Motion and Perception Strategies for Outdoor Mobile Robot Navigation in Unknown Environments

Simon Lacroix* and Raja Chatila
LAAS - CNRS
7, ave du Colonel Roche 31077 Toulouse Cedex - France
E-mail: {simon, raja}@laas.fr

Abstract

This paper presents an experimented approach to autonomous robot navigation in an unknown natural environment. The approach involves several levels of reasoning, several environment representations, and three different motion modes. We focus on the "navigation level" of the whole system, which is in charge of reaching a distant goal by selecting sub-goals to reach, motion modes to apply, and perception tasks to execute for this purpose. We present how a terrain model dedicated to the navigation process is built on the 3D data acquired by the robot, and we describe an approach to tackle the difficult problem of planning perception and motion tasks. Experimental results on a realistic test site are presented and discussed.

1. Introduction

The problem of long range navigation in unknown outdoors environments is not very frequently addressed. Systems that were actually experimented were demonstrated for road following [1], or motion in rather limited environment conditions [2, 3]. Two important achievements in totally unstructured environments were Ambler [4] and the navigation of the UGV [5].

A canonical task a robot should be able to achieve in such an environment is the `''Go To [goal]''` task (or *navigation task*), where `goal` is a distant point to reach autonomously : any more complex robotic mission (exploration, sample collecting...) will include one or more instances of this task. Our approach to achieve this navigation task in an unknown natural environment involves 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 sub-goal to reach, and which motion mode to apply. The paper is especially devoted to this level, which is a key component of the system, since it controls the perception and motion activities of the robot.

The paper is organised along the following outline : the following section presents more precisely our general adaptive and hierarchical approach to autonomous navigation in unknown outdoor environments, pointing out the importance of the navi-

*Currently on a post-doc funded by INRIA at the Centre for Intelligent Machines, McGill University, Montréal (simon@cim.mcgill.ca) 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 4. Experimental results conclude the paper.

2. A General Strategy for Navigation in Outdoor Unknown Environments

For any robotic task in general, and for the navigation task in particular, we favour an approach where the robot explicitly reasons on environment representations and its capabilities to decide the actions to perform [6, 7]. As opposed to a behaviour-based approach, we are convinced that such an approach is mandatory to develop machines that can be *controlled* and *programmed* by a remote human operator.

2.1. An adaptive approach

According to a general "economy of means" principle due to limitations of on-board processing capacities, memory and energy that we put on the system (realistic constraints for several applications were the robot has to be operational on a remote site, *e.g.* for planetary exploration of scientific missions in hostile places such as the Antarctic), and to achieve a time-efficient behaviour, we choose an *adaptive* approach in which the robot adapts its behaviour to the nature of the terrain [8, 7]. Hence, three motion modes are considered :

- A **reflex** motion mode : on large flat and lightly cluttered zones, the robot locomotion commands are determined 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** motion mode : when the environment is mainly flat, but cluttered with obstacles, the robots locomotion commands are determined by a trajectory plan. The trajectory planner reasons on a binary description of the environment, which is described in terms of *Crossable/Non-Crossable* areas.
- A **3D planned** motion mode : when the environment is highly constrained (uneven terrain), 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 (a numerical terrain model - NTM [10]) ;

Choosing to endow the robot with three different motion modes enables "smart" and efficient behaviours, but complicates quite a lot the system : it must be able to deal with several different terrain representations and planning processes. It must especially have the ability to determine which motion mode to apply, and therefore which terrain representation to update : this is performed thanks to a specific planning level, the *navigation planner*.

2.2. The navigation planner

We assume that the terrain on which the robot must fulfill a navigation task is initially unknown, or mapped with a very low resolution. It is then only possible robot has to move autonomously. In this context, the navigation task ``Go To'' is achieved thanks to three layers of planning (figure 1) :

- The *route planner* chooses long-term paths to the goal on the basis of the initial informations (the route graph that covers the whole area in which the mission takes place). The route planner selects a sub-goal for the navigation planning level ;

- The *navigation planner* (or *path planner*) reasons on a global qualitative representation of the terrain (the *region map*), built from the data acquired by the robot's sensors. It selects *(i)* the next perception task to perform, *(ii)* the sub-goal to reach and *(iii)* the motion mode to apply (which comes to select and control the trajectory planners) ;

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

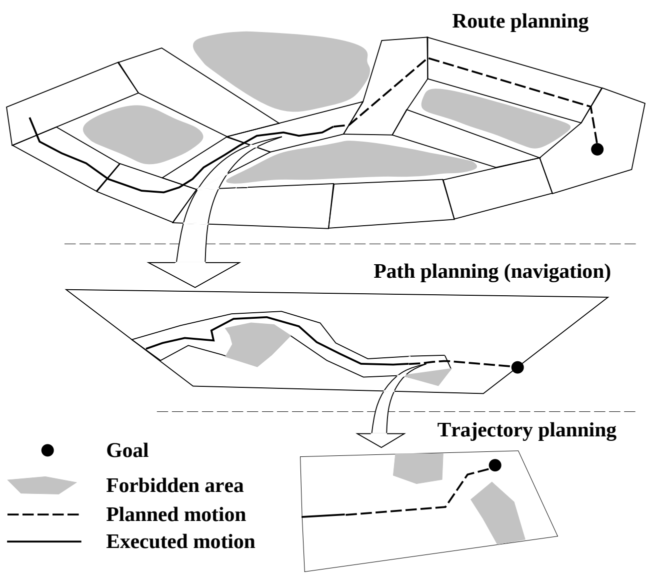

Figure 1. *Hierarchical organisation of the the three levels of planning involved in our approach, and corresponding terrain representations : the route planner reasons on the initial coarse environment model, that can cover several kilometres. The navigation planner reasons on a qualitative terrain representation built from the robot sensor's data (the* region map, *that can cover several hundreds of meters) ; and finally the trajectory planners reason on a local and precise description of the environment.*

Splitting the decisional processes into three layer of planning has the advantage to structure 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 analyse and solve failing cases : when a planner fails to reach its goal, it means that the environment representation of the upper layer is erroneous and therefore that it has to be revised.

The navigation planner is *systematically* activated at each step of the incremental execution of the task : each time 3D data are acquired, they are analysed to provide a description of the perceived zone in terms of navigation classes, and this description is fused to update the *region map*. The introduction of the navigation planning layer defines a particular instance of the usual "perception-decision-action" loop, where the "decision" part is split into two distinct processes : navigation and trajectory planning.

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 *multi-layered* and *heterogeneous*. Several perception processes coexist then in the system, each dedicated to the extraction of specific representations : perception is *multi-purpose*.

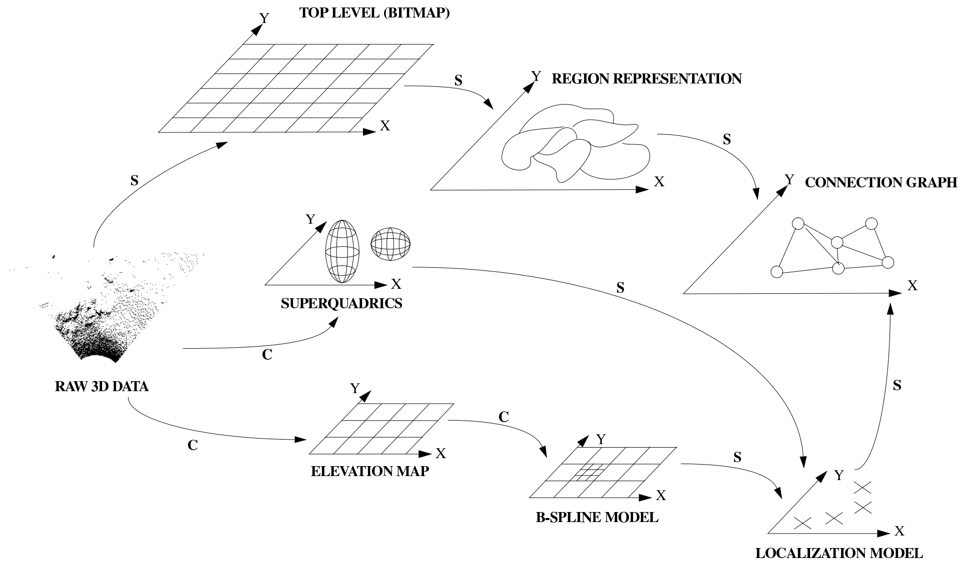

Figure 2. *The various representations used in the system : one can distinguish the numerical terrain model [10] necessary to the 3D trajectory planner, the region map (and connection graph) dedicated to the navigation planner, and three different ways to build a localisation model :*(i) *by modelling objects (rocks) with superquadrics [11],* (ii) *by detecting interesting zones in the NTM thanks to a B-spline based model [12], or* (iii) *by detecting poles in the 3D raw data*

Figure 2 presents the various terrain representations required during navigation : arrows represent the constructive dependencies between them. Coherence relationships between these various representations are to be maintained when necessary, which remains an open issue.

3. Building the region map

3.1. 3D data classification

Applied each time 3D data are acquired (either by a laser range finder or a correlation stereo-vision algorithm), the classification process produces a description of the locally perceived area in term in *terrain classes* [13]. It relies on a specific discretization of the perceived area that respects the sensor resolution (figure 3), that defines "cells" on which different characteristics (attributes) are determined : density (number of points contained in a cell compared with a nominal density defined vector and corresponding variances...

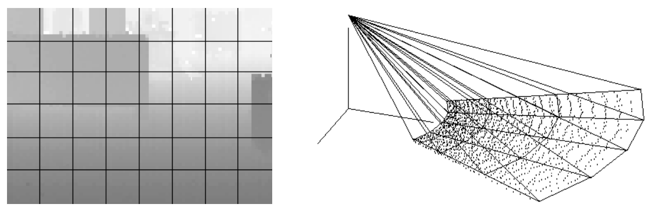

Figure 3. *Discretization used for the classification procedure : a regular discretization in the sensor frame (left : a 3D image is represented as a video image, where the gray levels corresponds to the points depth) defines a discretization of the perceived zone that respects the sensor resolution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

A non-parametric Bayesian classification procedure is used to label each cell : a learning phase based on prototypes classified by a human lead to the determination of probability density functions, and the classical Bayesian approach is applied, which provides an estimate of the partial probability for each possible label. A decision function that privileges 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.

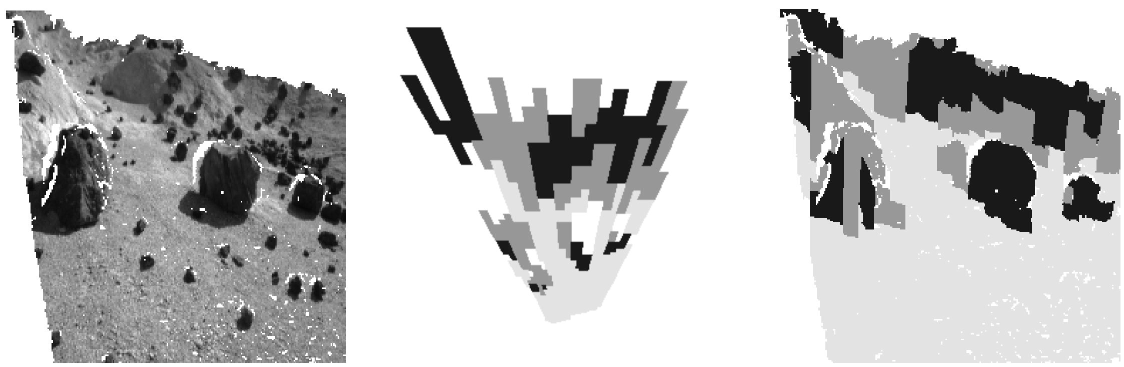

Figure 4. *Classification of a correlated stereo image : correlated pixels (left) ; classification of the perceived zone (center), and reprojection of the result in the camera frame (right). From clear to dark : unknown, flat, uneven and obstacle)*

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 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 their elevation allow to perform a fusion procedure of several classified images, provided the robot position is known (figure 5). 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 structuration 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

Figure 5. *The model structuration procedure. From left to right : a terrain model resulting from the fusion of 8 classified laser images ; the same model after constrained zones growing, and the nodes of the connection graph defined by the region segmentation*

constraints on the robot : we simply consider a robot point model, and therefore perform a constrained zones 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 5). The terrain is described as a bitmap only in the surroundings of the robot's current position, whereas the region model (much more compact) is kept in memory during the whole mission.

4. Navigation planning

Each time 3D data are acquired, classified and fused in the global model, the robot has to answer autonomously 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 connection graph defined by the region map.

4.1. Planning motion versus planning perception

A straightforward fact is that motion and perception tasks are strongly interdependent : executing a motion requires to have formerly modelled the environment, and to acquire some specific data, a motion is often necessary to go the adequate observation position. Planning motion tasks in an environment modelled as a connection graph is easily solved by classical search techniques that find optimal paths minimising some criteria (in our case time and energy, that are respectively related to the terrain classes and elevation variations).

To plan perception tasks, 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 to achieve. If the utility of a localisation task can be modelled by a simple function that expresses the gain on the the robot position precision as a function of the number and distance of perceivable landmarks, estimating the utility of a modelling task is a much more difficult issue (figure 6).

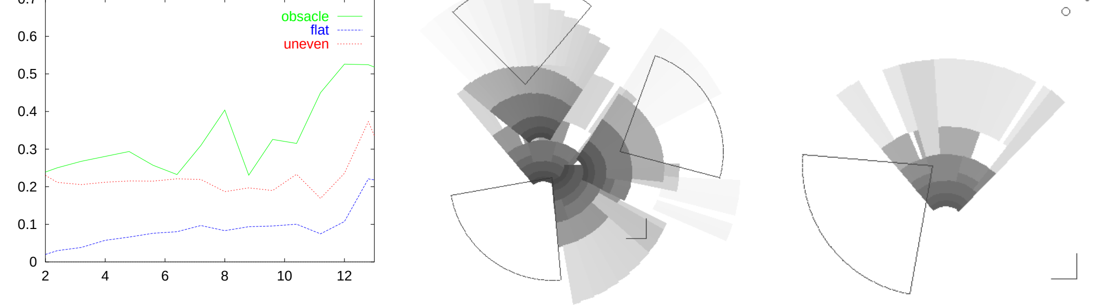

Figure 6. *The confidence model of the classification procedure (left : mean cell entropy as a function of the perceived distance) and the labelling confidence of the terrain model (represented as grey levels in the images) allow to determine the classification task that maximises the gain in confidence from any given position in the model (center). But the result of such a task may be of a poor reach interest to reach a specified goal (right). In such a case, one should be able to determine* **where** *a better information is required* **in order to** *reach the goal*

4.2. 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 analysed afterwards to deduce the perceptions tasks to perform. 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 7).

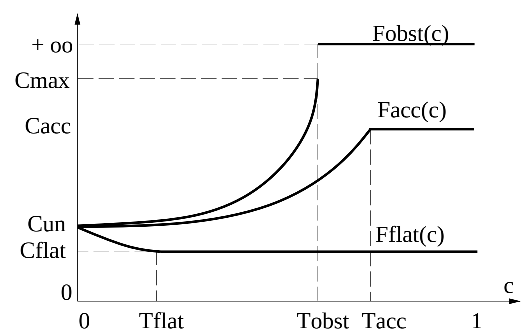

Figure 7. *Weighting functions of an arc cost, as a function of the region label confidence. An experimental statistical analysis helped to define thresholds on the confidence labelling over which the classification result can be considered faithful (thresholds T_{flat}, T_{acc} and T_{obst} respectively for regions labelled flat, uneven and obtacle). When the region confidence is over these thresholds, their crossing cost is equal to a* nominal *cost (C_{flat}, C_{acc} and $C_{obst} = \infty$) ; when the region confidence is under these thresholds, the costs decreases to reach the unknown region cost C_{un} when $c = 0$.*

The influence of these cost weighting functions is illustrated on figure 8. They come 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 ;

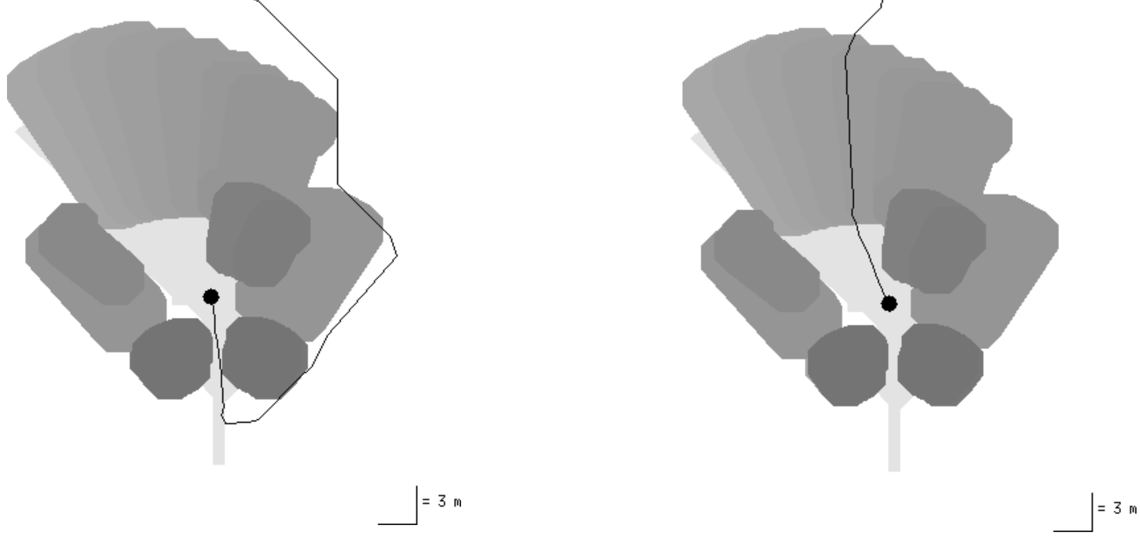

Figure 8. *Influence of the cost weighting functions (the gray levels represent the confidence of the region labelled "obstacle"). When not considering this confidence, the optimal path returns backward to reach the goal (left). When the cost weighting functions are applied, the optimal path crosses an obstacle area labelled with a low confidence (right)*

2. The label of the regions crossed to reach this sub-goal determines the motion mode to apply ;

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

One can interpret the result of this analysis as a way to answer the question asked in figure 6 : the cost weighting functions help to find the interesting regions to model in order to reach the goal.

Figure 9 presents a way to determine paths along which are planned localisation tasks. This approach has only been tested in simulation experiments, but seems to be helpful to plan safe paths.

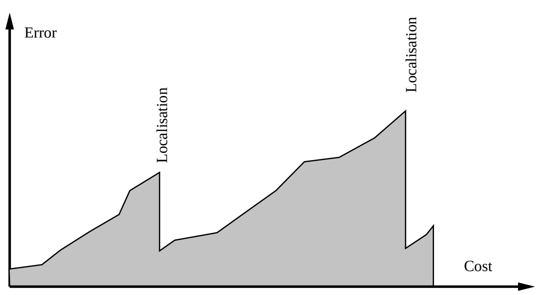

Figure 9. *Introduction of the uncertainty on the robot position in the cost to control localisation tasks. Provided a model of the evolution of the robot position uncertainty as it moves and a model of the localisation task, minimising the integral of the robot position precision as a function of the cost expressed in term of time and energy determines paths that gets closer to landmarks. The choice of minimising an integral (as opposed to a linear combination for instance) comes to defines motions with a big uncertainty on the position as more "risky" than motions with a low uncertainty*

5. 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 the

[1] Advanced Demonstrator for Autonomy and Mobility, is property of Framatome and Matra Marconi Space and is currently lent to LAAS

Figure 10. *ADAM in the Geroms test site*

Geroms test site in the French space agency CNES[2], 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 [8]).

Figure 11 presents two typical behaviours of the navigation algorithm in a dead-end, and figure 12 shows the trajectory followed by the robot to avoid this dead-end, on the elevation model built after 10 data acquisitions.

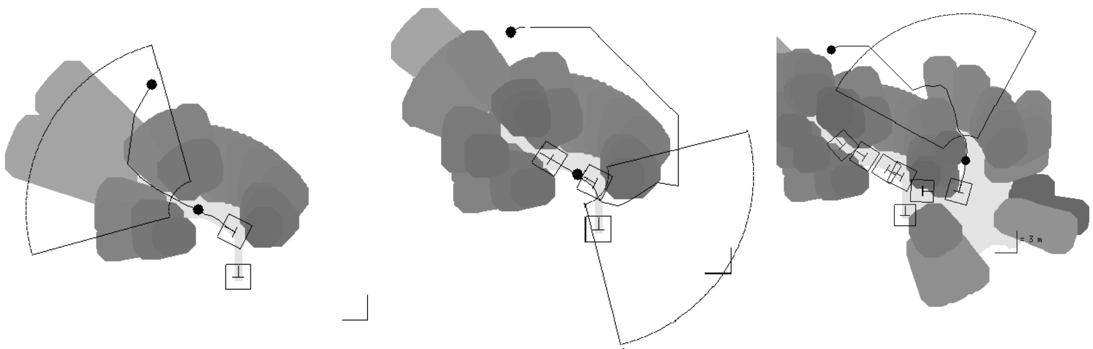

Figure 11. *The navigation planner explores a dead-end : it first tries to go through the bottom of the dead-end, which is modelled as an obstacle region, but with a low confidence level (left) ; after having perceived this region and confirmed that is must be labelled as obstacle, the planner decides to go back (center), and finally finds a way that reaches the goal(right)*

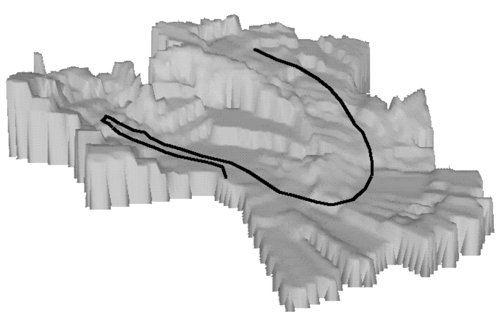

Figure 12. *Elevations encoded in the terrain model built during the dead-end exploration, and trajectory executed to reach the global goal. 80 meters have been travelled, and 10 perceptions have been activated.*

These experiment have proved the possibility to build and maintain quickly

[2]Centre National d'Études Spatiales

navigation algorithm proposed proved its efficiency on most of our experiments. It includes the sub-goal and next perception task determination, two issues that should never be considered indenpedently.

The problem of navigation as we understand it nevertheless requires some more attention. In particular, we are considering the framework of the decision theory to express the utility of a classification task. By considering more explicitly the sensor model (detectability and uncertainty models), we hope to have more optimal decisions.

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, M. Hebert, M. Buffa, F. Cozman, and L. Robert. Stereo friving and position estimation for autonomous planetary rovers. In *IARP 2nd Workshop on Robotics in Space, Montreal, Canada*, 1994.

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

[6] 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.

[7] 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.

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

[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. Betge-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] S. Lacroix, P. Phillatreau, F. Nashashibi, R. Chatila, and M. Devy. Perception for autonomous navigation in a natural environment. In *Workshop on Computer Vision for Space Applications, Antibes, France*, Sept. 1993.