

Compact Modeling Technique for Outdoor Navigation

Cristina Castejón, Dolores Blanco, and Luis Moreno, *Member, IEEE*

Abstract—In this paper, a new methodology to build compact local maps in real time for outdoor robot navigation is presented. The environment information is obtained from a 3-D scanner laser. The navigation model, which is called traversable region model, is based on a Voronoi diagram technique, but adapted to large outdoor environments. The model obtained with this methodology allows a definition of safe trajectories that depend on the robot's capabilities and the terrain properties, and it will represent, in a topogeometric way, the environment as local and global maps. The application presented is validated in real outdoor environments with the robot called GOLIAT.

Index Terms—Navigation map, outdoor, robot mapping, traversability, 3-D scanner laser.

I. INTRODUCTION

THE NEW mobile robotic applications, such as human service or planetary exploration, make the robots work in changing outdoor environments. Outdoor environments are less structured, less controlled, and less predictable than indoor environments. This new challenge has prompted a change in robotics navigation philosophy, where path planning and modeling were always obtained *a priori*. The evolution over unknown and changing environments leads the study toward computing time reduction and the robot's safety [1]. To give the robot autonomy skill, knowledge about the environment, where the robot is going to move, to manipulate objects and to navigate without collision or plan trajectories is needed.

The great outdoor environment navigation problem is the variety of terrains that exist (rough terrains, high obstacles density, partially structured terrains, or completely unknown environments). That is, for all outdoor environments and robots, a single model does not exist. For this reason, a new philosophy about the robot capabilities, such as perception, localization, and modeling, must be developed.

In this paper, we present a method to model wide outdoor areas in terms of the mobile robot's ability to move through them. The methodology allows different types of models to be built, depending on the robot task. An application for navigation tasks is also developed in this paper. The model obtained for navigation is called traversable region model (TRM), and it

represents the crossable areas of the environment perceived by a 3-D scanner laser. The model is robot-oriented, because some terrains can be crossable or not depending on the robot's capability. In that sense, the map will use a 2-D Voronoi-based model to represent the \mathcal{XY} environment information and a qualitative description of the terrain traversability to represent the \mathcal{Z} environment information. This description is highly compact and very easy to manage with the planning or navigation system. The system will use a differential global positioning system (GPS) to integrate local information into a global map.

In Section II, a state of art related to outdoor modeling and the traversability concept is presented. In Section III, we present the general structure of the modeling methodology. Then, in Section IV, the terrain analysis used to obtain the robot's traversability skill is developed, highlighting the equations needed to obtain the basic parameters. In Section V, the TRM based on Voronoi diagrams (VDs) and the 3-D data preprocessing is calculated and presented. In Section VI, the global model is developed, which is based on the local VDs integration and a GPS. Section VII shows the results of experimental tests carried out in real outdoor environments and with a large size outdoor robot called GOLIAT. Finally, conclusions will be presented in the last section.

II. PREVIOUS WORKS

The majority of work in navigation maps has dealt with robots operating on flat terrains such as planetary environments [2], highways [3], airports, etc. In these cases, the terrain has a low degree of roughness, and the obstacles have a known shape (usually spherical or cubic), and they are placed on a flat area; therefore, they are easy to model with geometrical primitives. In general, in mobile robot navigation, the occupancy-based approach is one of the most commonly used methods [4]. This representation is easy to manipulate, and discretization facilitates a sorted and specific information for the robot tasks. For the discretization construction, the sensor information is transformed using techniques such as occupancy grids [5], digital elevation maps (DEMs) [6] or special representations such as octrees, quadtrees [7], etc.

On the other hand, other approaches refer to the level of environment representation that we divide into geometrical, topological, and topogeometric levels. Geometrical level approach has been shown to be not of interest for navigation modeling. The 3-D reconstruction approach for indoor and outdoor environments can be found in literature [8], [9]. In these works, the large quantity of information required to

Manuscript received March 14, 2005; revised November 20, 2005 and February 2, 2006. This work was supported by the Spanish Government through the MICYT project DPI2003-01170. This paper was recommended by Associate Editor A. Ollero.

The authors are with the University Carlos III of Madrid (UC3M), 28911 Madrid, Spain.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCA.2007.904786

model the environment is stated. Their main disadvantage is the huge computational cost, when it is used in wide and poorly structured outdoor environments. The advantage of geometrical models derives from the ability to generate accurate maps, close to the real environment. Nevertheless, these representations are of limited use in outdoor modeling due to its high time computing, the lack of geometrical structure in some kinds of environments, and the bulky models that are generated. Another problem is the computational cost required to interpret these maps for navigation.

In topological models, the numerical information is replaced by symbolic data, and the map is represented as a graph. Topological models have been proven to be useful in navigation along corridors and in topological localization. Topological model applications in outdoor environments have not been found, despite their effectiveness in large environments [10], [11]. They allow path planing in a fast way and reduce the amount of data and the computing time. However, the graph representation needs repetitive elements (such as doors, walls, corners, etc.) that do not exist in outdoor environments. This technique is of interest for modeling large indoor environments where a great deal of common objects exists. The objects allow the robot to localize and to move. However, for outdoor environments, *a priori* unknown, these models are not successful.

Hybrid representations are a good alternative to the methods described above. This method combines different modeling techniques to take advantage of each method and to overcome the disadvantages. Currently, it is a good choice for navigation map modeling. Some authors work with topogeometrical models [12] where they carry out a terrain discretization using a sensor-based model. Thrun [13] builds a topological graph based on an occupancy grid. Betgé-Brezetz *et al.* [14] describe, in a topological way, the locations starting with the geometrical information. The authors build planes to represent the flat and superquadrics for the obstacles representation. With digital image techniques and border extraction, the objects are separated from the environment to obtain the topological model. This description is limited to easy environments with circular-form obstacles on the terrain. Other authors build topological models from the geometrical information provided by the sensor system. Simon and Dudeck [15] propose a global hybrid topological model, which is built from local maps called islands of reliability. Each local map has a quantitative information about the environment, which is based on the metric information from the sonar. The same sensor is used in easy indoor environments by Choset *et al.* [16], which describe the sensor-based exploration with VDs. The system builds incrementally a connected graph with points that belong to the Voronoi edges. This technique is used by Blanco *et al.* [17], with a 2-D scanner laser, and it will be used in this paper to model traversable regions as we will justify in Section V-A. The model will be represented as a VD [18] based on the sensor information provided by a 3-D scanner laser. Some modifications will be necessary in order to adapt this approach to start from the 3-D data.

Previous to the navigation model, objects from the 3-D robot workspace must be determined. Several authors have studied different classification methods in order to separate objects from

the scene. Vandapel *et al.* [19] develop a method to segment 3-D points, provided by a ladar, into three classes such as surfaces, linear structures, and scatter in order to classify a vegetated terrain. The method is based on statistical classification and learning techniques [20]. As authors comment, the approach is too expensive for on-board processing due to the number of 3-D points. For our purposes of building a compact navigation model, so much detail is not necessary.

A. Traversability

The traversability attribute represents the terrain's suitability to be navigated. Several authors have studied this attribute. The traversability concept was introduced by Langer *et al.* [3], and it highlights the region's capacity to be crossable or not by a robot.

Most of the works related to outdoor navigation try to divide the perceived terrain into regions with different characteristics (in a segmentation process). Langer *et al.* [21] perform a terrain segmentation and generate a list of crossable regions for the robot, with depth images provided by a stereovision system and the sorted information with a DEM. The classification allows the robot in [3] to navigate over roads and highways, that is to say, partially structured easy environments. For planetary environments, Seraji [22] introduces the concept of a traversability index. The index expresses the suitability of a terrain to be crossed based on the physical properties such as slope and roughness.

A number of authors work with two base parameters: the terrain slope and the roughness degree. To obtain the slope, algorithms have already been developed, for example, the general slope calculation using the horizontal line obtained in 3-D coordinates, with a stereovision system and interpolating only one plane, as presented by Howard and Seraji [23], or the use of neural networks with complex training, which is based on the presentation of different terrain patterns to obtain a numerical slope [24].

The last two works presented above have been developed specifically for planetary type environments, that is to say, flat terrain with defined form obstacles (basically rocks over a flat surface). Other authors compute the slope in each point. The number of operations is higher, but it is not necessary to perform interpolations or to preprocess the sensor information. Among these methods, we highlight the interpolation of four connected points performed by Nashashibi *et al.* [25] and the use of vision masks in a neighborhood of a point [26]. The last method has been chosen to evaluate the slope in each point in this paper. Other authors, namely Goldberg *et al.* [27], model the workspace as a regularly spaced cell grid. In each cell, data from different traversability tests are stored. The traversability tests are integrated in a software for planetary exploration. The time computing spent, in order to perform all the traversability tests, makes the model not be of interest for our purposes.

On the other hand, roughness is treated as a measurement of the surface variation. Most of the authors evaluate roughness as the height variation. For example, Nashashibi *et al.* [25] and Langer *et al.* [21] evaluate the discontinuities with the gradient calculation and the sorted data provided by a range finder.

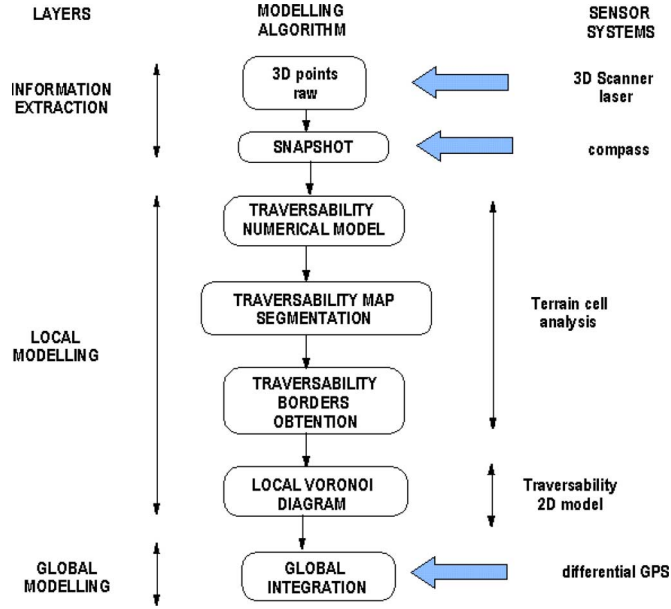


Fig. 1. Modeling system scheme.

Other authors calculate the dispersion points with respect to an interpolated plane [28]. For Seraji [29], the roughness depends on the number of rocks in the environment, and the parameter is computed using artificial vision techniques and fuzzy rules.

In this paper, the traversability numerical model (TNM) will be built in a low level. The terrain will be divided into crossable and noncrossable areas for the robot. We must take into consideration that the traversability characteristic not only depends on the terrain properties but also on the robot's physical restrictions and the robot task.

III. MODELING SYSTEM: GENERAL STRUCTURE

Fig. 1 presents the robot-oriented philosophy to obtain a map in order to navigate in different environment types. The philosophy divides the problem into several modeling levels, which allows to obtain intermediate models which can be useful for other robot tasks. The aim is to build a model to represent wide areas with the minimum information needed for navigating. The model proposed is compact and represents the areas that the robot can cross. The layers are described as follows.

- 1) In the first layer, which is called information extraction layer, the 3-D information is captured, in this case, by a 3-D scanner laser and is computed with the aid of a compass in a snapshot representation. In this layer, algorithms used to select the coordinate reference systems are developed. This information is useful for 3-D objects modeling needed for manipulation tasks.
- 2) In the local modeling layer, the snapshot model is enhanced with the terrain cell analysis. The 3-D data are classified depending on the traversability characteristic. The model obtained will be called TNM, which will be defined in Section IV. The TNM divides the 3-D environment into regions that can be crossable or noncrossable by the robot. To determine the traversability property, both the terrain's and robot's characteristics are taken

into account. Inside this layer, the 3-D measurements are projected on an \mathcal{XY} plane, and they are discretized in a DEM to obtain a qualitative description of each cell. In addition, the borders of the nontraversable regions are extracted to build the traversability 2-D model called TRM. The TRM is representative by definition because it is a representation of the environment's maximum security path. Obstacles are represented as nontraversable areas (NTAs), and they are not considered for the map representation.

- 3) At last, in the global modeling layer and owing to the global information provided by a differential GPS, an integration with the previous local TRM, which is obtained in the successive perceptions during an exploration task, can be done. This model is compact and can be computed in real time while the robot is moving.

IV. TRAVERSABILITY ANALYSIS

In outdoor navigation, it is interesting to estimate some parameters that define the difficulty of the terrain to be crossed. Path planning in large and outdoor environments is a complex task because there are a lot of parameters that define the traversability, for example, as follows.

- 1) The task the robot is going to perform in the environment. Manipulation, planetary exploration, etc., need different models and information. In this application, the robot's task is to build a sensor-based model to move in a large outdoor environment, *a priori* unknown. The chosen model (TRM) is of the topogeometric type.
- 2) The experimental platform characteristics, its locomotion system, and physical restrictions affect future decision taking. In this case, the locomotion system, which shows the maximum height that the robot can go through in its movement, and the robot size, which allows determination of the minimum elevation that the robot can go under (the 3-D free space), have been chosen as important robot characteristics.
- 3) The terrain characteristics, which determine the difficulty of the terrain to be crossed. Most authors focused their research on two basic parameters to specify the terrain characteristics: the terrain slope and the roughness degree. The two parameters are used not only in robotics but also in the topography field and, specifically, in geographical information systems, but in another level and with different sensor systems. Both parameters (slope and roughness) will be used in this paper to define the traversable zone for the robot and for an outdoor environment that can be nonstructured.

A. Terrain Slope Analysis

The surface inclination or slope can be defined as the existing angle between the surface normal vector \vec{N} and the vector \vec{W}_π , perpendicular to the horizontal surface, as shown in Fig. 2

$$\xi = \arccos \frac{\vec{N} \cdot \vec{W}_\pi}{|\vec{N}| \cdot |\vec{W}_\pi|}. \quad (1)$$

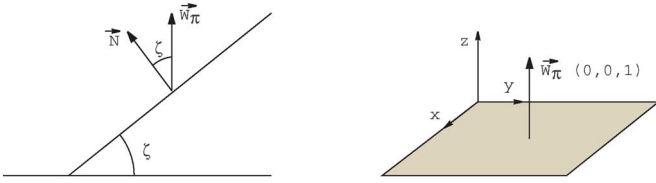
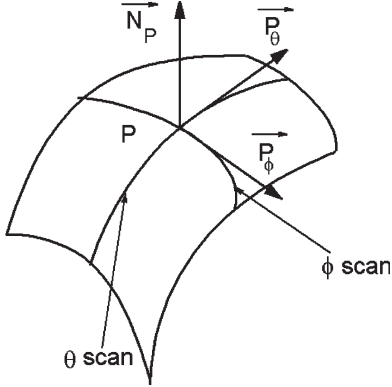


Fig. 2. Terrain slope definition.

Fig. 3. Normal in the point P estimation.

The algorithm used to calculate the slope in each point was developed by Betgé-Brezetz in [26]. It consists of applying, over a sorted data image, a Sobel filter in the horizontal and vertical directions to obtain the tangents in each point along the two scanning directions, which are defined in Fig. 3 as θ and ϕ scans.

Each 3-D point (P) is represented in a Cartesian reference system

$$P = \begin{pmatrix} x(\rho, \theta, \phi) \\ y(\rho, \theta, \phi) \\ z(\rho, \theta, \phi) \end{pmatrix}. \quad (2)$$

The slope in each point is calculated based on the previous normal vector calculation, which is shown in Fig. 3 as \vec{N}_P .

The lines ϕ and θ scans are defined as the horizontal and vertical scans, respectively. In addition, the 3-D point image is defined as $I(\theta, \phi)$, such that the depth ρ is the value in each pixel (I). In Fig. 3, the normal vector is observed to be the normalized cross product of the tangent vectors to the lines ϕ and θ (called P_ϕ and P_θ). That is to say

$$\vec{N}_P(\theta, \phi) = \frac{\vec{P}_\theta \wedge \vec{P}_\phi}{\|\vec{P}_\theta \wedge \vec{P}_\phi\|} \quad (3)$$

with

$$\vec{P}_\theta = \begin{pmatrix} \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial \theta} \\ \frac{\partial z}{\partial \theta} \end{pmatrix} \quad \vec{P}_\phi = \begin{pmatrix} \frac{\partial x}{\partial \phi} \\ \frac{\partial y}{\partial \phi} \\ \frac{\partial z}{\partial \phi} \end{pmatrix}. \quad (4)$$

The surface orientation depends on the cross product sign; the orientation is chosen such that \vec{N}_P goes always in the sensor direction.

Mathematically, the tangent vector is obtained when the curve in the point is derived. If a discretization is done, and

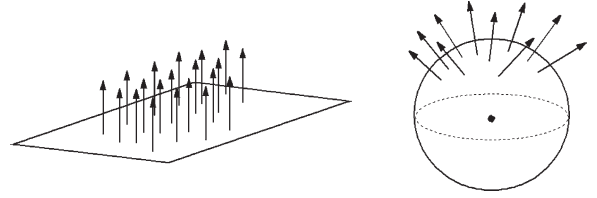


Fig. 4. Spherical variance analysis.

we apply digital image-processing techniques, the continuous derivatives that appear in P_θ and P_ϕ are calculated by multiplying a mask and by adding the elements.

This method gives us the possibility of calculating the normal vector in each 3-D point in a fast and easy way. It is suitable in unknown outdoor and nonstructured environments, because the operations are performed in a local neighborhood. Furthermore, despite the fact that the slope values are not exact, because of the filter used, its accuracy is sufficient in solving the navigation problem, as demonstrated by the experimental results in Section VII.

Due to the robot's physical restrictions, basically the robot mass and its locomotion system, maximum and minimum slopes that the robot can go up and down exist. With the robot information, the traversable slope analysis is performed as follows.

Definition 1: Given a 3-D point image $I = \{p_i(x_i, y_i, z_i)\}$, the traversable area (TA) is defined as the subset of 3-D points, $p_i \in I$, such that the slope in each point ξ_i fulfills the following property:

$$\text{TA}^{\text{slope}} = \{p_i(x_i, y_i, z_i) / \xi_{\min} < \xi_i < \xi_{\max}\}. \quad (5)$$

That is to say, the slope analysis considers as NTAs those regions where the slope in each 3-D point exceeds two thresholds.

This analysis is useful in flat terrains with different slopes, but the method presents some problems with surfaces such as steps (nontraversable by the experimental platform, but considered as traversable by this method) or little rocks (easy to cross for the robot locomotion system, but considered as obstacles by the method). In that case, it will be necessary to study another parameter to complete the traversability study.

B. Roughness Analysis

Most of the definitions found in the literature about roughness make a reference to irregularity measurements in the terrain surface, and they usually refer to elevation irregularities.

The method used in this paper to calculate the roughness degree is not a classical method in the robotics field, rather, it is based on the normal vector deviation in each point, with the calculus of a statistic called spherical variance [30].

In this case, we take advantage of the previous normal vector calculation to evaluate the variation inside a local region.

The spherical variance is obtained from the orientation variation of the normal vector in each point. The study uses the following reasoning.

- 1) In a uniform terrain (low roughness), the normal vectors in a surface will be approximately parallel, and for this reason, they will present a low dispersion (see Fig. 4 left).

- 2) On the other hand, in an uneven terrain, the different changes in the orientation of normal vectors will present great dispersion (see Fig. 4 right).

Definition 2—Spherical Variance: Given a set of vectors $\{\vec{N}_i\}$ corresponding to the normal vectors in a neighborhood inside the perceived space, the spherical variance ω is defined as complementary to the normalized mean vectors module

$$\omega = 1 - \frac{R}{n}. \quad (6)$$

Where ω is a measure of the vectors' dispersion. The vectors are defined by their module and direction in the 3-D space. The method used to obtain the parameter is detailed below.

- 1) Given a set of n normal vectors to a surface, defined by their three components $\vec{N}_i = (x_i, y_i, z_i)$, the module of the sum vector R is calculated as

$$R = \sqrt{\left(\sum_{i=1}^n x_i\right)^2 + \left(\sum_{i=1}^n y_i\right)^2 + \left(\sum_{i=1}^n z_i\right)^2}. \quad (7)$$

- 2) The mean value is normalized by dividing between the number of data n . In this way, the resulting value is between the range $[0, 1]$

$$\frac{R}{n} \in [0, 1]. \quad (8)$$

- 3) Finally, the complementary of the result is calculated to give sense to the statistic in (6).

Therefore, the values are standardized, and they are distributed in a theoretical range between 0 and 1. When $\omega = 1$, there exists a maximum dispersion that can be considered as the maximum roughness degree, and when $\omega = 0$, a full alignment exists, and the terrain will be completely flat.

The spherical variance has not been used before in the robotics field. It is of great interest to consider the normal vector variation as a roughness measurement, instead of the elevation gradient, as different authors have done [3], [25]. A comparative study between the spherical variance and gradient elevation techniques can be seen in [31]. The advantage of our method over traditional ones can be found, above all, in the 3-D points sensed by the scanner laser, where the density information decreases with distance. An example of this can be clearly found in those environments with a negative slope [32], where the scanner does not sense too much information.

Therefore, the 3-D traversability model is defined as follows:

$$\text{TA}^{\text{roughness+slope}} = \{P_i(x, y, z) \in \text{TA}^{\text{slope}} / \omega < \omega_{\text{th}}\}. \quad (9)$$

The terrain will be considered as nontraversable, due to its roughness, when the spherical variance in a point exceeds a threshold (ω_{th}) depending on the robot's locomotion system. Therefore, "little rocks," which are considered in the slope analysis as nontraversable, will now be considered as crossable by the robot due to their low dispersion, and in the "steps" case, the high dispersion, when the slope changes from 0° to 90° , let them to be considered obstacles.

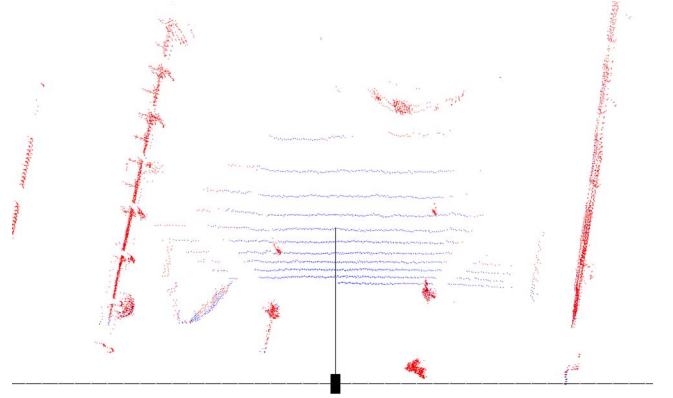


Fig. 5. TNM. Top view.

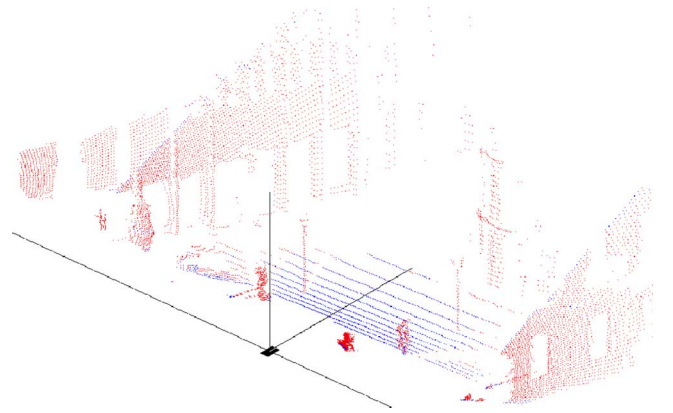


Fig. 6. TNM. Side view.

V. TRAVERSABLE REGION MODEL (TRM)

When the 3-D model is defined (as the one in Figs. 5 and 6, where the terrain considered TA is represented in blue and the NTA is represented in red), the free space can be extracted to build a TRM for the robot navigation and path planning. For the regions' representation, a VD technique has been chosen.

The TNM provides a geometrical information that can be useful for object modeling or virtual model construction. Moreover, the bulky information provided by the sensor system makes the model difficult to manipulate because of the time-computing cost. Previous to obtaining the TRM, a 3-D information must be reduced and simplified. The information needed for the model constructor is the 2-D coordinates of the borders that separate the TA from the NTA. To achieve this goal, digital image-processing techniques have been chosen.

In this paper, we define the image as a 2-D matrix where each pixel is defined by its traversability characteristic. The 3-D image is, in this way, transformed into a binary image of traversable and nontraversable regions. The image defines the robot's free space.

To achieve the model, it is necessary to follow these steps.

- 1) DEM calculation: DEM allows the 3-D information of the TNM to be arranged in cells, in the \mathcal{XY} plane, storing in each cell only the information necessary to represent a navigation model (see Fig. 7). The parameters necessary for the environment model construction are the number (position) of cell, traversability information, the number

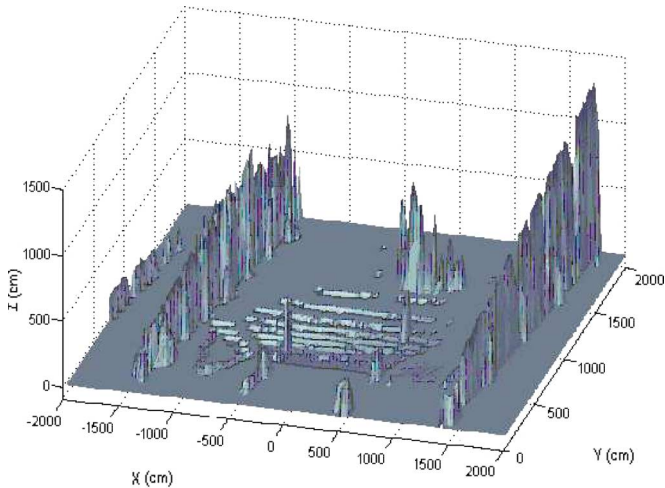


Fig. 7. DEM.

of obstacles in a cell, and the elevation measure. The traversability information and the elevation measure will depend on the number of obstacles in the cell and on their position. The sensor will provide an information about the land, the objects that exist over the land, and the overhanging objects. To obtain the number of objects, the chain map algorithm is applied [33]. In a cell, we said that only one object exists when all the points of the cell are continuously distributed. In the case where more than one object exist, the continuity will be broken. Therefore, the information from each cell is calculated as follows.

- a) The chain map algorithm is applied, determining the number of objects presented in the cell, which, in this case, will be one or two.
 - b) The elevation value is assigned. If only one object has been detected, the elevation corresponds to the maximum height, meaning the z coordinate of the last point in the chain. If two objects have been detected, the robot can navigate between both, and therefore, the highest object is rejected, and the maximum elevation value of the first object (the closer object to the soil) is assigned to the cell.
 - c) Finally, the value of traversability, which corresponds to the considered object in each case, is added to the cell.
- 2) Visibility map (VM) calculation: With the DEM, only those cells that the sensor system perceives can be labeled. Then, there are some cells in the discretized environment without information, because the sensor maximum range has been overloaded or because another cell is occluding it. For this reason, the VM is built (see Fig. 8). To obtain the information in all the cells, an elevation and the traversability estimation are assigned to the occluded cell. The estimation is performed with the extrapolation of the 3-D segment that joins the sensor with the center of the visible cell. In the occluded cells, the traversability value is assigned by means of the elevation and the evaluation of its neighbor cell's information.
 - 3) Binary map obtention: With the workspace divided into cells with the traversability characteristic, the image is

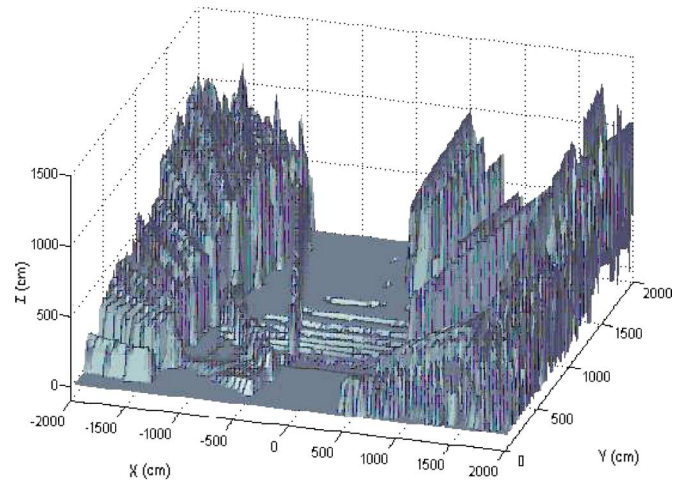


Fig. 8. Visibility map.

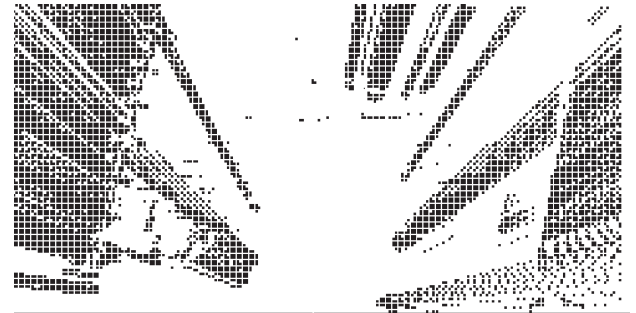


Fig. 9. Binary image.

transformed into a binary matrix (see Fig. 9) that represents the traversable (0) or nontraversable regions (1). The image is filtered with the classical morphological operations (closing operation) to smooth the shapes without losing the geometrical information (see Fig. 10). The closing operation consists of performing a dilatation and a later erosion. Upon having performed first the dilatation operation, we ensure that small obstacles are not lost, and upon performing the erosion operation, we achieve a smoothing of the shapes, but without losing the real dimensions. This preprocessing technique enhances the subsequent edge detection. The boundaries that separate the TAs from the nontraversable ones are needed to build the TRM. In this case, different filters have been tested, giving good results the Prewitt, Robert, and Isotropic filters. These filters are developed as a matrix with a dimension of 3×3 , and a convolution is performed with the binary image (the result is shown in Fig. 11).

In previous steps, the information needed to represent the TRM is obtained.

A. Traversable Region Modeling Algorithm

For region representation, a VD technique has been chosen. This approach is used for the following reasons.

- 1) The model is built based on the sensor information, its construction is fast, and the model can be used in real time.



Fig. 10. Morphological operation. Closing.

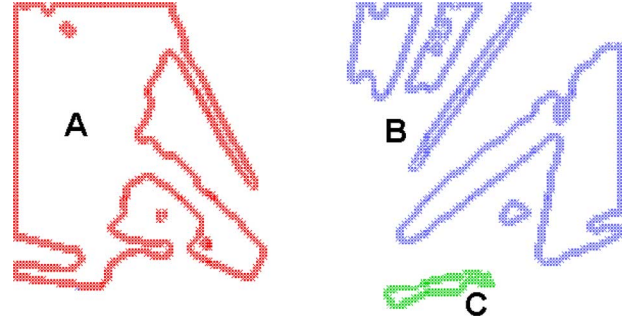


Fig. 12. Data grouping to obtain generators.



Fig. 11. Edge detection.



Fig. 13. Real image.

- 2) The VD, by definition, gives safe trajectories that maximize the distance between the robot and the obstacles. The model obtained is very useful for navigation.
- 3) The environment model is either a simplification of the robot's free space or it represents a free-space scheme. It does not imply memory-storing problems.

The VD divides the free space into a set of different regions called Voronoi regions, each of them formed by the set of points that is placed closer to an object than the rest [34]. This roadmap algorithm is generally used for the topological map construction because it uses a segmentation region. The use of this technique in the robotics field began in the early 1980s in the 2-D path planning [35], first with *a priori* established models, and then, in the 1990s, directly from sensor information, and this technique has been used in robot navigation and planning tasks [34].

In this paper, we take advantage of the digital image obtained in the previous steps to work with the algorithm proposed above, where the input is an image with the borders between the traversable and nontraversable zones. The visibility model (VM) information determines which cells belong to the free space and which ones do not.

The workspace \mathcal{W} is represented as a 2-D binary image $B(i, j)$, where each position (i, j) is assigned a field value (0 or 1) that indicates if a pixel belongs to a traversable region (field 1) or not (field 0). For each cell belonging to the free space, $(i, j) \in \mathcal{FS}$ are, at least, one point closer to the occupied space (\mathcal{FS}).

The LVD is obtained from the distances to the generated points which belong to the objects' borders. To obtain the local model, the steps are presented in the following sections.

1) *Clustering*: The cluster determination, in the 3-D information case, is not trivial. The sorted 2-D information from a scanner laser is easily separated in clusters, using the distances between scanned points [17], [36]. In this paper, the 3-D information cannot be dealt with as in the case of other authors, and a labeling technique is used to obtain the clusters. The kernel applied is a circumference. The radius is the robot's size. If there is a distance greater than the robot's size between two points which belong to the nontraversable region border, then the robot can cross between them, and the points will be considered as belonging to different clusters.

In Fig. 12, the data grouping of the real environment presented in Fig. 13 is shown. The result of this step is three clusters called A, B, and C.

2) *Distance to the Cluster Computing*: For each free-space cell $(i, j) \in \mathcal{FS}$, that is to say, the cells where the robot can move, the distance between the cell and each point, which belongs to each cluster, is calculated. The minimum Euclidean distance between the cell and the points to one cluster will be considered for the VD construction.

3) *LVD Obtention*: The label cell is evaluated based on the minimum distance as follows (see Fig. 15).

- 1) If the distance to the A cluster is less than the rest, the cell is evaluated as belonging to the Voronoi region associated to cluster A.
- 2) If there are two equidistant clusters in a cell (for example, A and B), then the cell is labeled as Voronoi edge.
- 3) For greater equidistances, it will be labeled as Voronoi node.

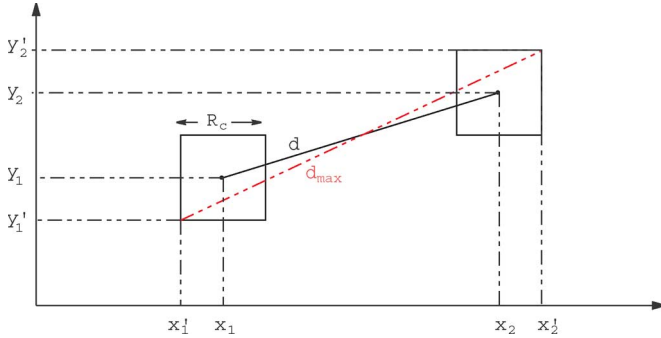


Fig. 14. Maximum distance between two cells.

4) *Error Caused by Discretizing*: The labeling of each cell belonging to the \mathcal{FS} is carried out by computing the distance between the center of the cells. Nevertheless, the real measurement can be located in any position inside the cell. This must be taken into account when the distance to the generators is calculated. The maximum discretized error is calculated based on Fig. 14

$$d_{\max} = \sqrt{(x_2' - x_1')^2 + (y_2' - y_1')^2}. \quad (10)$$

In the same Fig. 14, we can see that

$$\begin{aligned} x_1' &= x_1 - \frac{R_c}{2} & x_2' &= x_2 + \frac{R_c}{2} \\ y_1' &= y_1 - \frac{R_c}{2} & y_2' &= y_2 + \frac{R_c}{2} \end{aligned} \quad (11)$$

and by replacing (11) in (10), we obtain

$$d_{\max} = \sqrt{(x_2 - x_1 + R_c)^2 + (y_2 - y_1 + R_c)^2}. \quad (12)$$

Therefore, d_{\max} is the maximum distance that we must consider when the labeling step is performed. That is to say, the maximum error E when two cells are evaluated is

$$E = d_{\max} - d. \quad (13)$$

In addition, the LVD obtention step is modified, for each cell (i, j) belonging to the free space, as follows.

- 1) The distance to all the points belonging to each generator is computed.
- 2) The minimum distance to each generator is obtained. We consider, for example, in Fig. 12, the distance difference between two generators A and B as $E_c(A, B) = \mathbf{d}_{\mathbf{E}}(g_A, p) - \mathbf{d}_{\mathbf{E}}(g_B, p)$, where p is the center of the cell (i, j) with coordinates (x, y) .
 - a) If $E_c(A, B) > 2 \times E$ for all $A \neq B$, the cell is considered as belonging to the Voronoi region associated to the object A .
 - b) If $E_c(A, B) < 2 \times E$ y $E_c(H, B) > 2 \times E$ for all $H \neq A \neq B$, the cell is considered as Voronoi edge between two objects A and B .
 - c) For greater equidistances, the cell will be considered as Voronoi node.

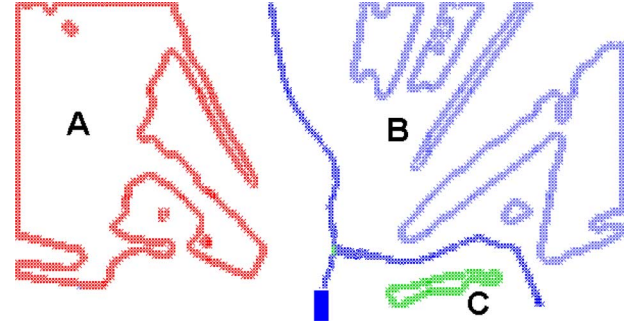


Fig. 15. LVD representation.

In Fig. 15, the result of the LVD computing from Fig. 12 is presented. In the figure, the Voronoi edges and nodes are highlighted.

VI. INCREMENTAL GLOBAL MAP

To explore, with autonomy skill, wide unknown outdoor areas, the robot needs to build a global model to know where it is at each instant and where it can go. The model can be introduced *a priori* in the robot, or it can be built while the robot is traveling, fusing local models. Davison and Kita [37] distinguish two different methodologies in order to build a global map: Batch and sequential methodologies. In the first case, the model is built based on the complete information obtained by the sensors of the robot during an exploration task. The method builds the model offline, and it is not possible to update the model with new data. On the other hand, sequential method offers an update representation of the environmental state, where the size does not change in time. The model is updated when new data are provided, but sensors' uncertainty must be carefully taken into account when the model is built.

In this paper, the methodology developed is near to sequential one; nevertheless, in our case, the environment size is changing during the robot exploration. Therefore, the model is obtained by incremental fusion. The incremental global model will be built by assembling a set of local data obtained in different acquisitions in a unique model.

To obtain an incremental model, which is based on the robot successive perceptions, the following steps must be carried out [38].

- Step 1) Environment information sensing: The robot, stopped, uses its external sensors (in this case, a 3-D scanner laser and a compass) to acquire all the needed information and to build a local model (LVD).
- Step 2) The local model (LVD), which is useful for navigation, is transformed in a global coordinate system, with a differential GPS, and the local model is integrated with the global current model.
- Step 3) The robot moves to the next position by path planning or guiding, and goes back to Step 1).

The exact global model construction, which is based on the successive perceptions, is, in general, difficult to obtain due to the sensor uncertainty. Moreover, in the majority of the cases, it is not interesting not only for the imprecision but also for the

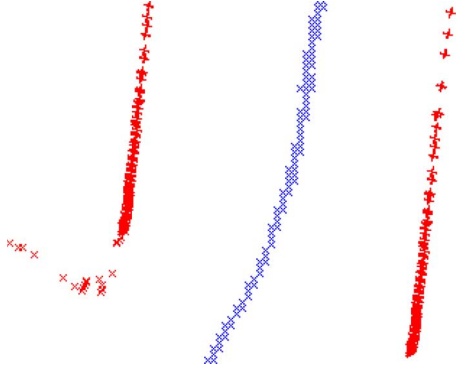


Fig. 16. LVD.

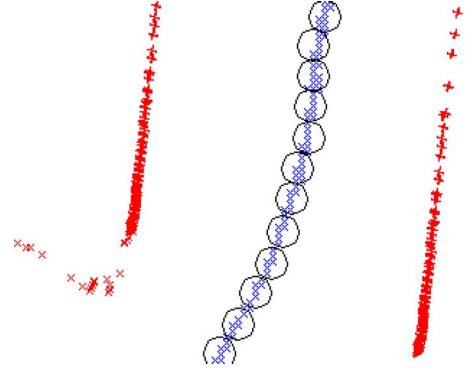


Fig. 17. Data map clustering.

CPU time computing, and the high memory storing makes the model not interesting for real-time objectives.

For the integration, a transformation in the global Cartesian system is needed. A GPS is used to perform the transformation for each LVD point (x_i, y_i) . The algorithm is detailed as follows:

- 1) For the first local map (transformed into global reference system, as shown in Fig. 16):
 - a) To obtain a seed point $P_0(x_0, y_0) \in \text{LVD}$ and to set it as the first center of mass ($\text{CM}_0 = P_0$).
 - b) Calculate the Euclidean distance between all the points $P_i(x_i, y_i) \in \text{LVD}$ and the CM_0 as (15).
 - i) If the distance r_i is less than an error value called r_{\max} , the point P_i is included in the list belonging to the CM_0 , and a new CM'_0 is recalculated with a set of points.
 - ii) If the distance is greater than r_{\max} , then a new list is formed, and it is set as another center of mass $\text{CM}_1 = P_i$. This operation is shown in Fig. 17.
 - c) To finish, in order to generate the global model information, the center-of-mass coordinates and the distance to the clusters of the TRM are stored (see Fig. 18).
- 2) For the rest of the LVD maps (transformed into global reference system):
 - a) For each point $P_i \in \text{LVD}$, we try to find the CM in the global map to which P_i belongs (the Euclidean distance between P_i and CM is less than r_{\max}).
 - b) If we do not find it, a new list is formed with $\text{CM} = P_i$.

Equation (14) calculates the CM for a set of N points. The Euclidean distance is calculated in (15) to know if a point belongs to the CM

$$\text{CM} = \left(\frac{\sum_1^N x_i}{N}, \frac{\sum_1^N y_i}{N} \right) \quad (14)$$

$$r_i = \sqrt{(x_i - x_{\text{CM}})^2 + (y_i - y_{\text{CM}})^2}. \quad (15)$$

The r_{\max} value must be chosen considering the map cell resolution, and it always depends on the robot's velocity movement.

In Figs. 16–18, the steps performed to build the global map are presented.

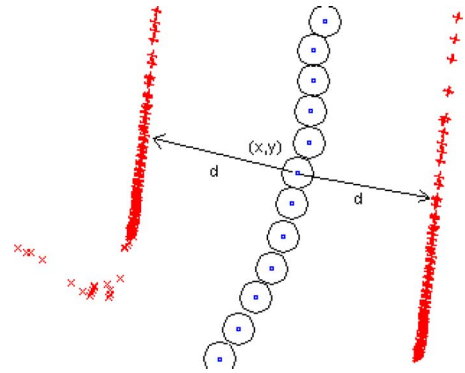


Fig. 18. Incremental Voronoi map.

VII. EXPERIMENTAL RESULTS

In this section, we present the experimental results carried out in three different real outdoor scenes. The first scene (shown in Fig. 20) is a partially structured environment (on the right-hand side) with a rough terrain (on the left) with an aerial obstacle. The second scene represents a terrain with different slopes (see Fig. 24). Finally, the third scene represents a rough terrain with NTA for GOLIAT (Fig. 29).

As experimental results for each scene, we present the following.

- 1) The TNM obtained, where those points that belong to the NTA are represented in bold (or in red).
- 2) The VM obtained, which is based on the TNM. VM will be presented to better understand the image.
- 3) The LVD as the TRM representation.

The models built are adapted not only to the terrain type and the navigation task but also to a long-size intelligent autonomous robot called GOLIAT (see Fig. 19). GOLIAT is a 1.5-t diesel crawler robot with caterpillars. The devices, which give the robot the autonomous intelligent characteristic (such as sensors and computers dedicated to perception, planning, and movement control), are placed onboard. The robot senses the environment, owing to a 3-D scanner laser assembled in the Carlos III Robotics Lab [39], which provides a 3-D information in horizontal and vertical scans. These 3-D points are filtered and transformed into a traversability information for the roadmap constructor.



Fig. 19. Outdoor robot GOLIAT.

TABLE I
PHYSICAL PARAMETERS OF GOLIAT

Maximum Robot height (including aerial DGPS)	H_{robot}	219.50 cm
3D scanner laser distance to base reference system	H_L	152.30 cm
Robot width	D_{robot}	188.50 cm
Maximum height the locomotion system can go through	U_{min}	15.00 cm
Maximum slope to climb	ξ_{max}	+54°
Minimum negative slope to descend	ξ_{min}	-49°
Maximum roughness degree	ω_{th}	0.2



Fig. 20. Environment with aerial obstacles. Real image.

The parameters associated with the outdoor platform that is needed to represent the traversability of GOLIAT are shown in Table I.

These parameters were obtained from the physical robot characteristics. Roughness threshold is an experimental value obtained from the experience of the robot fieldwork.

A. First Test Scene: Aerial Obstacles Environment

In Fig. 20, a partially structured environment with aerial obstacles is shown. The environment presents a roughness degree which is low and accessible for GOLIAT. In Fig. 21, the TNM is presented. The TA is represented in red, and the NTA is represented in blue. The aerial obstacle is evaluated as belonging to the NTA. Nevertheless, the aerial beam is placed

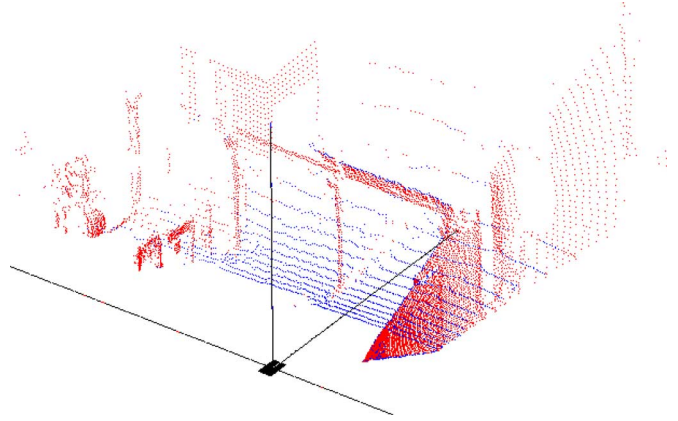


Fig. 21. Environment with aerial obstacles. TNM.

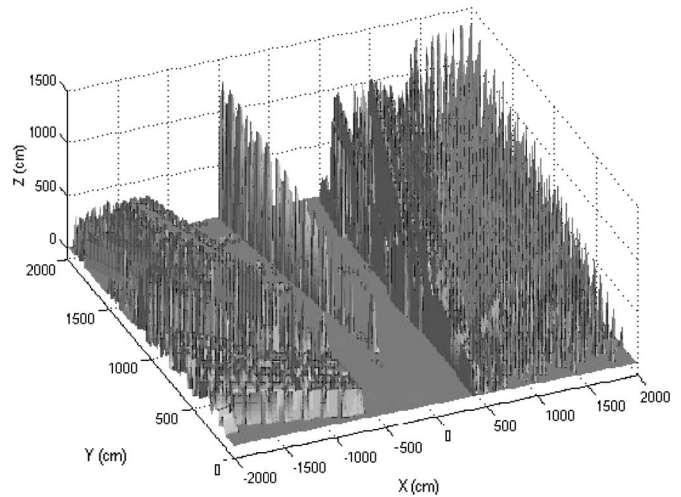


Fig. 22. Environment with aerial obstacles. Visibility map.

higher than the robot height. With the DEM construction, this obstacle is eliminated and considered as free space.

Starting from TNM, the 3-D points are sorted in a DEM and then completed with the VM calculation. The objective of this process is to obtain and separate the accessible regions and the nonaccessible ones. Fig. 22 shows the VM obtained when the numerical model, as shown in Fig. 21, is discretized. In this test, the cell resolution is 20 cm.

To finish, after the digital image treatment (closing and edge detection), a clustering and an LVD calculation are performed. In the LVD, the safest path for the robot is represented. The LVD stores the coordinates (x, y) of each point belonging to it, which is labeled as node (represented in Fig. 23 in green) or edge (represented in the same figure in blue), and gives an additional information to the path planner.

In Table II, the time computed in milliseconds is detached for each process. The experiments were carried out in a personal computer AMD-K6 3D processor, with 64-MB RAM and 450 MHz. The table notation is shown in Table III.

The experiment shown in Table II was carried out with a cell resolution of 20 cm. The images have a 100×200 size. To reduce the time computed in the LVD algorithm, we select, for each generator, a representative data set. With this, the distance calculation between each cell (belonging to the free space)

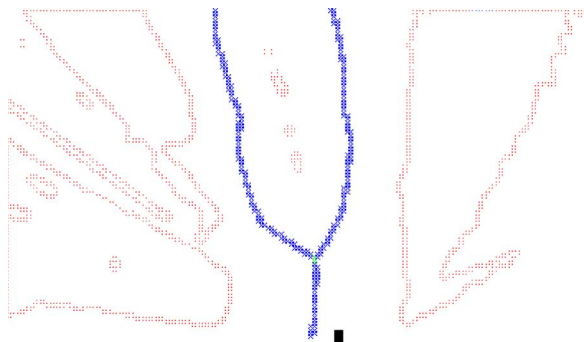


Fig. 23. First scene. LVD.

TABLE II
FIRST SCENE. PROCESSING TIME (IN MILLISECONDS)

TNM	DEM	VM	CLOS
694.06	277.97	258.29	123.11
EDG. D	CLUST	LVD	t_{TOTAL}
74.70	4 clusters 1011.22	1 nodes 1815.70	4834.90

TABLE III
RESULTS NOTATION

<i>DEM</i>	Digital Elevation Map.	<i>VM</i>	Visibility Map.
<i>CLOS</i>	Morphological operation (Closing).	<i>EDG. D</i>	Edge Detection.
<i>CLUST</i>	Clustering in generators.	<i>LVD</i>	Local Voronoi Diagram.
t_{TOTAL}	Processing time (in ms).		

and each point from the generator is reduced to the distance computed between each free-space cell and a set of points.

The time computed in process CLUST and LVD depends on the number of points belonging to each generator more than the number of generators. In particular, in the LVD process, it depends on the number of points belonging to the generators and the number of free-space cells.

B. Second Test Scene: Different Slopes

In Fig. 24, a semistructured environment is presented. The most relevant characteristic in the scene is the big difference in the slopes drawn in Fig. 24.

Figs. 25 and 26 represent the TNM top and side view. The scanner laser is able to sense enough information to obtain a dense environment representation with a range of 30 m. This can be seen in the trees and the building walls detection.

In the VM shown in Fig. 27, the regions perceived by the sensor system and the occluded cells are presented. The LVD obtained in the scene is shown in Fig. 28, and in Table IV, where the time computed in milliseconds is detached for each process. Notice that the time computed in the last two processes has decreased despite the fact that the number of generators is greater.

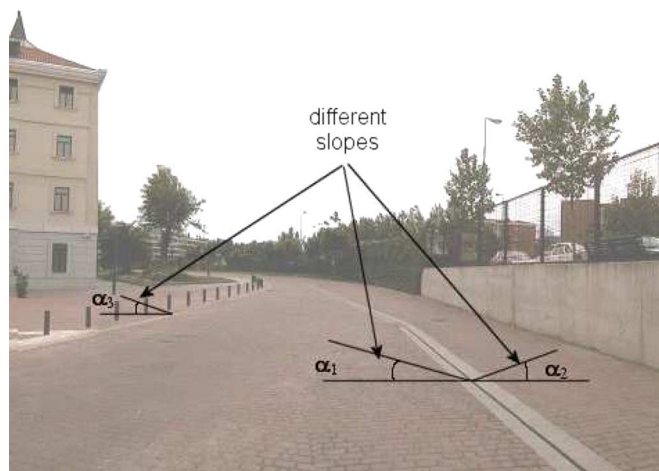


Fig. 24. Environment with different slopes. Real image.

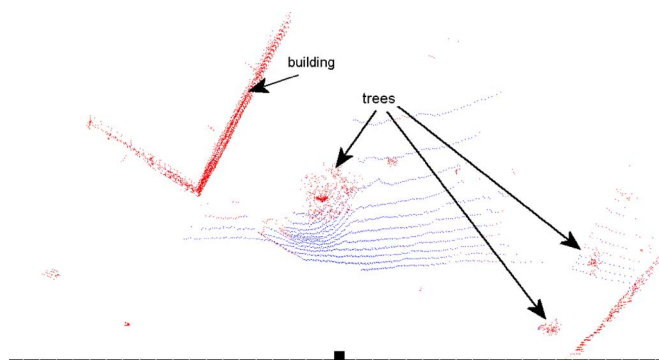


Fig. 25. Environment with different slopes. TNM. Top view.

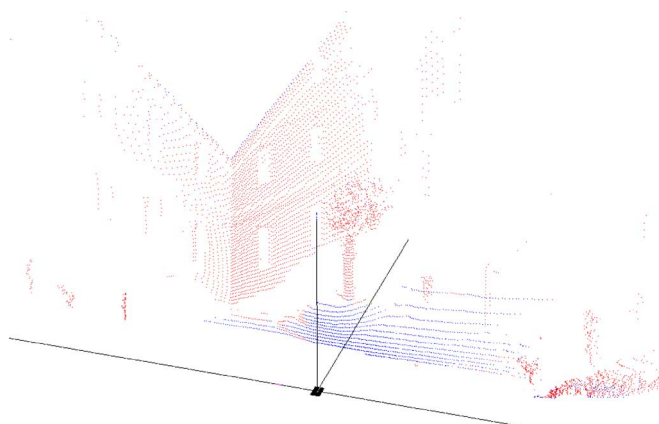


Fig. 26. Environment with different slopes. TNM. Side view.

C. Third Test Scene: Rough Terrain

In Fig. 29, a partially structured environment with a rough terrain is presented. The environment perceived by the scanner laser is transformed in the TNM using the topographical analysis explained above, and the result is shown in Figs. 30 and 31, where the TA is represented in red, and the NTA is represented in blue.

In these figures, the method classifies as nontraversable those regions with a high degree of roughness, and vertical walls. In this model, the iron beams have been detected as obstacles.

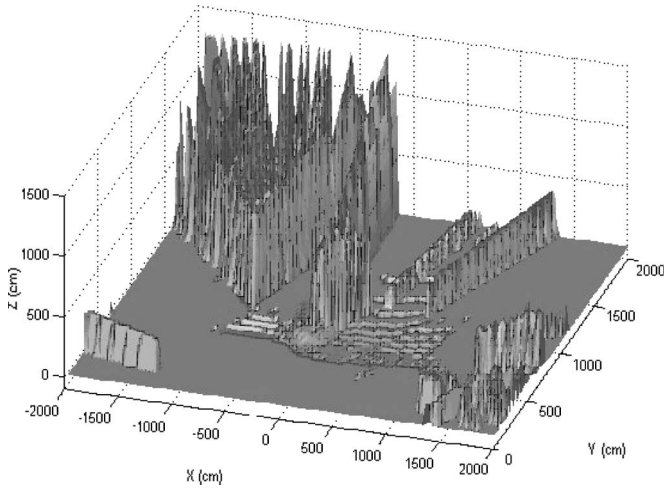


Fig. 27. Environment with different slopes. Visibility map.

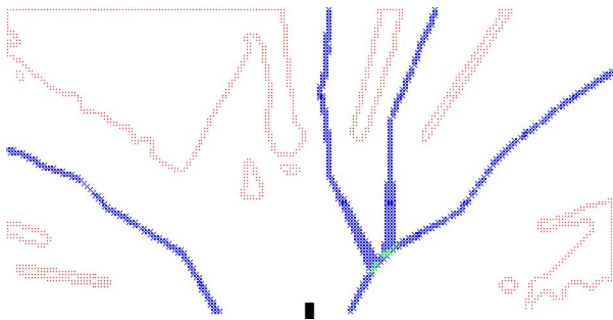


Fig. 28. Environment with different slopes. LVD.

TABLE IV
SECOND SCENE. PROCESSING TIME (IN MILLISECONDS)

TNM	DEM	VM	CLOS
685.77	353.49	58.19	123.48
EDG. D	CLUST	LVD	t_{TOTAL}
74.36	5 clusters 934.19	14 nodes 1674.96	4488.39



Fig. 29. Environment with rough terrain. Real image.

Fig. 32 shows the VM obtained when the numerical model is discretized. In this test, the cell resolution is 20 cm.

To finish, after the digital image treatment (closing and edge detection), a clustering and the LVD calculation are performed. In the LVD, the safest path for the robot is represented. The LVD stores the coordinates (x, y) of each point belonging to it, which is labeled as node (represented in Fig. 33 in green)

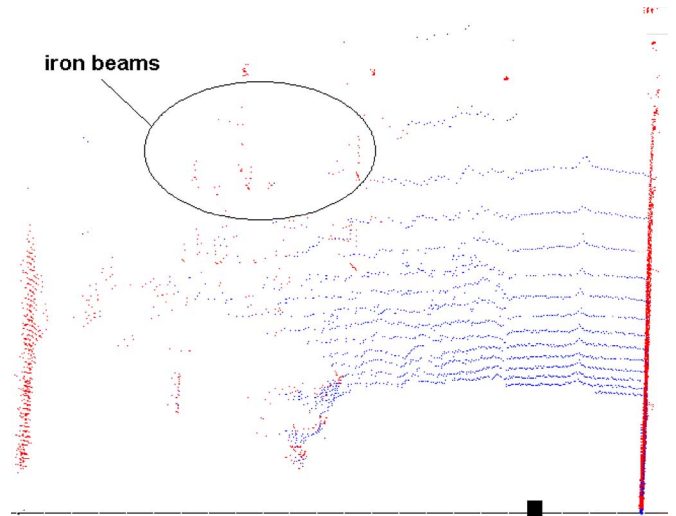


Fig. 30. TNM third scene. Side view.

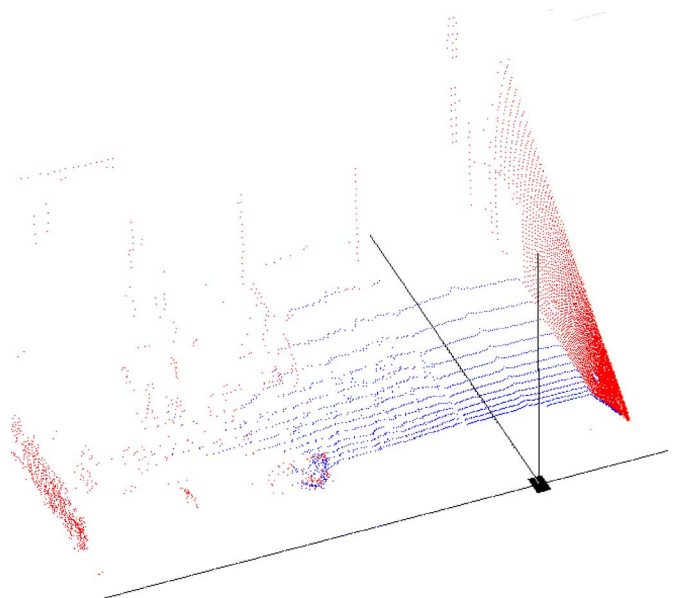


Fig. 31. TNM third scene. Top view.

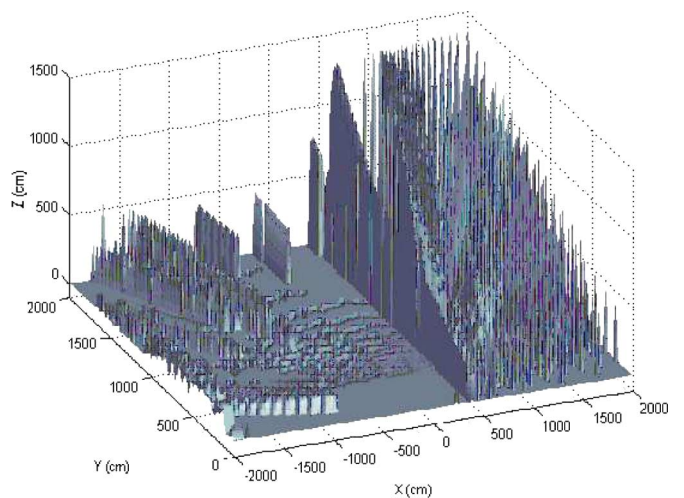


Fig. 32. Third scene. Visibility map.

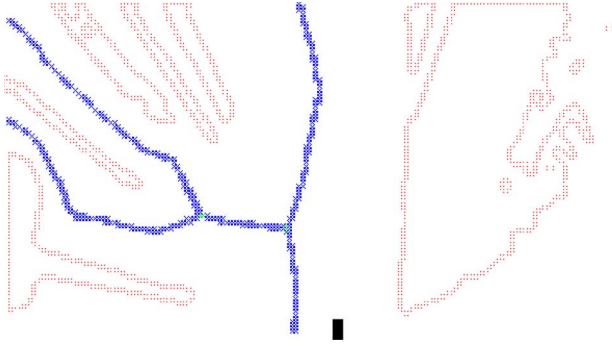


Fig. 33. Third scene. LVD.

TABLE V
THIRD SCENE. PROCESSING TIME (IN MILLISECONDS)

TNM	DEM	VM	CLOS
693.10	298.73	247.78	123.43
EDG. D	CLUST	LVD	t_{TOTAL}
74.80	4 clusters 1057.15	4 nodes 1707.91	4808.42

or edge (represented in the same figure in blue), and gives additional information to the path planner.

The experiments have been carried out in a personal computer AMD-K6 3-D processor, with 64-MB RAM and 450 MHz.

In Table V, the time computed in milliseconds is detached for each process. The table notation is the following.

The experiment shown in Table V was carried out with a cell resolution of 20 cm. The images have a 100×200 size. To reduce the time computed in the LVD algorithm, we select, for each generator, a representative data set. With this, the distance calculation between each cell (belonging to the free space) and each point from the generator is reduced to the distance computed between each free-space cell and a set of points.

The time computed in process CLUST and LVD depends more on the number of points belonging to each generator than the number of generators. In particular, in the LVD process, it depends on the number of points belonging to the generators and the number of free-space cells.

D. Cell Size Influence

A priori cell size influences time computing and model accuracy. To test the influence, two cell sizes (suitable for the environment and the robot size and for the robot speed that, for GOLIAT, is 11 km/h) have been chosen. The cell sizes are 20 and 50 cm. The experimental results have been carried out in an environment presented in Fig. 34, and the TNM is presented in Figs. 35 and 36.

The LVD obtained is presented in Fig. 37, where a 20-cm cell resolution has been implemented, and in Fig. 38, where a 50-cm cell resolution has also been implemented. Figs. 39 and 40 present the VM in order to compare the cell size.

As can be seen in the previous figures, the graphical result does not change; in other words, the LVD shape has not been



Fig. 34. Environment with high obstacle density.

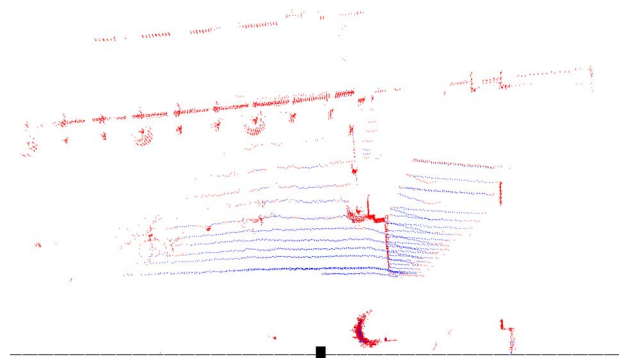


Fig. 35. TNM. Top view.

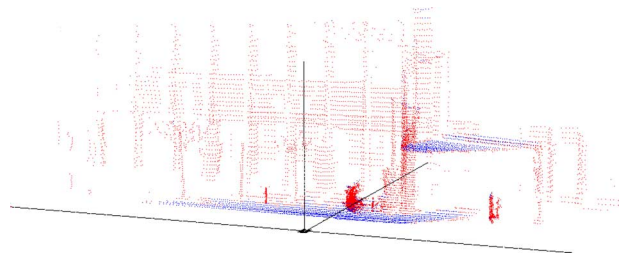


Fig. 36. TNM. Side view.

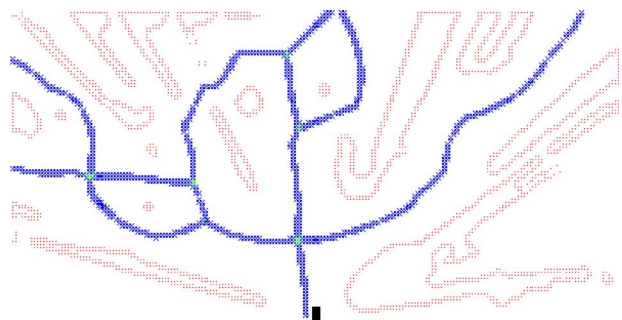


Fig. 37. LVD for 20-cm cell size.

lost, only the number of points has decreased. In the 50-cm cell size, the number of nodes decreases, and above all, the difference between the two resolutions, which S increases with

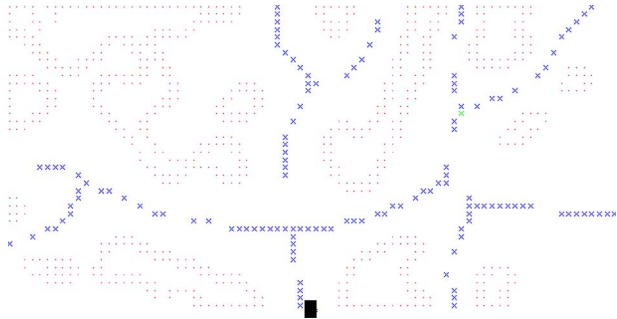


Fig. 38. LVD for 50-cm cell size.

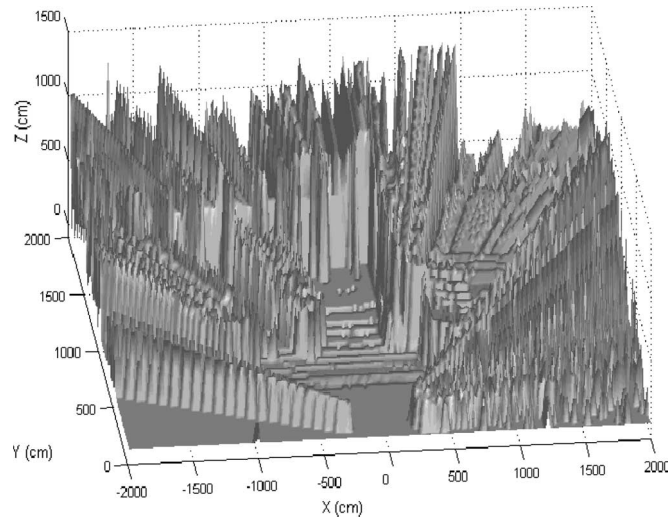


Fig. 39. Visibility map. Cell size of 20 cm.

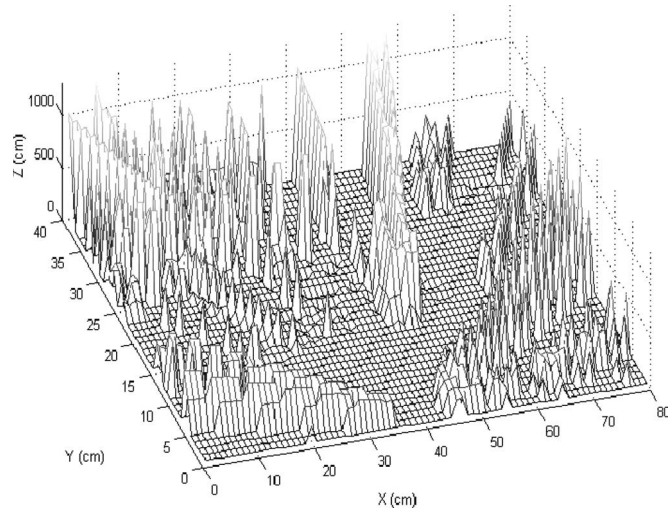


Fig. 40. Visibility map. Cell size of 50 cm.

the distance to the sensor system, can be appreciated. Of course, the processing time in steps with digital image-processing algorithms is reduced because of the increase in cell size. The image changes size from 100×200 to 40×80 (see Table VI).

TABLE VI
DIFFERENT CELL SIZE TIME-COMPUTED

	t for $R_c = 20$ cm in ms	t for $R_c = 50$ cm in ms
DEM	273.82	706.73
VM	116.72	42.72
CLOS.	127.90	14.18
EDG. d	77.15	19.14
CLUBS	(8 clusters) 1941.22	(9 clusters) 11.84
LVD	(15 nodes) 2261.88	(1 nodes) 99.34
TOTAL	4798.69	893.22



Fig. 41. Real environment.

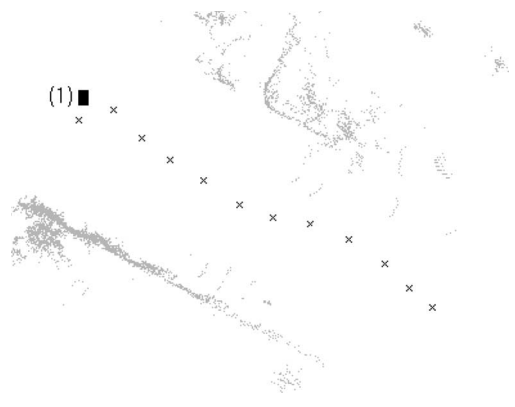


Fig. 42. Robot exploration sequence. Position 1.

In Table VI, the time computed for the 50-cm cell size is reduced to 20% of the time for the 20-cm cell size.

E. Incremental Maps

In the next figures (first, see Fig. 41), an exploration task for an outdoor robot is presented. The robot senses the environment and builds the LVD (Fig. 42); then, it moves 3 m ahead and repeats the process 11 times, building the incremental model (in Figs. 43–46, the most representative steps are presented).

VIII. CONCLUSION

Three-dimensional information has been used to apply a new methodology in modeling outdoor environments, which is

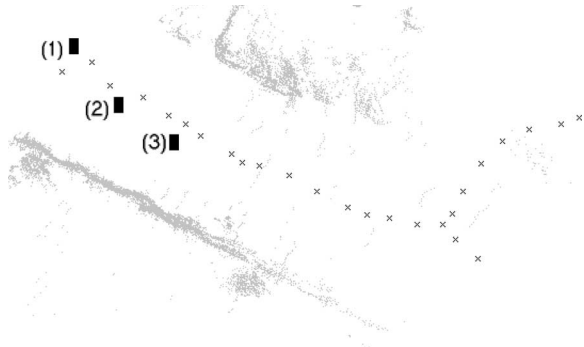


Fig. 43. Robot exploration sequence. Position 3.

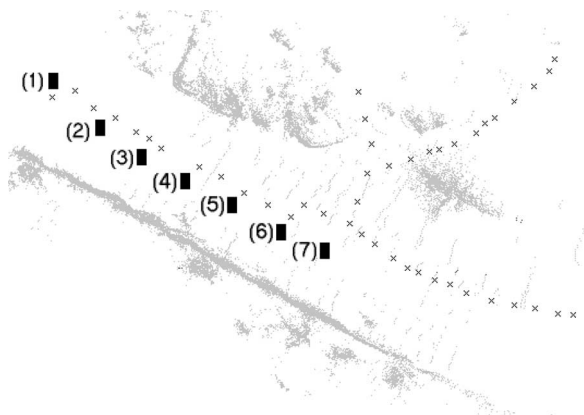


Fig. 44. Robot exploration sequence. Position 7.

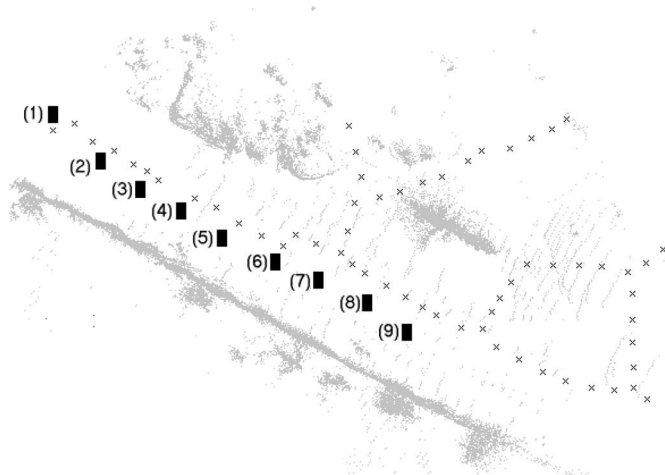


Fig. 45. Robot exploration sequence. Position 9.

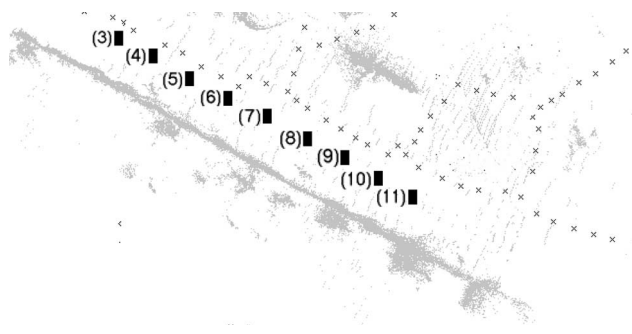


Fig. 46. Robot exploration sequence. Position 11.

based on a topographical analysis. The model can be built in real time while the robot is moving, and it is possible to carry out a local model integration in order to build an environment's knowledge database in a global map, using a GPS system onboard GOLIAT.

As conclusions based on the experimental results presented in this paper, we can highlight the following.

- 1) The algorithm developed is highly time-efficient. The time required to extract a 50-cm cell size local model (40×20 m) in a cluttered outdoor area is of 0.89 s in an AMD-K6 450-MHz processor. The time to integrate the local map in the global environment is less than 50 ms. This low computing time allows this solution to be used online.
- 2) The model is very easy to use in navigation because it detects noncrossable cells, and the local Voronoi maps provide safe local paths to avoid obstacles. The map can be directly used to obtain global paths by using the global Voronoi map.
- 3) The LVD process time increases when the number of free-space cells increases and when the number of points belonging to the generators increases.
- 4) The spherical variance has been proven to be a new and suitable alternative to obtain a roughness degree in large outdoor environments.
- 5) The 3-D scanner laser has proved to be a good choice to obtain information to model poorly structured outdoor environments. It senses the terrain surface with enough accuracy to obtain a 3-D dense map, which is then processed to obtain a qualitative description of each individual environment cell that is integrated in the local map.
- 6) Nevertheless, the sensor system presents problems in negative sloped terrains, where the information density is less when the distance between the measurements and the sensor increases. This is because of the scanner laser's nonlinearity when a vertical scan with a constant increment is done. Different solutions can be set out to solve this problem, such as the local model size reduction or the sensor placing (in a more elevated position) and the use of a nonconstant increment in the vertical scans to obtain more information in the slope area.
- 7) The cell size used in the model discretization has an influence on time computing. For a same size environment, the increase in the resolution cell will decrease the number of cells present on the map, and then, all the digital image algorithms will reduce the process time. Based on the experimental results, we have concluded that, for the robot size and the environment type, we are going to work with a good size cell that will be between 20 and 50 cm.
- 8) The incremental global Voronoi map of the outdoor environment is extremely compact, which allows to be used to model very wide outdoor areas. For the robot speed (GOLIAT speed is 11 km/h), the robot can compute the global map while it is moving.

REFERENCES

- [1] V. Fernández, C. Balaguer, D. Blanco, and M. A. Salichs, "Active human-mobile manipulator cooperation through intention recognition," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seoul, Korea, 2001, pp. 2668–2673.
- [2] M. Maurette, "Mars Rover autonomous navigation," *Auton. Robots*, vol. 14, no. 2/3, pp. 199–208, Mar.–May 2003.
- [3] D. Langer, J. K. Rosenblatt, and M. Hebert, "A behavior-based system for off-road navigation," *IEEE Trans. Robot. Autom.*, vol. 10, no. 6, pp. 776–782, Dec. 1994.
- [4] A. Yahja, S. Singh, and A. Stentz, "An efficient on-line path planner for outdoor mobile robots," *Robot. Auton. Syst.*, vol. 32, no. 2/3, pp. 129–143, Aug. 2000.
- [5] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," in *Proc. 6th Conf. Uncertainty AI*, 1990, pp. 60–70.
- [6] I. S. Kweon and T. Kanade, "High resolution terrain map from multiple sensor data," in *Proc. IEEE Int. Workshop Intell. Robots Syst.*, 1990, pp. 127–134.
- [7] M. Macri, S. De, and M. S. Shepard, "Hierarchical tree-based discretization for the method of finite spheres," *Comput. Struct.*, vol. 81, no. 8–11, pp. 789–803, May 2003.
- [8] J. Gomes-Mota and M. I. Ribeiro, "Mobile robot localization on reconstructed 3D models," *Robot. Auton. Syst.*, vol. 31, no. 1/2, pp. 17–30, Apr. 2000.
- [9] D. Hähnel, W. Burgard, and S. Thrun, "Learning compact 3-D models of indoor and outdoor environments with a mobile robot," *Robot. Auton. Syst.*, vol. 1, no. 44, pp. 15–27, 2003.
- [10] P. Ranganathan, J. B. Hayet, M. Devy, S. Hutchinson, and F. Lerasle, "Topological navigation and qualitative localization for indoor environment using multi-sensory perception," *Robot. Auton. Syst.*, vol. 41, no. 2/3, pp. 137–144, Nov. 2002.
- [11] U. Nehmzow and C. Owen, "Robot navigation in the real world: Experiments with Manchester's 42 in unmodified, large environments," *Robot. Auton. Syst.*, vol. 33, no. 4, pp. 223–242, Dec. 2000.
- [12] B. L. Boada, D. Palazon, D. Blanco, and L. Moreno, "Voronoi based place recognition using hidden Markov models," in *Proc. 15th IFAC World Congr. Int. Fed. Autom. Control*, 2002.
- [13] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artif. Intell.*, vol. 99, no. 1, pp. 21–71, Feb. 1998.
- [14] S. Betgé-Brezetz, R. Chatila, and M. Devi, "Natural scene understanding for mobile robot navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, vol. 1, pp. 730–736.
- [15] S. Simon and G. Dudeck, "A global topological map formed by local metric maps," in *Proc. Int. Conf. Intell. Robots Syst.*, Victoria, BC, Canada, 1998, pp. 1708–1714.
- [16] H. Choset, S. Walker, K. Eiamsa-Ard, and J. Burdick, "Sensor-based exploration: Incremental construction of the hierarchical generalized Voronoi graph," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 126–148, Feb. 2000.
- [17] D. Blanco, B. L. Boada, L. Moreno, and M. A. Salichs, "Local mapping from on-line laser Voronoi extraction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2000, pp. 103–108.
- [18] D. Blanco, B. L. Boada, C. Castejón, C. Balaguer, and L. E. Moreno, "Sensor-based path planning for a mobile manipulator guided by the humans," in *Proc. 11th ICAR*, 2003, vol. 1.
- [19] N. Vandapel, F. Hubert, A. Kapuria, and M. Hebert, "Natural terrain classification using 3-D lidar data," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 26–May 1, 2004, vol. 5, pp. 5117–5122.
- [20] J. F. Lalonde, N. Vandapel, and M. Hebert, "Data structure for efficient processing in 3-D," in *Robotics: Science and Systems*, vol. 1. Cambridge, MA: MIT Press, Jun. 2005.
- [21] D. Langer, J. K. Rosenblatt, and M. Hebert, "An integrated system for autonomous off-road navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, vol. 1, pp. 414–419.
- [22] H. Seraji, "Traversability index: A new concept for planetary rovers," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999, pp. 2006–2013.
- [23] A. Howard and H. Seraji, "Real-time assessment of terrain traversability for autonomous rover navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2000, pp. 58–63.
- [24] A. Howard and H. Seraji, "Vision-based terrain characterization and traversability assessment," *J. Robot. Syst.*, vol. 18, no. 10, pp. 577–587, 2001.
- [25] F. Nashashibi, M. Devy, and P. Fillatreau, "Indoor scene terrain modeling using multiple range images for autonomous mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1992, vol. 1, pp. 40–46.
- [26] S. Betgé-Brezetz, "Modélisation Incrementale et Localisation Pour la Navigation D'un Robot Mobile Autonome En Environnement Naturel," Ph.D. dissertation, Laboratoire d'analyse et d'architecture des systèmes du CNRS. Univ. Paul Sabatier de Toulouse, Toulouse, France, Ferrier 1996.
- [27] S. B. Goldberg, M. W. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Proc. IEEE Aerosp. Conf.*, 2002, pp. 2025–2036.
- [28] D. B. Gennery, "Traversability analysis and path planning for a planetary rover," *Auton. Robots*, vol. 6, no. 2, pp. 131–146, Apr. 1999.
- [29] H. Seraji, "Fuzzy traversability index: A new concept for terrain-based navigation," *J. Robot. Syst.*, vol. 17, no. 2, pp. 75–91, 2000.
- [30] K. V. Mardia and P. E. Jupp, *Directional Statistics*, ser. Wiley Series in Probability and Statistics. Hoboken, NJ: Wiley, 1999.
- [31] C. Castejón, D. Blanco, B. L. Boada, and L. Moreno, "Traversability analysis technics in outdoor environments: A comparative study," in *Proc. 11th ICAR*, 2003.
- [32] C. Castejón, B. L. Boada, and L. Moreno, "Topographical analysis for Voronoi-based modelling," in *Proc. 28th Annu. Conf. IEEE Ind. Electron. Soc.*, 2002, pp. 2548–2553.
- [33] J. C. Russ, *The Image Processing Handbook*, 2nd ed. Boca Raton, FL: CRC Press, 1994. EEUU.
- [34] H. Choset, "Sensor based motion planning: The hierarchical generalized Voronoi graph," Ph.D. dissertation, California Inst. Technol., Pasadena, Mar. 1996.
- [35] J.-C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer, 1991.
- [36] Y. D. Kwon and J. S. Lee, "A stochastic map building method for mobile robot using 2-D laser range finder," *Auton. Robots*, vol. 7, no. 2, pp. 187–200, Sep. 1999.
- [37] A. J. Davison and N. Kita, "Sequential localisation and map-building for real time computer vision and robotics," *Robot. Auton. Syst.*, vol. 36, no. 4, pp. 171–183, Sep. 2001.
- [38] K. Nagatani and H. Choset, "Toward robust sensor based exploration by constructing reduced generalized Voronoi graph," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1999, pp. 1678–1698.
- [39] C. Castejón, L. Moreno, and M. A. Salichs, "Traversability modelling in 3-D environments," in *Proc. 3rd Int. Conf. FSR*, Helsinki, Finland, Jun. 2001.



Cristina Castejón received the B.Sc. degree in industrial engineering and the Ph.D. degree in industrial technologies from the University Carlos III of Madrid (UC3M), Madrid, Spain, in 1998 and 2002, respectively.

She is currently a Research Assistant with the Mechanical Engineering Unit, UC3M. She has developed a new methodology to model outdoor environments for mobile robots based on 3-D information and a VD technique.



Dolores Blanco received the B.S. degree in physics from the University Complutense of Madrid, Madrid, Spain, in 1992 and the Ph.D. degree in mechatronics from the University Carlos III of Madrid (UC3M), Madrid, in 2002.

From 1996 to 1999, she was a Fellowship Student with the Department of Systems Engineering and Automation, UC3M, where she has been an Assistant Professor since 1999. She is also a member of Mobile Manipulator Group at UC3M. Her current research interests include sensor-based path planning, localization, and control for mobile manipulators.



Luis Moreno (M'91) received the B.S. and Ph.D. degrees in systems engineering and automation from the Polytechnical University of Madrid (UPM), Madrid, Spain, in 1984 and 1988, respectively.

From 1984 to 1994, he was with the Department of Systems Engineering and Automation, UPM. Since 1994, he has been an Associate Professor with the Department of Systems Engineering and Automation, University Carlos III of Madrid (UC3M), Madrid. He is a member of the Robotics Laboratory at UC3M. His current research interests include sensor-based path planning, localization, and control for mobile manipulators.