# Reactive Navigation in Outdoor Environments using Potential Fields

H. Haddad, M. Khatib, S. Lacroix and R. Chatila
**LAAS-CNRS**
7, Avenue du Colonel Roche, 31077 Toulouse Cedex 4 - France
e-mail : {haddad, maher, simon, raja}@laas.fr

## Abstract

The paper presents an approach to reactive navigation in cross-country terrains. The approach relies on a particular probabilistic obstacle detection procedure, that describes the area perceived by a pair of stereo cameras as a set of polygonal cells. To generate the motion commands on the basis of this terrain description, we present some improvements and adaptations to the classical potential fields technique. Results on real stereo data illustrate our contribution throughout the paper, and simulated long range traverses are discussed.

## 1 Introduction

Most of the contributions to autonomous navigation in outdoor environments consist in reactive sense-action loops (*e.g.* [8]. Indeed, building and maintaining global models of such environments is a quite difficult task. There has been several work dedicated to outdoor environments modeling, either to localize the robot, plan navigation tasks, or complex trajectories (see for instance [1, 7]). However, such global models are not required when the environment is essentially flat: in such cases, it is more efficient to reactively generate motion commands.

We consider in this paper an approach to reactive cross-country navigation in lightly cluttered environments. Our approach relies on potential fields and is sketched in figure 1 : a pair of stereo images is acquired, and a correlation procedure produces a dense 3D points image. This image is then analyzed to determine a polygonal map of the perceived area, for which the probability for each cell to correspond to an obstacle is estimated. The elementary motion commands are finally generated thanks to potential fields defined on *(i)* this local map and to *(ii)* on the result of a visual goal tracker.

This sequence is repeated as long as the goal is not reached, with a non-constant rate that is determined by the time necessary to process the data: when available, a new local map (or a new goal position) simply replaces the former one. The local maps are not fused into a global model, and the robot position is only defined relatively to the goal by the goal tracker: *no dead-reckoning is required.*

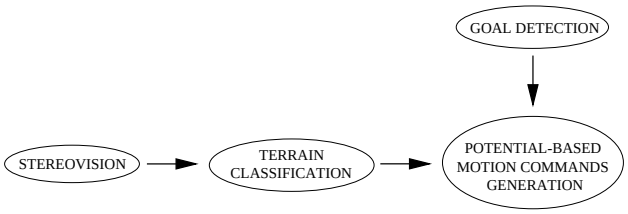



Figure 1: *The "reach-goal" reactive loop*

The paper essentially focusses on the generation of motion commands, and is organized as follows : the next section briefly presents the probabilistic obstacle detection procedure, that builds a polygonal obstacle map. Section 3 presents some improvements and adaptations to the classical potential fields technique, that ensure safe and smooth robot motions within this local map. Results on real data illustrates this section, and section 4 presents first long range motion results, obtained in simulation.

## 2 Obstacle Detection

The difficulty of representing outdoor environments comes essentially from their geometrical and *physical nature* diversity. However, several obstacle detection techniques on the basis of 3D data have been proposed [3, 9]. Most of these techniques produce a *binary* description of the environment, in term of traversable and non-traversable areas. In order to ensure safe robot motions on such representations, one must force the number of non-detection to be as close as possible to zero, which often induces false alarms, and thus "over-constrains" the problem. We present here a classification procedure that produces a polygonal map, in which a probability for each cell to correspond to an obstacle is estimated: it allows to consider costs based on a *risk* to plan paths or to generate motion commands.

This 3D data classification procedure, initially developped for complex terrains [1], relies on a specific discretisation of the perceived area, that defines a *cell image*. Attributes are computed for each cell, and are used to label the cells in term of *navigation classes*, thanks to a supervised Bayesian classifier.

### 2.1 Cells definition

The sensors that produce 3D points, be it a laser range finder or a stereo-vision correlation algorithm, generally have a constant scanning rate whithin the *sensor frame*. But when perceiving a ground plane with such sensors, the resolution of the perceived points decreases dramatically with their distance to the sensor. This fundamental point lead us to choose the discretization of the perceived zone presented in figure 2, instead of a cartesian one such as in [3]. This discretization corresponds to the central projection on a virtual horizontal ground of a regular (cartesian) discretization in the sensor frame.

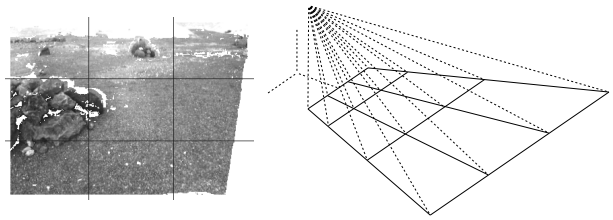


Figure 2: *Discretisation of a 3D stereo image : regular Cartesian discretisation in the sensor frame (left - only the correlated pixels are shown), and its projection on the ground (right - the actual discretisation is much finer)*

The fundamental property of this discretization is the "conservation of density" : on a perfect flat ground corresponding to the reference plane, the number of points that belong to a cell - *i.e.* whose vertical projection coordinates are bounded by the cell's border - is equal to a constant *nominal density*, defined by the discretization rates. On the other hand, a cell covering an obstacle area contains much more points than the nominal density. The number of points contained in a cell is then an important feature to determine the nature of the perceived zone covered by this cell.

Other features are used to identify a cell : the elevation standard deviation and the maximum elevation difference computed on the cell's points give an information on the "flatness" of the cell ; the mean normal vector and the variances on its coordinate are useful to determine if the cell covers a regular slope or has an unregular surface. The cell distance to the sensor is also an important feature : there is actually no correlation between this distance and the cell's class, but all the former features strongly depend on it. The introduction of this distance feature comes to implicitely taking into account the sensor's uncertainty and resolution properties in the classification process.

### 2.2 Cell classification

A supervised bayesian classification procedure is used to label each cell: during an off-line learning phase, a human prototyped a set of cell images. On line, once 3D data are acquired, the cells image is defined by the discretization, and the feature vector $F$ is computed for each cell. The Bayes theorem is then applied to determine the partial probabilities $P(C_i|F)$ for a cell to correspond to each of the $M$ terrain classes $\{C_1, \ldots, C_M\}$ :

$$P(C_i \mid F) = \frac{P(F \mid C_i)P(C_i)}{P(F)} = \frac{P(F \mid C_i)P(C_i)}{\sum_{i=1}^{n} P(F \mid C_i)P(C_i)},$$

where $P(F \mid C_i)$ are the pdf's computed thanks to a nearest neighbour technique applied in the feature space filled with the prototyped cells, and $P(C_i)$ are the *a priori* probabilities to perceive an area that belongs to the class $C_i$.

This classification procedure is very fast and robust : we tested it on hundreds of images, using various prototype data bases. In the case of the application considered in this paper, we only consider two terrain classes, *i.e.* flat and obstacle (Figure 3).

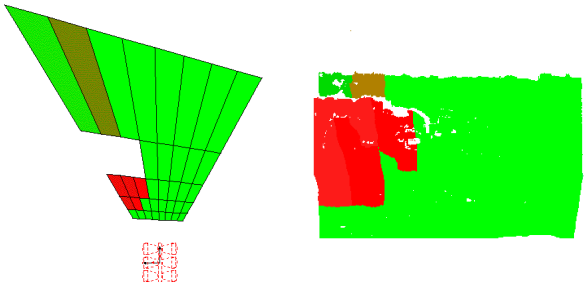


Figure 3: *Classification of the stereo image of figure 2 : perceived area (left) and reprojection in the sensor frame (right - only points closer than 15m are considered). The grey levels represent the partial probability* $P_{Obst}$.

## 3 Obstacle avoidance

### 3.1 Potential functions

The artificial potential field method originally proposed in [6] is one of the most widely used techniques for mobile robot local collision avoidance: it is well adapted for real-time motion control. Its principle is simple: the robot motion is defined by the application of *(i)* an attractive force generated by the goal and of *(ii)* repulsive forces generated by the obstacles.

Consider an obstacle $\mathcal{O}$, represented by an analytical function in the plan $(O, x, y)$, or as shown later by their enveloping segments. If $X_g$ represents the goal position and $X$ the current robot position, the artificial potential function defined at $X$ is of the form:

$$U(X) = U_g(X) + U_r(X)$$

where $U(X)$ is the resultant potential, $U_g(X)$ the attractive potential produced by the goal at $X$ and $U_r(X)$ the repulsive potential induced by the obstacle at $X$. The resultant force $F$ is then:

$$F(X) = F_g(X) + F_r(X)$$

where $F(X) = -\vec{\nabla}\ U(X)$. $F_g$ is an attractive force which guides the robot to the goal and $F_r$ is an artificial repulsion induced by the obstacle surface.

### 3.1.1 Attractive potential

The attractive potential such as proposed in [6] induces a force whose intensity is only function of the distance between the robot and the goal. As shown in [4], with such an attractive force, it is difficult to define the repulsive forces parameters to obtain a "constant" behavior with respect to the obstacles, be they far or close to the goal. To cope with this, an attractive potential function having a quadratic behavior at the goal neighborhood and an asymptotic linear behavior away from it has been proposed in [4]:

$$U_g = K_g\sqrt{d_g^2 + R^2}$$

where $d_g$ denotes the distance to the goal, $K_g$ is a positive gain constant, and $R$ is a weighting function. The force deduced from this potential is thus:

$$F_g = -K_g \frac{d}{\sqrt{{d_g}^2 + R^2}} \frac{\partial d_g}{\partial X}.$$

where

$$R = d_m\sqrt{\frac{1}{{F_g^*}^2} - 1},$$

is a function of a user-defined limit distance $d_m$ and a maximum attractive force $F_g^*$. Thus, $F_g$ is asymptotically constant for distances greater than $d_m$, and tends to zero as $d_g$ tends to zero (figure 4).

### 3.1.2 Repulsive potential

Most of the proposed repulsive potential functions in the literature depend only on the distance to the obstacles. A major drawback of such potentials is that obstacle segments have an influence on the robot even if the robot is moving in a parallel direction. This can lead to irregular motions, especially in our case where the environment description is polygonal. To cope with this, some authors introduced some parameters describing the geometrical situation between the robot and the obstacles to define the repulsive function [2].

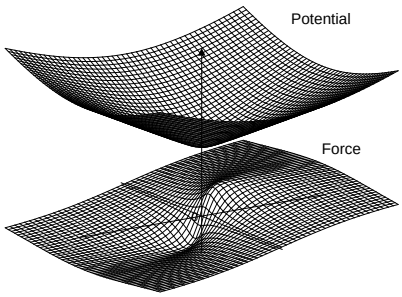



Figure 4: *Shape of the attractive potential and corresponding force we consider*

We have chosen to use the *rotational potential* approach, which initially proposed in [5]. In this approach, the resultant repulsive potential is a linear combination of two repulsive potentials $p_1(x)$ and $p_2(x)$:

$$U^\alpha(X) = \begin{cases} \alpha^2 p_2(X) + p_1(X) & \text{if } d < d_0, \\ 0 & \text{else-wise.} \end{cases} \quad (1)$$

where $d_0$ is a limit distance over which the obstacle has no influence on the robot motion, and $\alpha = \parallel \sin(\Delta\theta) \parallel$ ($\Delta\theta$ is the relative angle between the robot direction and the nearest cell segment - figure 5).

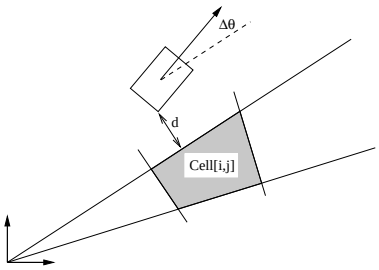
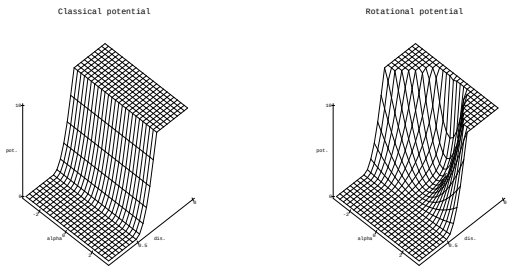



Figure 5: *Cell's repulsive force computation.*

$p_1$ and $p_2$ are defined as follows:

$$p_1(X) = \frac{1}{2}k_1\left(\frac{1}{d} - \frac{1}{d_0}\right)^2,$$
$$p_2(X) = \frac{1}{2}k_2\left(d - d_0\right)^2.$$

$p_1(x)$ has a quadratic behavior and $p_2(x)$ a linear one. The aim of $p_1(x)$, which dominates in the neighborhood of the obstacle surface, is to guarantee the non-collision ($p_1(x) \rightarrow \infty$ when $d \rightarrow 0$). $p_2(x)$ having a linear behavior, it dominates in the region near the limit distance boundary and acts to smoothly pre-avoid the obstacle considering its relative orientation expressed by the parameter $\alpha$ (figure 3.1.2).



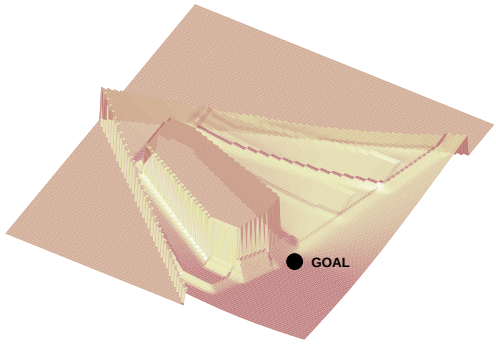




Figure 6: *Potential field for a cell image.*

## 3.2 Adaptation to the cell images

### 3.2.1 Cells Repulsive Potential

Our purpose is to guide the robot safely in the perceived area using the classified image. For security reasons, we divide the cells into two categories:

• **obstacle cells:** are the non classified cells or cells with $P_{Obst} \geq \mathcal{S}^o$ where $\mathcal{S}^o$ is an obstacle threshold. The repulsive potential function associated to such cells is the rotational potential defined in 1;

• **traversable cells:** are the classified cells with $P_{Obst} < \mathcal{S}^o$. For a traversable cell $[i, j]$, the repulsive potential function is given by :

$$U_r[i,j](X) = \begin{cases} \alpha^2 \; P_{Obst}[i,j] \; p_t(X) & \text{if } d < d_0 \\ 0 & \text{else-wise.} \end{cases}$$

where $p_t(X) = k_t \frac{1}{2} (d - d_0)^2$. The force issued form this potential is linear and has a limit intensity as $d$ tends to zero. This intensity is also modulated by the parameter $P_{Obst}$ , the gain of this potential, which represents the cell traversability danger. As a consequence, the robot subjected to such a potential can either avoid these cells and come into regular terrain, or traverse them if no better solution exists. The choice of $k_t$ must insure that the traversability of these cells is still possible when the robot is subjected to forces coming from obstacle cells.

### 3.2.2 Resultant potential

The resultant repulsive potential $U_r$, generated by the cell image, is the sum of all cells repulsive potentials $U_r[i, j]$:

$$U_r(X) = \sum_{i,j} (U_r[i,j])$$

Note that and $U_r[i, j] = 0$ if there is an intersection between the robot and the $cell[i, j]$. Figure 6 shows the resultant potential $U = U_g + U_r$ for the cell image presented on figure 7, with the same goal position.

Figures 7 illustrates the advandage of the rotational potential over the classical repulsive potential. We use an obstacle threshold $\mathcal{S}^l$ equal to $0.75$ : the two dark grey cells on the right are obstacle cells, whereas the grey cell on the left is a traversable cell ($P_{Obst} = 0.45$) that could be traversed if necessary.

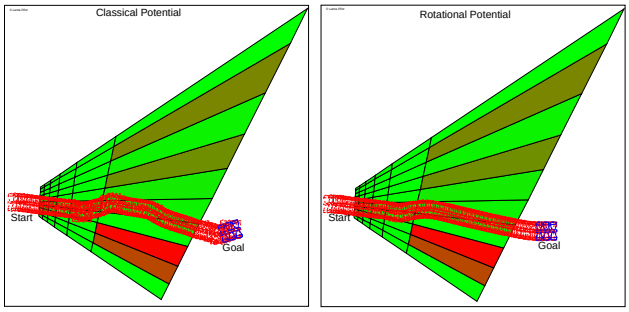



Figure 7: *"Go to goal" with classical repulsive potential (left) and with the rotational potential (right)*

## 3.3 Dynamic adaptation of the limit distance

The robot's safety is the most important argument which influences our control parameters choice, the most important one is the limit distance $d_0$: it determines the influence area rang of a cell on the robot behavior. Using a rather big value of this distance would ensure safe and smooth motions of the robot, that would avoid getting to close to the obstacle (and therefore eliminate irregular maneuvers). Unfortunately, this solution constrains too much the robot motion if there are many obstacles around, because the range of influence will be sufficiently big to include obstacles which do not disturb the robot.

To tackle this problem, we propose to use a dynamic limiting distance rather than a static one:

$$d_0(\phi) = d_m f(\phi) = d_m \left( \frac{1 + e^{(\cos(\phi))}}{1 + e} \right)$$

where $d_m$ is the security distance corresponding to $d_0$ in the static case, and $\phi$ is the angle between the robot motion direction and the heading angle of the closest point to the

obstacle. With such a definition of the weighting function $f(\phi)$, the obstacle influence on the robot depend on their relative position to the robot (figure 8): the limit distance $d_0(\phi)$ is maximal for $\phi = 0$ (the obstacle is in front of the robot), and is minimal for $\phi = \pi$ (the obstacle is behind the robot).

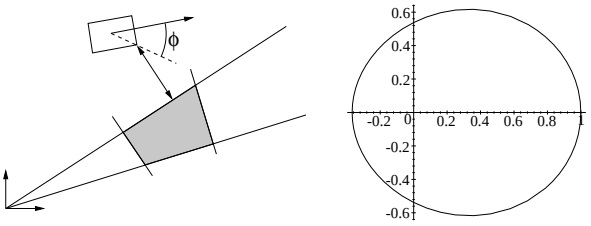



Figure 8: *Definition of the angle $\phi$ (left), and weighting function $f(\phi)$ in polar coordinates (right)*

Figure 9 illustrates the advantage of the adaptation of the limit distance, in the case where the robot moves between two obstacles. Note also that using a static limit distance, the robot can hardly reach its goal because of an important repulsive force generated by the obstacle close to the goal. Using $d_0(\phi)$, the robot motion is smoother between the obstacles, and the influence of the obstacle close to the goal is reduced considerably.

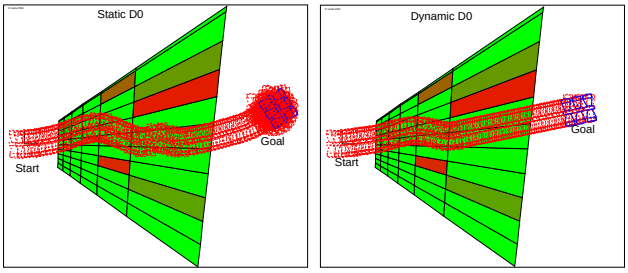



Figure 9: *"Go to goal" with static $d_0 = 2m$ (left) and with dynamic $d_0(\phi)$ (right, $d_m = 2m$). The robot width is 1 meter*

# 4 First results

## 4.1 Dealing with a sequence of images

The difficulty of dealing with a sequence of images arises from the fact that the processing time $t_{acq}$ needed to obtain a cell image is not negligible with respect to the robot speed.

Let us assume that at the instant $t$ of the image acquisition, the robot is in a cell image $V_{t-1}$. From $t$ until the availability of the next cell image $V_t$, the robot is bounded in $V_{t-1}$, since it is the only area on which it has informations. At the time $t$ of the new image acquisition, only the boundaries of the future cell image $V_t$ are available. Therefore, the robot motion executed during the processing time $t_{acq}$ must constrain the robot to reach the area $V_{t-1} \cap V_t$ (figure 10).

This constraint reflects an important implication: the time $t_{acq}$ and the robot speed being given, one must make sure that at any time there is enough free space in front of the robot such that the intersection between the current cell image and the next one still accessible. This constrains the robot to avoid sharp turns and get close to ahead obstacles. This is made possible thanks to the various improvements presented in section 3.1: it was very easy to determine the value of the limit distance $d_m$ to solve the problem in our context (the actual maximum speed of Lama[1] is around $0.2m/s$, and image processing time is around $5$ seconds).

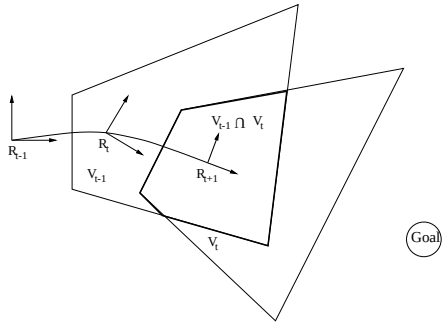



Figure 10: *Linking an image sequence*

## 4.2 Results

To integrate and intensively test our algorithms, we have completed a simulation system on Unix platforms: a ray-tracing algorithm run on a digital elevation map produces 3D points images, at the resolution of the cameras that equips Lama. The classification procedure being rather qualitative, its results on simulated 3D images are similar to those on real images. In the simulation, we impose the maximum robot speed to $(0.2m/s)$ and image processing time to $(5s)$. With such values, a new cell image is provided to the system almost every meter (figure 11). On figure 12, one can note the smoothness of the executed paths.

We ported the algorithms under the real-time OS VxWorks to test them on the robot Lama, and performed several experiments, that showed smooth and sage path executions.

# 5 Conclusion

We developed a complete reactive navigation system that relies on the artificial potential fields method combined with a probabilistic terrain classification procedure. We proposed several improvements and adaptations of the

[1] Lama is a Marsokhod-like robot, that is lent to us by Alcatel

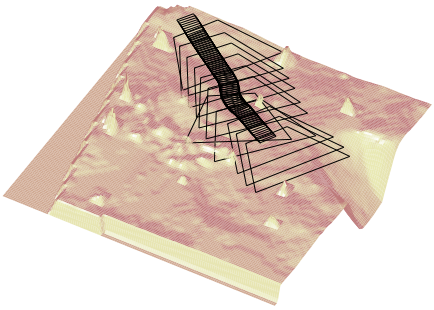


Figure 11: *Complete image sequencing simulation*

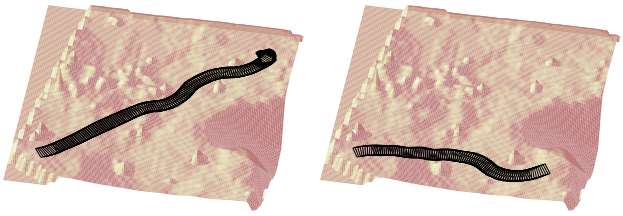


Figure 12: *Simulation results*

classical potential fields technique to ameliorate the robot behavior on outdoor terrains. The main advantage of our navigation algorithm is expressed by its behavior with respect to the obstacles and traversable zones: experiments showed that it produces smooth, safe and steady paths.

Future work will essentially focus on an intensive experimental evaluation of the method, under various environments setups, and considering various robot speeds and data acquisition parameters (size and resolution of the images). Dynamic cameras orientation control in order to satisfy both local map building and visual goal detection will also be tackled. Following these short term objectives, we would like to implement other reactive navigation approaches (using other means to detect the obstacles or to generate the motion), in order to experimentally evaluate them.

## References


[1] R. Chatila, S. Lacroix, T. Siméon, and M. Herrb. Planetary exploration by a mobile robot : Mission teleprogramming and autonomous navigation. *Autonomous Robots Journal*, 2(4):333–344, 1995.

[2] D.N. Green, J.Z. Sasiadek, and G.S. Vukovich. Path traking, obstacle avoidance and position estimation by an autonomous, wheeled planetary rover. In *IEEE International Conference on Robotics and Automation, San Diego California, (USA)*, pages p. 1300–1305, 1994.

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

[4] M. Khatib. *Sensor-based motion control for mobile robots*. PhD thesis, LAAS-CNRS, Toulouse, France, December 1996. – Ref.: 96510.

[5] M. Khatib and R. Chatila. An extended potentiel field approach for mobile robot sensor-based motions. *Intelligent Autonomous Systems (IAS'4), Karlsruhe (Germany)*, pages 490–496, 1995.

[6] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[7] I.S. Kweon and T. Kanade. High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):278–292, Feb. 1992.

[8] D. Langer, J.K Rosenblatt, and M. Hebert. A behavior-based system for off-road navigation. *IEEE Transactions on Robotics and Automation*, 10(6):776–783, 1994.

[9] L. Matthies, A. Kelly, and T. Litwin. Obstacle detection for unmanned ground vehicles: A progress report. In *International Symposium of Robotics Research, Munich (Germany)*, Oct. 1995.