Proceedings of the 1998 IEEE
International Conference on Robotics & Automation
Leuven, Belgium • May 1998

Range Data Merging for Probabilistic Octree Modeling of 3-D Workspaces

P. Payeur†, D. Laurendeau†, C. M. Gosselin‡
Computer Vision and Digital Systems Laboratory
†Dept of Electrical and Computer Engineering
‡Dept of Mechanical Engineering
Laval University
Sainte-Foy, Québec, Canada, G1K 7P4
http://www.gel.ulaval.ca/~vision
[ppayeur, laurend, gosselin]@gel.ulaval.ca

Abstract

In a previous paper, probabilistic occupancy modeling has been successfully extended to 3-D environments by means of a closed-form approximation of the probability distribution. In this paper, the closed-form approximation is revisited in order to provide more reliable and meaningful models. A merging strategy of local probabilistic occupancy grids originating from each sensor viewpoint is introduced. The merging process takes advantage of the multiresolution characteristics of octrees to minimize the computational complexity and enhance performances. An experimental testbed is used to validate the approach and models computed from real range images are presented.

1 Introduction

The modeling of a 3-D scene is a critical issue in computer vision and image processing. With the continuous increase in computational power, computer vision is now intended to provide a wider autonomy to robotics systems such as mobile vehicles and telemanipulators. Complex scenes found in these applications involve huge amount of data which must be processed rapidly. This results in a need for compact storage schemes and efficient ways to handle workspace models.

One of the major approaches in 3-D modeling is polyhedra modeling which fits geometrical primitives to parts of the scene [6]. Another approach is octree modeling which is based on a hierarchical data structure [1]. The spatial representation of this structure is called the occupancy grid and results from the tessellation of 3-D space into voxels. Each voxel is tagged with the state of occupancy for the area it encloses. Both approaches produce compact data structures. The difference resides in the nature of the information they each convey. Polyhedric models provide information about the arrangement of objects into the workspace. They are usually built by finding the edges and faces of objects on intensity or range images. These models involve a segmentation effort which is computationally intensive. Octree models contain information about the state of each small piece of space independently of the nature, shape and complexity of the objects.

For guiding a robot in a complex environment, discrete octrees that only provide *empty*, *occupied* or *unknown* states of the cells may not always be appropriate [3]. A numerical estimate of the occupancy probability ranked between 0.0 and 1.0 should be associated with each voxel in order to provide a more complete information to path planning algorithms for instance. Elfes [4, 5] has initially proposed a framework for building 2-D probabilistic occupancy maps of the environment using a sensor error model and a Bayesian probabilistic approach to combine data from multiple viewpoints. The complexity of the estimation of the occupancy probability as well as the merging phase of data from various viewpoints depends on the resolution of the map and increases exponentially with the number of cells. This leaves the method impractical for 3-D applications.

In a previous paper [10], a new formalism of the Bayesian occupancy probability estimation procedure has been presented. A closed-form approximation of the occupancy probability function has been proposed to circumvent the computational complexity associated with the estimation of this probability from the range data. Combining this strategy with an optimized merging approach which was summarily introduced results in a computationally tractable algorithm for building 3-D occupancy maps of the environment.

In this paper, refinements of the closed-form approximation are presented. Thereafter, emphasis is put on the merging of local probabilistic occupancy grids built for each viewpoint visited by the sensor. This approach takes advantage of the multiresolution property of octrees to avoid useless volume matching during the merging phase. Furthermore, the approach has recently been tested on an experimental testbed. Results provided from real range images are presented for the typical application of concern: guidance of a telemanipulator in servicing on-line electricity distribution equipments.

0-7803-4300-x-5/98 $10.00 © 1998 IEEE

Section 2 proposes a brief review of the closed-form approximation of the occupancy probability function as well as some refinements which have been brought to the initial implementation. Sections 3 and 4 describes the merging strategy of data from various viewpoints. Finally, results from real images are presented and performances are analyzed.

2 Enhanced closed-form approximation of the probability distribution

Building a probabilistic occupancy grid model of objects involves the estimation of the probability of occupancy of each part of space. Using range data, Elfes [4] has proposed a building scheme in which measurements are interpreted by means of a probabilistic model of the sensor. The resulting occupancy grid cell probabilities are finally merged using a Bayesian procedure. This approach revealed to be computationally untractable for 3-D space in spite of the assumption of independent cell states and the discretization of these states to *empty* and *occupied* values.

We have recently proposed a strategy to reduce the heavy computational task implied in evaluating conditional probabilities for the entire set of possible configurations of the grid [10]. This approach uses a closed-form approximation of the characteristic occupancy probability distribution function (OPDF) as obtained by Elfes' approach. When applied directly to range measurements, this approximated OPDF allows a direct computation of occupancy probability $P(\rho, \theta, \phi)$ for a given cell taking into account the noise characteristics of the sensor. For a generic range sensor mounted on a two degree-of-freedom mechanism (azimuth, θ, and elevation, ϕ), the closed-form approximation of the OPDF for a single measurement is expressed as follows:

$$P(\rho,\theta,\phi) = \frac{1}{2}\left(1 + e^{-\left(\frac{2((\rho-\bar{\rho})+2\sigma_\rho)}{\sigma_\rho} + \frac{(\theta-\bar{\theta})^2}{\sigma_\theta^2} + \frac{(\phi-\bar{\phi})^2}{\sigma_\phi^2}\right)}\right)^{-1} + A_G \cdot e^{-\left(\frac{(\rho-\bar{\rho})^2}{\sigma_\rho^2} + \frac{(\theta-\bar{\theta})^2}{\sigma_\theta^2} + \frac{(\phi-\bar{\phi})^2}{\sigma_\phi^2}\right)} \quad (1)$$

where $(\sigma_\rho^2, \sigma_\theta^2, \sigma_\phi^2)$ are the variance characterizing the sensor along each axis and $(\bar{\rho}, \bar{\theta}, \bar{\phi})$ are the coordinates of the actual measurement provided by the sensor. The coefficient of the Gaussian term, A_G, has been readjusted to a value of 3/20 (the original one was 1/3) in order to provide a less saturated occupancy model when an object is measured repeatedly from the same viewpoint. When a given point is sampled several times, the occupancy probability of this point progressively grows as a result of the Bayesian merge as shown in figure 1 for an object located at 50 mm from the sensor. If A_G is too high, the probability reaches 99% too rapidly, thus producing a model which is saturated in terms of the occupancy probability. Reducing the value of this coefficient allows a slower growth and thus for a more realistic behavior of the modeling approach. For a match with the Elfes' probabilistic distribution evolution, the weighting coefficient should have been set to 1/20. But we have observed that the confidence in the model grows too slowly with this value. Tuning of this parameter is made on an experimental basis and depends on the application.

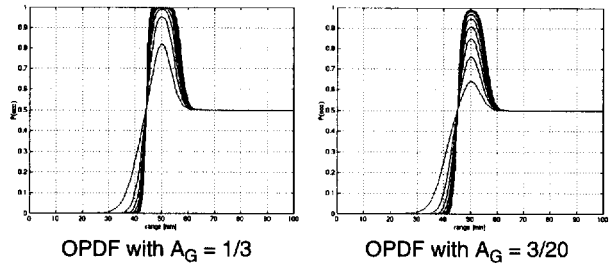

Figure 1 : *Evolution of the approximated OPDF for multiple measurements of the same object from the same viewpoint. Each curve corresponds to one measurement being added to the previous occupancy model.*

Because of the scanning pattern of the range sensor (a laser line scanner based on a rotary mechanism), the resulting occupancy probability distribution is temporarily stored into a spherical probabilistic occupancy grid. Such a spherical grid is obtained for each viewpoint from which the sensor gathers range data as shown in figure 2. In order to build the desired cubic model of the scene containing the entire set of probability distributions, the local spherical grids must be integrated into one global Cartesian occupancy grid of cubic shape. Figure 3 shows an overview of the approach. The steps located on the first row have been described in [10] while the merging process from spherical grids to the final octree is detailed in sections 3 and 4.

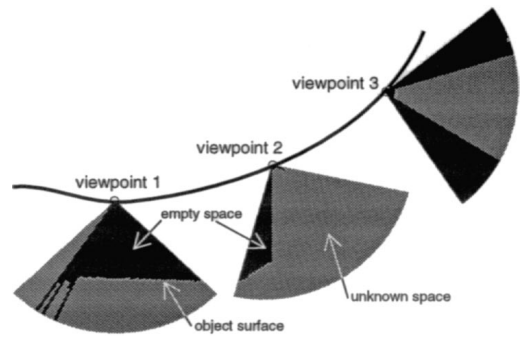

Figure 2 : *Set of spherical occupancy grids resulting from the application of the approximated OPDF to measurements from a sequence of viewpoints.*

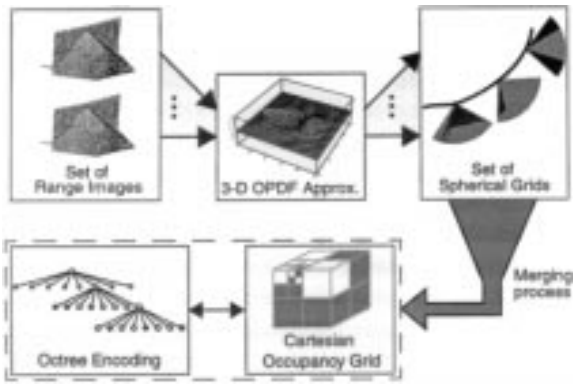

Figure 3 : *Processing steps from range data acquisition to the probabilistic occupancy model encoded as an octree in Cartesian space.*

3 Merging spherical probabilistic occupancy grids in a multiresolution Cartesian model

To merge data provided by a sensor from multiple viewpoints, most approaches use 2-D projections of the objects to build the octree [2, 3, 7]. They require the computation of geometrical or Boolean intersection to find the corresponding volumes which must be tagged as occupied. These approaches result in octrees with discrete entries (*empty* or *occupied*).

To deal with probabilistic grids, Elfes proposed a basic approach to merge a set of local grids. It consists in scanning the entire workspace covered by the global occupancy grid subdivided to the highest resolution level. Each cell in the global grid must be matched with its corresponding cells into the local grids to update the probabilistic occupancy state. The actual occupancy probability contained in a given cell of the global occupancy grid, $P[s(c_i)]$ is combined with the probability read from each local grid by means of the Bayes theorem [9]:

$$P[s(C_i) = OCC|P_1, P_2] = \frac{P_1 P_2}{P_1 P_2 + (1-P_1)(1-P_2)} \quad (2)$$

where P_1 and P_2 are the respective occupancy probabilities of the occupancy grid cells that are merged. Initially, the occupancy probability, which varies between 0.0 and 1.0, is set to 0.5 everywhere for unknown state.

This procedure leads to an algorithm of complexity $O\left(m \cdot Q^n\right)$ where m is the number of independent local occupancy grids, n is the number of dimensions of space and Q is the number of cells to visit for a one-dimensional subspace of the global occupancy grid of width *ModelSize* and resolution *Resolution*:

$$Q = \left\lceil \frac{ModelSize}{Resolution} \right\rceil \quad (3)$$

While such an algorithm remains tractable in 2-D space (e.g. planar grids), experience reveals that it is not suitable for 3-D modeling (volumetric grids) because of the combinatorial explosion in the number of cells.

Considering the 3-D spherical grids resulting from the application of the closed-form approximation of the OPDF, the counterpart of the previous approach would consists in *i)* successively scanning the cells of each spherical grid, *ii)* finding the corresponding cells in the Cartesian grid and *iii)* updating their occupancy probability with equation (2). However, since spherical cell boundaries do not coincide with Cartesian cells, some holes may appear in the Cartesian grid. Interpolation techniques must then be applied to update the occupancy probability of those cells.

In this work, it is rather proposed to take advantage of the multiresolution property of octrees to avoid useless volume matching where no data had been gathered. Elfes limited is work to quadtrees of single resolution. This explains why the entire set of voxels must always be scanned for the integration of each new local grid. Our approach prevents the algorithm from visiting a large number of maximum resolution voxels in most cases.

The merging algorithm searches for intersections between cubic cells and spherical grids to identify the regions where they share common space. Starting at the coarsest level of resolution of the Cartesian occupancy grid, this search is performed in a recursive and descending order in the octree. For each level of resolution of the octree, intersections are searched for between one of the eight Cartesian grid voxels and each spherical grid. When the Cartesian voxel does not match any part of any spherical grid, it is discarded and its occupancy probability is left unchanged. On the other hand, when a spherical occupancy grid intersects with the voxel of interest, the intersection check is repeated for each of the eight children of this voxel at the following level of resolution in the octree.

Consider for instance a Cartesian occupancy grid of size 8x8x8 when subdivided to the highest resolution level and a spherical occupancy grid intersecting with a corner of the Cartesian grid as shown in figure 4. The corresponding octree model is also illustrated. The white nodes in the tree represent areas where no intersection is found. Since the search is based on the octree corresponding to the Cartesian grid and starts from the lowest resolution cell which is the root of the tree, the search stops at the second level of resolution for all the white nodes. This means that 7/8 of the total space covered by the model is processed in only 7 search steps.

Children number 5 needs more refinement since it intersects with the spherical grid. Therefore the search is recursively pursued in this subspace until the highest resolution level is reached. The occupancy probability of the Cartesian grid is finally updated with equation (2) in accordance with the data stored in the spherical grid in this area.

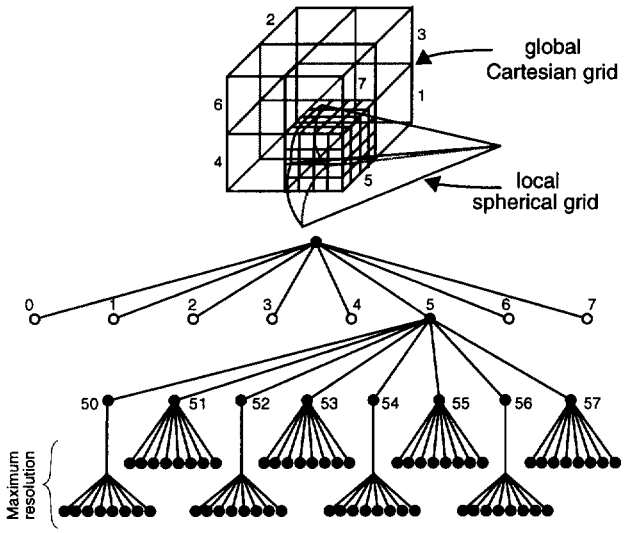

Figure 4 : *A spherical occupancy grid intersecting with a corner of a Cartesian multiresolution grid and the corresponding octree model.*

This example clearly shows the advantage provided by the use of a multiresolution occupancy grid. If a grid of a single resolution would have been used, the entire space covered by the model would have been processed at the highest resolution. This would have resulted in a lengthy merging procedure.

To really benefit from the proposed approach, the overhead created by the required recursive check for intersections must be kept to a minimum. Intersection checking is thus based on a progressive refinement such that the fastest parts of the search are made first.

4 The search for intersections

Even though several types of intersection can occur between a cube and a section of sphere as shown in figure 5, these can be grouped into three categories. The first one includes intersections which can be detected by an examination of the eight corners of the cube, such as types 1 and 5. The second class is concerned with intersections which can be found by the examination of vertices of the section of sphere, such as types 2 and 4. Finally, there are some intersections which only imply a shared volume of space delimited by the faces of the cube and the section of sphere, such as type 3.

In order to optimize the intersection search, the algorithm first looks for intersections of types 1 and 5 since the coordinates of the corners of a given cell of the Cartesian occupancy grid are easily computed. If this search fails, intersections of types 2 and 4 are checked for. Finally, and only if necessary, intersections between bounding planes are searched for. Intersection search and boundary validation are based on scalar products between the normal vectors to the bounding planes of the Cartesian and spherical cells. The normal vectors as well as the spherical grid vertices are precomputed to speed up the procedure.

4.1 Intersection finding by cubic cell corners inclusion in the section of sphere (types 1 and 5)

Let us assume that a cube is defined by the Cartesian coordinates of its eight corners, $corner_i$, as shown in figure 6. A set of eight vectors $\vec{v}_{ci}$ is defined that join the origin of the section of sphere, $vertex_0$, to each of these corners. The inclusion of a given corner in the section of sphere can then be easily verified by means of a logical test based on scalar products between the vector corresponding to this corner and the normal vectors to the planes bounding the section of sphere.

$$\begin{aligned} &[(\vec{n}_{s1} \bullet \vec{v}_{ci}) \cdot (\vec{n}_{s2} \bullet \vec{v}_{ci}) \le 0] \\ \perp\ &[(\vec{n}_{s3} \bullet \vec{v}_{ci}) \cdot (\vec{n}_{s4} \bullet \vec{v}_{ci}) \le 0] \\ \perp\ &[\vec{v}_{center} \bullet \vec{v}_{ci} \ge 0] \end{aligned} \tag{4}$$

where $\bullet$ is the scalar product, $\perp$ is the logical AND operator.

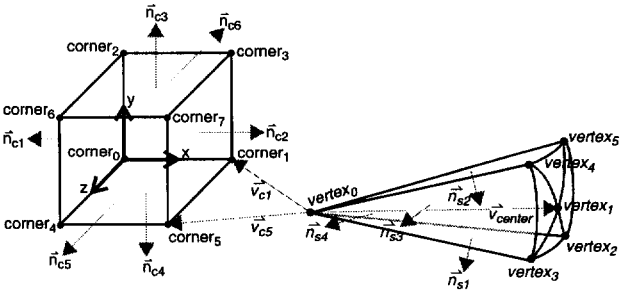

Figure 6 : *Definition of a cube and of a section of sphere by corners, vertices and normal vectors.*

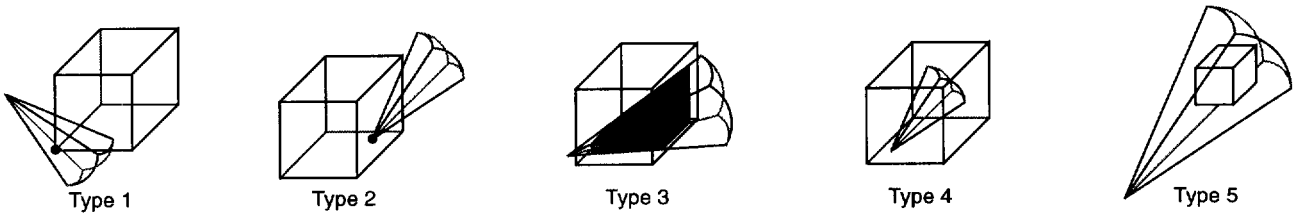

Figure 5 : *Types of intersection occurring between a cube and a section of a sphere.*

When equation (4) is true, an intersection of type 1 or 5 occurs. This test is repeated successively for each of the eight corners of the cube or until it is validated for one of them. In this case, the intersection search is pursued directly to the next level of resolution without testing any other corner. If the test is not validated for any corner, the search continues as described in the next section.

4.2 Intersection finding by spherical grid vertices inclusion in the cube (types 2 and 4)

The section of sphere can be represented by a set of six vertices defining its extent in space as shown in figure 6. Given that the coordinates of these vertices are known, their inclusion into the cube is easily tested as follows for the *ith* vertex.

$$corner_0|_x \le vertex_i|_x \le corner_1|_x \quad (5)$$

$$corner_0|_y \le vertex_i|_y \le corner_2|_y \quad (6)$$

$$corner_0|_z \le vertex_i|_z \le corner_4|_z \quad (7)$$

When these three criteria are simultaneously verified, an intersection of type 2 or 4 occurs. The test is applied successively to each of the six vertices (i = 0, ..., 5) of the section of sphere or until it is validated for one of them thus resulting in a recursive intersection search at the next resolution level. If the test is not validated for any vertex, intersections between bounding planes must be checked for.

4.3 Intersection finding on bounding planes (type 3)

Infinite planes in 3-D space can be defined by their normal vector and one point belonging to the plane. Considering that a cube and a section of sphere are delimited respectively by six and four planes, the closed-form expression of the straight line resulting from the intersection between two of these planes is obtained as follows:

$$\hat{n}_{ci} \bullet \hat{x} = \hat{n}_{sj} \bullet \hat{x} \quad (8)$$

where $\hat{n}_{ci}$ and $\hat{n}_{sj}$ are respectively the normal vectors of the *ith* face of the cube and of the *jth* face of the section of sphere as shown in figure 6 and $\hat{x}$ is the coordinate vector of a set of points belonging to both of these planes.

Locating the intersection with equation (8) provides the closed-form expression of an infinite straight line in 3-D space where one plane of the cube intersects with a plane of the section of sphere. The cube and the section of sphere are bounded volumes. An intersection between these volumes occurs only if a segment of this straight line is included simultaneously into the cube and the section of sphere. Therefore boundary checking is necessary to validate an intersection.

In order to satisfy this requirement, coordinates of the points which bound the segment of the straight line that is included into the cube are computed. Each face of the cube can contain up to two bounding points as illustrated in figure 7 for various types of intersection. For a valid intersection to occur, the straight line segment must simultaneously cross the section of sphere and the cube. A logical test based on scalar products is used to verify this condition.

Considering a straight line segment delimited by two bounding points and a pair of facing planes bounding the section of sphere, four types of intersection can occur. This is easier to visualize in a 2-dimensional case such as in figure 8. The partial or total inclusion of the segment into the section of circle, given the normal vectors to each angular boundaries, is tested with the following expression:

$$\{ [(\hat{n}_1 \bullet \hat{v}_{b0}) \cdot (\hat{n}_2 \bullet \hat{v}_{b0}) \le 0] \perp [\hat{v}_{center} \bullet \hat{v}_{b0} \ge 0] \}$$

$$\| \{ [(\hat{n}_1 \bullet \hat{v}_{b1}) \cdot (\hat{n}_2 \bullet \hat{v}_{b1}) \le 0] \perp [\hat{v}_{center} \bullet \hat{v}_{b1} \ge 0] \} \quad (9)$$

$$\| \{ [(\hat{n}_1 \bullet \hat{v}_{b0}) \cdot (\hat{n}_1 \bullet \hat{v}_{b1}) \le 0] \perp [\hat{v}_{center} \bullet \hat{v}_{b0} \ge 0] \}$$

where vector $\hat{v}_{bi}$ joins the origin of the section of circle to the *ith* bounding point and $\|$ is the logical OR operator.

A distance check is also required to ensure that the bounding points are not too far away from the origin of the section of circle.

In a 3-dimensional space, the test in equation (9) is repeated for each combination of the six bounding planes of the cube with the four spherical grid bounding planes. As soon as one test is validated, the search on this cube is stopped since an intersection with the spherical grid is ensured. As usual, the search is then pursued on the eight children of the cube under analysis until the occupancy probability can be updated at the highest resolution level for those children who actually intersect the spherical grid.

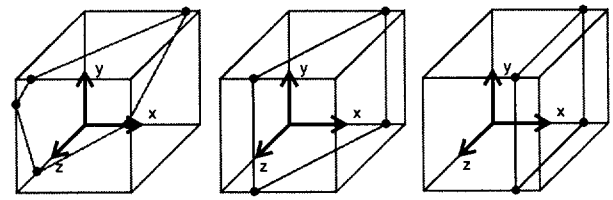

Figure 7 : *Boundaries of the intersection between the cube and one of the planes bounding the section of sphere.*

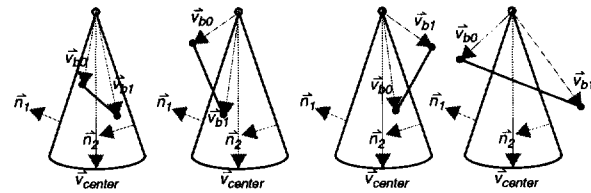

Figure 8 : *Four configurations in which a straight line segment is included into a section of circle.*

In some particular cases where a spherical cell bounding plane is aligned with one of the reference frame axes as shown in the middle and the right drawings of figure 7, the logical test does not need to be repeated for each of the six faces of the cube. A single test on four bounding points is rather used only once for each of the four spherical grid bounding planes.

5 Results

To demonstrate the validity of the proposed probabilistic octree building algorithm, a synthetic scene containing an electricity distribution pole has been used to generate a series of 128x128 range images from 10 different viewpoints. A simulated range finder was used to generate the range images. Wire frame objects of exact dimension are superimposed to the model as seen in figure 9. Both the pole, the beam and the wire match, within the range of precision, the size of the original objects from which range data were acquired.

In figure 9, gray shading of the cells corresponds to the probability of occupancy; white being 100% (*occupied*) and black 0% (*empty*). For the purpose of displaying the entire modeled volume, only voxels having a probability of occupancy higher than 50% (*unknown*) are displayed.

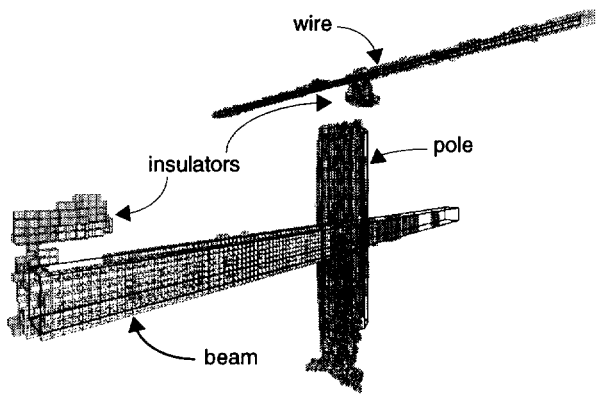

Figure 9 : *Matching of the size of each part of the model with their respective actual component.*

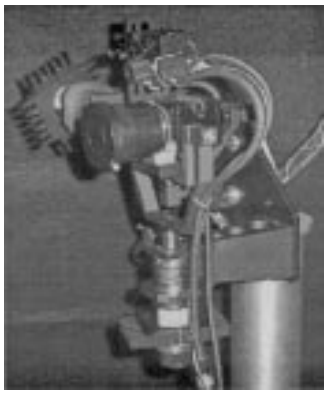

Figure 10 : *The TRID range sensor mounted on the two degree-of-freedom aiming device.*

Moreover, in order to ensure that the approach is appropriate for real world applications, it has been tested with real range images acquired with an actual range sensor. The range sensor (TRID) [8] recently developed in our laboratory has been used to gather data. TRID measures range using an active approach based on a combination of "range from defocusing" and active triangulation with a laser beam. It provides a single range reading at a rate of ~15 samples/second with an accuracy of 0.3% at 1 meter. The sensor is mounted on a two degree-of-freedom (azimuth and elevation) mechanism which allows a fast scanning of the scene. The TRID setup is depicted in figure 10.

To demonstrate the behavior of the occupancy probability when parts of the scene are measured repeatedly, as presented in figure 1, three partial images of a pyramid have been sampled from a single viewpoint. Figure 11 illustrates the setup with the range sensor device and the pyramid. The scanning procedure consisted in three steps. The first step consists in scanning the entire pyramid. Building the occupancy probability model with these data results in the model of figure 12a. The sides of the cells have not been drawn for clarity. The gray level indicates the occupancy probability. The probability is approximately equal for each part of the scene where obstacles have been encountered. The maximum value is around 0.64. Second, only the top part of the pyramid is scanned. The occupancy probability in this area is then updated to about 0.76 as shown in figure 12b. Finally, a third scan has been gathered on a strip in the middle on the pyramid. The occupancy probability is augmented to 0.85 since the presence of objects in this region has been detected three times and these informations have been merged. The final probability distribution is illustrated in figure 12c. Figure 12d shows the resulting model when the three images are merged. The pyramid and the backplane are easily recognized.

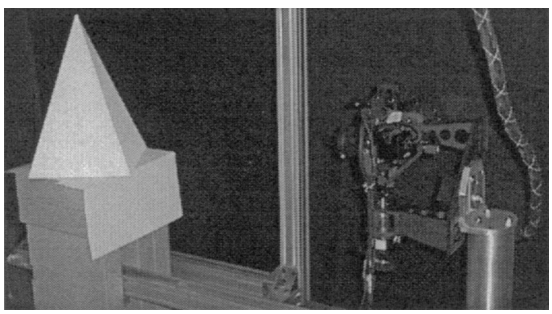

Figure 11 : *Setup used to demonstrate the increase in probability occupancy when given points are measured many times.*

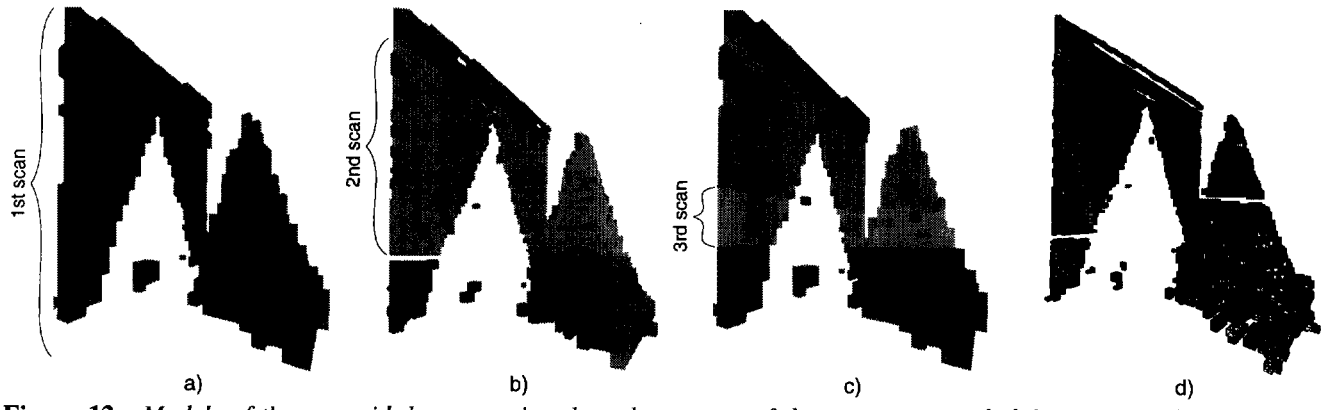

Figure 12 : *Models of the pyramid demonstrating the enhancement of the occupancy probability in accordance with the multiplicity of scans.*

Computation on these probabilistic octrees demonstrates the quality of models which can be obtained with the proposed approach and the behavior of the occupancy probability in function of the number of scans gathered in a given area of an object. This confirms the validity of the closed-form approximation of the OPDF presented in section 2.

As a final step in the validation process of the approach, real range images have been gathered on a typical scene of an electricity distribution network as shown in figure 13. 50x50 range images from two different viewpoints were used to compute the model. "Real world" constraints are significant in this case. Specularity of some surfaces like the insulator and the wire as well as the orientation of surfaces like the top of the beam limit the performance of TRID. Nevertheless, in spite of the reduced set of data and the fact that the sensor is still under development, the major components of the scene can easily be located in the model as shown in figure 14. Holes in the surfaces correspond to points where the sensor did not succeed in estimating the distance. This problem is mainly concentrated on the edges of the objects or on thin pieces like the pin that supports the insulator. On the other hand, the sizes of the modeled parts match with the real dimension of the objects.

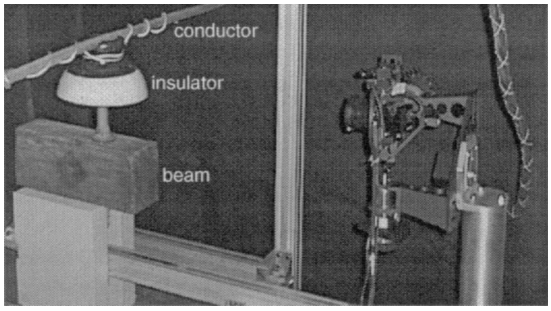

Figure 13 : *Setup used to gather data on the electricity distribution network scene.*

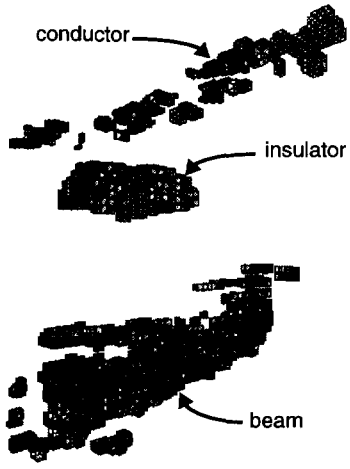

Figure 14 : *Model of the beam, the insulator and the wire resulting from merging of two real range images.*

All models illustrated in this section have been computed to a 5 *mm* resolution. The time required to merge one range image depends on the size of this image and varies from about 2 minutes for each pyramid model to about 20 minutes for the simulated electricity distribution pole in figure 9. The average time to merge one range measurement into a current model is estimated to 0.08 second. These computations were performed on a O_2 Silicon Graphics workstation. Memory requirements appear to be a critical issue during computation since cells of similar occupancy probability are merged only at the end in order to provide a compact octree.

6 Complexity

At first glance, the merging process seems to be quite complex because of the search for geometrical intersections. But considerable reduction of the computing load is provided by the use of multiresolution models which allows large areas to be discarded early in the merging process. In the worst case, the complexity is

similar to that of the original Elfes' scheme $O\left(m \cdot Q^n\right)$ where m is the number of local spherical grids, n is the number of dimensions of the model and Q is the number of cells along a side of the model for a given resolution as expressed in equation (3). Nevertheless, such a situation rarely occurs in practice since the external size of the Cartesian model is chosen as a bounding box which includes the entire set of local spherical grids given their respective location and orientation in 3-D space. Therefore, a complete scan of the Cartesian grid is never necessary and the total merging complexity is reduced compared to the original scheme.

The specific complexity of the intersection search between the cubes and the sections of sphere is hard to evaluate. It depends on the extent of objects, on the number of objects in the scene as well as on the maximum desired resolution and on the type of intersections that occur. Selecting viewpoints that provide range images with a rich content is of prime importance. It is thus advantageous to concentrate the scan on a zone of interest. This contributes to reduce the total volume of space to model and therefore the size of the root cell of the Cartesian occupancy grid. Resolution overestimation should also be prohibited because of its impact on the computing time.

Experiences also revealed that range data grouped has complete raster scan images are more efficient than a series of line scans. Range images such as those provided by the two degree-of-freedom sensor of figure 10 result in a smaller number of spherical grids and therefore in less intersection searches and validations.

7 Concluding remarks

In this paper, enhancements to the proposed scheme for probabilistic occupancy grid building in 3-D space have been presented. Models which are less saturated in term of occupancy probability can now be obtained. This characteristics reveals to be a critical issue when the model is used to drive a robot since some decisions must be taken on the basis of the occupancy probability in each cell. When multiple free paths are available for the robot arm to reach a target, the selection of the safer one is a one step procedure since the occupancy probability from the volumetric model is directly used as the main criterion. Therefore, the proposed closed-form approximation of the OPDF appears to be a more flexible approach than the original scheme because of its adaptability to specific applications.

The merging process of data gathered from various viewpoints has also been detailed. Multiresolution characteristics of octrees and characterization of intersection types between a cube and a section of sphere allows significant reduction in the time that is required for merging local spherical grids with the global Cartesian grid describing the workspace.

Acknowledgments

The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Centres of Excellence Program of the Government of Canada, the Natural Sciences and Engineering Research Council (NSERC) and the participation of PRECARN Associates Inc. Special thanks are also addressed to Mr. Sylvain Comtois whose help in gathering experimental data was greatly appreciated.

References

[1] H. H. Chen and T. S. Huang, "A Survey of Construction and Manipulation of Octrees", *Computer Vision Graphics & Image Processing*, 43(3):409-431, September 1988.

[2] C.H. Chien, Y.B. Sim and J.K. Aggarwal, "Generation of Volume/Surface Octree from Range Data", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 254-260, Ann Arbor, MI, June 1988.

[3] C. I. Connolly, "Cumulative Generation of Octree Models from Range Data", in *Proceedings of the IEEE International Conference on Robotics*, pp. 25-32, Atlanta, GA, March 1984.

[4] A. Elfes, *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 1989.

[5] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation", *IEEE Computer*, 22(6):46-57, June 1989.

[6] F. P. Ferrie and M. D. Levine, "Deriving Coarse 3D Models of Objects", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 345-353, U. Michigan, Ann Arbor, MI, May 1992.

[7] A. Li and G. Crebbin, "Octree Encoding of Objects from Range Images", *Pattern Recognition*, 27(5):727-739, May 1994.

[8] F. Loranger and D. Laurendeau, "TRID, a Fast and Accurate Range Sensor", *International Conference on Advances in 3-D Digital Imaging and Modeling*, pp. 51-58, Ottawa, Canada, May 1997.

[9] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd Edition, McGraw-Hill, 1991.

[10] P. Payeur, P. Hébert, D. Laurendeau and C. M. Gosselin, "Probabilistic Octree Modeling of a 3-D Dynamic Environment", in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1289-1296, Albuquerque, NM, April 1997.