ABSOLUTE MAP-BASED LOCALIZATION FOR A PLANETARY ROVER

Bach Van Pham, Artur Maligo, and Simon Lacroix

CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France

Université de Toulouse ; UPS , INSA , INP, ISAE ; LAAS ; F-31077 Toulouse, France

ABSTRACT

Navigating autonomously several kilometers in a planetary context without any communication with Earth operators raises several challenges. Among those, precise localization is of essential importance, and the availability of orbital data yields the possibility to estimate absolute position information. This paper proposes an approach that exploits a priori Digital Elevation Maps derived from orbiter data and on-board stereo imagery to estimate the absolute rover position. The approach extends the classic particle-filter based global localization scheme to the context of planetary rovers, where the environment is not structured. The article details the various required functions, the way each particle likelihood is evaluated, and means to manage a short number of particles. Experimental results illustrates the approach.

1. INTRODUCTION

Autonomous long range navigation for a planetary rover calls for a localization approach whose drift is bounded – otherwise the rover is not able to reach its goal or follow the planned itinerary. Localization using dead-reckoning techniques such as inertial navigation, wheel odometry or visual odometry Howard (2008) do not suffice for long range navigation as they eventually drift over time or the distance travelled. By memorizing and exploiting the detected landmarks, SLAM approaches (see *e.g.* Marks et al. (2009) in a planetary context) reduce the drift of dead-reckoning approaches, however navigating over long ranges does not yield loop-closures, and hence the drift remains unbounded.

The availability of high resolution orbiter imagery, such as provided by the HIRISE camera of the Mars Reconnaissance Orbiter, yields the possibility to develop map-based absolute localization approaches for planetary rovers, with a precision that is independent of the travelled distance. In this paper, we propose an approach which allows to localize the rover within a known map ("absolute navigation") with a precision of the order of 2 meters. The proposed system uses on-board stereo vision and a global Digital Elevation Map (DEM) of $1m$ resolution, similar to the DEM derived from HIRISE imagery. The approach relies on the use of a particular filter, as often proposed for map-based localization approaches, and the two main contributions are *(i)* the estimation of each localization hypothesis likelihood using an on-board built DEM from range data, and *(ii)* a specific management of the particles to maintain tractable computation times.

Outline the next section presents previous work on absolute navigation in planetary contexts, and the principle of particle filter approaches to map-based localization. Section 3 is the heart of the paper: it depicts the way a particle filter is designed to efficiently yield precise absolute position estimates. Section 4 presents field trials, using aerial imagery to mimic orbiter maps. Section 5 concludes the article, and presents an adaptation required to exploit the approach in Earth environments.

2. BACKGROUND

2.1. Absolute localization in planetary contexts

Skyline-based approaches First attempts to estimate the global position of the rover using global DEM relied on skyline matching Stein & Medioni (1992); Cozman & Krotkov (1997): the principle is to match a perceived skyline with a database of skylines pre-computed on the basis of the DEM. The database encodes skyline signatures, that are matched with the skyline extracted from a panoramic image taken by the rover. This solution has proved to efficiently find the global position of the rover in the "lost-in-space" situation, *i.e.* when the rover has no information on its initial position. It yields an estimation accuracy of 100-150 meter on a 30 m/pixel DEM Cozman et al. (2010), but the DEM must cover an area be large enough to allow the prediction of the skyline. Also the approach is not effective if the rover is within very rough areas or narrow corridors, where the horizon is hardly detectable.

Feature-based approaches Hwangbo et al. (2009) presents and approach that exploits large rocks extracted

from the high resolution satellite images. Surface rocks are extracted from the 3D point cloud produced by the on-board stereo camera Li et al. (2007), and the rover position is determined by finding a 2D transformation that gives the maximum number of matches between the rocks detected from the two sources. In order to reduce the search area for the 2D transformation, the authors propose to firstly match the local DEM with the global DEM. Experiments presented in Hwangbo et al. (2009) show that the system is capable of matching rock with a success rate of up to 50 percent. This approach is limited by the nature of the terrain: desert areas with no rocks, or on the contrary very rough ares, in which rocks are hardly defined are difficult to handle.

Carle & Barfoot (2010); Carle et al. (2010) proposed the use of a LIDAR to estimate the global position of the rover. An initial global DEM is processed to obtain a global database: topographic peaks (local maxima) are extracted with morphological operators, and a similar process is used to extract the topographic peaks from the local DEM. Peak matches are established with the "DARCES" process: several hypotheses of 3 matches are firstly obtained by comparing the inter-distance of the peaks. Then, these hypotheses are compared to each other to select the one with the 2D transformation that gives the higher number of matches. The estimation is then passed to a SLAM module. The experimental results obtained with 1.5 km range and 360 degree LIDAR scan give an average precision of 22 meter with a global DEM of 17 m/pixel resolution. Other noteworthy work compare global and local DEM with spin-images Vandapel et al. (2006). Note however that planetary contexts preclude the use of LIDARs, due to their heavy weight and high power consumption. In addition, LIDARs provide data at much longer ranges than stereovision, making this approaches hardly applicable to planetary rovers.

2.2. Map-based localization using a particle filter

A well-known framework used to estimate the global position of a robot is provided by Monte Carlo Localization (MCL) Dellarert et al. (1999), also called particle filter. In this method, the posterior probability density function corresponding to the robot global position is approximated by a set of samples, the so-called particles. By doing this, MCL can take into account the multiple possible positions that arise in the lost-in-space problem. Particles are associated to weights which are computed using an observation model and a prior global map of the environment. In Guivant & Katz (2007), an urban context implementation of the method is proposed, in which the authors use a Route Network Description File (RNDF) as the prior map. RNDF maps provide specific information about the roads structure and thus can be efficiently used to localize vehicles in this context. In another urban environment example Levinson et al. (2007), infrared reflectivity maps of road surfaces are built, and LIDAR sensors are used to acquire reflectivity information that are then matched with the maps data. Both approaches make use of urban oriented prior models and matching techniques, which are clearly not adapted to planetary unstructured landscapes.

Outdoor generic environments are considered in Silver & Stentz (2011), where the authors show that observation models can be learned directly from the alignment between the prior map and the sensors data. However, this alignment is computed using high quality GPS information, which is not available in a planetary context. A variation of the particle filter, the Rao-Blackwellized particle filter Doucet et al. (2000); Grisetti et al. (2007), is often used in SLAM applications where each particle has its own global map to which observations are compared. In our problem, the prior global map is provided by an orbiter and such a variation is not adapted.

3. PROPOSED APPROACH

In our approach, a local DEM is built from stereovision data gathered by the rover, and is matched to the global DEM provided by the orbiter. We chose a particle filter estimation scheme to cope with the fact that the robot initial position parameters can be totally unknown, or only bounded, without any further information. For computation efficiency purposes, instead of sampling particles in a continuous space, our implementation considers a discretized space. Additionally, particles carry an uncertainty with them, modelled by a Gaussian distribution. The discretized sampling and the Gaussian nature of particles is used to manage them as the rover moves.

3.1. Discretized Particle Filter

Given a map M, odometry measurements u and observations z, the key idea of the particle filter is to estimate the posterior density of the current rover state x_k based on all the information available: $p(x_k|z_{1:k}, u_{0:k}, M)$. The state is often chosen as $x = [x, y, \theta]$, in environments where we a planar surface assumption can be made, or as $x = [x, y, z, \psi, \theta, \phi]$, in outdoor complex environments. The posterior is computed in three steps: *(i)* prediction, where particles are sampled according to a motion model, *(ii)* update, where the compute particle weights are computed using a likelihood function given by an observation model, and *(iii)* re-sampling, where particles are re-sampled according to their previously computed weights. Our discrete particle filter, however, includes an initialization step and adds an extra sampling step after the prediction, as further explained in this section.

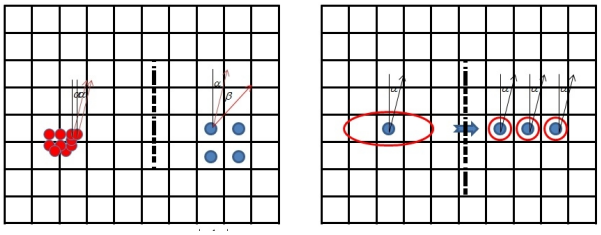
Unlike the classic particle filter, where particles are sampled in a continuous manner, we choose to sample them in a discrete manner. The discrete space has the same (x, y) resolution as the global DEM, while the angular resolution of the heading can be defined arbitrarily[1]. In

[1]The roll and pitch angles are nor considered, assuming that they are precisely provided by the inertial navigation sensors of the rover.

the discretized filter, only one particle is assigned to a discrete position: this is is done by merging particles that occupy the same cell, as shown Figure 1.a). This is meant to reduce the number of particles, as there would be no difference between the predicted observations of particles belonging to the same cell. The discrete position of a particle i is defined according to its coordinates (x_i, y_i) and its heading θ_i:

$$S_i = (\lfloor \frac{(x_i - x_O)}{r_x} \rfloor, \lfloor \frac{(y_i - y_O)}{y_x} \rfloor, \lfloor \frac{\theta_i}{r_\theta} \rfloor) \qquad (1)$$

were x_O, y_O is the origin of the global map, r_x, r_y is the resolution of the global map, and r_θ is the angular resolution of the heading. This discretization principle is similar to the one of grid-based Markov localization approaches Burgard et al. (1998, 1996), with the difference that such approaches consider all cells and angles, while our particle filter only maintains the state of existing particles.



a) *b)*

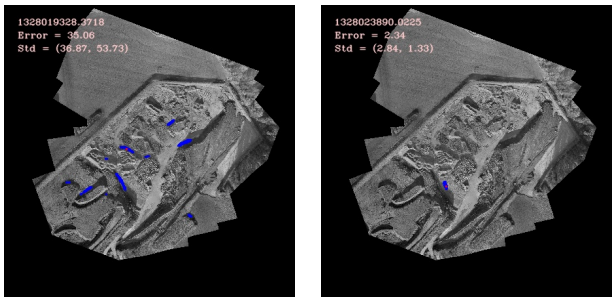
Figure 1. a): comparison between continuous particles (left), and discretized ones (right). b): Gaussian uncertainty and particle sampling by decomposition.

In addition to the discretization process, our filter models the uncertainty associated to a particle by a Gaussian distribution. This technique complements the filter and could be seen as a way of modeling the information loss caused by the discretization. The uncertainty is also taken into account in the additional sampling step, when particles are added next to particles whose distributions have high variances, as illustrated Figure 1.b).

3.2. Algorithm Outline

The overall algorithm consists of 5 steps. The initialization is only applied at the beginning of the mission or when the filter is reset. The other four steps, run sequentially and repeatedly, are prediction, sampling, update and re-sampling. Apart from the sampling step which is specific to our implementation, these are the classic particle filters steps.

1. **Initialization**: To avoid convergence to a wrong local maximum, the correct position of the rover must be occupied by one of the initial particles. To ensure this, a full search is performed and the likelihood of all the discrete cells is evaluated with the same map-matching process used in the update step of the algorithm. Particles are then generated in the positions that obtained the highest likelihoods. Figure 2.a) shows the initial set of particles in a typical lost-in-space situation. This global evaluation is necessary when no prior knowledge about the rover position is known. The Gaussian distributions of the particles are initialized by setting their standard deviations to half of the global DEM resolution and to half of the predefined angular resolution. The weight of each particle is set to $\frac{1}{N}$, N being the initial number of particles. The initialization step is the most computationally expensive: this can be alleviated by prior knowledge on the robot position, and by discarding of the global DEM physically impossible areas such as steep slopes, cliffs, etc.



a) *b)*

Figure 2. a) initialization of particles. b) distribution of particles along a steep slope (the ground truth is the red spot on both images).

2. **Prediction**: when a new motion measurement u_k is received, the position of each particle i is modified according to:

$$x_k^i = x_{k-1}^i \oplus u_k \qquad (2)$$

In our experiments, we use the Visual Odometry (VO) DROID Harris (1993) motion measures. The Gaussian distribution of each particle is modified by simply adding the VO variances. In the classic particle filter, particles would be sampled according to this motion measures and its associated uncertainty. Here, however, we represent this uncertainty explicitly in the Gaussian distribution, and hence we can directly increase the distribution variances to take the uncertainty into account. The sampling step complements this operation and uses the Gaussian modeling to sample additional particles around highly uncertain ones.

3. **Sampling**: this step is introduced to add new particles and ensure that the rover actual position is still represented by one of the considered particles. Given a particle i, its Gaussian distribution standard deviations are compared with the coordinate resolution and with the angular resolution. If the standard deviation is sufficiently large, new particles are created in the discrete positions around the original one

(Figure 1.b). After this process, the Gaussian distribution of the particle i and the newly created particles are reset to half of the global resolution and angular resolution. New particles are added in a deterministic manner, which is not the case in normal sampling operations where particles would be sampled in a random manner. By doing this, we ensure that the filter will always consider all the possible positions.

4. **Update**: in our application we use a pair of stereo cameras and thus our observations z_k consist of the acquired 3D point clouds. These measurements are not directly used but are merged within a local DEM m_k centered around the rover. We keep updating the local DEM with new observations and analyzing its quality before actually using it in the likelihood calculation (details on the local map building process are given in section 4.2). Using the global DEM, we define a predicted local map t_k^i for each particle i using its associated position x_k^i. The likelihood of a particle i is evaluated with the computation of the Zero mean Normalized Cross-Correlation score (ZNCC) of the DEMs:

$$ZNCC_k^i = \frac{1}{n-1} \sum_{x,y} \frac{(m_k(x,y) - \bar{m_k})(t_k^i(x,y) - \bar{t_k^i})}{\sigma_{m_k} \sigma_{t_k^i}} \tag{3}$$

where (x, y) is the index of the non-empty cells of the local map m_k, t_k^i is the predicted map for particle i, and $\bar{m_k}$, $\bar{t_k^i}$, σ_{m_k} and $\sigma_{t_k^i}$ are respectively the means and standard deviations of the local and the predicted DEMs. Simpler scores could have been used (*e.g.* the Sum of Absolute Differences), but ZNCC proved to give more discriminant measures. Since ZNCC scores belong to $[-1; 1]$, negative values are set to zero to avoid assigning negative weights to particles. The weight of the particle i is updated using

$$w_k^i = w_{k-1}^i * ZNCC_k^i \tag{4}$$

5. **Re-Sampling**: as in Doucet et al. (2000); Grisetti et al. (2007); Marks et al. (2009), the effective number of particles is calculated by

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (\tilde{w}_k^i)^2} \tag{5}$$

where $\tilde{w}_k^i$ represents the normalized particles weights. If the value of N_{eff} is smaller than $\frac{N}{2}$, the re-sampling operation is carried out and particles are re-sampled with probabilities given by their weights. Weights are then reinitialized to $\frac{1}{N}$. Figure 2.b) shows the remaining particles after applying a few iterations of the filter: they converged around the actual robot position, near a steep slope.

4. EXPERIMENTS

4.1. Generation of a global DEM with a UAV

Our experiments were carried on in a sand quarry. To mimic the resolution of the global DEM of a planetary mission (namely 1 m/pixel resolution with HIRISE DEM), we used a UAV to build a global DEM of the test area from a sequence of aerial images (Figure 3). The global DEM is built thanks to a commercial bundle adjustment software, that produces 5 cm/pixel DEM and orthoimage (Figure 4). Ground markers whose position has been assessed with a cm-accuracy RTK-GPS have been used to geo-reference the global map, and finally the global DEM is down-sampled to 1 m/pixel.

Figure 3. Examples of aerial images used to generate the global DEM

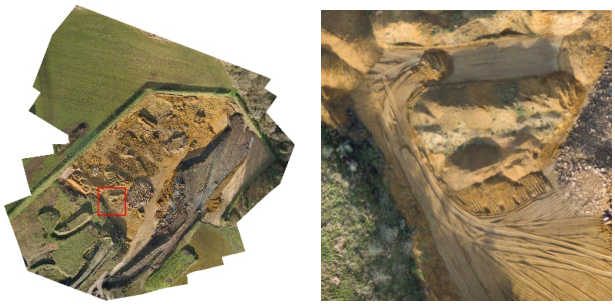


Figure 4. Orthoimage of the $450 \times 450m^2$ mapped area (the right inset is a zoomed view of the red square in the left image).

4.2. Trials

4.2.1. Parameter setup

We used the robot Rimmer of RAL Space Woods et al. (2012), that was manually driven in the mapped area of the sand quarry. The stereovision bench is a Point Grey Bumblebee VGA system, tilted down of 15^o. The rover speed is 0.2 m/second, and the pictures are acquired at 5 Hz (according to a requirement of the VO). The VO, stereovision, DEM building and absolute localization algorithms run on-board of a Core i7 2GHz Linux PC.

Stereovision runs at 1 Hz on VGA images, and the 3D points within a range of 15 meters from the rover are re-

tained[2], and merged in a local 1 m/pixel resolution DEM of $30 \times 30m^2$. This size has been defined considering two factors: it requires very little memory and makes the absolute localization run faster, and within such a range, the errors of VO have a negligible impact on the building of the DEM.

We assume that the absolute heading (yaw) angle of the rover is known thanks to the use of a Sun sensor, which is mimicked with GPS ground truth measures, that yield a heading precision of ± 3 degrees. However, our approach can handle no prior heading information, at the cost of more computational time. The filter update step is only applied when two requirements are satisfied: with respect to the previous update, the rover must move far enough and the local DEM must have been significantly updated. The particles position and Gaussian distribution are however continuously updated as the rover moves. All of the tests presented here were made with the lost-in-space situation, the search is made on the entire global map. For all tests, the initial number of particles is 500, and the maximum number of particle allowed is 1000.

4.2.2. Results

Estimation error To estimate global position of the rover, we calculate the mean and standard deviation of the global position of the particles. Figure 5 illustrates the estimation results obtained with (case A) and without (case B) heading information. It is important to note that the heading information is only used in the particle initialization step. The top image in Figure 5 shows that the estimation error of case A initially is higher than case B. In reality, this initial error depends on the shape of the global DEM or more concretely on the similarity between the local DEM and some particular regions in the global DEM. Then, as the rover starts to move, the error of case A raises: this is a particular case, caused by the fact that a cluster of wrong particles are removed, which here shifts the mean position further away from the true position.

Figure 5 bottom shows the evolution of the standard deviation. In contrast to the rise of the estimation error due to the shift of the mean position, the reduction in the standard deviation value means that the particles converge. The standard deviation values shows that the algorithm in case A converges faster and stabilises after 58 meters of traveled distance, in comparison of 78 meters of case B. Once the estimation is stabilised (with an error of about 1.0 m), there is no difference between the two cases. The algorithm has naturally more chance to converge with prior heading data, but it does not necessarily converges faster: the convergence rate depends much more on the shape of the traversed terrain.

Influence of the angular resolution The angular resolution plays a very important role. Fine angular reso-

[2]The Bumblebee stereo bench has a $24cm$ baseline, and points below $15m$ have errors compatible with a 1 m/pixel DEM.

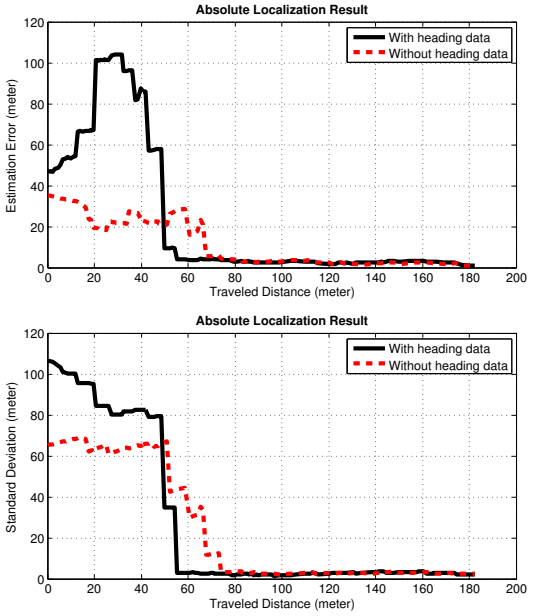


Figure 5. Absolute localization results with ("case A") and without ("case B") heading information with 1 m/pixel resolution global DEM and 3 degrees angular resolution.

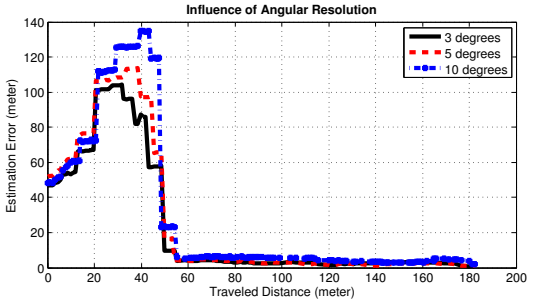


Figure 6. Influence of the angular resolution: results with 3, 5 and 10 degrees angular resolutions, without prior heading information.

	Known heading			Unknown heading		
Ang. res.	3^o	5^o	10^o	3^o	5^o	10^o
Run-time(s)	4.78	4.77	4.75	186.95	111.29	56.76

Table 1. Initialization time with 3, 5, 10 degrees angular resolutions, with and without prior heading information.

lution allows better result both in global position and in absolute heading estimation. A coarse angular resolution, in contrast, needs less computational time. The top image in Figure 6 compares the estimation result of 3 , 5 and 10 degrees angular resolution with the case without prior heading information. As shown in the plot, the case with 3 degrees angle converges a little faster than the other cases, whereas the average estimation error of the 3 cases is almost similar. However, there are some turn-on-the-spot moments when the yaw estimation error raises, the cases with 3 and 5 degrees angle work better than the one with 10 degrees.

Performance The particle initialization is run used once and is by far the most computational expensive step. Computations depend on the size and resolution of the local and global DEMs and on the angular resolution. discretization. Table 1 presents the initialization run time for the 3, 5, 10 degrees angular resolutions, with and without prior heading information. It shows that the angular resolution has no influence when prior heading information is used.

The time required by the particle updates are much smaller, and reaches at most 0.1 seconds for the worst cases.

Figure 7 shows the number of particle used during three tests without prior heading data. All tests are initialized with 500 particles. Once the rover moves, the Gaussian distribution are updated using the VO estimation, and new particles are generated by decomposing this Gaussian distribution, while wrong particles are removed during the re-sampling step. These generations and removals of the particles cause the zig-zag appearance of the curves, which is more marked with 3 degrees angular angle, as more particles are generated given the same Gaussian distribution.

5. CONCLUSION

5.1. Discussion

Absolute localization is an essential functionality for autonomous long range navigation of planetary rovers, but has received little attention. Skyline matching approaches are rather suited for flat areas with mountainous landscapes on the horizon, where the skyline is well defined, whereas feature-based approaches require the

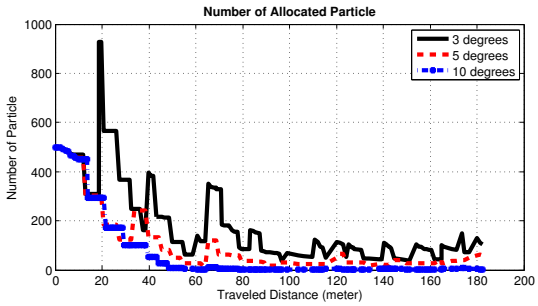


Figure 7. Number of particles during 3 without-prior-heading-data tests.

presence of easily extractable features. The approach proposed in this paper complements well with these approaches, as it works particularly well on areas with strong terrain reliefs. An ideal absolute localization system would required the integration of the three approaches: one could then define a pre-processing of the global DEM that would state where each approach is expected to behave well. Experiment results show that the method can provide localization estimate with $2m$ accuracy on a $1m$ resolution global DEM. In addition, the adaptations made to the particle filter yield a system with a low computation footprint. There remain however some drawbacks: during the initialization phase, the actual position of the rover must be among the particle candidates, and like any particle filter based method, good particles can be removed during the re-sampling phase.

5.2. Future Work

The proposed algorithm has been adapted to urban environments and to the use of panoramic long range LIDAR. Figure 8 presents results obtained on a 0.1 m/pixel DEM of a parking lot built by the robot during a learning phase during which cm-accuracy GPS was available. Two different resolutions of the global DEM are used: a global DEM with 1 m/pixel resolution is exploited to initialize the particles, and a global DEM with 0.1 m/pixel is used to update the particles. To further reduce computations, a Hough-based ground extraction algorithm segments 3D point of the local $30 \times 30m^2$ DEM into ground and off-ground points, and only off-ground 3D points are used to update the particles.

ACKNOWLEDGMENTS

Most of the work presented in this paper were sponsored by the ESA-founded SEEKER project Woods et al. (2012) with the participation of Roke Manor, Scisys, RAL Space, MDA, BAE System and LAAS-CNRS.

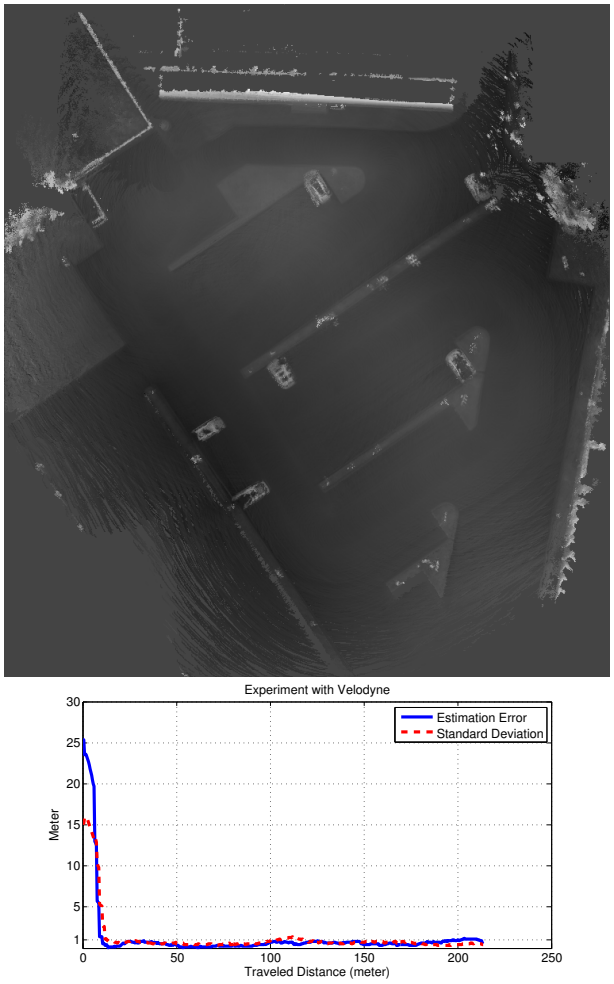


Figure 8. Top: 0.1m/pixel DEM of a $100 \times 100m^2$ *parking lot. Bottom: localization results on a* $200m$ *long trajectory.*

REFERENCES

Burgard, W., Derr, A., Fox, D., & Cremers, A. 1998, in Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on, Vol. 2, 730 –735 vol.2

Burgard, W., Fox, D., Hennig, D., & Schmidt, T. 1996, in In Proceedings of the Thirteenth National Conference on Artificial Intelligence, Menlo Park (AAAI, AAAI Press/MIT Press), 896–901

Carle, P. J. & Barfoot, T. D. 2010, in International Conference on Robotics and Automation

Carle, P. J., Furgale, P. T., & Barfoot, T. D. 2010, Journal of Field Robotics, 27, 344

Cozman, F. & Krotkov, E. 1997, in International Conference on Robotics and Automation

Cozman, F., Krotkov, E., & Guestrin, C. 2010, Outdoor Visual Position Estimation for Planetary Rovers

Dellarert, F., Fox, D., Burgard, W., & Thrun, S. 1999, in International Conference on Robotics and Automation

Doucet, A., De Freitas, N., Murphy, K., & Russell, S. 2000, in Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence

Grisetti, G., Stachniss, C., & Burgard, W. 2007, Robotics, IEEE Transactions on, 23, 34

Guivant, J. & Katz, R. 2007, in Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, 1079 –1084

Harris, C. 1993 (Cambridge, MA, USA: MIT Press), 263–284

Howard, A. 2008, in Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, 3946 –3952

Hwangbo, J. W., Di, K., & Li, R. 2009, in ASPRS Annual Conference

Levinson, J., Montemerlo, M., & Thrun, S. 2007, in Proceedings of Robotics: Science and Systems, Atlanta, GA, USA

Li, R., Di, K., Howard, A. B., et al. 2007, J. Field Robot., 24, 187

Marks, T. K., Howard, A., Bajracharya, M., Cottrell, G. W., & Matthies, L. H. 2009, J. Field Robot., 26, 26

Silver, D. & Stentz, A. T. 2011, in IEEE International Conference on Intelligent Robots and Systems (IROS)

Stein, F. & Medioni, G. 1992, in International Conference on Robotics and Automation

Vandapel, N., Donamukkala, R. R., & Hebert, M. 2006, The International Journal of Robotics Research, 25, 31

Woods, M., Shaw, A., Maddison, B., et al. 2012, in International Symposium on Artificial Intelligence, Robotics and Automation in Space, Turin, Italy