A PERSPECTIVE PROJECTION ALGORITHM WITH FAST EVALUATION OF VISIBILITY FOR DISCRETE THREE-DIMENSIONAL SCENES

H.OSWALD, W.KROPATSCH, F.LEBERL

TECHNICAL UNIVERSITY
AND
GRAZ RESEARCH CENTER
INSTITUTE FOR IMAGE PROCESSING AND COMPUTER GRAPHICS

Abstract

We have developed a three-dimensional display algorithm for discrete objects in a discrete space. Objects may be concave or convex, have holes (including interior holes) and can consist of disjoint parts. The input for our algorithm is a binary scene derived from a consecutive set of segmented and interpolated computed tomography (CT) cross-sections. The output is a shaded three-dimensional appearance of an organ from a user specified viewing direction.

Introduction

Certain spatial imaging problems represent the object space by a set of volume elements or voxels in a three dimensional array. With the advent of a number of three dimensional imaging technologies, three dimensional digital images have become quite common in many scientific fields. A very large and popular source of such three dimensional data arrays presents the area of computed tomography. For a human observer it is not easy to recognize features in this three-dimensional array, if they are presented only as numerical data in an array. Therefore there exist some techniques to represent three-dimensional data, which often can be interpreted as objects in a space, with shaded-surface display. Especially in the area of computed tomography, we want to display organs or parts of the skeleton with a shaded surface.

There are essentially two approaches to the shaded-surface display problem. The difference lies in the methods of representing objects in the three dimensional image. In the first approach, a set of contours is used to define the object regions in a sequence of slices. The boundary surfaces of the object are produced either by tiling contours in succesive slices with triangular patches or by fitting surface elements to the contours by the method of lofting using cardinal splines (1).

In the second approach, a set of volume elements represents an object in the three dimensional image. A boundary surface of the object is a set of faces of voxels, which are determinded by tracking through the connected set of faces comprising the boundary surface. The representation in this approach can handle objects of very complex shapes, which are commonly encountered in medical applications (2).

Our approach is closely related to the second, but with the essential difference of interpreting the cubie voxels as single points. A surface of an object in this case is formed by a set of points sampled in some appropriate sense.

Having represented a boundary, surface by this approach, the hidden surface removal and the shading algorithm can be greatly simplified. For the hidden surface algorithm the object boundary data are stored in the order of increasing coordinates. Thus a dynamic mode of display is possible, without changing the boundary data format. This feature makes it possible to keep the computations required in assign shading to an object to a minimum. Furthermore this display algorithm can also generate synthetic stereo-scopic views from an object scene. The algorithms are written in FORTRAN in a portable fashion and are implemented on a PDP 11/34 minicomputer.

1. The Object Space Model

We first describe our object space model or universe, which is similar to the cuberille model by Herman (3). The universe is defined as a discrete three dimensional image or scene, built up by cubic formed voxels of discretisation unit size. We assume that together with the scene a rectangular coordinate system has been determined which assigns to each voxel a triple (i,j,k) of integers. The origin of the universe is the point of intersection of the coordinate axis. An important feature of our model is that we use a discrete metric.

In many application areas the scene is digitized in the sense that the values must be integers in a finite interval, for example the CT-Scene has values in the range of - 1024 to + 1024. For our object space model we assume a binary scene, that means that the value set is (0,1).

2. Objects and Their Surface

For a human observer there is no difficulty to identify objects in three dimensional scenes. It is therefore surprising that it is far from trivial, to give a precise mathematical definition

CH1804-4/82/0000/0464$00.75©1982 IEEE

of an "object" and its "surface".

Our approach to define an object in the binary scene is that we interpret a voxel V(i,j,k)= 1 as a single point in the universe, disregarding its finite surface. An object is formed by a set of points, which are O(3)-adjacent (see Appendix A) and it is therefore a connected subset of voxels of the universe. Furthermore an object is always of the same order as the universe, in our case of order three it is composed of discrete units of the three-dimensional space. All objects in a third-order universe must occupy volume. A 2-D object could not exist there.

We now indicate an approach to define surfaces of objects in our universe. We need this surface for the shading algorithm in our display model.

The surface S of an Object O is a set of points V with the following properties:

$$(1)\quad S:=\{V(i,j,k) \mid V(i,j,k)=1 \text{ and } |V(i-1,j,k)-V(i+1,j,k)| + |V(i,j-1,k)-V(i,j+1,k)| + |V(i,j,k-1)-V(i,j,k+1)| \geq 1\}$$

$$(2)\quad |S| \geq 3 \quad \text{(numbers of connected points)}$$

In other words the surface of an object consists of points which are border points of the object and that means that the O(1)-adjacent to a single object point is not symmetric. To avoid singularities of the surface, the numbers of connected surfacepoints has to be not less than three. With these properties we have given an exact definition of a surface for our object in a binary scene.

3. Display of Object Surfaces

A further problem of the true display of an object is the calculation of the shading of the surface. We use a shading model according to the Lambertian law, which consists of diffuse illumination and diffuse reflection. In our opinion this model is satisfying in medical application for the display of organs of the CT-Scene.

The light energy E_p emanating from a surface point is the sum of the diffuse illumination E_{pd} and the diffuse reflection E_{ps} :

$$E_p = E_{pd} + E_{ps}$$

$$E_{pd} = R_p * I_d$$

$$E_{ps} = (R_p * \cos \varphi) * I_{ps}$$

In these equations, R_p is the reflectance coefficient at P, in the range of O to 1, I_d is the diffuse illumination falling on the entire scene, I_{ps} the energy arriving from a light source. The angle φ is the angle between the surface, normal N_p and a ray to the light source(4). To calculate the diffuse reflection, the normal vector in a surface point is used,but with our definition of the surface the normal vector is implicitly defined in the form of

$$N_p = \begin{bmatrix} n_i \\ n_j \\ n_k \end{bmatrix} = \begin{bmatrix} V(i-1,j,k) - V(i+1,j,k) \\ V(i,j-1,k) - V(i,j+1,k) \\ V(i,j,k-1) - V(i,j,k+1) \end{bmatrix}$$

This definition of the normal vector generates a sample of 26 distinct directions of a surface. As we can show in our examples, this discretisation of an object surface is satisfying for our problem. For a finer discretisation with a more smoothed surface, one can define the normal vector in a greater neighbourhood, than the first order neighbourhood. This problem in 3-D is similar to chain-coding in 2-D spaces (5).

To a surface point EL of our object O consists now of the six - tuple (i, j, k, n_i n_j, n_k). We calculate these values slice by slice from our binary scene, and get a linear list of surface elements EL, sorted by increasing coordinates. The fact that coordinates increase is important for our hidden surface algorithm.

4. Perspective Projection

The projection, we had implemented is the perspective projection because it corresponds to natural view and to camera imaging. But in general there is no restriction on projection of our object model.

The universe in our definition is a discrete three dimensional space, and also the image-space onto which we are projecting is a discrete, but two dimensional space. In order to get a "true" projection of our object we have to guarantee, that the projection of voxels, which are O(3)-adjacent, yields O(2)-adjacent pixels in the image space. To do this we are projecting the diameter of the object space unit by unit onto connceted units of the image plane. By projecting also the 8 corners of the universe to the discretizised image space, we can define our discrete image plane. It is very important, that the image plane has square formed discretisation units to provide a deformation of the object.

By observing these considerations, our projection of an object, sampled by points, will give a correct view of this object.

5. Display of Object Surfaces

In order to create a realistic image, we must apply a hidden-surface algorithm. One of the simplest is the depth-buffer algorithm (see Appendix B). Instead of calculating the depht of each surface-point, we use a sorting of the surface elements in the way of increasing coordinates and project it in the same order. Thus surface elements, which are closer to the observer will, during the process, cover those which are farther away. Therefore the resulting image shows only the visible surface of an object.

The sorting of the surface elements need not be changed, if the observer position moves in an octant. However if the direction of view changes from one octant to another you had to re-sort the surface elements.

6. Examples

For illustration of our results we have chosen a CT-Scene from the head. It consists of 10 slices, each slice with a thickness of 5 mm. (Fig. 2)

Our aim was to display the objects skull and lesion. The first step was the interpolation of thinner subslices, to get a CT-scene of cubic voxels. For this CT-Scene we used linear interpolation.

In the second step, we generated a binary scene by segmentation of the CT-Scene (6). This we did by thresholding over the range of values according to object's CT numbers (7). The binary scene consists of 160 x 160 x 46 elements and the surface of our object has about 76600 elements.

The three-dimensional displays show the skull and the lesion in shaded perspective views from different directions. (Fig.3, Fig.4)

7. Outlook and Conclusion

We have outlined an efficient algorithm for a realistic display of surfaces based on CT-scannes. In a next step we want to implement a more efficient data storage for the surface elements, namely the octree representation (8). This technique of octree encoded objects will allow us: to implement the shading feature transparency, to remove object parts and to intersect two or more objects to generate a single display of them.

8. Acknowledgements

The research reported in this paper is supported by the Fond zur Förderung der wissenschaftlichen Forschung. We are grateful to Mr. Gell for his many interesting comments and helpful discussion and for supplying the CT-data for our example. Thanks also to Mr. Ranzinger for his comments on an early draft.

Appendix A

Adjacency and Connectivity in 2 and 3-Dimensional Discrete Space (9).

A pixel or point p in a 2-dimensional space R^2 is a tuple of integers (p_1, p_2). Two points p and p' are

(a) 1-adjacent if and only if

(i) $0 \leq |p_i - p_i'| \leq 1, \quad i=1,2$

(ii) $\sum_{i=1}^{2} |p_i - p_i'| = 1$

and (b) 2-adjacent if and only if

(i) $0_2 \leq |p_i - p_i'| \leq 1, \quad i=1,2$

(ii) $\sum |p_i - p_i'| = 2$

Two points are O(1)-adjacent if they are 1-adjacent and they are O(2)-adjacent if they are 1-adjacent or 2-adjacent.

For the 3-dimensional space we can define the n-adjacent and O(n)-adjacent for two voxels or points in R^3 similar to the 2-dimensional case. The geometric appearance of this situations is shown in Fig. 1.

Appendix B

The depth-buffer algorithm given below requires two arrays for intensity and depth, each of which is indexed by pixel coordinates (x,y).

```
1   For all pixels on the screen, set DEPTH (x,y)
    to a maximum value and INTENSITY (y,x) to a
    background value
2   for all surface elements EL(i,j,k,n_i,n_j,n_k) do
3   begin
4   calculate perspective projection D3(i,j,k)→
    D2(x,y)
5   calculate the depth d of the surface element EL
6   if d  depth (x,y) do
7   begin
8   depth (x,y) = d
9   calculate shading  s of surface element EL
10  intensity (x,y) = s
11  end
12  end
```

The modified depth-buffer algorithm needs only one array for the intensity of the resulting image.

```
1   For all pixels on the screen, set INTENSITY
    (x,y) to a background value
2   for all surface elements EL(i,j,k,n_i,n_j,n_k)  do
3   begin
4   calculate perspective projection D3(i,j,k)→
    D2 (x,y)
5   calculate shading s of surface element EL
6   INTENSITY (x,y) = s
7   end
```

References

(1) G.T.Herman et al, Three Dimensional Display of Medical Objects, Technical Report MIPG53, Medical Image Processing Group, 1981, State University of New York at Buffalo.

(2) G.T.Herman, Three-Dimensional Imaging from Tomograms, Digital Image Processing in Medicine, Proceedings, Hamburg, October 1981, Springer

(3) J.K. Upuda, Interactive Segmentation and Boundary Surface Formation of 3-D Digital Images, Computer Graphics and Image Processing, p.213-235, Vol 18, 1982

(4) W.H. Newmann and R.F. Sproull, Principles of Interactive Compter Graphics, International Student Edition, 1979

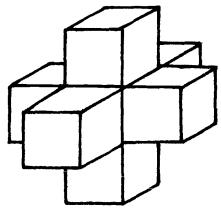
(5) H. Freeman and J.A. Saghri, Comperative Analysis of Line-Drawing Modeling Schemes, Computer Graphics and Image Processing, p.203-223, Vol 12, 1980

(6) H. Oswald, Merkmalserkennung in Dreidimensionalen Szenen der Computer Tomographie, p.219-233, ÖAGM-Tagung 1981, OCG-Proceedings

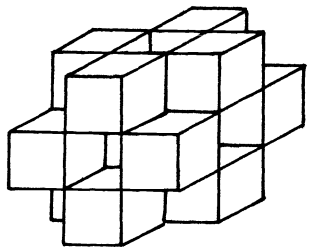
(7) H. Oswald, Segmentieren von Dreidimensionalen Computertomographischen Szenen und räumliche Darstellung der Ergebnisse, DIBAG-Publication Nr. 3, 1981, FZG, Graz, AUSTRIA.

(8) D. Meagher, Geometrie Modeling Using Octree Encoding, Computer Graphics and Image Processing, p.129-147, Vol19, 1982

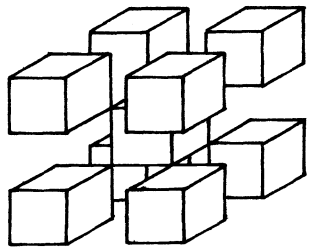
(9) J.K. Upuda et al, Boundary Detection in Multidimensions, IEEE Trans.on Pattern Anal. and Machine Intell., PAM-4, Nr.1, January 1982.



a)



b)



c)

Figure 1. The geometric appearance of the three possible adjacencies in R^3: a) 1-adjacent, b) 2-adjacent and c) 3-adjacent.

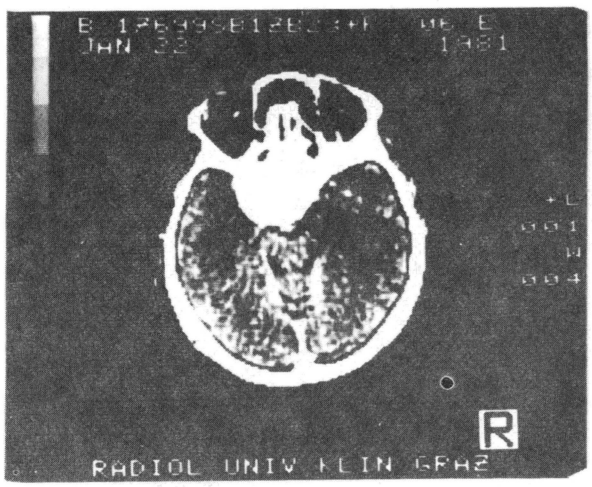


Figure 2. Slice 30 out of the 46 1.0 mm thick slices of a head scann. This slice is produced by an EMI CT1010 scanner.

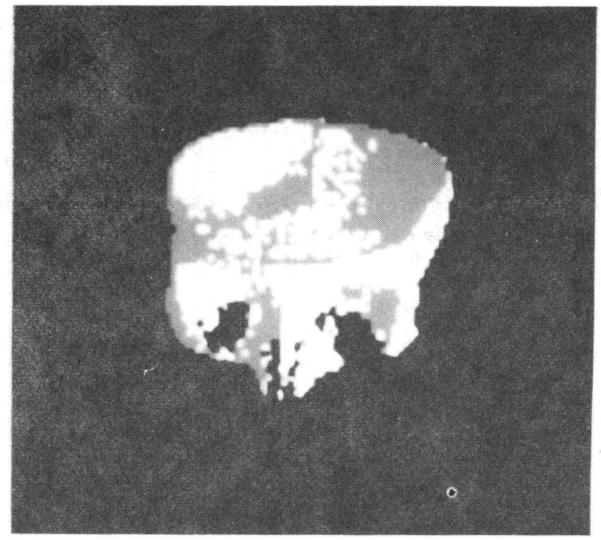


Figure 4. A single view of the detected surface of the skull and lesion. This picture is zoomed with the factor 2 by linear interpolation.

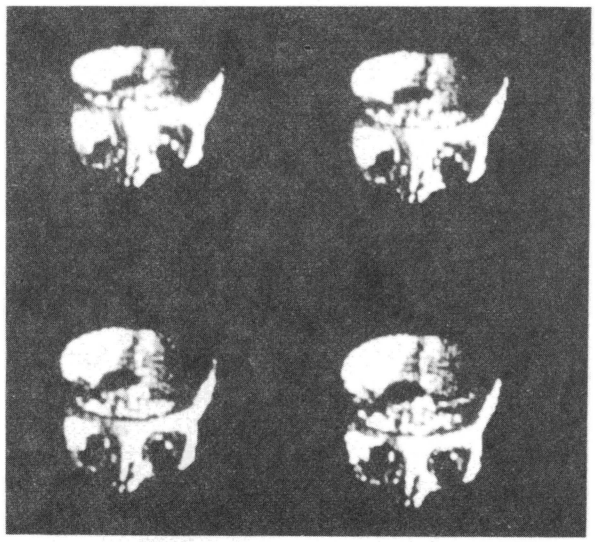


Figure 3. Four views of the detected surface of the skull and lesion as displayed by our program.