

N90-29064

Building an Environment Model Using Depth Information*

Y. Roth-Tabak and Ramesh Jain

Artificial Intelligence Lab

Electrical Engineering and Computer Science Department

The University of Michigan, Ann Arbor MI 48109

Abstract

Modeling the environment is one of the most crucial issues for the development and research of autonomous robot and tele-perception. Though the "physical robot" operates (navigates and performs various tasks) in the *real world*, any type of reasoning, such as situation assessment, planning or reasoning about action, is performed based on information in its *internal world*. Hence, the robot's intentional actions are inherently constrained by the models it has. These models may serve as interfaces between sensing modules and reasoning modules, or in the case of telerobots serve as interface between the human operator and the distant robot.

A robot operating in a known restricted environment may have *a priori* knowledge of its whole possible work domain, which will be assimilated in its *World Model*. As the information in the *World Model* is relatively fixed, an *Environment Model* must be introduced to cope with the changes in the environment and to allow exploring entirely new domains.

We introduce here an algorithm that uses dense range data collected at various positions in the environment to refine and update or generate a 3-D volumetric model of an environment. Our model which is intended for autonomous robot navigation and tele-perception, consists of cubic voxels with the possible attributes: Void, Full, and Unknown. We present experimental results from simulations of range data in synthetic environments. The quality of the results show great promise for dealing with noisy input data. The performance measures for the algorithm are defined, and quantitative results for noisy data and positional uncertainty are presented.

1 Introduction

Modeling the environment is one of the most crucial issues for intelligent autonomous system development and research. Though a "physical system" operates (navigates and performs various interactive tasks) in the *real world*, any type of reasoning, such as situation assessment, planning or reasoning about action, is performed based on information in its *internal world*. Hence the system's self and intentional action (non accidental or non human supervised) is inherently constrained by the internal models it has or may have of its environment and of the world. By the 'models a system may have' we mean the extent of the power of the representation scheme as opposed to a particular instance of a model which might be partial or incorrect.

The model used by such autonomous agents is referred to as *World Model* (WM), and it represents relatively fixed information about the world in which the system has to work. However, at any given time, only a small portion of the world model, called *environment* is used by an autonomous agent in its operation. The *Environment Model* (EM) at a given time instant should contain more detailed and explicit task-oriented information. The ultimate modeling scheme should consist of hierarchical decompositions on various scales such as a resolution scale and an abstraction scale. The resolution scale allows detailed (high resolution) inspection and reference of parts of the environment as well as a more general (low resolution) view. The abstraction scale, on which sensory data is on one side and a symbolic representation on the other, allows communication in both top-down and bottom up modes. Many researchers [5],[7], [10],[18],[1],[2],[3],[13], [21],[23],[17],[14],[26], have suggested various modeling schemes. In this work we concentrate on the volumetric level of the EM, on which information about *free* and *occupied* space

*This work is sponsored in part by NASA Contract No. 958086

is represented explicitly. At this model level, updating operations can take place using raw sensory data, in case of range sensors, or processed data from any stereo or other depth recovery technique. In addition, although this type of model 'resides' on the very low level of the EM's abstraction hierarchy, it can be used directly by path planning and navigation modules, and object recognition and manipulation modules as well.

Herman and Kanade [10] reconstruct 3D scenes from a sequence of images where 3D wire frame descriptions help to construct surface based models. Chien and Aggarwal [6], Srivastava and Ahuja [25], and Potmesil [22] construct 3 dimensional object models from silhouettes from different views. Elfes [8], and Moravec [19] construct a 2 dimensional map of an environment using sonar readings and a Bayesian probabilistic approach to combine information from various sensor positions. Jain et al [11] use sparse range data provided by any stereo or other depth recovery technique and some worst-case assumption to construct a 3D model. All these techniques rely on relatively 'poor' data (external boundaries of objects, ultrasonic readings or sparse range data) to construct 2D or 3D maps of environments or objects, and hence need a relatively large number of views to obtain reliable models (relative to the complexity of the object or domain to be modeled), and usually require either some worst-case analysis or probabilistic analysis.

In the last few years range sensing technology has come a long way; faster, more accurate, and more reliable systems are now available [4]. In view of the nature of the information in the range images it is considered as a major source for 3-D model generation. As opposed to the methods described above, we consider a dense range sensor as the source of 'rich' information for our technique. In the work by Goldstein et al [9], dense range data was also used as a source for updating a world model, however, they were using a Combinatorial Geometry (CG) model (also known as Constructive Solid Geometry CSG) for describing objects in a scene, in particular they were using spheres as their initial building blocks. In our system we partition the space into a 3D matrix of cubic voxels, which imposes a different implementation and possible use. We introduce here an algorithm that uses dense range data from multiple viewpoints in an environment to refine a 3-D volumetric model of that environment. The locations and orientations of the sensor which are used for the updating process will be determined using reasoning, however, here the emphasis is on information assimilation. Our model is intended for autonomous robot navigation and tele-perception, and each voxel may have one of three attributes: Void, Full, or Unknown. The third attribute *Unknown* represents the voxels in space of which no information is known, and it can help guide an exploring module to locate a next good position for observation.

At the present we assume a static environment and the work described here follows this assumption, however, we intend to be able to relax this assumption and cope with dynamic environments using conflicting information between expected scene and a viewed scene. Such comparisons between expected and viewed scene may be also used to estimate the real position of the sensor.

1.1 Organization of the article

The following sections describe the method and some experimental results with simulated data. Section 2 describes the modeling scheme and the method to create and update the model. In this section a general review on World and Environment Models is introduced as well. In section 3 experimental results are described. These experiments include a generation of an environment volumetric model from several viewpoints, and a comparison with results obtained using noisy data and location uncertainty. In section 4 an evaluation of the results is provided followed by some conclusive remarks regarding the method's performance in different conditions. Section 5 describes possible extensions and directions for future work, and Section 6 provides a brief conclusion.

2 The Model and the Method

2.1 The model

We concentrate here on the lower abstraction levels of the EM, to demonstrate the use of sensory data to initiate a model. The model is a 3D volumetric grid of cubic voxels. The voxels are assigned three possible values: VOID - for empty voxels, FULL - for occupied voxels, and UNKNOWN for voxels for which no meaningful information has been obtained yet.

Other researchers such as [20] use different attributes in their grids and it might be claimed that the notion of Unknown may be captured by using uncertainties. We claim, however, that these notions are distinct and each

has its own importance in such models. These claims resemble claims of the Shafer-Dempster formalism [24] where Belief Functions are introduced to represent the "strength of the evidence" that specifically favors some proposition, in contrast to the Bayesian approach in which a unit of probability must be apportioned between the possible propositions.

Uncertainty provides information about the degree of confidence assigned to a certain piece of information about a voxel. The attribute *Unknown*, on the other hand, declares that there is no previous information available about a voxel. This attribute may be the key to a decision module searching for the next position for the sensor so as to capture as much as possible new or required information from a domain.

As opposed to [8],[19],[20] in our model we avoided assigning certainty levels to the attributes for the following reasons: 1. Range data produced by a dense range sensor is used, as opposed to ultrasonic sensors which usually have a 'wide' opening angle and impose more uncertainty on the location of the actual obstacles. 2. Dense range data is used which, as opposed to various stereo and depth recovery techniques from grey level images, provides a range reading for all the pixels in the image, and hence there are no 'spatial gaps' of depth information. 3. The updating technique is based on intersection with the previous model which speeds the operation as the model is being constructed; working with certainty levels would force us to scan the whole grid for every updating step. 4. This method treats the uncertainties and errors within itself, and provides a simple model, free of uncertainties, for use by other modules. However, we do not argue against using uncertainties, but merely point out to the fact that for the specific engineering task presented a model without uncertainty is sufficient.

2.2 Updating the Model

The model is initially entirely UNKNOWN. The position and orientation of the robot with respect to some general frame are assumed to be known for each view, however, some location uncertainty can be tolerated.

The operations performed include various transformations from a fixed Cartesian, coordinate system to a translated and rotated coordinate system, representing the sensor coordinate system. These transformations are used to determine the location of a point given in the global coordinate system in the sensor coordinate and vice-versa. Another required transformation is between the sensor Cartesian coordinate system and a sensor spherical coordinate system which represents the actual range image of the sensor.

A global Cartesian coordinate frame was defined as $x, y,$ and $z,$ where zx is the horizontal plane and y is the vertical axis (see figure 1).

The location of the sensor is defined as the coordinate point (z_0, x_0, y_0) in this global coordinate frame. The orientation of the sensor is specified in a sensor centered coordinate frame where z' is the viewing direction and y' is the vertical axis of the sensor's image plane. x' is defined so as to have a right-handed coordinate system (see figure 1).

The first transformation is between the global coordinate frame and the sensor centered Cartesian coordinate frame:

$$\begin{cases} x' = l_1(x - x_0) + m_1(y - y_0) + n_1(z - z_0) \\ y' = l_2(x - x_0) + m_2(y - y_0) + n_2(z - z_0) \\ z' = l_3(x - x_0) + m_3(y - y_0) + n_3(z - z_0) \end{cases}$$

and the inverse transformation:

$$\begin{cases} x = l_1x' + l_2y' + l_3z' + x_0 \\ y = m_1x' + m_2y' + m_3z' + y_0 \\ z = n_1x' + n_2y' + n_3z' + z_0 \end{cases}$$

Where $l_1, m_1, n_1; l_2, m_2, n_2; l_3, m_3, n_3;$ are the direction cosines of the x', y', z' axes relative to the x, y, z axes respectively, and (x_0, y_0, z_0) is the linear translation of the center of the coordinate system.

However, it is desirable to specify the transformations in terms of the *pan, tilt,* and *swing* angles, so that the transformation can be specified in terms of the orientation parameters explicitly available for the sensor.

Here are the direction cosines specified in terms of the *pan*, *tilt*, and *swing* angles where $\alpha = \text{pan}$, $\beta = \text{tilt}$, and $\gamma = \text{swing}$:

$$\begin{aligned}
 l_1 &= \cos \alpha \cos \gamma - \sin \alpha \sin \beta \sin \gamma \\
 l_2 &= -(\cos \alpha \sin \gamma + \sin \alpha \sin \beta \cos \gamma) \\
 l_3 &= \sin \alpha \cos \beta \\
 m_1 &= \cos \beta \sin \gamma \\
 m_2 &= \cos \beta \cos \gamma \\
 m_3 &= \sin \beta \\
 n_1 &= -(\sin \alpha \cos \gamma + \cos \alpha \sin \beta \sin \gamma) \\
 n_2 &= \sin \alpha \sin \gamma - \cos \alpha \sin \beta \cos \gamma \\
 n_3 &= \cos \alpha \cos \beta
 \end{aligned}$$

In addition the following known transformation between the sensor centered Cartesian coordinate system to a spherical coordinate system, in which the sensor is actually defined, is used:

$$\begin{cases} x' = r \sin \theta \cos \phi \\ y' = r \sin \theta \sin \phi \\ z' = r \cos \theta \end{cases} \quad \begin{cases} r = \sqrt{(x')^2 + (y')^2 + (z')^2} \\ \theta = \cos^{-1}\left(\frac{z'}{r}\right) \\ \phi = \sin^{-1}\left(\frac{y'}{r \sin \theta}\right) \end{cases}$$

where θ is the 'opening' angle, and ϕ is the 'rotation' angle.

The Updating Algorithm can be described in words as following

1. Only voxels which are within the scope of the sensor are checked. By 'in the scope of the sensor' we mean both the 'angular' scope, i.e. being inside the cone in front of the sensor, and the 'distance' scope, i.e. being within the maximum range of the sensor.
2. Only voxels which are not yet VOID are checked (this implies an intersection with previous models).
3. For each of the voxels which are actually checked all of the 8 vertices are checked and compared to the actual pixel in the range image which points to their position in space.
4. If the maximum distance to any of the 8 vertices is smaller than the minimum range pointed by any of the range pixels, then that voxel is VOID.
5. If the 'range' of the vertices' distances intersects with the 'range' of the range pixels, and the difference between the maximum and the minimum range pixels is within a certain threshold, then a voxel is FULL.
6. Else it is unchanged.

This method has a few attributes worth mentioning. In step 3, the fact that 8 vertices are being checked has an inherent smoothing effect on the result. In most cases not all vertices will fall within the same range pixel (This depends on the size of the voxels and the resolution of the range image and can be guaranteed by controlling the size of the voxels), and hence noisy images to a certain extent will not have a strong impact on the result. In step 4, a certain threshold margin may be added to the above requirement in cases where there is some location uncertainty and its extent is known. This margin represents the worst case error which might result from such a location uncertainty. The threshold of Step 5 is introduced to avoid assigning FULL values to voxels which lie on or near sharp range discontinuities. This threshold is not critical since these voxels are not going to be assigned a VOID value, however it helps define the UNKNOWN regions in space and avoids assigning FULL values to the wrong voxels. Another quality of this method is that it is fully parallelizable in a straight forward manner.

3 Experimental Results

The experiments were performed with a 3D volumetric model of dimensions 64X64X16 voxels. However, there is no inherent restriction on the dimensions in the method.

To perform the experiments a synthetic domain was created (see figure 2) which is represented in a similar grid as the model is, with the attributes FULL and VOID only assigned to its voxels.

A range sensor which produces circular range images (see figure 3) was simulated. In the provided range images darker pixels represent closer range, and lighter color pixels represent further range. This range sensor has an opening cone of 60 spatial degrees (30 deg rotated around a symmetry axis). The sensor was described in spherical coordinates and the pixels were located by the 'opening' angle (distance from the center line of view) and the 'rotation' angle. This simulated sensor exhibited a resolution of 128X64 pixels, 128 on the rotation angle and 64 on the opening angle. The pixels values were integers between 0-128. Such a description of 'sensor' is not a common description for range sensors which usually define the parameters as elevation angle and azimuth angle [4]. However, the difference between the descriptions is only in the orientation of the spherical system (Z axis pointing to the viewing direction in the described sensor, as opposed to the Z axis being vertical to the viewing position). This type of description was chosen to produce higher resolution pixels in the center of an image and lower resolution at the image boundaries, and to produce a symmetric image form around its center line of view.. Nevertheless, the 'type' of range sensor (referring to the coordinate system in which the sensor is described) does not affect our method, but would merely require another transformation between the Cartesian coordinate system and the alternative spherical coordinate system. The sensor was 'positioned' in 12 different positions and orientations in the environment and the range images for these views were obtained (see table 1). The position is specified as

No. of View	z	x	y	pan (deg)	tilt (deg)	swing (deg)
1	56	6	10	180	-3	0
2	56	6	10	125	-3	0
3	56	6	10	100	-3	0
4	56	24	10	180	-3	0
5	56	52	10	100	-3	0
6	56	52	10	170	-3	0
7	36	46	10	235	-3	0
8	20	60	10	-20	-3	0
9	18	60	10	-30	-3	0
10	16	16	10	90	-3	0
11	5	28	10	-85	-3	0
12	8	8	10	60	-3	0

Table 1: The positions and orientations of the sensor used in the experiments

a 3-tuple coordinate (x,y,z) relative to a fixed global coordinate frame, and the orientation is specified as a 3-tuple (pan, tilt, swing), where pan specifies the azimuth, tilt is the vertical direction where 0 deg is parallel to the ground, and swing is a rotation angle around the center line of view (this degree of freedom is seldom used nevertheless it is specified to have all the degrees of freedom available). The model which was initially UNKNOWN was then updated using these range images.

Experiments were also performed with the same data with added noise and with some location uncertainty. The noise that was added was a pseudo-Gaussian noise with σ (which is specified in the results) as the square root of the variance. When location uncertainty was added to the experiment, the same images were used but the algorithm used wrong position information. This position was produced by a pseudo-uniform random generator on the range 0-radius (The diameter is specified as a parameter in the results) from the actual position, and a pseudo-uniform random generator for the angle from the actual position on the range 0-2 Π .

To evaluate the results numerically three parameters were introduced: *quality level*, *acquaintance level*, and *error level*. In the description of these parameters The 'EM' refers to the volumetric model that was generated using the Updating algorithm, and 'The No. of voxels' referenced in the denominator of the following expressions refers to the number of the (FULL or VOID) voxels in the real domain.

$$\text{quality level} = \frac{\text{No. of correct VOID voxels in the EM}}{\text{Total No. of VOID voxels}}$$

$$\text{acquaintance level} = 1 - \frac{\text{No. of UNKN EM voxels VOID in real domain}}{\text{Total No. of VOID voxels}}$$

$$\text{error level} = \frac{\text{No. of wrong VOID voxels in the EM}}{\text{Total No. of FULL voxels}}$$

All these parameters are presented in percentile. The *quality level* represents the percentage of the free space which was correctly found. For robot navigation and path planning this is an important aspect for evaluating the method. Since the goal is to correctly identify the clear passages in a domain, the percentage of the VOID voxels that were found gives a quantitative estimate of the quality of the model. The *acquaintance level* is 100 – (minus) the percentage of the UNKNOWN voxels which are actually VOID. This parameter represents the level of acquaintance the robot-model has with the environment and helps to evaluate the performance with regard to the number of views taken. The *error level* represents the percentage of wrong VOID voxels to total number of FULL voxels. This parameter actually specifies the level of confidence we can assign to the model's accuracy. The two parameters of quality and error combined are obviously the bottom line for evaluating the model.

The results for a lower resolution model were evaluated in the same manner too. The reason for this additional evaluation is that the above parameters do not specify the distribution of the wrong or correct voxels in the model. Moving to one lower level of resolution allows to evaluate the extent of the quality and of the errors in the model with regard to navigation and path planning in an additional perspective.

As for the time it takes to generate the model, since we were interested more in competence issues than in computational issues only rough run time measurements were performed, the results are still quite fast. The times provided here are real computer run time and not CPU time. We ran it on an Apollo DN4000 workstation, and for the first 3 views it took between 45-60 sec. to update the model, and then the time for updating the model drops down to 15-20 sec. for the last views. Considering the possibility of parallelism, since the algorithm is performed serially on all of the voxels, the increase in speed will be bound by the extent of the parallel machine. This means that any parallel machine can be used to its full potential with minor modifications to the algorithm.

4 Discussion

The results for added noise show that the method is not susceptible to noise. In full resolution up to noise levels of normal standard deviation $\sigma = 3$ an error level up to 1% only is produced. Furthermore, at the lower resolution level the error level drops below 1% even for high noise levels of $\sigma = 8$. As the range image pixel values are between 0-128, such noise levels are enormous. This resistivity to noise was expected due to the natural smoothing which is performed implicitly (see table 2). As for location uncertainty, at full resolution only up to diameter = 3 produces acceptable errors. At the lower resolution up to diameter = 4 the error levels are still low, but then they rise sharply, and the effect of the lower resolution we had on the noise does not appear here (see table 3). As mentioned before, a threshold may be added to deal with location uncertainty. However, there will be a trade-off between the quality level and the error level when adding this threshold.

5 Future Work

The overall performance of this method shows a lot of promise for producing an adequate volumetric model of an environment as exhibited in the results presented here. We intend to pursue this research in the following directions:

- Investigate this method with real data. This will involve both adapting the transformations to the geometry of a specific real range sensor, and also collecting the actual data in a controlled environment where the results can be properly evaluated.
- Investigate additional methods for dealing with location uncertainty. We think that an adaptive technique which changes its mode of operation according to the level of acquaintance with the environment may provide the desirable results.

- Investigating possible optimizations of the actual algorithm. The algorithm which was used to generate the results presented here was written with the intent to check its competence for the task of updating a volumetric model. We would like to investigate it from computational aspects too.
- Investigate algorithms and methods for guiding an autonomous agent carrying the range sensor to a potentially informative new position and orientation. The idea is to use the UNKNOWN voxels as indicators for parts of the space which should still be explored.
- Investigate dynamic domains. This must involve additional higher level information such as attributes for static voxels, potentially mobile voxels, and actively moving voxels.

6 Conclusion

We presented here a method for generating, refining and updating a volumetric Environment Model of a domain using dense range data. Such a method may be used by an autonomous intelligent system for navigation and path planning, as well as for object recognition and manipulation, or for transferring spatial information from a remote sensing agent to its operator. Performance measures for the algorithm were defined, and quantitative results with noisy data and positional uncertainty were provided. Both the quality of the results, the stability of the method under noisy conditions, the relative speed of computation, and the real 3D quality of the information acquired, demonstrate the potential of the method for autonomous intelligent systems.

Though at the present we are assuming a static environment, we intend to be able to cope with dynamic environments using conflicting information between expected scene and viewed scene. Such comparisons between expected and actually viewed scene may be also used to estimate the real position of a sensor under location uncertainty conditions.

References

- [1] K. M. Andress and A.C Kak, "A Production System Environment for Integrating Knowledge with Vision Data", *Proc. of the AAAI Sensor Fusion Workshop*, pp. 1-11, 1987.
- [2] Anthony Richard, "Spatial Reasoning Using an Object Oriented Spatial DBMS", *Proc. of the AAAI Sensor Fusion Workshop*, pp. 42-51, 1987.
- [3] Asada Minoru, "Building a 3-D World Model for a Mobile Robot from Sensory Data", *Report CAR-TR-332 CS-TR-1936 University of Maryland*, October 1987.
- [4] Paul J. Besl, "Active, Optical Range Imaging Sensors", *Machine Vision and Application (1988) 1*, pp. 127-152 1988.
- [5] Rodney A. Brooks, "Visual Map Making for a Mobile Robot", *Proc IEEE Int. Conf. on Robotics and Automation*, pp.824-829 1985.
- [6] C.H. Chien and J.K. Aggarwal, "Identification of 3D Objects from Multiple Silhouettes Using Quadtrees / Octrees", *Computer Vision, Graphics, and Image Processing* 36, pp. 256-273 1986.
- [7] James L. Crowley, "Representation and Maintenance of a Composite Surface Model", *Proc. of the 1986 IEEE International Conf. on Robotics and Automation*, pp. 1455-1462, April 1986.
- [8] Alberto Elfes, "Sonar-Based Real-World Mapping and Navigation", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No.3, pp 249-265, June 1987.
- [9] M. Goldstein, F.G. Pin, G. de Saussure, and C.R. Weisbin, "3-D World Modeling With Updating Capability Based on Combinatorial Geometry", *Proc. of the Workshop on Space Telerobotics*, pp. 273-282, 1987.
- [10] Martin Herman and Takeo Kanade, "Incremental Reconstruction of 3D scenes from Multiple, Complex Images", *Prager, USA*, 1981.

- [11] R. Jain, Y. Roth-Tabak and K. Skifstad, "Hyperpyramids for Vision-Driven Navigation", *Proc. AAI-VI (Orlando)*, April 1988.
- [12] Stephen Kaplan, and Rachel Kaplan, "Cognition and Environment, Functioning in an Uncertain World", *Proc. of the AAAI Workshop on Sensor Fusion*, pp. 390-404, 1987.
- [13] Benjamin J. Kuipers and Y. T. Byun, "A Qualitative Approach to Robot Exploration and Map-Learning", *Proc. of the AAAI Workshop on Sensor Fusion*, pp. 390-404, 1987.
- [14] Benjamin J. Kuipers and Tod S. Levitt, "Navigation and Mapping in Large Scale Space", *AI Magazine Vol.9 No.2*, pp. 25-43, 1988.
- [15] Daryl T. Lawton, Tod S. Levitt, Christopher C. McConnel, Philip C. Nelson, and Jay Glicksman, "Environmental Modeling and Recognition for an Autonomous Land Vehicle", *Proc. of the Workshop on Space Telerobotics*, pp.313-336, 1987.
- [16] Tod S. Levitt, Daryl T. Lawton, David M. Chelberg, Philip C. Nelson, and John W. Dye, "Visual Memory Structure for a Mobile Robot", *Proc. of the AAAI Sensor Fusion Workshop*, pp. 92-105, 1987.
- [17] Tod S. Levitt, Daryl T. Lawton, David M. Chelberg, Kerry V. Koitzsch, and John W. Dye, "Qualitative Navigation II", *Proc. of the DARPA Image Understanding Workshop*, pp. 319-326, 1988.
- [18] Herman Martin and Takeo Kanade, "Incremental Reconstruction of 3D Scenes from Multiple Complex Images", *Artificial Intelligence* 30 pp. 289-341, 1986.
- [19] H.P. Moravec, "Certainty Grids for Mobile Robots", *Proc. of the Workshop on Space Telerobotics*, pp. 307-312, 1987.
- [20] Hans P. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots", *AI Magazine Vol.9 No.2*, pp. 61-74, 1988.
- [21] I.J. Oppenheim, D.R. Rehak, W.T. Keirouz and R.F. Woodbury, "Robotic Task Planning, Domain Modeling and Geometric Reasoning", *EPRI NP-5525 Project 2515-1 Final Report*, December 1987.
- [22] Michael Potmesil, "Generating Octree Models of 3D Objects from Their Silhouettes in a Sequence of Images", *Computer Vision, Graphics, and Image Processing* 40, pp. 1-29 1987.
- [23] Nageswara S.V. Rao, S.S. Iyengar B. John Oommen and R. L. Kashyap, "On Terrain Model Acquisition by a Point Mobile Robot Amidst Polyhedral Obstacles", *IEEE Journal of Robotics and Automation, Vol.4, No.4*, pp. 283-288, August 1988.
- [24] Glen Shafer, "A Mathematical Theory of Evidence", *Princeton University Press*, 1976
- [25] Sanjay K. Srivastava and Narendra Ahuja, "An Algorithm for Generating Octrees From Object Silhouettes in Perspective Views", *Proc. of the IEEE Workshop on Computer Vision*, pp. 363-365, Nov 30 - Dec 2 1987.
- [26] Ellen L. Walker, Martin Herman, and Takeo Kanade, "A Framework for Representing and Reasoning about Three-Dimensional Objects for Vision", *AI Magazine Vol.9 No.2*, pp. 47-58, Summer 1988.
- [27] Wai K. Yeap, "Towards a Computational Theory of Cognitive Maps", *Artificial Intelligence* 34, pp. 297-360, 1988.

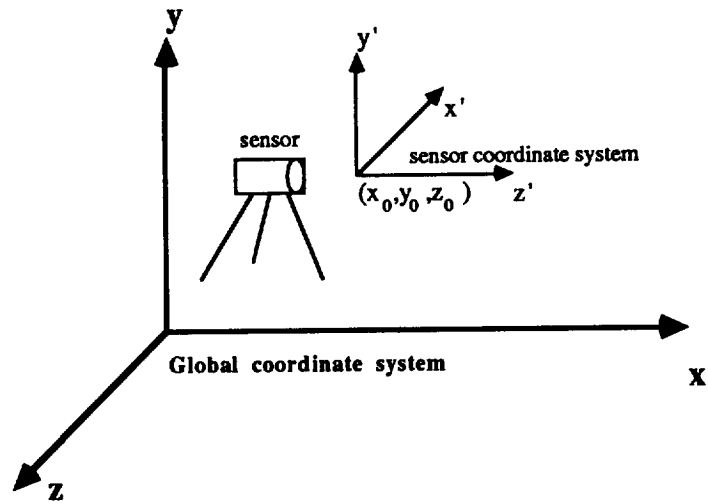


Figure 1: Transformation & rotation of Cartesian coordinates

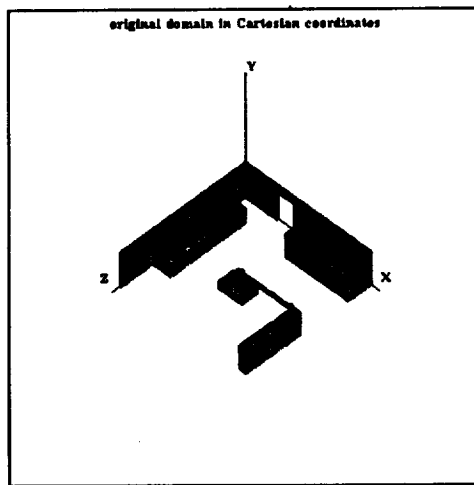


Figure 2: The synthetic domain

σ	Quality level %	Acquaintance level %	Error level %
0	94.7	99.9	0
1	94.5	99.9	0
2	93.2	99.9	0.07
3	92.2	99.9	0
4	91.5	99.9	0
5	90.9	99.9	0.07
6	90.1	99.8	0.13
7	89.8	99.5	0.13
8	89.3	98.9	0.53

Table 2: The effect of noisy range images on the model at a lower resolution

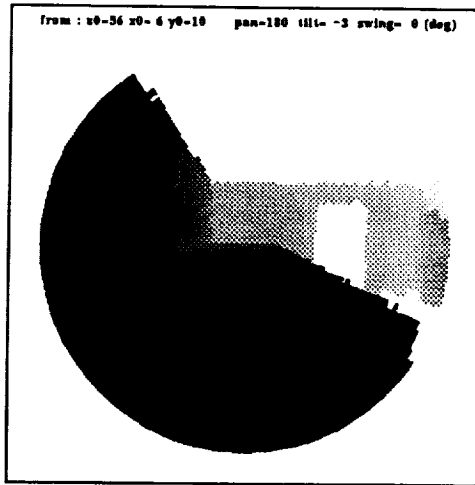


Figure 3: The synthetic range image from the first position

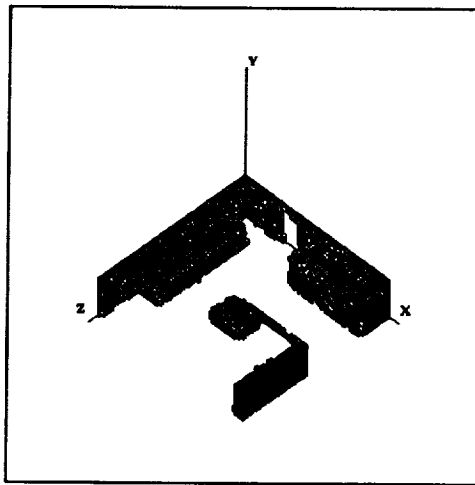


Figure 4: The model after 12 views

diameter of uncertainty	Quality level %	Acquaintance level %	Error level %
0	94.7	99.9	0
1	95.1	99.9	0
2	96.8	100.0	0
3	97.6	99.9	0.20
4	96.8	99.9	0.86
5	95.7	99.8	5.32
6	97.2	99.9	7.98
7	97.4	99.7	9.51
8	95.3	98.8	10.24

Table 3: The effect of location uncertainty on the model at a lower resolution

ROVERS

