3D DATA STRUCTURES AND APPLICATIONS IN GEOLOGICAL SUBSURFACE MODELING

R. Li, Y. Chen, F. Dong and L. Qian
Dept. of Geomatics Engineering, The University of Calgary
2500 University Dr., N.W., Calgary, AB, Canada
Tel: (403) 220-4112 Fax: (403) 284-1980
E-mail: rli@acs.ucalgary.ca
WWW: http://loihi.ensu.ucalgary.ca/

J.D. Hughes
Institute of Sedimentary and Petroleum Geology
Geological Survey of Canada
3303-33St NW, Calgary, Alberta T2L 2A7

KEY WORDS: 3D data structures, octree, Peano keys, subsurface, GeoView

ABSTRACT:

In this paper, a brief discussion is given to the existing problems in 3D geological subsurface modeling. Two groups of 3D data structures are then identified with the focus on the octree representation and its Peano coding. Two spatial operations based on Peano keys are described in details. Finally, the experimental results of the developed system, GeoView, are given.

1. BACKGROUND

Geographic Information Systems (GIS) have been widely used and have shown their power in spatial analysis, database management and various other applications. The extensive use of computers in oil and mining explorations has introduced the possibility of applying the GIS technology to the field of geology and geophysics.

However, geological subsurface modeling involves usually a large volume of digital information which is spatially referenced in three dimensions. The use of traditional interpretation methods (e.g., contour maps, cross sections, fence diagrams, isometric surfaces) limits the view of the geological world to two dimensions, or at best quasi-three dimensions (Fisher and Wales, 1992). The accumulation of the geological information calls for a GIS system capable of handling 3D data efficiently. This system should combine the search and analysis of databases with the versatility for visualization using computer graphics technology (Jones, 1989). Currently, most GIS systems are based on two dimensional data structures and could not handle information in the third dimension efficiently. Attempts have been made to extend spatial operations to 3D by adding the third dimensional information into 2D data structures, such as including elevation data as attributes (Arc/Info, 1992). Although such systems have 3D information in databases, the full 3D functionality cannot be expected, such as 3D modeling, query of 3D spatial elements and association of non-spatial data to solid elements, etc.

Geological applications pose special challenges to the development of GIS. Geological phenomena are 3-dimensional in nature. When fit into 2-dimensional GIS systems, they are not accurately modeled, analyzed or displayed (Smith and Paradis, 1989). In geological modeling, all applications require increasingly quantitative and accurate underground material characterizations within the 3D subsurface environment. The 3D data are required because the depth dimension is in the same general range as the surface dimensions, and the true spatial relationships are important to the problem analysis (Turner, 1989).

3D geological modeling has identified a number of requirements on GIS systems. An ideal system should provide a variety of facilities, such as data storage of a large amount of volume data, input to and output from models, support for a range of data structures and for transformation between different data structures, integration of data from different sources, assistance with construction of models, integration of complex spatial relationship between geological layers, geostatistical techniques for data interpolation before model input, determination of model parameters, design of sampling strategies, and error analysis (Mason, et. al., 1994). Additional important facilities may be dynamic visualization and animation of the third dimensional spatial information.

This paper presents the results of a research project initialized in 1993 and conducted jointly by The University of Calgary and the Geological Survey of Canada to model and visualize 3D geological subsurface information using 3D data structures.

In this paper, a review of 3D data structures is given. A few 3D spatial operations developed and implemented in our system are discussed. The system itself is introduced in the last section, which has been implemented on a Silicon Graphics Workstation by using C programming language, Motif and Graphic Language (GL).

2. THREE DIMENSIONAL DATA STRUCTURES

Although 3D GIS is badly needed in geological applications, progress in this field has been relatively slow. This might generally due to difficulties in defining complex geological information and specifically to the need for finding data models and data structures suitable for handling large quantities of 3D geoscientific data. Many recent advances in the design of GIS are applicable, but in general this technology is oriented towards two rather than three-dimensional information (Jones, 1989).

Depending on geometric characteristics, 3D objects can be described by two groups of geometric representations, namely, surface-based and volume-based representations (Li, 1994).

The surface-based 3D representations describe geometric characteristics of objects by surface entities. There are four types of them, namely, grids, shape models, facet models, and boundary representations. Grids and shape models are in raster format and the facet models and boundary representations are in vector format. In general, the grids, shape models and facet models are suited for describing object surfaces with irregular shapes; while boundary representations give the exact surface geometry of objects with regular shapes.

Volume-based representations describe the interior of objects by using the solid information instead of surface information. With these representations, the solid information of objects can be presented, analyzed and visualized. 3D binary arrays, needle models, octrees and CSG (Constructive Solid Geometry) belong to this group. CSG is in vector format. Generally speaking, 3D binary arrays, needle models and octrees are capable of modeling objects with irregularly shaped objects; while CSG is well suited for modeling regularly-shaped objects.

Most of the data structures mentioned above, to some extent, are characterized by a restricted domain of representable objects, due to the fact that the objects are constructed from a limited number of mathematically well-defined surfaces or solid primitives (Meagher, 1982). For example, surface-based representations may suffer from inefficiency when geometric and Boolean operations are of high priority.

Being an extension of a 2D quadtree, an octree models 3D volumetric objects by recursively subdividing the object space. An object is represented by the root node in the form of a cube, which is then subdivided into eight octants to form the first level of the tree. Octants that are not entirely occupied by the object are called partial octants and are further subdivided to smaller octants until each suboctant is either empty, fully occupied by the object, or until the desired resolution is reached. Extended octrees can model objects of any shapes (Laurini and Thompson, 1992). The storage requirement of octrees is much low in comparison to traditional raster methods. Octree remains also advantageous because of its raster structure and efficient geometric and Boolean operations based on spatial indexing and relational operations. Furthermore, multi-resolution modeling is another important characteristic of octrees.

Kavouras and Masry (1987) applied the octree method in a geological environment. They demonstrated the use of linear octrees for storing a model of a gold ore deposit, using a system called Daedalus. Algorithms for octree-based geometric operations, such as translation, scaling, rotation, and perspective transformation, have been investigated by Jackins and Tanimoto (1980), Meagher (1982), and Laurini and Thompson (1992).

In the presented research, octrees are adopted for geological subsurface modeling because of its aboved mentioned advantages.

3. SPATIAL OPERATIONS BASED ON PEANO KEYS

Indexing methods have a great impact on the efficiency of octree based spatial operations. The relative merits of five methods, including row, row prime, Hilbert, Morton (or Peano) and Gray code, were assessed by Abel and Mark (1990). It was concluded that the Morton ordering provides a great efficiency in window searching and has been used in most linear quadtree systems.

It is noted that a linear octree, a pointerless structure first proposed by Gargantini (1982), locates any node in a tree by a unique key which is obtained by interleaving binary coordinates of X, Y and Z. This can be implemented by using Peano coding. It reduces the amount of storage of octree representations and provides an efficient spatial indexing method. Spatial geological operations can be developed based on Peano keys for geometric operations, Boolean operations and visualization.

The Peano keys can be generated by interleaving binary coordinates of (X, Y, Z). For example, an octant with P(O)=14 has decimal coordinates (1, 1, 2). The corresponding binary coordinates are (01, 01, 10). Interleaving the binary coordinates in the order of Z, Y, and X for every bit starting from lower bits results in the binary Peano key 001110 which is 14 in decimal. This nature of the Peano keys makes the storage of Peano keys compact and the conversions between Peano keys and coordinates very efficient. In addition, neighbourhood searching in a Peano encoded octree is efficient because of the characteristics of the N curve (Laurini and Thompson 1992). An octant with a Peano key of P and a size of S is expressed as O(P, S).

Peano coding has been chosen for implementing octrees in this project due to its efficiency in storage and spatial indexing, inheritance of relationships with spatial geometry and its capability of preserving spatial contiguity. Out of a number of subsurface oriented algorithms, two particular ones based directly on the Peano keys are presented in this paper. The first algorithm is used for cutting an octree representation by a plane with any orientation. The second one performs a geological datum adjustment.

3.1 Plane cutting algorithm based on Peano codes

One of the important operations in the geological subsurface modeling is to build a fence diagram along a defined path on the subsurface model. The geologists are then able to visualize the profiles along the fence diagram and analysis the subsurface structures.

The basic component and a simplified case of the above-mentioned operation is to define a cutting plane on the subsurface model and to remove the front part of the model so that the user is able to visualize the profile along the cutting plane. An algorithm is developed to perform this function on the octree representation of the subsurface model based directly on the Peano codes. To explain the algorithm, the two dimensional case (quadtree representation) is used. The plane cutting algorithm is simplified to the line cutting.

Figure 1 shows a quadtree representation with three levels. Each smallest quadrant is labeled with its corresponding Peano code. A cutting line is defined as Ax + By + C = 0. For each cutting line, we get the slope K from the equation as K = -A/B. We further decompose K to the x-component K_x = 1 (or -B/A) and y-component K_y = -A/B (or 1).

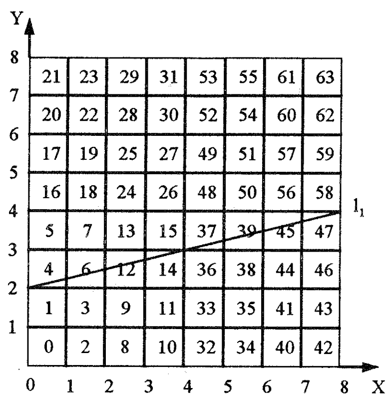


Figure 1 Line Cutting on a Quadtree Representation

For each cutting line, the intersecting points with the boundary of the quadtree representation are determined. The Peano codes of those intersecting points can then be calculated simply by interleaving the x and y coordinates. Once we get the two intersecting points, the next step is to search for those quadrants along the cutting line. The final step is to remove the front quadrants.

1) Determine the intersecting points

In Figure 1, the cutting lines l_1 is defined as:

l_1: x - 4y + 8 = 0, K = -A/B = 1/4,

K_x = 1, K_y = -A/B = 1/4.

The binary expressions of K_x and K_y are $K_x = 01_{(2)}$, $K_y = 2^{-2}{}_{(10)} = 0.01_{(2)}$. $K_{(2)}$ is then the interleaving of K_x and K_y, $K_{(2)} = 10.0001_{(2)}$.

Note the calculation of K_y is different because decimal digits are involved. We use the 2^n components to express the decimal part. If n = -1, we set a 1 at the first digit after the decimal point of binary number. If n = -2, we set a 1 at the second digit after the decimal point. This is similar for the subsequent digits. The order of interleaving for the decimal part is the same as for the integer part.

The starting intersecting point of the line with the boundary x = 0 is P_1 (0, 2).

2) Determine the Peano code for the starting quadrant

Once we have the starting intersecting point, the next step is to calculate the Peano code for the starting quadrant.

Let PC[0] denote the Peano code for the staring quadrant of the line. We could then calculate PC[0] by interleaving the x- and y-coordinate. For example, for point P_1 (0, 2)

$x = 0_{(10)} = 00_{(2)}$

$y = 2_{(10)} = 10_{(2)}$

$PC[0] = 0\,1\,0\,0_{(2)} = 4_{(10)}$

3) Search for the quadrants along the cutting line

To search for the remaining quadrants along the cutting line, we have to use the information about x- and y-components of the slope. To propogate from one point to its next point along the cutting line, the following equation is used:

$PC[i] = PC[i-1]_{(2)} + K_{(2)}$.

To demonstrate the usage of the above equation, the searching procedure along l_1 is shown as follows:

$PC[0] = 4_{(10)} = 0\,1\,0\,0_{(2)}$

$PC[1] = PC[0] + K_{(2)} = 0\,1\,0\,0_{(2)} + 1\,0\,.\,0\,0\,0\,1_{(2)}$
$= 0\,1\,1\,0\,.\,0\,0\,0\,1_{(2)} = 6.06_{(10)}$

In the above binary addition, if an increment occurs at a digit of x-component, then it will pass the increment to the higher digit of x-component, instead of the next direct digit. The same holds for y-components.

If the resulted Peano code is not an integer, the integer part is used as the code of the linear quadtree. The decimal part is kept for searching further quadrants along the cutting line.

$PC[2] = PC[1]\; 0\,1\,1\,0\,.\,0\,0\,0\,1_{(2)} + 1\,0\,.\,0\,0\,0\,1_{(2)}$
$= 1\,1\,0\,0\,.\,0\,1\,0\,0_{(2)} = 12.25_{(10)}$

$PC[3] = PC[2]\; 1\,1\,0\,0\,.\,0\,1\,0\,0_{(2)} + 1\,0\,.\,0\,0\,0\,1_{(2)}$
$= 1\,1\,1\,0\,.\,0\,1\,0\,1_{(2)} = 14.31_{(10)}$

$PC[4] = PC[3]\; 1\,1\,1\,0\,.\,0\,1\,0\,1_{(2)} + 1\,0\,.\,0\,0\,0\,1_{(2)}$
$= 1\,0\,0\,1\,0\,1\,.\,0\,0\,0\,0_{(2)} = 37_{(10)}$

$PC[5] = PC[4]\; 1\,0\,0\,1\,0\,1\,.\,0\,0\,0\,0_{(2)} + 1\,0\,.\,0\,0\,0\,1_{(2)}$
$= 1\,0\,0\,1\,1\,1\,.\,0\,0\,0\,1_{(2)} = 39.06_{(10)}$

$PC[6] = PC[5]\; 1\,0\,0\,1\,1\,1\,.\,0\,0\,0\,1_{(2)} + 1\,0\,.\,0\,0\,0\,1_{(2)}$
$= 1\,0\,1\,1\,0\,1\,.\,0\,1\,0\,0_{(2)} = 45.25_{(10)}$

$PC[7] = PC[6]\; 1\,0\,1\,1\,0\,1\,.\,0\,1\,0\,0_{(2)} + 1\,0\,.\,0\,0\,0\,1_{(2)}$
$= 1\,0\,1\,1\,1\,1\,.\,0\,1\,0\,1_{(2)} = 47.31_{(10)}$

Consequently, the Peano codes in the list are 4, 6, 12, 14, 37, 39, 45 and 47.

Once all the Peano codes along the cutting line are detected, the front quadrants can be removed.

3.2 Datum adjustment algorithm based on Peano codes

Datum adjustment is an operation in subsurface modeling to stretch the bottom surface of a layer S_0 to a plane. The top surface of the layer and other layers are changed accordingly by substracting the depths of the surface S_0. This operation is essential for estimating the volume of a lithology or viewing the other layers with respect to a certain layer.

The following are steps of this operations.

1) Save the depth offsets of the bottom octants $O_b(P, S)$
- Resolve the Peano key P and get the coordinates (x, y, z).
- Store the z to a working array as Z(x, y).
- Copy z to Z(x+i, y+j) with i = 1, 2, ...S-1 and j = 1, 2, ...S-1, if S > 1.

2) Adjustment operations on all octants O(P, S)

case 1 : the size of the octant O(P, 1) is 1.
- Resolve the Peano key P and get the coordinates (x, y, z).
- Subtract the bottom depth z' = z - Z(x, y).
- Interleave the coordinates (x, y, z') to calculate the new Peano key P'. The datum adjusted octant is O'(P', 1).

case 2 : the size of the octant O(P, S) is S ($S = 2^K$, K > 0)
- Resolve the Peano key P and get the coordinates (x, y, z).
- If Z(x, y) > 0 and Z(x, y) < S, subdivide the octant into 8 smaller octants and go to the beginning of step 2).
- If Z(x, y) > 0 and the Remainder[Z(x, y), S] ≠ 0, subdivide the octant into 8 smaller octants and go to the beginning of step 2).
- Check the values of Z(x+i, y+j) with i = 1, 2, ... S-1 and j= 1, 2, ... S-1. If there is any value different from Z(x, y), partition the octant and go to the beginning of the steps 2). Otherwise, subtract the bottom depth z' = z - Z(x, y). Interleave the coordinates (x,y,z') and get the new Peano key P'. The datum adjusted octant becomes O'(P', S).

3) Conformance check and aggregation
- Set the initial level i = 1.
- If the Remainder[P', 8^i] = 0 and $S = 2^{i-1}$, check the values of $P' + j*8^{i-1}$ (j = 1, 2, ...7). If all octants $O(P'+j*8^{i-1}, 2^{i-1})$ exist, aggregate them and create a new octant $O'(P', 2^i)$.
- i=i+1 and repeat the above procedure until all the Peano keys are processed and no aggregation can be made.

The following is an example based on a quadtree.

1) At first, we register the depth offset values of the bottom boundary. The octant O(4, 2) has the coordinates (0, 2). We have Z(0) = 2 and Z(1) = 2. Similarly, we have Z(2) = 1, Z(3) = 0, Z(4) = 1, Z(5) = 1, Z(6) = 2, Z(7) = 2.

2.1) The Peano key 18 can be resolved as its coordinates (1, 4). Since the size of the quadrant is 1, y' = 4 - Z(1) = 2. The new coordinates is (1, 2) corresponding to its Peano key 6.

Similarly, we process the octants with Peano keys 9, 10, 11, 33, 35, 44, 45, 46, 54 and 60. Their new Peano keys are 9→8; 10→10; 11→11; 33→32; 35→34; 44→40; 45→41; 46→12; 54→51; 60→56.

2.2) As for octants with Peano keys 4, 12, 24, 36, 48 and 56, their sizes are S = 2 (K = 1). For example, if P = 12, the coordinates are (2, 2). Since Z(2) = 1 < 2 = S, the quadrant is partitioned to 4 smaller quadrants 12, 13, 14 and 15, which have the size of 1. After processing, the smaller quadrants have the Peano keys of 9, 12, 14 and 15, respectively.

Similarly, quadrants with P = 24, 36, 48 produced the quadrants with P' = 13, 24, 26, 27, 33, 35, 36, 38, 37, 39, 48, 50.

For P = 56, the coordinates are (6, 4). Z(6) = 2 = S, Z(6) > 0 and the remainder [Z(6), 2] = 0. Since Z(7) = 2 = Z(6), y' = 4 - 2 = 2. The new coordinates are (6, 2). The new octant is O'(44, 2).

Finally, octant O(4, 2) becomes O(0, 2).

3) According to the procedure described above, octants O'(8, 1), O'(9, 1), O'(10, 1), O'(11, 1) can be aggregated to O'(8, 2). Similarly we have O'(12, 2), O'(32, 2), O'(36, 2).

At the level i = 2, no aggregation is required. The process is thus finished.

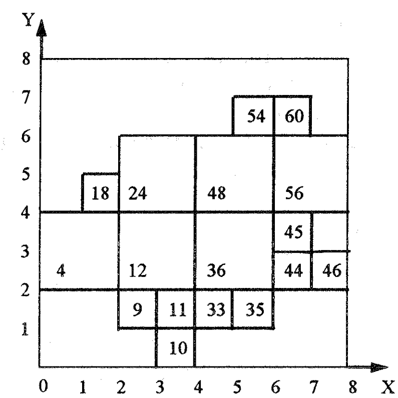


Figure 2. A quadtree before the datum adjustment

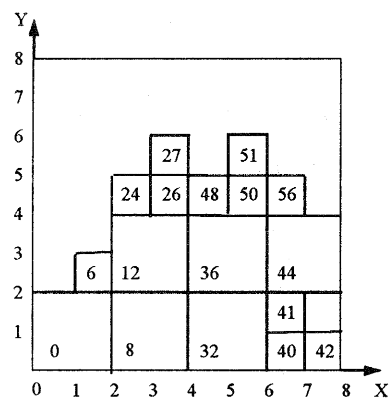


Figure 3. The quadtree after the datum adjustment

4. GEOVIEW SYSTEM

The presented system aims at 3D subsurface modeling to improve oil, gas, and mining explorations. Subsurface data are loaded into the system as 3D multi-layers. For each grid point on the x-y plane, depth values of top and bottom surfaces of each layer are registered in the z direction to form the geometric description of the subsurface layers to which layer attributes are attached. Efficient data structures and modeling tools are provided for visualising, analysing, and managing the 3D geological subsurface data. The system is developed in C programming language and based on a strong Graphics Library package which takes advantages of hardware of Silicon Graphics.

Two modelers coexist in the system: a surface modeler and an octree modeler. The surface modeler is used to convert the input data into a surface model according to the top-bottom layer-surface information; while the octree modeler transforms the same data into an octree model if necessary. Since the surface model and the octree model depict the same objects by using surface and solid geometric information respectively, the efficiency of modeling functions based on these two models is also different (Li 1994). For example, the surface model gives a relatively realistic shaded surface for visualisation because subtle normal vector changes of the surfaces can be represented. This is especially important when the lighting function is used. On the other hand, octants have six faces which, in turn, give only six normal vectors parallel to three principal axes. If the resolution of the octree model is set as the same as that of the input data, there is no loss of geometric information in the resulting octree model. However, the graphic quality of the octree model display is not comparative to the realistically shaded surface model because of the restriction of normal vector directions. Furthermore, layer-related topology can be constructed in surface models.

One of the major advantages of octree models is efficient Boolean operations because of the simple geometry and topology of octants. If encoded by Peano keys (Laurini and Thompson 1992), some spatial operations can be carried out at the bit level. In this system, octree models are, therefore, used to perform 3D spatial operations for analysis and simulations. Consequently, the system maintains two kinds of models for the same loaded object, namely the surface model for visualisation and the octree model for spatial operations.

Since most spatial operations are based on the octree representation the efficiency of octree operations often determines system responses to users requests. In special cases of geological subsurface modeling with large layer datasets, this is especially true. Among others, one of the critical and frequently used basic spatial operations is finding neighbour octants for a given octant. An application of this basic function in subsurface modeling could be to find all octants on the surface, for example, for a conversion from an octree to a surface model. Surface boundary lines can then be extracted from the boundary octants. The same basic function is also used in the operations to cut a subsurface model by a plane or a half cylinder face so that the intersection profile of the solid surface model is exposed for material queries and geological interpretations (Li and Xu, 1995). Figure 4 shows one of the examples in geological subsurface modeling. "Fences" are interactively defined on a 3D subsurface model. Spatial operations are required to perform the intersection between the "fence" (multi-planes) and the model. The front part of the model is then removed so that the defined profile can be visualised. In light of the above facts, it is necessary to develop an efficient algorithm for finding boundary octants in order to support quick system responses to users' octree-based requests.

5. ACKNOWLEDGMENTS

This project has been supported by the Geological Survey of Canada and the National Sciences and Engineering Research Council of Canada (NSERC).

6. REFERENCES

Arc/Info. 1992. ArcCAD GIS for AutoCAD Provides Full Complement of GIS Tools within AutoCAD Environment. ARC News Spring, 1992.

Abel, D.J. and D. Mark, 1990. A comparative Analysis of Some Two-dimensional Orderings, *International Journal of Geographical Information Systems (IJGIS)*, Vol. 4, No. 1, pp.21-31.

Arzty, K., G. Frieder and G.T. Herman, 1981. The Theory, Design, Implementation and Evaluation of a Three-dimensional Surface Detection Algorithm, *Computer Graphics and Image Processing*, 15, pp.1-24.

Fisher, T.R. and R.Q. Wales, 1992. Three Dimensional Solid Modeling of Geo-Objects Using Non-Uniform Rational B-Splines (NURBS), edited by A.K. Turner, Chapter 9 in *Three-Dimensional Modeling with Geoscientific Information Systems*, (Kluwer Academic Publishers), pp. 85-105.

Gargantini, I., 1982. Linear Octrees for Fast Processing of Three-dimensional Objects, *Computer Graphics and Image Processing*, 20, pp.365-374.

Jones, C.B., 1989. Data Structures for Three-dimensional Spatial Information Systems in Geology, *IJGIS*, Vol. 3, No. 1, pp.15-31.

Kavouras, M. and S.E. Masry, 1987. An Information System for Geosciences: Design Considerations, in *Proceedings of Auto Carto 8 held in Baltimore*, pp.60-65.

Laurini, R. and D. Thompson, 1992. Fundamentals of Spatial Information Systems. Academic Press, pp.507-511.

Li, R., 1994. Data Structures and Application Issues in 3-D Geographic Information Systems, *Geomatica*, Vol.48, No.3, pp.209-224.

Li, R. and C. Xu, 1995. An Algorithm for Searching Boundary Octants in 3D Geological Subsurface Modeling, *Geographic Information Sciences*, Vol. 1, No. 1, pp.23-32.

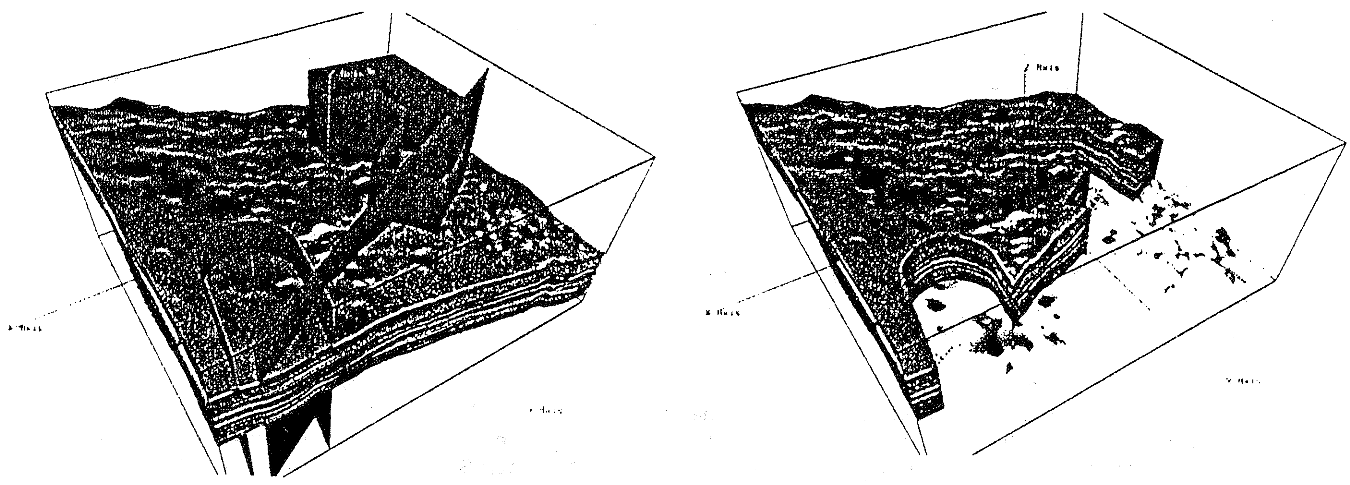
Liu, H.K., 1977. Two- and Three-dimensional Boundary Detection, *Computer Graphics and Image Processing*, pp.123-134.

Mason, D.C., M.A. O'Conaill and S.B.M. Bell, 1994. Handling Four-dimensional Geo-referenced Data in Environmental GIS, *IJGIS*, Vol. 8, No. 2, pp.191-215.

Meagher, D., 1982. Geometric Modeling Using Octree Encoding, *Computer Graphics and Image Processing*, **19**, pp.129-147.

Smith, D.R. and A.R. Paradis, 1989. Three-dimensional GIS for the Earth Sciences, edited by J.F. Raper, Chapter 11 in *Three Dimensional Applications in GIS*, (Bristol, PA: Taylor and Francis), pp.149-154.

Turner, A.K., 1989. The Role of Three-dimensional Geographic Information Systems in Subsurface Characterization for Hydrogeological Applications, edited by J.F. Raper, Chapter 8 in *Three Dimensional Applications in GIS*, (Bristol, PA: Taylor and Francis), pp.115-127.



(a) (b)

Figure 4 One of examples in geological subsurface modeling: Interactive definition of "fences" on a subsurface model (a) and removal of the front part of the model using spatial operations (b)