A Landmark-based Motion Planner for Rough Terrain Navigation

Alain Haït, Thierry Siméon, Michel Taïx

ISER'97, Barcelona, June 15-18, 1997

A Landmark-based Motion Planner for Rough Terrain Navigation

A. Hait T. Siméon M. Taïx

LAAS-CNRS
7, Avenue du Colonel-Roche
31077 Toulouse Cedex - France
{hait,nic,taix}@laas.fr

Abstract: *In this paper we describe a motion planner for a mobile robot on a natural terrain. This planner takes into account placement constraints of the robot on the terrain and landmark visibility. It is based on a two-step approach: during the first step, a global graph of subgoals is generated in order to guide the robot through the landmark visibility regions; the second step consists in planning local trajectories between the subgoals, satisfying the robot placement constraints.*

1. Introduction

Autonomous navigation on natural terrains is a complex and challenging problem with potential applications ranging from intervention robots in hazardous environments to planetary exploration. Mobility in outdoors environments has been demonstrated in several systems [16, 7, 8, 3].

The adaptive navigation approach currently developped at LAAS within the framework of the EDEN experiment [3] demonstrates fully autonomous navigation in a natural environment, gradually discovered by the robot. The approach combines various navigation modes (*reflex, 2d and 3d*) in order to adapt the robot behaviour to the complexity of the environment. The selection of the adequate mode is performed by a specific planning level, the navigation planner [11] which reasons on a global qualitative representation of the terrain built from the data acquired by the robot's sensors.

In this paper, we concentrate on the motion planning algorithms required for the *3D navigation mode* [13], selected by the navigation planner when very rugged terrain has to be crossed by the robot. On uneven or highly cluttered areas, the obstacle notion is closely linked with the constraints on the robot attitude, and therefore contrains the robot heading position. Planning a trajectory on such areas requires a detailed modeling of the terrain and also of the robot's locomotion system.

Several contributions recently addressed motion planning for a vehicle moving on a terrain [15, 9, 4, 2, 6]. In particular, the geometric planner we proposed in [15, 6] is based on a discrete search technique operating in the (x, y, θ) configuration space of the robot, and on the evaluation of elementary

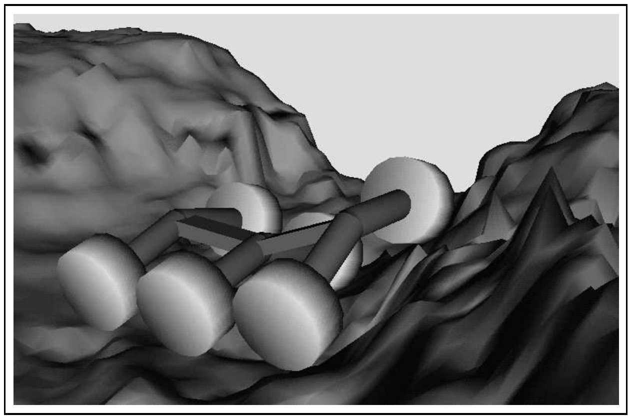

Figure 1. Placement of an articulated chassis

feasible paths between two configurations. The overall efficiency of the approach was made possible by the use of fast algorithms for placing the robot onto the terrain (Figure 1) and checking the validity of such placements.

In the navigation experiments previously conducted with a real robot [13], motion control was limited to executing the paths returned by this planner by relying on odometry and inertial data. The unpredictable and cumulative errors generated by these sensors, especially significant on uneven and slippery areas, often caused important deviations leading to a lack of robustness at execution. To overcome this problem, the robot has to be equipped with environment sensors (eg. cameras) that can provide additionnal information by identifying appropriate features of the terrain, and allow to use sensor-based motion commands. Such primitives are more tolerant to errors than classical position-controlled primitives, but their feasability has to be checked in term of visibility of the landmarks along the trajectory.

The contribution of this paper is to propose a motion planning approach which considers a set of given landmarks (eg. terrain peaks [5]) in the terrain model, and computes a partitionning of the terrain into regions where particular landmarks are visible. The approach allows to produce trajectories that remain, whenever possible, inside these *visibility regions* where the robot can navigate with respect to the given landmarks using closed-loop primitives relying onto the sensor's data.

2. The Motion Planning Approach

2.1. Problem statement

We consider a geometric model of an articulated robot illustrated by Figure 1. The robot is composed of several axles linked by passive joints allowing to adapt its shape to the terrain relief. The terrain is described by surface patches defined from an elevation map in z associated with a regular grid in (x, y). The terrain model also contains a set of point landmarks corresponding to major terrain features that the robot should be able to track at execution. The robot motions are constrainted by:

- **validity constraints** related to the feasibility of the motion (eg. stability of the vehicle, collision avoidance with the terrain, mechanical constraints).
- **visibility constraints** which traduce the ability of the robot to detect one or a set of landmarks from a given configuration.

While the first constraints need to be verified to guarantee the safeness of the motion, it would be too restrictive to only consider solutions satisfying the visibility constraint all along the path. Therefore, the planning approach should generate solutions alternating large portions verifying the visibility of the landmarks (sensor-based mode), and subpaths where the landmarks cannot be used at execution (position-controlled mode).

2.2. A two-step approach

In order to separate the integration of the two kinds of constraints, we propose a motion planner based on the following two-step approach:

- In the first step, subgoals are generated in order to guide the robot through landmark visibility regions computed from the terrain model and the landmark set. These subgoals and the start/final configurations are connected into a network of *possible* paths which minimize a cost reflecting the traversability of the terrain.
- In the second step, a motion planner based on [6] is used to transform the *possible paths* into feasible ones verifying the validity constraints.

The final trajectory is a sequence of trajectories from a subgoal to another one, alternately in and out of the visibility regions. We describe below the two steps of the approach respectively called **global planner** and **local planner**.

3. The Global Planner

3.1. Landmark visibility

The landmarks define a partition of the terrain into regions, called the **visibility regions**, where particular landmarks are visible. Consider a robot equipped with an omnidirectional sensor. A landmark $\mathcal{L}_i$ can be seen if the following constraints are respected:

- $\mathcal{L}_i$ is not hidden by a part of the terrain.
- the distance from the sensor to $\mathcal{L}_i$ is within some limit values.

For the first constraint, we use a hierarchical model of the terrain (see [15]) that allows an efficient collision checking: if the segment connecting the sensor center to the landmark does not collide with the terrain, then the landmark can be seen from this position. The test is performed for every position respecting the second constraint and the set of points obtained is the visibility region of the landmark.

For a given landmark $\mathcal{L}_i$, note that the visibility region may content several components. Each one is considered as a particular visibility region $\mathcal{R}_{ij}$.

3.2. Visibility graph

The visibility regions are then connected into a **visibility graph**. As illustrated by Figure 2, the graph has two type of edges. Some edges represent the possibility of a direct connection between two adjacent regions (ie. without crossing another region). These edges are first computed by a numerical propagation technique described in the next section. This propagation, issued from the boundary of the visibility regions, allows to associate to the edges a path minimizing a cost evaluated from the terrain shape. The endpoints of these paths define the nodes of the graph. They correspond to a set of subgoals located onto the boundary of the visibility regions. The graph is then completed by additionnal edges connecting the different subgoals generated in the same visibility region. The paths associated to these edges are also computed by a numerical propagation performed inside the visibility regions.

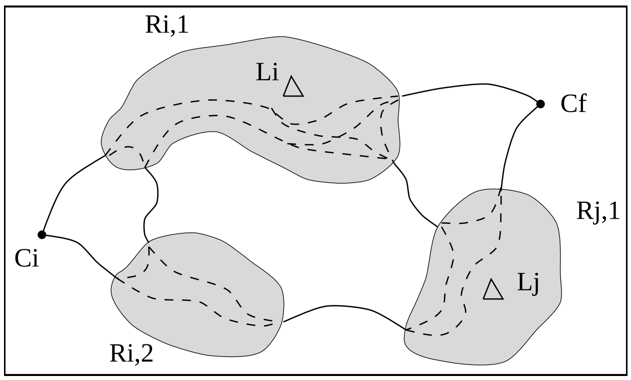

Figure 2. visibility regions and associated graph

3.3. The propagation algorithm

Let us now describe the propagation algorithm used to compute the visibility graph introduced in the previous section.

A **cost bitmap** is first computed by evaluating for each point of the terrain model, the slope and the roughness of a circular domain centered at this position, with a radius related to the size of the robot. It represents the difficulty for the robot to cross the domain.

The propagation consists in a wavefront expansion of a numerical potential obtained by integrating the cost across the bitmap, starting from each visibility region. The pixels belonging to the boundary of these regions are initialized to a null potential. At each iteration, the smallest potential pixel is expanded using an 8-neighbours propagation. Therefore, waves of increasing potential propagate from each region.

Whenever the waves issued from two regions meet together, an edge is created and the associated path is easily obtained by chaining back the pixels up to the associated region boundaries. The edge cost is the sum of the potentials computed at the meeting point. The propagation stops when all pixels have been expanded.

Path determination can take into account uncertainty growing outside the visibility regions by evaluating the maximum cost on a circular domain around the robot position, proportional to the uncertainty. This leads to a modification of the path that improve the robustness of the execution.

4. The Local Planner

The global planning step provides a sequence of subgoals between the initial and the goal configuration. The local planner is based on the algorithms described in [6]. Its role is to generate trajectories between the subgoals, while respecting the validity constraints (stability, non-collision, mechanical constraints).

The placement $\mathbf{p} = (\mathbf{q}, \mathbf{r}(\mathbf{q}))$ of the articulated robot (Figure 1) is defined by a configuration vector $\mathbf{q} = (x, y, \theta)$ specifying the horizontal position/heading, and by a vector $\mathbf{r}(\mathbf{q})$ of the joint parameters (the roll and pitch angles of the axles). For a given $\mathbf{q}$, the values of $\mathbf{r}(\mathbf{q})$ depend on the contact relations between the wheels and the terrain. Computing the robot placement is required for the evaluation of the validity constraints considered by the planner.

Planning is performed in two steps: a preprocessing step to compute robot placements on the terrain, and the planning step itself.

4.1. Preprocessing step

The preprocessing step is aimed at reducing the computational cost of the robot placement. It consists in slicing the orientation parameter, and in computing for each slice, two surfaces that characterize the evolution of the roll angle and elevation of a single axle, in function of its position onto the terrain. This step has only to be performed once, before the first call of the local planner.

4.2. Graph search

The planning step uses an approach similar to [1] in order to incrementally build a graph of discrete configurations that can be reached from the initial position by applying sequences of discrete controls during a short time interval. The arcs of the graph correspond to feasible portions of trajectory, computed for a given control. Only the arcs verifying the validity constraints are considered during the search.

The heuristic function used to efficiently guide the search is based on the potential bitmap computed by the global planner. It also allows to follow, whenever possible, the paths computed at the global level.

5. Results and Discussion

The motion planning approach has been implemented in C on a Silicon graphics indigo workstation with an R4000 processor. We describe below some simulation results.For these examples, the terrain is represented by a 114×179 elevation map (see Fig 3).

Several steps of the global planner are shown on Figure 4: the cost bitmap and potential bitmap computed from the terrain model and three landmarks (Fig. 4-a), and the obtained visibility graph (Fig 4-b). The landmarks visibility regions and the connecting paths are represented in dark on the figure.

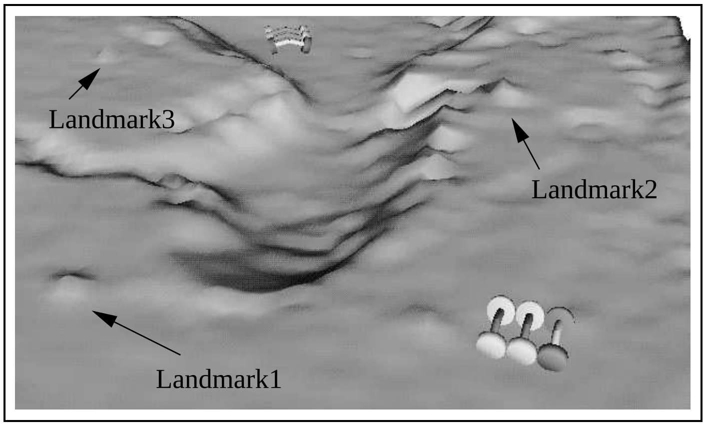

Figure 3. The terrain model, the selected landmarks and the initial/final configurations

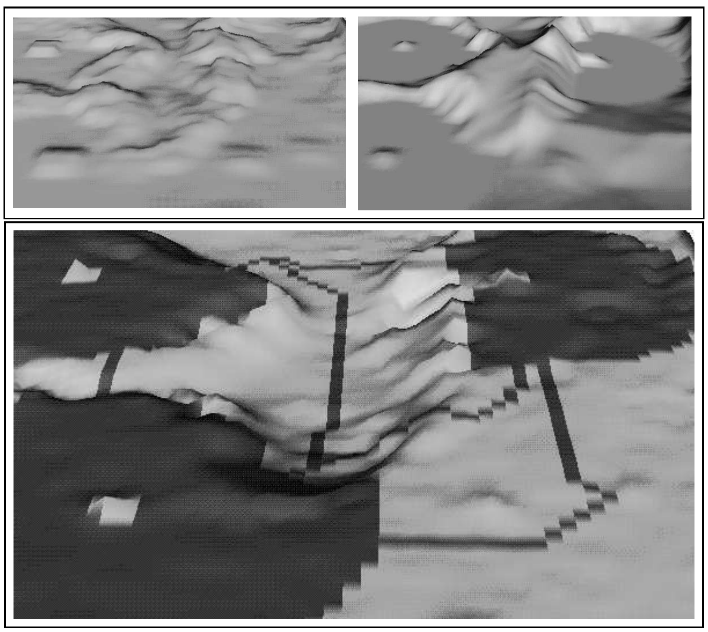

Figure 4. The global planning step. **a-** the cost and potential bitmaps and **b-** the computed visibility graph.

The final trajectory computed by the local planning step is displayed in Figure 5. One may note that a solution crossing the visibility regions computed for landmarks 3 and 1 would have been preferable for the visibility constraint. However, trajectories connecting the initial configuration to region 3 do not satisfy the validity constraints and the global planner had therefore to choose another path in the visiblity graph, leading to the final trajectory shown on

Figure 5. The final trajectory

		time (sec.)
Preprocessing	cost bitmap	3.260
	quadtree	1.540
	slices	13.850
Global planning	visibility regions	0.500
	potential bitmap	2.470
	visibility graph	0.930
Local planning	graph searchs	2.000

Table 1. Some computation times

the figure.
Computation times are reported in the following table. In the *preprocessing* step, data structures are computed from the terrain and the robot models: the cost bitmap, the hierarchical terrain model (quadtree) used by the collision checker and the orientation slices used by the local planner.

In the next example, only one landmark has been selected (Fig 6). The first trajectory represents planning without landmarks [6]. The interest of the landmark-based approach is to allow the robot localisation in the landmark visibility region. Computation times are similar to the previous example.

The last example shows the influence of uncertainty on path determination. Figure 7 represents cost bitmap for different uncertainty values. In Figure 8, taking into account uncertainty modifies the path by moving away from hazardous regions.

In the presented approach, the planned trajectories pass through the landmark visibility regions allowing robot localisation. Integrating uncertainty in cost bitmap is a first step toward planning with uncertainty. This should be completed in a future work.

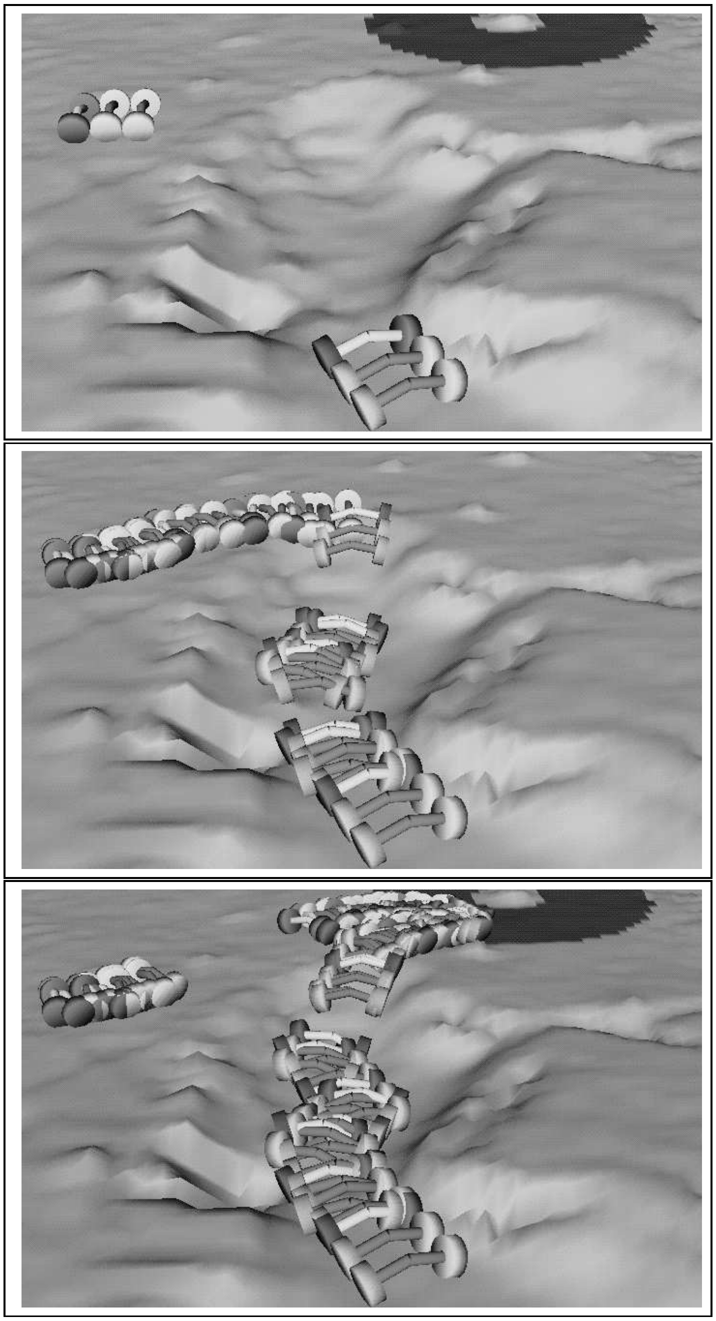

Figure 6. **a**-initial and goal configurations **b**-trajectory without landmark **c**-trajectory with landmark

6. Experiment

We are currently working on the integration of the presented motion planner in the EDEN experimental testbed. Experiments will be performed with the

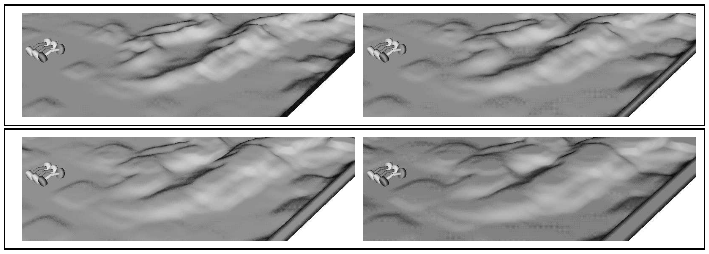

Figure 7. Cost bitmap with different uncertainty values

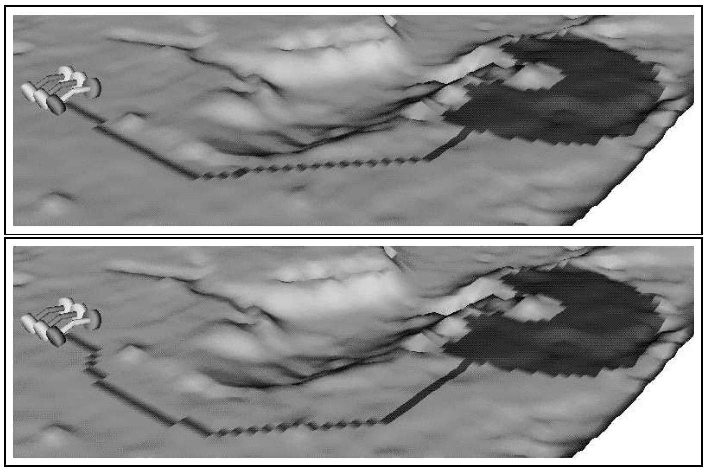

Figure 8. Path modification with uncertainty

LAMA Marsokhod (Figure 9) equipped with a range finder and a stereovision system for terrain modelling and landmark recognition. The purpose is to demonstrate autonomous navigation of a robot on rough terrain.

The robot is given a global goal in an unknown environement and perceives incrementaly the natural environment. At each step, a perception of a portion of the environment is done. This sensor information allows terrain modeling [13] and landmark extraction [5].

To use the landmark-based motion planner approach it is necessary to define a goal in the current perception. The goal position is chosen in order to ensure a good matching between the different perceptions.

At this level the landmark-based motion planner is called with this goal and landmark position. During the excution, landmark visual servoing is used inside the visibility regions, and classical odometry method outside these regions. After the path execution, a new step is done as far as the global goal is not reached.

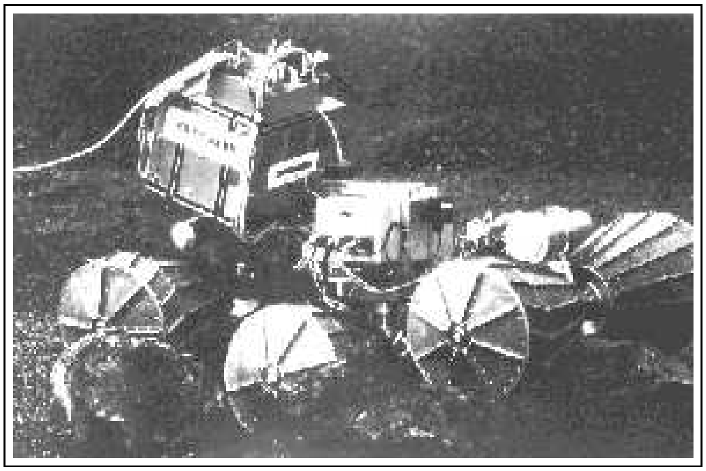

Figure 9. The mobile robot Lama

References

[1] J. Barraquand and J.C. Latombe. On non-holonomic mobile robots and optimal maneuvering. *Revue d'Intelligence Artificielle*, 3(2), 1989.

[2] F. Ben Amar, P. Bidaud. Dynamic Analysis of Off-Road Vehicles. In *Fourth International Symposium on Experimental Robotics, Stanford (USA)*, July 1995.

[3] R. Chatila, S. Lacroix, T. Siméon, M. Herrb. Planetary exploration by a mobile robot: mission teleprogramming and autonomous navigation. In *Autonomous Robots Journal, Vol. 2, n° 4, p 333-344*, 1995.

[4] M. Cherif, C. Laugier. Dealing with Vehicle/Terrain Interactions when Planning the Motions of a Rover. In *IEEE International Conference on Intelligent Robots and Systems, Munich (Germany)*, September 1994.

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

[6] A. Hait, T. Siméon. Motion planning on rough terrain for an articulated vehicle in presence of uncertainties. In *IEEE International Conference on Robots and Systems, Osaka (Japan)*, November 1996.

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

[8] E. Krotkov, M. Hebert and R. Simmons. Stereo Perception and Dead Reckoning for a Prototype Lunar Rover. *Autonomous Robots Journal, Vol. 2, n° 4, p 313-331*, 1995.

[9] T. Kubota, I. Nakatani and T. Yoshimitsu. Path Planning for Planetary Rover based on Traversability Probability. In *Seventh International Conference on Advanced Robotics, Sant Feliu (Spain)* September 1995.

[10] S. Lacroix, R. Chatila, S. Fleury, M. Herrb, T. Simeon. Autonomous navigation in outdoors environments: Adaptive approach and experiments. In *IEEE International Conference on Robotics and Automation, San Diego (USA)*, May 1994.

[11] S. Lacroix, R. Chatila. Motion and Perception Strategies for Outdoor Mobile Robot Navigation in Unknown Environments. In *Fourth International Symposium*

on Experimental Robotics, Stanford (USA), July 1995.

[12] J.C Latombe. *Robot Motion Planning.* Kluwer Pub., 1990.

[13] 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.

[14] Z. Shiller and J.C. Chen. Optimal motion planning of autonomous vehicles in three dimensional terrains. In *IEEE International Conference on Robotics and Automation, Cincinnati (USA)*, 1990.

[15] 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.

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