A Parametric Alternative to Grids for Occupancy-Based World Modeling

Dmitry O. Gorodnichy and William W. Armstrong
Department of Computing Science,
University of Alberta,
Edmonton, Alberta, Canada T6G 2H1
http://www.cs.ualberta.ca/ dmitri/WorldModeling

Abstract – **In the paper, we consider an occupancy-based approach for range data fusion, as it is used in mobile robotics. We tackle the major problem of this approach, which is the redundancy of stored and processed data caused by using the grid representation of the occupancy function, by proposing a parametric piece-wise linear representation. When applied to vision-based world exploration, the new representation is shown to have advantages, which include its suitability for radial range data, its efficiency in representing and fusing range data, and its convenience for navigation map extraction. The proposed technique is implemented on a mobile robot, Boticelli. The results obtained from running the robot are presented.**

Keywords – **Occupancy grids, mobile robots, vision-based world exploration.**

1 Introduction

In mobile robot world exploration, the occupancy-based approach is one of the most commonly used [9, 3, 8, 16, 4, 10, 14]. In this approach, the exploration policy is determined by the occupancy model of the world which is built from range data registered by robot sensors. The following information is usually extracted from the model: the observed obstacles, the navigation area, which is the area free of obstacles, and the unexplored area, which is area where insufficient range data has been acquired.

When this information is obtained, it is processed in order to produce the command for the robot. Such methods as potential fields [8], value iteration [13] and other reinforcement learning techniques [12] are most common at this stage.

As understood, no matter what a technique is used at later stages, the success of the occupancy-based world exploration depends on the quality of the occupancy model and also its suitability for extraction of the information required for navigation.

Occupancy function

The occupancy world model is defined by an *occupancy function* which maps 3D points of the world into a real interval so that higher values of the function indicate points that are more likely to be occupied[1]:

$$m = F(\vec{r}), \quad m \in [0,1], \quad \vec{r} \in \Re^3. \tag{1}$$

The major problem of occupancy-based world modeling concerns the representation of this function. A conventional grid representation, which represents the occupancy function as a multi-dimensional array, results in storing huge amount of data and very time-consuming calculations required to process these data. This makes modeling of 3D and large-scale environments in real time practically impossible.

Because of the problem described above, up till now in mobile robotics, where the issue of time is critical, only 2D occupancy models have been used. That is, instead of treating a world the way it is, a robot has to consider only a 2D shadow of it in making a navigation decision. And this is not the only problem encountered using the grids. Grid models are not suitable for radial range data, which is the most frequent case, and they are very inefficient for map extraction.

Thus, there is a need for another representation of the occupancy function which would suit range data well, be optimal space-wise and allow efficient navigation. This led us to propose a regression-based tech-

[1] In some cases, a similarly defined *emptiness* function is used, and sometimes both functions are used [10].

nique for range data fusion using a piece-wise linear representation for the occupancy function. While the regression issues of the approach are covered in [6], this paper focuses on the issue of the occupancy function representation and its application to the world exploration problem. We show how the occupancy model of the world can be efficiently represented using a min-max tree of combining linear functions and then we show how the information needed for exploration can be efficiently extracted from the constructed model. We demonstrate the validity and the promise of our approach by implementing it on the mobile robot Boticelli, which searches for objects in an unknown environment using a single camera stereo range sensor described in [7].

The paper is organized as follows. In the next section, we describe the problem of vision-based world modeling the way it is used for world exploration and define range data fusion as a regression problem. Parametrically represented occupancy models are built in Section 3 and extraction of the navigation maps from the models is described in Section 4. Discussions conclude the paper.

2 Vision-based world modeling

Single-camera range data

In [7] we present the design of a single-camera stereo range sensor, which registers visible 3D features around a robot. Along with a 3D vector $\vec{r}$ of a registered visual feature measured in the robot-centered system of coordinates (see Figure 1), the sensor provides the evidence value m of the feature, which is calculated according to the evidence theory paradigm [15]. More specifically, the evidence value of a feature is determined by the match error obtained during the stereo acquisition and is a value between zero and one.

An example of range data registered by a single-camera sensor is shown in Figure 5.a. These data are acquired by observing the room shown in Figure 4. Registered 3D features are shown projected on the floor (Oxy plane), the robot is located in the center. The features with higher evidence are shown brighter.

As can be seen, the range data acquired by this sensor are very suitable to be represented by a parametric occupancy function $m = F(\vec{r})$ defined in the robot-centered system of coordinates.

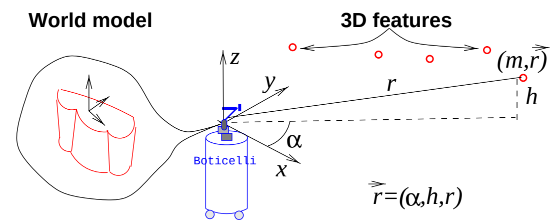

Figure 1: *Vision-based world modeling.*

Using the evidential approach

Our preference of the evidence theory over the probability theory in building the occupancy function is for to the following three reasons. First, as discussed in [7], the evidence approach provides a straightforward way of assigning the evidence values to features based on the uncertainty parameters of the sensor. Second, using only one occupancy function in Eq. 1 does not resolve the "contradictory *vs* unknown" problem, which occurs when $m = \frac{1}{2}$. This makes the extraction of the unexplored area, i.e. the area where no features have been observed yet, impossible. Therefore, two occupancy functions: belief and plausibility, – have been suggested [10] instead of the one occupancy function to combine range data, which is an approach supported by the evidential theory. Third, since we use regression for combining evidence values, the probability axioms do not hold for combined evidence values. In [6] we describe the reasons for using regression for range data fusion, the major one of which is the desire to handle dependent range data.

Exploration task

The task we consider for the robot is the following: to explore an unknown environment for the purpose of finding an object, where the exploration policy is determined by a world model obtained from range data registered by robot sensors. Let us describe this task in more detail.

Since we use a visual range sensor only, we consider an environment full of visual features in almost any direction the robot looks at. The range data registered by the visual sensor are pairs of numbers $\{m_k, \vec{r}_k\}$ provided by the sensor. The fusion of these data is a batch process, that is, the occupancy function $m = F(\vec{r})$ is built only after all visual features around the robot are observed, rather than being updated continuously with one observed feature at a time. This allows us to consider fusion of these data as a regression problem.

Since only the range data registered from the current robot position are used in fusion, the constructed world model will be a local model. The exploration policy of the robot is determined by obstacle and exploration points extracted from multiple local models.

Fusion as regression

We formulate the fusion of range data as a regression problem as follows. Given a set of sample points $\vec{r}$ along with their evidence values m, find a smooth approximation of the function $m = F(\vec{r})$ on the whole input domain.

The strategies to be used when choosing and applying a regression technique are described in [6]. Here, we concentrate on the issue of representing the models constructed by regression techniques and its application to navigation.

We use the Adaptive Logic Network (ALN) [2] as a regression tool. As a result of the fitting process, the ALN produces a binary tree of minimums and maximums of linear functions on polyhedra in 3D space as illustrated in Figure 2. This has certain advantages, the main ones being the controlled generalization and high-speed execution.

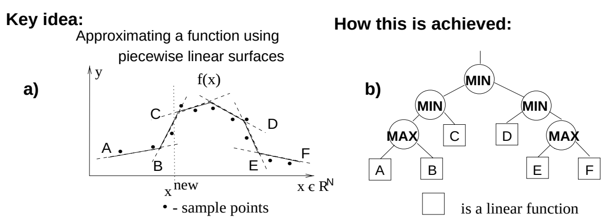

Figure 2: *Approximating a function with linear surfaces.*

3 Piece-linear representation

Choosing coordinate system

Each feature i registered by a sensor induces a set of sample points $\{m_i^j, \vec{r}_i^j\}$, which are the functions of the feature values $\{m_i, \vec{r}_i\}$ in accordance with what is called a *sensor model*. Figure 3.a shows a sensor model of the single camera stereo. According to this model, if a feature is registered at distance r from the camera with the evidence value m, then all points on the ray of view are given the values of believe evidence (Bel) and plausibility evidence (Pl) as shown in the figure. In this paper we use the plausibility evidence values only in constructing occupancy models. Belief evidence models can be built in a similar fashion yet require extra postprocessing.

The relationship of the range data defined by the sensor model determines the choice of the radial-based coordinate system as the most appropriate system to be used in representing the occupancy functions. In particular, we use the coordinate system[2] (see Figure 1), which uses the pan angle α (radians), the height h (decimeters), and the distance r from the pan-tilt unit center on the top of the robot (decimeters). In the next section we show the convenience of this system for local horizontal-plane straight-line navigation.

In this system of coordinates, the form of the occupancy function $m = F(\alpha, h, r)$ constructed by the ALN can be written as

$$m = tree_{l=1...L}^{(MIN,MAX)}\{a_l\alpha + b_l h + c_l r + d_l\}, \qquad (2)$$

where L is the number of linear pieces used in regression.

This system of coordinates allows us to incorporate the sensor model into regression by imposing constraints on the occupancy function. In particular, we impose the constraint

$$\frac{\partial F(\alpha, h, r)}{\partial r} > 0, \qquad (3)$$

which implies that the occupancy function is monotonic increasing in the horizontal plane in the direction from the robot. In the case of the ALN representation of the occupancy function, the monotonicity constraint of Eq. 2 reduces to ($c_l > 0$).

This constraint results in drastic reduction in the number of sample points needed in regression, as illustrated in Figure 3.b. In particular, we generate only five sample points per feature: two with evidence m to account for error in depth calculation, two with evidence less than m to correspond to decreasing occupancy values along the ray towards the robot, and one in the center of the robot with the evidence value equal to zero. The specific locations of these points are defined by the parameters of the sensor model. For comparison, grid-based approaches practically ignore the fact that many sample points used in regression are function of other points and generate sample points as many as there are grid cells between an observed point and the camera, as illustrated in Figure 3.a.

Figure 5.b shows a 3D occupancy model obtained from the range data shown in Figure 5.a using the ALN representation of the occupancy function described above with eight linear pieces ($L = 8$). The points with the occupancy value higher than 0.4 are

[2] Another very suitable system of coordinates is the spherical one.

shown projected onto the floor. The circular appearance of the data is due to uniform sampling in the coordinate system we use.

We would like to emphasize that the constructed model is the occupancy model of the environment and should not be confused with geometry-based models, which are very common in 3D reconstruction and which, in many cases, are also built by fitting piecewise linear surfaces.

As seen in the figure, the constructed occupancy model does not follow exactly the contour of the range data in the depth map. However, while using only a few parameters, it is able to show clearly the areas of high evidence of occupancy as well as the areas of insufficient occupancy information.

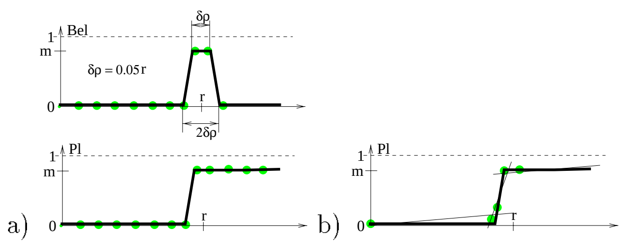

Figure 3: *The visual sensor model sampled according to the grid-based (a) and regression-based (b) techniques.*

4 Extracting Maps

Once a 3D occupancy model of the world is constructed as a tree of minima and maxima of linear functions of three variables, it is possible to determine a 2D polygon within which it is safe for the robot to navigate.

Occupancy Function Inversion

The volume with occupancy less than a certain threshold (e.g. 0.6), is considered unoccupied and therefore available for navigation. In order to find this volume, the first step is to invert the occupancy function. The inverse function returns a distance within which it is safe to move as a function of pan angle, height and occupancy. This can be done theoretically due to the strong monotonicity condition imposed on the occupancy function (Eq. 3) during ALN training.

The ALN max-min tree representation of the occupancy function allows the inversion to be done very efficiently. The inverted ALN is constructed as follows: in the original ALN tree, each maximum node is replaced by a minimum node, and each minimum node by a maximum. Then the weights on variables are normalized in such a way that the weight on the new output variable (occupancy) is –1. The simplicity and speed of this inversion is another advantage of using ALNs for robot navigation.

The inverse occupancy function, which can be now written as

$$r = F^{-1}(\alpha, h, m), \tag{4}$$

where the occupancy is fixed at some level, say $m = 0.6$, is now applied at several values of height h. The r values are converted to a horizontal distance by taking $\sqrt{r^2 - h^2}$. The result of this computation is a set of polygons obtained at different heights centered at the robot's current position. In addition to the monotonicity constraint imposed in the regression (Eq. 3), we also impose upper and lower bounds on the weights for pan angle and height. This allows us to use a finite set of height values and yet be sure that no point on the robot at any height will collide with any point of the environment exceeding a certain value of occupancy.

The final step in calculation of the polygon of the 2D local map of the area available for navigation consists in shrinking all polygons by the radius of the robot and taking the intersection of their areas. This ensures that the whole body of the robot can go in a straight line to any point inside the intersection polygon.

An example of a polygon obtained using the described techniques is shown in Figure 5.c. The polygon shown in the figure is extracted from the occupancy model shown in Figure 5.b using a threshold of $m = 0.6$.

Obstacle and exploration points

The polygon boundary far from the robot is most susceptible to error. This could be the result of absence of depth data or error in depth estimation. This led us to upper-bound the distance to the periphery of the polygon from the robot position (e.g. with distance at most one meter). Thus, the polygon's points lie on a circle of radius one meter, except where the occupancy of the environment causes incursions into the circle.

The points of the navigation polygon which lie inside the bounding circle represent obstacles. Obstacle points are defined for evenly spaced angles in angular sectors where obstacles occur. When making a decision where to navigate, these points will be given negative reinforcement values to keep the robot from hitting the obstacles.

Points of the navigation polygon on the periphery of the one meter circle represent points where the knowledge of the environment becomes undependable, so further data must be collected near them. A collection of exploration points is defined at evenly spaced angles in sectors where the navigation polygon lies on the circle. The density of exploration points is chosen to be adequate to find channels through which the robot could pass, but which may not be observable from the current robot position. Exploration points have positive reinforcement values, thus encouraging the robot to move near to exploration points.

More details on using obstacle and exploration points for planning the navigation can be found in [1].

Figure 4: *Robot Boticelli exploring the room.*

5 Discussions

We described the technique for representing the occupancy world model in a parametric way — using equations of linear surfaces. We showed that the parametric representation of the occupancy function can be very efficiently achieved and then used for world exploration. The proposed technique is implemented on the mobile robot Boticelli. The data obtained from actual runs of the robot are shown in Figure 5. In our application, the robot explores a room shown in Figure 4 in order to find a goal hidden behind a wall, which is a green triangle glued on white paper seen on the back wall in the figure. During the course of exploration, in each of its locations, the robot acquires the range data, which is then converted to a 3D occupancy world model. The constructed model is then used to provide the robot with the navigation map consisting of the list of obstacle and exploration points, which is used by the robot to decide where to go. A reinforcement learning method is applied at this stage.

With the technique described in the paper the robot was able to find the goal, while maintaining the world model. We used less than 32 linear pieces in the occupancy function representation. This appeared to be sufficient for the problem and did not take much time to evaluate. More specifically, it takes approximately the same amount of time to build a model as it takes to collect the range data in the vision stage, that is, about one minute on a Pentium Pro 200 MHz computer.

There was an assumption made about the environments the robot is exploring that there are visual features present all around the robot. In our experiments this is achieved by putting camouflage clothes on otherwise featureless walls, which can be seen in Figure 4. If there are no features available in a part of the environment, then, because of the linear regression fitting, this could result in undesirably high occupancy values in that part. This situation however can be handled by generating a few samples points all over the space with occupancy values equal zero.

Another thing to be mentioned is that the visual range data we used in the experiments contain a lot of imprecise data. Approximately 5% of the data are estimated to be outliers. This is another reason why we build a coarse model of the world, i.e. with only few linear pieces. However, if range data are obtained by more robust range sensors like laser range finders [5, 11], for instance, then there is reason to believe that the proposed piece-wise linear representation of the occupancy function would yield a better approximation of the world, if more linear pieces are used in regression.

Since the number of sample points used in building the occupancy model of an environment does not depend on the scale of the environment, it can be assumed that the proposed technique of representing occupancy models can be used equally well for representing environments at different scales. This is currently under investigation as well as the question of how to combine two local occupancy models obtained from two different locations.

Acknowledgments

The support by the Defense Research Establishment Suffield (contract W7702-6-R594/001) and by the Natural Sciences and Engineering Research Council of Canada (OGP 0172860) is gratefully acknowledged.

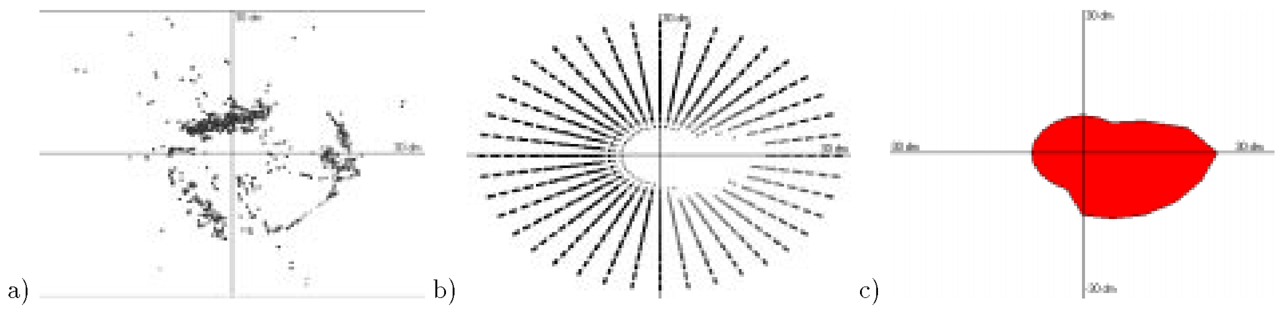

Figure 5: *Range data (a), the occupancy model (b) and the navigation polygon (c) obtained from these data.*

References

[1] W.W. Armstrong, B. Coghlan, and D.O. Gorodnichy. Reinforcement learning for robot navigation. In *International Joint Conference on Neural Networks (IJCNN'99) proceedings, Washington DC, July 21-23*, 1999.

[2] W.W. Armstrong and M.M. Thomas. Adaptive logic networks. *Handbook of Neural Computation, Section C1.8, IOP Publishing and Oxford U. Press*, 1997.

[3] J. Borenstein, H.R. Everett, L. Feng, and D. Wehe. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems, Special Issue on Mobile Robots*, 14(4):231–249, 1997.

[4] J. Buhmann, W. Burgard, A.B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun. The mobile robot rhino. *AI Magazine*, 16:1, 1995.

[5] S.F. El-Hakim, P. Boulanger, F. Blais, J.-A. Beraldin, and G. Roth. A mobile system for indoors 3-d mapping and positioning. In *Proceedings of the Optical 3-D Measurement Techniques IV, pp. 275-282*, 1997.

[6] Dmitry O. Gorodnichy. On using regression in range data fusion. In *Canadian Conference on Electrical and Computer Engineering (CCECE'99) proceedings, May 9-12, Edmonton*, 1999.

[7] D.O. Gorodnichy and W.W. Armstrong. Single camera stereo for mobile robots. In *Vision Interface (VI'99) conference proceedings, May 18-21*, 1999.

[8] C. Jennings and D.Murray. Stereo vision based mapping and navigation for mobile robots. In *Proc. IEEE International Conference on Robotics and Automation, pp. 1694-1699*, 1997.

[9] M.C. Martin and H.P. Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, CMU RI, 1996.

[10] D. Pagas, E. Nebot, and H. Durrant-Whyte. An evidential approach to probabilistic map-building. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 165–169, 1995.

[11] P. Payeur, P. Hebert, D. Laurendeau, and C.M. Gosselin. Probabilistic octree modeling of a 3d dynamic environment. In *Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1289-96*, 1997.

[12] R.S. Sutton and A.G. Barto. *Reinforcement Learning - An Introduction.* MIT Press, 1998.

[13] S. Thrun. An approach to learning mobile robot navigation. *Robotics and Autonomous Systems, 15: 301-319*, 1995.

[14] J. van Dam, B. Krose, and F. Groen. Neural network application in sensor fusion for an autonomous mobile robot. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 263–277, 1995.

[15] Frans Voorbraak. Reasoning with uncertainty in ai. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 52–90, 1995.

[16] B. Yamauchi, A. Schultz, and W. Adams. Mobile robot exploration and map-building with continuous localization. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium*, 1998.