RBE. VOL.6 N.2 1989

OBJECT HANDLING AND DISPLAY IN THE VOXEL SPACE.

LINCOLN MOURA[1] and RICHARD KITNEY[2]

ABSTRACT-- A considerable amount of work has been undertaken towards viewing the inner parts of the human body non-invasively. In this regard, Computerised Tomographic Scanners and Nuclear Magnetic Ressonance Systems allow the acquisition of very accurate anatomical information. Although this information is of three-dimensional nature, it has been presented, in practice, as a series of cross-sectional images taken at varying depths. Similar data is generated by serial slicing in Pathology, though in this case images are not necessarily physically aligned in relation to each other.

Owing to the increasing processing power of digital computers and graphic peripherals, a major interest has developed in the reconstruction of internal body structures so as to create *digital objects* which can be handled and displayed using techniques developed in Computer Graphics.

In this paper we present a system for handling and display of 3D medical structures which can handle up to 8 objects simultaneously. We also discuss future steps towards overcoming the system's major drawback: *processing time.*

INTRODUCTION.

In this paper the development of a system for handling and display of digital objects in the Voxel Space is discussed and a computer system for this task is presented. The aim of the system is to provide the user with a tool which allows the handling and display of computer-defined Voxel Space objects in an interactive and natural way (Figure 1).

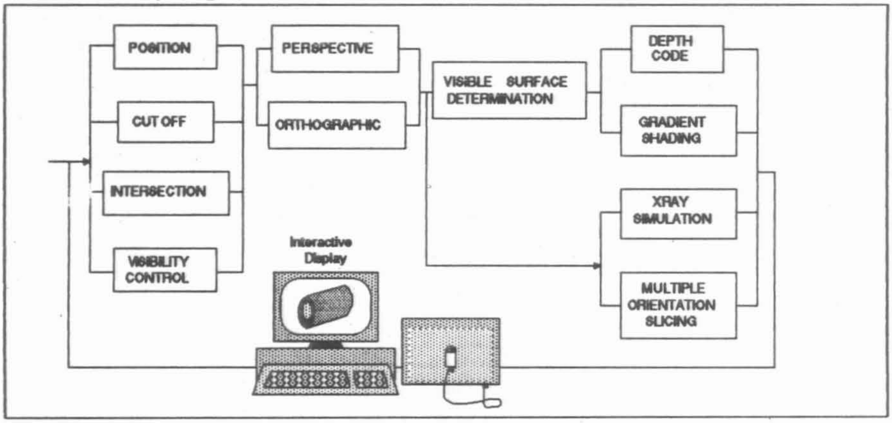

Figure 1: Block Diagram for the Object Handling and Display Module.

---

[1] Coordenadoria de Informática - InCor HCFMUSP, São Paulo, SP.

[2] Biomedical Systems Group - Imperial College, London SW7 2AZ, U.K.

Such a system has been implemented on a MicroVax-II computer and a Sigmex hi-res graphics terminal with resolution of 1448 x 1024 pixels in 256 colours.

Although the final objects to be handled and displayed are to be reconstructed from medical image data, the digital objects used in this paper have been created using either hand-drawn contours or geometric primitives. The aim of this paper is to approach object handling and display regardless of their origin.

VOXEL SPACE OBJECT DESCRIPTION.

In the Voxel Space an object is always described as a collection of voxels. Although there are three major ways in which the collection of voxels can be stored — Voxel Enumeration, Octrees and the Random-Access Voxel Space Database, we will discuss only the latter because of lack of space. The reader can refer to Moura (1988), for a comparison between these methods.

Random Access Voxel Database.

In this method a three-dimensional cubic matrix V of dimension N_v^3 has each of its elements $V(i,j,k)$ associated with each voxel on a one-to-one basis. A voxel at position (i,j,k) is then considered to be part of an object if $V(i,j,k)=1$ or not part of the object otherwise. More generally, it is possible to associate several objects with the Voxel Space, by associating different values of $V(i,j,k)$ with different objects, as in Equation (1) below

$$O_n = \{ v \,/\, V(v) = K_n,\ v \in VS \}, \tag{1}$$

where O_n denotes the n-th Object and K_n denotes the voxel value associated with the n-th Object.

In general, any operation on the Random Access Database requires that the N_v^3 voxels be scanned, independently of the model's complexity. This is a major advantage if models are expected to be complex. The dimension N_v of the Voxel Space defines the model complexity as it defines the Voxel Space resolution.

The major advantage of the Random Access Database is that given a voxel it is possible to access directly its associated voxel value. Consequently, given a voxel it is possible to determine immediately which object, if any, it belongs to. Likewise, if a voxel is to be excluded from an object and included into another, the only operation required is to set the proper element $V(i,j,k)$ to the voxel value associated with the new object.

Although all the methods of representing objects in the Voxel Space show advantages and disadvantages, the Random Access Database is the only one which is fully compatible with digital images, in the sense that the three-dimensional matrix $V(i,j,k)$ — which is the essence of the Random Access Database — can be used to store the pixel values of a sequence of digital images. Furthermore, even though the Random Access Database is the method which is more likely to require the largest amount of memory for object representation, it is the only method in which it is possible to move equally from object to voxel and from voxel to object.

These major advantages led us to decide to use the Random Access Voxel Database for object representation in our computer system.

OBJECT DISPLAY AND GEOMETRIC PROJECTIONS

Man-made devices for optical image capture use a planar light-sensitive surface as exemplified in Figure 2a This model — called *perspective projection* — is assumed to be the golden standard for distortion-free optical systems. In this model parallel straight lines may come out oblique because light-rays cross at the observer, as in the figure. As the observer moves away from the object this effect is less noticeable. Indeed, if the distance object-observer tends to increase indefinitely or if the distance object-observer is much greater than the object dimensions, then the light-rays can be assumed to be parallel and the model becomes an *orthographic projection*, as shown in Figure 2b. In orthographic projection parallelism is always retained.

Although perspective projection has been extensively used in CAD/CAM applications — Amanatides (1987), Faux (1985), Foley (1984) — its use in systems specifically designed for medical applications has not been reported in the literature. Perspective projection in the Voxel Space is not particularly difficult to implement though it is more time-consuming than orthographic projection because the latter can assume some simplifications - Moura (1988).

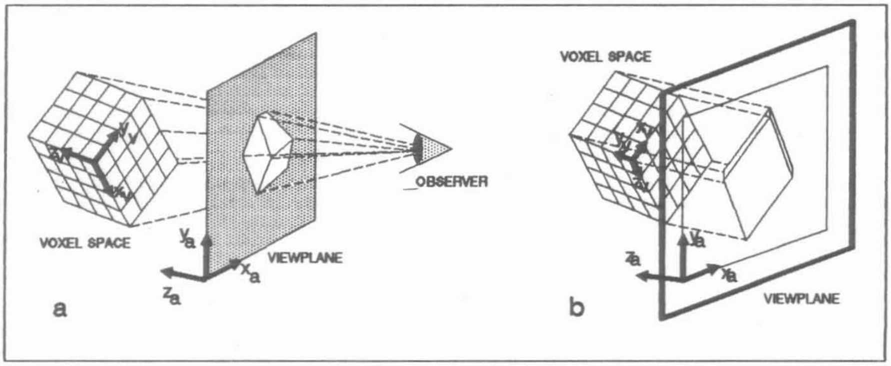

*Figure 2: **Geometric Projections.** As the observer moves away from the object the projection lines tend to become parallel to each other and orthogonal to the viewplane. The projection goes from perspective (a) to orthographic (b).*

Orthographic projection is a good model for medical systems because medical volumes tend to be small and are not handled from a much close distance — Herman (1983). The advantages brought about by the use of perspective projection in medical systems are not worth the additional cost in processing time and do not justify its use. The orthographic projection of the Voxel Space onto the absolute xy-plane can be easily computed using mathematics decribed elsewhere — Moura (1988).

IMAGE RENDERING

Although the geometry of image formation can be easily modelled, modelling the whole process of image formation to create realistic views of computer defined objects is a much more complex process. The techniques used for creating such views are called *image rendering techniques*.

Image rendering techniques consist basically of determining the surface seen by the observer and assigning proper colours and their shades to it. The quality of the rendered image is dictated by the method used for assigning the colour shades.

Whenever an opaque object is to render a shaded image, the visible surface must be determined. In general, determining the visible surface consists of locating the parts of the object surface which are closest to the observer along the projection lines.

The visible surface problem is traditionally solved by determining the voxel facets which are closest to the observer for every projection line. There are

One of the first simplifications which can be made to the model, is to suppress shadow formation. The easiest way to avoid shadows is to assume that there is some ambient light and only one point light source, centred at the observer's eye. This assumption brings a tremenduous reduction in model complexity yet retaining the basic image rendering features. Indeed, although shadows tend to add realism to images, they often mask important information in the picture. That is the reason for fitting hospital surgery theatres with shadow-free lighting.

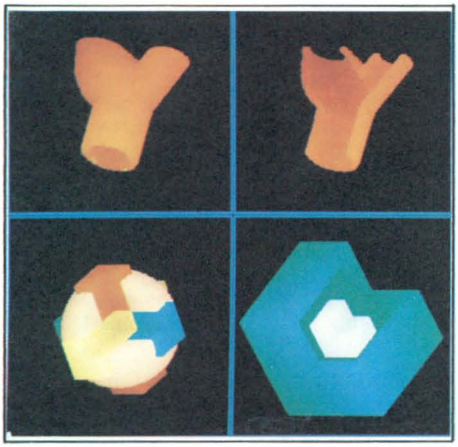

Figure 3
Four examples of Depth Code.
These images were produced using Depth Code.

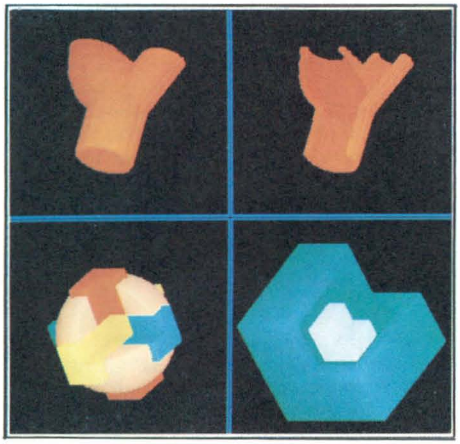

Figure 4
Four examples using the Empirical Diffuse Reflection Model.
This model produces more natural and informative images.

Gordon and Reynolds (1985) showed that instead of processing each voxel and its neighbours in search of the normal vector to the surface, it is far more effective to process the z-buffer image and use the associated z coordinates to compute the surface normal vector. This approach has been referred to as *image-based gradient shading* and it leads to image rendering in a much shorter time without major losses in image quality - Chen (1985).

basically two algorithms which have been used to determine the visible surface. The first — called *back-to-front* — consists of sorting all the facets in decreasing order of z-axis coordinate. Each facet is then projected onto the viewplane so that the closest facets will prevail. The second algorithm — called *z-buffer* — starts by setting the pixel values on the viewplane to a number D_{max} larger than the maximum possible z-coordinate. A voxel facet is then projected onto the viewplane leading to a pixel p. The distance between the facet and the viewplane is stored on pixel p if it is less than the value already stored on this pixel. After all the facets have been processed the pixels on the viewplane contain the visible surface defined by the z-coordinates.

It is possible to take advantage of some properties of the Voxel Space environment in order to make this computation run faster. Given any Voxel Space position, all the voxels have the same orientation, i.e. one voxel facet is parallel to all the other corresponding voxel facets. Also, for each voxel, at most three facets will be visible and the shape formed by the orthographic projection of each voxel onto the xy-plane is the same for the entire Voxel Space.

Depth Code

A very simple image rendering technique which has been widely used is Depth Code. This method is in no way supported by any law of Optics though it can be easily implemented and leads to satisfactory results, as testified by the number of systems described in the literature that have used this technique — Bresina (1986), Laschinger (1987), Trivedi (1987).

In Depth Code, the intensity of a pixel on the viewplane is a function of the distance between the viewplane and the point at the object surface which has been mapped onto the pixel. In fact, the closer a voxel is to the viewplane the brighter the pixel onto which the voxel is projected is assumed to be.

Therefore, the Depth Code image can be easily obtained from the visible surface algorithm previously described by replacing each pixel value with the constant D_{max} minus the pixel value — Moura (1988).

As can be seen in Figure 3, the results obtained when using Depth Code are, in fact, surprisingly good considering the simplicity of the model and its lack of theoretical support. However, Depth Code has its limitations. In some circumstances Depth Code tends to generate too smooth an image where rougher ones are expected. As an example, in the image in Figure 3c the edges of the cube are fainter than is appropriate. In fact, the edges are recognised more by the overall information present in the image, rather than the edges themselves.

If more realistic images are to be generated then it is necessary to search for a more sophisticated model.

Shading Models

Complex scenes are composed of several light sources, many objects and different types of surfaces. Reflecting surfaces can act as secondary light sources creating an endless interaction between surfaces. Complex patterns of shadows and textures can also be present. A realistic model for computer shading will take into account most of these factors and translate them into a *shading rule* which can be applied to the computer model, or more precisely to the visible surface.

Although some areas of Computer Graphics, such as Film-Making — Reynolds (1982), and Flight Simulation — Gardner (1984), Greene (1984), are mainly concerned with representing complex scenes as realistically as possible, medical systems are not intended to render images able to "fool the eye". Instead, medical systems must generate simple images which convey unmistakable information about the medical structure's geometry.

The shading model we have decided to use is described in detail by Moura (1988). It is an empirical model for diffuse reflection, assuming one point light-source placed at the observer, both infinitely distant from the model. The corresponding shading rule can be computed from the z-buffer image as:

$$G(x,y) = \frac{K_1}{Z(x,y)+K_2} \cos^u \theta, \qquad (2)$$

where K_1 and K_2 are constants related to environmental lighting, $u = 0.2$ is a constant, $Z(x,y)$ is the z-buffer distance and θ is the angle between the surface normal vector and the light rays at the visible surface point which maps onto (x,y).

Some results using this model can be seen in Figure 4. Note that images are natural and convey more information than those in Figure 3.

OBJECT HANDLING AND OTHER FEATURES

Apart from displaying the gradient shading image, the system allows that the model be handled and viewed in other ways. It is possible to create x-ray simulations, fast shading for positioning as well as object sectioning on any plane.

Object handling also includes intersecting the model with given objects such as blocks, cylinders and spheres. It is also possible to make measurements on the model so as to compute volumes, distances or surface areas.

For a complete system description, the reader should refer to Moura (1988).

CONCLUSIONS

In this paper we have presented some aspects of the Voxel Space computer system we have developed for handling and display of 3D models. Once defined the models can be rotated, intersected by primitives or used to generate cut-offs. The resulting 3D model can be viewed in a number of representations, such as Depth Code and Gradient Shading.

This system has been successfully applied to the handling and display of arteries modelled using ultrasound, as presented by Moura and Kitney (1988).

The most important drawback in the present system is posed by processing time and Voxel Space resolution. These two factors are interrelated. Broadly speaking, the 3D processing time is proportional to the cube of the Voxel Space size — N_v^3 — and the 2D processing time is proportional to N_v^2.

In fact, rather than computer memory, the major difficulty is posed by computer speed. About 20 seconds are required to produce a typical Gradient Shading image. This might not be a major problem if the system were not intended for medical use. In similar applications, however, physicians have required that images be displayed with little or no delay [7.08]. The requirement for speed is even greater if the system is to be used in real time applications.

Although the programs in the present system have not been fully optimised for speed, it is estimated that the processing time cannot be drastically reduced unless the hardware is upgraded. There are basically two ways in which this can be achieved. The first is by using very fast computers — called *array processors* — which can be plugged in or connected to a host computer.

Array processors do not operate on their own. They usually input and output data only to a host computer. However, once the data has been moved into the array processor, processing is done at very high speed, measured in terms of million of arithmetic operations per second — MIPS if the operations are integer or MFLOPS if the operations are floating point. Currently available array processors range from some MIPS to hundreds of MFLOPS depending on their cost.

The second alternative is the use of *transputers*. These devices are computers with a remarkable property. They can be connected to each other so that they operate in parallel — Stein (1988). Therefore, if enough transputers are interconnected together the Voxel Space can be scanned very quickly.

It seems that for mid and long term results the best approach is given by the use of personal computers and transputers. Personal computers have had their performances so much improved that they may compare with yesterday's mainframes. The state-of-the-art personal computer can execute more than 30 million operations per second and can address some tens Megabytes of memory. Furthermore, these systems can make use of a plethora of add-ons which include array-processors and transputers as well as high-resolution colour graphics cards.

On the other hand, transputers are truly modular and can be expanded indefinitely without major losses in performance. The use of microcomputers and transputers in the handling and display of medical objects is very promising and should be placed at the top of the agenda.

ACKNOWLEDGEMENTS

This work was partly supported by Fapesp, CNPq and InCor.

REFERENCES

AMANATIDES J. *Realism In Computer Graphics: A Survey.* Computer Graphics and Applications, 1987, 1, 44-56.

BRESINA SJ et al. *3-D Animation Of Human Wrist Motion.* NCGA Conference Proceedings, 1986, (III), 5, 124-131.

CHEN LS et al. *Surface Shading In The Cuberille Environment.* Computer Graphics and Applications, 1985, (5), 12, 33-42.

FAUX ID and PRATT MJ. *Computational Geometry For Design And Manufacture.* Ellis Horwood Ltd - John Wiley and Sons, 1985, 331.

FOLEY JD and van DAM A. *Fundamentals Of Computer Graphics.* Addison-Wesley, 1984, 664 pp.

GARDNER GY. *Simulation Of Natural Scenes Using Textured Quadratic Surfaces.* Computer Graphics, 1984, (18), 7, 11-20.

GORDON D and REYNOLDS RA. *Image Space Shading Of Three-Dimensional Objects.* Computer Vision Graphics and Image Processing. 1985, (29), 361-376.

GRAY JE et al. *A Skeptical Look At A (Digital) Sacred Cow.* SPIE Medical Imaging and Instrumentation, 1984, (486), 160-166.

GREENE N. *A Method Of Modelling Sky For Computer Animation.* Proceedings Intern Conf Engineering And Computer Graphics, 1984, 8, 297-300.

HERMAN GT and UDUPA JK. *Display Of 3D Digital Images: Computational Foundations And Medical Applications.* Computer Graphics and Applications, 1983, 8, 39-45.

LASCHINGER J et al. *Three Dimensional Reconstruction Of The Heart Using EKG-Gated Magnetic Resonance Images.* NCGA Conference Proceedings, 1987, (III), 3, 87-91.

MOURA L and KITNEY R. *Three Dimensional Solid Modelling Of Arterial Structures Using Ultrasound.* Proceedings IEEE IX Annual Conf Engineering Med Biology Soc, 1987, 11, 400-401.

MOURA L. *A System For Reconstruction, Handling And Display Of 3D Medical Structures.* Ph D Thesis, University of London, 306 pps.

REYNOLDS C. *Computer Animation With Script And Actors.* Computer Graphics, 1982, (18), 7, 289-296.

STEIN RM. *T800 And Counting.* Byte, Nov, 1988, 287-296.

TRIVEDI et al. *Cardiac Applications Of 3-D Display And Analysis.* NCGA Conference Proceedings, 1987, (III), 3, 129-137.