108

158110

3-D World Modeling With Updating Capability Based on Combinatorial Geometry

M. Goldstein, F.G. Pin, G. de Saussure, and C.R. Weisbin
Oak Ridge National Laboratory
Oak Ridge, TN 37831

ϕA 789470

1. Abstract

This paper describes a 3-D world modeling technique using range data. Range data quantify the distances from the sensor focal plane to the object surface, i.e., the 3-D coordinates of discrete points on the object surface are known. The approach proposed herein for 3-D world modeling is based on the Combinatorial Geometry (CG) Method which is widely used in Monte Carlo particle transport calculations. First, each measured point on the object surface is surrounded by a small sphere with a radius determined by the range to that point. Then, the 3-D shapes of the visible surfaces are obtained by taking the (Boolean) union of all the spheres. The result is an unambiguous representation of the object's boundary surfaces. The "pre-learned" partial knowledge of the environment can be also represented using the CG Method with a relatively small amount of data. Using the CG type of representation, distances in desired directions to boundary surfaces of various objects are efficiently calculated. This feature is particularly useful for continuously verifying the world model against the data provided by a range finder, and for integrating range data from successive locations of the robot during motion. The efficiency of the proposed approach is illustrated by simulations of a spherical robot in a 3-D room in the presence of moving obstacles and inadequate pre-learned partial knowledge of the environment.

2. Introduction

An autonomous robot must have sensory capability to deal with unknown or partially known environments. The sensor derived data need to be processed to an appropriate internal representation of the external world. The external world to be described is fundamentally three-dimensional, involving object occlusion. Most computer vision research performed during the past twenty years has concentrated on using intensity images as sensor data. The imaging hardware (cameras) for these studies typically project a three-dimensional scene onto a two-dimensional image plane, thus providing a matrix of gray level values representing the scene from a given viewpoint. These values indicate the brightness at points on a regular spaced grid and contain no explicit information about depth. Methods that use intensity information only for deriving 3-D structure are usually computationally intensive. This computationally expensive processing arises due to the fact that correspondence of points between different views must be established and a complex system of nonlinear equations must be solved([1]-[5]).

In recent years digitized range data have become available from both active and passive sensors, and the quality of these data has been steadily improving([6]-[8]). Range data quantify the distances from the sensor focal plane to an object surface. Since depth information depends only on geometry and is independent of illumination and reflectivity, intensity image problems with shadows and surface markings do not occur. Therefore, the process of representing 3-D objects by their shape should be less difficult in range images than in intensity images. The problem addressed by this paper is the external world modeling using range data. Unique requirements for such a model are:

a) Allow representation of a general 3-D unknown or partially known environment, based on range data.
b) Allow for minimal fast memory for storage.
c) Allow the introduction of a feedback loop for continuous verification of the world model against the data provided by the sensor (efficient distance calculation).
d) Allow for efficient integration of the range data from multiple views.
e) Allow for efficient navigation and manipulation.

A wide variety of techniques have been developed for representing 3-D objects for digital computing purposes. There are methods which describe the surface boundary and methods which represent the object's volume. The simplest boundary representation is using n-sided planar polygons (triangles, quadrilaterals, etc.) which can be stored as a list of 3-D node points along with their relationship information. Arbitrary surfaces are approximated to any desired degree of accuracy by using many planar polygons. This type of representation is popular because model surface area is well defined and all object operations are carried out using piecewise-planar algorithms. The next step in generality is obtained using quadric surface boundary representation. More advanced techniques for representing curved surfaces with higher order polynomials or splines are mentioned in the computer graphics and CAD literature([9]-[12]). There are many different techniques of this type; however

†On leave from the Nuclear Research Centre-Negev, Israel.

they are generally not very compact in terms of data storage, nor are they computationally efficient in calculating distances to boundary surfaces[13]. The best known volumetric representations are the oct-tree[14] generalized cylinder[15] and multiple 2-D projection views[16]. The generalized cylinder approach is well suited to many real world shapes. However, it becomes just about impossible to use this representation for large, thin objects. The multiple 2-D projection view is not a general purpose technique since different objects may have similar 2-D projections. The oct-tree representation is used in many world models. However, the indexing problem[17] is seriously affecting the efficiency of this tecnnique. In conclusion there is a need for a fast and robust technique for building 3-D models of arbitrarily shaped objects.

In Chapter 3, a proposed external world modeling procedure and an efficient distance calculation algorithm are presented. A technique for integrating the range data from multiple views and a continuous verification procedure of the world model versus the range data provided by the sensor is illustrated in Chapter 4. Finally, the feasibility of the proposed approach is illustrated in Chapter 5 by simulations of a spherical robot navigating in a 3-D room in presence of static and moving obstacles and inadequate pre-learned partial knowledge of the environment.

3. Representing 3-D Surfaces Using the Combinatorial Geometry

The basic problem addressed in this paper is one of representation. The proposed approach is based on the Combinatorial Geometry (CG) method[18] which is widely used in Monte Carlo simulation of particle transport in 3-D geometries. In CG (also known as Constructive Solid Geometry (CSG) in computer graphics and CAD literature) solids are represented as combinations of primitive solids or "building blocks" (i.e., spheres, cylinders, boxes, etc.) using the Boolean operations of union, intersection and difference. The storage data structure is a binary tree where the terminal nodes are instances of primitives and the branching nodes represent Boolean operators. Any 3-D known object can be represented by a (Boolean) combination of primitive solids or deformed superquadrics[19]. This representation is especially suitable for describing pre-learned partial knowledge of the environment. An example of describing an object composed of two boxes, one of them with a cylindrical hole is illustrated in Fig. 1. The result is a concise, unambiguous and complete representation of the object volume and boundary surface.

ORNL-DWG 86-16942

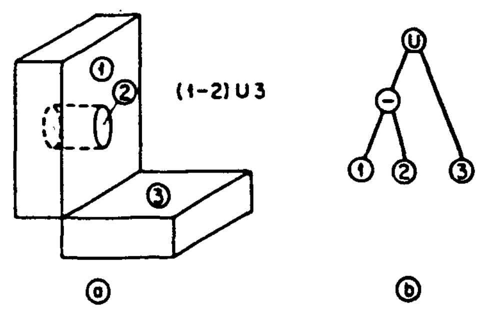


Fig. 1. Representing a 3-D object using Combinatorial Geometry.
a - given object and its CG representation.
b - the storage data tree.

The result of a range scan is a matrix of distances from the sensor focal plane to an object surface. In other words, the coordinates of discrete points on the "visible" parts of the boundary surfaces of different objects in the external world of the robot, are known. Let be the (small) angle between two successive "reading" directions of the sensor. First, each discrete point i, is surrounded by a small solid sphere with a radius, $r_i = \max(R_i \sin \alpha, \Delta R_i)$, where R_i is the range to point i, and ΔR_i is the associated measurement error. Then, the approximate 3-D shape of the visible boundary surfaces is obtained directly by taking the union of all the spheres (see Fig. 2).

ORNL-DWG 86-16943

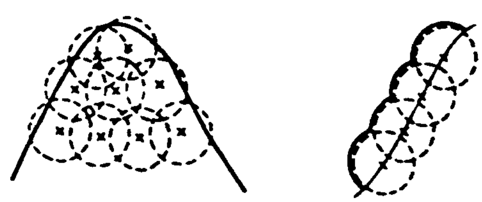


Fig. 2. Describing the shape of 3-D objects using spheres.

The reason for using spheres is to keep the representation as compact as possible. Describing the sphere for a particular discrete point in space means adding only one additional parameter (the radius) to the coordinates of the discrete point which are provided by the sensor. The radius r_i is defined as $r_i = \max(R_i \sin \alpha, \Delta R_i)$ to avoid the appearance of "holes" in the geometry and to take into account the range uncertainty. Using this definition for r_i, neighbor spheres overlap one another and the boundary surface of the union of all spheres is continuous (without holes) from the robot's point of view. Finally, it is obvious that using the "sphere" procedure, the shape of the boundary surfaces is distorted. However, the distortion is practically proportional to the range to each point since the range uncertainty is relatively small. In other words, the resolution of the model is improved as the range to the surface is decreased.

3.1. Distance Calculation in CG Representation

A very useful feature of the CG representation is its efficiency in calculating distances to 3-D surfaces in a desired direction. Observing discontinuities in the range data greater than the maximum size of the robot, the scene is partitioned in many different zones. A zone is defined as the union of small spheres located between two successive discontinuities in the range data. Using the storage data structure mentioned in Chapter 3, two tables are defined: the first one includes the spatial location of the small spheres; the second one identifies the different zones in terms of these spheres. The distance to 3-D surfaces in a desired direction from a given point, is calculated in two steps:

a) Each zone is surrounded tightly by a box (rectangular parallelepiped). Since the boxes are approximate bounding configurations, intersections of a given ray with a box does not necessarily imply intersection with any particular sphere. In addition, the different orientations of the sphere clusters imply that bounding boxes can intersect and therefore multiple boxes may have to be checked for penetration by a given ray. The box (boxes) penetrated by the ray is determined by calculating the intersection points between the boxes and the ray. A list consisting of the boxes physically penetrated by the ray, is defined. The corresponding list of zones is used to determine the penetration point.

b) Determine the small sphere penetrated by the ray and calculate the penetration point. This is done by considering only the spheres included in the zones listed in the first step. Using this approach, only a small number of spheres are checked for penetration, and therefore significant computation time is saved.

It should be mentioned that the boxes surrounding the zones are used only internally during distance calculations and they are not affecting the geometric description of the 3-D surfaces. During path planning, "tentative paths" are checked for potential collision by calculating the distances to object surfaces from scattered points on the robot's surface in the desired direction. These distances can be effectively calculated by using the CG representation, and the procedure outlined above.

4. Updating the World Model

Automatic construction of 3-D models of objects from multiple views is an important problem in computer vision. In the past, a number of different techniques have been used for representation and modeling of 3-D objects for computer vision applications([20]-[27]). However, there is an absence of a fast and robust technique for building 3-D models of arbitrarily shaped objects. The process of constructing 3-D models for objects involves integrating the range data from multiple views. In general, the integration process performs matching to establish correspondence between the views, determines the interframe transformations to register the views in a common reference coordinate system and then merges the data. The difficult and time consuming step in the above process is the matching step required to establish a correspondence. Much of the previous research efforts have been directed toward solving the difficult correspondence problem. The algorithm presented in this paper, does not require any correspondence between different views, because the world model uses a universal coordinate system with the origin arbitrarily located at the robot's initial position. According to the representation algorithm described in Chapter 3, the accuracy with which a certain point in space may be observed by the robot depends upon the distance between the robot and the point. This fact is translated to the radius of the sphere surrounding a particular point in the CG representation. Therefore, a point in space should be kept in memory along with the most accurate information (shortest observation distance). In other words, for each "measured point" in space, the shortest observation distance in the robot's history should be determined. The main problem in implementing this approach is the fact that since the sampling procedure of range data is discrete, the probability of a particular point to be sampled from two different positions of the robot is zero. In other words, each "measured point" is sampled just once during the robot history. The solution implemented in our approach follows an iterative algorithm using the "old data" acquired before the current scan and the "new data" acquired during the current scan:

a) The "old data" is checked from the current position of the robot. Using the world model based on the "old data", distances to 3-D surfaces from the robot's current position in the direction of points in the "old data" are calculated. If the distance to 3-D surfaces is smaller than the euclidean distance to the sphere surrounding the point then this particular point cannot be seen from the current position of the robot and the point representation is kept unchanged. Otherwise, the "old" radius of the surrounding sphere is compared with the "new" radius determined by the euclidean distance from the current position of the robot and the smallest radius is chosen between the old and the new radii.

b) The "new data" acquired from the current position of the robot is checked against the "old data". If the "new" point (provided by the sensor) is located within the world model based on the old data (within a sphere surrounding an "old" point) then the new point is rejected and therefore no new sphere is added to the world model.

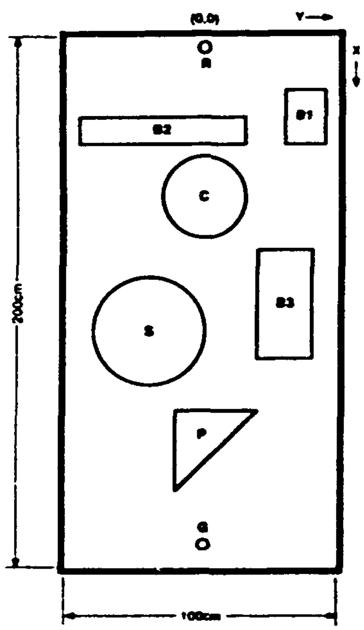
If the "new" point is outside the old world model, then the distance to 3-D surfaces in the "new" point direction is calculated (using the old world model). If the obtained distance is greater than the range to the "new" point (provided by the sensor), the "new" point is added to the world model as a sphere with a radius determined by the range to the point. Otherwise (the obtained distance is smaller than the range to the point) the "old" point (located approximately in the same direction, but closer to the robot) is erased from the model and the "new" point is added to the world model. This is the case of moving objects, in which the "old" data should be continuously verified and updated.

c) Verification of pre-learned geometric knowledge of the environment.

The sensor derived data is compared with the calculated distances obtained from scanning the pre-learned geometric environment. The pre-learned data is represented in a very concise way using the CG representation. If the "real" range in a certain direction is found to be similar (within the uncertainty of the pre-learned data) to the calculated range in the same direction, the representation is kept unchanged and no point is added to the world model. If the real range is smaller than the calculated range, the new real point is added to the world model. Finally, if the real range is greater than the calculated range, the entire pre-learned object is removed from the world model and the "real" point is added to the representation.

5. Sample Problems

The efficiency of the proposed world model is illustrated in several simulations of a spherical robot navigating in a 3-D room in presence of static and moving obstacles and inadequate pre-learned partial knowledge of the environment. The robot is assumed to move in a plane parallel to the floor, along straight lines. The origin of coordinates is arbitrarily located at the robot's starting position. The goal coordinates are known a-priori. The external world geometry, the robot starting position and the goal location are illustrated in Fig. 3. The radius of the spherical robot is 3 cm. The plane of motion is 30 cm off the floor. The navigation algorithm used in the sample problems is described in detail in Ref. 28.



R - The initial position of the robot (4,0,30)

G - The goal position (190,0,30)

B1 - BOX; dimensions (20 x 15 x 40); Center at (30,37.5,20)

B2 - BOX; dimensions (10 x 60 x 90); Center at (-15,35,45)

B3 - BOX; dimensions (40 x 20 x 90); Center at (100,30,45)

C - CYLINDER; Center of basis at (60,0,0)
Height 60, Radius 15

S - SPHERE; Center at (110,-20,30)
Radius 20

P - PRISM; dimensions (30 x 30 x 90); Vertex at (140,-10,0)

Room dimensions: 200 x 100 x 100

All dimensions are in centimeters

Fig. 3. The geometry of the room.

To illustrate the efficiency of the proposed technique for building the world model, four sample problems have been considered. In the first problem the 3-D environment is completely unknown and the robot is representing the surrounding environment using the range data provided by the sensor. Figures 4-8 illustrate the plane of motion during the robot's journey from its initial position to a final position where he can directly "see" the goal. The world representation is continuously updated using the information provided by the sensor from different reading positions of the robot. It can be seen that as the robot proceeds to the goal the world representation becomes more complete.

In the second problem (Figs. 9-13), the box B1 and the prism P are provided a-priori to the robot (pre-learned knowledge). The robot is verifying the accuracy of the pre-learned information and after finding it correct, is representing the two objects using two CG primitives (Box and Prism), without using the sphere type of representation. The representation of the overall 3-D world is thus more concise than in the first problem, in which the sphere procedure was used to represent all objects.

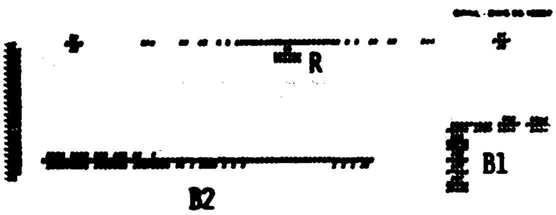


Fig. 4

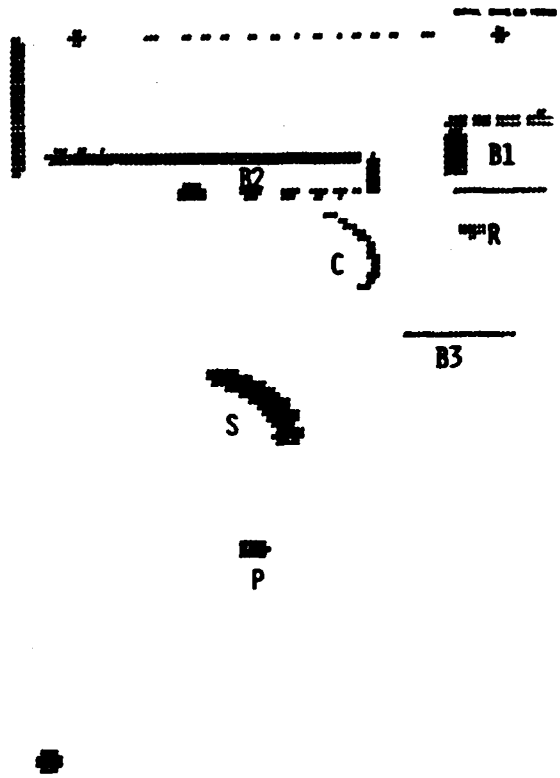


Fig. 5

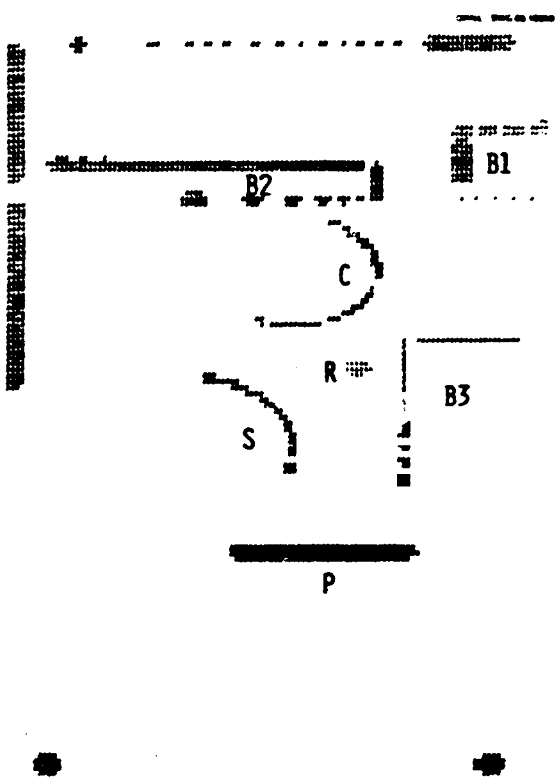


Fig. 6

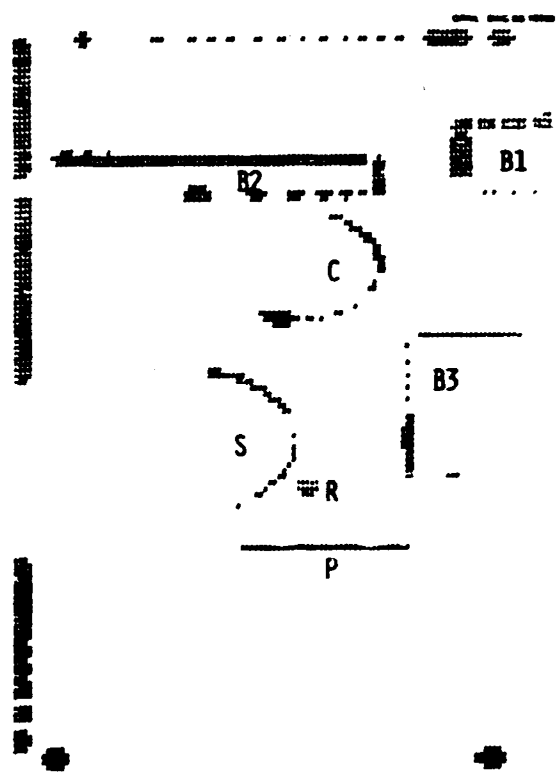


Fig. 7

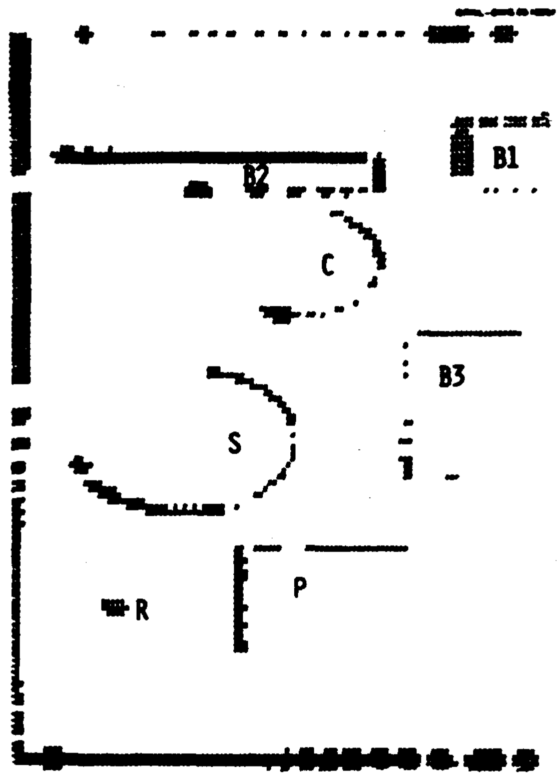


Fig. 8

The external world geometry considered in the two last problems is similar to the geometry of the first two problems, except that the boxes B1, B2 and the cylinder C are removed from the scene. In the third problem the box B3 and the prism P are defined as pre-learned information which (intentionally) was provided inaccurately to the robot. From Figs. 14-17 it can be seen that the robot is verifying the pre-learned data and finding that the box is inaccurately positioned (Fig. 14) is using the sensor (real) data only to represent it (Figs. 15 and 16). At a later stage, (Fig. 17) the robot can directly check the pre-learned information for the prism (which was previously occluded) and finding it incorrect is removing the prism from memory. The prism is then accurately represented using the data provided by the sensor.

In the last example, the box and the prism are correctly provided as pre-learned information, and the "unknown" sphere is moving forward and backward between successive positions of the robot. Figure 18 illustrates the environment with the sphere at its initial position. While the robot is moving to the second position (Fig. 19) the sphere is moving forward. The previous information about the sphere is then checked, found incorrect and removed from the robot's memory. Finally, when the robot is reaching the next (third) position, the sphere has moved back to its initial position. It can be seen (Fig. 20) that the robot is keeping the previous information about the sphere, since it is now occluded by the "real" data and therefore cannot be verified. If at a later stage the robot is again in a position to directly "see" the "old" position of the sphere, this previous information will be checked and eventually removed from the world model.

These and the following figures shown in this paper have been produced using a computer printer and a very simple plotting routine. Since the maximum resolution of the printer along the Y axis (across the page) is 130 characters, certain existing spheres having diameters smaller than the printer resolution are not printed and therefore some "holes" may artificially appear on surfaces which are in fact continuous.

6. Conclusion

The proposed approach for modeling the external world using the Combinatorial Geometry was found promising. The range data from successive locations of the robot during motion can be effectively combined and given an adequate world representation. The pre-learned knowledge and moving objects in the scene can be effectively verified and represented in the world model. The computation time per "picture" including the simulated range scan, modeling the geometry, trajectory planning and plotting the plane of motion was 30 s to 1 min CPU time of VAX-8600 Computer, depends upon the scene complexity and the number of tentative paths considered. More than 50% of the computation time is used for plotting the plane of motion and for calculating distances in a given

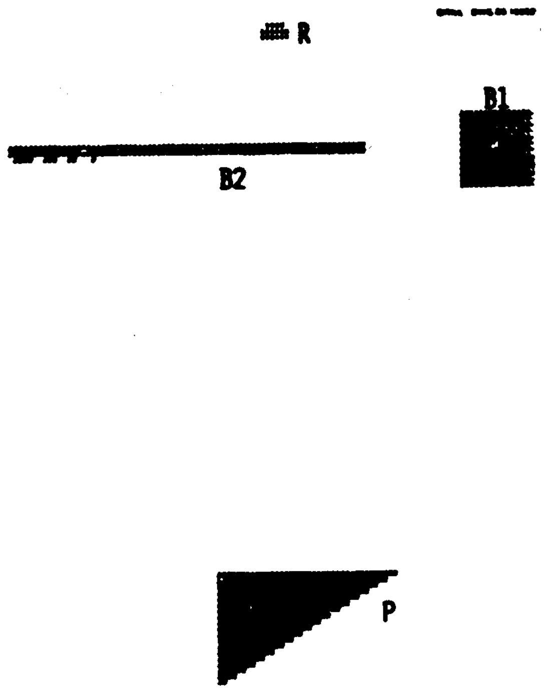


Fig. 9

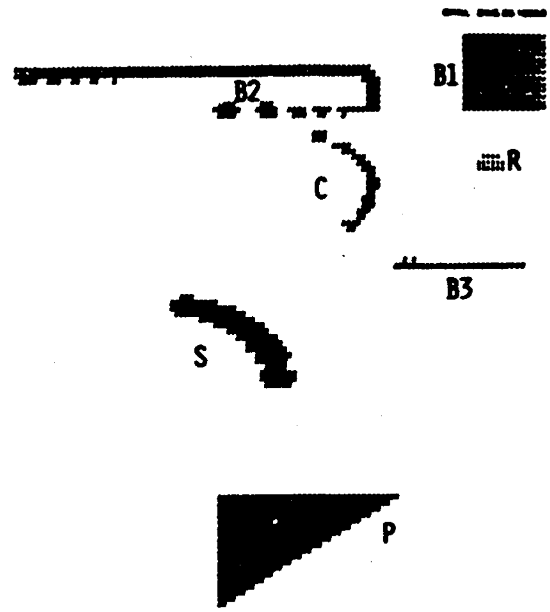


Fig. 10

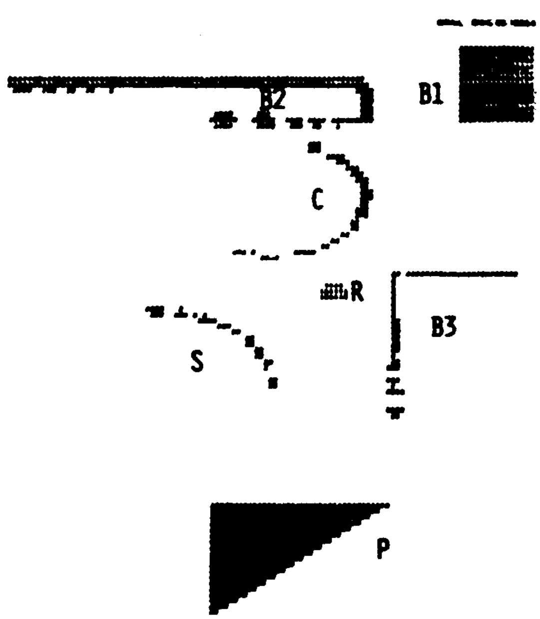


Fig. 11

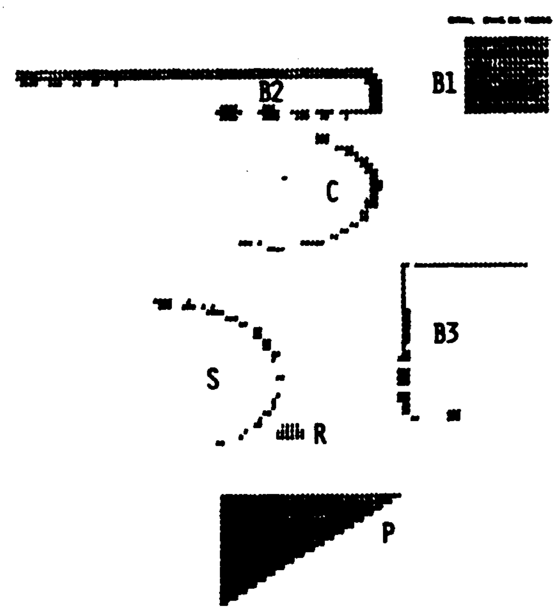


Fig. 12

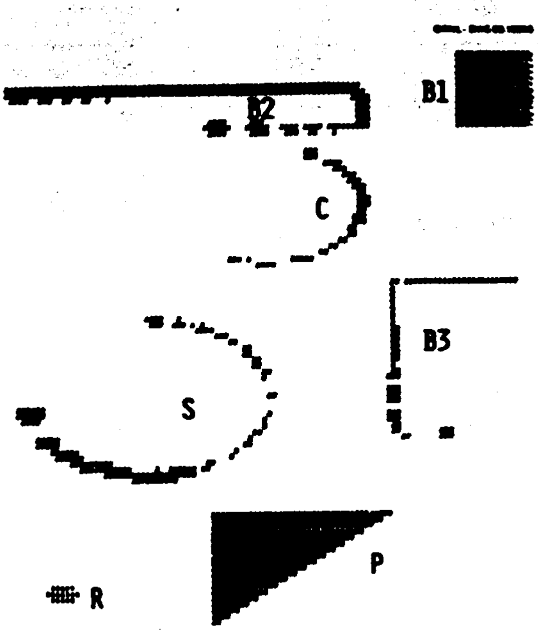


Fig. 13

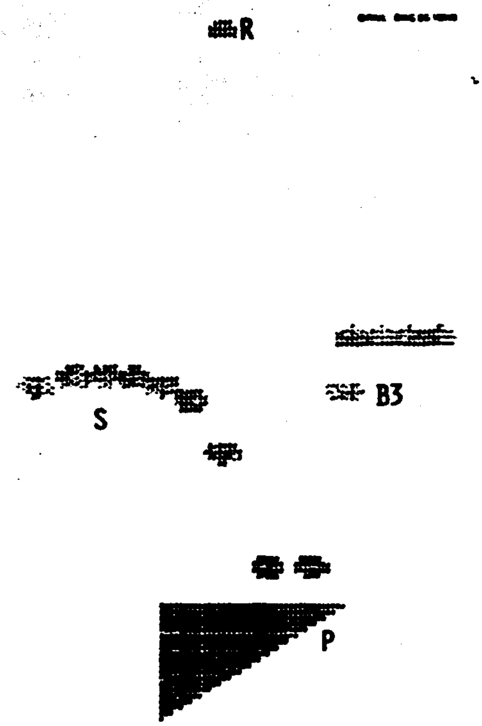


Fig. 14

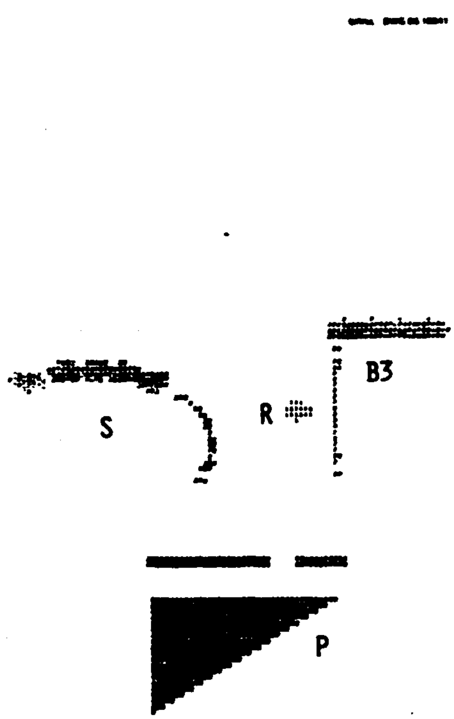


Fig. 15

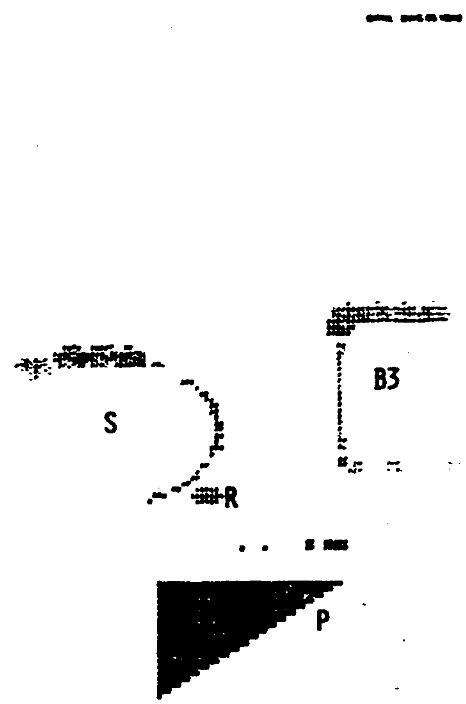


Fig. 16

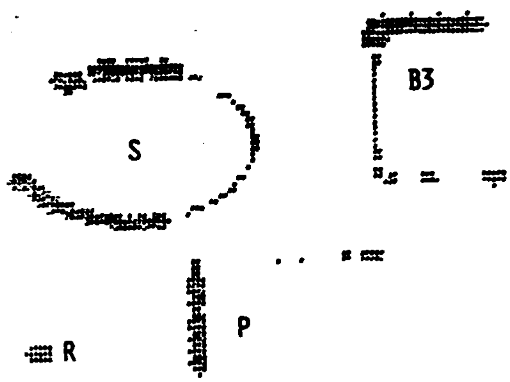


Fig. 17

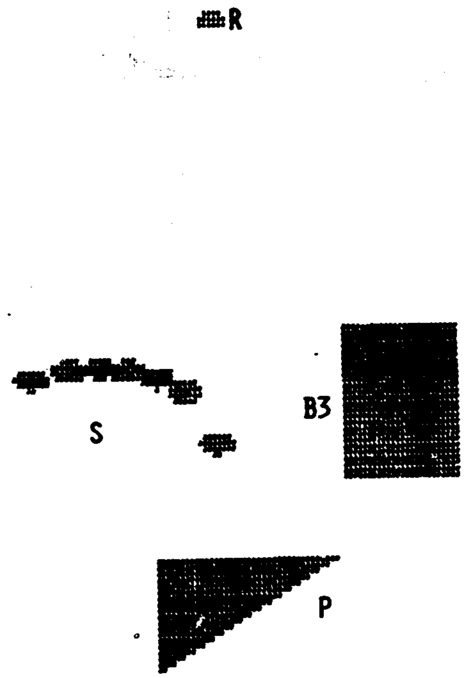


Fig. 18

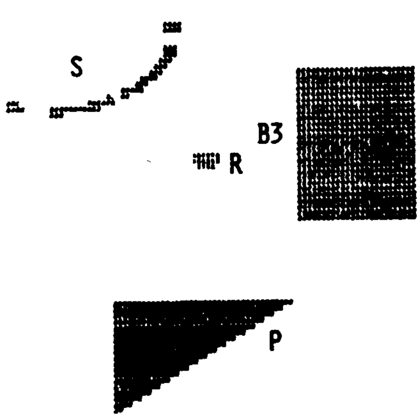


Fig. 19

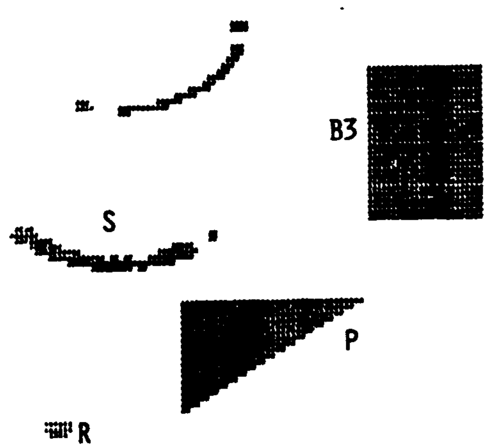


Fig. 20

direction from discrete points. These calculations can be executed independently and therefore, performing the same calculations on a parallel or concurrent computer may significantly reduce the computation time. Future work using the proposed external world modeling approach will focus on the following issues: scene segmentation into objects, feature point extraction, recognition of 3-D objects from range data, replacing the sphere representation with a more concise CG volumetric representation of the recognized objects and finally implementation of this method on the NCUBE Machine and experimental verification using the HERMIES-II robot.

7. Acknowledgements

The first author is grateful to the Nuclear Research Centre-Negev, Israel for the support and encouragement during the preliminary stage of this work. The authors gratefully acknowledge the support provided by Dr. O. Manley of the Department of Energy Office of Basic Energy Sciences, Dr. H. Alter of Office of Nuclear Energy, Dr. D. Bailey of Spent Fuel Management and Reprocessing Systems and to Karen Harber for preparing the manuscript. This research was performed under DOE contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

8. References

[1] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis," Wiley Interscience, New York (1972).
[2] S. Ullman, "The Interpretation of Visual Motion, MIT Press, Cambridge, Massachusetts (1979).
[3] J. W. Roach and J. K. Aggarwal, "Determining the Movement of Objects from a Sequence of Images," IEEE Trans. Pattern Anal. Machine Intell., PAMI-2. 554-564 (November 1980).
[4] H. H. Nagel, "Representation of Moving Rigid Objects Based on Visual Observations," Computer 14, 29-39 (August 1981).
[5] H. P. Moravec, "Rover Visual Obstacle Avoidance," in Proc. 7th Int. Conf. Artif. Intell., Vancouver, Canada, August 24-28, 1981.
[6] M. J. Magee and J. K. Aggarwal, "Determining Motion Parameters Using Intensity Guided Range Sensing," in Proc. 7th Int. Conf. Pattern Recogn., pp. 541-583, Montreal, Canada, July 30-August 2, 1985.
[7] D. Laurendeau and D. Poussart, "3-D Model Building Using a Fast Range Finder," IEEE Proc. of Comp. Vision and Pattern Recogn., pp. 424-426 (June 1986).
[8] M. Hebert, "Outdoor Scene Analysis Using Range Data," Proc. 1986 IEEE Int. Conf. on Robotics and Automation, 3, 1426-1432 (April 1986).
[9] R. T. Farouki and J. K. Hinds, "A Hierarchy of Geometric Forms," IEEE Comp. Graphics Appl. 5, 51-78 (May 1985).
[10] I. D. Faux and M. J. Pratt, "Computational Geometry for Design and Manufacture," Ellis Horwood, Chichester, U.K. (1979).
[11] W. Tiller, "Rational B-Spline for Curve and Surface Representation," IEEE Comp. Graphics Appl. 3, 61-69 (November 1983).
[12] T. W. Sedenberg and D. C. Anderson, "Steiner Surface Patches," IEEE Comput. Graphics Appl. 5, 23-36 (May 1985).
[13] P. Besl and R. Jain, "An Overview of Three-Dimensional Object Recognition," The University of Michigan, RSD-TR-19-84 (December 1984).
[14] D. J. Meager, "Efficient Synthetic Image Generation of Arbitrary 3-D Objects" IEEE Conf. Proc. Patt. Recogn. and Image Proc., pp. 473-478, 1982.
[15] J. G. Agin and T. O. Binford," Computer Description of Curved Objects," Proc. IJCAI-3, 629-640, 1973.
[16] J. J. Koenderink and A. J. Van Doorn, "Internal Representation of Solid Shape with Respect to Vision," Biological Cybernetics 32, 211-216, 1979.'
[17] G. B. Smith and T. M. Strat, "A Knowledge-Based Architecture for Organizing Sensory Data, Proc. of Int. Conf. Intelligent Autonomous Systems, pp. 106-117, Amsterdam, The Netherlands, December 1986.
[18] W. Guber, "The Combinatorial Geometry Technique for Description and Computer Processing of Complex Three-Dimensional Objects," MAGI Project 6915, MR-7004/2 (March 1970).
[19] A. P. Peutland, "Perceptual Organization and the Representation of Natural Form," AI Journal. 28(2), pp. 1-33, June 1986.
[20] B. Bhanu, "Representation and Shape Matching of 3-D Objects," IEEE Transactions on Pattern Analysis and Machine Intelligence, May 1984, PAMI-6(3), 340-351.
[21] B. Bhanu, T. C. Henderson and S. Thomas, "3-D Model Building Using CAGD Techniques," Proceedings CVPR, 1985, San Francisco, CA, pp. 234-239.
[22] M. Potmesil, "Generating Models of 3-D Objects by Matching 3-D Surface Segments," Proc. 8th IJCAI, August 1983, Karlsruhe, West Germany, pp. 1089-1093.
[23] F. P. Ferrie and M. D. Levine, "Piecing Together the 3-D Shape of Moving Objects: An Overview," Proceedings CVPR, 1985, San Francisco, CA, 574-584.
[24] B. A. Boyter and J. K. Aggarwal, "Recognition with Range and Intensity Data," Proc. of the Workshop on Computer Vision: Representation and Control, 1984, Annapolis, Maryland, pp. 112-117.
[25] M. J. Maggee, B. A. Boyter, C. H. Chien and J. K. Aggarwal, "Experiments in Intensity Guided Range Sensing Recognition of Three-Dimensional Objects," IEEE Trans. on Pattern Analysis and Machine Intelligence, November 1985, PAMI-7(6), pp. 629-637.
[26] B. A. Boyter and J. K. Aggarwal, "Recognition of Polyhedra from Range Data," IEEE Expert, Spring 1986 1(1), pp. 47-59.
[27] B. C. Vemuri and J. K. Aggarwal, "3-D Model Construction from Multiple Views Using Range and Intensity Data," Proc. CVPR, 1986, Miami Beach, Florida, pp. 435-437.
[28] M. Goldstein, F. G. Pin, G. de Saussure and C. R. Weisbin, "3-D World Modeling Based on Combinatorial Geometry for Autonomous Robot Navigation," to be published in Proc. of 1987 IEEE Int. Conf. on Robotics and Automation, March 1987.