3D-Odometry for rough terrain – Towards real 3D navigation

Pierre Lamon and Roland Siegwart

Swiss Federal Institute of Technology, Lausanne (EPFL)
Pierre.Lamon@epfl.ch, Roland.Siegwart@epfl.ch

Abstract

Up to recently autonomous mobile robots were mostly designed to run within an indoor, yet partly structured and flat, environment. In rough terrain many problems arise and position tracking becomes more difficult. The robot has to deal with wheel slippage and large orientation changes. In this paper we will first present the recent developments on the off-road rover Shrimp. Then a new method, called 3D-Odometry, which extends the standard 2D odometry to the 3D space will be developed. Since it accounts for transitions, the 3D-Odometry provides better position estimates. It will certainly help to go towards real 3D navigation for outdoor robots.

1. Introduction and motivation

Keeping track of position is one of the most important tasks for an autonomous mobile robot. For example, tasks like path planning need good position estimation in order to work well. For rough terrain this task is much harder and requires the estimation of six degrees of freedom. A lot of research on ego-motion using different kinds of sensors has been done. A computation of displacements by tracking pixels from one image frame to another and considering the corresponding 3D points sets produced by stereovision can be found in [1] and [2]. An extension of shape-from-motion to omnidirectional cameras is presented in [3]. Because of the camera's wide field of view and lack of degenerate motions, this method is more likely to produce robust motion estimates. The fusion of both inertial and visual cues can improve the motion estimation [4][5].

The methods presented above generally assume small displacements and angular changes between two acquisitions (feature tracking) and therefore limit their application to smooth environments and/or slow speeds. Furthermore, they generally fail when the images are underexposed or saturated because of reflection or direct sun exposure. In order to improve the robustness of the position tracking one has to fuse the data from different types of sensors, e.g. inertial, visual, proximity sensors and wheel encoders.

Although odometry is widely used for indoor (2D), its application is limited for rough terrain (3D). The wheels are more likely to slip because of the rough structure of the soil and the error in position estimation grows quickly. A model of the wheel-soil interaction allows accounting for wheel slippage and limits the error growth. Nevertheless those models are generally complex and won't give good results for every kind of terrain [6]. For these reasons, one generally avoids using odometry for rough terrain. One can look at the problem differently and ask: Why are the wheels slipping and how could we avoid this?

There are two different aspects on which we can act directly. The first one is to improve the mechanical structure of the robot. Indeed, a good mechanical design allows the rover to move smoothly across the obstacles and therefore limits the wheel slippage. The second one is to focus on the wheel controllers. A good balance of the torques and speeds between the wheels is essential for optimizing the robot's motion [7].

The off-road rover we are using for this research is presented in section 2. Section 3 describes a new motion estimation method based on the wheels' encoders called 3D-Odometry. The experimental results are presented in section 4 and future work in section 5.

2. System architecture

The following sections present the latest developments on our research platform. We will first present the newly mounted sensors and then the software tools and architecture.

2.1 Hardware architecture

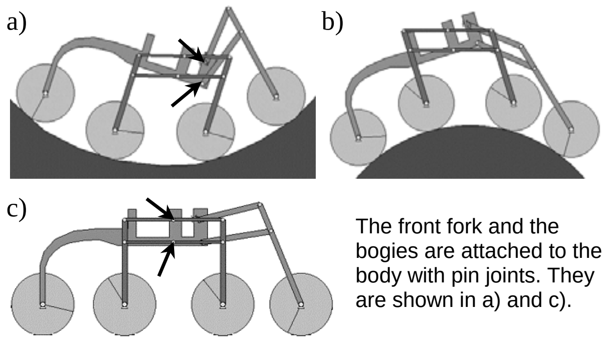

a)
b)
c)
The front fork and the bogies are attached to the body with pin joints. They are shown in a) and c).

Figure 1. Profile views of the Shrimp's mechanical structure

Our lab (Autonomous System Laboratory, ASL) developed an off-road rover called Shrimp which shows good climbing capabilities. Its passive and non-hyperstatic structure can adapt to a large range of obstacles and therefore allows to limit wheel slippage (see Fig. 1). It has six motorized wheels and is composed of four main parts: the body (on which the rear wheel is attached), the articulated front fork and the two side bogies. A detailed description of the mechanical design can be found in [8].

The control system is depicted in Fig. 2 and Fig. 3 shows an overview of the rover and the experimental setup. A lightweight laptop clocked at 600 MHz controls the whole system via three communication channels: two RS232 serial ports (com1 and com2) and one IEEE 1394 bus. One can also exchange information with a host computer through an Ethernet link. The first serial link is used to interface an I2C bus on which different slave modules developed at ASL are attached. The robot has a motor controller module for each wheel, one servo controller for the steering of the front and back wheel and one angle sensor module for sensing the state of the bogies and the front fork relative to the body. The modules dedicated to the motors provide speed and position control (PID).

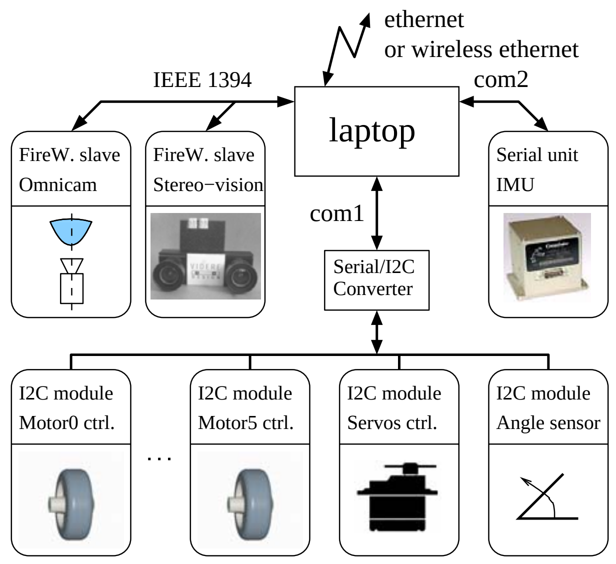


Figure 2. Overview of the Shrimp's actuators and sensors

In order to measure the angular rates and accelerations, we have mounted an IMU[1] on the system. This unit also provides stabilized roll and pitch angles in static and dynamic conditions. In our application it is interfaced via the second serial communication port. Finally, stereo and panoramic images can be acquired through an IEEE1394 bus. For the moment only the stereovision head[2] has been mounted on the robot.

[1] Inertial Measurement Unit, VG400CC-200 from Crossbow
[2] Stereo-vision module of type MEGA-D from Videre Design

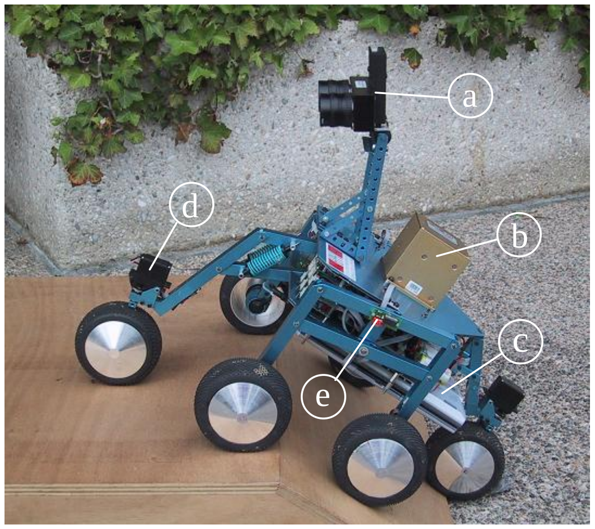


Figure 3. The Shrimp rover climbing a structured obstacle. (a) the stereovision module (b) the IMU (c) the laptop (d) the front wheel servo (e) the left bogie angle sensor

2.2 Software tools and architecture

The whole software has been programmed in C and C++ and runs under Linux. However, substantial effort towards portability has been done by choosing cross-platform components and libraries. The system is divided into three functional modules running as separate processes. We use the IPC messaging system developed at CMU for inter-process communication [9]. Fig 4. depicts the whole software architecture.

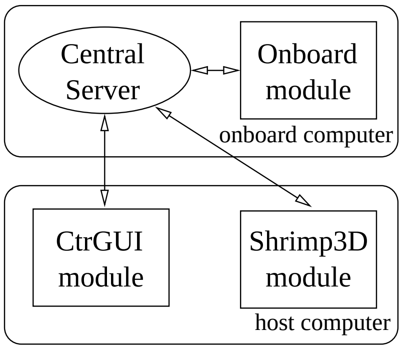


Figure 4. Layout of the IPC based distributed system

The central server is responsible for routing the messages and holds the system-wide information (such as defined message prototypes). The 'Onboard' module is the main program. It runs on the rover's computer and controls its hardware e.g. the actuators, the sensors and the IEEE1394 imagers. For controlling the rover through an ethernet link we have developed a dedicated module called 'CtrGUI'. This graphical user interface allows the user to drive the rover based on the source images provided by the stereovision module. Finally Fig. 5 shows two snapshots of the 'Shrimp3D' module. This module has been developed for visualizing and logging the data coming from the robot. It is based on the Open Inventor C++ library from Silicon Graphics that allows to easily manage the hierarchical transformations between the 3D objects in the scene e.g. the motion of the front fork and bogies relative to the body. All variables stored during the experiment can be plotted and analyzed in the variable browser shown in Fig. 5. This module is a precious tool for testing the system and the algorithms.

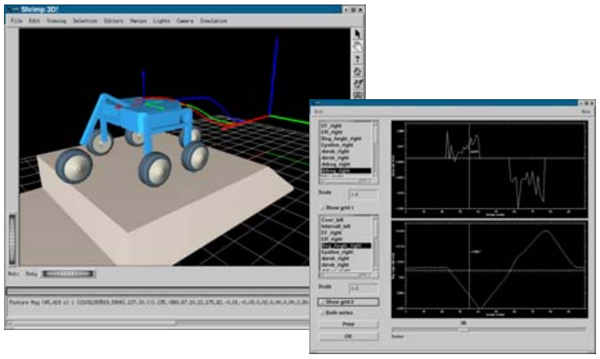


Figure 5. Snapshot of the main Shrimp3D module window (left) and the variables browser (right). This user interface allows the user to visualize the experimental data. It displays the robot's state, the elements of the scene and the computed trajectories. New objects are easily incorporated into the scene. The user can replay the experiment by handling the horizontal slider. All variables stored during the experiment can be plotted and analyzed in the variables browser. The user can choose a sample by means of the slider and the 3D scene is updated accordingly.

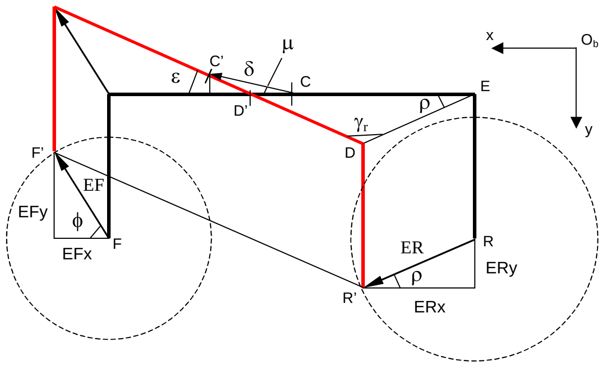
3. 3D-Odometry

The odometry is widely used for mobile robots moving on flat and even terrains. The equations are well known and allow estimating the position and the orientation of the robot (x, y, θ) in the plane. When the robot has to deal with ground slope changes, it is important to track the z coordinate also. The usual way to solve this issue is to add an inclinometer on the robot. This sensor provides the roll and pitch of the robot relative to gravity and allows to know the orientation of the plane on which the robot is currently traveling. Using this information and the standard 2D odometry equations it is possible to compute the 3D coordinates of the robot. This method, wich will be refered later as the standard method, works well under the assumption that the ground is relatively smooth and doesn't have too many slope discontinuities. Indeed, the system accumulates errors during transitions because of the planar assumption. In rough terrain this assumption is not verified by definition and the transitions problem must be addressed properly.

The main idea of the 3D-Odometry is to consider the displacement **and the direction of motion** of the wheels.

3.1 Bogie displacement

For the Shrimp we actually consider the translations of the left and right bogie. The following shows how to compute the displacement (δ and μ) knowing the translation of the wheels (encoder data ER, EF) and the change of the bogie angle (ε) between the initial and final state (see Fig. 6). The solutions provided by this method can be applied for any bogie design.



ER, EF	: rear/front wheel displacement, norms of vectors **RR'** and **FF'**
ρ, ϕ	: rear/front wheel angle, direction of displacement
R, F	: initial rear/front wheel center
R', F'	: final rear/front wheel center
C, C'	: initial/final state of the bogie center
ε	: angle change of the bogie (without pitch angle change)
h	: distance between the wheel centers
δ	: norm of the displacement, norm of vector **CC'**
δx, δy	: x/y components of vector **CC'**
μ	: direction of **CC'**, **C'CD'** angle
γ_r	: **EDD'** angle

Figure 6. Bogie's schematic and variables definition.

Since the distance between the wheels remains constant we can write the following equations.

$$\overrightarrow{RF} = \overrightarrow{RR'} + \overrightarrow{R'F'} + \overrightarrow{F'F}$$

$$\begin{pmatrix} h \\ 0 \end{pmatrix} = \begin{pmatrix} ER\cdot\cos(\rho) \\ ER\cdot\sin(\rho) \end{pmatrix} + \begin{pmatrix} h\cdot\cos(\varepsilon) \\ -h\cdot\sin(\varepsilon) \end{pmatrix} + \begin{pmatrix} -EF\cdot\cos(\phi) \\ EF\cdot\sin(\phi) \end{pmatrix} \quad (1)$$

These equations can be solved analytically for ϕ and ρ (with the displacements ER, EF and the change of the bogie angle ε as parameters). Unfortunately, since three parameters are needed for estimating two values there is a dependency between them. For example, if ε is zero then ER must be equal to EF because of the constant wheels' distance constraint. In practice, ER and EF can be different because they are provided by encoders and the wheels can slip and have different speeds. When such a case occurs we simply consider that the total bogie displacement is the average of the displacements of the two wheels.

We now apply the sine theorem in the **EDD'** triangle in order to find the equations for δx, δy and μ in the bogie reference frame $O_b$xy.

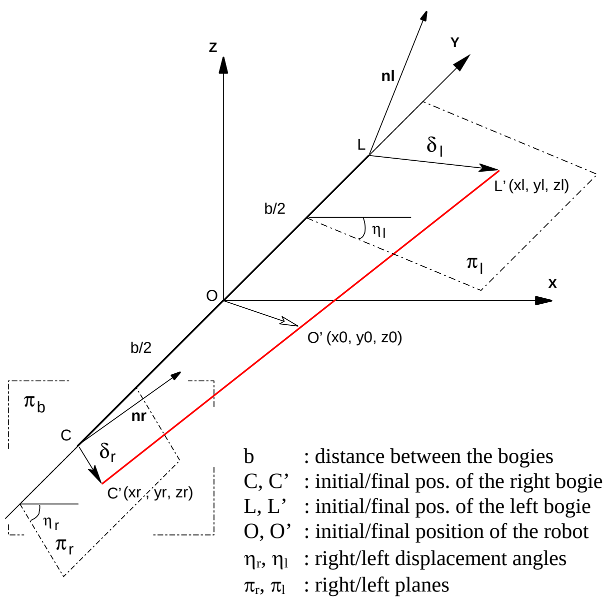
$$\delta x = t_1 + \left(\frac{h}{2} - t_2\right)\cos(\varepsilon) \qquad (2)$$

$$\delta y = \left(\frac{h}{2} - t_2\right)\sin(\varepsilon) \quad (3) \qquad \mu = \arctan\left(\frac{\delta y}{\delta x}\right) \quad (4)$$

where $t_1 = \dfrac{ER\cdot\sin(\gamma_r)}{\sin(\varepsilon)} - \dfrac{h}{2}$ and $t_2 = \dfrac{ER\cdot\sin(\rho)}{\sin(\varepsilon)}$

3.2 3D displacement

The previous section showed how to find the translations of the bogies. Now the computation of the robot's center displacement is presented (3D translation and yaw angle).



b	: distance between the bogies
C, C'	: initial/final pos. of the right bogie
L, L'	: initial/final pos. of the left bogie
O, O'	: initial/final position of the robot
η_r, η_l	: right/left displacement angles
π_r, π_l	: right/left planes
nr, nl	: normal vectors of π_r, π_l, // to Oxz
π_b	: plane // to O_{xz} and containing C

Figure 7. Main schematic for the 3D-Odometry

The first step consists in expressing the δ's and μ's in the robot's frame of reference (see previous section). η_r, η_l are the μ's angles expressed in the robot's reference frame. They define the planes π_r, π_l containing C' and L' (Eq. 5, 6). C' and L' are situated on circles centered in C and L with radius δ_r and δ_l in the planes π_r and π_l respectively. These considerations lead to the following equations:

$$\vec{n}_r \cdot \overrightarrow{CC'} = 0 \quad (5) \qquad \vec{n}_l \cdot \overrightarrow{LL'} = 0 \quad (6)$$

$$\delta_r^{\,2} = x_r^{\,2} + \left(y_r + \frac{b}{2}\right)^2 + z_r^{\,2} \qquad (7)$$

$$\delta_l^{\,2} = x_l^{\,2} + \left(y_l - \frac{b}{2}\right)^2 + z_l^{\,2} \qquad (8)$$

We now make the approximation that the smallest displacement takes place in the bogie plane. In the example of Fig. 7, δ_r is smaller than δ_l therefore $\overrightarrow{CC'}$ is in the plane π_b. This additional constraint is expressed by Eq. 9. Since the wheelbase b remains constant one can write Eq. 10. Finally, the vector $\overrightarrow{OO'}$ can be computed easily with Eq. 11.

$$\left(\frac{b}{2}\right)^2 + \delta_r^{\,2} = x_r^{\,2} + y_r^{\,2} + z_r^{\,2} \qquad (9)$$

$$b^2 = (x_l - x_r)^2 + (y_l - y_r)^2 + (z_l - z_r)^2 \qquad (10)$$

$$\overrightarrow{OO'} = \overrightarrow{OC'} + \frac{\overrightarrow{C'L'}}{2} \quad (11) \qquad Yaw = \frac{x_r - x_l}{b} \quad (12)$$

Solving the system of nine equations with nine unknowns formed by Eq. 5 to 11 leads to the solutions for $\vec{OC}'$, $\vec{OL}'$ and $\vec{OO}'$ (the nine unknowns). The yaw angle is computed with Equ. 12. The roll angle could also be computed but we use the value from the inclinometer since it is an absolute angle and therefore is not subject to error growth.

4. Experimental results

In order to test the equations presented above we drove the robot forward and computed the trajectory online with both our method and the standard method (see section 3). We set a proportional speed controller for the front bogies' wheels and added an integral term for the rear wheels. This allows the front and back wheels to have slightly different speeds and therefore limits the slippage during the transitions. We tested with proportionnal integral controllers on both bogie wheels but the results produced were worse than the one presented next.

The system has been tested in two different situations depicted in Fig. 10 (a) and (b) respectively. The true trajectory is an approximation. It is build with characteristic positions that can be easily computed knowing the shape of the obstacle, the geometrical model and the state of the robot.

Fig. 8 shows the z-coordinate corrections during the transitions in experiment 1. There is a positive correction when the bogie's first wheel starts climbing the obstacle and a negative one for the convex transition. These corrections are not present in the standard method: this is the main difference between the two methods.

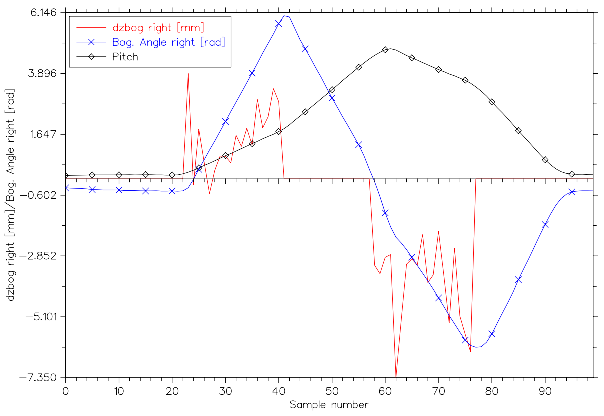


Figure 8. z-coordinate correction during the transitions of the first experiment. For clarity purpose both Pitch and Bogie Angle curve have been scaled (by a factor of 12 and 8 respectively)

The x and z coordinates of the robot's final position have been measured for every run. We did five runs for each experiment and computed the relative error. Fig. 9 shows the results corresponding to both experiments. One can see that the 3D-Odometry demonstrates much better performance. The sharper the transitions are, the better it does in comparison with the standard method. The errors accumulated by the 3D-Odometry method are due to different reasons. The first one we might think about is the wheel slippage. In case of slippage the calculated distance would be bigger than the measured one and the results presented in Fig. 9 can be interpreted that way.

Second experiment (1000 mm)

	Measured		3D-Odometry		Std. method	
	x	z	x	z	x	z
	1010	1000	1012	5	1056	19
	…	…	…	…	…	…
Average error			**0.2%**	**2.7%**	**5.5%**	**13.2%**

(a)

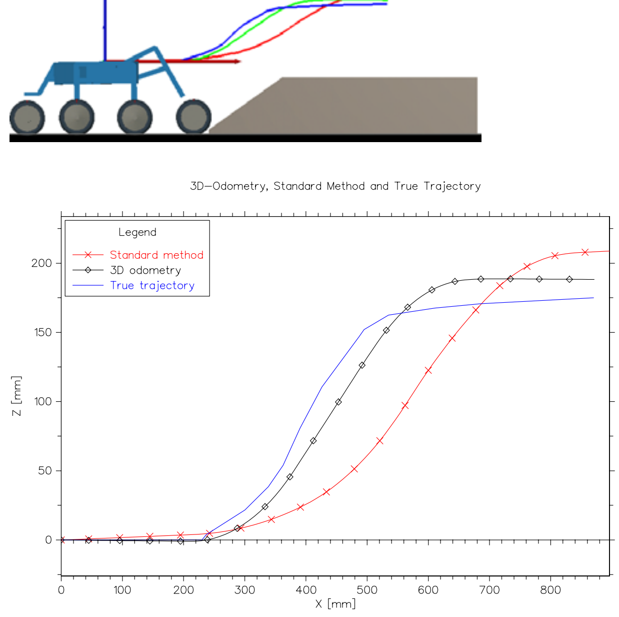
First experiment (870 mm)

	Measured		3D-Odometry		Std. method	
	x	z	x	z	x	z
	864	175	871	188	896	209
	873	175	876	187	904	210
	872	175	877	186	905	209
	875	175	878	185	908	208
	870	175	873	186	903	208
Average error			**0.5%**	**6.4%**	**3.7%**	**19.2%**

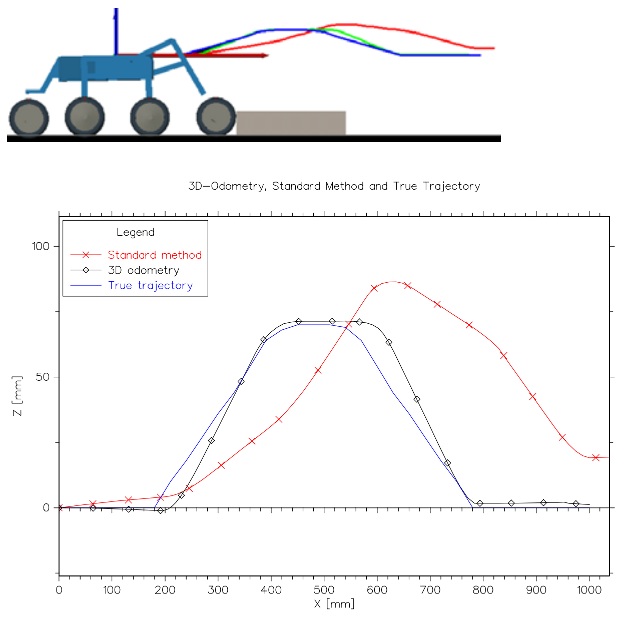
(b)

Figure 9. Average relative errors for the second and first experiment. Only the first run and the errors are represented for the second experiment. The difference between the two methods grows for sharper transitions.

However, slippage is not the biggest source of error in these experiments. We can see that the errors are mainly in the z direction and we are convinced that they are due to the sensors' offsets and non-linearities. Although we corrected the bogies' angle sensors for offsets we didn't account for non linearities. A difference of one degree leads to an error of around 15 mm in the z direction for a 870 mm horizontal motion. It is approximatively the



(a)



(b)

Figure 10. (a) First experiment. The robot starts in front of the obstacle, climbs a 35 degrees slope and stops on top after 870 mm in the x direction. The height of the obstacle is 175 mm. The transitions are relatively smooth for this experiment. (b) Second experiment. The robot goes over this 300 by 70 mm obstacle and stops after 1 meter. The transitions are sharp and since our method treats the transitions the 3D-Odometry curve respects the obstacle shape.

height error for the first experiment. An appropriate calibration will certainly improve the results. Finally, variation of the wheels' diameters and inaccuray in the mechanical dimensions are also factors of odometric errors.

Nevertheless, the 3D-Odometry produces better results than the standard method in both experiments. Accounting for transitions improves the position estimation significantly. This is even more obvious when considering sharp transitions like in the second experiment. Since there are a lot of discontinuities in rough terrain this will help to provide usable odometric information.

5. Future work

We are currently developing an error model for the 3D-Odometry which involves the creation of a 3D static model of the robot. The model will allow us to set a slippage probability which is inversely proportional to the normal forces on each wheel. Indeed, the less pressure on the tire, the more likely the wheel slips. This will also give a clue on how to correct for wheel diameter changes due to tyre compression.

We are also working on the low level trajectory controller for the robot. A good distribution of wheel speeds and torques based on the information provided by the model and the sensors allow optimizing the robot's motion. This reduces the wheel slippage, the overall energy consumption and increases the robot's climbing performances. We are investigating the use of torque control instead of speed control for the wheels.

As mentioned in the introduction, robust position tracking must integrate different types of sensors. For this reason we will fuse the motion information provided by the IMU, the stereovision system, the omnicam, and the odometry. Until now most of the applications integrate only two sensors e.g inertial and standard vision, inertial and stereovision, or odometry and vision.

6. Conclusion

Navigating in rough terrain is a complex task which requires the robot to be considered as a whole system. The quality of the odometry strongly depends on the mechanical design and the trajectory controller of the robot. Minimizing slippage not only limits odometric errors but also increases motion efficiency.

Thanks the passive and non-hyperstatic structure of Shrimp, odometric motion estimation can be used in rough terrain. It has been shown, that considering transitions significantly improves position estimates. Especially when the robot is overcoming sharp-shaped obstacles. Finally, the 3D-Odometry should expand the range of speed and surface roughness over which the rover should be able to go and keep track of its position.

ACKNOWLEDGEMENTS

The author would like to thank Stéphane Michaud, Daniel Burnier, Grégoire Terrien and Ralph Piguet for their help with the test bed vehicle. This project has been funded by the EPFL's research council and the ESA Solero project.

REFERENCES

[1] A. Mallet, S. Lacroix, and L. Gallo, "Position estimation in outdoor environments using pixel tracking and stereovision", *In International Conference on Robotics and Automation*, pages 3519-3524, San Francisco, CA (USA), April 2000.

[2] Olson C.F., Matthies L.H., Schoppers M., Maimone M.W., "Stereo ego-motion improvements for robust rover navigation", *IEEE International Conference on Robotics and Automation*, Proceedings, ICRA 01, Vol. 2, 2001.

[3] Strelow D., Mishler J., Singh S., Herman H., "Extending shape-from-motion to noncentral onmidirectional cameras", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Proceedings, Vol. 4 , 2001.

[4] Vieville T., Romann F., Hotz B., Mathieu H., Buffa M., Robert L., Facao P.E.D.S., Faugeras O.D., Audren J.T., "Autonomous navigation of a mobile robot using inertial and visual cues", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS '93, Proceedings, Vol. 1, 1993.

[5] Roumeliotis S.I, Johnson A.E., Montgomery J.F., "Augmenting inertial navigation with image-based motion estimation", *ICRA '02. IEEE International Conference on Robotics and Automation*, Proceedings, Vol. 4, Page(s): 4326 –4333, 2002.

[6] Andrade G., Amar F.B., Bidaud P., Chatila R**.,** "Modeling robot-soil interaction for planetary rover motion control", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Proceedings, Vol. 1, Page(s): 576 –581, 1998.

[7] Iagnemma, K., Shibly, H., Rzepniewski, A., and Dubowsky, S., "Planning and Control Algorithms for Enhanced Rough-Terrain Rover Mobility," *Proceedings of the Sixth International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS,* 2001.

[8] Siegwart R., Lamon P., Estier T., Lauria M., Piguet R., "Innovative design for wheeled locomotion in rough terrain", *Journal of Robotics and Autonomous Systems*, Elsevier, vol 40/2-3 p151-162.

[9] www-2.cs.cmu.edu/afs/cs/project/TCA/www/ipc/index.html

[10] E. T. Baumgartner, H. Aghazarian, A. Trebi-Ollennu, T. L. Huntsberger, M. S. Garrett, "State Estimation and Vehicle

Localization for the FIDO Rover", *Sensor Fusion and Decentralized Control in Autonomous Robotic Systems III*, SPIE Proc. Vol. 4196, Boston, MA, November 2000.

[11] Singh S., Simmons R., Smith T., Stentz A., Verma V., Yahja A., Schwehr K., “Recent Progress in Local and Global Traversability for Planetary Rovers”, *In International Conference on Robotics and Automation*, p1194-1200, San Francisco, April 2000.