AIAA-94-1201-CP

PERCEPTION SYSTEM AND FUNCTIONS FOR AUTONOMOUS NAVIGATION IN A NATURAL ENVIRONMENT

N94-30552

Raja Chatila, Michel Devy, Simon Lacroix, Matthieu Herrb
LAAS-CNRS
7, avenue du Colonel-Roche
31077 Toulouse Cedex - France
E-Mail : (raja|michel|simon|matthieu@laas.fr)

207649
6047
p. 11

Abstract

This paper presents the approach, algorithms and processes we developed for the perception system of a cross-country autonomous robot. After a presentation of the tele-programming context we favor for intervention robots, we introduce an adaptive navigation approach, well suited for the characteristics of complex natural environments. This approach lead us to develop an heterogeneous perception system that manages several different terrain representations. The perception functionalities required during navigation are listed, along with the corresponding representations we consider. The main perception processes we developed are presented. They are integrated within an on-board control architecture we developed. First results of an ambitious experiment currently lead at LAAS are then presented.

1 Context - Introduction

A large amount of results exists today on mobile robot navigation, most of them related to indoor environments. As for outdoor navigation, most of the works concern environments wherein obstacles are rather structured, and the terrain mostly flat (*e.g.* road following [1]). More recently, studies considering autonomous mobility in *natural unstructured* outdoor environments comes out [2] : several applications are considered, such as public safety [3] (fire fighting, chemical disaster...), sub-sea intervention or exploration, and planetary exploration [4, 5].

Several aspects make these kinds of interventions a demanding and difficult problem for robotics :

- The robot has to operate in a natural, unstructured, maybe hostile and a priori unknown environment ;
- There might be interaction discontinuities with the robot because of communication breakdowns, important delays or low bandwidth ;
- The information on the robot and the environment is mostly acquired through the robot's own sensors.

These constraints rule out direct teleoperation as well as telerobotics approaches, and point towards robots with **important autonomous capacities** : the environment being poorly known and the communication possibilities very poor, the mission can only be predefined at a *task-level* in general, not in its every details. The robot must then build and maintain its own representations of the environment, upon which it autonomously reasons and plans the actions to perform in order to fulfill the mission.

As opposed to behavior-based control schemes [6], we favor the development of a global architecture with two main parts to tackle this challenge [7, 2] : an operating station for mission programming and supervision, and a remote robot system[1] able to *interpret* the mission and *execute* it autonomously.

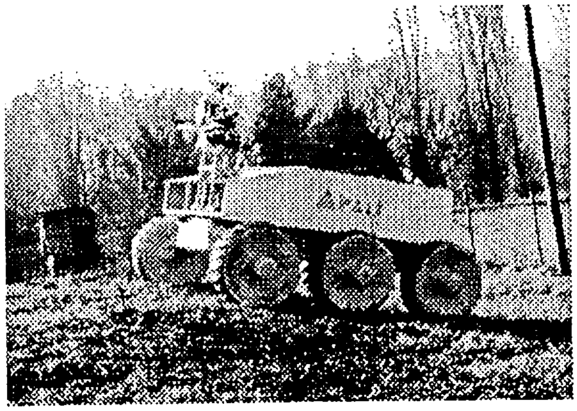

Figure 1: **The mobile robot ADAM in its environment**

The operating station includes the necessary functions that allow a human to (i) build an *executable robotic*

[1] not necessarily a single one.

mission that can be interpreted and executed by the robot, (as opposed to a higher level description of objectives) ; and to **(ii)** supervise its execution, taking into account the delays and communication constraints. Its presence essentially ensues from the following considerations :

- The mission is not defined once and for all : according to returned data, one should be able to change the objectives of the mission (when unexpected events occur for instance) or to decide the execution of a particular action (such as "pick this sample" in the case of a scientific exploration).
- The robot could fall into difficult situations wherein its own capacities are insufficient, a human intervention would then be necessary for troubleshooting.

As for the robot, its autonomy essentially relies on its ability to build faithful representations of its environment, which is obviously necessary for him to interpret the mission and decompose it into executable tasks, considering its actual context.

We focus in this paper on the development and organization of the perception functionalities an autonomous cross-country robot must be embedded with. The following section introduces the general adaptive approach we chose to tackle with outdoor environment navigation, that emphasizes the need to develop several perception processes. Section 3 presents the different perception functionalities required during navigations, and the corresponding terrain representations maintained by the robot. The processes we developed to build these representations are presented in section 4, and the way they are controlled and integrated within the context of our robot architecture is presented in section 5. We finally describe the first results of the EDEN experiment, currently developed at LAAS with the mobile robot ADAM[2] (figure 1).

2 A Multi-Purpose Perception System for Adaptive Navigation

The complexity of outdoor natural environments comes essentially from their diversity and lack of structure : some areas can be totally flat (maybe cluttered with easily detectable obstacles - big rocks lying on a prairie for instance), whereas others area can be much more cluttered, such as a landscape of smooth hills (sand dunes) or an uneven rocky area. This variety induces several different behaviors, and constrains both the perception and motion planning processes.

According to a general economy of means principle (on-board processing capacities, memory and time are always limited), we favor an *adaptive* approach [8, 9] : we aim at adapting the robot behavior of the robot to the nature of the terrain, and hence three navigation modes are considered :

- And a **reflex** navigation mode : on large flat and lightly cluttered zones, the robot locomotion commands are determined on the basis of *(i)* a goal and *(ii)* the information provided by "obstacle detector" sensors.
- A **2D planned** navigation mode : it relies on the execution of a planned 2D trajectory, using a binary description of the environment in terms of *Crossable/Non-Crossable* areas.
- A **3D planned** navigation mode : this mode requires a precise model of the terrain, on which a fine 3D trajectory is planned and executed.

Each of these navigation mode is suitable for a particular terrain configuration, and requires a specific representation. Besides this trajectory planning functionalities, there are some other important processes that also require a representation of the terrain : exteroceptive localization, often required to refine or correct the estimation of the robot position provided by its internal sensors ; and *navigation planning*, which is in charge of intermediate goal and navigation mode selection.

Several authors emphasized on the development of perception and motion planning processes able to deal with any terrain configuration [10, 11], trying to recover as much information as possible from the acquired 3D data. Besides the processing complexity, such an approach has a main drawback : it does not takes advantage of the variety of the environment. Although sometimes needed, the recovery of a complete and accurate 3D geometrical model may be often not necessary : more simple and approximative representations will be sufficient in many situations, when the terrain is mostly flat for instance.

We believe that aiming at building such a "universal" terrain model is extremely difficult and not efficient, and we therefore chose to endow the robot with a *multi-level* terrain modeling capacity : a particular representation is built or updated only when required by a given task. This involves the development of various perception processes, each of them being dedicated to the extraction of specific representations (*multi-purpose perception*).

At each step of the incremental execution of its mission, the navigation planner autonomously chooses an intermediate goal, along with the navigation mode to apply to reach it. This induces the choice of the representations it must update, which comes to answering these questions : which sensor to use ? With what operating modalities ? How should the data be processed ? Perception planning becomes in our case a

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

key component to enhance the robot autonomy and efficiency.

To achieve this, we propose to build and update systematically a *global qualitative* description of the environment on which all "strategic" decisions are taken. This representation is built thanks to a fast analysis of the raw 3D data acquired (either by a Laser Range Finder - LRF - or by a stereovision correlation algorithm), that provides a terrain description in term of navigation classes, and some other *qualitative* informations, such as the possible presence of a landmark, the mean altitude and slope of some areas... Each time this representation is updated, it is structured in order to produce a semantically significant model, from which navigation and perception plans are deduced.

3 Terrain Representations

After a brief presentation of the perception functionalities and the constraints brought by outdoor environments, we introduce in this section a *multi-level* environment model, that defines the relations between the various representations.

3.1 Outdoor Representations : characteristics and constraints

The difficulty of representing outdoor environments comes essentially from the fact that they are not intrinsically structured, as compared to indoor environments where simple geometric primitives match the reality. As a consequence, any representation based on geometric primitives (linear or second degree surfaces, super-quadrics...) is difficult to build and to maintain, and introduces an approximation of the reality via artificial structures. We therefore favored the development of simpler representations (polygonal maps, elevation maps...), easier to build and manage. Semantic informations are not explicitly contained in such representations, but can anyhow easily be extracted.

The other characteristics of the representations are related to the robot sensors and mission :

- The sensors are always imperfect : their data are incomplete (lack of information concerning existing features) and not precise. They generate artifacts (information on non-existing features) and errors (wrong information concerning existing features). The same area when perceived again can therefore be differently represented. Hence environment representations must tolerate important **variations** [12].
- The environment is initially unknown (or very poorly known) and is **incrementally discovered** : the robot must be able to manage local momentary representations, and merge them in global descriptions of the world. We are convinced that global representations are required [13], especially to recover from deadlocks that often appears when dealing only with local representations.

Finally, one must not forget that the system memory is limited, and so the representations must be as compact as possible.

3.2 Perception Functionalities and Corresponding Representations

3.2.1 Trajectory Planning

From the poorest to the richest, here are the representations required by the three navigation modes we retained :

- **Reflex Navigation** : The robot locomotion commands are determined on the basis of *(i)* a target value (heading or position) and *(ii)* the information provided by "obstacle detector" sensors. An obstacle avoidance procedure enables the robot to move safely, and the area to cross is essentially obstacle-free, so that there are poor chances that the robot fall into deadlocks. Strictly speaking, this mode does not requires any modeling of the terrain, but a description (a simple 2D polygon in our case) of a zone where it can be applied.
- **2D planned navigation** : This mode is applied on lightly cluttered environments, that can be represented by a binary description in term of *Crossable / Non-Crossable* areas. The crossable zones are the places where the robot attitude is not constrained, *ie.* where the terrain is mostly flat, or has an admissible slope for the robot to run safely, whatever its heading position is. A trajectory defined by a sequence of 2D positions is planned within the crossable areas. In our case, the 2D planner requires a binary *bitmap* description, on which a distance propagation method (similar to those presented in [14]) produces a Voronoi diagram.
- **3D planned navigation** : On uneven or highly cluttered areas, the "obstacle" notion is closely linked with the constraints on the robot attitude, and therefore constrains the robot heading position. Planning a trajectory on such areas is a much more difficult task [15] that requires a detailed modeling of the terrain. In our case, the 3D planner builds its own data structure on the basis of an elevation map, computed on a regular Cartesian grid (section 4.4).

3.2.2 Localization

The internal localization sensors of the robot (odometry, inclinometers, inertial platform...) generate cumulative errors, especially on uneven or slippery areas. A localization procedure based on exteroceptive sensors is often necessary for both the robot and the supervising operator : to plan safe trajectories on formerly perceived areas for instance, the robot obviously needs to know precisely where it stands ; and a false position value may mislead the operator.

Such a localization procedure requires a specific *global* representation of the environment, be it a set of 3D points in the case of a correlation-based localization (iconic matching [16]), or a global map of detected landmarks (that must then be modeled, using particular geometric descriptions) in the case of a feature-based localization [17]. These two kinds of representations can be viewed as maps of interesting zones for the purpose of localization. In our case, we developed an original localization procedure (section 4.5), that requires a B-Spline based model of the terrain.

We are also currently investigating the modeling of unstructured objects (rocks, bushes...) thanks to complex geometric primitives (super-quadrics [18]) : such a model could be used to perform landmark detection, and might provide a "qualitative" localization functionality, sufficient in reflex navigation mode.

3.2.3 Navigation Planning

Navigation planning consists essentially in the determination of an intermediate goal, as well as the mode to activate to reach it, considering the mission's objective and the partial (and unprecise) knowledge the robot has on its environment. Several different constraints can be taken into account to perform this "route" planning, depending on the context : one may prefer execute safe trajectories from the localization point of view, or one may choose the fastest trajectories (time constraint), the shortest (energy constraint)... A semantic significant description of the perceived environment is here necessary. We have chosen a topological connection graph (section 5.2.2) : such a structure can contain very rich informations, and a theoretical formalism, often applied in the robotic community [19], is available for its exploitation.

3.2.4 Perception Planning

Perception planning, which is closely linked to navigation planning, requires a prediction ability : given a sensor and a point of view, what can be perceived ? To answer this question, the perceptual constraints of the sensor (occlusion, field of view, specularity) must be checked considering an environment numerical model.

3.3 A Structural Scheme

Several data structures that represent the same entities in the environment must coexist in the system. In this *multi-layered heterogeneous model*, the different representations are easily managed and a global consistency can be maintained. The relationships between the various representations explicit their building rules, and defines a constructive dependency graph between them. The figure 2 illustrates these relationships : each thin arrow represents a data processing algorithm, and the thick straight arrows corresponds to the production of a structure required to a trajectory planner. We distinguish two kinds of dependencies :

- **Systematic** dependencies : Every time a representation is updated, all the representations that systematically depends on it (arrows labeled "S") are updated. As one can see on the figure, every time 3D data are acquired, the global bitmap representation, the region representation and the connection graph are updated. Let's also note that when a localization model is available, the informations it contains are merged in the connection graph (section 5.2.2).
- **Controlled** dependencies (labeled "C") : The representations that are not always necessary are only built under control of the navigation planner. For instance, an elevation is only required to cross an uneven zone.

The top level of this heterogeneous model is a "bitmap" description of the environment, built upon the results of the fast terrain analysis algorithm. A lot of information is available in every pixel of this bitmap, such as the terrain label and its confidence level, the estimated elevation, the identification of the region it belongs to... We have chosen such a structure for the following reasons : it is simple, rich, adapted to the lack of geometrical structure of the environment and to the Digital Elevation Map description (section 4.4), and flexible, in the sense that any supplementary information can easily be encoded in a pixel without reconfiguring the entire description and the algorithms that use it. Moreover, the techniques that allow to extract structured informations (regions, connexity...) from a bitmap are well known and easily implemented.

3.4 Memory Management

The main drawback of maintaining global representations is memory occupancy, that rapidly becomes huge if they covers large areas, especially when using bitmap representations and elevation maps. To cope with this, we are currently developing a "forgetting" functionality : the area surrounding the robot, with a size limited by the sensor capacities, is fully described, whereas the remaining already perceived terrain is structured in a more compact way. The key point here is to determine the informations one must not forget : for the purpose of long range navigation, we consider that only the connection graph and the localization model are necessary to maintain.

We consider two different ways to implement this : the first one is to take advantage of the global bitmap region structuration, or of any other classical data compression method. The precise informations brought by the possibly computed elevation maps is then totally lost. The second way is to use the B-Spline based representation : the B-Spline representation would then be systematically built (in parallel with trajectory execution for instance). Only the B-Spline representation,

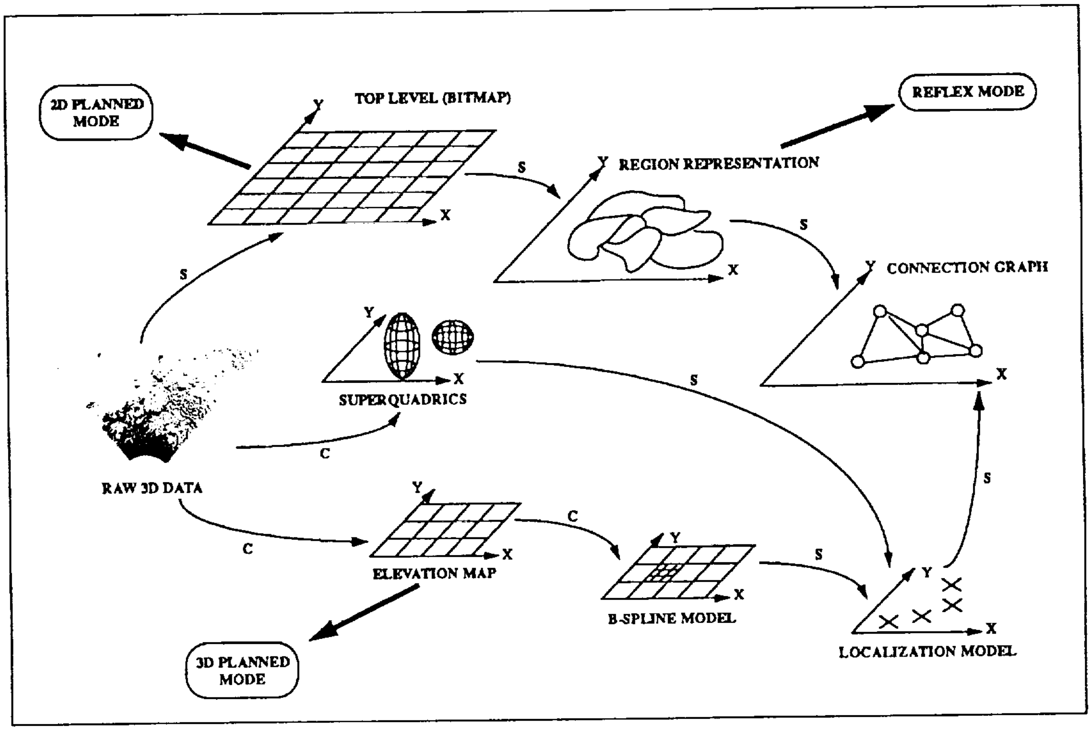

Figure 2: The representations used in the system

which is extremely compact, and that contains much more informations than the global bitmap representation, is kept in memory.

4 Building Representations

4.1 Fast Classification

Applied each time 3D data are acquired, this process produces a description of the perceived areas in term in *terrain classes*, along with some qualitative informations. It relies on a specific discretization of the perceived area in "cells", on which different characteristics that allow to label them are computed [9].

The discretization is the projection of a regular grid defined in the sensor frame (fig. 3). Its main characteristics are that it respects the sensor resolution, and that it points out a "density" attribute : the number of points of point contained in a cell, compared with a *nominal density* defined by the discretization rates, provides a useful information concerning the area covered by the cell : for instance, it is equal to the nominal density if the cell corresponds to a flat area. This information, along with other attributes concerning the cells (mean altitude, variance on the altitude , mean normal vector and corresponding variances) allows to heuristically label each cell as one of {*Flat, Slope, Uneven, Obstacle, Unknown*}.

This classification procedure, which complexity is $O(n)$, where n is the number of 3D points considered, takes around half a second on a Sparc-10 workstation to process a 10.000-points 3D image. It has proved its robustness on a large number of different images (fig. 4), produced either by the LRF or a stereovision correlation algorithm[3], and is especially weakly affected by the sensor noise (uncertainties and errors). An important point is that it is possible to estimate a confidence value on the labeling of each cell : this value generally decreases with the distance of the cell to the sensor, because of the decreasing accuracy on a 3D point coordinates with this distance. But this confidence also obviously depends on the label itself : for instance, a flat cell containing a few erroneous points can be labeled as an "uneven" one, whereas the probability that erroneous points perceived on an actu-

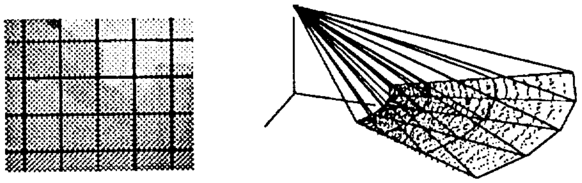

Figure 3: Discretization in the sensor frame, and projection on the ground

[3] The discretization then differs slightly from the one used for LRF images

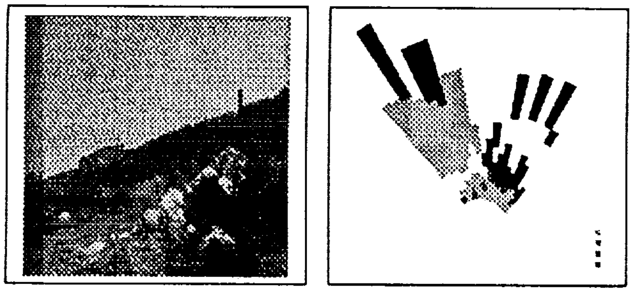

Figure 4: **Classification result on a complex scene. From clear to dark : Unknown, Flat, Slope, Uneven, Obstacle**

ally uneven zone lead to a "flat" label is very low. The quantitative estimations of this confidence value $P(error) = F(distance, label)$ are statistically determined, and constitute the useful model of the logical sensor "terrain classifier" (figure 5).

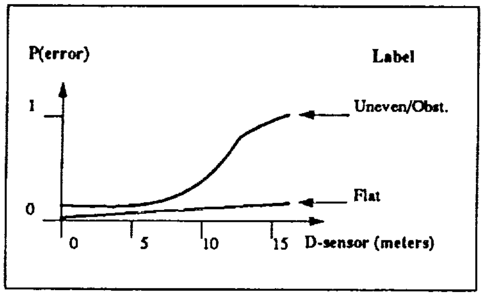

Figure 5: **Error probability on the cell labeling**

We are considering the application of a similar classification method on luminance images : global information concerning the same cells in the camera frame (color, texture...) should permit a fast determination of the terrain nature, and therefore produce a more significant description of the terrain. Another interesting thing to consider is the detection of areas of interest for the localization procedure (possible presence of landmarks or particular geometric features), using the attributes determined for each cell.

4.2 Global Model Building

In the incrementally built bitmap structure that represents the global terrain model, all the informations provided by the classification are encoded (label and corresponding confidence, elevation, slope). Fusion of the classifier output is a simple and fast procedure : each cell is written in the bitmap using a polygon filling algorithm. When a pixel has already been perceived, the possible conflict with the new perception is solved by comparing the label confidence values. This process is illustrated in figure 6 : the area originally labeled "obstacle" in front of the first position (left image) is split into two smaller obstacle areas plus a flat area when perceived from a smaller distance (right image). Many experiments have proved the robustness of this fusion method.

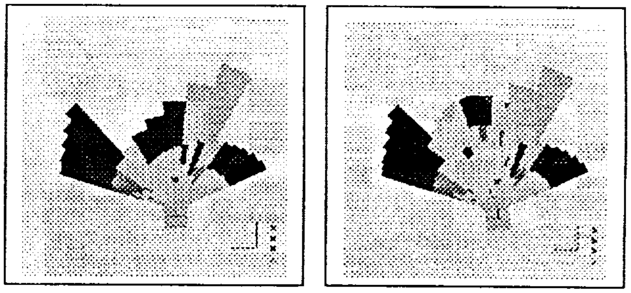

Figure 6: **Two steps of the global bitmap model building**

4.3 Connection Graph Building

Once the global bitmap representation is updated, it is structured in a "region model", thanks to classical image processing algorithms. Regions are areas of uniform label, uniform mean altitude and uniform confidence. If no precise geometrical informations are available in the description of a region, some useful qualitative informations can anyway easily be extracted, such as its surface or its including rectangle. A contour following algorithm provides all the neighborhood informations between the regions, that defines the topological connection graph. A node of the graph is related to the border between two regions, whereas an arc corresponds to the crossing of a region. Section 5.2.2 presents different ways to valuate the graph, considering the regions' attributes.

4.4 Fine Modeling

When an uneven area has to be crossed, it must be precisely modeled in order to plan a secure trajectory. We use for that purpose a generic interpolation method [20] that builds a discrete representation $z = f(x, y)$ on a regular Cartesian grid from a 3D spherical image $(\rho, \theta, \phi) = f(i, j)$.

Local Elevation Map (LEM) Building

Our method relies on the analysis of all sets of four neighboring points in the spherical image : they define *patches* in the Cartesian robot's redressed frame. Thanks to the fine grid resolution, a planar approximation is sufficient to represent a patch. The interpolation problem is then reduced to finding the intersection between each (x, y) "vertical" line and the plane that best approximate the patch. A test based on depth discontinuities allows to decide whether a patch can be interpolated or not, and leads to an estimation of the elevation Z_{Local} for the (x, y) interpolated points. An accuracy on each computed elevation is estimated, using Jacobian matrix of the sensor model to estimate variances on the raw Cartesian measurements, and a Kalman Filter to compute variances on the plane parameters [21].

Global Elevation Map (GEM) Building

A fusion of different LEM in a global elevation map may be needed for trajectory planning if the uneven area can not be entirely perceived from a single viewpoint. Once the estimation of the new robot's position is achieved (section 4.5), we combine the new LEM and the former Global Elevation Map into a new global map. The new elevation $(Z_{Global})_k$ after the k^{th} acquisition is updated by this ponderation equation :

$$(Z_{Global})_k = \frac{\sigma_{Z_G}^{-2}.(Z_{Global})_{k-1} + \sigma_{Z_L}^{-2}.(Z_{Local})_k}{\sigma_{Z_G}^{-2} + \sigma_{Z_L}^{-2}}$$

4.5 Localization Processes

Besides a localization process based on structured features [17], we developed a localization process that relies on the peak detection method [22], better suited for unstructured environments.

The specific terrain representation used here is a B-Spline surface based model, built upon an elevation map thanks to a least-square approximation. Such a model is very rich and compact, and provides a hierarchical description of the environment : a coarse level B-Spline representation is first computed on a uniform mesh, and a test based on the least-square errors points out the areas where some refinement is needed. A new mesh with smaller size patches is then defined, and a new B-Spline representation is computed, which ultimately leads to a tree model, in which each node corresponds to a B-Spline surface.

This analytic model allows to extract features such as high curvature points, valleys or ridges. We currently only implemented a peak extraction procedure based on a quick analysis of the *matrix* expression of the B-Spline surfaces. Once the peaks are extracted, we apply a feature matching localization method, cooperating with an iconic one : the iconic method is only performed in the neighborhood of the detected features. Hence, using small correlation windows, we avoid the long processing time usually encountered with such methods.

5 System Architecture and Control

The generic control architecture for the autonomous mobile robots developed at LAAS is organized into three levels [23, 24]. It is instantiated in the case of the EDEN experiment as shown in figure 7. The higher task planning level plans the mission specified by the operator in terms of tasks, with temporal constraints, executable by the robot. This operating station level, not currently used in the experiment, will be implemented in an specific environment to validate our teleprogramming approach.

Let's describe here the functional and decisional levels, and the way they are integrated.

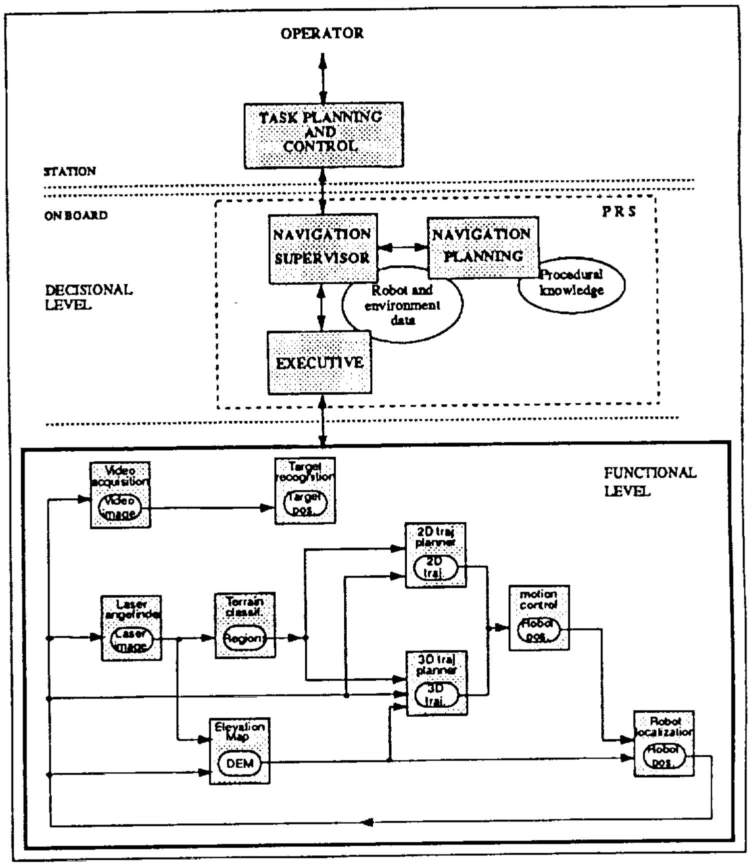

Figure 7: **Global control architecture. Connections between the modules at the functional level show data flow.**

5.1 The Functional Level

The Functional Level includes the functions for acting (wheels, perception platform), sensing (laser, cameras, odometry and inertial platform) and for various data processing (feedback control, image processing, terrain representation, trajectory computation, ...). To control robot functionalities and underlying resources, all these functions are embedded into modules defined in a systematic and formal way, according to data or resources sharing. Thus, modules are servers which are called via a standard interface, and allow to combine or to redesign easily the functions [25]. These modules can be viewed as a generalization of the logical sensor concept [26].

Figure 7 shows the set of modules used for the experimentation and the data flow during the progress of an iteration. The connections are dynamically established by the decisional level according to the context.

5.2 The Decisional Level

This level includes the navigation planner and a supervisor that establishes at run-time the dependencies between modules. It also controls their execution according to the context and the robot state, and installs the conditions/reactions in case of external events (watching for obstacles when executing a trajectory for instance). In our current implementation, the three entities of the decisional level have been simplified and merged together, using a Procedural Reasoning System [27].

5.2.1 The Supervisor and the Executive

The supervisor receives the task to be executed, described in terms of actions to be carried out and modalities. If the task is not directly executable (typically when the goal lies in an unknown area), the navigation planner refines it (section 5.2.2). The supervisor watches for events (obstacles, time-out, etc.) and reacts to them as planned, according to the dynamics of the situation and the state of the other running tasks. It sends to the Executive the different sequences of actions which correspond to the task, and sends back to the operator the informations related to task (*e.g.* specific data, and the report about its execution,etc.). The executive launches the execution of actions by sending the related requests to the functional level modules. It manages the access to resources and the coherence of multiple requests at the syntactic level, and can take into account the parallelism of some sequences (watching for obstacles while moving toward an intermediate goal for instance). It sends back to the supervisor reports about the fulfillment of those basic actions.

5.2.2 Navigation Planning

Generally speaking, the navigation planner uses procedures to carry out the task and decompose it into executable elementary actions, on the basis of the current environment and robot states. It is a key component of the decisional level : mixing both procedural knowledge and knowledge about the environment, it perform the decisions that provide the robot with a "smart" behavior. These decisions include *perception strategies*, *ie* the choice and the definition of the different perception tasks to perform, and *motion strategies*, that imply the definition of intermediate goals and the choice of navigation modes. The two problems are obviously closely linked, but to avoid a great complexity, we developed two independent techniques coupled afterwards.

Motion Strategies

The basic technique to plan a route in the known environment relies on the execution of an A^*-like search in the connection graph. This search selects a path, *i.e.* a succession of connected regions, that defines the intermediate goal and the motion mode to activate. The valuation of the arcs (that connect the region borders) is obviously determinant to implement different strategies. Our valuation is currently a heuristic mix between these criteria :

- **Arc label** : to plan a route that minimizes the execution time, the region label are taken into account. The planner then avoids to cross uneven areas when possible, since they require a fine modeling and a complex trajectory planning.
- **Arc confidence** : considering only the former constraint, the artifacts raised by the classification procedure (essentially badly labeled "obstacle" cells) would mislead the robot navigation. The arc label criterion is therefore pondered by its confidence, which allows the planner to cross some obstacles areas for instance, which actually triggers the execution of a new perception task when facing such areas.
- **Altitude variation** : For the purpose of energy saving, one may wish to minimize the positive altitude variations during trajectory execution, which increases the cost of climbing hills for instance.

Finally, let's note that a *localization ability* value can be taken into account while planning a route : from the localization model and the global bitmap model, landmarks (or interesting areas) visibility zones can be quickly computed, which produces a structure similar to a potential field. A localization ability value is then associated to each node of the graph, and a path that maximizes the sum of these values along the route can be determined.

Using some pre-defined rules, an analysis of the search result is then performed to define the next *perceptual need* among the three following : localization, discovery (perception of unknown area), and model refining (re-classification of an already perceived zone from a closest point of view or fine modeling).

Perception Strategies

Once the intermediate goal and the perceptual need are defined, the next perception task is performed according the following procedure [28] :

1. Perceptual **constraint checking** : characteristics on the sensor (field of view, resolution) and on the environment (visibility) constrains the observation ;
2. **Prediction** of the result of the perceptual task, *i.e.* estimation of the information it can provide ;
3. And finally **evaluation** of the contribution of the predicted task, in the context of the current need.

The main point here is to faithfully model the logical sensor to use ("classifier", "peak extractor",...), as in section 4.1.

As an example, let's examine a perceptual task selection : suppose the search in the graph derived a need to enhance the confidence value of a certain area. From the intermediate goal selected, the following procedure is run :

1. For each pixel of the global bitmap surrounding the sensor (within the LRF distance limit), the visibility constraint is checked using the elevation value encoded in the pixel ;

2. The current confidence label (Equal to zero if the pixel has not yet been perceived) of each perceivable pixel is compared to a theoretical "mean confidence value" the sensor can bring (deduced from the curves of figure 5). This comparison permits to estimate the amount of information the sensor can provide.

3. Finally, the usefulness of the predicted task is estimated, and the consideration of other constraints (allowed time, maximal sensor field of view...) defines its parameters, *ie.* perception direction, the LRF scanning mode, the field of view...

6 The EDEN Experiment

All the concepts and processes described in this paper are currently being integrated in the context of the "EDEN" experiment.

6.1 Experimental Test Bed

ADAM[4] has six motorized non directional wheels with passive suspensions, and is equipped with a "perception head" composed of a 3D scanning laser range finder with a deflecting mirror and two color cameras, mounted on a 1-axis pan platform.

The on-board computing architecture is composed of two VME racks running under the real time operating system VxWorks. They are connected to the operating station (a Sun SparcStation 10-41) by an Ethernet link. The first rack includes two 68030 CPUs and various I/O boards, and is dedicated to internal localization (thanks to odometry encoders and a inertial platform) and locomotion

The second rack is composed of two 68040 CPUS, three Datacube boards and some I/O. It is dedicated to sensing activities : video image acquisition, laser range finder command and acquisition, local processing of data.

During the experiments, most of the "high level" computing processes are run on the operating workstation to take benefit of a better debugging environment and of the pre-existence of the softwares under Unix. However, we have the possibility to embark all the softwares in a near future : some are already ported under VxWorks, and it is possible to use an on-board Sparc CPU under Sun-OS.

6.2 Experiments

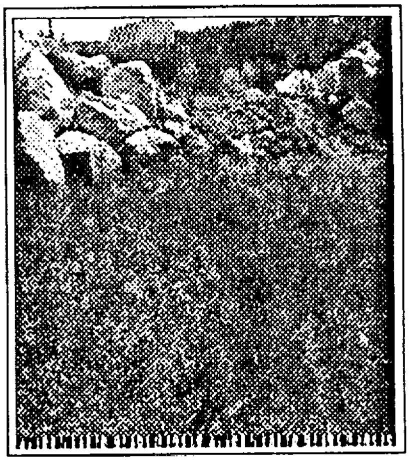

Figure 8: **ADAM's natural environment**

The figure 8 shows an illustrative image of ADAM's natural environment; it is a 20 by 50 meters area, composed of flat, sloppy, uneven rocky areas, and of big obstacle rocks. The canonical task is "GoTo Landmark", the environment being initially ***totally unknown***. The goal landmark is currently a 2D pattern detected and localized in a luminance image. We have performed several "reach that goal" experiments using only the 2D motion planner in the crossable zones, and a "discovery" strategy. After a few "perceive - analyze - plan" steps, (from 3 to 10, depending on the chosen path) Adam reaches the target located at an approximatively 30 meter distance from its starting point. The whole processing time does not exceed half a minute at each step, but due to the slow motion of the robot (its maximum speed is 28 cm/s) and the LRF image acquisition time, ADAM takes generally about 15 minutes to execute its mission.

We have also performed experiments using only the 3D motion planner; for this sake, we have partially integrated the following functions : fine terrain modeling, localization procedures and 3D trajectory planning on uneven terrain[5]

Figure 9 illustrates the position update and the terrain model updating performed after the third acquisition : the left figure shows the extracted features in the Local Elevation Map, built from the third depth image ; the right figure presents the corresponding correlated points (and the correlation windows) in the current Global Elevation Map. Figure 10 represents the new Global Elevation Map after the robot position updating and the fusion.

[4] Its chassis was built by VNII Transmach (S^t Petersburg, Russia)

[5] The computation time needed on a sparc II Sun station to build a Digital Elevation Map is about 2 sec.; the localization process takes about 3 sec., and the 3D planning process needs about 60 sec.

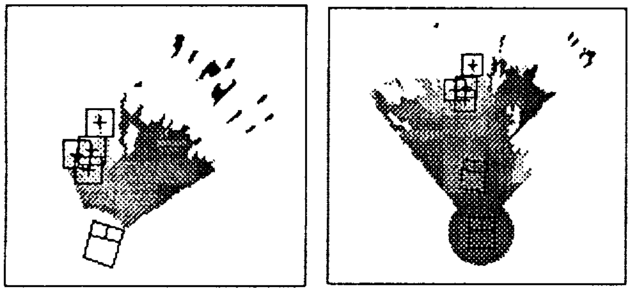

Figure 9: Position updating : how to merge the new LEM in the current GEM ?

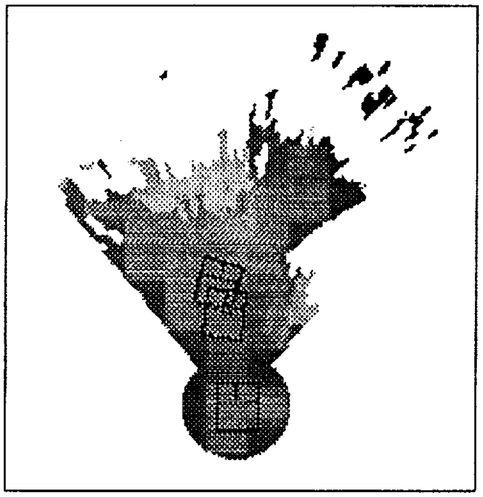

Figure 10: The new GEM after localization and fusion

Figure 11 is a perspective view of the reconstructed terrain on which the 3D trajectory of the robot has been planned and executed after 5 incremental steps (the grid discretization of the elevation map is 10 cm). The concatenation of the different 3D trajectories planned by ADAM to reach the goal is surimposed to the terrain model.

7 Conclusion and Future Work

We have presented an integrated multi-level perception system for an autonomous outdoor robot. This system points out several different modeling services, and enhances a lot the robot autonomy and efficiency. An ambitious experimental project, still under way, validates our adaptive approach and benefits to the development of highly demanding robotic applications, in particular planetary exploration.

A lot of difficult tasks have nevertheless still to be achieved, among which we retain the followings :

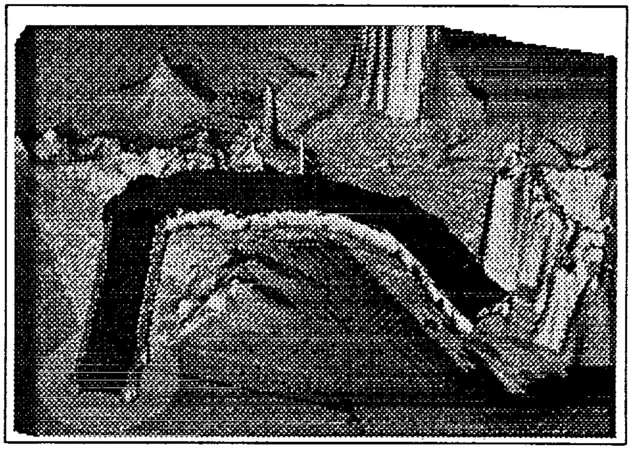

Figure 11: The GEM after 5 perceptions

- Besides the software complete integration of the whole system (and especially of the fine modeling and localization modules), each process performance needs to be improved and better validated. Feedback provided by the real data gathered during the experiments is here an essential information.
- The integration of a stereovision correlation algorithm would enhance the perception capabilities, by providing dense 3D and color data on a particular area. We could then address natural landmark recognition, and estimate the physical nature of the soil during the classification procedure.
- We currently only experimented the 2D navigation mode and the 3D navigation mode apart. Mixing both modes with a reflex one requires the development of "smart" navigation strategies. This topic needs particularly to be better formalized and tested ; the idea of developing exploration strategies in a topological connection graph whose arcs are valued with a certain confidence, while having the possibility of raising up this confidence (by acquiring data), appears to be promising.
- Memory management and consistency management of the models is a bottleneck to the execution of very long missions. The "sliding bitmap" concept we briefly presented has to be implemented and tested.
- Finally, improving the robot speed is fundamental, if not vital. The robot computing capacities should be better exploited, by implementing a kind of "pipeline" architecture.

References

[1] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Vision and navigation for the carnegie-mellon navlab", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 362–373, May 1988.

[2] G. Giralt, R. Chatila, and R. Alami, "Remote Intervention, Robot Autonomy, And Teleprogramming: Generic Concepts And Real-World Application Cases", *in IEEE International Workshop on In-*

telligent Robots and Systems (IROS '93), Yokohama, (Japan), pp. 314-320, July 1993.

[3] G. Giralt, R. Alami, R. Chatila, and P. Freedman, "Remote operated autonomous robots", *International Symposium on Intelligent Robotics, Bangalore (Inde)*, Jan. 1991.

[4] G. Giralt and L. Boissier, "THE FRENCH PLANETARY ROVER VAP: Concept and Current Developments", *in IEEE International Workshop on Intelligent Robots and Systems (IROS '92), Raleigh (North Carolina, USA)*, pp. 1391-1398, July 1992.

[5] F. D. Burke, "Past us studies and developments for planetary rovers", *in Misions, Technologies and Design of Planetary Mobile Vehicles, CNES, Toulouse, France*, Sept. 1992.

[6] R. A. Brooks, "A robust layered control system for a mobile robot", *IEEE Journal of Robotics and Automation*, Apr. 1986.

[7] R. Chatila, R. Alami, S. Lacroix, J. Perret, and C. Proust, "Planet exploration by robots : from mission planning to autonomous navigation", *in To be published in '93 International Conference on Advanced Robotics*, 1993.

[8] 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, Oct. 28-30*, 1993.

[9] S. Lacroix, P. Fillatreau, 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.

[10] M. Hebert, C. Caillas, E. Krotkov, I.S. Kweon, and T. Kanade, "Terrain Mapping for a Roving Planetary Explorer", *in IEEE International Conference on Robotics and Automation, Scottsdale, (USA)*, pp. 997-1002, 1989.

[11] M. Daily, J. Harris, D. Kreiskey, K. Olion, D. Payton, K. Reseir, J. Rosenblatt, D. Tseng, and V. Wong, "Autonomous cross-country navigation with the alv", *in IEEE International Conference on Robotics and Automation, Philadelphia (USA)*, 1988.

[12] E. Schalit, "Arcane : Towards autonomous navigation on rough terrains", *in IEEE International Conference on Robotics and Automation, Nice, (France)*, p. 2568 2575, 1992.

[13] R. Chatila and J.P. Laumond, "Position referencing and consistent world modeling for mobile robots", *in IEEE International Conference on Robotics and Automation, St Louis (USA)*, Apr. 1985.

[14] J. Barraquand and J.-C. Latombe, "Robot motion planning: A distributed representation approach", *in International Journal of Robotics Research*, 1991.

[15] T. Siméon and B. Dacre Wright, "A Practical Motion Planner for All-terrain Mobile Robots", *in IEEE International Workshop on Intelligent Robots and Systems (IROS '93) Japan*, July 1993.

[16] Z. Zhang, "Recalage 3d", Programme vap : Rapport final de phase 5, Institut National de Recherche en Informatique et en Automatique - Sophia Antipolis, 1992.

[17] P. Fillatreau and M. Devy, "Localization of an autonomous mobile robot from 3d depth images using heterogeneous features", *in IEEE International Workshop on Intelligent Robots and Systems (IROS '93), Yokohama, , Japan)*, July 1993.

[18] S. Betge-Brezetz, R. Chatila, and M.Devy, "Natural scene understanding for mobile robot navigation", *in Submitted to IEEE International Conference on Robotics and Automation, San Diego, California*, 1994.

[19] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.

[20] F. Nashashibi, M. Devy, and P. Fillatreau, "Indoor Scene Terrain Modeling using Multiple Range Images for Autonomous Mobile Robots", *in IEEE International Conference on Robotics and Automation, Nice, (France)*, pp. 40-46, May 1992.

[21] F. Nashashibi and M. Devy, "3D Incremental Modeling and Robot Localization in a Structured Environment using a Laser Range Finder", *in IEEE Transactions on Robotics and Automation, Atlanta (USA)*, May 1993.

[22] P. Fillatreau, M. Devy, and R. Prajoux, "Modelling of Unstructured Terrain and Feature Extraction using B-spline Surfaces", *in '93 International Conference on Advanced Robotics (ICAR),Tokyo (Japan)*, Nov. 1993.

[23] R. Chatila, R. Alami, B. Degallaix, and H. Laruelle, "Integrated planning and execution control of autonomous robot actions", *in IEEE International Conference on Robotics and Automation, Nice, (France)*, 1992.

[24] R. Alami, R. Chatila, and B. Espiau, "Designing an intelligent control architecture for autonomous robots", *in Third International Symposium on Experimental Robotics, Kyoto, Japan, Oct. 28-30*, 1993.

[25] R. Ferraz De Camargo, R. Chatila, and R. Alami, "A distributed evolvable control architecture for mobile robots", *in '91 International Conference on Advanced Robotics (ICAR),Pisa (Italy)*, pp. 1646-1649, 1991.

[26] T.C. Henderson, C. Hansen, and B. Bhanu, "A framework for distributed sensing and control", *in 9th International Joint Conference on Artificial Intelligence (IJCAI), Los Angeles, California (USA)*, 1985.

[27] M.P.Georgeff and F. F. Ingrand, "Decision-Making in an Embedded Reasoning System", *in 11th International Joint Conference on Artificial Intelligence (IJCAI), Detroit, Michigan (USA)*, 1989.

[28] K. Tarabanis, R.Y. Tsai, and P.K.Allen, "Automated sensor planning for robotic vision tasks", *in IEEE International Conference on Robotics and Automation, Sacramento, (USA)*, pp. 76-82, 1991.