VOLUMETRIC DESCRIPTIONS OF OBJECTS FROM MULTIPLE VIEWS

W. N. Martin[2]
J. K. Aggarwal[1,2]

TR-82-5 October 1982

[1]Department of Electrical Engineering
[2]Laboratory for Image and Signal Analysis
The University of Texas at Austin
Austin, Texas 78712

This research was supported by the Air Force Office of Scientific REsearch under grant AFOSR 82-0064.

ABSTRACT

Occluding contours from an image sequence with viewpoint specifications determine a bounding volume approximating the object generating the contours. The initial creation and continual refinement of the approximation requires a volumetric representation that facilitates modification yet is descriptive of surface detail. The "volume segment" representation presented in this paper is one such representation.

1. Introduction

Volumetric models have been the basis of numerous three-dimensional object modelling systems. In order to represent the desired volumes, various types of primitives have been specified. Initially, polyhedra were used as combinations of a small set of simple volumes [1] and later as general volumes [2]. Each face of a polyhedron may be considered to be part of the bounding plane of a half-space. The interior of the polyhedron can then be described procedurally by the "addition" and "subtraction" of these half-spaces. Requicha [3] defines "general regularized set operators" and a syntax for specifying the procedural description in a constructive solid geometry scheme. The operators will always yield "valid" representations if the primitives have finite volumes. Through additional operators general sweep representations can also be specified. Generalized cylinders are a class of sweep representations that have been studied extensively [4-8]. Elsewhere spheres [9] have been used in structures approximating symmetric surfaces, i.e., three-dimensional generalizations of symmetric axes [10].

Most of the systems referenced to this point are geometric modelling systems [11] primarily developed for computer aided design applications, such as BUILD [2] and PADL [12], or for image interpretation, such as MSYS [13], VISIONS [14] and ACRONYM [15]. These systems usually have interactive procedures for deriving the three-dimensional models and often have a major interest in the suitability of the model for graphical display [16-18]. The primary concern of this paper is the development of volumetric descriptions suitable for deriving three-dimensional object representations from two-dimensional images. Determining such representations has been the goal of many computer vision systems since Roberts' original paper [1]. Roberts demonstrated that three-dimensional information about the actual objects in a single image can be derived under two classes of constraints. The first class included overall scene domain constraints, e.g., the objects were planar faced, while the second class involved specific object constraints, e.g., the objects were combinations of a limited number of polyhedra. These polyhedra were the primitives instanced in various sizes and arrangements to form the actual objects in the scene.

Roberts' paper is extremely important for having established a paradigm that many researchers in scene analysis have followed. Attempts have been made to lessen the restrictions imposed by either or both classes of constraints. For example, the specific object constraints were replaced by more extensive domain constraints in the thorough work on line drawings of polyhedral scenes, see [19-22]. Different sorts of scene constraints have also been applied. For instance, known properties of special illumination conditions can indicate surface orientations, see [23-27].

The constraints can be reduced further by using multiple views of the scene, often with a time ordering resulting in an image sequence [28,29]. For each view in the sequence feature points can be detected and a correspondence [30] formed between the features in successive views. The image positions of the corresponding feature points provide additional constraints that can be written in the form of a set of non-linear equations [31-33]. Solving the set of equations yields a wire-frame model of the three-dimensional positions of the scene components underlying the image features. The multiple views presented in image sequences can result from either camera or object movement (or both) and provide an additional source of constraints through the control [34, 35] or restriction [36] of the exhibited motions. In the system we have developed the motion creating the multiple views will not be controlled but will be precisely known through viewpoint specifications.

Our work has been in the pursuit of two major goals. The first goal is the development of a system that is capable of deriving three-dimensional object description images, yet does not depend completely on feature point measurements. The second goal is the development of a scheme for representing three-dimensional objects that is descriptive of surface detail, while remaining functional in the context of a structure from multiple view system. The result of this work is a system which uses the occluding contours from multiple images with viewpoint specifications to construct a bounding volume approximating the actual three-dimensional structure of the rigid object generating the contours. The representation constructed to facilitate the creation and continual refinement of the bounding volume is referred to as a "volume segment" representation and will be discussed in Section 3.

2. Volumes from Contours

By an "occluding contour" we mean the boundary in the image plane of the silhouette of an object generated by an orthogonal projection. In an intensity image the silhouette can most often be formed by a simple thresholding of the intensity values. A connected component analysis [37] of the resulting binary valued image yields the boundary of the object silhouette.

Throughout this paper, the orientation of a view will be specified by a point and three mutually perpendicular directions that form a right-handed, three-axis coordinate system with the given point as origin. The y-axis of each coordinate system is considered to be the direction of the line of sight with the (x,z)-plane being the image plane, i.e., the silhouette is projected onto the image plane by lines parallel to the line of sight. Thus, an occluding contour is generated by the lines that are parallel to the y-axis and that intersect the object, but do so only at points which are on the object surface. For objects with smooth surfaces this means that the lines are in the tangent planes of their intersection points.

Clearly, then, the object surface must touch each contour generating line. The problem is to determine which points on the line are also object surface points. This problem is solved by using the constraints imposed by the contour generating lines from a second (and subsequent) view. The situation of two views is shown in Figure 1. The contours are displayed in their respective image planes and some representative contour generating lines are indicated by equal length sections of lines that

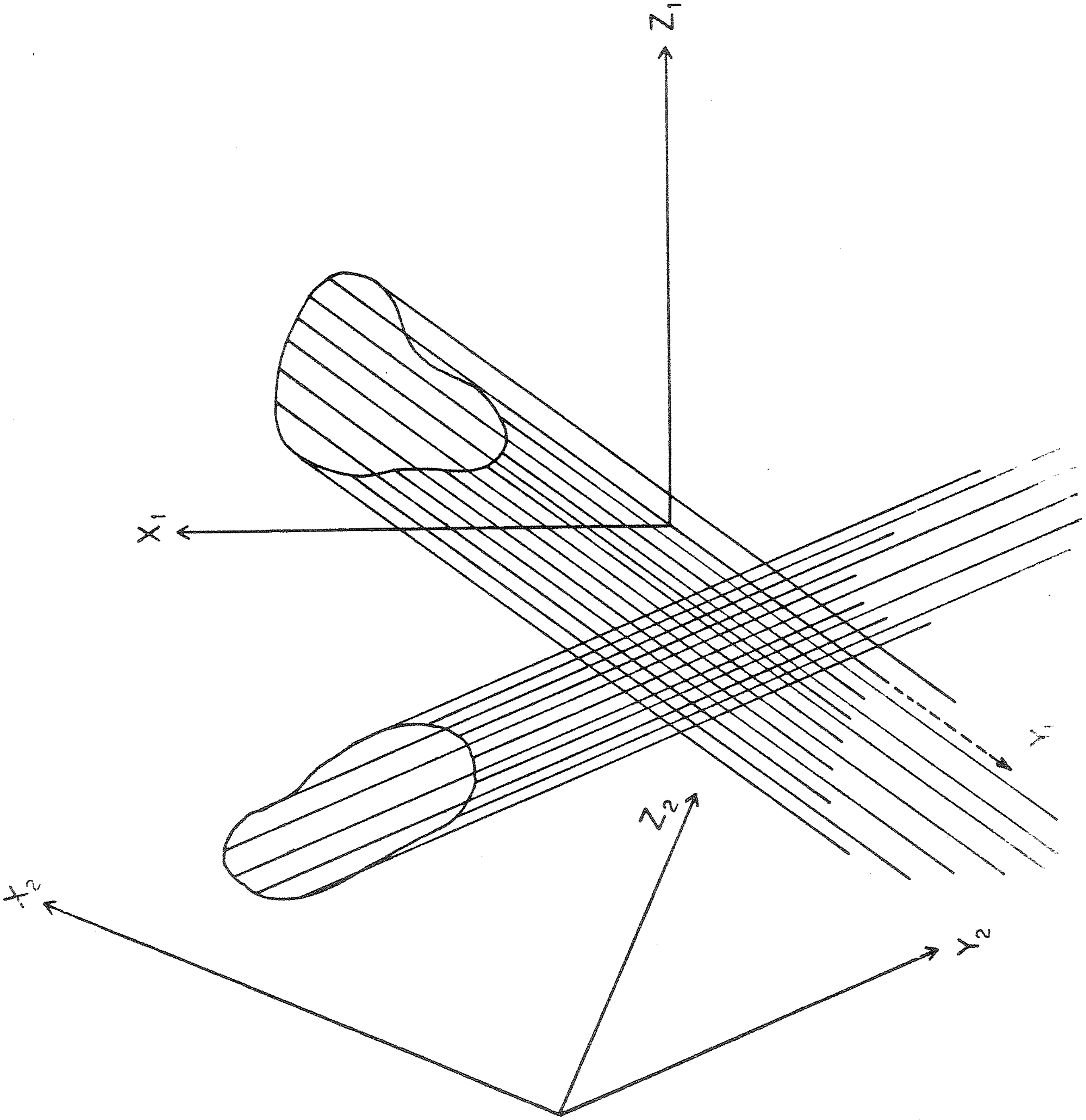

Figure 1. Silhouettes and contour generating lines for two views.

are parallel to the appropriate y-axis. The set of contour generating lines for a given view defines a volume which bounds the actual object, however, this volume is by itself infinite. A second view will also define an infinite volume that contains the object, and if the second view is distinct, i.e., the two lines of sight are not parallel, then the intersection of the two volumes will still encompass the object and will be of finite extent. Intersecting the volumes to define the approximation is in the spirit of the geometric modelling systems [2,3,12], yet is a process that can be applied to image data.

It is clear from Figure 1 that each contour generating line for one of the views serves to constrain the extent of the possible object surface points on some of the contour generating lines for the other view. In order to establish the initial estimate of the object, the system must determine how each contour generating line constrains the volume from the other view and create a structure that satisfies the various constraints simultaneously.

To see how these constraints are specified consider Figure 2. Again, two contours are shown in their respective image planes with the lines of sight corresponding to the y-axes. Since these directions are distinct, the image planes are not parallel and thereby must share a common line. The common line for the two image planes is shown on the right hand side of Figure 2. This line is in the direction of the cross-product of the two y-axes, and so is perpendicular to both axes, as indicated in the figure. Now, the upper right contour is in the (x1,z1)-plane

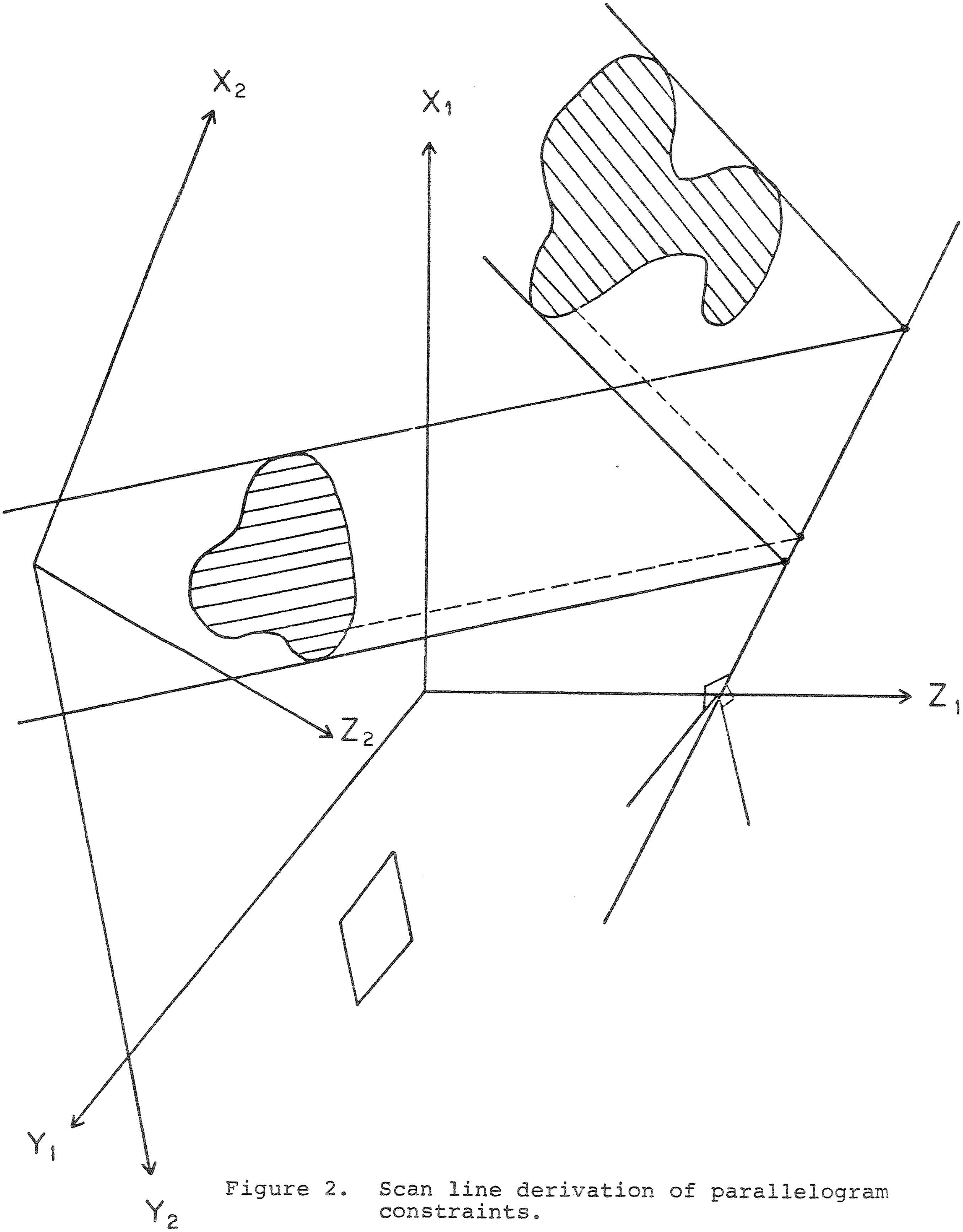

Figure 2. Scan line derivation of parallelogram constraints.

parallelogram in a given plane, but that is indicative of the cross section being disconnected with each connected subset bounded by a separate parallelogram. The locus of all the parallelograms in the planes of the various scan lines is the bounding volume which establishes the initial estimate of the object structure.

With the object structure described in this manner, the system is ready to form the volume segment representation which will be used in the subsequent refining process. The next section will define the volume segment representation and then discuss how it is derived from this initial parallelogram structure.

The process described in this section should be contrasted with computer aided tomography [40] and other cross-sectional methods [9,41,42] on two main points. First, the source of data is a sequence of simple binary images possibly from a television camera as opposed to the elaborate data acquisition mechanisms required for tomography and range measurement [43]. Second, neither the initial volume specification described above nor the refinement process explained in Section 4 depends on predefined cross-sectional decomposition of the data.

3. Volume Segment Representation

The three-dimensional structure to be derived from the sequence of occluding contours is a bounding volume approximating the actual object. For this reason the representation incorporated in this system is based on volume specification through a "volume segment" data structure. The volume segment representation is a generalization to three-dimensions of the rasterized area description. For the rasterized area, the segments each denote a rectangular area. The generalization to three dimensions is to have each segment represent a volume, i.e., a rectilinear parallelipiped with edges parallel to the coordinate axes. In addition to grouping colinear segments into lists, the set of segment lists is partitioned so that the subsets contain lists having coplanar segments. The primary dimension of the rectilinear parallelipiped specified by a segment is the length of the segment. The second dimension is given by the inter-line spacing within the plane of the segment, while the third dimension is the inter-plane distance. The latter two dimensions are specified to be uniform throughout the volume segment representation.

The structure then maintains an ordered pair of values for each volume segment: the values are the y-coordinates of the segment endpoints and are ordered into lists of segments having the same x-coordinate, i.e, colinear. The x-level lists are then coalesced into z-level ordered lists by common z-coordinate, i.e., coplanar. From the top down this structure is a set of "planes" parallel to the z=0 plane, that are ordered by z-value. Each "plane" contains a set of "lines" parallel to the y-axis, that are ordered by x-value. Each "line" comprises a set of disjoint segments that are ordered by endpoint y-value. Figure 3 shows a schematic of a volume segment structure using linked lists to order the various components.

In a general situation the primary advantage of this

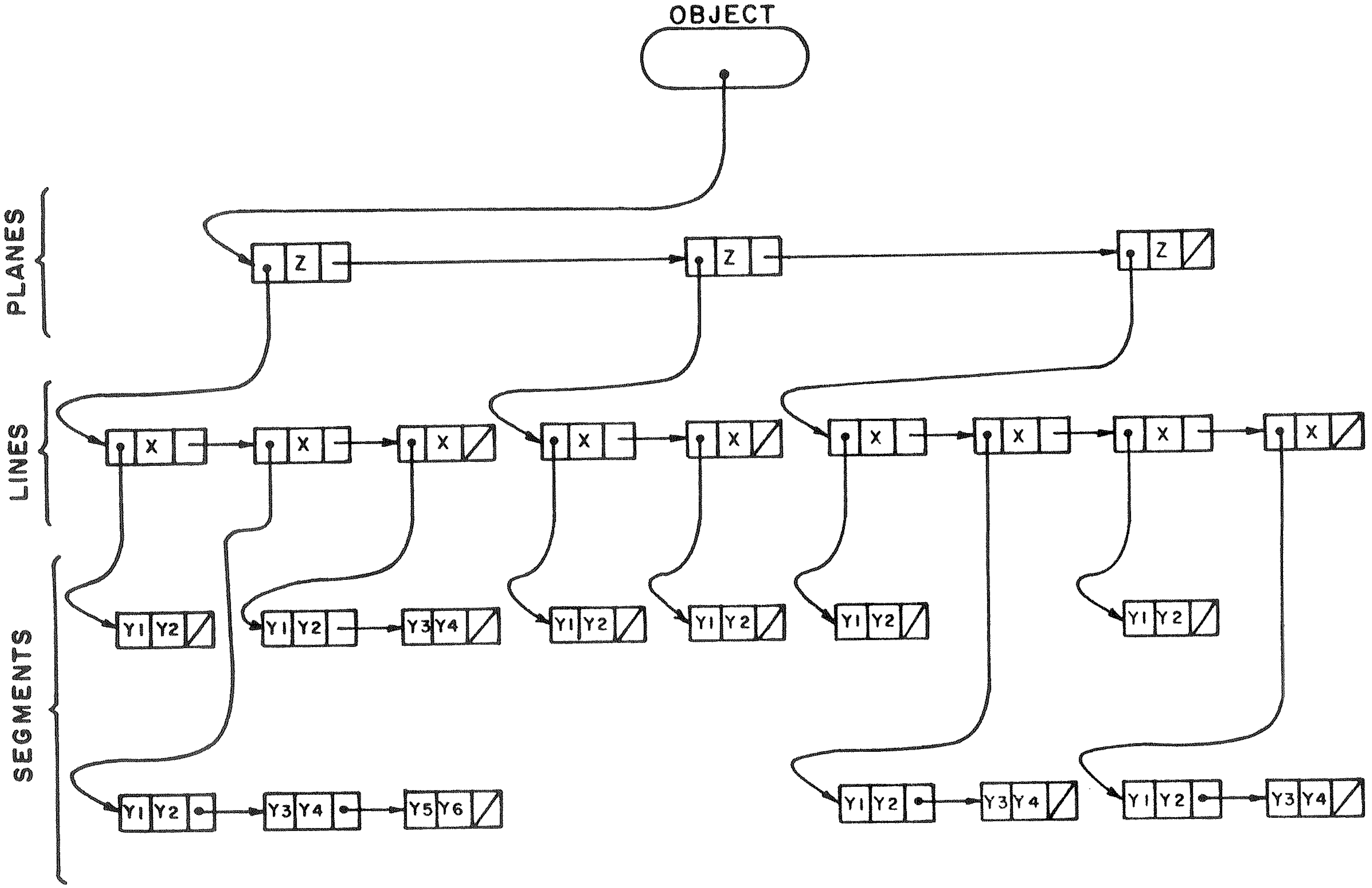

Figure 3. Data structure schema for the volume segment representation.

structure is that the process of determining whether an arbitrary point is within the surface boundary consists of a simple search of three ordered lists: select a "plane" by z-coordinate; select a "line" by x-coordinate; and finally, check for inclusion of the y-coordinate in a segment. The simplicity of this process is in contrast to that required by a representation scheme such as constructive solid geometry [3] wherein the object is represented using regularized set operations on primitive solids. For a particular object represented by given solids, point inclusion is specified by determining if the point is in each primitive solid (possibly a difficult problem itself) and then performing the logical operations corresponding to the set operations from which the object representation is constructed.

The volume segment structure can also provide a fairly succinct representation, particularly for objects that are elongated in the direction of the y-axis. It becomes more verbose as the elongation extends in the z-axis direction, however, it always remains more compact than surface-ordered enumerative representations, e.g., connected sets of "voxel" faces [44]. In comparing a volume segment structure to a connected voxel face structure one should observe that each leaf in the tree of the volume segment structure represents two voxel faces and remains only two levels from the root of the structure. In addition each voxel face must maintain its three-dimensional position and connections to four edge-adjacent faces. In balance, surface connectivity is not specified directly in the volume segment structure and may require a small amount of searching to compute. As will be shown by the example in Section 5, forming a surface description from the volume segments is possible and is computed as a matter of course by this system.

Before proceeding with the details of that system let us indicate a comparison with another volume oriented representation, that of Oct-trees [45]. Consider a simple cube measuring 2^n units on each edge that is embedded with standard orientation in a single octant of a space extending 2^{n+1} units in each direction. For the volume segment structure to represent the cube requires 2^n z-planes each having 2^n x-lines with 2 y-endpoints per line. This yields 2^{2n} volume segments represented by a tree having a total of $(1+2^n + 2^{2n}+1)$ nodes. Note the connected voxel face structure would require 2^{2n} voxel faces for each side of the cube, yielding $6(2^{2n})$ faces.

In contrast, the cube can be represented by an oct-tree of just 9 nodes. Of course, this example is a best case for oct-trees. Now, translate the cube one unit in the x direction so that the cube intersects two octants of the embedding space: the volume segment structure and connected voxel face structure do not change. However, the oct-tree representation expands to require $1+6+((4^{n+2}-10)/3)$ nodes. Unit translations of the cube in the remaining two principle directions will yield further expansion of the oct-tree. Rotation of the cube can cause expansion in both the volume segment and connected voxel face structures, but these expansions would not be of the magnitude described for the oct-tree structure.

4. Construction and Refinement for the Volume Representation

This section describes both the process by which the volume segment representation is initially constructed from the parallelogram structure detailed in Section 2 and the process that refines the representation using subsequent views. The first step in the initial construction process is to define a new coordinate system, relative to which the volume segment representation will be specified. Consider, now, Figure 4, in which the contours from two views are displayed in their image planes. Also shown are the lines common to both image planes and a few appropriate scan line segments for the contours. For the bottom scan line of each contour the limiting contour generating lines, i.e., the dotted lines parallel to the respective y-axes, are shown forming the parallelogram for that plane. Also shown are the parallelograms for the planes specified by the remaining scan line segments. Note that the top scan line of the triangular contour is a single point, resulting in a degenerate parallelogram. Again, each parallelogram is in a plane that is perpendicular to the common line. This observation suggests that the common line could be used as the z-axis of the volume segment representation's coordinate system. Making that choice, the y-axis of the new system is defined to be in the direction of the line of sight for the first view, while the x-axis is selected to complete the right-handed coordinate system, e.g., the (x',y',z')-axes in Figure 4. Of course, either line of sight could be used for the y-axis as it is known that both are perpendicular to the common line. The choice of one of these two

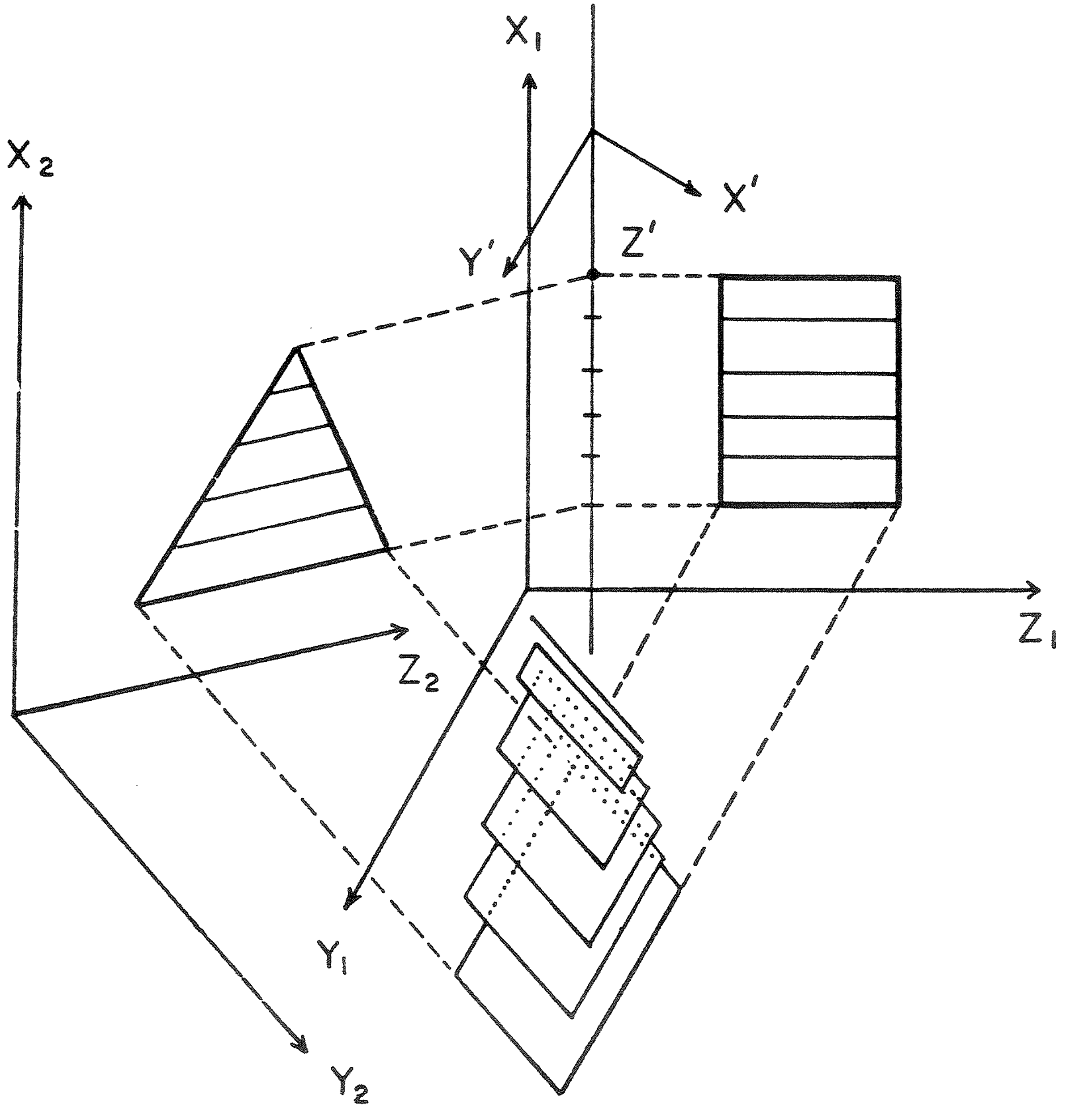

Figure 4. Parallelogram structure yielding the initial volume segment representation.

directions is made to simplify the next step: parallelogram rasterization.

Each parallelogram is taken as an area in the appropriate (x',y')-plane and rasterized along scan lines parallel to the y'-axis. For the plane of a parallelogram, the rasterization results in a set of line segments parallel to the y'-axis for each x'-value. Thus there is a set of planes each having various lines that are broken into segments. With this information all that is needed is to add the linking structure and the volume segment representation will be complete. Figure 5 displays the representation formed for the parallelogram structure of Figure 4. In Figure 5 the segments are drawn relative to the representation's coordinate axis system. The dotted line from the z-axis indicates the plane containing the segments derived from what was the bottom parallelogram in Figure 4. The other segments are in planes nearer the origin. Figure 6 displays a summary of the algorithm to create the initial volume segment representation that was just explained, while Figure 7 exhibits a summary of the volume segment representation refinement process that is described in the remainder of this section.

The refinement process is applied as each new frame (after the first two) of the dynamic image is obtained. As was stated earlier, this process makes extensive use of the clipping procedure that is based on rasterized area descriptions. To see how this is done consider Figure 8. The volume segment representation of Figure 5 is again shown, however, the contour, image plane and line of sight for a new frame are included in the

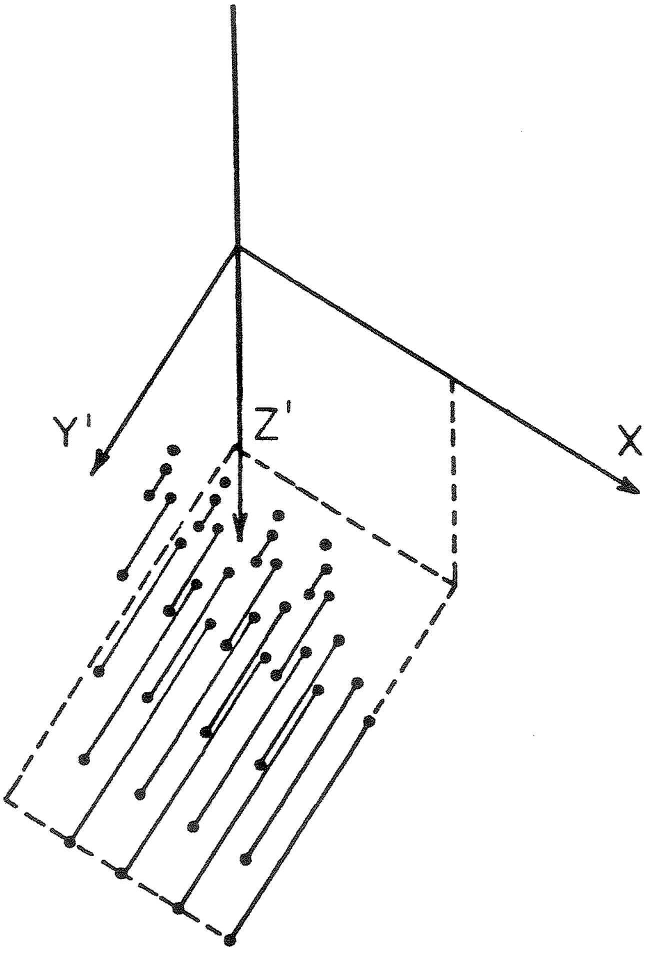

Figure 5. Volume segment representation in its own coordinate axis system.

Algorithm summary:

1. obtain image and viewpoint specification for first frame.
2. form silhouette and extract boundary coordinate list for first frame.
3. obtain image and viewpoint specification for second frame.
4. form silhouette and extract boundary coordinate list for second frame.
5. determine line common to the image planes from the first and second frames.
6. rasterize the boundaries along scan line direction perpendicular to the common line.
7. merge mutual constraints from corresponding scan line segments to form the parallelogram description of the object.
8. define the coordinate axis system for the volume segment representation having the z-axis in direction of the common line and the y-axis in the direction of the line of sight for the first frame.
9. rasterize the parallelograms in each z-plane of the new axis system to define the segments of the volume representation.
10. add the linkage structure to complete the volume segment representation.

Figure 6. Initial representation creation.

Algorithm summary:

1. obtain image and viewpoint specification for new frame.
2. form silhouette and extract boundary coordinate list for new frame.
3. project the y-axis of the volume segment representation's coordinate system along the line of sight and onto the image plane of the new frame.
4. rasterize the new boundary along scan lines that are parallel to the projection of the y-axis in the image plane.
5. for each segment in the volume segment representation do the following:
    - 5a. project the segment along the line of sight and onto the image plane of the new frame.
    - 5b. clip the projected segment to the rasterized boundary.
    - 5c. update the actual segment with respect to the clipped projection.
6. create surface description if desired.
7. continue at step 1 if there are more frames, else stop.

Figure 7. Continuing representation refinement.

figure. The overall process is to clip the volume segments by the new contour. For this to be done, the contour must be rasterized properly. The required direction for the scan lines is the direction in the image plane of the projection, along the new line of sight, of the y-axis for the volume segment representation. The dotted lines in Figure 8 from the y-axis into the (x',z')-plane indicate this projection. Note that the resulting direction will not normally be parallel to the original y-axis. The contour in Figure 8 is shown to be rasterized along scan lines parallel to this projected direction.

Given the properly rasterized area description of the new contour the refinement procedure is as follows. Each segment of the volume representation is projected onto the new image plane, again according to the direction of the new line of sight. The projected segment, then, is in the image plane and can be clipped as described previously for the two-dimensional case. The original volume segment can then be updated by modifying its length in a proportion equivalent to that by which the projected segment was clipped. In geometric terms, the clipped segment could be inversely projected onto the line of the original volume segment, with the resulting segment replacing the original segment. A special case occurs when the line of sight happens to be parallel to the y-axis of the coordinate system of the volume segment representation. However, in this case each volume segment projects onto a single point in the image plane and a simple point inclusion test on the area description (rasterized to an arbitrary direction) determines whether the entire volume segment is to be retained in or deleted from the representation.

5. An Example Dynamic Scene

To illustrate the volume segment representation and the process which constructs it, an example is presented in this section. The various stages of the example are shown in Figures 9 through 17. The four frames, with raster lines inserted, of the input dynamic scene are presented in Figures 9 through 12. In these frames the object rotates about a vertical axis so that it traverses 90 degrees between the first and last views. The combination of the first two views, i.e., Figures 9 and 10 results in the surface shown in Figure 13. Each pair of "raster" lines (one from each view) generates a parallelogram which lies in a plane parallel to the global (x,y)-plane. The parallelograms have been connected along corresponding corners with the top and bottom faces marked by crossing lines. The volume segment representation resulting from this surface is exhibited in Figure 14. As stated earlier, each segment is parallel to the global y-axis. In Figure 14 perspective cues have been added to provide the proper sense of depth. The third and fourth frames are then processed to constrain the volume segment representation to be as it is displayed in Figures 15 and 16, respectively. Finally, the volume segment representation of Figure 16 is transformed into a surface description, as illustrated in Figure 17. It should be noted that the surface representation of Figure 17 is derived directly from the volume segment representation and is much more general than the surface description shown in Figure 13.

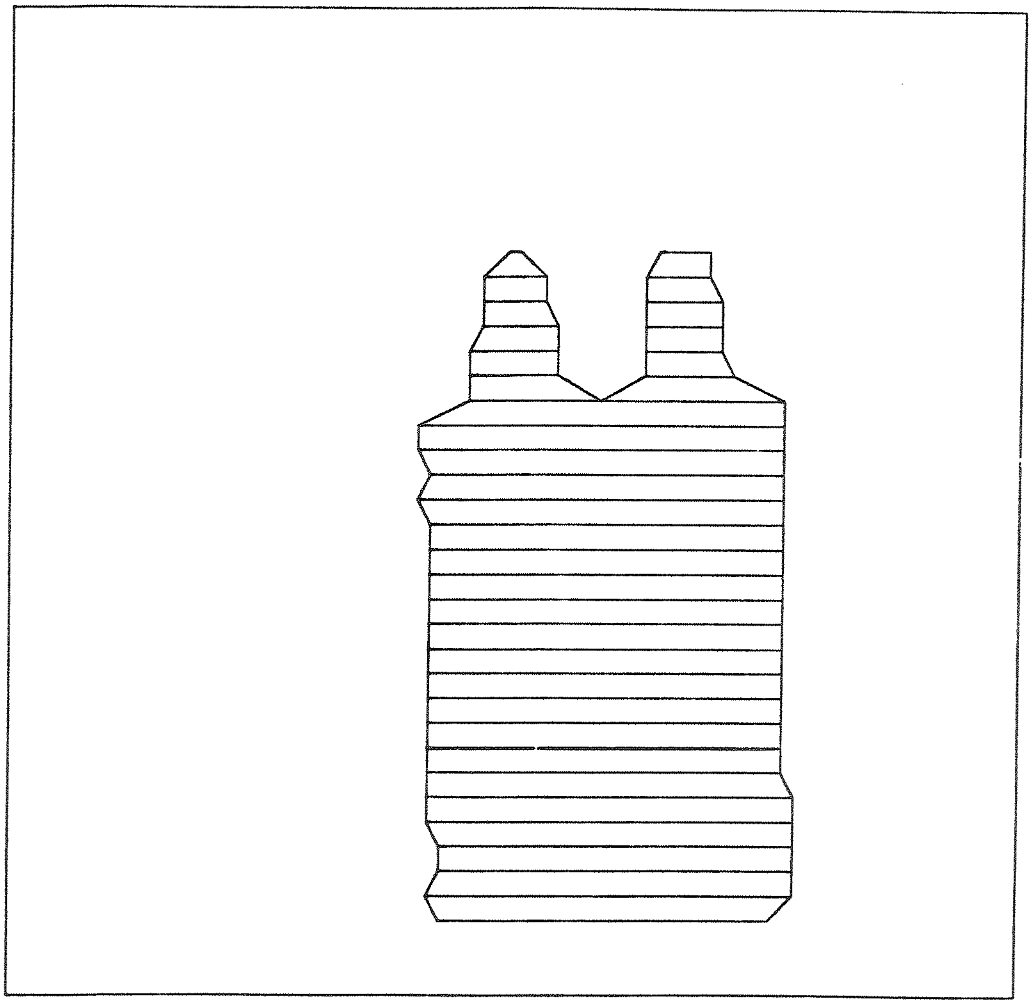

Figure 9: Contour from first view with "raster" lines inserted.

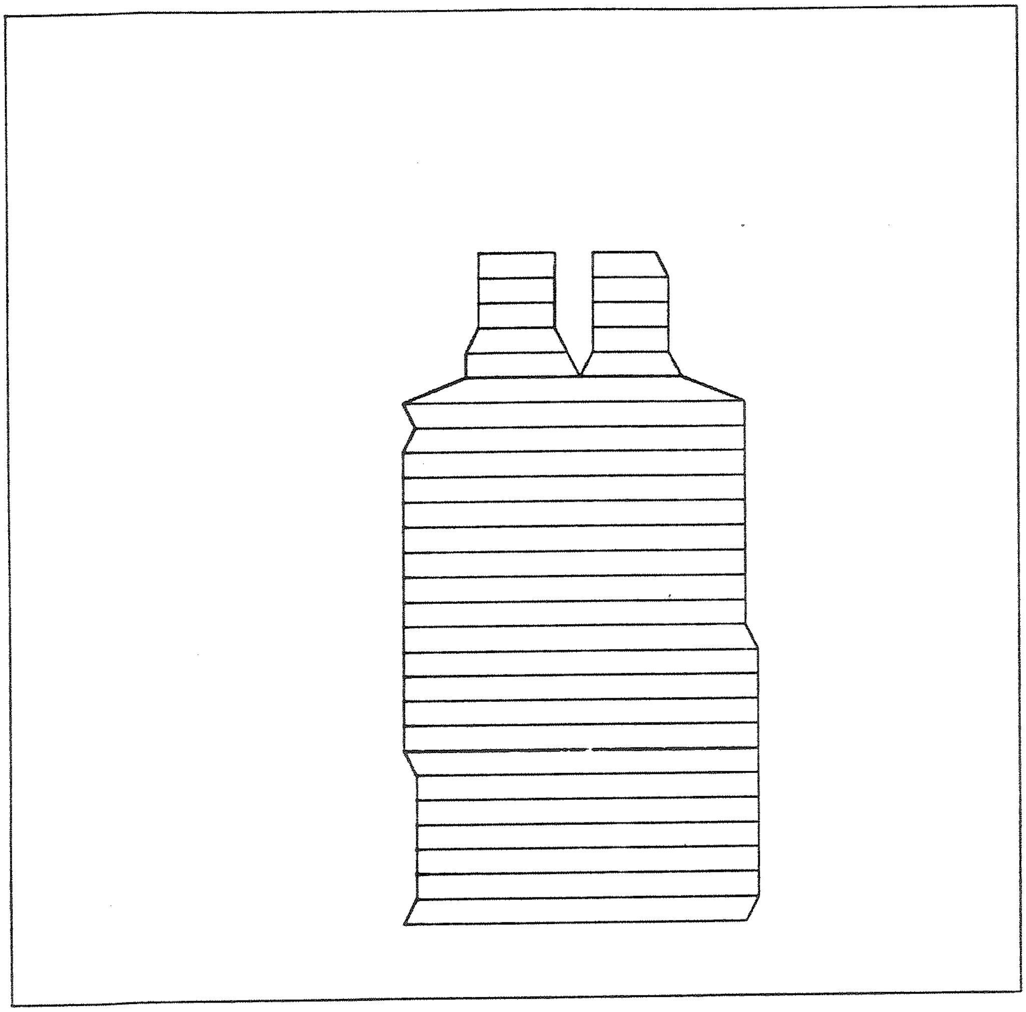

Figure 10. Contour from second view.

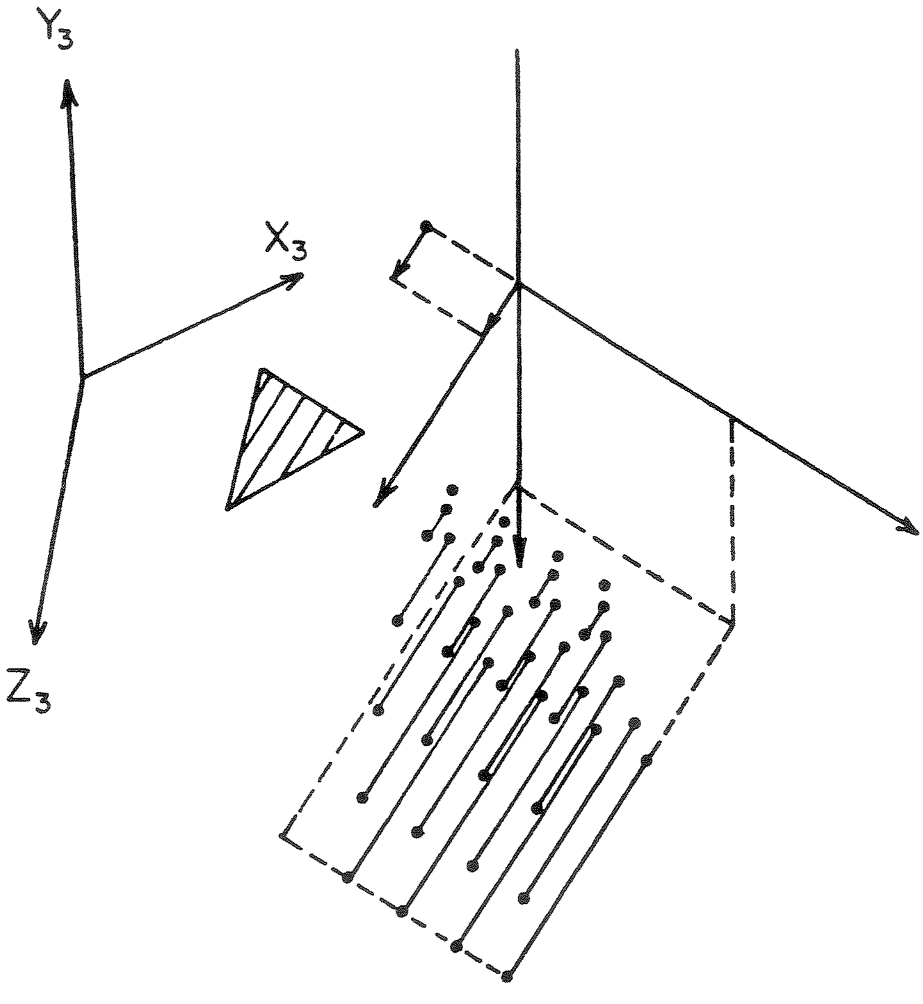

Figure 8. Refining the volume segment representation by subsequent views.

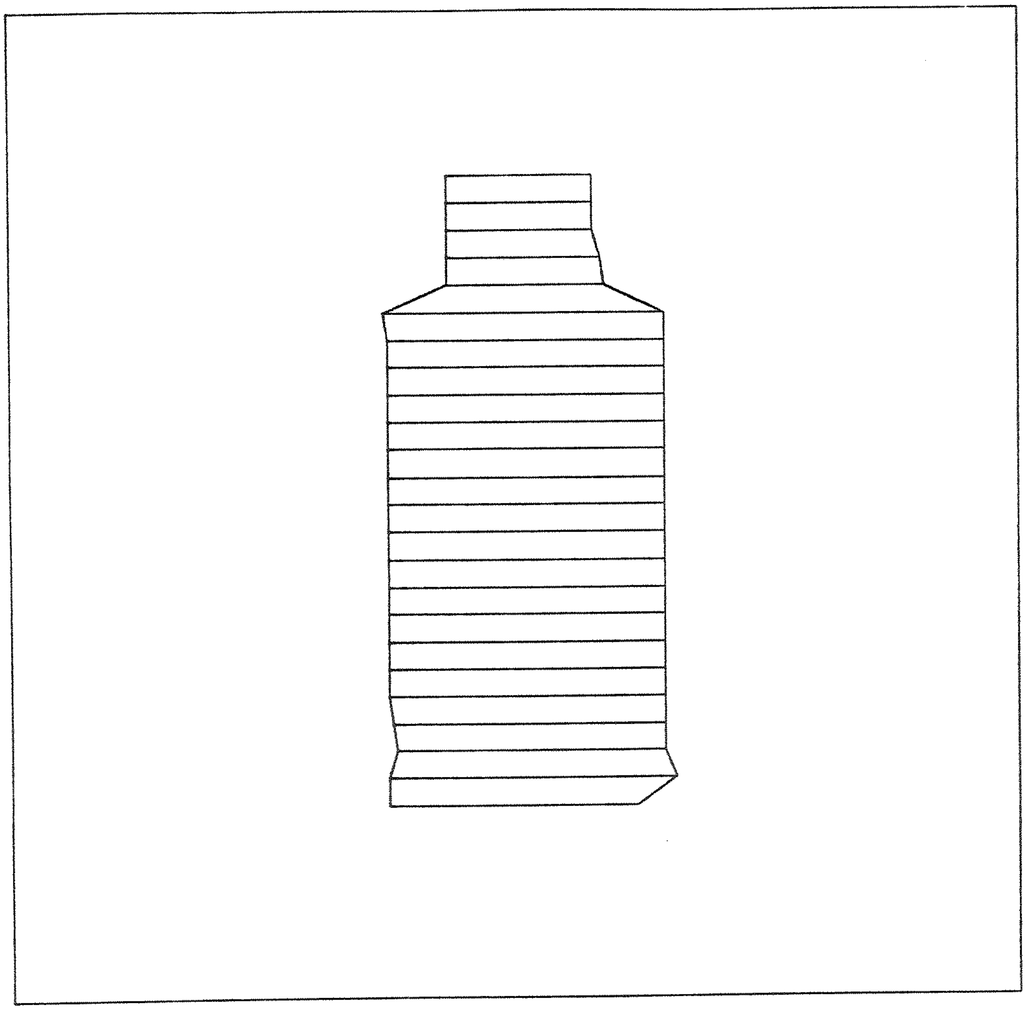

Figure 11. Contour from third view.

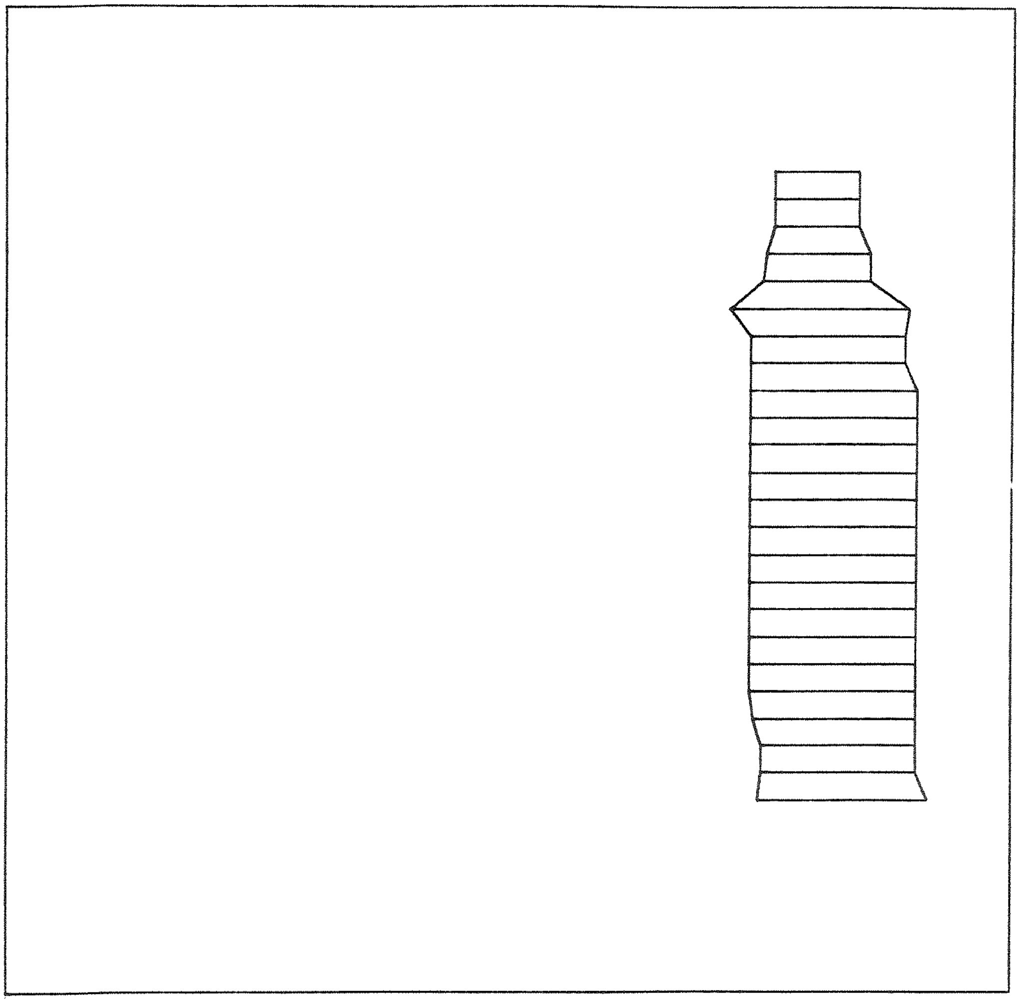

Figure 12. Contour from the fourth and last view.

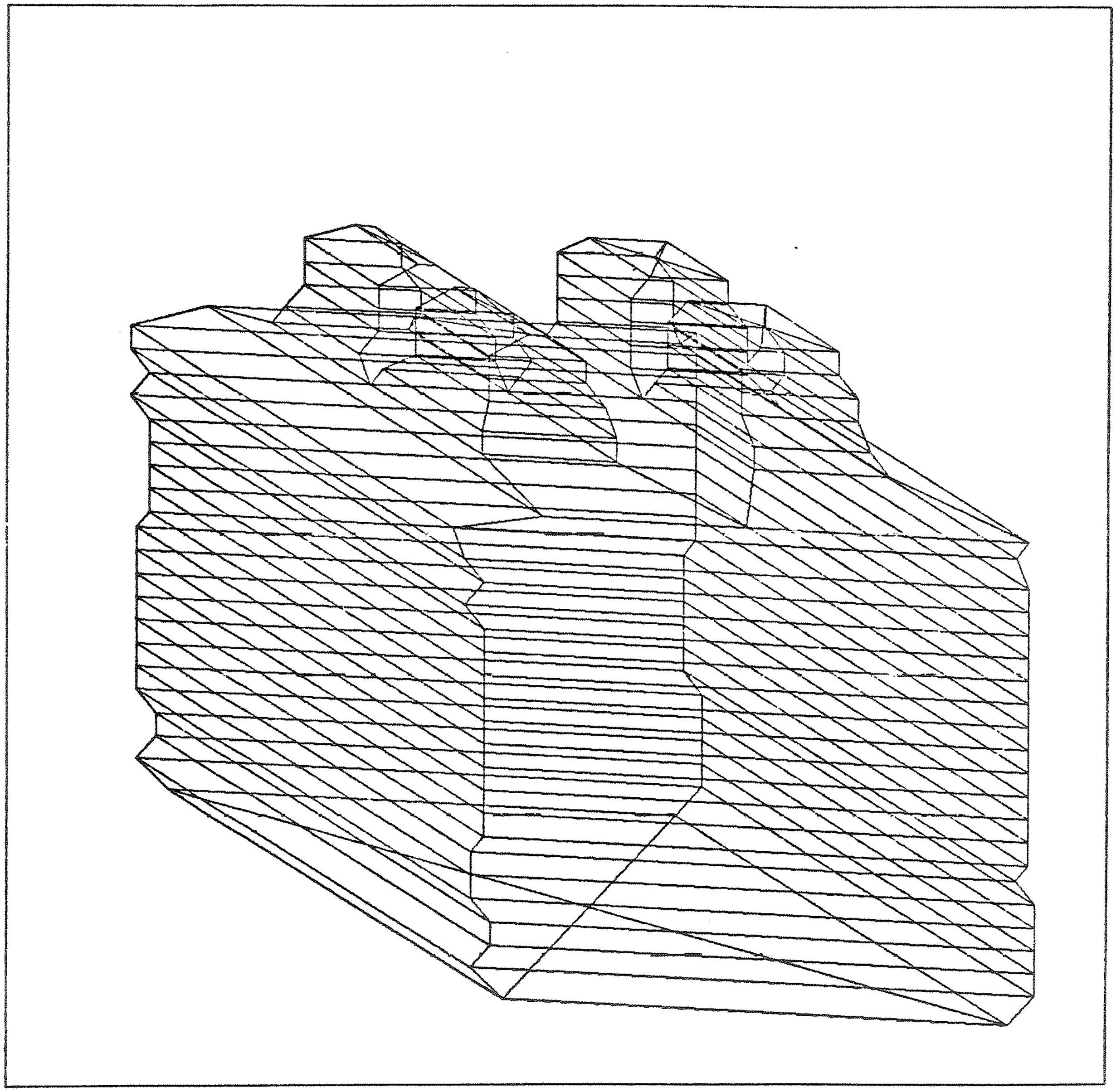

Figure 13. "Parallelogram cross-section" surface description derived from the views displayed in Figures 9 and 10.

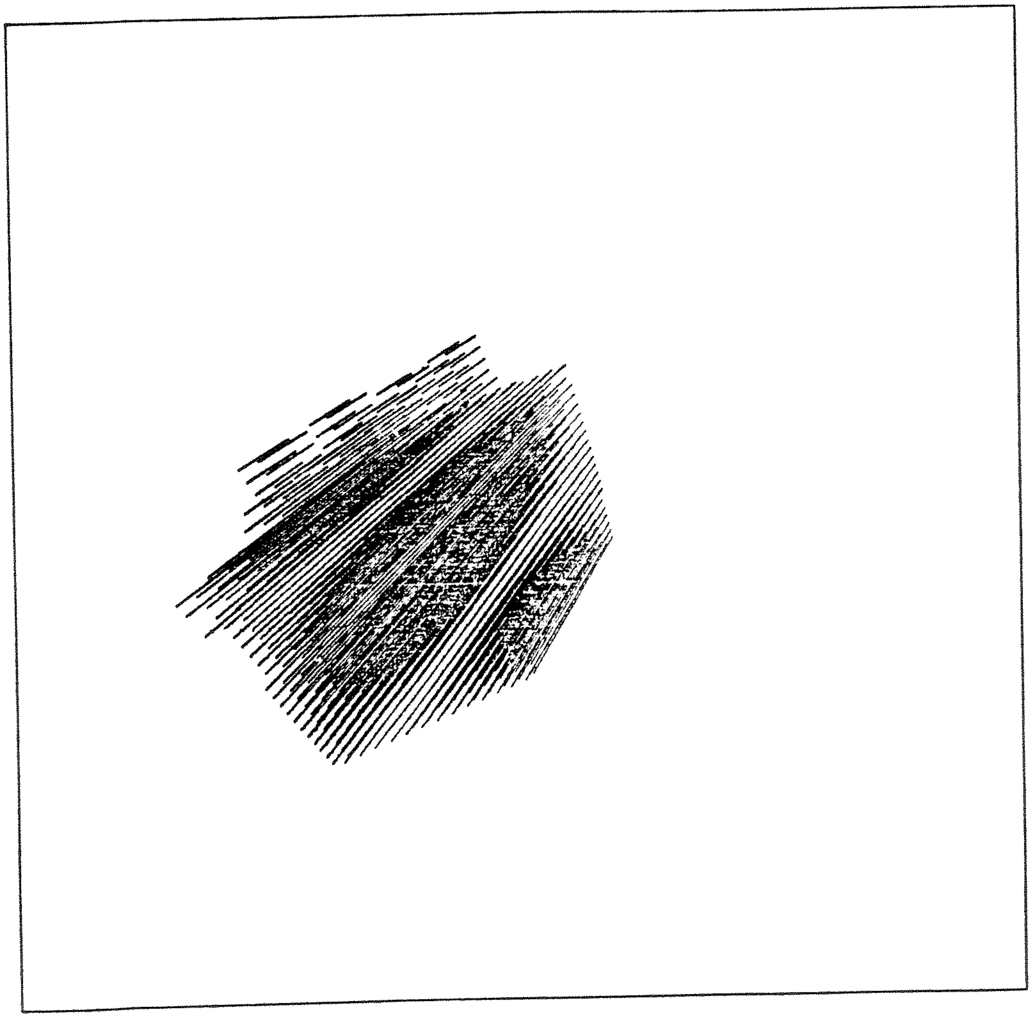

Figure 14. The volume segment representation of the surface in Figure 13.

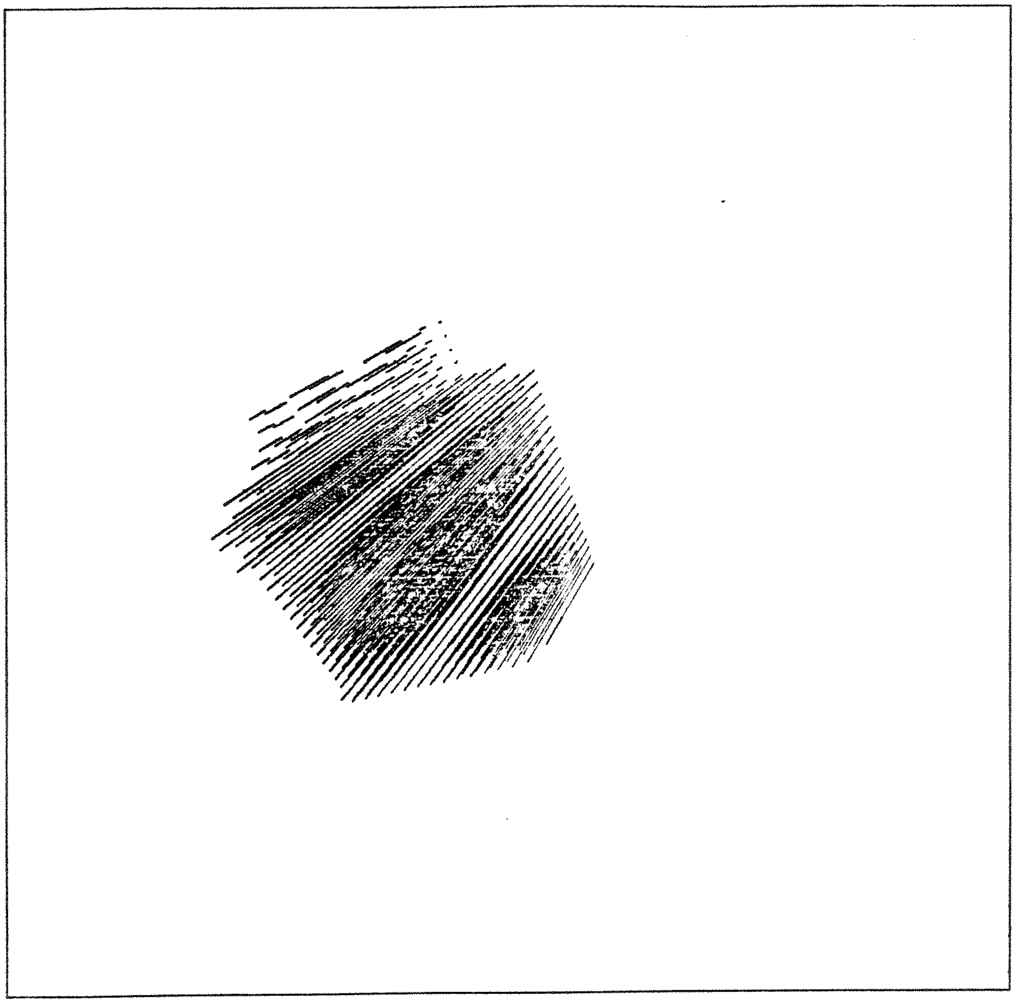

Figure 15. The volume segment representation derived by constraining the representation shown in Figure 14 by the contour displayed in Figure 11.

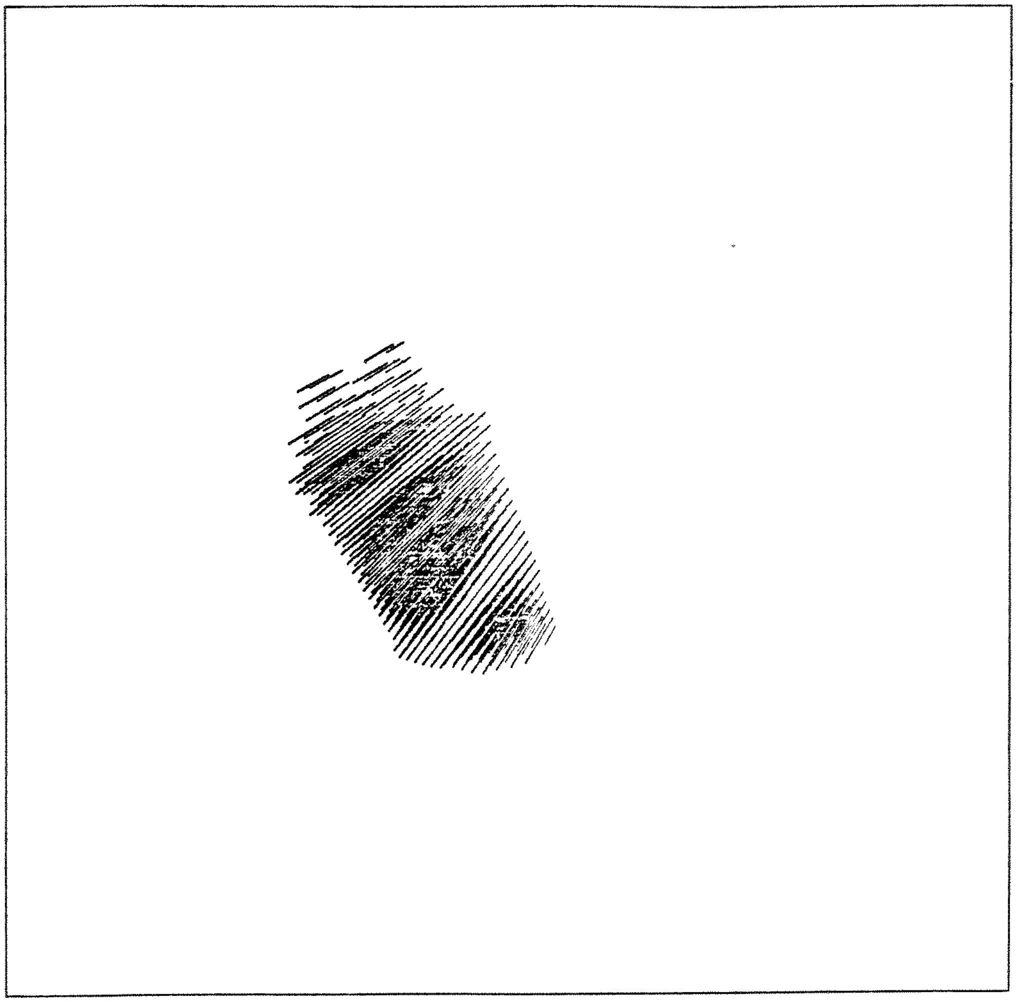

Figure 16. The final volume segment representation for the dynamic scene.

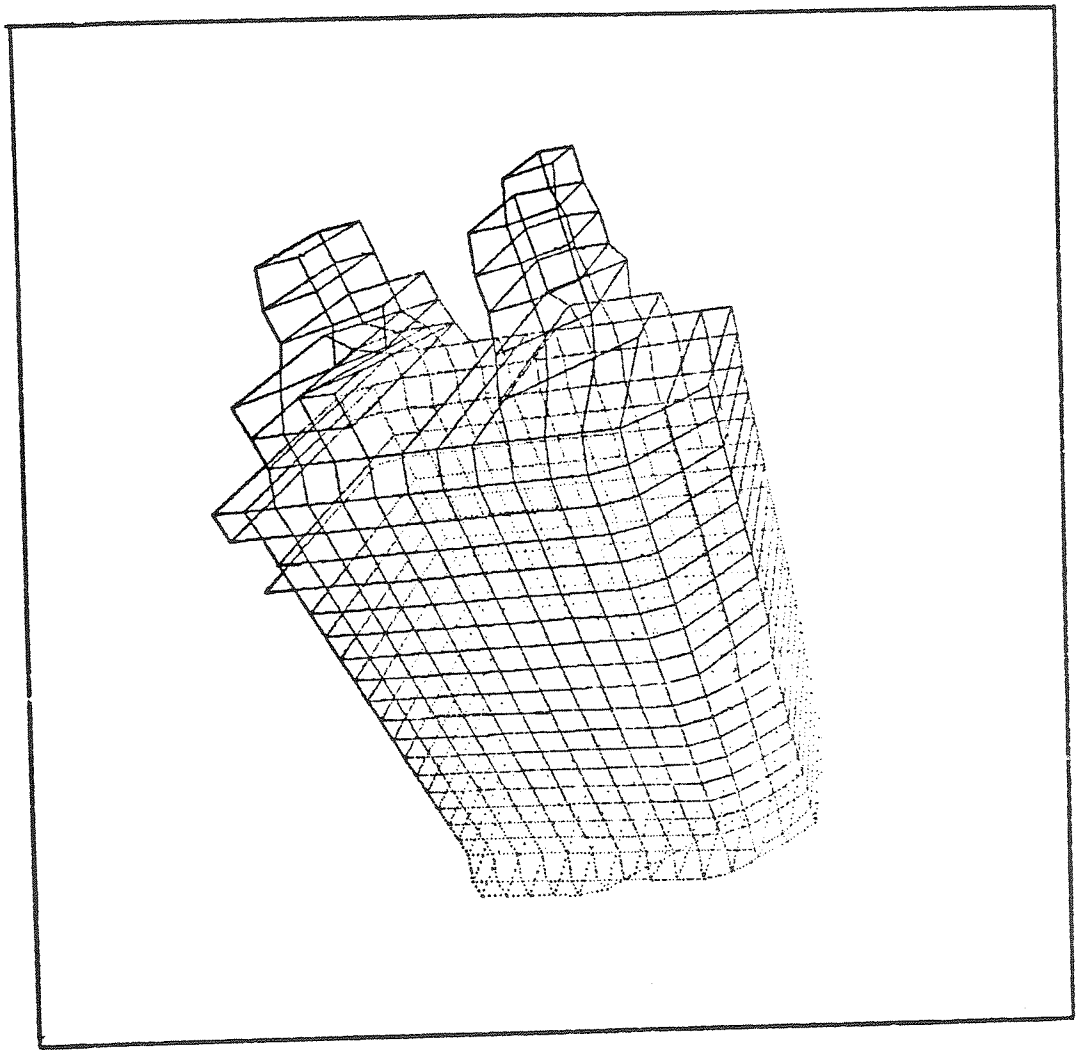

Figure 17. The general surface description derived from the volume segment representation illustrated in Figure 16.

6. Conclusion

The two major goals pursued in this work are, first, to lessen the dependence on feature point measurements in a structure from multiple views system, and second, to develop a descriptive three-dimensional object representation that was suitable for a dynamic process of volume refinement. The results are two fold: (i) a system has been developed which constructs a volumetric structure for an object from a sequence of occluding contours, and (ii) an algorithm has been formulated for the representation, and refinement of this structure.

The occluding contours with viewpoint specifications from a dynamic image are analyzed to initially form and continually update a description of the three-dimensional object generating the contours. The description is a bounding volume for the object and is successively refined to yield finer approximations to the actual object. Of course, from the silhouettes that form the occluding contours it is not possible to resolve certain kinds of concavities. In particular, object surface points for which every tangent line (in the tangent plane, any line that contains the given surface point) also intersects the object at some non-surface point cannot be resolved using silhouettes. However, the class of objects that can be described exactly is large, and in fact, the object surface may have saddle points and holes.

Clearly, to analyze the structure of objects a system must provide a representation scheme. For three-dimensional objects many different schemes have been proposed and used. The details of the representations usually are determined either by the data acquisition techniques or by the ultimate application of the system, with an important problem being the development of methods for transforming between structures of the first type and structures of the second type, see Aggarwal et al [46]. The volume segment representation described in this paper has been developed to facilitate the acquisition of three-dimensional information from dynamic images. The main attributes of the volume segment representation are that it is easy to update (as required by the continual refinement), maintains fine surface detail, simplifies the point inclusion test, and can be readily transformed into a surface representation.

For these reasons the work presented in this paper provides an excellent basis for further research. In particular, the work is appropriate to industrial automation applications. For example, selecting one of several parts on a conveyor using the views taken from several cameras fixed along the line of travel, or in conjunction with a manipulator arm that could successively reposition a part until an adequate approximation was derived. In such applications there are usually fixed sets of possible objects from which an unknown object must be recognized, implying the need for a representation scheme suitable for creating a library of possible objects, describing the unknown sample and matching it to the library entries. Future research should be directed toward exploiting the methods developed here in those applications.

References

1. L. G. Roberts, "Machine perception of three-dimensional solids," in Computer Methods in Image Analysis, J. K. Aggarwal, R. O. Duda, and A. Rosenfeld, Editors, IEEE Press, New York, 1977, pp.285-323.

2. I. C. Braid, "The synthesis of solids bounded by many faces," Communications ACM, vol. 18, no. 4, April 1975, pp. 209-216.

3. A. A. G. Requicha, "Representations for rigid solids: Theory, method and systems," Computing Surveys, vol. 12, no. 4, 1980, pp. 437-464.

4. G. J. Agin and T. O. Binford, "Computer description of curved objects," IEEET-Computers, vol. C-25, April 1976, pp. 439-449.

5. J. Hollerbach, "Hierarchical shape description of objects by selection and modification of prototypes," MIT AI-TR-346, Nov. 1975.

6. B. I. Soroka and R. K. Bajcsy, "A program for describing complex three-dimensional objects using generalized cylinders as primitives," IEEE Conference on Pattern Recognition and Image Processing, Chicago, IL, 1978.

7. R. Nevatia, and T. O. Binford, "Description and recognition of curved objects," Artificial Intelligence, vol. 8, 1977, pp. 77-98.

8. D. Marr, "Visual information processing: the structure and creation of visual representations," Proc. IJCAI-6, Tokyo, August 1979, 1108-1126.

9. J. O'Rourke and N. Badler, "Decomposition of three-dimensional objects into spheres," IEEET-Pattern Analysis and Machine Intelligence, vol. PAMI-1, no. 3, July 1979, pp. 295-305.

10. H. Blum, "A transformation for extracting new descriptors of shape," in Models for the perception of speech and visual form, W. Dunn, ed., MIT Press, Cambridge, MA, 1964, pp. 362-380.

11. B. G. Baumgart, "Geometric Modeling for Computer Vision," Stanford AI Lab Memo AIM-249, Oct. 1974.

12. H. B. Voelcker, A. A. G. Requicha, E. E. Hartquist, W. B. Fisher, J. E. Shapiro and N. K. Birrell, "An introduction to PADL: characteristics, status, and rationale," Production Automation Project Technical Memorandum TM-22, University of

Rochester, Dec. 1974.

13. H. G. Barrow and J. M. Tenenbaum, "MSYS: A System for Reasoning About Scenes," SRI AI Center, Tech. Note 121, March 1976.

14. A. Hansen and E. Riseman, "VISIONS: A computer system for interpreting, scenes," Computer Vision Systems, A. Hansen and E. Riseman, Editors, Academic Press, New York, 1978, pp. 303-333.

15. R. A. Brooks, "Symbolic reasoning among 3-D models and 2-D images," Report no. STAN-CS-81-861, Dept. of Computer Science, Stanford University, June 1981.

16. E. Miyamoto and T. O. Binford, "Display generated by a generalized cone representation," IEEE Conference on Computer Graphics and Image Processing, May 1975.

17. J. H. Clark, "Hierarchical geometric models for visible surface algorithms," Communications ACM, vol. 19, October 1976, pp. 547-554.

18. N. Badler and R. K. Bajcsy, "Three-dimensional representations for computer graphics and computer vision," Computer Graphics, vol. 12, no. 3, August 1978, pp. 153-160.

19. D. Huffman, "Impossible objects as nonsense sentences," in Machine Intelligence 6, B. Meltzer and D. Michie, Editors, Edinburgh University Press, Edinburgh, Scotland, 1971.

20. M. Clowes, "On seeing things," Artificial Intelligence Journal, vol. 2, no. 1, 1971, pp. 79-116.

21. D. Waltz, "Understanding line drawings of scenes with shadows," in Psychology of Computer Vision, P. H. Winston, Editor, McGraw-Hill, 1975, pp. 19-92.

22. A. Mackworth, "Interpreting pictures of polyhedral scenes," Artificial Intelligence Journal, vol. 4, 1973, pp. 121-137.

23. P. M. Will and K. S. Pennington, "Grid coding: a preprocessing technique for robot and machine vision," Second IJCAI, London 1971, pp. 66-70.

24. B. K. P. Horn, "Understanding image intensities," Artificial Intelligence Journal, vol. 8, 1977, pp. 201,231.

25. R. J. Woodham, "A cooperative algorithm for determining surface orientation from a single view," Proc. Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA, 1977, pp-635-641.

26. Y. Shirai and M. Suiva, "Recognition of polyhedrons with a range finder," Proc. Second International Joint Conference on Artificial Intelligence, London, 1971, pp. 71-79.

27. H. Freeman and M. Potmesil, "Curved surface representation utilizing data extracted from multiple photographic images," Proc. of the Workshop on the Representation of Three-dimensional Objects, Philadelphia, PA, 1979.

28. W. N. Martin and J. K. Aggarwal, "Survey: dynamic scene analysis," Computer Graphics and Image Processing, vol. 7, no. 3, June 1978, pp. 356-374.

29. H.-H. Nagel, "Image Sequence Analysis: What Can We Learn from Applications?," in Image Sequence Analysis, T. S. Huang, Editor, Springer-Verlag, Heidelberg, Germany, 1981, pp. 19-228.

30. J. K. Aggarwal, L. S. Davis and W. N. Martin, "Correspondence processes in dynamic scene analysis," Proc. of the IEEE, vol. 69, no. 5, May 1981, pp. 562-572.

31. S. Ullman, The Interpretation of Visual Motion, MIT Press, Cambridge, Massachusetts, 1979.

32. J. W. Roach and J. K. Aggarwal, "Determining the movement of objects from a sequence of images," IEEET-Pattern Analysis and Machine Intelligence, vol. PAMI-2, no. 6, 1980, pp. 554-562.

33. H.-H. Nagel, "Representation of moving rigid objects based on visual observations," Computer, vol. 14, no. 8, August 1981, pp. 29-39.

34. R. Nevatia, "Depth measurement by motion stereo," Computer Graphics and Image Processing, vol. 15, 1976, pp. 203-214.

35. H. H. Baker, "Three-dimensional modelling," Second IJCAI, London, 1977, pp. 649-655.

36. J. A. Webb and J. K. Aggarwal, "Visually interpreting the motion of objects in space," Computer, vol. 14, no. 8, August 1981, pp. 40-46.

37. A. Rosenfeld and A. C. Kak, Digital Picture Processing, Academic Press, New York, 1976, pp. 336-347.

38. W. E. Newman and R. F. Sproull, Principles of Interactive Computer Graphics, Second Edition, McGraw-Hill, New York, 1979, pp. 229-245.

39. W. N. Martin and J. K. Aggarwal, "Analyzing dynamic scenes," TR-81-5, Laboratory for Image and Signal Analysis, University of Texas at Austin, Dec. 1981, pp. 51-56.

40. G. T. Herman, Image Reconstruction from Projections: The Fundamentals of Computerized Tomography, Academic Press, New York, 1980.

41. C. Levinthal and R. Ware, "Three-dimensional reconstruction from serial sections," Nature, vol. 236, March 1972, pp. 207-210.

42. H. Fuchs, Z. M. Kedem, and S. P. Uselton, "Optimal surface reconstruction from planar contours," Communications ACM, vol. 20, October 1977, pp. 693-702.

43. D. Nitzan, A. E. Brain and R. O. Duda, "The measurement and use of registered reflectance and range data in scene analysis," Proceedings of the IEEE, vol. 65, Feb. 1977, pp. 206-220.

44. G. T. Herman and H. K. Liu, "Dynamic boundary surface detection," Computer Graphics and Image Processing, vol. 7, 1978, pp. 130-138.

45. C. L. Jackins and S. L. Tanimoto, "Oct-trees and their use in representing three-dimensional objects," Computer Graphics and Image Processing, vol. 14, 1980, pp. 249-270.

46. J. K. Aggarwal, L. S. Davis, W. N. Martin and J. W. Roach, "Survey: Representation methods for three-dimensional objects," in Progress in Pattern Recognition, vol. 1, L. Kanal and A. Rosenfeld, Editors, North-Holland Pub. Co., Amsterdam, The Netherlands, 1981, pp. 337-391.