Monocular Visual Odometry in Urban Environments

Jean-Philippe Tardif Yanis Pavlidis Kostas Daniilidis
GRASP Laboratory, University of Pennsylvania,
3401 Walnut Street, Suite 300C, Philadelphia, PA 19104-6228
{tardifj, pavlidis}@seas.upenn.edu, kostas@cis.upenn.edu

***Abstract*— We present a system for Monocular Simultaneous Localization and Mapping (Mono-SLAM) relying solely on video input. Our algorithm makes it possible to precisely estimate the camera trajectory without relying on any motion model. The estimation is fully incremental: at a given time frame, only the current location is estimated while the previous camera positions are never modified. In particular, we do not perform any simultaneous iterative optimization of the camera positions and estimated 3D structure (local bundle adjustment). The key aspects of the system is a fast and simple pose estimation algorithm that uses information not only from the estimated 3D map, but also from the epipolar constraint. We show that the latter leads to a much more stable estimation of the camera trajectory than the conventional approach. We perform high precision camera trajectory estimation in urban scenes with a large amount of clutter. Using an omnidirectional camera placed on a vehicle, we cover the longest distance ever reported, up to 2.5 kilometers.**

I. INTRODUCTION

Robot localization without any map knowledge can be effectively solved with combinations of expensive GPS and IMU sensors if we can assume that GPS cannot be jammed and works well in urban canyons. Given accurate robot poses from GPS/IMU, one can quickly establish a quasi-dense 3D map of the environment if provided with a full 2.5D laser scanning system and if the system can detect moving objects around. This is how vehicles, in their majority, navigate autonomously in recent DARPA Challenges. However, for massive capture of urban environments or fast deployment of small robots, we are faced with the challenge of how to estimate accurate pose based on images and in particular whether this can work for a very long route in the order of kilometers. We know that the best results can be obtained by combining sensors, for example cameras with IMUs. Because we address applications where video data is useful for many other purposes, we start from the assumption that we have cameras and push this assumption to the extreme: cameras will be the only sensor used.

We choose an omnidirectional camera array without overlap between the cameras, which is the reason why we talk about monocular vision. We will concentrate on the localization problem, which we call visual odometry. We simultaneously establish a metric map of 3D landmarks. In this work, no recognition technique is used for loop closing.

The main challenge for monocular visual odometry and for mono-SLAM is to minimize the drift in the trajectory as well as the map distortion in very long routes. We believe that instabilities in the estimation of 3D motion can be weakened or even eliminated with a very large field of view. Not only does the two view estimation become robust but landmarks remain in the field of view for longer temporal windows. Having a good sensor like a camera with almost spherical field of view, we believe that the main priority is not the propagation of the uncertainty but the correctness of the matches. Having thousands of landmarks in each omnidirectional view makes the correspondence problem hard. This is why we use an established RANSAC based solution for two-view motion estimation. Based on triangulation from consecutive frames we establish maps of landmarks. An incoming view has to be registered with the existing map. Here is the main methodological contribution in this paper: *given the last image and a current 3D map of landmarks, we decouple the rotation estimation from the translation in order to estimate the pose of a new image*.

This is the main difference to the state of the art in visual odometry [30]: instead of applying classic three-point based pose estimation, we compute the rotation from the epipolar geometry between the last and the new image and the remaining translation from the 3D map. We cascade RANSAC for these two steps and the main effect is that rotation error decreases significantly as well as the translation and in particular the heading direction. The intuition behind this decoupling is two-fold. Firstly, contrary to pose estimation from the 3D points, the epipolar constraint estimation is not subject to error propagation. Secondly, there is effectively a wider overlap of the field of views in epipolar geometry than the effective angle that the current 3D map spans. The difference are points far away which have not been reconstructed. Such points (like in the vanishing direction of the road) which are a burden according to [8], do contribute to the epipolar estimation and thus to a better relative rotation estimation.

We have been able to perform 2.5km long visual odometry with the smallest drift with respect to the results in the literature. This is verified by comparing it to GPS.

The main advantages of our approach can be summarized as follows:

- The data association problem is solved robustly and makes the algorithm unaffected by 3rd party motions like surrounding vehicles and humans occupying a significant part of the field of view;
- We do not apply any motion model and do not make any assumption about uncertainty in the prediction or in the measurements;
- At no stage do we need to apply an expensive batch approach like bundle adjustment

- The propagation of the global scale is robust and the underestimation in the length of the trajectory small;
- The orientation of the vehicle is estimated accurately.

We believe that considering the length of the route in our results and the complexity of the environment, we surpass the state of the art by using our new pose estimation algorithm and by employing an omnidirectional camera.

II. Related work

Progress and challenges in multi-sensor SLAM have been nicely summarized in the tutorial by Durrant-White [11], in Thrun's mapping survey [37], and in Frese's survey [15]. Here, we will concentrate on summarizing purely vision based SLAM systems, and in particular monocular SLAM systems. We will describe related work as follows: We will start with multiple frame structure from motion approaches from computer vision, characterized mainly by their application in short motion ranges. Then, we describe approaches capable of handling long-range sequences.

Since the eighties, several approaches have been introduced to extract the motion and the structure of an object rather than a scene from multiple video-frames either in recursive or batch filtering modes [3], [5]. Object features were visible over a long temporal window and the object projection occupied a significant part of the field of view. Systems employing motion models effectively smoothen the trajectory of the camera and constrain the search area for feature correspondence across frames. We list [6] as the latest approach in this category, and as the first who introduced the inverse depth in the state vector converting this way a nonlinear measurement equation to an identity. In all these approaches, the global scale is fixed for the first frame while the global reference frame is defined from three points in the first frame. Among the very early vision-based approaches, we have to mention [10] who used a camera over several kilometers for lane following but not for global localization or mapping. In the nineties, several approaches emerged with the goal of 3D modeling from video starting with approaches from Oxford [2], and culminating to state of the art systems by Pollefeys *et al.* [31] and Heyden and Kahl [19] whose main goal was a dense reconstruction of a 3D scene from uncalibrated views. Multiple view methods are now textbook material [17], [26].

We continue with approaches free of motion models. Nister *et al.* [30] were the first in recent history to produce a real-time monocular SLAM without motion model or any assumptions about the scene. They use the 5-point algorithm [28] with preemptive RANSAC for three consecutive frames and estimate subsequently pose with a 3-point algorithm [16]. Points are triangulated between the farthest viewpoints from which they are visible. A "firewall" prohibits the inclusion of frames for triangulation past a particular frame in history. Royer *et al.* [33] track Harris corners with normalized cross-correlation. Similar to [30], local 3D motion is computed from each incoming image triplet using RANSAC for inlier detection. The last image of the triplet is used in combination with the current estimate of the 3D structure to compute the pose of the most recent view. A hierarchical bundle adjustment is applied to subdivisions of the sequence with at least one overlapping frame among them. The established map is used for localization in a second pass. In similar spirit is the work by Kaess and Dellaert [20] who use an array of outwards looking cameras but they test in a short range sequence. Sipla-Anan and Hartley [34] use the same camera as we do but concentrate on the loop closing task.

All approaches mentioned in the last paragraph have the following in common with our approach: they do not use any motion model, a requirement for any filtering or Bayesian approach. Among filtering approaches, Davison's real-time monoSLAM system [8], [9] as well as Eade and Drummond's also monocular system [12] are the most competitive. Similar to [6], both groups use inverse depth parametrization with the advantages of being able to estimate depths of very far features and being less dependent on the initial estimates. A motion model via a combinaion of a particle filter and EF is used by Karlsson *et al.* [21], too, in their trademarked vSLAM system., tested in a tour of a two-bedroom apartment. Corke *et al.* [7] use a catadioptric vision system for visual odometry with Iterated EKF and report that tracking fails after 30m (300 frames).

Regarding localization but from a sparse set of views, we refer to Teller's work [36] who estimates all pairwise epipolar geometries from an unordered set of omnidirectional images and then fixes the translation scales with a spectral approach [4]. An interesting fusion is the adjustment of a purely visually obtained long range map [23] where visual odometry is adjusted with a rough map. We will not refer to approaches using an IMU [27], odometry, making assumptions about the terrain or using laser scanners for mapping. We refer though as state of the art to the longest sequence ($>$10km) that has been used by Konolige *et al.* [22] who achieved an error of only 10m after 10km using only a stereo camera and an IMU. Their results show that sparse bundle adjustment can improve pure stereo vision by halving the error while the use of an IMU can decrease the error by a factor of 20.

III. Overview

Monocular Visual SLAM in urban environments with a camera mounted on a vehicle is a particularly challenging task. Difficulties indeed arise at many levels:

- Many outliers are present (other moving vehicles, oscillating trees, pedestrians) and must not be used by the algorithm;
- Frequently, very few image landmarks are present, for instance when passing by a park or a parking lot;
- Occlusions, especially from the trees, make it almost impossible to track landmarks for a long period of time.

Under these circumstances, the use of an omnidirectional camera is almost unavoidable as large parts of the field of view are sometimes useless. We combine five cameras for a total of roughly 4 mega-pixels. Because, their optical centers are virtually aligned, this can be seen as a high resolution omnidirectional camera. In particular, no depth information

can be recovered without motion. To avoid confusion in the rest of the paper, we will refer to those cameras and their acquired images as a single omnidirectional camera and single image. This high resolution comes at the cost of a rather low frame-rate: a maximum of 10 frames per second. Our system can handle this kind of frame-rate; in fact, we could achieve very good results using only 3 frames per second (see §V).

The system we present follows the approach of Nister *et al.* [30] with some important differences, the most important of which is the decoupled estimation of the rotation and translation using epipolar geometry. Furthermore, we use SIFT features instead of corners. Similarly to them, motion estimation is completely incremental and no sophisticated optimization step is performed.

IV. Detailed description

In this section, we detail each step of our algorithm summarized in figure 1.

A. Landmark detection

The traditional approach for landmark detection is the use of image corners such as Harris. It has been shown that these can be computed very efficiently and matched by normalized cross-correlation as long a the image deformation remains low. Recently, Barfoot demonstrated that Scale Invariant Feature Transform (SIFT) detector [24] could provide more robust image matches in the context of SLAM [1], [13].

We tested both the Fast Corner Detector [32] and the SIFT detector and found the latter to perform better. This is most likely because of the low frame-rate of our camera. Furthermore, the presence of trees, sometimes covering a large portion of the image, yields large numbers of corners of poor quality. This effect was not as important in the case of SIFT detector. We use the implementation of the SIFT detector by Vedaldi and Fulkerson, available online[1], and set the peak-threshold to one, which is lower than the default value suggested in Lowe's original implementation. On average, we obtain, between 500 to 1000 landmarks per image.

B. Landmark tracking

Landmark matching between two images is the basis for our SLAM algorithm. Again, many possibilities were considered before settling on the one described below. First, we considered a KLT based tracker [25] which did no provide sufficient quality at our frame rate. Better results were obtained using either normalized cross-correlation of corners or SIFT matching, but the latter gave the best results. The traditional approach for SIFT matching is to compute the squared difference between SIFT descriptors. A match is accepted if the ratio between its score and the score of the second best match is at least 1.5. We found this criterion to be far too restrictive for our application. Indeed, valid matches between regions with repeating patterns, that are common in urban scenes (*e.g.* buildings), were ofter rejected. Lowering the threshold caused more landmarks to be matched, but the quality of the matching was also reduced.

[1]http://vision.ucla.edu/ vedaldi/code/vlfeat/vlfeat.html

OUPUT

- List of 3D points: $\mathbf{X} = \{\}$
- List of cameras: $\mathbf{P} = \{\}$

INITIALIZATION

- Tracking until keyframe is found
- Preemptive RANSAC for relative pose estimation
- Triangulation of landmarks

MAIN LOOP

- $\theta_{\text{min}} = 5$
- $D_{\text{min}} = 20$ pixels
- At the current image I_n:

  Tracking:
  - Detect and compute SIFT features
  - Match with view I_{n-1}
  - Preemptive RANSAC for relative pose $\rightarrow \mathsf{E}'$
  - Refine matches using E'
  - If 95% of landmarks are within D_{min}, drop frame, repeat tracking

  Motion estimation
  - Preemtive RANSAC for relative pose $\rightarrow \mathsf{E}_{n-1,n}$
  - Remove bad tracks
  - Preemptive RANSAC for camera translation $\rightarrow \mathbf{P}\{n\}$

  Structure computation
  - For each landmark l:
  - If camList$_l =$, camList$_l = \{n-1\}$
  - N-view triangulation with views from camList$_l$ and view $n \rightarrow$ 3D point $\mathbf{M}$
  - Reject point if high re-projection error, goto next landmark
  - Compute angle θ between N, $\mathbf{P}\{\text{camList}_l(end)\}$ and $\mathbf{P}\{n\}$
  - If $\theta > \theta_{\text{min}}$,
    $\mathbf{X}\{l\} \leftarrow \mathbf{M}$
    add $\{n+1\}$ to camList$_l$

Fig. 1. Overview of our visual odometry algorithm. Details are provided in §IV.

To improve the quality of the matching, we make a simple assumption similar to instantaneous constant velocity as proposed by Davison *et al.* among others [8]. Our assumption is weaker and we only use it to speedup the search. Denote the current and two previous images by I_n, I_{n-1} and I_{n-2}. We constrain our search for correspondences between I_n and I_{n-1} , by assuming that the epipolar geometry changes relatively little between neighboring images. In other words, $\mathsf{E}_{n-2,n-1} \approx \mathsf{E}_{n-1,n}$.

We divide the matching into two steps. First, for every landmark, we only consider landmarks in the second image that are within a relatively high distance ($D_E = 20$ pixels) to the epipolar line. Furthermore, no minimum score ratio is used, but only matches of a very good score (below 40000)

are kept. These matches are used to robustly estimate the epipolar geometry between the two frames (see §IV-E). Then, we recompute the matches, this time, setting D_E to two pixels while doubling the matching score threshold.

C. Initialization

Initialization of the algorithm is very simple. Tracking is done until a keyframe is found, as described above. We then robustly estimate the relative orientation of the second frame with respect to the first one. The distance between the two cameras is set to one which also fixes the overall scale of the reconstruction. Then, two-view triangulation of the landmarks is performed.

D. Keyframe selection

Our keyframe selection is akin to the one of Royer *et al.* [33] and does not rely on Geometric Robust Information Criteria [38]. A frame is dropped if more than 95% of the landmarks moved within a threshold $D_{\min}$, set to 20 in all our experiments. In practice, almost all acquired images are keyframes when the vehicle runs at full speed.

E. Robust pose estimation

Our pose estimation procedure is the key to the high quality of the trajectory estimation. At this step, we are given a set of tracked landmarks. In addition, an estimate for the location of a 3D point, previously obtained by triangulation (see §sec:trian), is given for some of these tracks.

We proceed with a geometric hypothesize-and-test architecture similar to Nister *et al.* [30] and Royer *et al.* [33]. However we obtain superior results in practice. Before going into details, it is useful to outline the approach used in the two aforementioned works.

The first step is to remove bad correspondences between the two frames. Their approach, which we also adopt, is to rely on Random Sampling and Consensus (RANSAC)[14] with the minimal relative pose algorithm (so called 5-point algorithm)[35]. Our implementation is a fast variation of RANSAC called preemptive RANSAC [29]. Only landmarks with known 3D points are retained for the second step. Again, preemptive RANSAC is used with an algorithm for computing the pose requiring three 3D-2D correspondences [16]. Note that for this approach to be successful, the 3D structure must be recovered with high accuracy. For this reason, Royer *et al.* resort to iterative refinement of the camera position and orientation with the current visible 3D points (local bundle-adjustment). This step is the most computationally intensive and avoiding it significantly speeds up the algorithm. In the case of Nister *et al.* , they use of a firewall to reduce error propagation with the drawback of reducing the number of landmarks available for triangulation.

We propose decoupling the estimation of the camera orientation and position without implementing bundle adjustment or firewall. First, the following observations are made. In general, and especially in urban environments, the number of landmarks with known 3D points is much lower than the actual number of correspondences. This is because

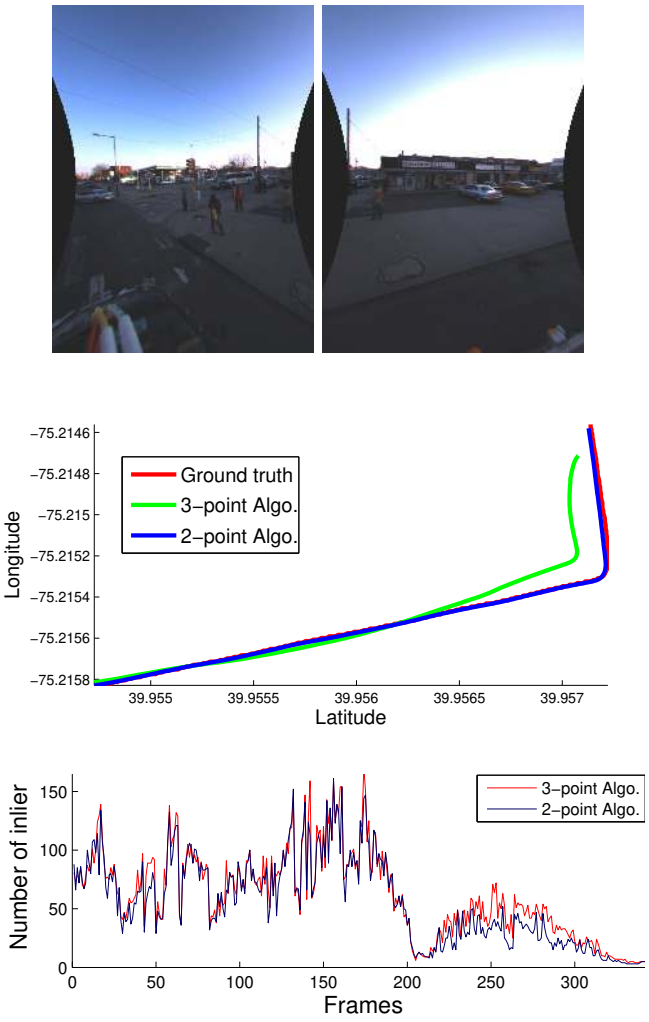

Fig. 2. Comparison between using a 3-point algorithm for pose estimation and the proposed approach of obtaining the relative orientation using 2-view epipolar geometry and camera position with the 2-point algorithm. **Top)** Two images of location where the number of landmarks decreases significantly. **Middle)** Camera trajectories with respect to ground truth given by the GPS (350 meters). **Bottom)** Inliers used for pose estimation.

triangulation can be accurately performed only for the part of the scene located on the side of the vehicle. Furthermore, we observe that the quality of the reconstruction is typically quite low for a large portion of the 3D points, since landmarks only appear in a few frames before being occluded. Typically, the number of available 3D points varies between 20 and 300. On the other hand correspondences between two views are much more abundant, typically between 500 and 1500, and more uniformly distributed. Thus, it seems natural that the epipolar constraint should not be "ignored" while estimating the pose from the 3D points.

As a matter of fact, epipolar geometry already provides the relative position of the new camera up to one degree of freedom: only the direction of the translation is estimated [17]. In theory, only one 3D point is required to recover that scale. In doing so, the estimated camera position is consistent with both 3D points and the epipolar geometry. One may

argue that the recovered translation direction can be of poor quality, especially under small camera motion. This is indeed what we experienced in practice. For this reason, instead of only estimating the translation scale, we actually estimate the full camera position while fixing its orientation. Solving for this is straightforward and requires only 2 (1 and a half) 3D-2D correspondences. Preemptive RANSAC is performed followed by iterative refinement.

Figure 2 shows a 350 m. trajectory estimated using both a 3-point and the 2-point algorithm, as well as the ground truth given by our GPS. In this sequence, the number of landmarks is significantly reduced around frame 200, resulting in a drift in the 3-point algorithm. To better understand why our pose estimation algorithm is less sensitive to drift we analyzed the inlier count given by the two algorithms. In figure 2, we observe that when drifting occurs, the number of inliers for a 3-point algorithm is on average higher than that for the 2-point algorithm. This means that the drift is caused by the use of some of the 3D points whose tracks are consistent with the epipolar geometry but whose 3D points are erroneous.

F. Triangulation

As explained above, structure computation is only needed to estimate the camera position, but not its orientation. A small set of points of high quality is thus preferred over larger one of lower quality. Thus, our triangulation procedure is conservative. When only two landmarks are available, we rely on a fast close-form two-view triangulation algorithm [28]. Otherwise, we perform N-view triangulation using the DLT algorithm [17]. Any attempt of relying on only a two view triangulation algorithm, including the one minimizing the re-projection error [18], gave results of lower quality. No iterative refinement of the 3D points using the re-projection error is used in our experiments.

We propose a simple heuristic to select the cameras used for triangulation. We do so in order to distribute back-projection rays as evenly as possible. Our goal here is twofold. Firstly, we make sure that 3D points far away from the camera are never used for estimating the position of the camera. However, their corresponding 2D landmarks are still used for estimating the camera orientation as explained earlier. Secondly, as the distance of a 3D point from the current camera increases, it is not re-triangulated over again.

We proceed as follows. For each landmark l, we keep a list of indices camList_l corresponding to the cameras used for triangulation. When a landmark first appears in two frames, denoted I_n and I_{n+1}, we add $\{n\}$ to camList_l. Note that $n+1$ is not yet added. We then estimate the 3D points using camera n and $n+1$ and compute the angle between the back-projection rays with respect to the 3D points. If the angle is larger than a given threshold, θ_{min}, the 3D point is kept and $n+1$ is added to camList_l. When performing triangulation for a landmark observed before frame f, we proceed similarly. We compute a 3D point using camList_l as well as the current camera. We then compute the angle of the emerging rays from the camera corresponding to the last element of camList_l and the current one. Again, if the angle is too small, this newly triangulated point is rejected and the one previously estimated is kept. At any of these steps, if the re-projection error is higher than our threshold, we completely reject the 3D point.

Fig. 3. **Top)** Ladybug 2 camera mounted on a vehicle. **Bottom)** Six images acquired from the six cameras, after radial distortion correction.

V. Experiments

A. System configuration

We tested our system using a Pointgrey Ladybug 2 camera mounted on a vehicle as illustrated in figure V-A. The SDK was used to rectified the six fish eye cameras and also provided the relative orientation of the cameras. Each image has a resolution of 768×1024. Only the first five were actually used. They also show a large amount of clutter.

B. results

Two sequences, respectively of 1 km and 2.5 km, and both containing loops were tested. The frame rate of the camera during each experiment was set to 3 and 10 images per second. A GPS unit was used and synchronized with the image acquisition. It was used to test the accuracy of the

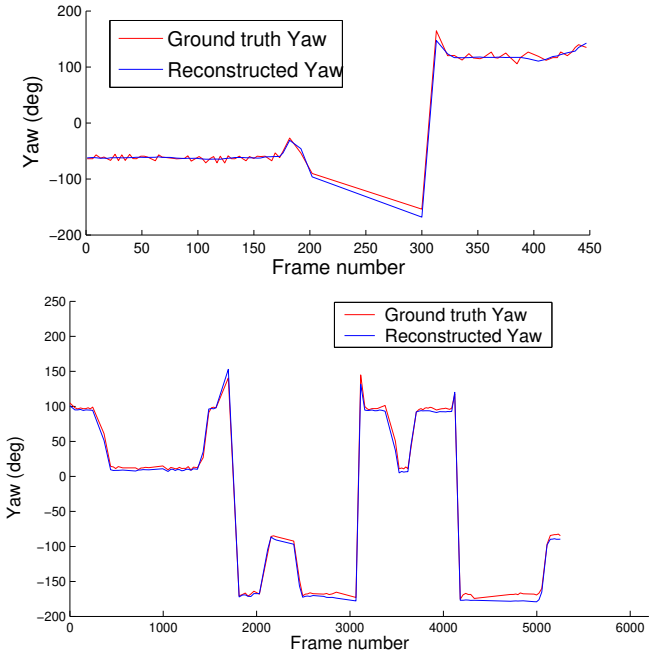

Fig. 4. Comparison between GPS-estimated yaw versus reconstructed yaw from visual odometry. **Top** 1 km sequence. **Bottom** 2.5 km sequence.

TABLE I

TOTAL DISTANCE IN METERS FOR THE TWO SEQUENCES.

Sequence	GPS dist.	Reconstructed dist.	error %	# of frames
1	784	763	2.86	450
2	2434	2373	2.47	5279

recovered vehicle trajectories by aligning them to the GPS coordinates. However, it sometime gave inaccurate results as observed at the end of the km sequence (figure 5a). Observe that the estimated trajectory is properly aligned with the street. Results for both sequences are shown in figure 5. In figure 6, a close up of the camera trajectory and projected 3D points on the map and figure 7 shows the 3D map estimated as well as the camera trajectory.

As a second measure of accuracy, we also compute the yaw of the vehicle at every frame. However, no IMU was available during our tests and an approximate ground truth yaw was computed from the GPS. The results are shown in figure 4 and visual inspection shows similar results. Finally, the total distances of the trajectories computed from GPS and visual odometry are given in table I. Note that for the 1km sequence, we compared the yaw and total distance only for the part where the GPS was accurate.

VI. CONCLUSION

We presented a system for motion estimation of vehicle using a Ladybug 2. The most important difference with prior art is the decoupling of the rotation from translation for the estimation of the pose. Rotation is estimated using robust computation of the epipolar geometry, and only the camera position is estimated using the 3D points.

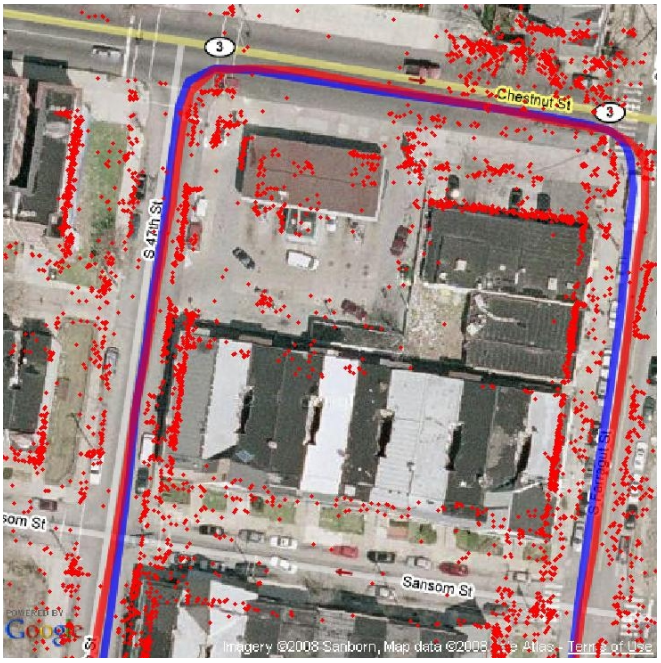

Fig. 6. 3D points overlay on the satellite image for the 2.5 km sequence (figure 5b)

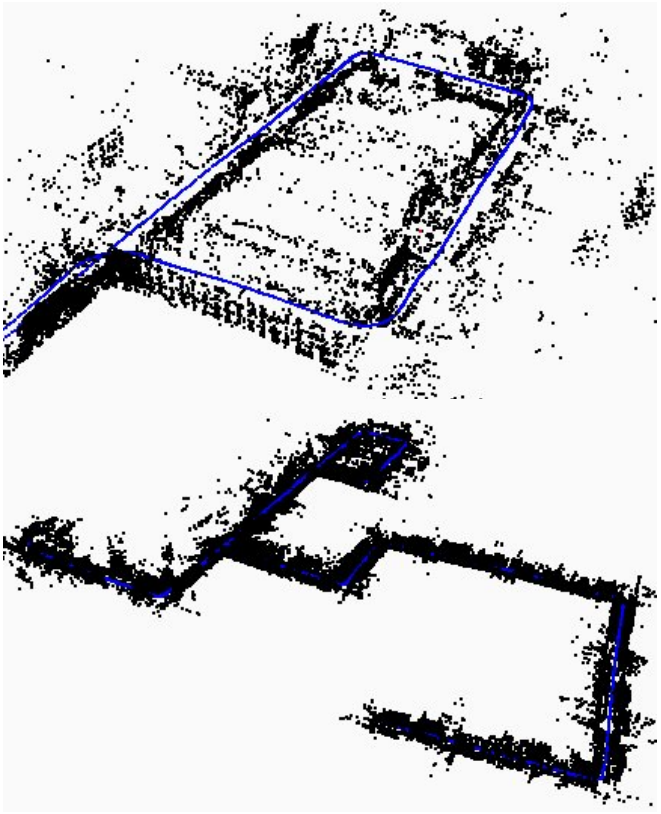

Fig. 7. Recovered 3D map (black) and camera position (blue) for the 2.5 km sequence. **Top)** Complete model. **Bottom** Zoom of the area show in figure 6

(a) (b)

Fig. 5. Comparison between the trajectory estimated by visual odometry and the GPS. **a)** 1 kilometer sequence. Notice the erroneous GPS coordinate towards the end of the sequence. **b)** 2.5 kilometer sequence. Overlay circle corresponds to close-up show in 6.

The results were quantitatively compared to ground truth GPS on sequences of up to 2.5 km, showing high accuracy.

REFERENCES

[1] T. D. Barfoot. Online visual motion estimation using fastslam with sift features. *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 579–585, 2005.

[2] Paul A. Beardsley, Philip H. S. Torr, and Andrew Zisserman. 3d model acquisition from extended image sequences. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume II*, pages 683–695, London, UK, 1996. Springer-Verlag.

[3] T. Bonde and H. H. Nagel. Deriving a 3-d description of a moving rigid object from monocular tv-frame sequence. In J.K. Aggarwal & N.I. Badler, editor, *Proc. Workshop on Computer Analysis of Time-Varying Imagery*, pages 44–45, Philadelphia, PA, April 5-6, 1979.

[4] M. Brand, M. Antone, and S. Teller. Spectral Solution of Large-Scale Extrinsic Camera Calibration as a Graph Embedding Problem. *Computer Vision, ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004: Proceedings*, 2004.

[5] T.J. Broida and R. Chellappa. Estimation of object motion parameters from noisy image sequences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:90–99, 1986.

[6] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):523–535, 2002.

[7] P. Corke, D. Strelow, and S. Singh. Omnidirectional visual odometry for a planetary rover. *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 4, 2004.

[8] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1052–1067, 2007.

[9] A.J. Davison and D.W. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, 2002.

[10] E.D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1:223–240, 1988.

[11] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part i the essent ial algorithms. *Robotics and Automation Magazine*, 13:99¡80¿¡93¿110, 2006.

[12] E. Eade and T. Drummond. Scalable Monocular SLAM. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 1, 2006.

[13] P. Elinas, R. Sim, and J. J. Little. sslam: Stereo vision slam using the rao-blackwellised particle filter and a novel mixture proposal distribution. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, page 1564–1570, 2006.

[14] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[15] U. Frese. A discussion of simultaneous localization and mapping. *Autonomous Robots*, 20:25–42, 2006.

[16] Haralick, Lee, Ottenberg, and Nõlle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13:331–356, December 1994.

[17] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[18] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68:146–157, 1997.

[19] A. Heyden and F. Kahl. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point, June 1997.

[20] M. Kaess and F. Dellaert. Visual SLAM with a Multi-Camera Rig. Technical report, Georgia Institute of Technology, 2006.

[21] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and ME Munich. The vSLAM Algorithm for Robust Localization and Mapping. *Robotics and Automation, 2005. Proceedings of the 2005 IEEE International Conference on*, pages 24–29, 2005.

[22] K. Konolige, M. Agrawal, R.C. Bolles, C. Cowan, M. Fischler, and BP Gerkey. Outdoor mapping and navigation using stereo vision. *Intl. Symp. on Experimental Robotics*, 2006.

[23] A. Levin and R. Szeliski. Visual odometry and map correlation. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004.

[24] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[25] B.D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Int. Joint Conf. on Artificial Intelligence*, pages 674–679, 1981.

[26] Y. Ma, J. Kosecka, S. Soatto, and S. Sastry. *An Invitation to 3D Vision*. Springer Verlag, 2003.

[27] A. I. Mourikis, N. Trawny, S.I. Roumeliotis, A.E. Johnson, and L.H. Matthies. Vision-aided inertial navigation for precise planetary landing: Analysis and experiments. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, June 2007.

[28] D. Nister. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26:756–770, 2004.

[29] D. Nistér. Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16:321–329, 2005.

[30] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23:3–20, 2006.

[31] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *Int. J. of Computer Vision*, 59(3):207–232, 2004.

[32] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2, 2005.

[33] E. Royer, M. Lhuillier, M. Dhome, and J. M. Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74:237–260, 2007.

[34] C. Silpa-Anan and R. Hartley. Visual localization and loop-back detection with a high resolution omnidirectional camera. *Workshop on Omnidirectional Vision*, 2005.

[35] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294, 2006.

[36] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master. Calibrated, Registered Images of an Extended Urban Area. *International Journal of Computer Vision*, 53(1):93–107, 2003.

[37] S. Thrun. Robotic mapping: a survey. In *Exploring artificial intelligence in the new millennium*, pages 1–35. Morgan Kaufmann, Inc., 2003.

[38] P. H. S. Torr, A. W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32:27–44, 1999.

Acknowledgement. We would like to thank Friedrich Fraundorfer (Swiss Federal Institute of Technology Zurich) for providing his vocabulary tree implementation. The support by the following grants is gratefully acknowledged: NSF-EIA-0324977, NSF-IIS-0713260, NSF-IIP-0742304, ARO/MURI DAAD19-02-1-0383, ARLCTA DAAD 1901-2-0012.