MATHEMATICAL MODEL FOR SOLIDS UNDER THE SURFACE OF THE EARTH*

Z. SIKI

Department of Surveying, Institute of Geodesy, Surveying and Photogrammetry, Technical University, H-1521 Budapest

Received July 3, 1989

Presented by Prof. Dr. F. Sárközy

Abstract

Geometric models for solids under the surface of the Earth are discussed in this paper. We have developed a two and a half dimensional model and program for geologic applications. We introduce a conceptual scheme for the full 3D geometric model as well.

Preface

Many systems were developed to handle and manipulate three dimensional objects. The developers of these systems tried to implement general purpose systems for mechanical engineering or architecture. There is no standard to model mineral resources for geological or mining aspects.

We have been trying to develop a two and a half and a three dimensional geometric model for these purposes. We examined the method to model three dimensional objects, then developed a mixed model to describe natural and artifical objects under the surface of the earth. We considered some of the necessary algorithms to manipulate our model and the restrictions involved in it.

An overview of solid modeling methods

Well known geometric models [5, 8, 9, 16, 17, 19] are discussed shortly in this paragraph. We have considered only the methods usable for our aims:

- sweeping
- Constructive Solid Geometry (CSG)
- cell decomposition
- Boundary Representation (BR)

Sweeping is useful to model prism-like or solid of rotation-like objects. An object is created by mowing a volume or polygon along a trajectory. There are two types of sweeping, translational and rotational.

* The National Scientific Research Foundation gives financial assistance to the research discussed in this paper.

Constructive Solid Geometry (CSG) uses set operations to represent a solid. Generally three operations are used: section ($\cap$), union ($\cup$) and set difference (—). Sometimes a fourth operation is introduced called glueing (G). A solid is represented as a binary tree where the nodes are set operations and the leaves are primitives (half spaces or previously defined solids). A half space divides the space into two parts inside and outside. CSG is a very good model for regular objects and the set operations are useful not only for storing but creating and modifying them.

Cell decomposition is a special case of CSG. There is only one type of primitives (cells) topologically and the cells are glued (no holes or common interiors among the adjacent primitives). Two types of primitives are usually used: regular (cube or block) or irregular (tetrahedron). Cube cells are usually encoded into octree. In case of tetrahedrons the points of the solid to model are used to make the decomposition. We get a linear approxmation of the solid. With regular cells the approximation is worse but the regular elements can be handled more easily. Cell decomposition is used to model sculptured objects and it is suitable for finite element analysis.

Boundary representation is the nearest to human sense. The solids are described by their boundaries. Boundary is represented by faces, edges and vertices with the topology. BR indicates a hierarchic data structure with one to many relations.

The developed models

At the beginning of our research we divided our task into two parts:

- not fully 3D model
  (so called $2^1/_2$ D model)
- full 3D model

A two and a half dimensional model for geologic applications

The $2^1/_2$ D model supports geologic applications for larger areas with lower resolution. The model is based on layers. Each layer has an upper and a lower boundary which must be single valued, with only one z value for each x, y pair. The boundaries which do not fulfil this condition have to be divided into single valued parts. The boundaries must be given by scattered points or contour lines. From these data a triangle mesh is generated [14] for each boundary. The model is limited to a block called model space. A layer can be defined for the whole model space or for one or more parts of it. The boundaries of partly defined layers must be closed (e.g. the upper and lower border must meet exactly) otherwise the calculated volumes and surfaces will be false.

Besides layers faults are allowed. This part of the model is being worked out. Faults can be stored as planes with a validity polygon. The elevations at the two sides of the face of a fault can be calculated in two ways. It can be interpolated from points of border on the same side of the face of fault or from given values at the intersection of the fault and the boundary.

A program was developed for IBM PC XT/AT or compatibles to manipulate the $2^1/_2$ D model. This program is really a graphic inquiry system which can be used with some IBM PC standard graphic cards CGA, Hercules, EGA, VEGA. The program makes it possible to display the whole model or a part

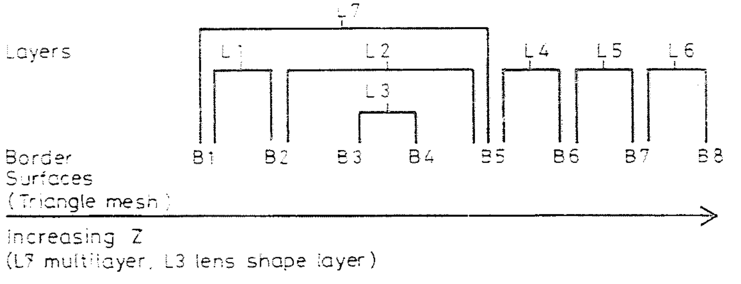

Fig. 1. Logical structure of the $2^1/_2$D model

of it and to modify the borders and faults locally. Each function of the program can be reached through pop-up menus which can be manipulated with keyboard or mouse.

Different display methods are implemented to get the most from stored data. The following types of projection are supported:

— perspective
— axonometry
— axonometry to the xy plane (map)
— axonometry to the plane of section (profile)

A part of the model can be displayed cut by vertical planes which may be given by a convex or concave polygon in the xy plane. The graphic screen can be divided into several viewports to view the model from different points of view at the same time. The resolution of the graphic screen highly influences the rational numbers of non-overlapping viewports on the screen.

There are different methods to display the upper boundary of the highest perspectivic or axonometric projected layers. Triangle mesh, rectangle mesh, contours can be selected to make the picture more spectacular.

Only the selected layers are included in the visualisation. Selection can be made by attributes of layers or individually. The surface of the borders and the volume of selected layers can be calculated.

Besides the screen only Epson printers are allowed as output device (alphanumeric or graphic hardcopy) but we plan to implement plotter drivers and logical output to other systems.

3D Model for mining and geological applications

Many 3D solid modeling systems were developed for engineering applications. However, they cannot be applied in our case because complete information is needed for the modeling solid. We have only incomplete information about the distribution of substances under the surface of the Earth. Information can be collected from different sources:

— drill holes
— maps
— geophysical observations
— surveying of caves, mines, the surface of the Earth.

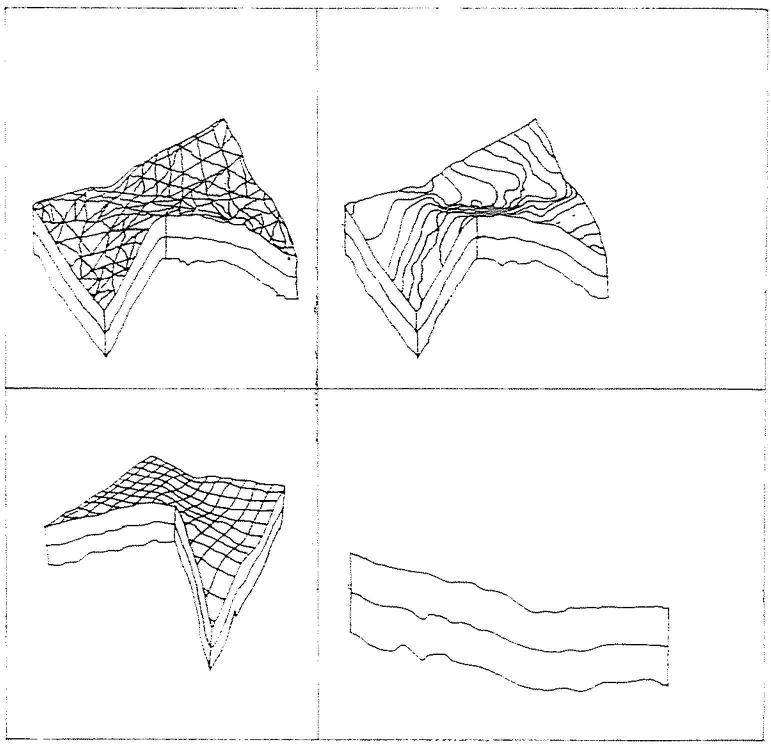

Fig. 2. Output of the $2^1/_2$D system

Another significant difference to engineering applications is that the border among different minerals cannot be drawn unambiguously. The border can be defined by a cut off value (a given percent of a mineral). The cut off value depends on the kind of mineral, economic mining etc. From this point of view scalar fields would be the most suitable to model them. The distribution of minerals can be estimated from the incomplete information like a scalar-vector function:

$$U = f(\mathrm{v}) = f(x, y, z)$$

The scalar-vector function and a cut off value determine the shape of the solid. The function and the cut off value define a set (half space) as:

$$f(x, y, z) \leq \text{cut off value}$$

These types of sets can be used with CSG because only an inside-outsids function is needed for CSG to manipulate solids or can be transformed to cell decomposition.

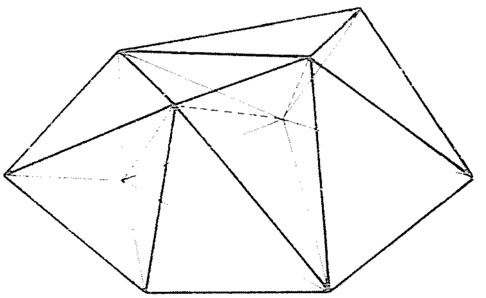

Fig. 3. Tetrahedron cells

If we consider that not only natural but artificial (man-made) objects are under the surface of the earth (drifts, pits etc.) it would be effective to use the same or similar geometric-model for both natural and artificial objects. In the case of artificial objects BR, CSG or sweeping would be suitable, in contrast with natural objects where cell decomposition would be used as the most efficient method to store sculptured solids. BR would be good in both cases if the volume were not so important in mining applications. So it seems that different methods would be good to model the two types of objects. But if we consider that cell decomposition is a special case of CSG then we can realize that cell decomposition and CSG can be combined easily without the permanently repeated transformations from one model to the other.

Only linear geometric elements are allowed in our model except scalar fields. This linear approximation [23] does not deform significantly our model because:

a) any geometric element can be approximated by linear elements with arbitrary precision

b) this error is smaller than the error of the estimation from the incomplete information.

We have to distinguish caves from other natural objects because they can be surveyed. If representative cross sections are measured in the cave then its boundary can be approximated with a spatial triangle mesh [11, 14]. It is true for existing artificial objects as well.

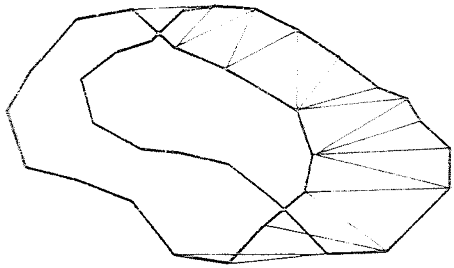

Fig. 4. Triangle mesh from cross sections

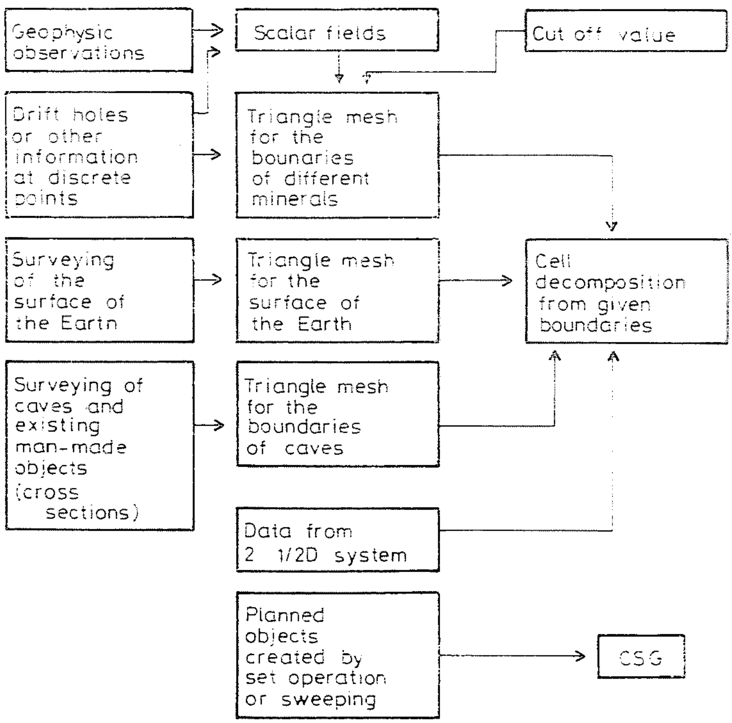

Fig. 5. Conceptual scheme for 3D model

We prefer tetrahedrons in cell decomposition to blocks because tetrahedrons give a better approximation and the number of tetrahedrons is independent of the resolution. Trigonal prisms may be used as cells if only vertical drift holes are given. A trigonal prism can be decomposed into three tetrahedrons.

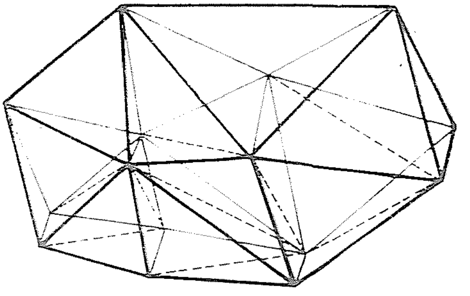

Fig. 6. Prisms and tetrahedrons

Tetrahedrons can be stored simply by their four vertices. No topology is needed because any three points of the tetrahedron make a face. Another possibility to store tetrahedrons is by their four planes like CSG. There is no need to store the binary tree because it is the same for every tetrahedron.

Manipulation of solids given by CSG binary tree

Binary trees are widely used in computing because many data structures can be modeled by them and their structure is fairly simple. Recursive data structures and algorithms are generally used in connection with binary trees. The following Pascal declarations define a minimal binary tree structure to store solids with plane faces.

```
type
    node_type = (section, union, difference, plane);
    node_ptr  = ^node;
    node      = record
    case typ: node_type of section, union, difference:
          (left_child: node_ptr;
           right_child: node_ptr);
           plane: (a, b, c, d: real);
    end;
```

Using this data structure a recursive inside outside function can be defined for a spatial point.

```
function inside (x, y, z: real; root: node_ptr): boolean;
begin
      inside: = false; {suppose outside}
if root < > NIL then  {shift out empty solid}
with root ^ do
case typ of
  section:
      inside: = inside (x, y, z, left_child) and
                inside (x, y, z, right_child);
  union:
      inside: = inside (x, y, z, left_child) or
                inside (x, y, z, right_child);
  difference:
      inside: = inside (x, y, z, left_child) and
                not inside (x, y, z, right_child);
  plane:
      inside: = (a*x + b*y + c*z + d)> = 0;
  end; {case}
end; {inside}
```

The set operations can easily be made between two solids. Only their binary trees have to be connected with the given operator (section, union, difference). No calculation is needed at all. Besides this benefit there are some disadvantages of CSG. We cannot display a solid directly stored in binary tree. Another problem with CSG is that we cannot find out easily if the solid is empty (e.g. empty set). Unfortunately this type of description of solids is not unique, the same solid can be described by several different binary trees. But this problem can be solved by transforming the tree into a standard format [12, 20, 24].

Let us go down to the displaying problem of solids stored in CSG structure. Two solutions can be imagined:

a) by the transformation of the CSG structure to another for example to BR

b) the binary tree contains implicitly every information needed to display it, let us try to pull it out.

First let us deal with the transformation of the CSG structure to BR. The faces and the vertices of the solid can also be described by the same binary tree. We can define the co-planar one or more faces of the solid by the same binary tree only one half space has to be replaced by its border (e.g. instead of $a * x + b * y + c * z + d > 0$ use $a * x + b * y + c * z + d = 0$). To get co-linear one or more edges two half spaces have to be replaced by their border. At last three half spaces have to be replaced by their border to get a vertex. Replacing half spaces by their border (in our case it is a plane) the result may be an empty set. This way the faces, edges, vertices of a solid can be defined as set operations. It is not enough to draw them yet. Further recursive algorithms have to be applied to solve the whole problem [21, 22]. It is not a very easy and quick way to display a solid.

Let us try to solve the displaying problem from the original CSG tree without transformations. The z buffer algorithm can help us [7, 15]. It is a hidden surface eliminating algorithm. By the help of z buffer algorithm a shaded picture of the solid can be displayed. This algorithm supposes a colour raster display device. The z buffer algorithm calculates depth information at each pixel for every geometrical object which hides it (e.g. the projecting ray to the pixel intersects the border of solid) and displays the nearest with a convenient colour. Since the border of solid is not known in CSG we have to extend the algorithm. We can only calculate the point of intersection between projection ray and the border of half spaces (plane). After it we have to decide whether the point of intersection belongs to the solid by the help of inside function. The colour of the pixel can be calculated from the normal vector of the border at the point of intersection and a light direction [7, 15]. This modified z buffer algorithm can be used for nonlinear half spaces as well, e.g. sphere, cylinder, hyperboloid, torus.

References

1. Barnnhill, R. E.: Surfaces in Computer Aided Geometric Design, North Holland, 1983.
2. Bayse, J. W—Gilchrist, J. E.: GMSolid: Interactive Modeling for Design and Analysis of Solids, IEEE Computer Graphics and Applications, Vol. *2*, March, 1982.
3. Cardenas, A. F.: Data Base Management Systems, Allyn and Bacon, Boston, 1979.
4. Encarnacao, J.: File Structures and Data Bases for CAD, North Holland, Amsterdam, 1982.
5. Encarnacao, J.: Computer Aided Design, Modeling, Systems, Engineering, CAD Systems, Lecture Notes in Computer Science, Springer-Verlag Darmstadt, Berlin, 1980.
6. Encarnacao, J.—Schlechtendahl, E. G.: Computer Aided Design, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1983.
7. Foley, J. D.—Van Dam, A.: Fundamentals of Interactive Computer Graphics, Adison-Wesley Publ. Co, New York, 1982.
8. Hillyard, R.: The Built Group of Solid Modelers, IEEE Computer Graphics and Applications, Vol. *7*, March, 1987.
9 Jackins, C. L.—Tanimoto, S. L.: Octrees and their Use in Representing 3D Objects, Computer Graphics *F* Image Processing, Vol. *14*, No 5, 1980.
10. Kalay, Y. E.: Worldview: An Integrated Geometric Modeling/Drafting System, IEEE Computer Graphics and Applications, Vol. *7*, February, 1987.
11. Karonen, O.: Geometric Mine Modeling: Modeling of Three-Dimensional Objects Based on Incomplete Information, Acta Polytechnica Scandinavica, Helsinki, 1985.
12. Lee, Y. C.—Fu, K. S.: Machine Understanding of CSG: Extraction and Unification of Manufacturing Features, IEEE Computer Graphics and Applications, Vol. *7*, January, 1987.
13. Mantyla, M.—Sulonen, R.: GWB: A Solid Modeler with Euler Operators, IEEE Computer Graphics and Applications, Vol. *2*, March, 1982.
14. Mirante, A.—Weingarter, N.: The Radial Sweep Algorithm for Constructing Triangulated Irregular Networks, IEEE Computer Graphics and Applications, Vol. *2*, May, 1982.

15. NEWMAN, W. M.—SPROULL, R. F.: Principles of Interactive Computer Graphics, Mc-Graw-Hill Inc, New York, 1979.
16. REQUICHA, A. A. G.—VOELCKER, H. B.: Solid Modeling: A Historical Summary and Contemporary Assessment, IEEE Computer Graphics and Applications, Vol. *2*, March, 1982.
17. REQUICHA, A. A. G.: Representation for Rigid Solids: Theory, Methods and System, ACM Computing Surveys, Vol. *12*, December, 1980.
18. SAMET, H.—WEBBER, R. E.: Hierarchial Data Structures and Algorithms for Computer Graphics Part I, IEEE Computer Graphics and Applications, Vol. *8*, May, 1988.
19. SAMET, H.—WEBBER, R. E.: Hierarchial Data Structures and Algorithms for Computer Graphics Part II, IEEE Computer Graphics and Applications, Vol. *8*, July, 1988.
20. SEGAL, M.—SEQUIN, C. H.: Partitioning Polyhedral Objects into Nonintersecting Parts, IEEE Computer Graphics and Applications, Vol. *8*, January, 1988.
21. TILOVE, R. B.: Set Member Classification: A Unified Approach to Geometric Intersection Problems, IEEE Transactions on Computers, Vol. *C-29*, No 10, 1980.
22. TILOVE, R. B.—REQUICHA, A. A. G.: Closure of Boolean Operations on Geometric Entities, Computer Aided Design, Vol. *12*, No 5, 1980.
23. TURNER, J. U.: Accurate Solid Modeling Using Polyhedral Approximations, IEEE Computer Graphics and Applications, Vol. *8*, May, 1988.
24. WOODWARK, J. R.: Eliminating Redundant Primitives from Set-Theoretic Solid Models by a Consideration of Constituents, IEEE Computer Graphics and Applications, Vol. *8*, May, 1988.

Zoltán SIKI H-1521, Budapest