Autonomous Mobile Robot Navigation for Planet Exploration The EDEN Project

Raja Chatila Simon Lacroix
Stéphane Betgé-Brezetz Michel Devy Thierry Siméon

LAAS-CNRS
7, Ave. du Colonel Roche – 31077 Toulouse Cedex, France
email: {raja,simon,betge,michel,nic}@laas.fr

Abstract

A planetary rover cannot rely in general on permanent and immediate communications with a control station. This precludes direct teleoparation of its motions. It has therefore to be endowed with a large autonomy in achieving its navigation. We have designed and experimented with the mobile robot ADAM a complete system for autonomous navigation in a planetary-like environment. We describe this work in this paper. The approach is primarly based on the adaptation of the perception and motion actions to the environment and to the status of the robot. The navigation task involves several levels of reasoning, several environment representations, and several action modes. The robot is able to select sub-goals, navigation modes, and perception actions according to the situation.

1 Introduction

Navigation is the basic task that has to be solved by a planetary rover. Effectiveness in achieving the task is essential given the constraints of energy.
Navigation is in general an incremental process that can be summarized in four main steps:

1. Environment perception and modelling: any motion requires a representation of the local environment at least, and often a more global knowledge. The navigation process has to decide where, when and what to perceive.
2. Localization: the robot needs to know where it is with respect to its environment and goal.
3. Motion decision and planning: the robot has to decide where or which way to go, locally or at the longer term, and if possible compute a trajectory for avoiding obstacles and terrain difficulties;
4. Motion execution: the commands corresponding to the motion decisions are executed by control processes - possibly sensor-based and using environment features.

The complexity of the navigation processes depends on the general context in which this task is to be executed (nature of the environment, efficiency constraints,...) and should be adapted to it.
Navigation in outdoors environments was addressed either for specific tasks, e.g., road following [1], or motion in rather limited environment conditions [2; 3]. Ambler [4] is a legged machine and the main problem was computing footholes. The UGV [6], as well as Ratler [4] achieve autonomous runs avoiding obstacles, but not coping to our knowledge with irregular terrain.
In a planetary-like environment, the robot has to cope with different kinds of terrain: flat with scattered rocks or irregular/uneven in which its motion control system should take into account its stability. Limitations of computing capacities (processing power and memory) and of power consumption on the one hand, and the objective of achieving an efficient behaviour on the other hand, have lead us to consider an *adaptive* approach to mobile robot navigation in natural environments.
The objective of the EDEN project described here is to achieve a canonical navigation task, i.e., the task: `‘‘Go To [goal]’’`, where the argument `goal` is a distant target to reach autonomously. Any more complex robotic mission (exploration, sample collection...) will include one or more instances of this task. Given the variety of terrains the robot will have to traverse, this task involves in our approach several levels of reasoning, several environment representations, and three different motion modes. It raises a need for a specific decisional level (the *navigation* level), that is in charge of deciding which environment representation to update, which subgoal to reach, and which motion mode to apply. This level, which is a key component of the system, controls the perception and motion activities of the robot for this task.
The paper is organised as follows: the next section

presents the adaptive approach to autonomous navigation in unknown outdoor environments. Section 3 then presents how the terrain representations required by the navigation decisional level is incrementally built on the basis of 3D data, produced either by a Laser Range Finder (LRF) or by stereo-vision. The algorithms that perform the selection of sub-goals and perception tasks (*i.e.* that compose the navigation level) are described in section 5. Finally, we present experimental results and conclude the paper.

2 A General Strategy for Navigation in Outdoor Unknown Environments

2.1 An adaptive approach

Using its own sensors, effectors, memory and computing power efficiently is certainly a feature that we would like to implement in a robot. This becomes even more a necessity for a planetary rover which has important limitations on its processing capacities, memory and energy. To achieve an efficient behavior, the robot must adapt the manner in which it executes the navigation task to the nature of the terrain and the quality of its knowledge on it [13; 8]. Hence, three motion modes are considered:

- A **reflex** mode: on large flat and lightly cluttered zones, it is sufficient to determine robot locomotion commands on the basis of a goal (heading or position) and informations provided by "obstacle detector" sensors. The terrain representation required by this mode is just the description of the borders of the region within which it can be applied;

- A **2D planned** mode: when the terrain is mainly flat, but cluttered with obstacles, it becomes necessary for efficiency reasons to plan a trajectory. The trajectory planner reasons on a binary description of the environment, which is described in terms of empty/obstacle areas.

- A **3D planned** mode: when the terrain is highly constrained (uneven), collision and stability constraints have to be checked to determine the robot locomotion commands. This is done thanks to a 3D trajectory planner [9], that reasons on a fine 3D description of the terrain (an elevation map or numerical terrain model [10]);

The existence of different motion modes enables more adapted and efficient behavior, at the price of complicating the system since it must be able to deal with several different terrain representations and motion planning processes. It must especially have the ability to determine which motion mode to apply: this is performed thanks to a specific planning level, the *navigation planner* which requires its own representations.

2.2 The Navigation Planner

We assume the terrain on which the robot must navigate is initially unknown, or mapped with a very low resolution. In this last case, it is possible for a user to specify a graph of possible *routes*, *i.e.* corridors that avoid largedifficult areas, and within which the robot has to move autonomously. In this context, the navigation task `''Go To''` is achieved at three layers of planning (figure 1):

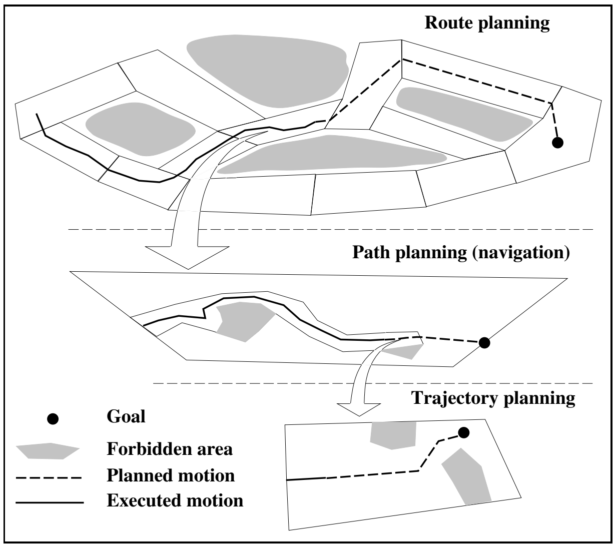

Figure 1: *Three levels of planning*

- *route planning* which selects long-term paths to the goal on the basis of the initial informations when available (the route map that may cover several kilometers). The route planner selects a sub-goal for the navigation planning level;

- *navigation planning* which reasons on a global qualitative representation of the terrain, built from the data acquired by the robot's sensors. The navigation planner selects the next perception task to perform, the sub-goal to reach and the motion mode to apply: it selects and controls the trajectory planner;

- *trajectory planning* which determines the trajectory to execute (in one of the above-mentioned three motion modes) to reach the goal defined by the navigation planning level.

Splitting the decisional process into three layers of planning has the advantage of structuring the problem: each planning layer controls the one that is directly below by specifying its goal and its working domain. It has also the great advantage of helping to analyze failures: when a planner fails to reach its goal, it means that the environment representation of the immediatly higher layer is erroneous and therefore that it has to be revised.
The navigation planner (layer 2) is *systematically* activated at each step of the incremental execution of the

task: each time 3D data are acquired, they are analyzed to provide a description of the perceived zone in terms of navigation classes. This description is fused with the model acquired so far to maintain a *global qualitative representation* of the environment, (the *region map*), on which the navigation planner reasons to select a subgoal, a motion mode, and the next perception task to execute.

2.3 Several terrain representations

Each of the three different motion modes requires a particular terrain representation. The navigation planner also requires a specific terrain representation, and during navigation, an exteroceptive localisation process has to be activated frequently, which requires an other terrain representation. Aiming at building a "universal" terrain model that contains all the necessary informations for these various processes is extremely difficult, inefficient, and moreover not really useful. It is more direct and easier to build different representations adapted to their use: the environment model is then *multi-layered and heterogeneous*. Several perception processes coexist in the system, each dedicated to the extraction of specific representations: perception is *multi-purpose*.

Figure 2 presents the various terrain representations required during navigation: one can distinguish the numerical terrain model [10] necessary to the 3D trajectory planner, the region map dedicated to the navigation planner, and three different ways to build a localisation model: *(i)* by modelling objects (rocks) with ellipsoids or superquadrics [11], *(ii)* by detecting interesting zones in the elevation map represented by a B-spline based model [12], or *(iii)* by detecting poles in the 3D raw data. Coherence relationships between these various representations are to be maintained when necessary.

3 Building the region map

For the purpose of navigation planning, a global representation that describes the terrain in terms of navigation classes (flat, uneven, obstacle, unknown) is required. This representation enables to select the adequate motion mode. We focus in this section on the algorithms developed to build such a model from 3D data (produced either by a laser range finder or a correlation stereovision algorithm).

3.1 3D data classification

Applied each time 3D data are acquired, the classification process produces a description of the perceived areas in term in *terrain classes*. It relies on a specific discretization of the perceived area respecting the sensor resolution (figure 3), that defines "cells" on which different characteristics (attributes) are determined: density (number of 3D points in a cell compared with a nominal density defined by the discretization rates), mean altitude, variance on the altitude, mean normal vector and corresponding variances...

A non-parametric Bayesian classification procedure is used to label each cell: a learning phase, based on prototypes classified by a human, leads to the determination of probability density functions, and the classical Bayesian approach is applied, which provides an estimate of the probability for each possible label. A decision function that prefers false alarms (*i.e.* labelling a flat area as obstacle or uneven) instead of the non-detections (*i.e.* the opposite: labelling an obstacle as a flat area) is used (figure 4). A simpler but faster technique based on thresholds on the cell attributes has also been implemented.

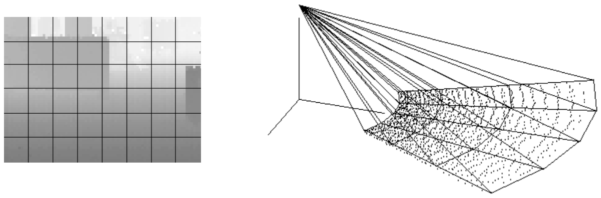

Figure 3: *Discretization used for the classification procedure: a regular discretisation in the sensor frame (left: a 3D image is represented as a video image, where the gray levels corresponds to point depth) defines a discretization of the perceived zone that respects the sensor resolution (right)*

This technique proved its efficiency and robustness on several hundreds of 3D images. Its main interest is that it provides an estimate of the confidence of its results: this information is given by the *entropy* of a cell. Moreover, a statistical analysis of the cell labelling confidence as a function of its label and distance to the sensor (directly related to the measure uncertainty) defines a predictive model of the classification process.

3.2 Incremental fusion

The partial probabilities of a cell to belong to a terrain class and the variance on its elevation allow to perform a fusion procedure of several classified images. The fusion procedure is performed on a bitmap, in the pixels of which are encoded cell attributes determined by the classification procedure (label, label confidence, elevation and variance on the elevation).

3.3 Model structure and management

For the purpose of navigation planning, the global bitmap model is structured into a region map, that defines a connection graph. Planning a *path* (as opposed to planning a *trajectory*) does not require a precise evaluation of the static and kinematic constraints on the robot: we simply consider a robot point model, and therefore perform an obstacle growing in the bitmap before segmenting it into regions. The regions define a connection graph, whose nodes are on their borders, and whose arcs correspond to a region crossing (figure 6).

In order to satisfy memory constraints, the global model is explicited as a bitmap only in the surroundings of the

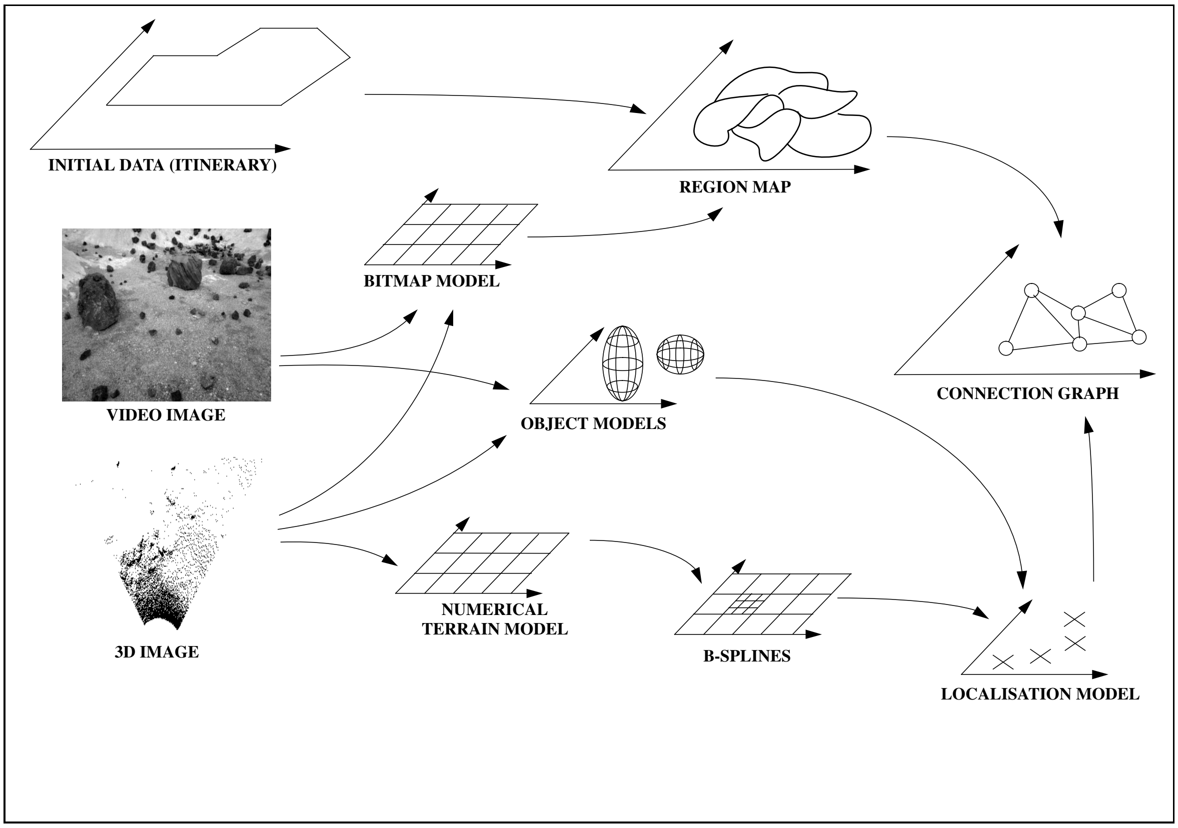

Figure 2: *The various representations used in the system. Arrows represent the constructive dependencies between them*

robot's current position, and the region model (much more compact) is kept in memory during the whole mission (figure 7).

4 Object-based representations

In order to be able to localize itself with respect to environment features, it is necessary for the robot to build representations of objects that it could recognize and use as landmarks.

When the terrain has been classified as flat but scattered with obstacles, we extract by a segmentation process the objects that are lying on the ground. Some of these objects that are salient and easily recognisable (e.g., peak-like) can be considered as landmarks for localisation.

To each object corresponds a referential - within which its surface is to be represented - this referential being itself located in a global coordinate frame in which it is represented by an uncertain state vector and a variance-covariance matrix. During navigation, the robot selects some objects possessing remarkable properties (visibility, selectivity, accuracy) that make them easily recognisable, and uses them as landmarks for anchoring the environment model and to locate itself. Landmarks will actually be specific features on such objects. As the robot moves the updating of the model is based on an extended Kalman filter. To recognize objects and landmarks, we use the two following complementary criteria:

- Comparison of the *global shape and features* of the objects (i.e., their intrinsic parameters).
- Comparison of the *relative position* of the objects themselves in order to obtain a *consistent* set of matchings. This is more interesting for landmarks because we could extract *precise* features on them.

A general approach will consist in exploiting the two criteria, but in our case, objects (mainly rocks) have similar shapes, and the position criterion will be proeminent.

The feature that defines the landmark is the object top, if it is salient enough. Landmarks here are local and should be *lower* that the robot sensor so that we can guarantee that the topmost point is perceived without ambiguity.

In a segmented 3D image, an object is selected as candidate landmark if:

1. It is not occluded by another object or by the image contour. If this object is occluded, it will be both difficult to recognize and to have a good estimate of its top, which may be not in the visible part.
2. Its topmost point is accurate. This is function of sensor noise, resolution and object top shape.
3. It must be in "ground contact". This is not an essential condition, but it reduces the recognition ambiguity.

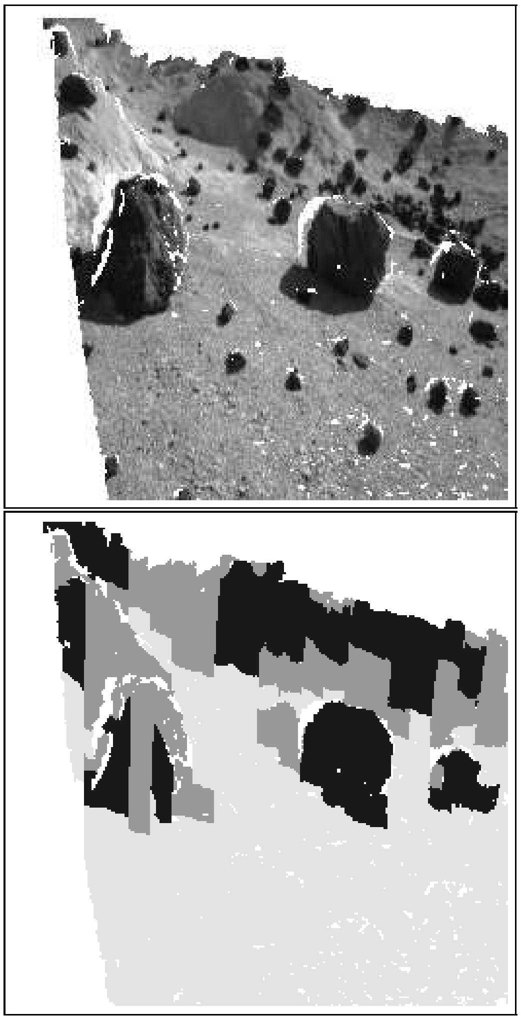

Figure 4: *Classification of a correlated stereo image: correlated pixels (top) and reprojection of the result in the camera frame (bottom - from clear to dark: unknown, flat, uneven and obstacle)*

In the model, an object is a landmark if it has been classified as landmark in at least one of its perceptions.

4.1 An Example of Landmark Selection

The top part of Figure 8 represents ten objects segmented in a 3D image. Six objects are selected as landmarks by the robot (objects number 1 to 6). Object 7 has not been selected because it is occluded by the image contour. This is also the case for object 9 (the hill) which top is not enough precisely defined. Object 8 is not occluded but it is not precise. Finally, object 10, which is on the hill, has not been selected because it is not in contact with the ground.

The bottom part of Figure 8 shows the environment map with the six selected landmarks and their corresponding uncertainties (ellipsoid projections corresponding to Gaussian distributions drawn at 99%). We can easily notice that precision decreases when the distance increases.

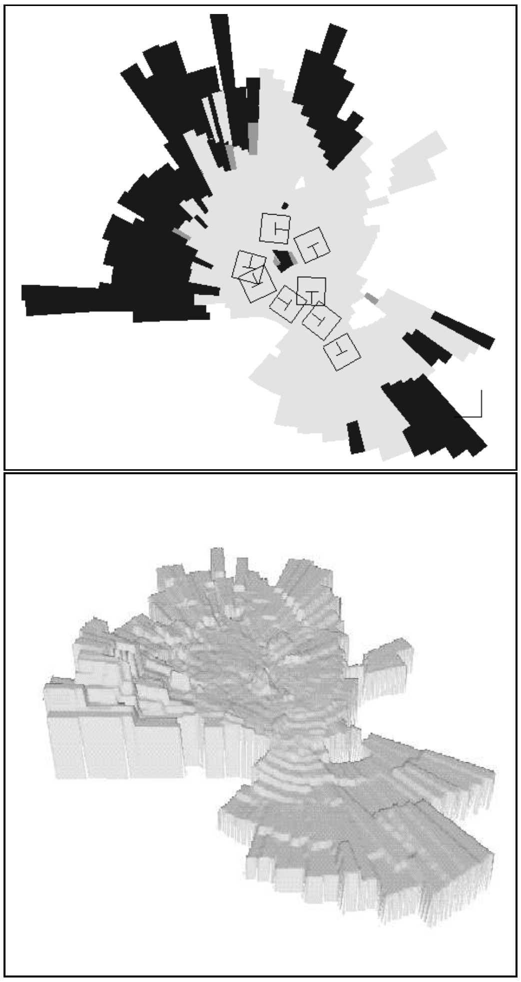

Figure 5: *Fusion of 8 different classified laser images: terrain classes (top) and elevation (bottom)*

5 Navigation planning

Each time 3D data are acquired, classified and fused in the global model, the robot has to answer the following questions:

- Where to go? (sub-goal selection)
- How to go there? (motion mode selection)
- Where to perceive? (data acquisition control)
- What to do with the acquired data? (perception task selection)

For that purpose, the navigation planner reasons on the robot capabilities (action models for perception and motion tasks) and the global terrain representation.

5.1 Planning motion versus planning perception

A straightforward fact is that motion and perception tasks are strongly interdependent: planning and executing a motion requires to have formerly modelled the environment, and to acquire some specific data, a motion

Figure 6: *The model of figure 5 after obstacle growing (top) and the nodes defined by the region segmentation (bottom)*

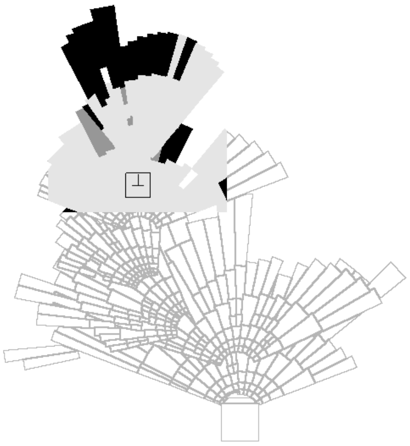

Figure 7: *Robot surroundings explicited as a bitmap*

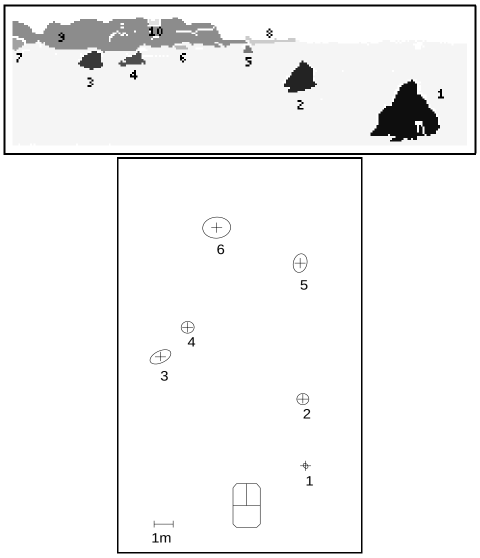

Figure 8: **Top***: Segmented objects.* **Bottom***: Selected landmarks and their uncertainty according to sensor noise and resolution and object shape.*

is often necessary to reach the adequate observation position.

Planning motion tasks in an environment modelled as a connection graph is quite straightforward: it consists in finding paths in the graph that minimise some criteria (time and energy, that are respectively related to the terrain classes and elevation variations). This is easily solved by classical graph search techniques, using cost functions that express these criteria.

Planning perception tasks is a much more difficult issue: one must be able to predict the results of such tasks (which requires a model of the perception processes), and the *utility* of these results to the mission:

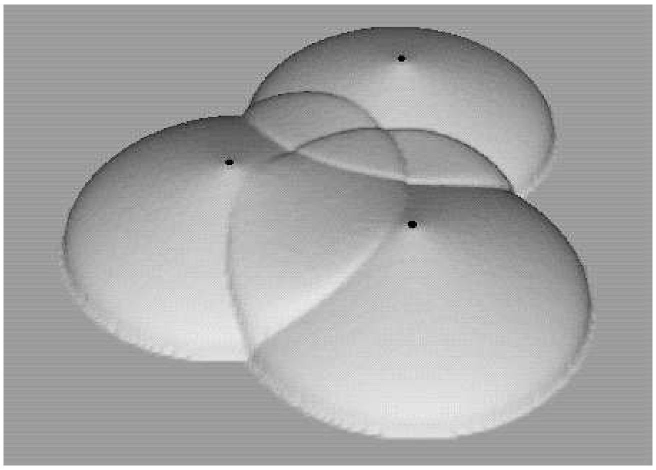

Figure 9: *A simple localisation model: the gain on the robot position precision increases as the distance of the perceived landmarks decreases, and is better as the number of perceivable landmarks increases. This figure presents this gain in an environment with three landmarks and no occlusions*

- Localization processes can be modelled by a simple function that expresses the gain on the robot position accuracy, as a function of the number and distances of perceivable landmarks - assuming all landmarks are intrinsically equally informative (figure 9);

- With the confidence model of the 3D data classification process, one can predict the amount of information a classification task can provide. But it is much more difficult to express the utility of a classification task to reach the goal: the model of the classification task cannot predict *what* will be effectively perceived. It is then difficult to estimate the interest of these tasks (figure 10).

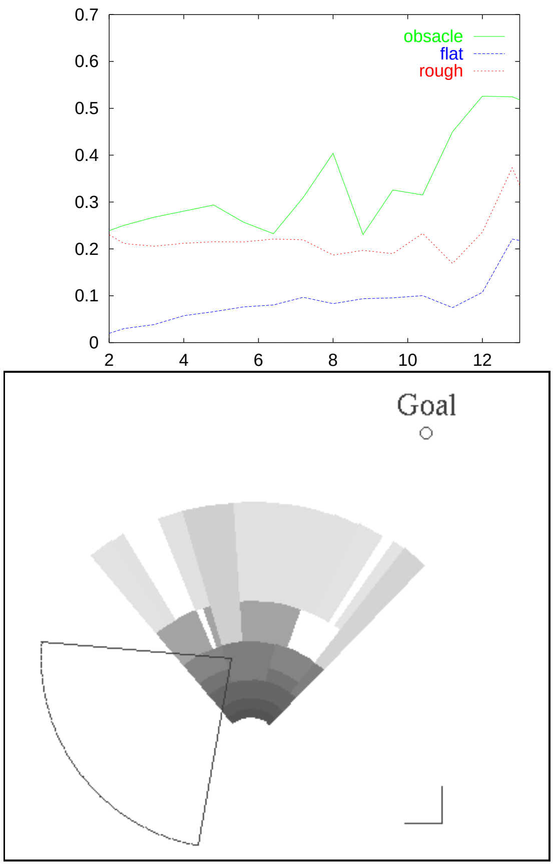

Figure 10: *The confidence model of the classification procedure (top) and the labelling confidence of the terrain model (represented as grey levels in the bottom image) allow to determine the classification task that maximises the gain in confidence from a given view point. But the result of this task is of a poor interest to reach the goal*

5.2 Approach

A direct and brute force approach to answer the former questions would be to perform a search in the connection graph, in which *all* the possible perception tasks would be predicted and evaluated at *each* node encountered during the search. Besides its drastic algorithmic complexity, this approach appeared unrealistic because of the difficulty to express the utility of a predicted classification task to reach the goal.

We therefore choose a different approach to tackle the problem: the perception task selection is *subordinated* to the motion task. A search algorithm provides an *optimal* path, that is analyzed afterwards to deduce the perceptions tasks to perform along it. The "optimality" criterion takes here a crucial importance: it is a linear combination of time and energy consumed, weighted by the terrain class to cross and the confidence of the terrain labelling (figure 11).

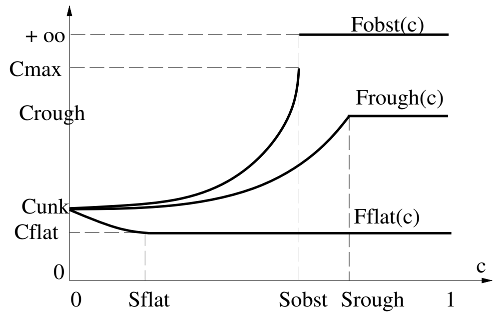

Figure 11: *Weighting functions of an arc cost, as a function of the arc label and confidence*

Introducing the labelling confidence in the crossing cost of an arc comes to consider *implicitly* the modelling capabilities of the robot: tolerating to cross obstacle areas labelled with a low confidence means that the robot is able to acquire easily informations on this area. Off course, the returned path is not executed directly, it is analysed according the following procedure:

1. The sub-goal to reach is the last node of the path that lies in a crossable area;

2. The labels of the regions crossed to reach this sub-goal determine the motion modes to apply;

3. And finally the rest of the path that reaches the global goal determines the aiming angle of the sensor.

Controlling localization: the introduction of the robot position uncertainty in the cost function allows to plan localization tasks along the path. The cost to minimise is the integral of the robot position accuracy as a function of the cost expressed in terms of time and energy (figure 12)

6 Results and discussion

The terrain modelling procedures and navigation planning algorithm have been intensively tested with the mobile robot Adam[1]. We performed experiments on

[1]ADAM is property of Framatome and Matra Marconi Space currently lent to LAAS

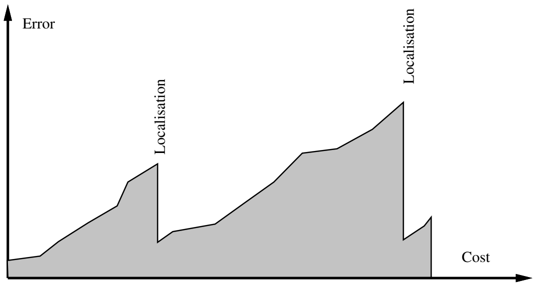

Figure 12: *Surface to minimise to control localisation tasks*

Figure 13: *ADAM in the Geroms test site*

the Geroms test site in the French space agency CNES, where Adam achieved several ``Go To [goal]'' missions, travelling over 80 meters, avoiding obstacles and getting out of dead-ends (for more details concerning Adam and the experimental setup, refer to [13]).

Figure 14 presents two typical behaviours of the navigation algorithm in a dead-end, and figure 15 shows the trajectory followed by the robot to avoid this dead-end, on the terrain model built after 10 data acquisitions.

The navigation planner proved its efficiency on most of our experiments. The adaptation of the perception and motion tasks to the terrain and the situation enabled the robot to achieve its navigation task efficiently. By possessing several representations and planning functions, the robot was able to take the adequate decisions. However, some problems raised when the planned classification task did not bring any new information: this happened in some very particular cases where the laser range finder could not return any measure, because of a very small incidence angle with the terrain. In these cases, the terrain model is not modified by the new perception, and the navigation planner re-planned the same perception task. This shows clearly the need for an explicit sensor model to plan a relevant perception task. And this generalizes to all the actions of the robot: the robot control system should possess a model of the motion or perception actions in order to select them adequately.

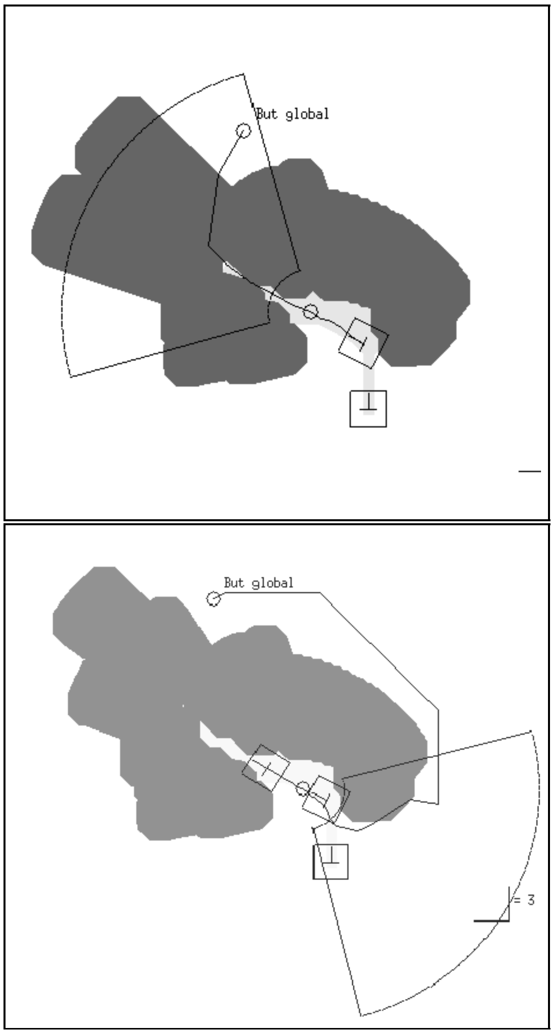

Figure 14: *The navigation planner explores a dead-end: it first tries to go through the bottom of the dead-end, which is modelled as an obstacle region, but with a low confidence level (top); after having perceived this region and confirmed that is must be labelled as obstacle, the planner decides to go back (bottom)*

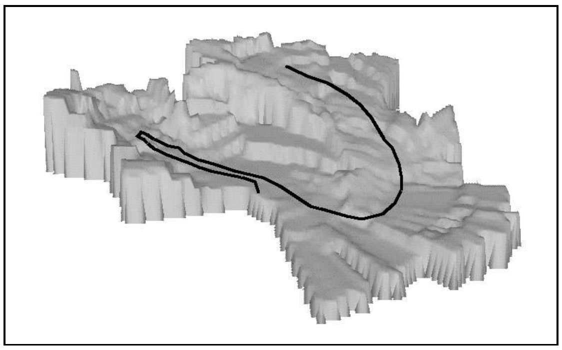

Figure 15: *A trajectory that avoids a dead-end (80 meters - 10 perceptions)*

References

[1] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer. Toward autonomous driving : the cmu navlab. part i : Perception. *IEEE Expert*, 6(4), August 1991.

[2] C.R. Weisbin, M. Montenerlo, and W. Whittaker. Evolving directions in nasa's planetary rover requirements end technology. In *Missions, Technologies and Design of Planetary Mobile Vehicules. Centre National d'Etudes Spatiales, France*, Sept 1992.

[3] B. Wilcox and D. Gennery. A Mars Rover for the 1990's. *Journal of the British Interplanetary Society*, 40:484–488, 1987.

[4] E. Krotkov and R. Hoffman. Terrain Mapping for a Walking Planetary Rover. *IEEE Trans. on Robotics and Automation, 10 (6)*, pp: 728-739, Dec. 1994.

[5] E. Krotkov and M. Hebert. Mapping and Positioning for a Prototype Lunar Rover. In *IEEE International Conference on Robotics and Automation, Nagoya, japan*, May 1995.

[6] M. Hebert. Pixel-based range processing for autonomous driving. In *IEEE International Conference on Robotics and Automation, San Diego, California*, 1994.

[7] G. Giralt, R. Chatila, and R. Alami. Remote intervention, robot autonomy and teleprogramming : generic concepts and real-world application cases. In *IEEE International Conference on Intelligent Robots and Systems, Yokohama (Japan)*, July 1993.

[8] S. Lacroix, R. Chatila, S. Fleury, M. Herrb, and T. Simeon. Autonomous navigation in outdoor environment : Adaptative approach and experiment. In *IEEE International Conference on Robotics and Automation, San Diego, California*, 1994.

[9] T. Simeon and B. Dacre-Wright. A practical motion planner for all-terrain mobile robots. In *IEEE International Conference on Intelligent Robots and Systems, Yokohama (Japan)*, July 1993.

[10] F. Nashashibi, P. Fillatreau, B. Dacre-Wright, and T. Simeon. 3d autonomous navigation in a natural environment. In *IEEE International Conference on Robotics and Automation, San Diego (USA)*, May 1994.

[11] S. Betgé-Brezetz, R. Chatila, and M. Devy. Natural scene understanding for mobile robot navigation. In *IEEE International Conference on Robotics and Automation, San Diego (USA)*, May 1994.

[12] P. Fillatreau, M. Devy, and R. Prajoux. Modelling of unstructured terrain and feature extraction using b-spline surfaces. In *International Conference on Advanced Robotics, Tokyo (Japan)*, November 1993.

[13] R. Chatila, S. Fleury, M. Herrb, S. Lacroix, and C. Proust. Autonomous navigation in natural environment. In *Third International Symposium on Experimental Robotics, Kyoto (Japan)*, October 1993.