AUTONOMOUS VISUAL NAVIGATION FOR PLANETARY EXPLORATION ROVERS

Manolis Lourakis, Georgios Chliveros, and Xenophon Zabulis

Institute of Computer Science
Foundation for Research and Technology - Hellas (FORTH)
P.O.Box 1385, GR-711 10, Heraklion, Crete, GREECE
{*lourakis,chlivero,zabulis*} *at ics.forth.gr*

ABSTRACT

SPARTAN (SPAring Robotics Technologies for Autonomous Navigation) and its extension SEXTANT (Spartan EXTension Activity - Not Tendered) are two robotic exploration technology development activities funded by the European Space Agency (ESA). They target the development of computer vision algorithms for visual navigation that will be suitable for use by Martian rovers. This paper summarizes our on-going efforts in the context of SEXTANT for developing dependable and efficient solutions for two key ingredients of visual navigation, namely terrain mapping and localization.

Key words: Terrain Mapping, Visual Odometry, Visual Localisation, StereoSLAM.

1. INTRODUCTION

Planetary exploration scenarios require that the optimal route between two far-away locations is computed in several iterations and with multiple intermediate waypoints since perfect knowledge of the environment is generally not available. Thus, a rover is asked to navigate in an unknown terrain several hundreds of meters away from a starting position. Therefore, it must perceive its surroundings, determine the best trajectory and execute it until reaching the desired target whilst maintaining an estimate of its current position. At each navigation cycle, sizeable volumes of data need to be processed with as little computing power, memory footprint and communication overhead as possible. Two key elements of visual navigation are terrain mapping and localization.

Terrain mapping concerns the use of several stereo images for incrementally producing a three-dimensional map of the environment that will be utilized for obstacle avoidance and path planning. Mapping makes extensive use of dense stereo matching and subsequent merging of partial maps. Localization, on the other hand, refers to the use of imagery for updating a feature-based map, whilst maintaining correct position estimates within locally maintained sub-maps. Localization resorts to visual simultaneous localization and mapping (vSLAM) techniques, using as priors motion estimates computed by visual odometry. This is because commonly used sensory inputs for localisation priors are either not available in extraterrestrial environments (e.g. GPS) or are prone to highly erroneous output (e.g. IMU).

Visual Odometry (VO) refers to the process of estimating the egomotion (i.e. position and orientation) of a vehicle by analyzing onboard-camera images [1]. In broad terms, solutions to VO need to address three distinct sub-problems, namely feature detection, feature matching and motion estimation. Feature detection concerns the automatic extraction of sparse point features from a general scene, feature matching involves tracking them across a set of successive image frames and motion estimation regards the recovery of the relative pose of the employed camera(s) as well some partial scene 3D information using structure from motion algorithms. With respect to planetary exploration, most approaches use stereo or monocular cameras [2, 3, 4]. However, monocular cameras are not the most popular choice due to the well-known depth/scale ambiguity that prevents the recovery of absolute scale [5, 6]. Use of stereo cameras, on the other hand, permits the recovery of truly Euclidean 3D pose and scene structure and entails less need for keyframe selection. VO operates incrementally by computing the motion between consecutive frames and integrating it over time [1].

After extracting salient and repeatable feature points (e.g. Harris corners, MSERs) and subsequent descriptors (e.g. SIFT, SURF) from images, features are matched according to some similarity measure. However, matched features are usually contaminated by outliers due to erroneous data association caused by environmental phenomena such as image noise, occlusions, blur, clutter or viewpoint and illumination changes. How these outliers are removed, is of utmost importance for the quality of the motion estimates [7]. Recent field trials on Mars analogue environments by Bakambu et al. [8], have indicated that, on average, image regions have greater stability than local corner features. In a number of test site environments (e.g. sand-dunes, boulders, mudflat), maximally stable extremal regions (MSERs) outperform Harris and SIFT local features. However, MSERs correspond

to blobs of high contrast with respect to their surroundings, thus most past works use instead local features that respond to strongly textured areas in an image [9, 10, 11]. Furthermore, Tong and Barfoot [7] indicate that local feature descriptors (e.g. SIFT, SURF) provide increased performance in outlier rejection for motion estimation.

This paper focuses predominantly on the mapping and localization aspect via dense stereo and visual odometry estimation, respectively. By critically reviewing the published literature, we have selected mapping and localization building blocks whose performance characteristics fulfill the application requirements while at the same time are amenable to efficient implementations. The remainder of the paper is organized as follows. Sect. 2 presents our approach for dense stereo matching. Sect. 3 discusses the point features employed for representing image motion and Sect. 4 details their use for obtaining sparse 3D structure. Sect. 5 explains the estimation of egomotion from 2D image projections and reconstructed 3D points. Experimental results based on the use of synthetic data are reported in Sect. 6. The paper is concluded in Sect. 7.

2. DENSE 3D RECONSTRUCTION

Stereo matching is a fundamental problem in computer vision that despite having been the subject of intense study for more than thirty years, still remains an active area of research. The archetypal stereo problem is restricted to the use of only two images, a case also known as binocular stereo. Binocular stereo aims to establish pixel-wise, or otherwise dense, correspondences across the images of a stereo pair. Capitalizing on rectified epipolar geometry, this search for correspondence is restricted to corresponding scanlines in the two images. Binocular stereo algorithms can be classified as local or global. In local methods, the disparity computation at a given point depends only on intensity values within a small, local window. Global methods make explicit smoothness assumptions that involve solving a costly optimization problem to disambiguate potential matches. Compared to global methods, local ones exhibit only minor inaccuracies, are less computationally intensive and exhibit easily exploitable data parallelism. Therefore, they are often the preferred solution for robotic applications. A comprehensive review of binocular stereo approaches can be found in [12].

To obtain dense correspondences, the plane sweeping local stereo algorithm has been adopted in this work [13]. Plane sweeping is a general re-sampling algorithm that performs multi-image stereo matching with arbitrary camera configurations. It works by sweeping a set of hypothetical planes at increasing distances through the scene and measuring the photoconsistency of the synthetic images generated by back-projecting the input images onto these planes. Back-projection on the sweeping plane is achieved with the aid of the homographies it induces and does not require prior rectification of images. Photoconsistency is evaluated using normalized cross correlation (NCC). NCC can be efficiently computed by precomputing sums of squared and pixel-wise products of back-projected image intensities over the correlation kernel. The range of distances covered by the sweeping plane are set to bracket the working volume. Parabola fitting on the correlation profiles defined as the depth ranges over acceptable values is used to calculate disparities with subpixel accuracy.

Plane sweeping is attractive since it is amenable to parallel (i.e. GPU) implementation that can achieve real-time performance [14]. Besides, it has low memory requirements, which can become a critical issue on robotic platforms. Owing to these reasons, parallelized implementations of plane sweeping have been extensively employed in the real-time reconstruction of large-scale environments from binocular pairs mounted on mobile vehicles [15]. The accuracy of plane sweeping can be increased when the orientation of major structures in the scene, such as the ground plane, can be assumed known. Furthermore, its computational complexity can be directly modulated with respect to the precision of the obtained depth map, both regarding pixel resolution as well as depth precision. In this manner, it is possible to dedicate shorter computational times for a coarser reconstruction of the scene, thus obtaining an algorithm with anytime characteristics (an algorithm is said to be anytime when it can return a valid solution to a problem even if it is stopped before it normally ends). This feature is something which is not trivially feasible with other local stereo approaches.

3. FEATURE EXTRACTION AND MATCHING

During the past decade, significant progress has been made in the development of local invariant features. These features permit the detection of local image structures in a repeatable fashion and their encoding in a way invariant to various image transformations. However, despite their robustness, local feature detectors often entail considerable computational overhead. This seriously limits their applicability on planetary rovers, due to the limited computational capacity of the later. On the other hand, since image acquisition is frequent, the distortions between successive images are not expected to be large and, therefore, can be accommodated by simpler and hence faster to compute interest point detectors.

In light of these considerations, features are detected in this work with the Harris corner detector [16]. Harris, also known as Plessey, is a popular interest point detector that is based on the local auto-correlation function (i.e. intensity variation) of an image. The local auto-correlation function measures the local changes of the image using patches shifted by a small amount in different directions around a point. The shifted patches are approximated by a Taylor expansion truncated to the first order terms, which gives rise to a 2×2 matrix known as the structure tensor. The eigenvalues of this matrix capture the intensity structure of a point's local neigh-

borhood. More specifically, when both eigenvalues are larger than some threshold, a corner is present in the image. This is because the eigenvalues are proportional to the principal curvatures of the image surface, therefore shifts in any direction result in significant change. Various corner strength (i.e. "cornerness") measures have been proposed, avoiding the costly explicit computation of the eigenvalues [17].

The Harris detector is generally considered as the best operator available with respect to detecting true corners. This is because it behaves very well with respect to detection and has a high repeatability rate. Its implementation involves separable 2D convolutions, therefore it can easily be implemented on hardware. To improve the spatial distribution of the detected Harris corners, the adaptive non-maximal suppression (ANMS) scheme of Brown et al. [18] has been employed, efficiently implemented as suggested in [19]. This scheme retains only those corners whose strength is locally maximal (i.e., in a neighbourhood of radius r pixels).

For each feature point, the local image appearance in its vicinity is captured using the BRIEF descriptor [20]. BRIEF (short for Binary Robust Independent Elementary Features) is an efficient feature point descriptor based on binary strings extracted directly from image patches. BRIEF is based on performing several pair-wise intensity comparisons on an image patch and encoding the comparison outcomes using a bit vector. It was inspired by earlier work that achieved effective recognition of patches seen from different viewpoints by using a relatively small number of pair-wise intensity comparisons to train randomized classification trees. BRIEF abandons the randomized tree and simply creates a bit vector from the test responses. The spatial locations of the pixels compared by BRIEF in each patch are selected at random.

BRIEF descriptors are compared using the Hamming distance, which counts the number of positions at which the corresponding bit strings differ. The Hamming distance of two binary strings a and b is equal to the number of ones (i.e. population count) in a XOR b, which can be computed very efficiently [21]. Compared to more elaborate descriptors such as SIFT [22], BRIEF is less discriminant but much faster to compute and match. Furthermore, it is robust to lighting changes, blur, and perspective distortion. Despite not being designed to be rotationally invariant, BRIEF can tolerate small amounts of in-plane rotation. A truly rotation invariant extension of BRIEF is proposed in [23]. The stability and repeatability of BRIEF descriptors is increased by smoothing the image patches with a Gaussian of $\sigma = 1.5$ prior to their computation. In our implementation, image patches were 53×53 and 512 binary tests were performed in each, giving rise to BRIEF descriptors that were 64 bytes long.

Prior to estimating 3D motion, 2D motion of feature points has to be determined by matching them across images. Matching of BRIEF descriptors is performed using the distance ratio test originally proposed for matching SIFT descriptors [22], which proceeds as follows. Given an image pair, matches are identified by finding the two nearest neighbors of each keypoint from the first image among those in the second, and only accepting a match if the distance to the closest neighbor is less than a fixed threshold of that to the second closest neighbor. This threshold can be adjusted to leniently establish more matches, or conservatively select the most reliable ones. To make the matching more discriminative, a maximum disparity and epipolar line distance limit should be satisfied in addition to the distance ratio being sufficiently small.

At regular time intervals, stereo image pairs are acquired. The procedure described above can be used to establish point correspondences between images within the same stereo pair or between images from stereo pairs taken at consecutive points in time. In the following, such correspondences will be referred as spatial and temporal matches, respectively.

4. SPARSE 3D RECONSTRUCTION

Feature detection and matching between the two stereo views captured at a certain moment in time yields a set of spatial matches. Knowledge of the stereo calibration parameters, allows the estimation via triangulation of the 3D points giving rise to these spatial matches. Triangulation recovers 3D points as the intersections of back-projected rays defined by the matching image projections and the camera centers. Since there is no guarantee that back-projected rays will actually intersect in space (i.e. they might be skew), matched image points should be refined prior to triangulation so as to exactly satisfy the underlying epipolar geometry. This is achieved by computing the points on the epipolar lines that are closest to the original ones. The computation involves minimizing the distances of points to epipolar lines with a non-iterative scheme that boils down to solving a sixth degree polynomial [24]. Since this is rather costly in terms of computation, we employ an approximate but much cheaper alternative relying on the Sampson approximation of the distance function [6, 25].

As the rover moves, temporal matches between the left images of the stereo pairs acquired at times t and $t + 1$ induce correspondences among the known 3D points reconstructed in stereo at time t and their 2D projections at time $t + 1$. These correspondences are illustrated in Fig. 1 and suffice to estimate the relative motion of the stereo rig between times t and $t + 1$, as will be detailed in the following section.

5. POSE ESTIMATION

Pose estimation concerns determining the position and orientation of a camera given its intrinsics and a set of n correspondences between known 3D points and their 2D

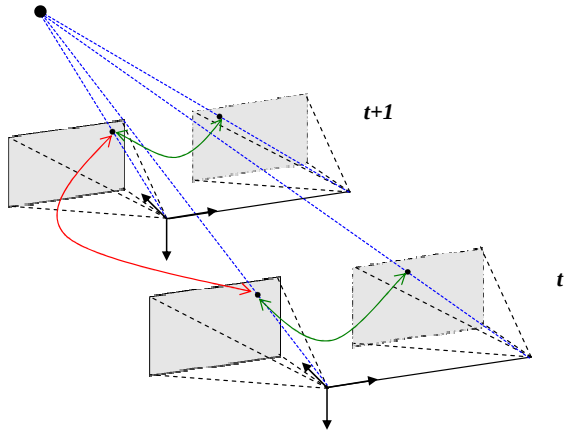


Figure 1: Establishment of 2D-3D correspondences for a moving stereo rig. Projection rays are shown in blue, spatial matches are indicated with green arrows and temporal matches with red. Pose estimation relies on the 3D points reconstructed at time t and their image projections at time $t+1$.

image projections. This problem, also known as camera resectioning or the Perspective-n-Point (PnP) problem, has received much attention due to is wide applicability in various domains. PnP is typically solved using non-iterative approaches that involve small, fixed-size sets of 3D-2D correspondences. For example, the basic case for triplets ($n = 3$ thus known as the P3P problem), was first studied in [26] whereas other solutions were later proposed in [27, 28]. P3P is known to admit up to four different solutions, whereas in practice it usually has just two. As a result, a fourth point is used in practice for disambiguation. Minimal solutions to PnP are particularly important for estimating pose in a robust estimation framework, as the cardinality of each random sample is directly related to the total number of samples that need to be drawn in order to find a solution with acceptable confidence. On the other hand, being unable to combine more than the minimal number of correspondences, minimal solutions ignore much of the redundancy present in the data.

5.1. Monocular robust pose estimation with non-linear refinement

This section describes in more detail our approach for pose estimation from a single image. Starting with a set of 2D-3D point correspondences, a preliminary pose estimate is computed first and then refined iteratively. This is achieved by embedding a P3P solver into a RANSAC [27] framework that uses the MSAC redescending cost function for hypothesis scoring [29]. Applied to the problem of pose estimation, RANSAC repetitively draws random quadruples of points and uses one triple with the P3P solver of [26] and the fourth point for verification to obtain a pose estimate. The best scoring pose hypothesis is retained as RANSAC's outcome and used to classify correspondences into inliers and outliers. By minimizing the reprojection error pertaining to all inliers, the pose computed by RANSAC is next refined to take into account more than three correspondences. Since it involves a non-linear objective function, this minimization is carried out iteratively with the Levenberg-Marquardt (L-M) algorithm [30], as will be explained shortly.

Denoting by $\mathbf{K}$ the 3×3 intrinsic calibration matrix and n corresponding 3D-2D points by $\mathbf{M}_i$ and $\mathbf{m}_i$, the pose computed with RANSAC is refined by using it as a starting point to minimize the cumulative image reprojection error defined as

$$\min_{\mathbf{r},\mathbf{t}} \sum_{i=1}^{n} d(\mathbf{K} \cdot [\mathbf{R}(\mathbf{r}) \,|\, \mathbf{t}] \cdot \mathbf{M}_i - \mathbf{m}_i)^2, \qquad (1)$$

where $\mathbf{t}$ and $\mathbf{R}(\mathbf{r})$ are respectively the sought translation and rotation matrix parameterized using the Rodrigues rotation vector $\mathbf{r}$, $\mathbf{K} \cdot [\mathbf{R}(\mathbf{r}) \,|\, \mathbf{t}] \cdot \mathbf{M}_i$ is the predicted projection on the image of the homogeneous point $\mathbf{M}_i$ and $d(\mathbf{x}, \mathbf{y})$ denotes the reprojection error, i.e. the Euclidean distance between the image points represented by vectors $\mathbf{x}$ and $\mathbf{y}$. The Jacobians required by L-M were provided analytically by performing symbolic differentiation of the objective function in Maple and automatically generating source code for their computation. The formulation in (1) assumes that no gross outliers (i.e. mismatched) points exist among the employed data. This is because the employed squared distance allows a single very erroneous observation to have a devastating effect on the total reprojection error and hence to its minimizer. In practice it is very difficult to guarantee the absence of outliers, therefore the non-linear refinement should be applied after RANSAC to ensure that the influence of outliers is mitigated.

5.2. Binocular pose refinement

Estimating the pose as described in Sect. 5.1 employs a single image. To improve accuracy with little additional overhead, a second image can be employed and the estimation can be extended to the binocular case by combining the reprojection error in two images. More specifically, assuming that two calibrated cameras are available, monocular pose estimation is carried out independently for each image as in Sect. 5.1. Knowledge of the camera extrinsic calibration parameters allows the pose of one of the cameras (e.g. right) to be related to that of the other (e.g. left). Indeed, if the pose of the left camera is defined by $\mathbf{R}$ and $\mathbf{t}$, the pose of the right camera equals $\mathbf{R_s}\mathbf{R}$ and $\mathbf{R}_s\mathbf{t} + \mathbf{t}_s$, where $\mathbf{R_s}$ and $\mathbf{t_s}$ correspond to the pose of the right camera with respect to the left. Assuming a rigid stereo rig, $\mathbf{R_s}$ and $\mathbf{t_s}$ remain constant and can be estimated offline via extrinsic calibration. The most plausible left camera pose is determined via the minimization of the binocular reprojection error that consists of two additive terms, one for each image. Denoting the intrinsics

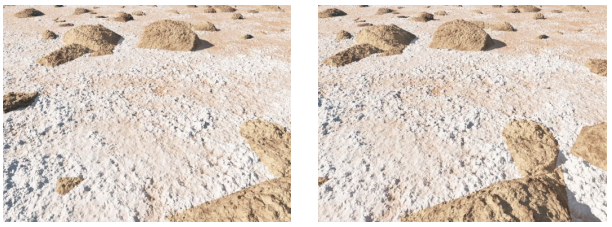


Figure 2: The first two frames from the simulated stereoscopic sequence. Image courtesy of Marcos Avilés, GMV.

for the left and right images by $\mathbf{K}^L$ and $\mathbf{K}^R$, the binocular reprojection error for n corresponding points in the left image and m in the right is defined as:

$$\min_{\mathbf{r},\mathbf{t}} \left(\sum_{i=1}^{n} d(\mathbf{K}^L \cdot [\mathbf{R}(\mathbf{r}) \,|\, \mathbf{t}] \cdot \mathbf{M}_i - \mathbf{m}_i^L)^2 + \sum_{j=1}^{m} d(\mathbf{K}^R \cdot [\mathbf{R}_s\mathbf{R}(\mathbf{r}) \,|\, \mathbf{R}_s\mathbf{t} + \mathbf{t}_s] \cdot \mathbf{M}_j - \mathbf{m}_j^R)^2 \right), \quad (2)$$

where $\mathbf{t}$ and $\mathbf{R}(\mathbf{r})$ are the sought translation and rotation, $\mathbf{K}^L \cdot [\mathbf{R}(\mathbf{r}) \,|\, \mathbf{t}] \cdot \mathbf{M}_i$ is the projection of homogeneous point $\mathbf{M}_i$ in the left image, $\mathbf{K}^R \cdot [\mathbf{R}_s\mathbf{R}(\mathbf{r}) \,|\, \mathbf{R}_s\mathbf{t} + \mathbf{t}_s] \cdot \mathbf{M}_j$ is the projection of homogeneous point $\mathbf{M}_j$ in the right image and $\mathbf{m}_i^L, \mathbf{m}_j^R$ are the 2D points corresponding to $\mathbf{M}_i$ and $\mathbf{M}_j$ in the left and right images, respectively. The minimization in Eq. (2) is performed with the L-M algorithm, employing only the inliers of the monocular estimations to ensure resilience to outliers.

It is noted that (2) circumvents the error-prone reconstruction of points via triangulation and does not limit the baseline of the two views nor calls for sparse feature or 3D point matching. It can also be extended to an arbitrary number of cameras. One possibility for initializing the minimization of (2) is to start it from the monocular pose computed for the left camera. However, this initialization does not treat images symmetrically as it gives more importance to the left image. Therefore, if the pose with respect to the left camera is erroneous, there is a risk of the binocular refinement also converging to a suboptimal solution. To remedy this, the refinement scheme is extended by also using the right image as reference and refining pose in it using both cameras, assuming a constant transformation from the left to the right camera. Then, the pose yielding the smaller overall binocular reprojection error is selected as the most accurate one.

6. EXPERIMENTAL RESULTS

The accuracy of the VO pipeline presented in Sections 3-5 was evaluated with the aid of a simulated stereoscopic sequence for which the ground truth egomotion is precisely known. Towards this end, a sequence of synthetic images with a resolution of 512×384 pixels and a field of view of $66^\circ \times 52^\circ$ was employed. The first stereo pair of this sequence is shown in Fig. 2. The stereo system had a baseline of $12\,cm$ and moved in a sufficiently textured simulated environment with a speed of $6\,cm$ per frame. The motion was predominantly in the forward direction combined with a shallow turn to the right. In the following, HARRIS+BRIEF will refer to Harris corners coupled with BRIEF descriptors. Moreover, due to the fact that SIFT features and descriptors comprise a very popular combination for feature extraction and matching [31], we have chosen to include the SIFT+SIFT detector/descriptor combination in the comparison. By doing so, we can directly quantify the impact on performance of a more powerful (but also more computationally expensive) combination. To demonstrate the improvements brought about by the binocular pose estimation of Sect. 5.2, HARRIS+BRIEF was also tested with the monocular pose estimation scheme of Sect. 5.1.

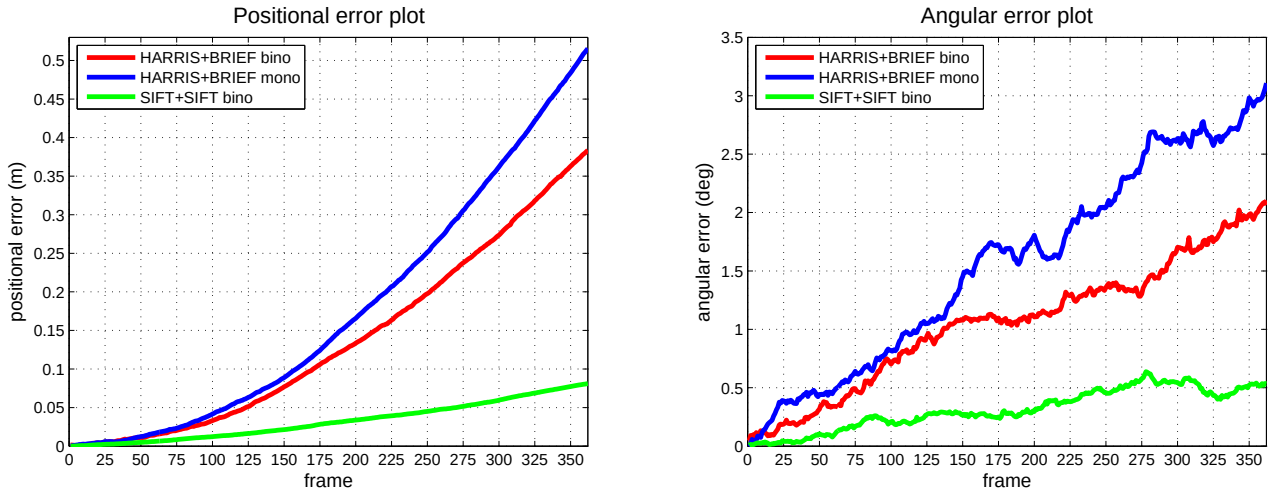
For each detector/descriptor and pose estimation choice, the camera motion was estimated for 363 stereo frames. This number of frames corresponds to a total travelled distance slightly less than $22\,m$ and a change in orientation of about 9.5° degrees. Since the remaining parameters involved in pose estimation were kept unchanged in all cases, all observed differences in performance should be attributed to the different detector/descriptor and pose estimation algorithm choices.

For each stereo pair, the incremental motion with respect to its previous pair was estimated and then transformed to the world coordinate system which was taken to coincide with that of the left camera in the first stereo pair. Given an estimated rotation $\widehat{\mathbf{R}}$ and translation $\widehat{\mathbf{t}}$, the error with respect to the true motion $\mathbf{R}$, $\mathbf{t}$ consists of a translational and rotational component, defined respectively as

$$||\mathbf{t} - \widehat{\mathbf{t}}||, \qquad \arccos\left(\frac{\mathrm{trace}(\mathbf{R}^{-1}\widehat{\mathbf{R}}) - 1}{2}\right). \quad (3)$$

Thus, the translational error is the Euclidean distance between translation vectors, whereas the rotation error corresponds to the amount of rotation about a unit vector that transfers $\mathbf{R}$ to $\widehat{\mathbf{R}}$.

Figures 3a and 3b illustrate respectively the translational and rotational errors pertaining to the binocular and monocular variants of HARRIS+BRIEF as well as to the binocular SIFT+SIFT. As can be clearly seen from them, binocular pose estimation with the HARRIS+BRIEF detector/descriptor combination comes at the cost of reduced accuracy compared to SIFT+SIFT. On the other hand, HARRIS+BRIEF has considerably lower computational requirements (e.g. it is about 32 times faster than SIFT+SIFT in our C code) and is amenable to efficient implementation for real-time performance, which makes it attractive for use in VO estimation. For example, our implementation of binocular VO with HARRIS+BRIEF runs at 4 fps on an Intel Core 3GHz without any special hardware optimizations (e.g. SSE). Furthermore, HARRIS+BRIEF yields a relative translational error that is



(a) Translational errors

(b) Rotational errors

Figure 3: Translational and rotational errors for the motion estimated with monocular HARRIS+BRIEF and binocular HARRIS+BRIEF & SIFT+SIFT. The well-known drift due to error accumulation over time is evident.

still less than 2%. It should also be noted that the errors for the binocular HARRIS+BRIEF are much lower compared to those for the monocular HARRIS+BRIEF combination, sharply exposing the improvements in accuracy gained by the binocular pose estimation scheme. It is stressed that all reported errors correspond to the raw output of VO, i.e. no attempt was made to temporally smooth the motion estimates with windowed bundle adjustment or similar procedure.

Sample dense reconstruction results from the application of plane sweeping with 101 depth planes and a 15×15 correlation kernel to the images of Fig. 2 are shown in Fig. 4.

7. CONCLUSION

The paper has presented our on-going efforts for vision-based mapping and localization solutions for use by Martian rovers. Future work will address the merging of partial dense reconstructions into larger environment representations and the incorporation of VO outputs as priors in a visual simultaneous localization and mapping (vSLAM) framework.

Acknowledgments: This work was partially supported by the ESA SPARTAN Extension Activity (SEXTANT) (ESA/ESTEC reference 4000103357/11/NL/EK).

REFERENCES

[1] D. Scaramuzza and F. Fraundorfer. Visual Odometry: Part I - The First 30 Years and Fundamentals. *IEEE Robot. Automat. Mag.*, 18(4):80–92, 2011.

[2] D. Nistér, O. Naroditsky, and J. Bergen. Visual Odometry for Ground Vehicle Applications. *J. Field. Robot.*, 23(1), 2006.

[3] M. Maimone, Y. Cheng, and L. Matthies. Two Years of Visual Odometry on the Mars Exploration Rovers: Field Reports. *J. Field. Robot.*, 24(3):169–186, 2007.

[4] M. Bajracharya, M.W. Maimone, and D. Helmick. Autonomy for Mars Rovers: Past, Present, and Future. *IEEE Computer*, 41(12):44–50, Dec. 2008.

[5] R. Szeliski and S.B. Kang. Shape Ambiguities in Structure from Motion. In *Proc. of ECCV'96*, volume I, pages 709–721, 1996.

[6] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[7] C.H. Tong and T.D. Barfoot. Evaluation of Heterogeneous Measurement Outlier Rejection Schemes for Robotic Planetary Surface Mapping. *Acta Astronaut.*, 2012.

[8] J.N. Bakambu, C. Langley, G. Pushpanathan, W.J. MacLean, R. Mukherji, and E. Dupuis. Field Trial Results of Planetary Rover Visual Motion Estimation in Mars Analogue Terrain. *J. Field. Robot.*, 29 (3):413–425, 2012.

[9] S. Se, T. Barfoot, and P. Jasiobedzki. Visual Motion Estimation and Terrain Modeling for Planetary Rovers. In *ESA 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2005.

[10] J. Wang. *Modeling and Matching of Landmarks For Automation of Mars Rover Localisation*. PhD thesis, Geodesic Science, The Ohio State University, 2008.

(a) Stereo reconstruction front view

(b) Stereo reconstruction top view

Figure 4: Sample views of the reconstruction obtained with plane sweeping applied to the images of Fig. 2.

[11] L. Matthies, M. Maimone, A. Johnson, Y. Cheng, R. Willson, C. Villalpando, S. Goldberg, A. Huertas, A. Stein, and A. Angelova. Computer Vision on Mars. *Int. J. Comput. Vis.*, 2007.

[12] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Int. J. Comput. Vis.*, 47(1-3):7–42, 2002.

[13] R.T. Collins. A Space-Sweep Approach to True Multi-Image Matching. In *Proc. of CVPR'96*, pages 358–363, 1996.

[14] R. Yang and M. Pollefeys. Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware. In *Proc. of CVPR'03*, pages 211–218, 2003.

[15] M. Pollefeys et al. Detailed Real-Time Urban 3D Reconstruction from Video. *Int. J. Comput. Vis.*, 78 (2-3):143–167, Jul. 2008.

[16] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proc. of the 4th Alvey vision conference*, pages 147–151, University of Manchester, UK, Sep. 1988.

[17] A. Noble. *Descriptions of Image Surfaces*. PhD Dissertation, Department of Engineering Science, Oxford University, UK, 1989.

[18] M. Brown, R. Szeliski, and S. Winder. Multi-Image Matching Using Multi-scale Oriented Patches. In *Proc. of CVPR'05*, pages 510–517, 2005.

[19] S. Gauglitz, L. Foschini, M. Turk, and T. Höllerer. Efficiently Selecting Spatially Distributed Keypoints for Visual Tracking. In *Proc. of ICIP'11*, pages 1869–1872, 2011.

[20] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proc. of ECCV'10*, pages 778–792, 2010.

[21] S.E. Anderson. Bit Twiddling Hacks, Counting bits set. [web page] `http://graphics.stanford.edu/~seander/bithacks.html#CountBitsSetParallel`, 1997. [Accessed on 14 Apr. 2013].

[22] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *Proc. of ICCV'11*, pages 2564–2571, 2011.

[24] R. Hartley and P. Sturm. Triangulation. *Comput. Vis. Image Und.*, 68(2):146–157, 1997.

[25] P.D. Sampson. Fitting Conic Sections to Very Scattered Data: An Iterative Renement of the Bookstein Algorithm. *CVGIP*, 18:97–108, 1982.

[26] J.A. Grunert. Das pothenotische Problem in erweiterter Gestalt nebst über seine Anwendungen in Geodäsie. *Grunerts Archiv für Mathematik und Physik*, 1841.

[27] M.A. Fischler and R.C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24:381–395, 1981.

[28] L. Kneip, D. Scaramuzza, and R. Siegwart. A Novel Parametrization of the Perspective-three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation. In *Proc. of CVPR '11*, pages 2969–2976, 2011.

[29] P.H.S. Torr and A. Zisserman. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Comput. Vis. Image Und.*, 78(1): 138–156, 2000.

[30] M.I.A. Lourakis. levmar: Levenberg-Marquardt Nonlinear Least Squares Algorithms In C/C++. [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, Jul. 2004. [Accessed on 14 Apr. 2013].

[31] S. Se, D.G. Lowe, and J.J. Little. Mobile Robot Localization and Mapping with Uncertainty Using Scale-Invariant Visual Landmarks. *I. J. Robotic Res.*, 21(8):735–760, 2002.