Agrawal, K. Konolige, and R. Bolles, "Localization and mapping for autonomous navigation in outdoor terrains: A stereo vision approach," in *Proc. IEEE Workshop in Applied Computer Vision (WACV)*, February 2007, p. To Appear.

[6] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzibbon, "Bundle adjustment - a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. LNCS. Springer Verlag, 2000, pp. 298–375.

[7] K. Konolige and M. Agrawal, "Frame-frame matching for realtime consistent visual mapping," in *Proc. International Conference on Robotics and Automation (ICRA)*, 2007, p. To Appear.

[8] C. Engels, H. Stewnius, and D. Nister, "Bundle adjustment rules," *Photogrammetric Computer Vision*, September 2006.

[9] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3d reconstruction," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, June 2006, pp. 363 – 370.

[10] M. Agrawal, K. Konolige, and L. Iocchi, "Real-time detection of independent motion using stereo," in *IEEE workshop on Motion (WACV/MOTION)*, January 2005.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.