Vision-Based Motion Estimation for the ExoMars Rover

F. Souvannavong*, C. Lemaréchal*, L. Rastel** and M. Maurette**

*Magellium, France
e-mail: fabrice.souvannavong@magellium.fr

**CNES (The French Space Agency), France
e-mail: laurent.rastel@cnes.fr

Abstract

Localization is one of the most critical function for the ExoMars rover autonomy. Both the daily distance crossed and the number of iterations needed to reach a site of interest depend on its precision. The Visual Motion Estimation system presented in this paper is designed to satisfy the ExoMars project needs on the localization function. Its main features are : the respect of the very constrained resources of the flight computer, an accurate short term (20cm) and long term (100m) localization given at 0.1Hz against more than 1Hz for terrestrial systems, efficient algorithms to fit time and memory requirements and a modular design to balance computation time and localization accuracy. The algorithm is tested on more than 300 real images acquired in a Mars-like environment every 200mm. All images are precisely localized in 6D by a laser tracker. We therefore give the real performances of our VME function that can later be used to specify a visual localization function for an autonomous navigation system. Our function demonstrates a localization accuracy of less than 2m after a traverse of 100m, when accelerometers are used to correct rover pitch and roll. Adding two extra azimuth correction steps allows to get an error inferior to 1m.

1 Introduction

CNES acts as a technical support to ESA in the ExoMars project, and has already made available to the project the results of its past developments (Mars96, MSR missions, R&D on rovers). This includes stereo bench design, perception and path planning algorithms, and tests facilities. We have been investigating visual motion estimation issues for 7 years as it was identified as a key component of autonomous navigation systems.

ExoMars context ExoMars will deploy a rover carrying a comprehensive suite of analytical instruments dedicated to exobiology and geology research. The rover will have to demonstrate high mobility to allow the efficient exploration of sites of scientific interest, and this with limited human intervention [1]. This implies the on-board and autonomous execution of two tasks : the path planning and the trajectory servo-control. The correct execution of these tasks requires an accurate localization function for two reasons. Firstly, it is critical during trajectory servo-control to insure the safety of the rover. The localization system provides to the locomotion system the current localization of the rover in its local navigation map which contains obstacles or important slopes to avoid. Secondly, it is important to plan efficient and optimal trajectories. On one hand, the computation of local navigation maps takes into account localization errors to enlarge detected obstacles. High errors will then result in bigger virtual obstacles and reduced navigable areas. On the other hand, the localization function gives the rover position with respect to its goal and allows to update the right path to reach it. The ExoMars rover requirements stipulate that the rover should execute journeys up to 100 meter length with a relative localization accuracy better than 5% of the path length and with a design goal of 1%.

The localization function has to give 6D coordinates of the rover at high frequency (about 5Hz) to the locomotion system and at lower frequency (about 0.01Hz) to the path planing system. Usually, an IMU is used to satisfy high frequency requirements for on-board self localization. Unfortunately, integrating accelerometers signal did not provide accurate results in our past experiments to localize a rover. The integration of an acceleration signal along X and Y axis of a rover moving at 1 cm/s is very challenging as shocks of its wheels on the soil surface along Z axis create accelerations hundred times bigger than accelerations of the motion. Other on-board solutions are wheel odometry and Vision Motion Estimation (VME) techniques. We present in this paper a VME algorithm to be combined with a wheel odometry algorithm to satisfy frequency and accuracy constraints. It is designed to measure the rover displacement from distant images of an unstructured rough ground. The algorithm is composed of two loops to optimize the execution time and the localization accuracy.

Related work As far as we know, the idea of estimating camera motion from a sequence of images appeared in early ’80s [2]. It has been a very active field since camera quality and computation power have greatly improved. This technology is, in particular, very attractive to solve the problem of self localization of rovers targeting autonomous planetary exploration [3, 4, 5]. Indeed, there are few external localization means, IMU-based localization is not adapted, and wheel odometry suffers from the difficulty to model and measure the wheel/soil contact.

We can identify two main issues in designing a visual motion estimation system : the matching or tracking of features in images, and the optimization process that estimates the motion and optionally the scene structure.

Concerning the matching or tracking, most solutions found in the literature assume small displacements in pixels between acquisitions, or relatively good a priori estimates. This facilitates image feature tracking or matching in 2D or 3D, but implies high frequency processing [6] or a poor stereoscopic configuration. In this way, our work differs by focusing on large displacements. The proposed method is related to the work presented in [7]. Yet, it mainly differs on two points. Firstly, they compute an initial set of matching landmarks using a visual similarity measure, whereas we use only geometrical constraints considering that good relative attitude measurements are provided by gyros. Secondly, we introduce a different matching score that is more precise to select matching landmarks.

Concerning the optimization, three types of techniques are identified in the literature : (VO) the optimization of solely two acquisitions (Visual Odometry field), (SLAM) incremental approaches using either Kalman filtering or particle filters (Real-Time Localization field), and (BA) global approaches using bundle adjustment (Structure From Motion field). In [8], they present three variations of the SLAM formulation for planetary exploration rovers. They assume that images are taken every 20mm which represents in the end 10 times more images than in the approach discussed here. We propose an hybrid VO/BA on-board approach to satisfy speed and accuracy requirements. VO is executed after each acquisition (every 200mm), while a local BA is executed after each path planing (2.5m). A similar approach, yet more complete, is presented in [9]. Nevertheless, we focus here our attention on on-board autonomous solutions. We include accelerometers to constraint the BA process. And we propose to track features using homographic transformations. Local bundle adjustment is now in great boom as it allies accuracy and real-time constraints [10, 11, 12].

The challenge is now to improve long range localization with loop closing and SLAM methods [13, 14, 15] or data fusion (lander, orbiter, multiple rovers) [9]. These systems have the ability to recognize visited places by potentially different sensors and consequently to adjust model parameters to get a very accurate localization. We do not present such a system that requires more memory and computation time, or on-earth computation assistance. In this article, we focus on the localization accuracy at 100m, which is the nominal distance that will be covered by the ExoMars rover to reach its daily goal.

Proposed method and its evaluation We propose in this paper a VME algorithm to be combined with a wheel odometry algorithm to satisfy frequency and accuracy constraints. It is designed to measure the rover displacement from distant images of an unstructured rough ground. The algorithm is composed of two loops to optimize the execution time and both short and long range localization accuracies : VO-SYS (composed of subsystems VO3D and VO2D) and LSBA. In the first loop, VO3D is responsible for finding a good set of matching features between two acquisitions. It was initially presented in [16]. It uses the output of wheel odometry and gyros to speed up the matching process and increase its robustness. This sub-system is next completed to improve the rover localization by two ways. Firstly, point features are tracked in images and there re-projection error is minimized with respect to the camera motion in the subsystem VO2D. Secondly, a bundle adjustment is applied on tracked points between two path planing operations in the second loop (LSBA).

Presented methods and their variations are compared and evaluated on simulated and real images. The evaluation first concerns computation time and memory needs, next localization accuracy. Computation time and memory needs are indicated for the first loop, i.e. VO3D and VO2D, that are the most critical ones. The bundle adjustment process is not evaluated here as it is not yet fully optimized. Nevertheless, this function benefits from relaxed constraints as it is ran when the rover is stopped. We do not identify any major difficulty to have this optimization executed on-board in a reasonable amount of time. The random process behind incremental localization is very complicated and requires a huge amount of data to be characterized. To tackle this issue, a validation method which includes: a limited set of images fully localized, the simulation of gyros data, the pseudo-random generation of trajectories from real data, and the use of our rover simulator for Monte Carlo like tests, is introduced.

The paper is organized as follows. Section 2 gives a summary of requirements on the localization function and a top down description of the localization system we designed. Section 3 presents the evaluation framework. Section 4 reports on evaluation results. Finally, section 5 concludes with recommendations for the implementation of the vision-based localization function of the ExoMars mission and suggests future research directions.

2 Localization System Requirements and Design

The localization system is designed to meet tight constraints on computation time, memory needs and accuracy specified by the ExoMars project. We first present the autonomous navigation process to better understand localization issues. And we provide a summary of the ExoMars mission requirements on the localization function. Next, we carry on with an overview of the localization system to end with a detailed description of the VME function.

2.1 Context and Requirements

The localization system is a key component of the autonomous navigation process described in figure 1. The rover is equipped with a stereo bench at the top of its mast for navigation purpose (NavCams). It is designed to perceive the 3D environment around the rover up to a distance of 6m. A navigation map is build from these 3D data. It identifies navigable areas and obstacles according to the rover crossing capabilities. The system is then capable of finding a safe path of 2.5m length in the perceived environment. This step is called path planing and is executed at each P_i position where P_o is the initial rover position of the day. It is repeated every 2.5m due to the vanishing reliability of the stereo vision process beyond 3m from the rover (in the ExoMars context). The rover localization is required at each P_i position to compute the optimal path toward the goal and to finally assert the goal is reached. Between two positions P_i and P_{i+1} the rover moves blinded and has to rely on its localization system and navigation map, i.e. map of navigable areas defined relatively to the P_i position and computed at the last path planing step P_i. The rover location has to be accurate to safely reach the next P_{i+1} position and provided in real-time to insure the stability of the trajectory servo-control. For this purpose, a second stereo bench (LocCams) is taken on board and dedicated to the visual localization function.

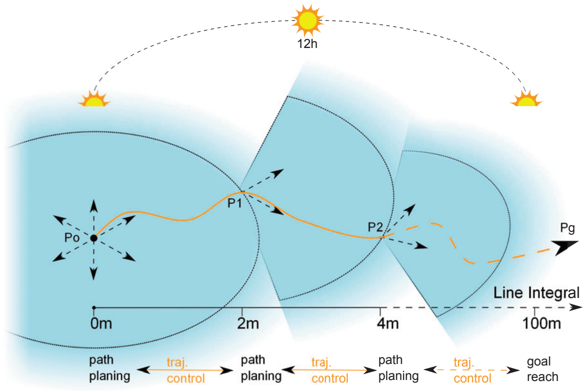


Figure 1. : Autonomous Navigation

Given the autonomous navigation iterative process and the rover hardware configuration, the following requirements were identified :

Trajectory servo-control : real-time accurate localization of the rover at 5Hz. A relative localization accuracy of 5% is specified and a design goal of 1% is expected.

Goal reach : the daily goal P_g specified by the ground segment has to be reached with a relative accuracy of 5% and a design goal of 1%. It is nominally located at 100m from P_o.

Computation time : as we will see later, visual motion estimation has to be executed every 10 seconds.

2.2 Localization System Overview

Wheel odometry suffers from the difficulty to model and measure the wheel/soil contact and it results in important localization errors. We learned from the NASA MER mission that wheel odometry estimates rover positions and orientations with a nominal accuracy of 10%. An alternative solution is to use visual motion estimation (VME). Unfortunately, it does not run in real time on the targeted platform. It thus comes as a complementary module to wheel odometry. This latter provides real-time localization and is adjusted every 200mm by visual odometry. 200mm is the distance between two rover axles. The VME function we present, takes as input two stereoscopic images taken at time instants $(t - 10s)$ and (t), the corresponding attitude variations provided by gyros integration and the translation estimate from wheel odometry. These inputs are processed at 0.1Hz (every 200mm for a rover speed of 20mm/s). Accelerometers are applied at each position P_i when the rover is stopped to correct estimated pitch and roll. In addition, a sun sensor can be used to correct the estimated rover yaw. An overview of the localization system is given in figure 2.

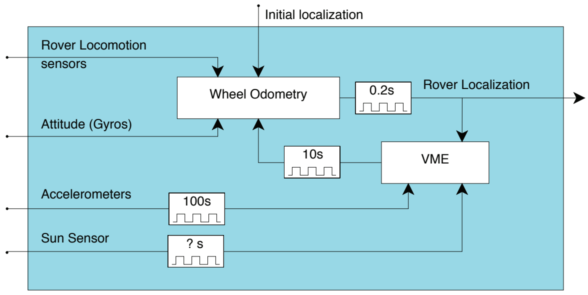


Figure 2. : Localization System Overview

2.3 Visual Motion Estimation

We designed two basic functions. The first one, called VO-SYS, provides the localization of the rover after each acquisition. Its accuracy is limited in time. The second one, called VME-SYS, integrates a bundle adjustment to improve the long range accuracy.

2.3.1 Visual Odometry

The visual odometry system called VO-SYS in figure 3 is composed of two sub-systems : the first one VO3D is designed to find a subset of good matches and to provide a first motion estimate; the second VO2D provides a refined motion estimate based on many tracked features. The overall allows to provide an accurate motion estimate at the frequency required by the mission.

Perception Input images are first degraded by 4 to a resolution of 320x256 pixels. This step is mainly done to reduce the execution time of the dense stereo process and of the Harris corner detector. Figures are given in section 4 Degraded images are next rectified thanks to an fine correction bitmap that implicitly models geometrical distortions and camera positions to fit a perfect model of pinhole

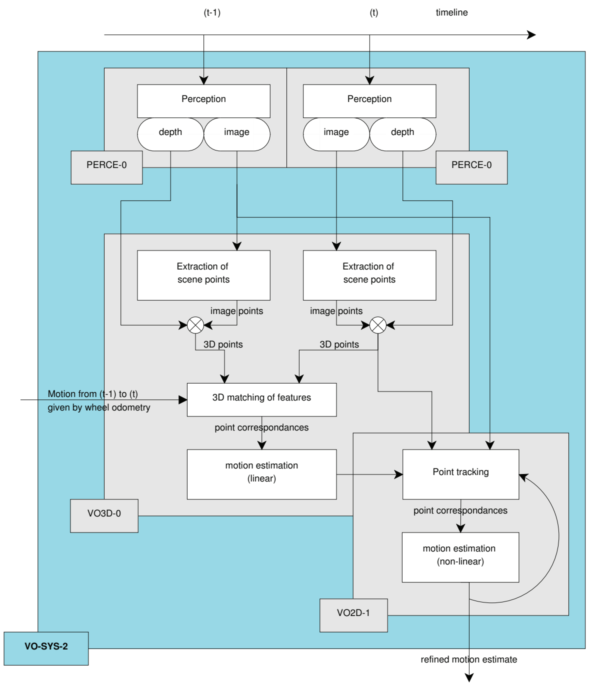


Figure 3. : Visual Odometry

cameras mounted on a rectilinear stereo rig. An optimized stereo process is next applied on gradient images to get the depth of each pixel of the left view.

Point matching The matching aims at identifying similar landmarks a priori detected in left images at $(t-1)$ and (t). This step is crucial for motion estimation. Landmarks are detected with the Harris corner detector, that was modified to select 300 points well distributed in images. It is crucial in presence of shadows or important variations of textures. Next, points that are geometrically compatible, knowing the rotational motion R provided by an external sensor (gyros), are matched in 3D. The algorithm relies on the observation that lengths and angles are preserved in the euclidean space. It then requires a sufficiently accurate stereoscopic triangulation. But, this approach has the main advantage of being invariant to image variations. A detailed description of the matching process is given in [16]. A first motion estimate is finally computed from these initial matches.

Point tracking The Harris corner detector is a good operator to detect features to be matched in two images. However, corner localization is not sufficiently accurate for motion estimation. This issue is then solved by tracking features. As we have a first good motion estimate, the tracking is efficiently done by block-matching and a ZNCC operator. The matching is either executed on degraded images or on raw images at full scale. In this latter case, a local rectification is applied at the proper resolution. This approach is identified by the prefix HR in coming figures. We have also investigated patch deformation by a local homography to improve the tracking accuracy and robustness.

Accurate motion estimate Finally, the motion is estimated by minimizing the projection error of tracked points in the past left acquisition. Given o_i the 2D observation of the feature i at $(t-1)$, M_i the 3D feature at (t), and P the projection matrix. The motion estimate is the solution of the following equation :

$$[R,T] = \arg\min_{[R,T]} \left(\sum_i \|o_i - P.(R.M_i + T)\|^2 \right) \tag{1}$$

The Levenberg-Marquardt optimization technique is applied to find R and T. About 25 iterations are performed to reach a stable solution. In addition, we have further constrained the optimization process by including observations on n past images. The objective is to limit integration errors while keeping the optimization process simple. The previous equation is thus modified as follows without changing the number of parameters to optimize :

$$[R,T] = \arg\min_{[R,T]_{t\to(t-1)}} \tag{2}$$

$$\left(\sum_{j=t-1}^{t-n} \sum_i \|o_{ij} - P.([R,T]_{(t-1)\to j}[R,T]_{t\to(t-1)}M_i)\|^2 \right)$$

In our experiments, n is set to 4. It corresponds to the mean number of images where a point can be identified.

2.3.2 Local Bundle Adjustment

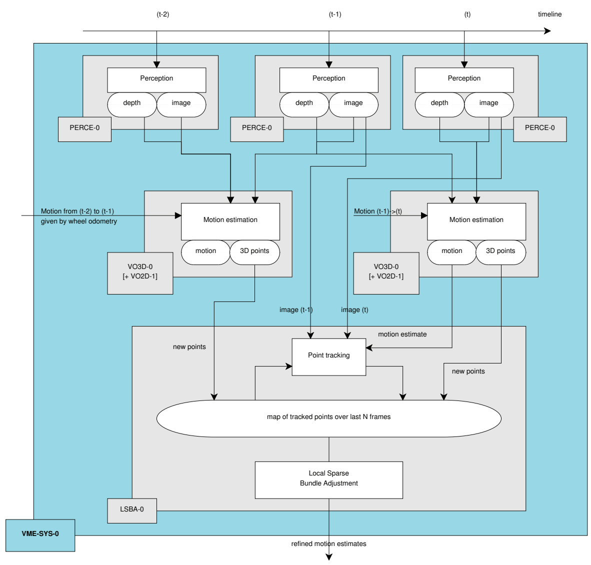


Figure 4. : Bundle Adjustment

The visual odometry system is sufficient to give the short-term localization of the rover but the error grows rapidly after few meters. A first solution is to use accelerometers to correct the rover pitch and roll each time it stops. A second solution is to optimize by bundle adjustment the parameters of the overall image sequence, i.e. rover and features positions. But it would require too much computation time and power to be run on-board over data of a complete trajectory. Nevertheless, a local bundle adjustment is acceptable. Therefore, the optimization technique is applied on data gathered between two path planing steps, i.e. every 2.5m. These key positions have also the advantage of giving access to accelerometers to measure rover pitch and roll as it is stopped. These constraints can be included in the optimization process as explained in [11]. We use the library "A Generic Sparse Bundle Adjustment C/C++ Package" [17] to evaluate this approach. Unfortunately, pitch and roll constraints can not be easily included with this library and we removed "known" angles from the optimization for now.

3 Evaluation Framework

Visual Motion Estimation (VME) is a complex system difficult to design and evaluate. On one hand, there are manifold parameters that can be tuned : field of view, stereo bench baseline and position, algorithms possibilities. On the other hand, the variability of possible observations, i.e. images, is huge. The influence of parameters can not always be tested on the terrain for practical and cost reasons. We thus use the terrain and rover simulator to study the design of the VME system. It gives an overview of VME performances and allows to compare methods. Simulations are next completed by experiments with the IARES rover. These experiments are primordial to assess for system performances. This section first present the simulator, next the evaluation process that was setup to evaluate VME performances.

3.1 Simulations

The validation and study of algorithms and cameras arrangements is conducted on simulated images given by our rover simulator (RS). RS is a real-time functional rover simulator which includes 3D modeling and kinematic simulation of IARES and ExoMars chassis, 3D modeling of the terrain and rocks and shadowing. The rendering is done by the Ogre engine[1].

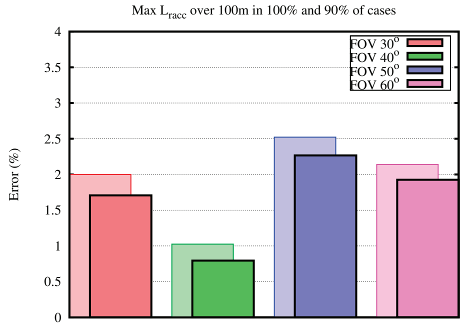
For now, a straight path is executed by the simulated rover. Besides, we have fixed the position of the stereo bench to the expected position on the ExoMars rover. The stereo bench baseline is also set to 80mm. Optimal field of view and pitch values of the stereo bench are then studied. Simulations, see figure 5, reveal that a field of view of 40^o and a pitch of -45^o allow to get best performances. This is of course valid for the presented algorithm. In addition, performances degrade slowly around these values as soon as accelerometers are in the loop. We will see that compared to performances on real images a factor of five is observed.

3.2 Real Images Exploitation

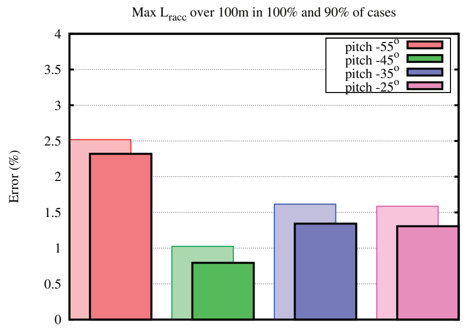
3.2.1 Test Data

A test campaign was conducted on the SEROM site under the following conditions :

[1] http://www.ogre3d.org



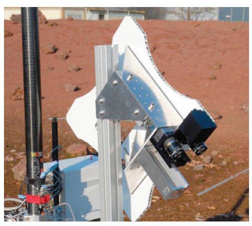
(a) Performance evolution wrt. cameras field of view for a pitch of -45^o.



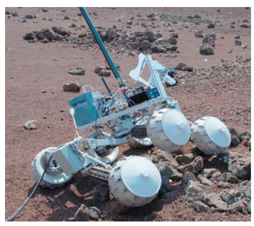
(b) Performance evolution wrt. cameras pitch for a field of view of 40^o.

Figure 5. : Performances on simulated images.

- stereo bench position and orientation compatible with the configuration of the ExoMars rover at that time (elevation of ˜900mm, pitch of 45^o, focal length of 8mm),
- precisely localized acquisitions : 0.05mm in position and 0.02^o in attitude over each direction thanks to the SMR plane calibrated and localized by a laser tracker,
- continuous trajectory of 70m executed by the IARES rover,
- ˜200mm between acquisitions (330 acquisitions), rover stopped,
- various types of ground, (panel of rock densities present on the CNES Mars yard)



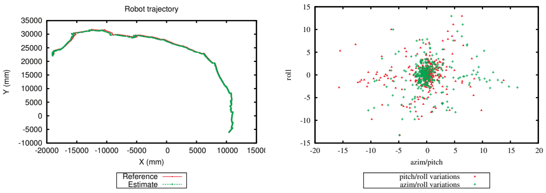
(a) The stereo bench and its SMR plane mounted on IARES



(b) IARES going through a rugged area

Figure 6. : Pictures of the acquisition campaign. The position and attitude of cameras are measured by the laser tracker.

Figure 7 shows the trajectory as it was executed by the rover. This is the reference trajectory that is used to construct our simulated trajectories as we will see. The figure also gives an overview of the large panel of attitude variations that were measured between acquisitions.



(a) Trajectory executed by the rover.

(b) Summary of attitude variations of the rover in ^{o}.

Figure 7. : Trajectory properties

3.2.2 Simulated Trajectories

Path Until now, our reference trajectory is 70 meter long which is not sufficient to evaluate the VME function. In addition, a single trajectory is not representative as localization performances are tightly linked to the integration of attitude variations errors and dead reckoning. A bigger acquisition campaign is scheduled for august 2010. In the meantime, we generate new artificial trajectories derived from the original one, in order to combine estimation errors in different ways. This is simply achieved by segmenting the initial trajectory into N sections (5m long each) that are afterward recombined randomly to form new artificial trajectories of 100m length. Furthermore, a new realization of the stochastic process describing gyro and calibration errors is picked for each generated trajectory. More details about the noise model are given later. New trajectories are optionally corrected in yaw to favor a rectilinear path. This kind of trajectory was identified as a worst case for evaluations based on the relative error. The evaluation ends up with a Monte-Carlo simulation that allows to qualify the localization error.

Noise model The attitude data is subject to calibration errors coming from the measurement noise and the uncertainty on the gravity vector orientation. Measurement errors are modeled independently by a zero mean Gaussian distribution. They are added to laser tracker measurements to model a real sensor. Note that accelerometers are used at each P_i position when the rover is stopped. This calibration is necessary to monitor the stable position of the rover. By contrast, the sun sensor can be used less often as the heading measurement is not critical. Nevertheless, it is preferable to use the sun sensor several times a day to ensure a good long range localization of the rover. In our experiments, a variance of 0.1^o is used for all noise sources.

4 Results

Memory The total amount of memory needed by VO-SYS' data is around 3MB. If the tracking is done at full image scale, then an extra of 2.5MB is needed. It can be reduced to 640KB if the tracking is done at half image scale. Last, the LSBA process requires to save all tracked points between two path planing steps, which should not exceed some dozen of KB.

Time Table 1 presents the actual time consumption of algorithms. One iteration of VO-SYS takes 260ms from image pre-processing to motion estimation on our Intel core 2 Duo E6550 @ 2.33GHz . Extra work is required to get more accurate figures targeting the ExoMars mission platform, the best way being to run algorithms directly on a Leon-like architecture. However, a first extrapolation allows us to say that one VO-SYS iteration should run in less than 10s.

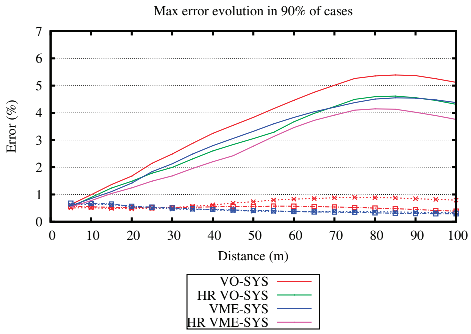
Table 1. : Execution time of V0-SYS-2 modules. Experiments were executed on a CPU Intel core 2 Duo E6550 @ 2.33GHz with 4096 KB of cache memory.

Sub-system	Module	Time on Linux (ms)
Perception For NavCams	Down sampling by 2	15
	Rectification	56
	Stereo (SGRAD LINE)	97
	Filtering	7
	Total	175
PERCE-0	Down sampling by 4	22
	Rectification	20
	Stereo (SGRAD LINE)	75
	Filtering	3
VO3D-0	Point detection	50
	3D matching, motion estimation	80
V02D-1	Tracking, motion estimation	10
	Total for the system V0-SYS-2	260

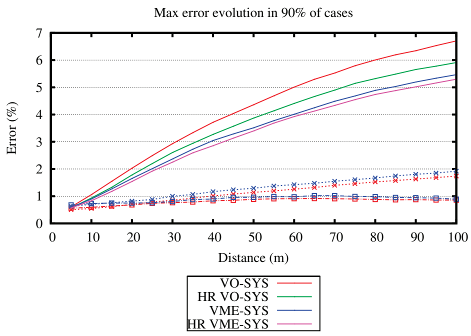
Accuracy Figure 8 shows the localization performances that we get with VO-SYS and VME-SYS on real images and with our evaluation framework. About 700 trajectories of 100m were generated. We can say that :

- First, it confirms that the original trajectory, as it is, is not adapted to provide a good evaluation of localization performances. A difference of 1% is observed when the evaluation is ran on the repeated trajectory or on the randomized and yaw corrected trajectory,
- As expected, the error grows rapidly if no calibration with an absolute reference is done. In that case, the error process is similar to a random walk process that is known to have a variance $\propto \sqrt{t}$. Of course, this is not observed on the top figure (repeated original trajectory),
- Correcting the attitude estimate with accelerometers allows to greatly improve localization performances as it partially breaks the random walk process. A localization accuracy of 2% is obtained on our worst case scenario and less than 1% on the repeated original trajectory,
- Correcting the attitude estimate twice with a sun sensor gives finally a localization accuracy of less than 1%,

- The bundle adjustment does not provide best performances on random trajectories. This was expected as it should perform better on continuous sequences of images.



(a) Original trajectory repeated



(b) Original trajectory randomized and corrected in yaw

Figure 8. : Localization performances on real images with our evaluation framework. Calibration noises have a standard deviation of 0.1^o. **No points** : blinded VME, i.e. without attitude calibration. **Crosses** : the attitude given by VME is corrected by accelerometers every 2.5m. **Squares** : the attitude given by VME is corrected by the accelerometers every 2.5m and by the sun sensor twice.

Figure 9 shows the attitude error evolution over time. As opposed to stellar sensors that observe points at infinity, roll and yaw are the best estimated angles while the pitch is less accurate. In fact, it depends on the size of the image overlap, i.e. the motion amplitute between two acquisitions. Attitude errors are the main source of long range drift and then can be reduced by looking more upward at the expense of the short range accuracy. However, simulations show that the localization error is not improved with our algorithm. In addition, VME cameras might be used to detect obstacles. In that case, they should point at the close surrounding of the rover.

5 Conclusions and Perspectives

The localization system is a critical component of autonomous rovers targeting planetary exploration. The lack of external means to localize them implies the on-board implementation of a self localization function. Visual motion estimation is then a key module to obtain an accurate

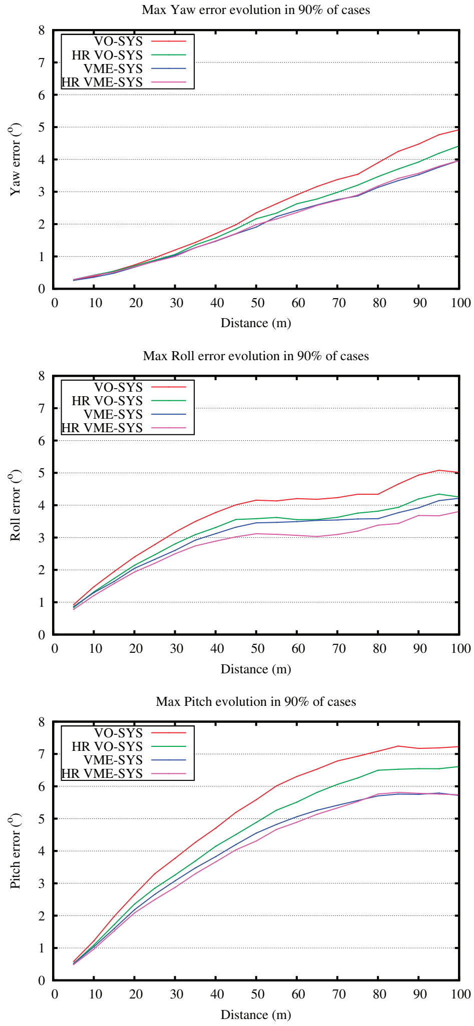


Figure 9. : Attitude error evolution on real images with our evaluation framework. Accelerometers are not used here.

localization.

We have investigated ways to design an efficient visual motion estimation system that answers to the ExoMars mission needs. We have presented in this paper the results of our studies. It is now clear what can be the architecture of a vision-based localization function for ExoMars. In particular, a double loop estimation of the motion is necessary to get both short range and long range accuracy. The same camera sensor can not provide both short range motion length accuracy and long range attitude accuracy. The first estimation loop, VO-SYS in this paper, should then be designed to guarantee a precise motion length estimation. Simulations suggest that a field of view around 40^o and a pitch around -45^o corresponds to a good stereo bench configuration for this loop. Next estimation loops are added to improve the long range attitude estimation. It can simply consist of using accelerometers, preferably in a bundle adjustment. And it is still better if a sun sensor is used regularly.

The selected system design meets the ExoMars project needs as it was demonstrated with our prototype and evaluation framework. VO-SYS, corrected in pitch and roll, allows to get a localization accuracy of 2% after a 100m traverse. Adding two extra calibrations with a sun sensor allows to get a relative error under 1%. Moreover, one iteration of the function should take less than 10s to estimate the camera motion.

In future works, we will carry on with the study of our bundle adjustment process. The objective is to identify the best window size on which the optimization should be conducted. Besides, we think that using overlapping windows could improve the overall accuracy and this will be investigated. In parallel, our bundle adjustment process will be generalized to include navigation cameras images. There use can improve attitude estimation accuracy as they look farther away. In addition, a target tracking strategy could further improve the accuracy of the goal reaching function.

References

[1] M. Maurette. Mars rover autonomous navigation. *Autonomous Robots*, 14(2-3):199–208, 2003.

[2] L. Matthies and Steven Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, 3(3):239 – 250, June 1987.

[3] S. Se, T. Barfoot, and P. Jasiobedzki. Visual Motion Estimation and Terrain Modeling for Planetary Rovers. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, volume 603 of *ESA Special Publication*, August 2005.

[4] Cheng YANG, Mark W. MAIMONE, and Larry MATTHIES. Visual odometry on the mars exploration rovers. *Proceedings of the IEEE International Conference on robotics and automation*, 13(2):54–62, 2006.

[5] Clark F. Olson, Larry H. Matthies, Marcel Schoppers, and Mark W. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215 – 229, 2003.

[6] A. Mallet, S. Lacroix, and L. Gallo. Position estimation in outdoor environments using pixel tracking and stereovision. *Proceedings of the IEEE International Conference on Robotics and Automation*, 4:3519–3524 vol.4, 2000.

[7] H. Hirschmuller, P.R. Innocent, and J.M. Garibaldi. Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics. *Proceedings of the International Conference on Control, Automation, Robotics and Vision*, 2:1099–1104 vol.2, Dec. 2002.

[8] J.N. Bakambu, C. Langley, and R. Mukherji. Visual motion estimation: Localization performance evaluation tool for planetary rovers. In *In Proceedings of ISAIRAS Conference*, Feb. 2008.

[9] Ron Li, Kaichang Di, Jue Wang, Sanchit Agarwal, Larry Matthies, Andrew Howard, and Reg Willson. Incremental bundle adjustment techniques using networked overhead and ground imagery for longrange autonomous mars rover localization. In *Proceedings of ISAIRAS 2005 Conference*, 2005.

[10] Z. Zhang and Y. Shan. Incremental motion estimation through modified bundle adjustment. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–343–6 vol.3, Sept. 2003.

[11] N. Sunderhauf, K. Konolige, S. Lacroix, and P. Protzel. *Visual Odometry using Sparse Bundle Adjustment on an Autonomous Outdoor Vehicle*. Tagungsband Autonome Mobile Systeme. Springer Verlag, 2005.

[12] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Monocular vision based slam for mobile robots. *Pattern Recognition, International Conference on*, 3:1027–1031, 2006.

[13] Thomas Lemaire, Cyrille Berger, Il-Kyun Jung, and Simon Lacroix. Vision-based slam: Stereo and monocular approaches. *Int. J. Comput. Vision*, 74(3):343–364, 2007.

[14] G. Silveira, E. Malis, and P. Rives. An efficient direct approach to visual slam. *IEEE Transactions on Robotics*, 2008.

[15] Pantelis Elinas, Robert Sim, and James J. Little. σslam : Stereo vision slam using the rao-blackwellised particle filter and a novel mixture proposal distribution. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2006.

[16] F. Souvannavong, C. Lemarchal, E. Remetean, L. Rastel, and M. Maurette. Visual odometry for autonomous localization on mars. In *10th Workshop on Advanced Space Technologies for Robotics and Automation*, Nov. 2008.

[17] M.I.A. Lourakis and A.A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug. 2004.