Proceedings of IMECE2007
2007 ASME International Mechanical Engineering Congress and Exposition
November 11-15, 2007, Seattle, Washington, USA

IMECE2007-41397

ROUGH-TERRAIN MOBILE ROBOT LOCALIZATION USING STEREOVISION

Annalisa Milella
Institute of Intelligent Systems for Automation
National Research Council
Via Amendola 122 D/O, 70126 Bari, Italy
milella@ba.issia.cnr.it

Giulio Reina
Department of Innovation Engineering
University of Salento
Via Monteroni, 73100 Lecce, Italy
giulio.reina@unile.it

ABSTRACT

Mobile robots are increasingly being used in high-risk rough terrain situations, such as reconnaissance, planetary exploration, safety and rescue applications. Conventional localization algorithms are not well suited to rough terrain, since sensor drift and the dynamic effects occurring at wheel-terrain interface, such as slipping and sinkage, largely compromise their accuracy. In this paper, we follow a novel approach for 6-DoF ego-motion estimation, using stereovision. It integrates image intensity information and 3D stereo data within an Iterative Closest Point (ICP) scheme. Neither a-priori knowledge of the motion and the terrain properties nor inputs from other sensors are required, while the only assumption is that the scene always contains visually distinctive features, which can be tracked over subsequent stereo pairs. This generates what is usually referred to as visual odometry. The paper details the various steps of the algorithm and presents the results of experimental tests performed with an all-terrain rover, proving the method to be effective and robust.

1. INTRODUCTION

In order for a mobile robot to navigate autonomously over long distances on uneven surfaces, a method for accurately tracking the pose of the robot is primarily needed.

Dead reckoning, based on data coming from wheel encoders, is a widely used localization method. This technique is easy to implement, and allows good short-term accuracy and very high sampling rate. However, dead reckoning systems are not well suited to long-range navigation and rough terrains, since they generally do not consider the physical characteristics of the vehicle and of the terrain it is traversing. Moreover, wheel slippage, sinkage, and sensor drift may cause errors that accumulate without bound over time unless an additional absolute localization system is employed for sporadic robot position updates [1, 2].

In this work, we follow a different approach, called visual odometry or ego-motion [3]. The basic idea of visual odometry is that of estimating the motion of the robot by tracking features of the environment detected with an on-board camera. Similarly to conventional dead reckoning, this technique can lead to error accumulation. However, since video sensors are exteroceptive devices, that is, they acquire information from the robot's environment, visual odometry is not affected by wheel slippage and sinkage. Moreover, it has been demonstrated that vision allows better results for most sensor combinations [4, 5]. Several visual odometry methods have been proposed in the last decades, using either single cameras [5, 6, 7] or stereo vision [4, 7, 8, 9], which mainly differ depending on the feature tracking method and the transformation applied for estimating the camera motion. For instance, in [4], odometry provides an estimation of the approximate robot motion that allows a search area to be selected for improved feature tracking, and a maximum-likelihood formulation is employed for motion computation. In [5], the visual module uses a variation of Benedetti and Perona's algorithm for feature detection, and correlation for feature tracking. Robustness is improved by integrating visual data with IMU using a Kalman filter. Finally, in [7], robust visual motion estimation is achieved using preemptive RANSAC [10], followed by iterative refinement.

In this paper, an algorithm for 6-DoF ego-motion estimation is proposed, which incorporates image intensity information and 3D stereo data in the well-known Iterative Closest Point scheme. ICP was originally introduced by Besl [11], for the registration of digitized data from a rigid object with an idealized geometric model. Here, the potentialities of ICP are investigated for the case of visual odometry using stereovision. Specifically, two basic problems of ICP are addressed: the susceptibility to outliers, and the failure when dealing with large displacements. As an extension of these issues, another drawback of ICP is its inability to segment input data [11]. Typical solutions use odometry information for

Figure 1: The rover Dune

predicting the displacement between consecutive frames and providing initial motion estimate before ICP registration [12]. Conversely, the method described here allows overcoming both problems, using the information deriving from a single stereo device, without previous knowledge of the motion. The only assumption is that the scene always contains visually distinctive features. The method can be summarized as follows. First, for each acquired stereo pair, a dense disparity map is generated; then, interesting pixel points are selected in the left image and only the visual landmarks with an associated high stereo-confidence level 3D point are retained. Potential matches between two consecutive frames are established using image intensity information, providing a first approximate estimation of the motion. Finally, ICP is applied for correction and refinement. A similar approach is employed in [13] for 3D simultaneous localization and modeling. Iterative methods combining intensity and 3D information can also be found in [14] for map building and in [15] for the registration of 3D partial surface models. This work focuses, instead, on the visual odometry issue. Various image processing and 3D registration techniques are efficiently combined for improving outlier rejection in both stereo matching and feature tracking, so that accurate motion estimates can be achieved, though using a few interesting points and preserving real-time constraints.

For the extensive testing of our visual odometry system, we used a rover that was built at the University of Salento and named Dune. The vehicle is shown in Figure 1. It is an independently controlled 4-wheel-drive/4-wheel-steer mobile robot with an envisaged operation speed ranging from 0.2 to 1 m/s. This configuration provides high mobility allowing the vehicle to perform special maneuvers, such as turn-on the spot and crab motion. The rover also features a four-wheel passive suspension system, commonly called rocker suspension. Details of rocker-bogie suspension characteristics can be found in [16, 17]. It ensures that all four wheels remain in contact with the ground all the time, despite one wheel moving higher or lower than the others, avoiding a very soft spring suspension. This minimizes the ground pressure at any one wheel while maximizing traction and the rover's ability to climb obstacles. The differential also serves to mediate the difference in ground terrain between both sides of the rover allowing the body of the rover to see only half of the disturbance, which is generally beneficial for the vision sensors. Dune is equipped with a Videre Design color digital stereo head with two ½" CMOS sensors using firewire interface, and four wheel encoders and steer potentiometers for conventional odometry.

Section 2 details the ICP-based visual odometry algorithm. Experimental tests, performed in both an indoor and an outdoor environment and showing the feasibility of our approach, are discussed in section 3, while section 4 concludes the paper.

2. VISUAL ODOMETRY USING ICP

In this section, we present our algorithm for 6-DoF ego-motion estimation. The method combines intensity and 3D information using an ICP-based frame. In order to apply ICP for localization purposes, two critical issues need to be addressed: the susceptibility to gross statistical outliers, and the failure when dealing with large displacements. As an extension of these issues, another drawback of ICP is addressed, i.e. its inability to perform the segmentation of input data points: if data points from two shapes are intermixed and matched against the individual shapes, registration fails [11]. These limitations are intrinsic in ICP basic concept and become particularly restrictive for robot self-localization and navigation purposes, as, while the vehicle moves, different parts of the scene become occluded and, conversely, new objects may appear. Therefore, vast regions may be present in only one of

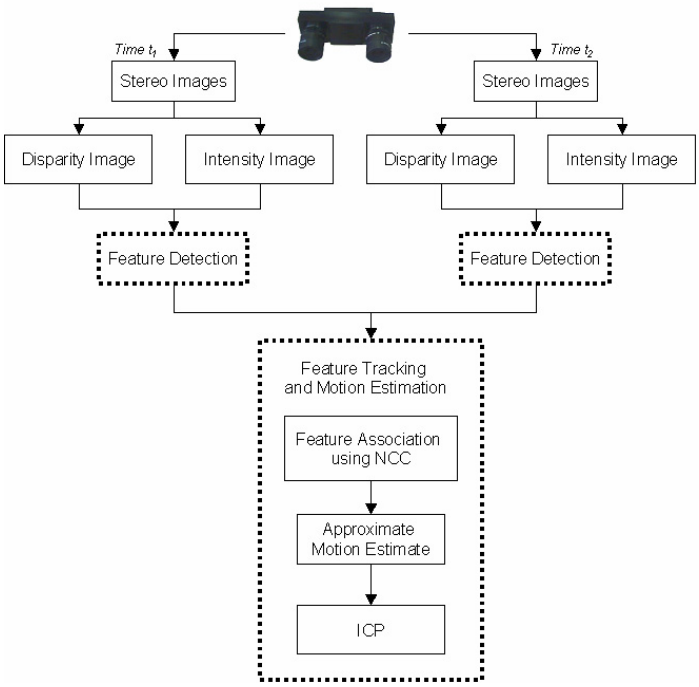

Figure 2: Block diagram of the visual odometry algorithm using two consecutive image pairs corresponding to time t_1 and t_2

two consecutive point clouds, and, if an outlier region is too close to a valid region, there is no possibility for ICP to perform a correct matching process [18].

Our method involves three main phases (see Figure 2): 1) Feature detection, 2) Feature tracking, and 3) Motion estimation. In the remainder of this section, each phase is discussed separately. Further theoretical details about the method can be found in [19].

Feature detection – Let us denote with $S_i=(I_{l,i}, I_{r,i})$ the stereo pair acquired at a given time t_i, being $I_{l,i}$ and $I_{r,i}$ the image produced by left and right camera, respectively. A dense disparity map is generated to obtain 3D points. The Stanford Research Institute Stereo Engine algorithm is employed [20]. It consists of an area correlation-based matching process, followed by a post-filtering operation that uses a combination of a confidence filter and left/right check to reject areas with insufficient texture, where bad matches are very likely to appear. The Shi-Tomasi feature detector [21] is then applied to the left image $I_{l,i}$ to select interesting pixel points. Only pixels with an associated high stereo-confidence level 3D point are retained for further processing. Two point clouds are in the end available for each stereo pair: the pixel point cloud and its associated 3D point cloud.

Feature tracking - The tracking of visual landmarks between two consecutive frames $S_i= (I_{l,i}, I_{r,i})$ and $S_{i+1}= (I_{l,i+1}, I_{r,i+1})$ acquired at time t_i and t_{i+1} respectively, is performed using a normalized cross-correlation (NCC) algorithm applied to the left image. The NCC allows determining the degree of similarity between two image portions f and w of dimension $L\times K$ by means of the coefficient C defined as

$$C=\frac{\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}\left(w(x,y)-\bar{w}\right)\cdot\left(f(x,y)-\bar{f}\right)}{\left[\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}\left(w(x,y)-\bar{w}\right)^2\right]^{1/2}\cdot\left[\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}\left(f(x,y)-\bar{f}\right)^2\right]^{1/2}} \quad (1)$$

where (x, y) represent the coordinates of an image point, $f(x, y)$ and $w(x, y)$ are the intensity value of f and w at the point (x, y), and $\bar{f}$ and $\bar{w}$ are the average intensity in f and w. C ranges between 0 and 1: the greater the value of C, the greater the similarity between f and w [22]. Based on this criterion, corresponding points are established as follows. Let us denote with L_i and L_{i+1} the two sets of visual landmarks detected in $I_{l,i}$ and $I_{l,i+1}$, respectively. Each point in L_i is paired with the point in L_{i+1} that generates the maximum normalized cross-correlation coefficient C in a 5×5 pixels window centered at the point. To speed up and improve the searching process, only features within a certain pixel distance from each other are matched. A minimum value for the correlation coefficient is also established. False matches are then rejected using two strategies: the mutual consistency check and robust statistics. The former consists in applying the cross-correlation-based pairing from both L_i to L_{i+1} and L_{i+1} to L_i; only pairs that mutually have each other as preferred mate are accepted as valid matches and are stored together with their correlation value. A final selection is accomplished based on the median and the standard deviation from median of the computed correlation coefficients; pairs whose correlation differs from the median by more than two times the standard deviation are rejected. Eventually, the tracking process leads to two main results: features that do not belong to both frames are discarded, i.e. the segmentation of input data is performed; two clouds of corresponding 3D points are produced, which will be employed in the successive motion estimation stage.

Motion estimation - The problem of estimating the motion that the camera has undergone between two consecutive stereo acquisitions can be expressed as finding the rotation matrix R and the translational displacement t that minimize the mean-squares objective function

$$F(R,t)=\frac{1}{N_p}\sum_{i=1}^{N_p}\left\|\left(R\bar{p}_i+t\right)-\bar{p}_{i+1}\right\|^2 \quad (2)$$

where $\bar{p}_i$ and $\bar{p}_{i+1}$ indicate corresponding 3D points at two successive time instants, and N_p is the number of pairs. Every optimization algorithm can be used to estimate least-squares rotation and translation. In our implementation, we use the dual number quaternion method [23], as it is an efficient solution for this kind of problem. The rotation matrix R is expressed in terms of a set of three independent Euler angles. In our system the *RPY* or *ZXY* convention is chosen. Figure 3 shows the three Euler angles ϕ, θ, and ψ, which are usually referred to as roll, pitch, and yaw, respectively. The motion estimate is, first, performed using the 3D correspondences found in the cross-correlation pairing process. Based on this estimate, the two 3D point clouds are aligned. Since, we can not be sure that all the correspondences found using NCC were correct and, therefore, we can not be certain about the accuracy of the estimated motion, we apply ICP for motion estimate refinement. Let us denote with $P_i=\{p_{i,1}, p_{i,2},\ldots, p_{i,N}\}$ and $P_{i+1}=\{p_{i+1,1}, p_{i+1,2},\ldots p_{i+1,M}\}$ two 3D point clouds. ICP allows finding corresponding points and estimating the motion between P_i and

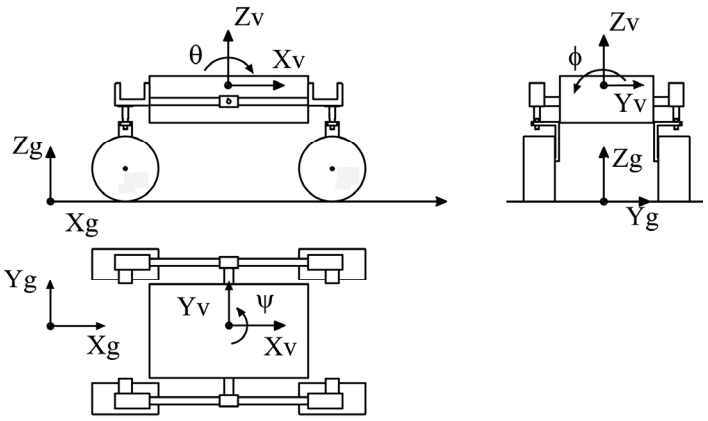

Figure 3: Dune's orientation using the RPY Euler angles

P_{i+1}, based on the following iterative scheme: *1)* associate each point of P_i with its closest point in P_{i+1}; *2)* discard false matches, based on a predefined rejection principle; *3)* estimate the motion; *4)* align the point clouds using the computed transformation; *5)* repeat steps *1* to *4*, until a convergence criterion is satisfied.

In our implementation, the Euclidean distance is used as metric for point association. Specifically, each point $p_{i,j}$ is associated to the point $p_{i+1,k}$ that satisfies the condition

$$\left\| p_{i,j}, p_{i+1,k} \right\| = \min_{k=1,2,\ldots,M} \left\| p_{i,j}, p_{i+1,k} \right\| \quad (3)$$

To discard false matches, the rejection scheme proposed by Zhang [24] is employed. It allows setting adaptively the value of the maximum distance between corresponding points using the statistics of the distances, namely the mean and the sample deviation. Least-squares rotation and translation are computed using the dual number quaternion approach. The process stops when the change in motion estimate between two successive iterations is less than 1%.

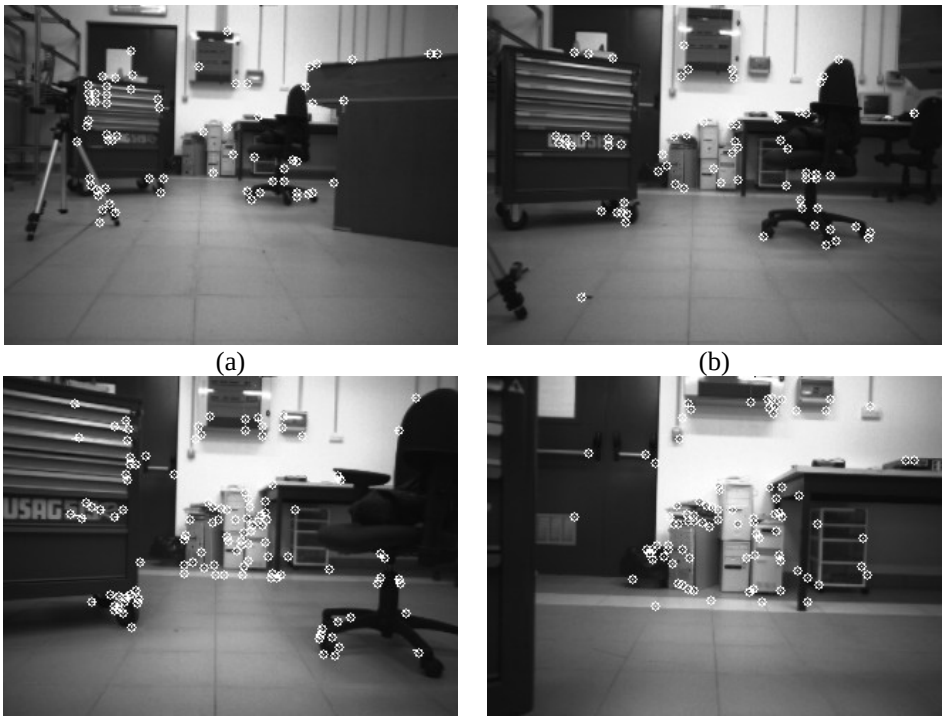

Figure 4: Sample frames during a typical run along a 3-m straight path: (a) 171st frame; (b) 471st frame; (c) 671st frame; (d) 1071st frame. The white markers denote the tracked features

3. EXPERIMENTAL RESULTS

The visual odometry algorithm was validated on real scenes using our all-terrain rover Dune (see Figure 1). Experimental trials were performed in both an indoor and an outdoor environment. In laboratory experiments, visual odometry was applied during a straight-line maneuver and the negotiation of a ramp. An experiment in the field was also performed driving the rover along a closed path on non-homogenous agricultural terrain with embedded rocks where the vehicle experienced substantial slip, tilt, and roll during the test. In all the experiments, the vehicle was remotely controlled using a joystick, and driven with a constant travel speed of 15 cm/s.

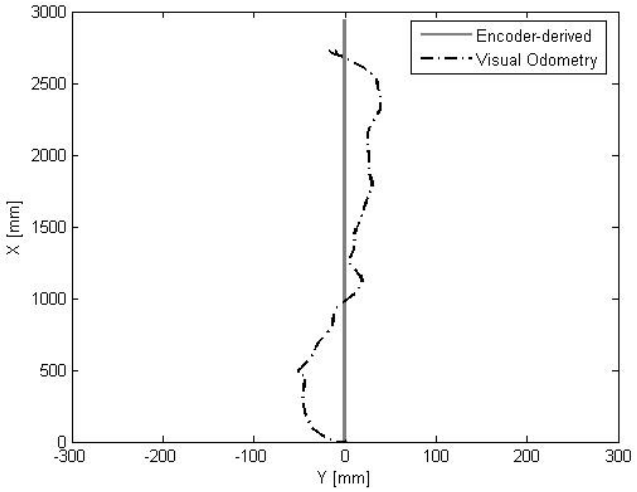

Figure 5: Path estimated by visual odometry compared with encoder-derived data for a test on flat surface

Indoor environment: straight path

We performed a set of experiments along a 3m straight path on flat surface. Figure 4 shows some frames captured during a typical test. Feature matches between consecutive frames are displayed using white circles and lines. White circles denote the current position of the matched features whereas the lines indicate how each feature has moved from one frame to the next, in a way similar to sparse optic flow representation. Note that, as the vehicle was driving straight, the lines in the circles may appear as single points. The path obtained from visual odometry in the same test is shown in Figure 5 compared with the encoder-derived path, which we can consider reliable and accurate on smooth surface and short distance. The final vehicle position as estimated by encoders and vision is also reported in Table 1. The test was repeated five times. For each repetition, a relative percentage error was calculated as

$$E_r^i = \frac{\sqrt{\left(x_o^i - x_{vo}^i\right)^2 + \left(y_o^i - y_{vo}^i\right)^2}}{\sqrt{\left(x_o^i\right)^2 + \left(y_o^i\right)^2}} \times 100 \quad (4)$$

where (x^i_o, y^i_o) and (x^i_{vo}, y^i_{vo}) denote the final robot position at the *i-th* run as estimated from odometry and visual odometry, respectively. This error also serves as an indication of error

	X [mm]	Y [mm]	Z [mm]	Yaw [°]	Pitch [°]	Roll [°]
Odometry	2944	0	_	0	_	_
Visual Odometry	2715	-17.3	28.1	5.0	2.0	0.2

Table 1: Result of visual and encoder-based odometry for a 3-m straight line run on flat surface

accumulation. In all experiments, E^i_r was less than 8.0%. Since, the algorithm was able to detect correct matches in all frames, such discrepancy can be mainly attributed to camera calibration errors.

Indoor environment: ramp

In this experiment, the robot was driven along a wooden ramp for a total length of 0.8m along the global *x*-axis and a maximum height of 0.15m. The test was repeated five times. Figure 6 shows the path of the robot in the *x*-*z* plane for a typical run. Table 2 reports the final position of the vehicle as estimated by the visual odometry algorithm and the actual position measured by a measuring tape, for the same test. In all tests, the relative percentage error, defined as in Eq. (2) using *x*-*z* coordinates in place of *x*-*y* coordinates, was less than 8.0%.

Outdoor environment

This test was performed in the field with the robot moving on uneven agricultural terrain along a closed path, resulting in a total travel distance D of 15m, and a total of 360 degrees of turning. Figure 7 shows the robot during operation.

Dune started at a marked location (0, 0) and, at the end of the test, was stopped at the same position. The discrepancy between the start/stop position and the estimated one derived from visual odometry is the so-called return position error. A short image sequence captured during the experiment is reported in Figure 8, showing a left turn followed by a right turn. In all the images, white markers are used to indicate the tracked features. The result of motion estimation is reported in Figure 9. Specifically, Figure 9(a) and (b) show the variation of the six DoF of the robot in terms of (*x*, *y*, *z*) coordinates and

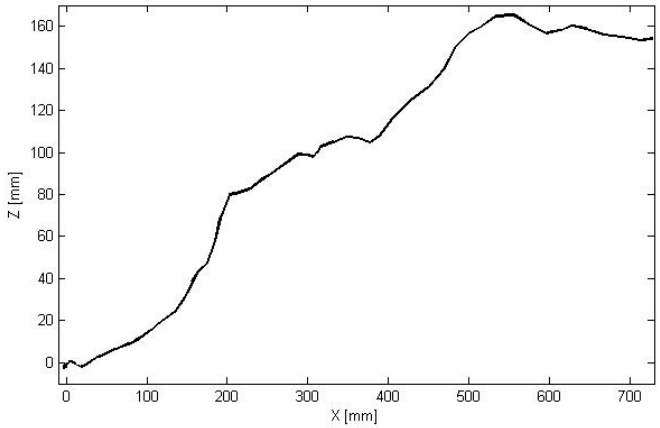

Figure 6: Path estimated by visual odometry for a test on a ramp

	X [mm]	Y [mm]	Z [mm]	Yaw [°]	Pitch [°]	Roll [°]
Actual Position	800.0	0	150.0	0	0	0
Visual Odometry	743.5	-55.9	156.2	-1.9	-0.3	-0.1

Table 2: Result of visual odometry compared with ground-truth measures for a test on a ramp

Euler angles, respectively. Figure 9(c) displays, instead, the estimated path in the *x*-*y* plane. Denoting with (x_{vo}, y_{vo}) the final position estimated by the visual odometry algorithm, we can compute the absolute return position error as

$$E = \sqrt{x_{vo}^2 + y_{vo}^2} \tag{5}$$

The same error can be also expressed as a percentage of the total travel distance D as

$$E_{\%} = \frac{E}{D} \times 100 \tag{6}$$

In our experiment, a percentage error of 15.2% was obtained. This relatively high value is partly due to error in camera calibration. In outdoor environment and rough terrain motion, however, a significant amount of error also derives from sudden variations in lighting conditions, vibrations, and quick movements due to rocks and bumps which can cause the feature tracking process to fail. In order to take into account these issues, the computed motion was accepted and a binary flag was raised, if the number of matches found was greater than 10 and the least squares error in motion estimation was less than 3cm. Otherwise, it was discarded and the relative motion was assumed to be equal to the identity matrix. The graph of the binary flag for the described experiment is reported in Figure 10, showing that the method failed in 16.4% of cases.

4. CONCLUSIONS

In this paper, a visual odometry algorithm for 6-DoF ego-motion estimation of a rough terrain mobile robot was presented. The method integrates image intensity and 3D stereo information in the Iterative Closest Point scheme. It was tested on an all-terrain rover in both an indoor and an outdoor

Figure 7: The rover driving in the field on agricultural terrain

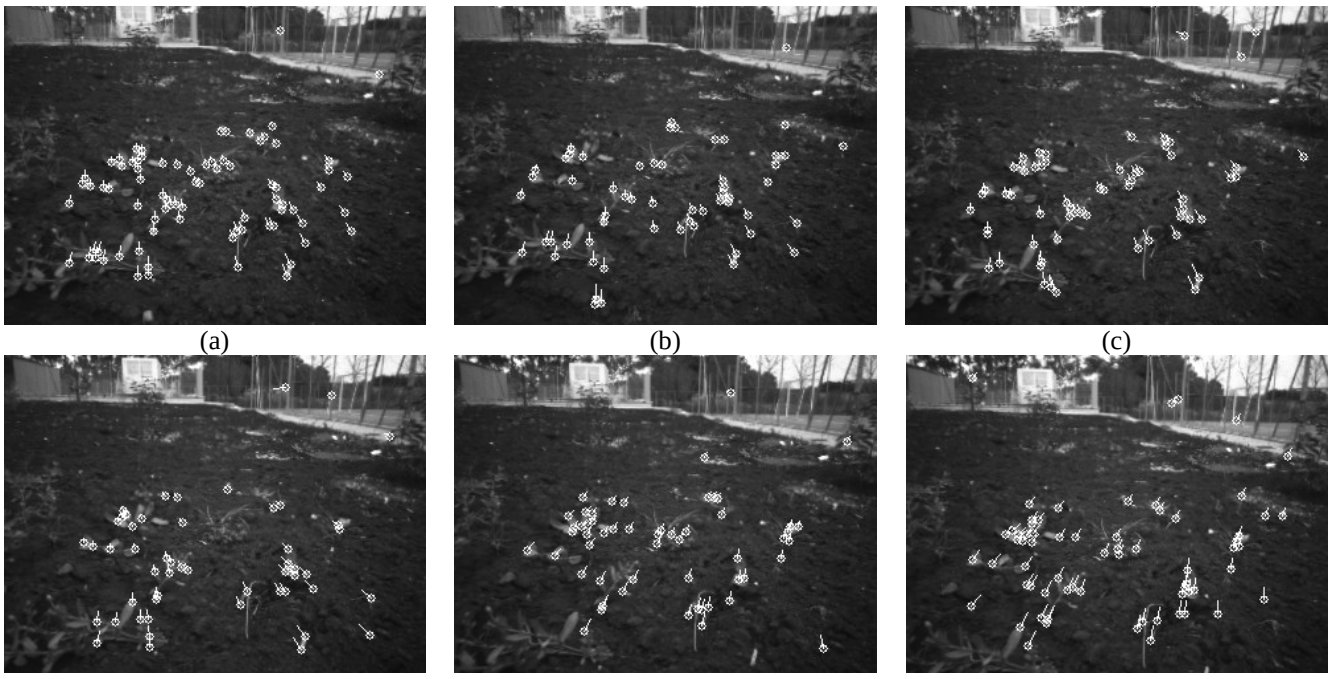

Figure 8: Image sequence captured during the outdoor experiment: forward motion with slight (a)-(c) left and (d)-(f) right turning. White markers denote the tracked features

environment showing the good performance in terms of accuracy and robustness to lighting conditions and external disturbances.

REFERENCES

[1] Borenstein, J., Everett, B., and Feng, L., 1996. Navigating Mobile Robots: Systems and Techniques, A. K. Peters, Ltd., Wellesley, MA, ISBN 1-56881-058-X.

[2] Ojeda, L., Reina, G., and Borenstein, J., 2004. Experimental Results from FLEXnav: An Expert Rule-based Dead-reckoning System for Mars Rovers, *Proc. IEEE Aerospace Conference*, Big Sky, MT, USA.

[3] Matthies, L.H., 1989. Dynamic Stereo Vision, PhD thesis, Carnegie Mellon University.

[4] Olson, C.F., Matthies, L.H., Schoppers, M., and Maimone, M.W., 2003. Rover Navigation Using Stereo Ego-Motion, *Robotics and Autonomous Systems*, 43, pp. 215-229.

[5] Roumeliotis, S. I., Johnson, A.E., and Montgomery, J.F., 2002. Augmenting Inertial Navigation with Image-Based Motion Estimation, *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, pp. 4326-4333.

[6] Corke, P.I., Strelow, D., and Singh, S., 2004. Omnidirectional Visual Odometry for a Planetary Rover, *Proc. IROS 2004*, Japan.

[7] Nistér, D., Naroditsky, O., and Bergen, J., 2004. Visual Odometry, *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition.*

[8] Dunbabin, M., Usher, K., and Corke P., 2005. Visual motion estimation for an autonomous underwater reef monitoring robot, *Proc. Field and Service Robotics Conf.*, Port Douglas, Qld., pp. 57-68.

[9] Mallet, A., Lacroix, S., and Gallo L., 2000. Position Estimation in Outdoor Environments using Pixel Tracking and Stereovision, *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, USA, pp. 3519-3524.

[10] Nistér, D., 2003. Preemptive RANSAC for Live Structure and Motion Estimation, *Proc. IEEE Int. Conf. on Computer Vision*, Nice, pp. 199-206.

[11] Besl, P.J. and McKay, N.D., 1992. A Method for Registration of 3-D Shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239-256.

[12] Surmann, H., Nüchter, A., and Hertzberg, J., 2003. An Autonomous Mobile Robot with a 3D Laser Range Finder for 3D Exploration and Digitalization of Indoor Environments, *Journal Robotics and Autonomous Systems*, Vol. 45, pp. 181-198.

[13] Garcia, M.A. and Solanas, A., 2004. 3D Simultaneous Localization and Modeling from Stereo Vision, *Proc. IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, pp. 847-853.

[14] Sáez, J.M. and Escolano, F., 2004. A global 3D map-building approach using stereo vision, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1197-1202.

[15] Weik, S., 1997. Registration of 3-D Partial Surface Models using Luminance and Depth Information, *Proc. of the Inter. Conf. on Recent Advances in 3-D Digital Imaging and Modeling*.

[16] Bickler, D., 1998. Roving over Mars, *Mechanical*

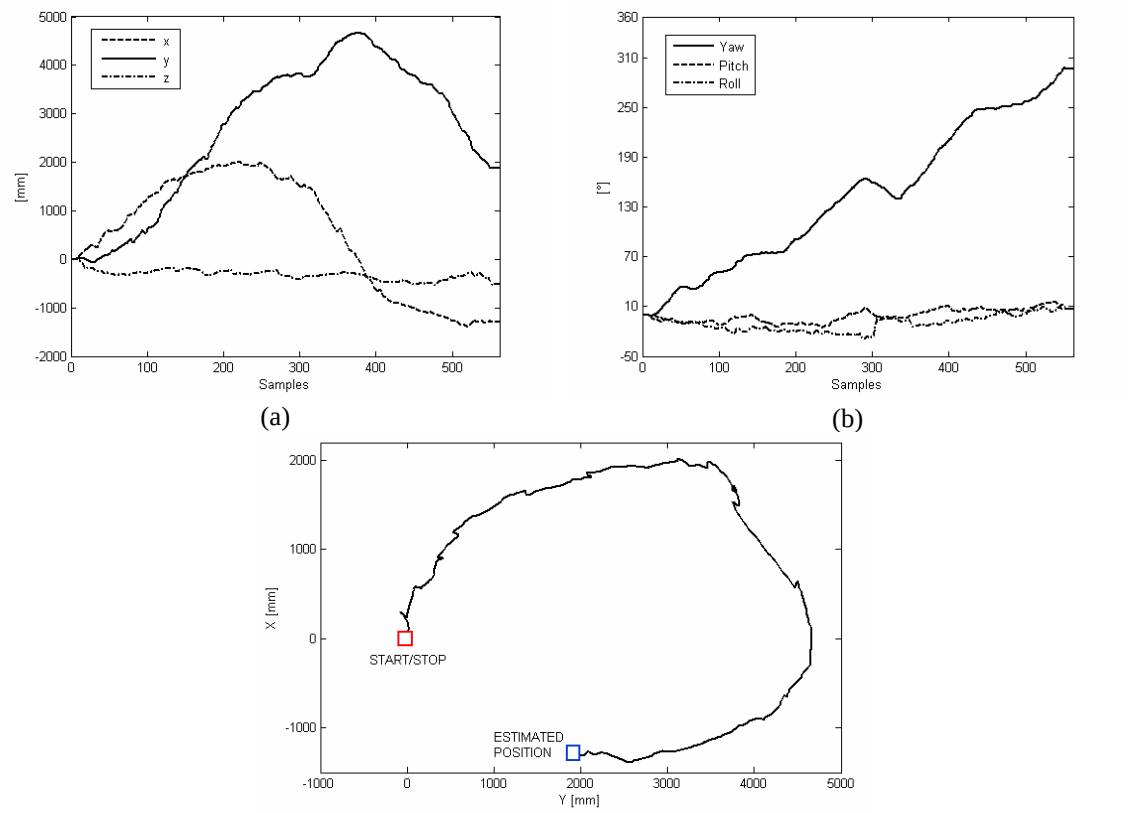

Figure 9: Results of a test on uneven terrain. Turning path: (a) variations of the robot position; b) variations of Euler angles; c) path in the x-y plane

Engineering Magazine, American Society of Mechanical Engineers.

[17] Reina, G., Messina, A., Gentile, A., and Foglia, M., 2007. Dune: a Mobile Robot for Rough-Terrain Applications, *Proc. National Congress of Theoretical and Applied Mechanics (AIMETA)*, Brescia, Italy.

[18] Nesnas, I.A.D., Bajaracharya, M., Madison, R., Bandari, E., Kunz, C., Deans, M., and Bualat, M., 2004. Visual Target Tracking for Rover-based Planetary Exploration, *Proc. IEEE Aerospace Conference*, Big Sky, Montana.

[19] Milella, A., Reina, G., and Siegwart, R., 2006. Computer Vision Methods for Improved Mobile Robot State Estimation in Challenging Terrains, *Journal of Multimedia*, Vol.1, No.7.

[20] Konolige, K., 1997. Small Vision Systems: Hardware and Implementation, *Proc. Int. Symposium on Robotics Research*, Japan.

[21] Shi, J. and Tomasi, C., 1994. Good Features to Track, *Proc. IEEE Conf. of Computer Vision and Pattern Recognition*, CA, pp. 593-600.

[22] Gonzalez, R., and Woods, R., 2002. Digital image processing, Prentice Hall, 2nd Edition.

[23] Walker, M.W., Shao, L., and Volz, R.A., 1991. Estimating 3-D Location Parameters using Dual Number Quaternions, *CVGIP: Image Understanding*, 54, pp. 358-367.

[24] Zhang, Z., 1992. Iterative Point Matching for Registration of Free-Form Curves, *IRA Rapports de Recherche N° 1658 Programme 4 Robotique, Image et Vision*.

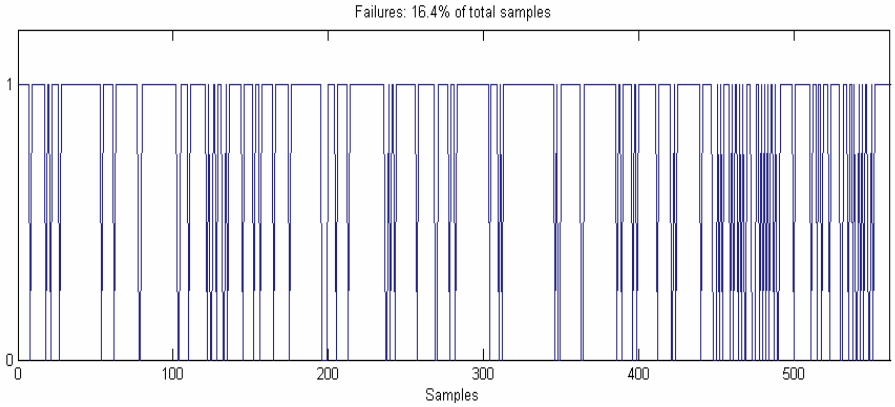

Figure 10: Binary flag representing the failure (*flag*=0) or success (*flag*=1) of the visual odometry algorithm during the test in outdoor environment