

**3D DISPLAY OF ORGANS
USING CT DATA IMPLEMENTATION
ON AN IBM PC**

by
Mutlu Hüner
B.S. in E.E. Boğaziçi University, 1985

Submitted to the Institute of Biomedical Engineering
in partial fulfillment of the requirements
the degree of
Master of Science
in
Biomedical Engineering

Boğaziçi University
1987

Bogazici University Library



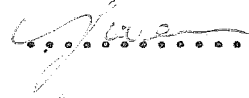
39001100312985

14

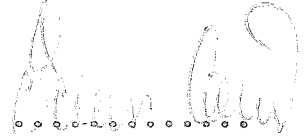
3D DISPLAY OF ORGANS
USING CT DATA-IMPLEMENTATION
ON AN IBM PC

APPROVED BY

Yard.Doç.Dr.Albert Güveniş
(Thesis Supervisor)



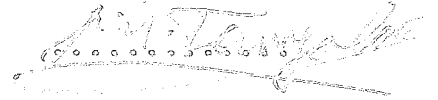
Doç.Dr.Ömer Cerid



Doç.Dr.Yusuf Tan



Prof.Dr.Necmi Tanyolaç



DATE OF APPROVAL 25.6.1987

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Dr. Albert Güveniř for his advice and encouragement throughout the preparation of this thesis. I am grateful to Dr. Yusuf Tan for his support at various stages of my work. I would like to thank Dr. Ömer Cerid and Ahmet Ulubilgen for designing the graphics terminal and to Cüneyt Gemiciođlu for the hardware connection between the terminal and IBM PC. I would also like to thank Dr. Necmi Tanyolaç for his support during my studies in the Biomedical Engineering Institute. Finally, I wish to thank Engin Atik, Fatih Çolgar, and Levent Akın for their help and advice in using Pascal.

ABSTRACT

Three dimensional display of organs
using CT data-Implementation
on an IBM PC

AUTHOR: MUTLU HÜNER

Advisor: Dr.Albert Güveniş

This thesis presents a low cost software package for the three dimensional display of organs using an IBM microcomputer. First existing 3D display techniques and their application in medicine are given. Then the particular algorithm selected for implementation is described and explanations pertinent to this specific implementation are discussed. The boundary detection algorithm used is the one developed by G.T. Herman et.al.

Standard computer graphics methods are used for displaying 3D objects. A grid technique is introduced when performing the scan conversion in order to increase image quality. Some of the results obtained by using mathematical phantoms are presented and discussed. The results indicate that this package can display high resolution images without antialiasing effect that is typically encountered in similar packages. It is shown that low pass filtering is only needed in extreme cases.

ÖZET

Bu çalışmada IBM kişisel bilgisayarı kullanılarak organların üç boyutlu görüntülenmesi için ucuz bir sistem geliştirilmiştir. Önce, üç boyutlu görüntüleme teknikleri ve tıptaki uygulamaları incelenmiştir. G.I.Herman tarafından geliştirilmiş sınır sezim algoritması kullanılmıştır.

Üç boyutlu cisimlerin görüntülenmesinde standart bilgisayar grafik yöntemleri kullanılmıştır. Görüntü kalitesini arttırmak amacı ile kullanılan tarama çevrimi, örgü tekniği ile gerçekleştirilmiştir. Matematiksel fantomlar kullanarak elde edilen bazı sonuçlar sunulmuş ve tartışılmıştır.

Sonuçlar, benzeri paketlerde rastlanan örtüşmenin engellendiği, yüksek ayırma güçlü görüntülerin bu paketle elde edilebileceğini göstermiştir. Alçak geçirgen süzgeçe ancak aşırı durumlarda gerek duyulduğu gösterilmiştir.

TABLE OF CONTENTS

	PAGE
ACKNOWLEDGEMENTS	
ABSTRACT	
ÖZET	
LIST OF FIGURES	
LIST OF TABLES	
I. INTRODUCTION.....	1
1.1 Objective of this thesis	3
1.2 Description of the 3D display method and equipment used in this thesis	5
1.3 Plan of thesis	6
II. STATE OF THE ART	7
2.1. State of the art	7
A. Reorganization	7
1. Reslicing	7
2. Reprojection	8
B. Surface display	8
1. 1D unit based approach	8
2. 2D unit based approach	9
a) Slice by slice	9
b) True 3D	15
3. 3D unit based approach	17
a) octree encoding	17
b) back to front display method	22
2.2 Medical applications of 3D imaging	25

III. TERMINOLOGY	34
3.1. General terminology for 3D	34
3.2. Graph theoretical terminology	35
3.3. Definitions and theorems for the boundary detection algorithm	40
A. Two dimensional case	40
B. Three dimensional case	44
IV. BOUNDARY DETECTION	49
4.1 An intuitive description of the boundary	49
4.2 Two dimensional boundary detection	53
4.3 Three dimensional boundary detection	54
V. HIDDEN SURFACE REMOVAL	56
VI. GEOMETRICAL TRANSFORMATIONS	59
6.1 Translation	59
6.2 Scaling	60
6.3 Rotation	60
VII. PROJECTIONS	64
7.1. Parallel and perspective projections	64
7.2 Orthographic and axonometric projections	66
VIII. SHADING	68
8.1 Diffuse and specular reflections	68
8.2 The shading rule used in this thesis	70
8.3 Shading using CT numbers	70
8.4 Smoothing	72
IX. IMPLEMENTATION OF THE 3D DISPLAY ALGORITHM	74
9.1 Hardware environment	74
9.2 Software environment	74
9.3 Implementation	74
9.3.1 User Interaction and I/O	74
9.3.2 Program Modules	75

X. RESULTS AND DISCUSSION	83
10.1 Illustrative results	83
10.2 Discussion	84
10.3 Future work	86
Bibliography	93

LIST OF FIGURES

	PAGE
FIG.1.1 A slice through the body is imaged by CT	2
FIG.1.2 Representation of a CT slice image	3
FIG.1.3 The arbitrarily rotated stack of arrays are projected onto the screen in order to obtain a 3D visualization	4
FIG.2.1.1 Illustration of the concepts used in triangulation techniques	9
FIG.2.1.1 Object representation by directed contours. Object subregions lie to the left of the contours	14
FIG.2.1.3 Definition of direction Codes for an 8-chain	14
FIG.2.1.4 Illustration of 8 contours	15
FIG.2.1.5 Simple object represented in octree encoding format Numbering sequence, label definitions, three level representation of an object.	18
FIG.2.1.6 Linear octree representation of an object	20
Fig.2.1.7 Back-to front voxel readout in two dimensions. For the purpose of illustration, assume the origin is at the corner farthest from the observer. The voxels are traversed in order of increasing values of x and y; either x or y may be chosen as the faster running index.	23

- Fig.2.1.8 Slice-by-slice back-to-front voxel readout. 23
 Slices are read out starting with the slice farthest from the observer. Within each slice, voxels are read out starting with the corner farthest from the observer. The voxels are labeled with three digits, identifying the x,y, and z coordinates in object space. For the object orientations shown the voxels can be read out in several slice-by-slice back-to-front sequences including 000,100,200...010,110,210...133, 233,333.
- FIG.3.2.1 A simple graph G 35
- FIG.3.2.2 A general graph 36
- FIG.3.2.3 Jordan Curve and closed Jordan Curve 36
- FIG.3.2.4 $v w x y z z y w$ is an edge sequence 37
 from v to w
- FIG.3.2.5 a digraph whose arcs are (u,v) (v,v) (v,w) 38
 (v,w) (w,u) and (w,z) . The ordering of the vertices in an arc is indicated by an arrow.
- FIG.3.2.6.a A digraph D 38
- FIG.3.2.6.b Underlying graph of D 38
- FIG.3.3.1 Four pixels in an two dimensional picture 40
- FIG.3.3.2 v is face adjacent to v' ($v v'$) 45
 v is edge adjacent to v'' ($v v''$)
- FIG.3.3.3 The two possible edges of face (b,w) are shown. 46
- FIG.3.3.4 The predicate describing (V,V') is $f_{2,-1}$ 47
- FIG.3.3.5 The pair (v,v') is a face of the voxel v . The 47
 pair (v,v'') is the right edge and the pair (v,v''') is the left edge of the face (v,v') .

FIG.3.3.6 Four voxels.	48
Fig4.1.1a A voxel with faces marked by numbers 1-6. There are also 12 arrows: the outgoing edges are the ones on which arrows terminate, the incoming edges are those on which arrows originate. Every face has two outgoing and two incoming edges, and every edge is the outgoing edge of one face and the incoming edge of another face. b) The digraph associated with. a) The correspond to the faces of the voxels. The continuous arrows corresponds to left adjacency, the broken arrows corresponds to right adjacency.	50
Fig.4.1.2 An organ of three voxels	50
Fig.4.1.3 The digraph of the organ in fig.4.1.2	50
FIG.6.1.1 point (1,2,0) translated to (7,4,3)	59
FIG.6.3.1 Point P rotated O_z around the origin to position p.	60
FIG.6.3.2 rotation matrix	62
FIG.7.1.1 parallel and perspective projections	65
FIG.7.2.1 orthographic projection	65
FIG.8.1.1 Direction of source of light, the normal of the surface. Diffuse and specular reflections	68
FIG.8.3.1 Voxels of a bone surface	71
FIG.8.4.1 The eight neighbors of G_1 used in low pass filtering	72

FIG.9.1.1	Flow diagram of the 3D display system	76
FIG.9.3.1	Numbering of faces	79
FIG.9.3.2	Offset values	80
FIG.9.3.3.	A 5x5 grid representing a rotated face	82
FIG.10.1.1	Rectangular mathematical phantom	90
FIG.10.1.2	Object rotated	90
FIG.10.1.3	Object rotated	91
FIG.10.1.4	Image without using grids. Jagged edges and holes due to antialiasing are seen.	91
FIG.10.1.5	Image obtained by using grid and lowpass filter	92

LIST OF TABLES

	PAGE
TABLE 2.1.1- The possible types of edge voxel a voxel can be in an 8-contour	16
TABLE 2.1.2- Labeling octants	21
TABLE 2.1.3- Priority order for octant labeling according to $VPN=(x,y,z)$	21
TABLE 4.1.1- Step by step results of the boundary detection algorithm for the organ in fig.4.1.2	52

I. INTRODUCTION

One of man's greatest dreams has always been the non-invasive visualization of body organs for diagnosing various diseases. The invention of X-Rays at the end of the nineteenth century has started an era in this direction. Conventional X-Ray imaging has been used extensively during our century for the evaluation of anomalies. With the rapid advance in computer technology a number of new imaging modalities have emerged in the past two decades. The invention of Computed Tomography in 1971 has revolutionized the world of radiology. Similar new techniques like Magnetic Resonance Imaging and Positron Emission Tomography make it now possible to obtain high contrast and high resolution images of a body organ in a very different way than conventional X-Ray Imaging.

Tomographic modalities allow us to select a plane through the organ of interest and visualize normal and abnormal structures without the annoying effect created by superposition of underlying objects. CT and MRI are gaining more popularity in detecting anomalies in various parts of the body in a non-invasive way. The high spatial and contrast resolution obtained by this technology has made it the tool of preference in diagnostic radiology in many cases.

The tomographic modality in general is used to depict the image of an organ along any desired plane. (See fig.1.1)

By examining several contiguous slices, one can determine exactly the location of a lesion or a tumor in a three dimensional environment.

Lately, there has been an interest in "displaying" an organ on a CRT screen in a truly three dimensional way.

This new technology which developed as a result of a cooperative effort by computer graphics specialists, biomedical engineers, and medical doctors consists of processing the CT or MRI data and projecting a three dimensional view of the organ onto the screen. One can then "see" the organ from any angle as if the organ has been completely removed from the body.

3D display of organs has started a number of important medical applications. For example, one can now plan surgery and rehearse it on the computer. This way exact surgery strategies are planned, customized prothesis are designed and a polyethylene model of the organ is manufactured using a milling machine which is directly connected to the graphics computer. Another important application of 3D imaging is to present the CT data to the radiologist in a form which would make the diagnosis easier. A number of patients have already benefited from this new technology which is still evolving rapidly in terms of both technique and application.

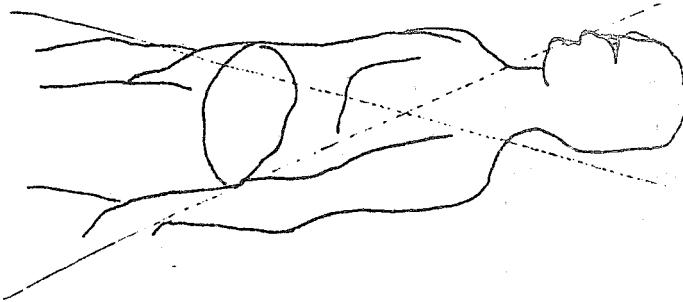


Fig.1.1 A slice through the body is imaged by CT

1.1 OBJECTIVE OF THIS THESIS

In CT, two dimensional reconstruction techniques are used to produce a slice image. Each 2D digital image is essentially made up of a 2D array of rectangular volume elements called voxels. (see fig.1.2)

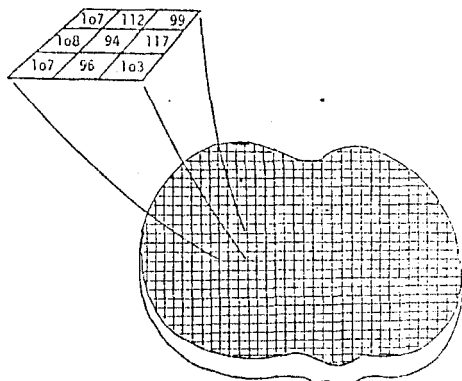


Fig.1.2 Representation of a CT slice image

A CT number is associated with each voxel, which is directly proportional to the relative linear attenuation coefficient of the organ at that location. Conventional (2D) CT display consists of displaying each slice by making the gray values proportional to the CT number.

The objective of 3D display is to build up a three dimensional array of numbers which contain spatial information rather than cross-sectional. By making use of these 3D array of densities, the method will display a particular organ as if it were removed from the body (see fig.1.3).

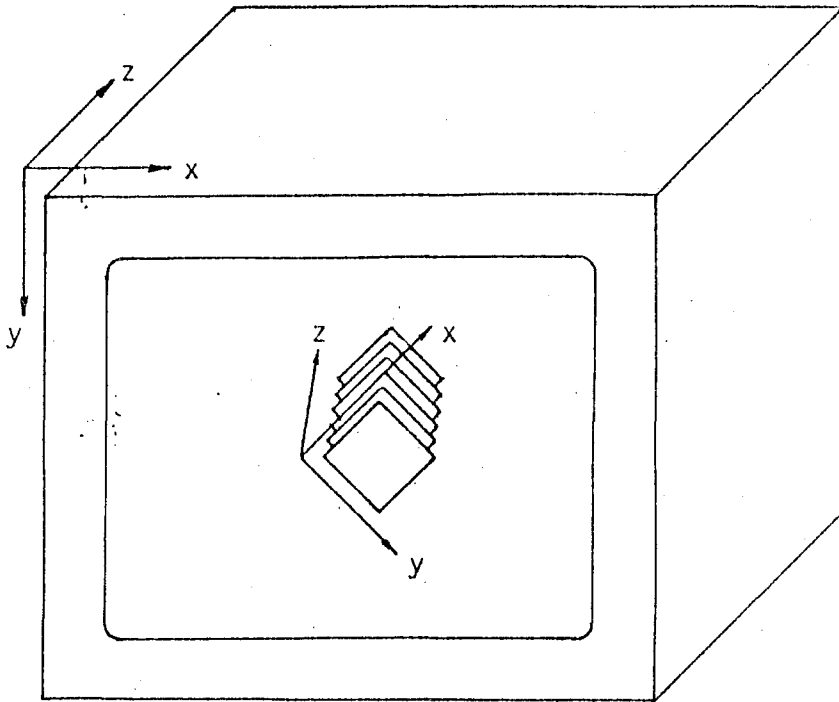


Fig1.3- The arbitrarily rotated stack of 2D arrays are projected onto the screen in order to obtain a 3D misualization.

The specific problem to be solved can be expressed as follows:

Given the slice data obtained by CT scans, find the display points and their gray values corresponding to the three dimensional image of the desired organ from any viewing angle.

Most implementations so far have been on rather large minicomputers and their cost consequently high. This thesis aims to develop a 3D display system with the following characteristics:

- 1.Low cost
- 2.Flexibility
- 3.High image quality

1.2 DESCRIPTION OF THE 3D DISPLAY METHOD AND EQUIPMENT USED IN THIS THESIS

The algorithm used in this thesis consists of the following steps:

a) Interpolation:

Generally the pixel size is smaller than the slice thickness. In order to obtain cube shaped voxels, one needs to interpolate between slices. A simple linear interpolation method is used in this thesis.

b) Identification of the Organ of Interest (Thresholding):

In this step the range of CT numbers characterizing the organ of interest is determined. This way, a binary 3D array is obtained.

c) Boundary Detection:

The boundary of the organ surface is detected by using the algorithm developed by G.T. Herman et. al. (See ref.1.2) The output obtained is a set of cube faces making up the surface of the organ.

d) Rotation and Projection:

The object is rotated by using a 3×3 rotation matrix. Then the perspective feeling is given by using orthogonal projections.

e) Hidden Surface Removal:

To give realism to the projected image, the parts that are behind the closer parts should not be displayed. This is achieved by the z-buffer algorithm.

f) Shading:

A simple shading rule is used which determines the gray

value at each point according to the depth and orientation of the surface elements.

The microcomputer used for this application is an IBM PC with 512 K Bytes of core memory, a 5 1/4 " disk drive and a 20 M Bytes of hard disk. The software package has been written by using Turbo Pascal (Version 3.0). An inhouse built image display terminal has been used for displaying the 3D images. This terminal can display a 256x256 image with 256 gray levels.

1.3 PLAN OF THESIS:

The next chapters of this thesis are organized as follows:

In Chapter II a review of existing 3D display techniques and their medical applications is given.

In Chapter III the terminology, the definitions and the theorems needed for the description of the 3D display algorithm is given.

Chapter IV presents the detailed description of the boundary detection algorithm.

Chapter V presents the hidden surface removal technique.

Chapter VI describes the geometrical transformation procedures.

In chapter VII the various projection techniques are discussed.

In Chapter VIII, shading rules are described.

In Chapter IX, the 3D implementation used in this thesis is described.

In Chapter X, the main results are discussed and suggestions for future study are made.

II.STATE OF THE ART

2.1 STATE OF THE ART

The computer technique for the representation of 3D object on a 2D screen such that a perceptual impression of a truly 3D image is created . A categorization of the techniques (3) used are shown below:

A. Re-organization

- 1- Reslicing
- 2- Reprojection

B. Surface display

- 1- 1-D unit based
- 2- 2-D unit based
 - a) Slice by slice
 - b) True 3-D
- 3- 3-D unit based
 - a) octree encoding
 - b) back to front display method

A. Reorganization

A discrete 3D scene may be considered as sets of 2D information. The methods used in this section, reorganize the 2D data so that a more desirable scene is obtained. No graphics processing is included because each item of 2D information may be displayed on a 2D monitor.

1) Reslicing (3)

The basic idea is, given a scene, to produce a set of slices belonging to arbitrarily oriented parallel planes that intersect the scene. In other words, a subset of voxels intersected by a specified plane are determined. This technique is useful if the scanner allows imaging at only specific orientations.

2.Reprojection (3)

The density of any point on the projection plane is determined by a weighted sum of the densities of all points of the scene along the line perpendicular to that point. By a proper choice of weights, obscuring features may be completely suppressed. Features that are obscuring but are anatomic landmarks may be made transparent, and the structure of interest may be highlighted.

The result is a 2D image requiring no graphics processing.

B. Surface display

The objective is to simulate the appearance of the surfaces of an object. This is the way humans perceive the world. The technique is made up of following steps : object identification, representation of the object, detection/formation of the surface of the object, and display of the surfaces. The classification of techniques under this approach is based on the object representation scheme. That is, in the 1D-unit approach the object is represented as a set of one dimensional borders on the slices. In the 2D-unit approach, it is represented as a set of patches and the 3D-unit-approach as a set of surfaces of 3D primitives.

1. 1D-Unit-Based Approach

Considering the k^{th} slice of a scene, the intersections of the surface of an object with this slice, gives the border of the object in that slice. Therefore, the surface of the object and the object itself can be represented as a stack of borders.

Object identification is done by thresholding and a binary scene results. Those voxels that belong to the object are assigned "1", while the others "0". The stack of borders may be rotated in three space to bring the object to any

desired orientation. After hidden-line elimination, the image may be projected, to produce a wire-frame type of surface display. Although this type of display can not produce fine images, it is computationally simpler than the 2D and 3D-unit-based approaches.

2. 2-D-unit based approach

In this approach the surface is made up of surface elements, referred to as patches

a) Slice-by-slice approach

Two methods will be describes here which share the main properties of the slice by slice approach.

Triangulation (3,4)

Here, the representation of the object is a stack of borders. Either a border following algorithm is used or the borders are traced out on each slice using a joystick.

The next step is the formation of the surface. The approach is to "tile" between a pair of borders. One border is selected on one slice and the other selected on the next slice. In the triangulation techniques, a triangle is formed by using a pair of points on one border and the third point on the other border see fig.2.1.1. First of all, the border is "resampled" so that the number of point are reduced, and the curve smoothed. The complexity of the algorithms is a

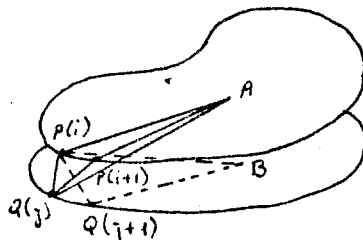


Fig.2.1.1 Illustration of the concepts used in triangulation techniques

function of the number of points on the border. These resampled borders are called contours.

The problem can be expressed as follows: Given a set of contours, produce a set of triangular patches by choosing triples of points. The result must be an "acceptable" surface. If we assume that the contours in a pair of successive slices produce an acceptable surface then the problem is much simplified.

Suppose a sequence of points $P(i), i=1, \dots, m$ representing the contour on the upper slice, and a set of points $Q(i), i=1, \dots, n$, representing the contour on the lower slice are given. The aim is to produce the acceptable triangles between the two slices.

Keppel's solution (3) is based on the graph-theoretical formulation of the problem.

All triangular patch arrangements are represented as a directed graph G .

The span from $P(i)$ to $Q(j)$ is denoted by $v_{i,j}$

$G = (V, E)$, where

$$V = \left\{ v_{i,j} / i = 1, \dots, m-1 \wedge j = 1, \dots, n-1 \right\}$$

$$E = \left\{ (a,b) / a = v_{i,j} \wedge (b=v_{i,j-1} \vee b=v_{i-1,j}) \right\}$$

The problem of triangulation is to find the appropriate directed path. Keppel determines the optimal triangulation by maximizing the volume enclosed. However complications arise in such a technique. Convex and concave segments do not match. Several different methods to find the appropriate surface are introduced. They try to choose one out of the many tiling arrangements in a systematic way. There is no

guarantee that the result will be acceptable. So searching is given up and a heuristic scheme is used. In the method of Cook et al. (4), the centroid of each region bounded by each contour is calculated. Let A and B be the calculated centroids (see fig.2.1.1), $V(i)$ is the unit vector along AP(i), $W(j)$ is the unit vector along BQ(j). An initial span P(1) Q(k) is chosen such that

$$V(1) \cdot W(k) > V(1) \cdot W(j) \quad 1 \leq j \leq n.$$

If P(i) Q(j) is the current span, the next span chosen is P(i+1) Q(j) if

$$V(i+1) \cdot V(j) > V(i) \cdot W(i+j)$$

otherwise, P(i) Q(j+1) is chosen

The intuitive idea is to select spans whose orientations are similar, as closely as to the orientation of the line joining the centroids. If the contours are of similar shape, and oriented similarly, this method produces acceptable results, and not as complex as the ones that predict the best arrangement.

After the patches are produced, graphics processing is introduced to produce the image.

The procedures stated above have some serious drawbacks. There is no guarantee that the tiling done in an automatic fashion will produce acceptable images. Medical objects are complex in shape. The assumption that the contours are similar in shape and orientation from slice to slice is not reasonable. The complex cases may be taken care of interactively, but this may become cumbersome. Also, processing time is an important factor.

Directed Contour Method (3,5)

Another slice by slice approach which is not confronted with such problems has been proposed by J.K. Udupa (5) . The surfaces produced by this method are faces of voxels. Although 8-adjacency is defined in the terminology section, it is repeated here for the case of the slice by slice approach.

An 8-contour is an ordered sequence v_1, v_2, \dots, v_n of voxels satisfying the following conditions:

- (i) $n \geq 3$;
- (ii) for $j = (i \pm 1) \bmod n$, v_j is 8-adjacent (on the same slice) to $v_{i,j}$
- (iii) for $i = 1, 2, \dots, n$, $v_i = v_j$ iff $i = j$.

One can then say that an 8-contour has at least three points. It is connected and closed. And the last property means that it is not self-intersecting.

Two voxels v_1 and v_2 are

- i) x-adjacent (denoted as $v_1 \times v_2$) if

$$|x_1 - x_2| = 1 \text{ and } y_1 = y_2 \text{ and } z_1 = z_2$$

- ii) y-adjacent ($v_1 \text{ y } v_2$) if

$$|y_1 - y_2| = 1 \text{ and } x_1 = x_2 \text{ and } z_1 = z_2$$

- iii) Z-adjacent ($v_1 \text{ z } v_2$) if

$$|z_1 - z_2| = 1 \text{ and } x_1 = x_2 \text{ and } y_1 = y_2.$$

The boundary set $B(0)$ (boundary of organ) of 0 is defined as

$$B(0) = \left\{ (v, \bar{v}) / v \in 0 \wedge \bar{v} \in \bar{0} \wedge (v_x \bar{v} \vee v_y \bar{v} \vee v_z \bar{v}) \right\}$$

(v, \bar{v}) is a face of v .

Two other sets are defined as

$$B_{x,y}(0) = \left\{ (v, \bar{v}) / v \in 0 \wedge \bar{v} \in \bar{0} \wedge (v_x \bar{v} \vee v_y \bar{v}) \right\}$$

$$B_z(0) = \left\{ (v, \bar{v}) / v \in 0 \wedge \bar{v} \in \bar{0} \wedge v_z \bar{v} \right\}$$

The set $B_{x,y}(0)$ consists of faces perpendicular to the x-and y-axis and $B_z(0)$ consists of faces perpendicular to the z-axis. From the above equations,

$$B(0) = B_{x,y}(0) \cup B_z(0)$$

If $B_{x,y}(0)$ represents the subset of $B_{x,y}(0)$ on the k^{th} slice then

$$B_{x,y}(0) = \bigcup_{k=1}^Z B_{x,y}(0_k)$$

as a result the boundary of the organ $B(0)$ can be expressed as:

$$B(0) = \left(\bigcup_k B_{x,y}(0_k) \right) \cup B_z(0).$$

object is identified by thresholding.

The region bounded by a set of contours in a slice is defined to be the region to the left of the contours on that slice including the points on the contours. (see fig.2.1.2)

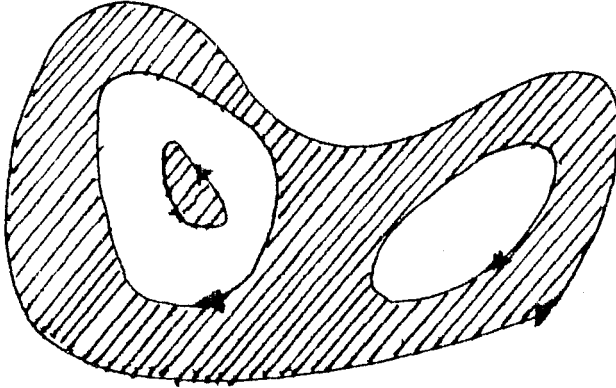


Fig.2.1.2 Object representation by directed contours. Object subregions lie to the left of the contours.

As can be seen, concentric contours are alternatively counterclockwise and clockwise. A set C^z is defined to represent the 8-contours associated with a slice. To assign a direction to the 8-contour, the notion of 8-chain is defined. 8-chain of an 8-contour v_1, v_2, \dots, v_n is a sequence $c_1, c_2, \dots, c_n (=c_0)$ of direction codes (see fig.2.1.3) such that for $1 \leq i \leq n$, c_i

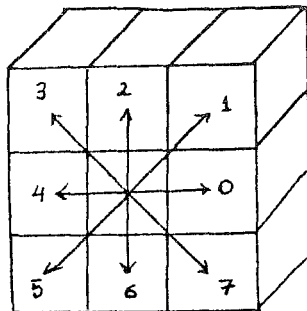


Fig.2.1.3 Definition of direction codes for an 8-chain

c_j is the direction code for $v_i v_j$, where $j = (i+1) \bmod n$. The 8-chain of the 8 contour $v_1 \dots v_{12}$ in fig.2.1.4 is 711235445716.

The notation is $c_i = (v_i, v_{i+1})$

Based on the definitions above, -x-edge voxel, +x-edge voxel, -y-edge voxel and +y-edge voxel are defined. i.e. a +x-edge voxel is one whose +x face is on the boundary. However a creation and use of a look up table for finding the -x, +x, -y, +y edge voxels is more efficient in determining the edge voxel and its face to see if v_i is can edge voxel, c_{i-1} and c_i must be considered (see table 2.1.1).

Let us consider v_6 in fig.2.1.4 $c_5=3$ and $c_6=5$ from the look-up table the boundary faces are the -x, +x, and +y faces of the considered voxel. For the k^{th} slice, $B_{xy}(0)_k$ is computed by determining the contribution of each element v of C_k using the look up table. The computation of $B_z(0)$ can also be done in a slice by slice fashion. This requires in addition to C_k the knowledge of the set C_{k-1} for the previous slice.

The surfaces formed this way are exactly the same way as the true 3D approach.

b) True 3D

This approach is implemented in this project and is explained in detail in the following sections.

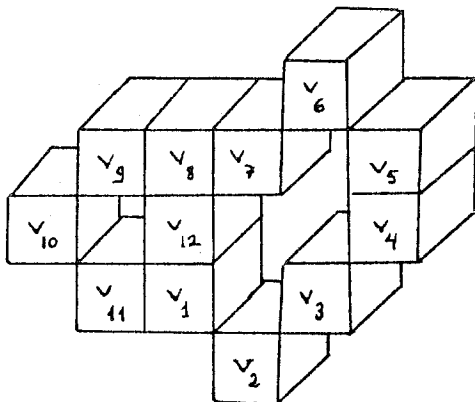


Fig 2.1.4 An illustration of 8-Contours

$c_i \backslash c_{i-1}$	0	1	2	3	4	5	6	7
0	-y	+x -y	+x -y	+x +y -y	-			-y
1	-y	+x -y	+x -y	+x +y -y	+x +y -y	-		-y
2		+x	+x	+x +y	+x +y	+x -x +y	-	
3		+x	+x	+x +y	+x +y	+x -x +y	+x -x +y	-
4	-			+y	+y	-x +y	-x +y	-x +y -y
5	-x +y -y	-		+y	+y	-x +y	-x +y	-x +y -y
6	-x -y	+x -x -y	-			-x	-x	-x -y
7	-x -y	+x -x -y	+x -x -y	-		-x	-x	-x -y

TABLE 2.1.1- The possible types of edge voxel a voxel can be in an 8-contour

3. 3D Unit-Based Approach.

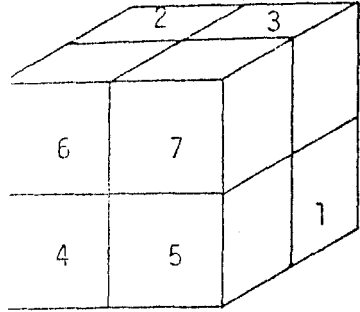
The techniques under this approach are derived from applications in computer-aided design and computer vision.

a) Octree encoding (6,7)

The octree for a picture is generated by recursive subdivision of the raster into octants. (see. fig.2.1.6). The root of the octree represents the whole raster. If the whole picture is of a single attribute (1 or 0), then the root is also a leaf and is assigned that attribute. Otherwise, it has eight sons, one representing each of the octants into which the raster can be divided. The same process is applied recursively to each of the eight sons. The recursion must terminate after at most R levels of recursion, because at that point the corresponding octant is a single voxel.

An encoded object B is defined as a family of ordered pairs $B(R) = (P, E(R))$. $E(R)$ are the octants at level of resolution R . In octree encoding, the octants are the nodes of the tree structure. The tree contains all octants with increasing resolution up to the maximum resolution R . The root is at the top node structure and all other nodes exist below the root. P is a set of properties. The property describes the state of the octant. It may have the value EMPTY, PARTIAL, and FULL, meaning that the octant is free from object, partially filled by the object and completely occupied by the object, respectively.

The position of a child node is indicated by a value 0 to 7. (see fig.2.1.6). Each node can be uniquely identified by its node address which is a string of numbers from the child number set 0,.....7. The root is represented by an empty string. So the node address of a child is its number preceded by the node address of its parent.



EMPTY (E) - cube is empty

FULL (F) - cube is completely enclosed by the object

PARTIAL(P) - cube partially intersects the object

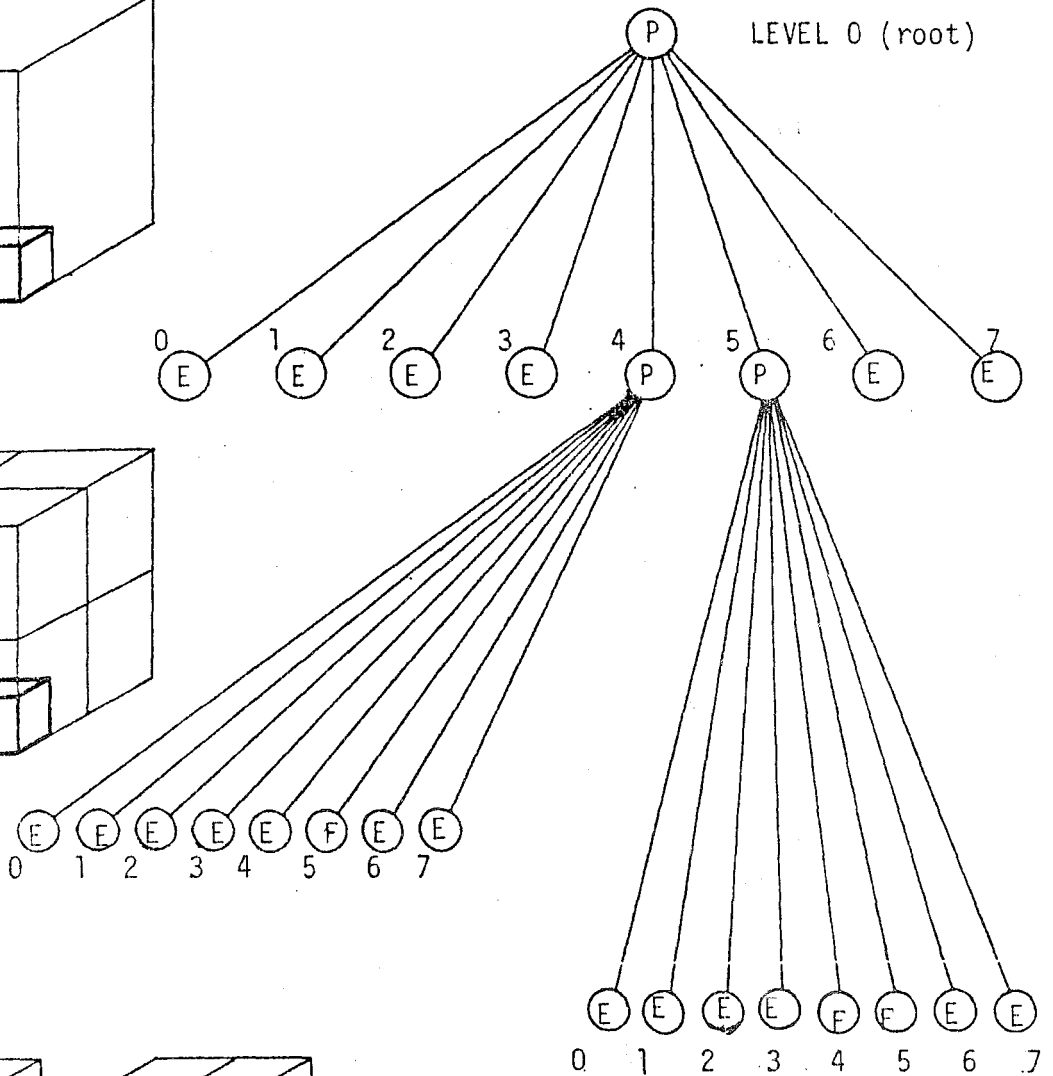
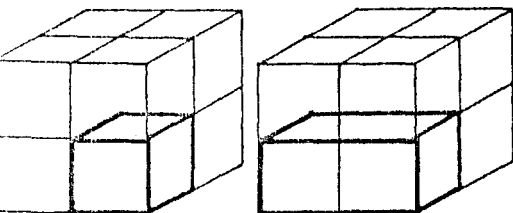
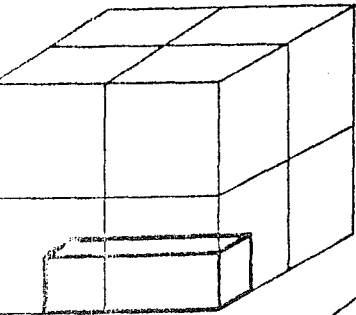
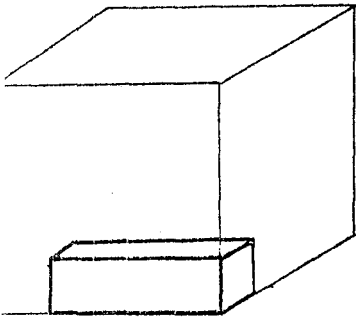


FIG.2.1.5 Simple object represented in octree encoding format Numbering sequence, label definitions, three level representation of an object.

The sequence of child node numbers identify a particular node and determines the traversal necessary to locate the node relative to the root. This also tells the section of space covered by the node .

As a result of this procedure 0 and 1 attributes making up the raster are all encoded and stored.

Linear octrees are a compressed form of regular octrees described above (8). Here, only the leaves belonging to the object are stored and each are labeled with a key consisting of a string of R digits and an integer from 0 to R. This key allows one to find the path from root to leaf in the corresponding regular octree. As a result, he can also find the subdivision of the raster that produced that octant, its size and its location in the raster. The octants can be labeled in any way. Suppose the conversion is as given in table.2.1.2.and 2.1.3. Consider in fig.2.1.7 the node labeled 721. This designation means that the corresponding voxel is in the front-north-east octant of the first division. In the backnorth-west octant of the second subdivision, and in the backsouth-east octant in the third (and final) subdivision.

Octants that are leaves and not single voxels are labeled with the key of the back-south-west corner voxel, and an integer called grouping factor follows. This number is the distance of that octant from the root or equivalently, the logarithm of the number of voxels contained in that octant. For example, in fig.2.1.7 octant (160,1) represents the groupign of 8 voxels. ($\log_8 8 = 1$, therefore the integer has a value of 1).

The address of the octant (160,1) is 16. The position of the back-south-west corner voxel is 0. So the representation (160,1) is obtained.

For the regular octree in fig.2.1.7, the linear octree is the array of codes is the array of codes written as:

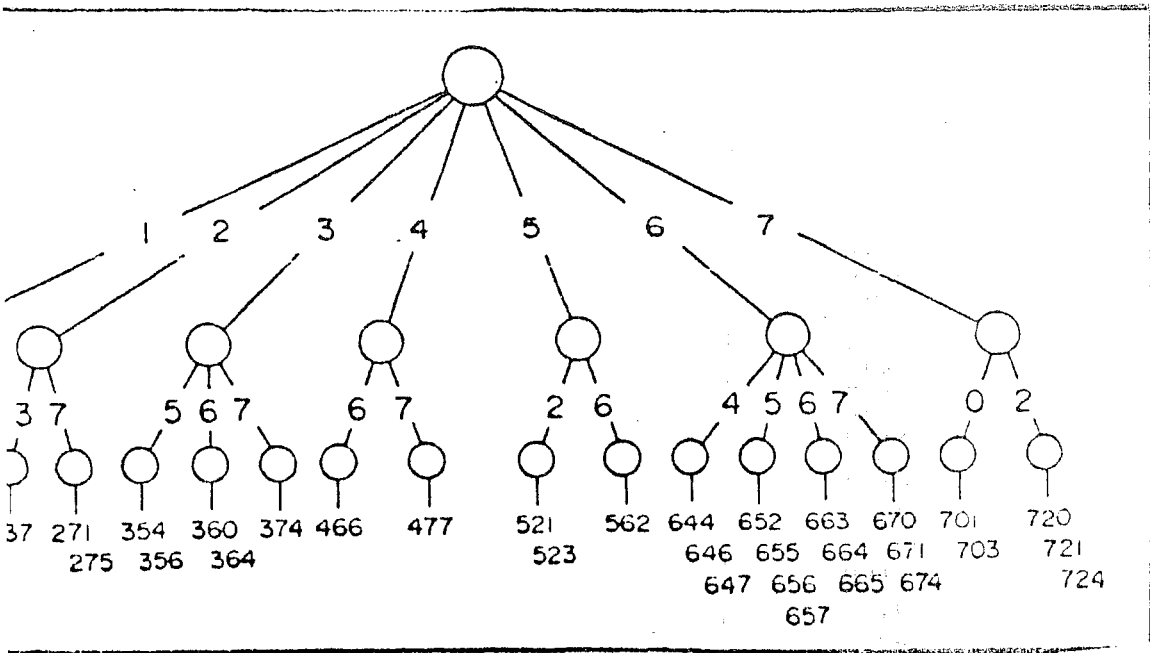
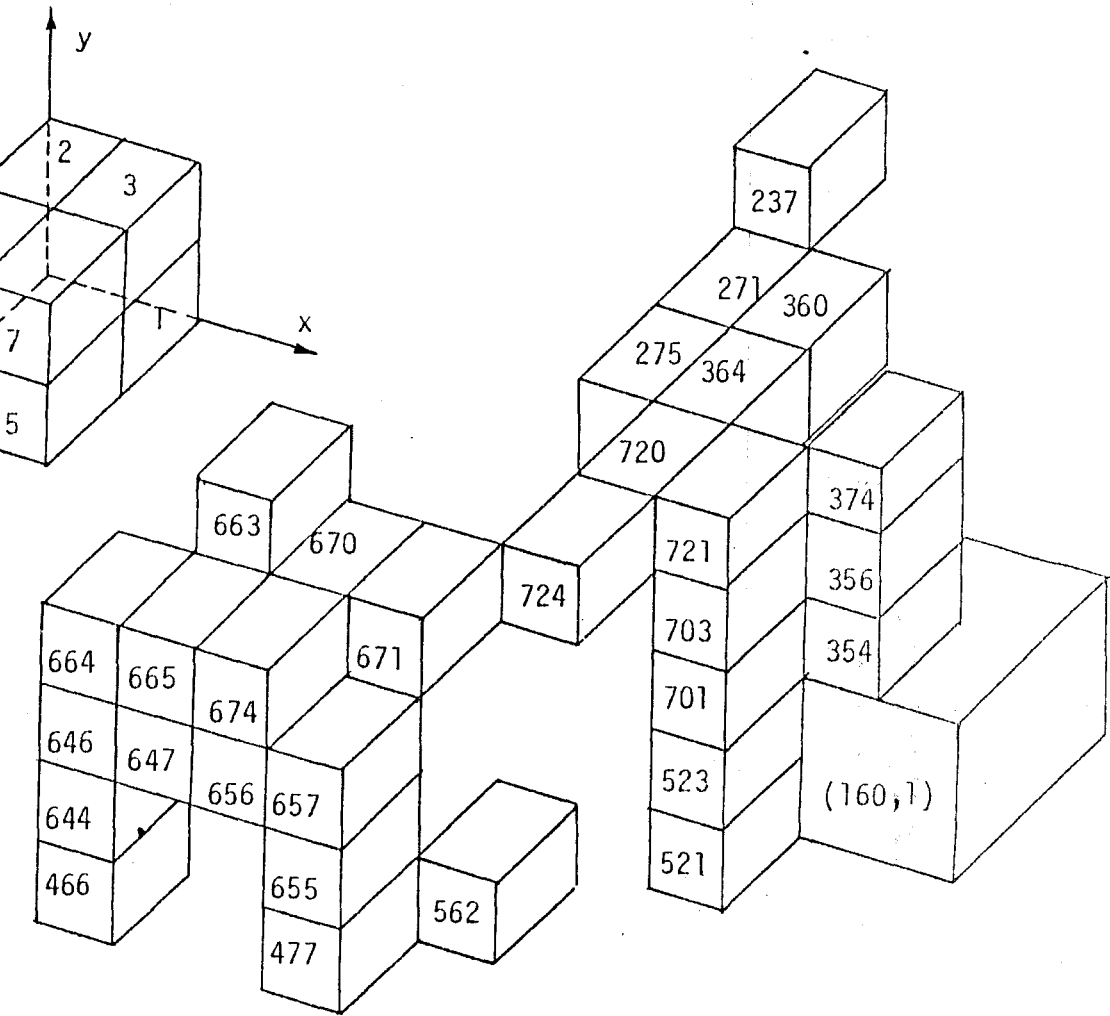


FIG. 2.1.6 Linear octree representation of an object

Table 2.1.2 Labelling octants.

Octant	Octal digit	3-bit digit
back-south west	0	000
back-south-east	1	001
back-north-west	2	010
back-north-east	3	011
front-south-west	4	100
front-south-east	5	101
front-north-west	6	110
front-north-east	7	111

Table.2.1.3 Priority order for octant labeling according to $VPN = (x_1, y_1, z_1)$.

z_1	y_1	Octal digit	Priority order
0	0	0	7, 6, 5, 3, 4, 2, 1, 0
0	0	1	6, 7, 4, 2, 5, 3, 0, 1
0	0	2	5, 4, 7, 1, 6, 0, 3, 2
0	0	3	4, 5, 6, 0, 7, 1, 2, 3
0	0	4	3, 2, 1, 7, 0, 6, 5, 4
0	0	5	2, 3, 0, 6, 1, 7, 4, 5
0	0	6	1, 0, 3, 5, 2, 4, 7, 6
0	0	7	0, 1, 2, 3, 4, 5, 6, 7

(160,1),237,271,275,354,356,360,364,374,466,477,521,523,562,
644,646,647,652,655,656,657,663,664,665,670,671,674,701,703,
720,721,724 .

Note that each node labeled full represents, in general, a parallelepiped. Therefore, subscenes represented by parallelepipeds labeled full may be displayed on the screen. If the leaves labeled full are displayed in a sequence such that the ones farthest away from the observer are displayed first and the next ones are displayed in the order of diminishing distance, over painting the parts already painted, the visible parts of the object will be displayed. Thus, no hidden surface removal algorithm is used. A priority coding of the voxels to be displayed can be prepared as a look up table. (See table 2.1.2 and 2.1.3 For example, if the observer is in the front-north-west corner of the view volume, the closest suboctant is 1,0,3,5,2,4,7,6. So, since (160,1) is the only octant which is starting with "1", it is displayed first. Since there is not a node starting with "0", the nodes starting "3" are searched. They are 354, 356,360,364,374. Among these the ones whose second octal digit is "5" comes first (354,356) and then whose second digit is "7" (374) and then comes "6" (360 and 364). Thus, the nodes would be displayed in the following sequence.

(160,1),354,356,374,360,521,523,562,237,271,275,477,466,701,
703,721,720,724,655,652,657,644,646,671,670,674,663,665,664 .

b) Back to Front Display Method

Another display approach based on the idea of spatial presortedness described above, is developed (9), in which the voxels are traversed slice by slice to project each voxel on the screen. The approach called the back-to-front display method and proposed by Gordon, is based on traversing the slices, rows and columns of the array in order of decreasing distance from the observer. The 2D analog of the problem is shown in fig.2.1.7. Assume the voxels A,B,C and D

Fig.2.1.7 Back-to front voxel readout in two dimensions. For the purpose of illustration, assume the origin is at the corner farthest from the observer. The voxels are traversed in order of increasing values of x and y ; either x or y may be chosen as the faster running index.

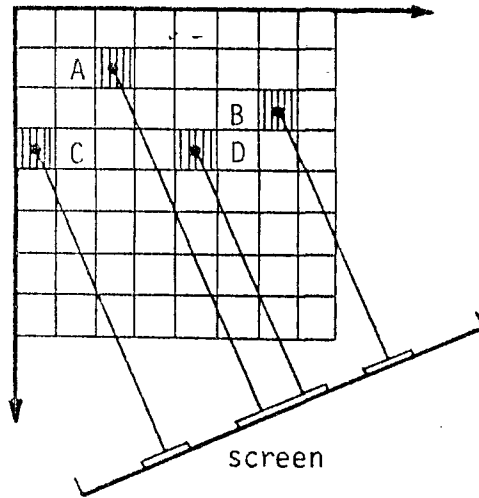
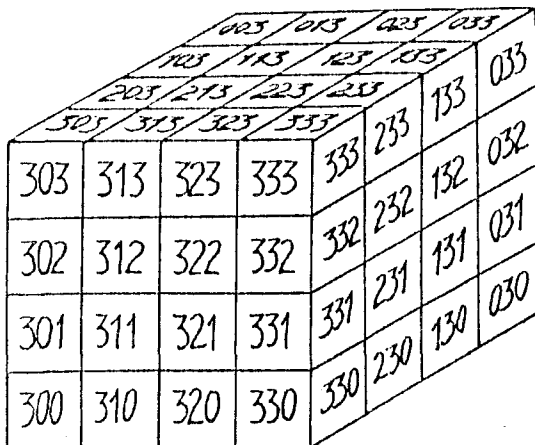


Fig.2.1.8 Slice-by-slice back-to-front voxel readout. Slices are read out starting with the slice farthest from the observer. Within each slice, voxels are read out starting with the corner farthest from the observer. The voxels are labeled with three digits, identifying the $x, y,$ and z coordinates in object space. For the object orientations shown the voxels can be read out in several slice-by-slice back-to-front sequences including 000,100, 200...010,110,210...133,233,333.



are full and are to be projected on the screen. The other voxels are empty. The voxels are traversed in the increasing order of x and y . Either x or y may be the fastest changing index. So the sequence A,B,C,D (x changes fastest), and C,A,D,B (y changes fastest) will give a correct picture. The reason for this is that if (part of) a voxel (x,y) is obscured by (part of) a voxel (x',y') , then $x < x'$ or $y < y'$, so by projecting (x,y) before (x',y') , the correct image is obtained. The choice of which index out of x,y , or z should change fastest is arbitrary, so a slice by slice approach can be taken. (See fig.2.1.8)

2.2 MEDICAL APPLICATIONS OF 3D IMAGING

3D display algorithms have found a wide range of applications in medical diagnosis and treatment. Surgical planning, quantitative bone densitometry, radiation therapy planning, prosthesis design are some of the current applications. Research on how to use 3D techniques in the most effective way and a search for other potential applications are still on the way. This section summarizes some of these application areas.

Sites of evaluation and diagnostic applications of the 3D display technique

The applications and results described below are performed and reported by the researchers at the University of California at San Diego.

Shoulder- 3-D images of the shoulder have produced excellent pictures of the osseous structures, including the scapula, humerus and distal portion of the clavicle. In the previous anterior dislocation of the glenohumeral joint, 3-D imaging allows for the definition of size and location of secondary Hill-Sachs, and osseous Bankart lesions, 3-D display is also capable of the precise size, location and configuration of intraarticular osseous fragments.

Comminuted fractures of scapula or proximal region of the humerus are easily diagnosed by routine radiography, but the precise relationship between osseous fragments is difficult to define with this method. Routine axial CT images show intraarticular extension of scapular fractures. 3-D display shows optimally the relationship of each fragment to its neighbors.

Hip: Three dimensional analysis of several types of hip diseases are performed at UCSD(10) It has been found most useful in ischemic necrosis of the femoral head, osteoarthritis, post traumatic abnormalities, and acetabular dysplasia. In the congenitally malformed hip, 3-D technique

2.2 MEDICAL APPLICATIONS OF 3D IMAGING

3D display algorithms have found a wide range of applications in medical diagnosis and treatment. Surgical planning, quantitative bone densitometry, radiation therapy planning, prosthesis design are some of the current applications. Research on how to use 3D techniques in the most effective way and a search for other potential applications are still on the way. This section summarizes some of these application areas.

Sites of evaluation and diagnostic applications of the 3D display technique

The applications and results described below are performed and reported by the researchers at the University of California at San Diego (10). Shoulder- 3-D images of the shoulder had produced excellent pictures of the osseous structures, including the scapula, humerus and distal portion of the clavicle. In the previous anterior dislocation of the glenohumeral joint, 3-D imaging allows for the definition of size and location of secondary Hill-Sachs, and osseous Bankart lesions, 3-D display is also capable of the precise size, location and configuration of intraarticular osseous fragments.

Comminuted fractures of scapula or proximal region of the humerus are easily diagnosed by routine radiography, but the precise relationship between osseous fragments is difficult to define with this method. Routine axial CI images show intraarticular extension of scapular fractures. 3-D display shows optimally the relationship of each fragment to its neighbors.

Hip: Three dimensional analysis of several types of hip diseases are performed at UCSD (10). It has been found most useful in ischemic necrosis of the femoral head, osteoarthritis, post traumatic abnormalities, and acetabular dysplasia. In the congenitally malformed hip, 3-D technique

defines the location and extent of acetabular deformity, the dimensions of the true acetabulum and false acetabulum. This information allows the categorization of the dysplasia present. Also, the the degree of femoral deformity can be determined.

Conventional radiography is used in the initial diagnosis of osteoarthritis, demonstrating classical patterns of head migration. Axial CT with sagittal and coronal reformations further define areas of joint space narrowing and osteophytosis. The location of osteophytosis is precisely determined by 3-D imaging.

Surface modeling of the diseased region provides orthopedic surgeons with an an accurate preoperative assessment of the abnormal hip and allows them to rehearse the surgery.

Early diagnosis of ischemic necrosis of the femoral hip is best achieved by scintiagraphy or magnetic resonance imagine (MRI), With CT, the extent of bone necrosis and the degree of asseous collapse of the femoral head in patients with advanced disease can be determined. The 3-D image precisely shows the extent and location of surface collapse as it might appear during a surgery and may influence the choice of therapy (10)

Spine 3-D imaging is best suited for the evaluation of the spine especially the analysis of the injured vertebral column. Complex fractures and fractures with intraspinal fragment represent the types of injures that are evaluated with this technique. Comminuted fractures of the vertebral bodies can be visualized by radiography. Only CT allows the examination of posterior displacement of fracture fragments, However, 3D display best demonstrates the relationship between osseous fragment and the precise location of the bone within the spinal canal.

A variety of degenerative diseases of the vertebral column can be evaluated with 3D CT images. Abnormalites of

spinal curvature can result in distortion of spinal anatomy and causes the interpretation of radiographs and CT scans to be difficult. The conditions that produce such confusions are bilateral spondylolysis with spondylo listhesis, associated narrowing of the intervertebral disc space kyphosis, scoliosis, and conditions that produce such confusion. 3-D surface images readily reveal the degree of narrowing of the intervertebral disc bone sclerosis, vertebral slippage and compromise of the spinal canal. Three dimensional CT is especially useful in degenerative disc disease both surface and sectional image displays are used. In the postoperative spine, 3-D CT can define sites of nonosseous union and pseudarthrosis. In this situation, 3-D technique produces superior images to axial CT scans, and better than sagittal coronal reformatted images.

Surgical Planning

Preoperative modeling of diseased articulations is extremely useful in planning of specific surgical strategies. The likelihood of a successful surgery is then increased tremendously.

Interactive Surgical Planning

Anatomic structural deformities in three dimensions can be visualized, and manipulated by inspecting the CT images of the patient. The three dimensional distortion is usually predicted by mentally constructing the image after reviewing the CT slices. 3D display enables a more effective description of the organ, and allow for a better physical examination, and intraoperative findings than the 2D modalities.

Even when using 3D display mode some parts can never be examined. However, with interactive manipulation, the parts that obscure the places that the physician is interested in, can be removed from the display. In craniofacial surgery, the planning of anomalies is based on the use of life size photographs and cephalometric radiographs of the patient.

Parts of prints are cut and removed, to find out the results of such skeletal changes. However, since the third dimension is not included, such techniques are not accurate.

In the ISP (interactive Surgical Planning) scheme (10) interactive manipulation techniques are used so that a surgical procedure can be simulated on a CRT. This will allow the surgeon to consider the probable intraoperative problems beforehand, determine the requirements for the bone graft to be used, and the duration of the operation.

In the ISP scheme, with a graphics input device, the surgeon carries out manipulations on the organ and the result is displayed on the screen. The manipulations continue until the desired result is obtained on the screen. Then the system produces a quantitative description of the actions to be taken to produce the result.

The manipulations are mostly of two types. Either any arbitrary part is removed from the organ or any arbitrary part is added. The amount and type of removal or addition operations depends on the model used. The modified organ is the organ before the manipulation combined with the model. The model can be a part of an organ, or a model can be defined. In the cuberrile approach the model is a set of voxels.

When a model M and an organ O are combined additively, the modified organ O' is

$$O' = O \cup M$$

when the model M is the part to be removed from the organ O , the modified organ

$$O' = O - M.$$

The ISP consists of the following steps:

- 1) Generate a display using the method chosen
- 2) Inspect the organ. Choose a model suited for the desired

manipulation.

- 3) Bring the model to a proper bearing with respect to the organ, by a composite display of the organ and the model.
- 4) Create and display the modified organ that result from the operations carried out with the model.
- 5) Get the necessary and important informations about the operation,

The above steps are repeated until the desired display is obtained.

The manipulations are carried out on a single slice but the result is shown on a 3D display.

The models can be geometric shapes such as squares and circles whose variables can be changes. The models can be combined to produce new models. Another way to generate a model is by using a graphics input device, such as trackball or joystick. The user creates the model and it can be of any shape. Yet, some geometric curves are difficult to generate in this way.

Another way to represent the model is by borders. Eventually, the model is a set of directed contours

(11)

Alternatively, a model can be represented as a set of pixels, called masks. They constitute the regions of the model.

Before the selected operation, the model should be correctly spaced. This is achived interactively, by the use of the composite display of the organ and the model.

One method of composite display is superimposing the display of the model on the organ. In such a display 3D space relation of the model and the organ must be consistent.

The method of manipulation depends on the schemes of representation of the organ and the model. If both the model

and the organ are represented by masks, then the operation is simply a set union and subtraction. If one of the organ or the model is represented by directed contours, the representations is first converted into masks then the union and subtraction operations take place.

If both are represented by contours, then the organ is manipulated interactively, as a result of the modification of contours in each slice.

In the three dimensional ISP scheme, 3D techniques are used throughout.

The drawbacks of the two dimensional scheme are that the user must have a good perception of the structure of the manipulated organ, if the organ is complex and relating slice information is difficult. Also, it may take a lot of user time if the number of slices is large.

Two types of models can be generated. In the first method, the model is generated from the algebraic compositions of primitive objects of predefined shapes. In the second method a model of arbitrary shape is produced interactively using a graphic input device. But to input such a shape by drawing on a screen is very difficult. So the following technique is used. The following is assumed:

- . The cutting surface is perpendicular to the screen
- . The user cuts the screen along a curve drawn by the user on the screen.
- . The scene is seperated into two
- . The user should specify the region in which the modified organ is in.

The model could be represented as a set of directed contours or in terms of masks. If both the model and the organ are represented with the same method then the ISP scheme runs faster, i.e. the representation of the organ and the model and the computation of the modified organ is quick.

The composite display of the organ and the model may be done in several ways. The organ may be displayed as transparent while the organ opaque. The organ and the model may be of different colors. All the information regarding the operations-size, shape, type, location etc., are recorded. The information should be complete so that the planning done during an ISP experiment can be duplicated in an actual surgery. (This aspect of ISP requires much further work.)

Accurate measurement of distances, angles, and volumes is very useful in presurgical planning and assesment of bone resorption, assesment of intracranial or orbital volumes (in cranio facial operations) with cranical vault remodeling and orbital advancement.

-Prosthesis manufacturing

CAD/CAM systems are combined with medical image data to manufacture structure models or prosthesis. In such systems, bone contour files and an algorithm to generate instructions for milling machine are used to manufacture models in precise dimensions.

The most common prosthesis designed with the technique used by M.L.Rhodes et al. (12) is a resurface plate for the patello-femoral groove-the surface over which knee caps glide as we flex our knees. The people who need such prosthesis are the ones who have pain because of bone irregularities on this surface. The femur at this point is very thin, with soft vascular marrow below. No standard resurface plate can be fitted without removing cortical bone, which weakens the knee. Only custom-designed plate is appropriate. Cemented implants are used extensively in total hip replacement surgeries. Although they perform well in less active patients, in young patients high implant failure rates occur within just five years. These, and the poorer long-term results of revised cemented implants, simulated design of cementless implants.

According to clinical and laboratory evidence the long term success rate of cementless hip stem is improved with an implant that provides maximal filling of the femoral canal and resistance to axial torques. The reasoning is that the maximum area of implant contact on the internal cortical bone surface produces tolerable levels of stress, reduces bone resorption and sinkage, and restricts micromotion. These design considerations are included in the newer, more anatomically shaped hip stem designs.

So far the conventional custom implants have fallen short because they have been designed with only two radiographic views (anterior-posterior and lateral) to model the bone. Failure to consider all the bones and the 3D geometry reduces implant-bone fit significantly. An interactive algorithm to design optimal-fit custom implant is described in (13) .

III. TERMINOLOGY

3.1 GENERAL TERMINOLOGY FOR 3D

The terminology introduced by Herman et al is used(1,14) A discrete 3D scene(or scene) consists of a X,Y,Z rectangular parallelepiped which is referred to as the region of the scene. This rectangular parallelepiped is subdivided by three sets of parallel planes into smaller identical volume elements called voxels. Each voxel has a value assigned to it called the density of the voxel. In many applications, the density is a set of integers $\{L, L+1, \dots, u-1, u\}$. If $L=0$ and $U=1$ then the scene is a binary scene. A rectangular coordinate system is used to describe object so that a voxel v may be described by a triple (x,y,z) of integers, where $1 \leq x \leq X$, $1 \leq y \leq Y$, and $1 \leq z \leq Z$. The $i \times j \times 1$ scene formed by all voxels (i,j,k) for a fixed k is called the k th slice of the scene.

In surface display procedure, the aim is to find and describe the surface of an object in the scene. This involves segmenting the original scene into an object and background. For example, a binary scene can be created where a "1" assigned to a voxel indicates that the voxel is in the object, while a "0" value assigned to a voxel means that it is outside the object.

3.2 GRAPH THEORETICAL TERMINOLOGY

The definitions in this section follow (15). The graph theoretical terminology used in boundary detection are explained.

Simple graph G is a pair $(V(G), E(G))$, where $V(G)$ is a nonempty finite set of elements called vertices (or nodes, or points), and $E(G)$ is a finite set of unordered pairs of distinct elements of $V(G)$ called edges, (or lines).

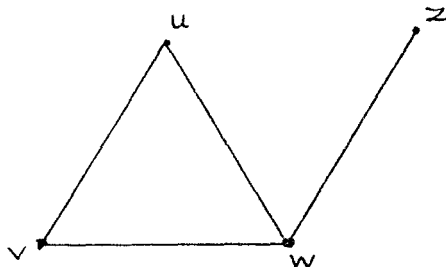


Fig.3.2.1 A simple graph G .

Fig.3.2.1 is a simple graph. The nodes are the set $V(G) = \{u, v, w, z\}$, and its edge set consists of the pairs $\{u, v\}$, $\{v, w\}$, $\{v, w\}$, $\{u, w\}$ and $\{w, z\}$. The edge $\{v, w\}$ joins the nodes v and w .

An edge may not join two distinct nodes. Then loops may occur, i.e. edges may join nodes to themselves. An object with loops and multiple edges is called a graph (general graph). So, a graph, is the pair $(V(G), E(G))$ where $V(G)$ is nonempty set of finite elements called nodes, and $E(G)$ is a family of unordered pairs of (not necessarily distinct) elements of $V(G)$ called edges. The word family allows the existence of multiple edges.

In fig.3.2.2 $V(G) = (u, v, w, z)$
 $E(G) = ((u, v), (u, v), (u, v), (v, w), (v, w), (u, w))$
 and w, z

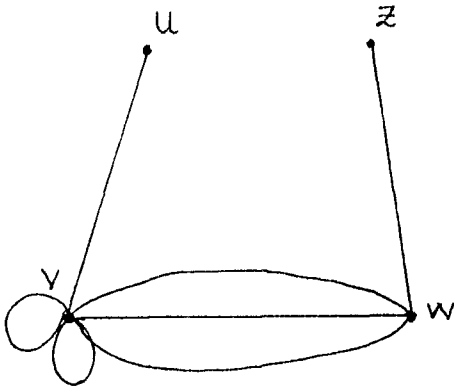


Fig.3.2.2 A general graph

The pair $\{v, w\}$ joins the nodes v and w .

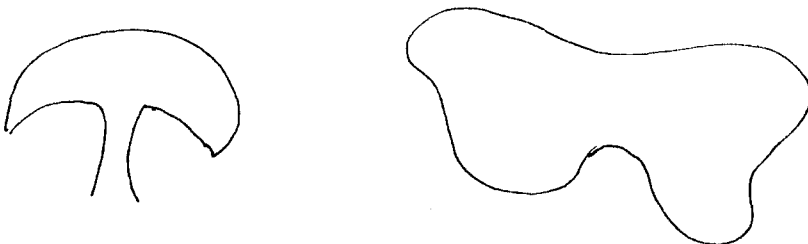
A subgraph of a graph G is a graph, all of whose nodes belong to $V(G)$ and all of whose edges belong to $E(G)$.

Two nodes v and w of a graph G are called adjacent if there is an edge jointing them. (i.e. there is an edge of the form (v, w)). The nodes v and w are then said to be incident to such an edge.

A Jordan Curve is a curve which does not intersect itself. A closed Jordan Curve is one whose end points coincide. (See Fig.3.2.3) Jordan curves can be defined in 3-space.

Jordan Curve Theorem (16)

A closed Jordan curve J in a plane divides the plane into two nonempty sets A and B , neither meeting J , such that any two points in A or B can be joined by a polygonal arc in A or B respectively and any polygonal arc joining a point of A with a point of B meets J . (see fig.3.2.3)



Given any graph G , an edge-sequence in G is a finite sequence of edges of the form $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{m-1}, v_m\}$. An edge sequence has the property that any two consecutive edges are either adjacent or identical. An edge sequence determines a sequence of nodes v_0, v_1, \dots, v_m ; v_0 is the initial node and v_m is the final node of the sequence, and it is an edge sequence from v_0 to v_m . (See fig.3.2.4)

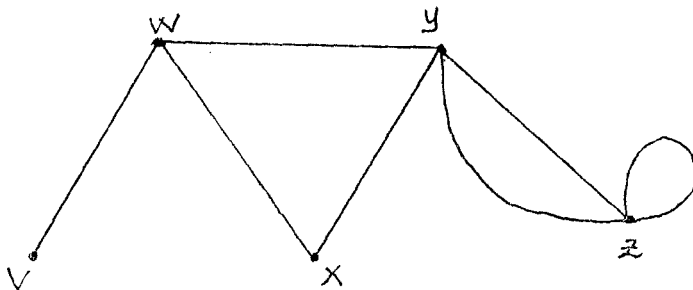


Fig.3.2.4 $v \rightarrow w \rightarrow x \rightarrow y \rightarrow z \rightarrow z \rightarrow y \rightarrow w$ is an edge sequence from v to w .

Edge sequence is a too general concept, so some restrictions are introduced. An edge sequence in which all the edges are distinct is called a path if in addition the vertices v_0, v_1, \dots, v_m are distinct ($v_0 = v_m$ is possible) then the path is a chain. A path or chain is closed if $v_0 = v_m$.

A graph G is said to be connected if given any pair of vertices v, w of G , there is a chain from v to w . A digraph D is the pair $(V(D), A(D))$. $V(D)$ is a nonempty finite set of elements called nodes (vertices), and $A(D)$ is a finite family of ordered pairs of elements of $V(D)$ called arcs. An arc whose first element is v and whose second element is w is called an arc from v to w , and is written as (v, w) . So the two arcs (v, w) and (w, v) are distinct.

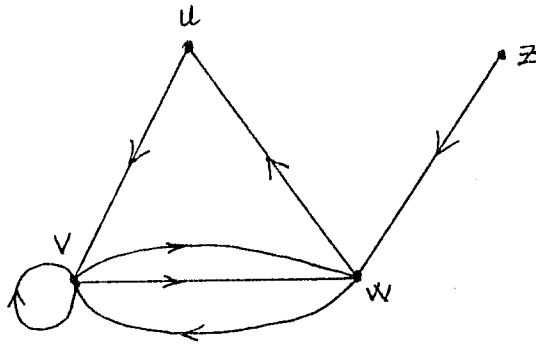


Fig.3.2.5 shows a digraph whose arcs are $(u,v), (v,v), (v,w), (v,w) (w,u)$ and (w,z) . The ordering of the vertices in an arc is indicated by an arrow.

If D is a digraph, the graph obtained from D by removing the arrows (i.e. by replacing each arc of the form (v,w) by a corresponding edge v,w) is called the underlying graph of D . See fig.3.2.6

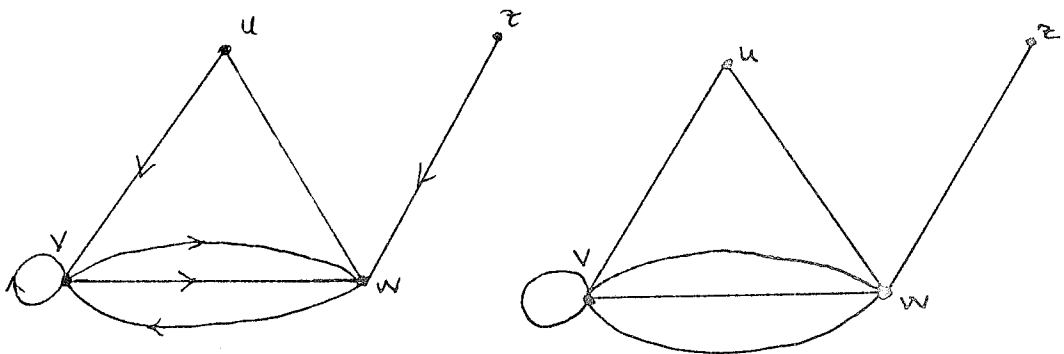


Fig.3.2.6a A digraph D b. underlying graph of D

An arc-sequence in a digraph D is a finite sequence of arcs of the form $(v_0, v_1), (v_1, v_2), \dots, (v_{m-1}, v_m)$. This sequence may be written as $v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_m$ and called an arc-sequence from v_0 to v_m . Analogously, directed path (dipath) directed chain (dichain) can be defined for example in fig.3.2.6 says $z \rightarrow w \rightarrow v \rightarrow w \rightarrow u$ is a dipath.

A digraph D is connected if the underlying graph of D is a connected graph. Suppose in addition, if for any two vertices v and w of D there is a directed chain from v to w then D is strongly-connected. So every strongly connected digraph is connected but all connected graphs are not strongly connected. For example, consider fig-3.2.6 This is a connected digraph which is not strongly connected because there is not a chain from v to z .

Consider the road map of a city, where all the roads are one way. Then the map is connected, if one can drive from any part to another ignoring the road directions. The map is strongly connected if he can drive from any point to anywhere, always going the right way

If v is a node of a digraph D , the out-degree, of v is the number of arcs of D of the form (v,w) . Similarly, the in-degree of v is the number of arcs of D of the form (w,v) . Then, it follows that the sum of the indegrees of all the nodes of D is equal to the sum of their outdegrees.

3.3 DEFINITIONS AND THEOREMS FOR THE BOUNDARY DETECTION ALGORITHM (1)

3.3.A THE TWO DIMENSIONAL CASE

In two-dimensional display the universe is a picture, V , defined as a set of pixels

$$V = \left\{ v \mid v = (v_1, v_2), \text{ where for } 1 \leq j \leq 2, v_j \text{ is an integer and } 0 \leq v_j \leq b_j \right\}$$

v_1 and v_2 are the first and second components of v . The pixels v , for which $v_j = 0$ and $v_j = b_j$ are called border pixels, and the rest of the pixels are called the interior pixels. The set of interior pixels are denoted by V^* .

$$V^* = \left\{ v \mid v \in V \quad 1 \leq j \leq 2, 0 < v_j < b_j \right\}$$

we assume that, for $1 \leq j \leq 2$ and so V is nonempty. Predicates

$$f_{1,s}(v, v') \Leftrightarrow [v_i = v'_i \text{ if } i \neq 1 \text{ and } v_1 = v'_1 + s]$$

$$v \sim v' \Leftrightarrow [f_{1,s}(v, v') \text{ for some } 1, 1 \leq 1 \leq 2 \text{ and } s \in \{-1, 1\}]$$

Hence, for every interior pixel v there are four pixels v' such that $v \sim v'$. These are sometimes referred to as 4-neighbours of v , and the relationship is referred to as 4-adjacency (17).

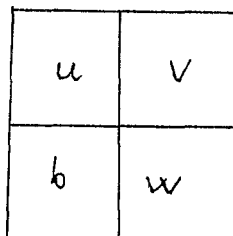


Fig.3.3.1 Four pixels in a two dimensional picture

The notion of 4-adjacency is extended to sets of pixels. Any two sets of pixels B and W are said to be adjacent if there is a pixel b in B and w in W such that $b \approx w$. (see fig.3.3.1). In the figure $b \approx u$, $b \approx w$.

Another predicate, \sim is defined by

$$v \sim v' \Leftrightarrow [v \approx v' \text{ or } |v_i - v'_i| = 1 \text{ for } 1 \leq i \leq 2]$$

So for any interior pixel there are eight pixels v' such that $v \sim v'$. They are sometimes referred to 8-neighbours of v . The relationship is referred to as 8-adjacency (17). In fig.3.3.1 $b \sim w$, $b \sim v$, $b \sim u$.

DEFINITION : A sequence $v^1 \dots v^k$ in V is said to be a path (resp. path) from $v^1 \dots v^k$, if for $1 \leq i \leq k$, $v^{i-1} \sim v^i$ (resp. $v^{i-1} \approx v^i$).

It is seen that any sequence of pixels of length 1 are both a path and a path.

In the following sections, and nonempty subset of V^* is an interior subset and any subset of V which contains border pixels, but not all elements of V is referred to as an exterior subset. If "Q" is an interior subset, then its complement (\bar{Q} i.e. $V-Q$) is an exterior subset. The term component is used for both interior and exterior subsets. In the case of medical application, the components of a given interior subset "Q" is referred to as "organs of Q" and components of the exterior subset \bar{Q} are called "organs of \bar{Q} " (is pronounced as "complement of Q", "organ" as co-organ, and a path as a co-path)

DEFINITION: Let Q be an interior subset and S be any subset of V . S is called an organ (resp., organ) of Q , if

- (i) $S \subset Q$ (resp. $S \subset \bar{Q}$),
- (ii) if v and v' are in S , there is a path in Q , (resp., a path in \bar{Q}) from v to v' ,
- (iii) S is not a proper subset of any subset of V which satisfies (i) and (ii).

LEMMA 1. Let Q be any interior set of V . If $v \in Q$ (resp., $v \in \overline{Q}$) there is one and only one organ (resp. $\overline{\text{organ}}$) of Q which contains v .

DEFINITION . Let Q be any interior subset of V and let S be any subset of V . Then S surrounds an organ (resp., $\overline{\text{organ}}$) O of Q if every path (resp., $\overline{\text{path}}$) from an element O to an element of the border $V-V$ intersects S .

An important result of this definition is as follows

LEMMA.2 Let Q be an interior subset of V . if B is an organ of Q and W is an $\overline{\text{organ}}$ of Q such that B and W are adjacent, then either B surrounds W or W surrounds B , but not both.

As a result, the boundary between an organ B and an adjacent $\overline{\text{organ}}$ W becomes clear. It is a set of edges separating B from W , with an "inside" and "outside" where one of B and W is entirely inside and the other is entirely outside the boundary. The definitions below are related to the above discussion.

DEFINITION . Any ordered pair of pixels (b,w) such that b and w are adjacent is called an edge. Any nonempty set of edges is called a 1-chain.

DEFINITION: Let B and W be any two adjacent subset of V . The boundary $P(B,W)$ between B and W is defined to be 1-chain.

$$P(B,W) = \{(b,w) \mid b \in B, w \in W, b \approx w\}$$

DEFINITION. Let b,u,v , and w be pixels such that $b \approx u$, $v \approx w$, $u \approx w$ and $w \approx b$, Then the four pixels define a vertex and this vertex is incident to one of the edges $(b,u), (u,b), (u,v), (v,w), (w,v), (w,b)$ and (b,w) .

DEFINITION. A 1-chain C is said to be a 1-cycle, if the following two conditions are satisfied:

(i) Whenever an edge (b,w) is in C , then the edge (w,b) is not in C .

(ii) Every vertex is incident to an even number of edges. If every vertex is incident to either no or exactly two edges in a 1-cycle C than C is said to be nonsingular; else C is said to be singular.

Nonsingular 1-cycles have an important property. They are Jordan curves in the following sense.

DEFINITION- A 1-chain C is said to be a Jordan Curve if and only if there exists an interior subset I (the interior of C) and an exterior subset E (the exterior of C) satisfying the following conditions.

$$(i) E = \bar{I}$$

$$(ii) C = P(I, E)$$

(iii) I is an organ of I and E is an organ of I .

The third condition in this definition says that two elements of I (resp. \bar{I}) are connected by a path (resp. a path). The reason for the name is that Jordan curves satisfy the analog of the Jordan curve theorem.

LEMMA 3. Let C be a Jordan curve, to let b and w be pixels in its interior I and exterior E respectively, and let v^1, \dots, v^k be a path from b to w . Then there exists an i , $1 < i \leq k$, such that $(v^{i-1}, v^i) \in C$.

DEFINITION. Let b, w and v be in V , such that $b \approx w$. The ordered pair (b, v) is a vertex of the edge (b, w) is $b \approx v$, and $w \approx v$. Clearly every edge has two vertices. As will be described later, a direction for edge following should be defined. This is accomplished by introducing the notion of an outgoing vertex, which is based on the predicates $f_{l,s}$.

DEFINITION. Let b, w and v be in V , such that $f_{l,s}(b, w)$ and (b, v) is a vertex of (b, w) we say that (b, v) is the outgoing vertex of (b, w) if

$$v_i = w_i \quad (-1)^l s, \quad \text{if } i \neq l,$$

$$= w_i, \quad \text{other wise}$$

for example in fig.3.3.1, $f_{1,1}(b, w)$ and so for the outgoing vertex (b, v) , $v = (w_1, w_2^{-1})$ as shown on the figure.

Intuitively this definition means that to find the outgoing vertex of an edge (b,w) , one has to stand in b , look towards w , and then the outgoing vertex is on the left.

DEFINITION. Let Q be an interior subset of V , let (b,w) be in $P(Q,\bar{Q})$ and let (b,v) be the outgoing vertex of the edge (b,w) . Let u ($u \neq w$) be the element of V such that $b \preceq u$ and $v \preceq u$ (see fig.3.3.1). Then, the element p in $P(Q,\bar{Q})$ is said to be adjacent to (b,w) if and only if one of the following holds.

- (i) $v \in \bar{Q}$, $u \in \bar{Q}$ and p is (b,u)
- (ii) $v \in \bar{Q}$, $u \in Q$ and p is (u,v)
- (iii) $v \in Q$, and p is (v,w) .

As a result, for any (b,w) in $P(Q,\bar{Q})$ there is one and only one p in $P(Q,\bar{Q})$ which is adjacent to it.

DEFINITION. Let Q be an interior subset of V . The boundary digraph $G(Q)$ of Q is the digraph $\{P(Q,\bar{Q}), E\}$, where $E = \{(p_1, p_2) \mid p_1, p_2 \in P(Q,\bar{Q}) \text{ and } p_2 \text{ is adjacent to } p_1\}$.

3.3.b THREE DIMENSIONAL CASE

The notions necessary to deal with 3-dimensional boundary detection is stated in this section. The previous section is followed closely not repeating the definitions where the three-dimensional case is closely analogous to the two-dimensional case.

The universe is a scene V , which is defined as a set of voxels,

$$V = \left\{ v \mid v = (v_1, v_2, v_3), \right. \\ \left. \text{where, for } 1 \leq j \leq 3, v_j \text{ is an integer and } 0 \leq v_j \leq b_j \right\}$$

Border voxels, interior voxels and the set V of interior voxels are defined in an analogous fashion to the two dimensional case.

The definitions of $f_{L,s}$ ($1 \leq L \leq 3, s \in \{-1, 1\}$) and \bar{v} are analogous to the two dimensional case but has to be redefined as

$$v \sim v' \Leftrightarrow \left[v \approx v' \text{ or ,for some } l, 1 \leq l \leq 3, v_l = v'_l \text{ and } |v_i - v'_i| = 1 \text{ if } i \neq l \right]$$

Geometrically, this means that 4-adjacency is replaced by face -adjacency, 8-adjacency is replaced by edge adjacency, and vertex adjacency is not needed for this subject (See fig.3.3.2)

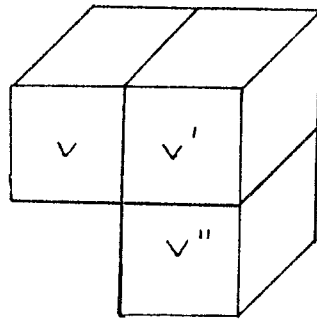


Fig. 3.3.2 v is face adjacent to v' ($v \approx v'$)
 v is edge adjacent to v'' ($v \sim v''$)

Interior subsets, exterior subsets, adjacency between subsets, paths, paths, organs and organs are defined exactly as in the two dimensional case. However the notion of "edge" and "1-chain" have to be replaced by the notions of "face" and "2-chain".

DEFINITION. Any ordered pair of voxels (b,w) such that b w is called a face. Any nonempty set of faces is called a 2-chain. The boundary between two adjacent subsets B and W of V is the 2-chain

$$P(B,W) = \{ (b,w) \mid b \in B, w \in W, b \approx w \}$$

The definitions of 1-cycle and Jordan curve can now be replaced by that of a 2-cycle and a Jordan surface (as a 2-chain satisfying conditions (i),(ii),(iii) listed there).

DEFINITION. Let b , w and v be in V , such that $b \approx w$. The ordered pair (b,v) is said to be an edge of the face (b,w) if $b \sim v$ and $w \approx v$ (see fig.3.3.3).

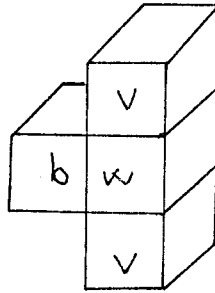


Fig.3.3.3 The two possible edges of face (b,w) are shown.

Clearly, each voxel has six faces. They are determined by the six possible values of the pair (l,s) , $1 \leq l \leq 3$ and $s \in \{-1,1\}$. Each face has four edges. For reasons to be explained in boundary detection algorithm two of the edges are distinguished and called right edge and left edge (The words do not indicate that the edges are physically on the left or right of anything.)

The definitions are as follows:

Suppose that $f_{l,s}(v,v')$ and (v,v'') is an edge of the face

(v,v') . Then (v,v'') is the left edge of (v,v') if

$$\begin{aligned} v''_i &= v'_i + s \quad \text{if } i = 1 \bmod (3) + 1 \\ &= v'_i, \quad \text{otherwise.} \end{aligned}$$

(v,v'') is the right edge of (v,v') if

$$\begin{aligned} v''_i &= v'_i - s \quad \text{if } i = (1+1) \bmod (3) + 1 \\ &= v'_i, \quad \text{otherwise} \end{aligned}$$

In (Fig 3.3.4) consider v, v' the u_2 coordinates of the two voxels are different so $l=2, v'_2$ is one unit less than v_2 ,

therefore $s=-1$ and $f_{2,-1}(v,v')$

Since $v''_1=v'_1+1$, $v''_2=v'_2$ and $v''_3=v'_3$

(v,v'') is the right edge of (v,v') .

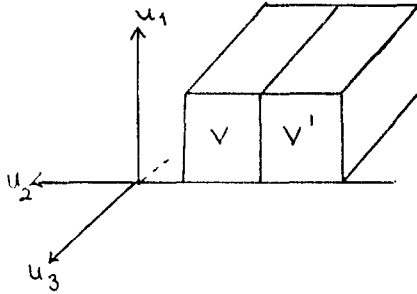


Fig.3.3.4 The predicate describing (v,v') is $f_{2,-1}$

To find the left edge (v,v'') find

$$i = 2 \bmod 3 + 1 = 3$$

$$s = -1$$

therefore

$$v''''_1 = v'_1$$

$$v''''_2 = v'_2$$

$$v''''_3 = v'_3 + (-1) = v'_3 - 1$$

v'''' is shown in fig 3.3.5

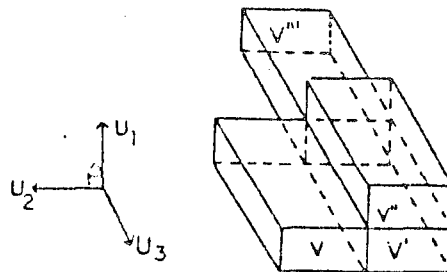


Fig. 3.3.5 The pair (v,v') is a face of the voxel v . The pair (v,v'') is the right edge and the pair (v,v''') is the left edge of the face (v,v') .

There are 12 distinct edges of the six faces of a voxel. When checked, it is seen that each of these 12 edges is the left or right edge of some face of the voxel.

Let Q be an interior subset of V , let (b,w) be in $P(Q, \bar{Q})$ and let (b,v) be the left (respectively, right) edge of (b,w) . Let u ($u \neq w$) be such that $b \approx u$ and $v \approx u$. (see fig.3.3.6)

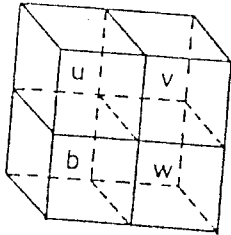


Fig.3.3.6 Four voxels

Then the face p in $P(Q, \bar{Q})$ is left (respectively right) adjacent to (b,w) if and only if the following holds:

- (i) $v \in \bar{Q}, u \in \bar{Q}$ and p is (b,u)
- (ii) $v \in \bar{Q}, u \in Q$ and p is (u,v)
- (iii) $v \in Q$ and p is (v,w)

p is adjacent to (b,w) , if it is either left or right adjacent to (b,w) .

With this new definition of "adjacent to" the definition of boundary digraph $G(Q)$ is exactly the same as in the two dimensional case. However, in the two dimensional case every node of $G(Q)$ had outdegree 1, while in the three dimensional case every node of $G(Q)$ has outdegree 2. (see figs.4.1.1 and 4.4.2)

IV. BOUNDARY DETECTION

4.1 AN INTUITIVE DESCRIPTION OF BOUNDARY DETECTION ALGORITHM

Before stating the algorithm an informal description would be more descriptive. As mentioned before, a subset Q of the voxels in the three-dimensional scene is assumed to be specified, and the problem is the detection of the boundary of an object in the scene. The output of one algorithm is a list of faces on the boundary. Consider a single voxel. (See fig. 4.1.1a) For any one of its faces, two of the four edges are the outgoing edges. The other two are the incoming edges. When done properly, this procedure covers 6 faces and the 12 edges of a cube, where each edge will be an outgoing edge of one face and an incoming edge of an adjacent face. If each face is viewed as node and each edge as an arc, the boundary will be represented as a digraph (directed graph) in which each node has indegree two, (corresponding to the incoming edges) and outdegree two (corresponding to the out-going edges) (see fig. 4.1.1b)

Surfaces of objects consists of faces of voxels. In the boundary detection procedure, it is shown that the notion of a digraph whose nodes are faces and whose arcs are edges can be generalized from a single voxel to arbitrary scenes. As a result, a "connected" part of the surface of a object gives rise to a connected component of an digraph and vice versa. Hence, given one of its elements, finding a boundary surface is traversing a connected component of a digraph. The algorithm makes use of the fact that every node in the digraph has outdegree two and indegree two.

Every connected component of such a digraph is strongly connected. This and the fact that every node has outdegree two implies that for every node there is a binary tree rooted at the node. This node is a subgraph of the digraph, and spans the connected component which contains the given node. So a procedure for traversing binary spanning trees can be used. However this introduces difficulty because of the size

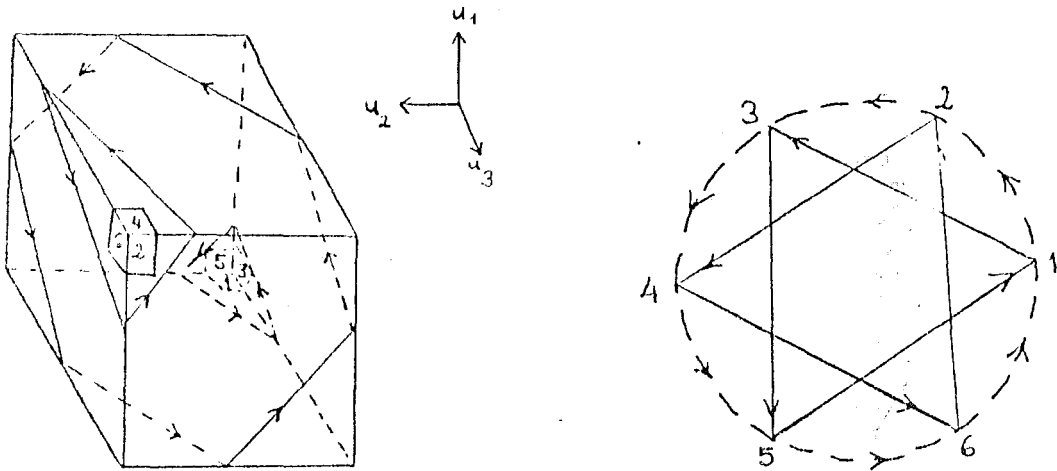


Fig.4.1.1a A voxel with faces marked by numbers 1-6. There are also 12 arrows: the outgoing edges are the ones on which arrows terminate, the incoming edges are those on which arrows originate. Every face has two outgoing and two incoming edges, and every edge is the outgoing edge of one face and the incoming edge of another face. b) The digraph associated with. a) The correspond to the faces of the voxels. The continuous arrows corresponds to left adjacency, the broken arrows corresponds to right adjacency.

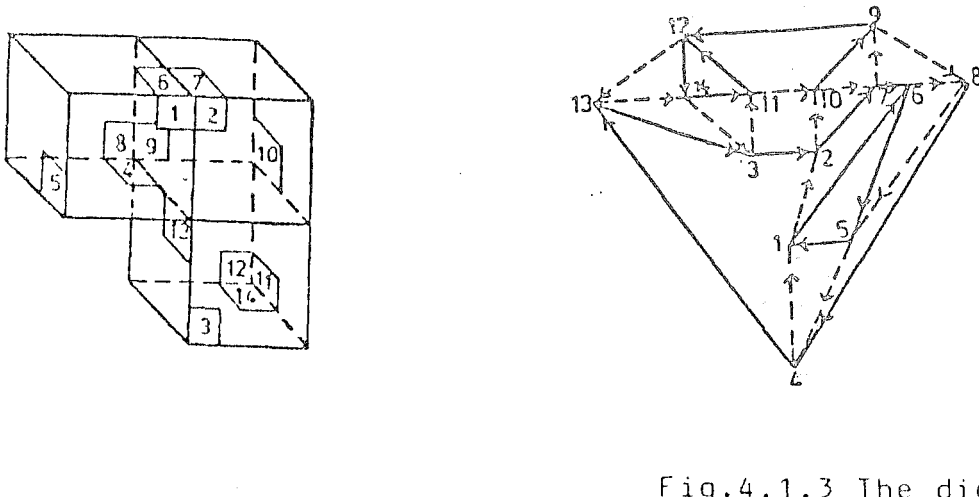


Fig.4.1.2 An organ of three voxels

Fig.4.1.3 The digraph of the organ in fig.4.1.2

of the problem. A binary spanning tree exists but, the information that is stored is the original picture. The spanning tree gets specified during the graph traversal process.

A general binary spanning tree traversal algorithm has to "mark" all the nodes which it had visited. Whenever an arc leads from a visited node to a "new", node this "new" node has to be checked against a list of marked nodes, so that infinite loops are avoided. Keeping the number of those nodes low is very important. This is done because each node has outdegree two and indegree two and because the digraph is traversed directly. Before stating the algorithm precisely an example would illustrate the procedures and techniques used.

Fig.4.1.2 shows a body made up of three voxels. Each face is identified by a number. A boundary digraph is created by marking each edge on the body as an incoming or outgoing edge (see fig.4.1.3) Each node represents a face and each arc represents a connection between one face and one of its neighbors. As seen from the figure each node of the digraph is both of indegree two and out degree two. (See also that for each node one of the outgoing arcs is "left adjacent" while the other one is "right adjacent". When a node is visited it is checked against a list of marked nodes. If it is not on the list, it must be marked (added onto the list). If it is on the list, it is removed since it will never be visited again. This way the list of marked nodes are kept to a computationally acceptable size. In the example given by Artzy, Frieder and Herman (1). In a 180x128x52 scene, the total number of voxels in the input three dimensional scene is 1,198,080. The number of voxels within the organ of interest is 153,547, and the number of faces enclosing the organ (i.e. the number of nodes of the connected subgraph. 112,130. The list of marked nodes, though, never exceeds 1744 which is less than 1.6% of the total nodes considered.

Table 4.1.1 below, shows the step by step result of the algorithm for the organ made up of three voxels.

ode	Queue of nodes to be considered	List of marked	Output list
	(1)	(1,1)	(1)
1	(6,2)	(1,1,6,2)	(1,6,2)
6	(2,5,8)	(1,1,6,2,5,8)	(1,6,2,5,8)
2	(5,8,7,10)	(1,1,6,2,5,8,8,10)	(1,6,2,5,8,7,10)
5	(8,7,10,4)	(1,6,2,5,8,7,10,4)	(1,6,2,5,8,7,10,4)
8	(7,10,4)	(1,6,2,8,7,10)	(1,6,2,5,8,7,10,4)
7	(10,4,9)	(1,2,8,7,10,9)	(1,6,2,5,8,7,10,4,9)
10	(4,9)	(1,2,8,10)	(1,6,2,5,8,7,10,4,9)
4	(9,13)	(2,8,10,13)	(1,6,2,5,8,7,10,4,9,13)
9	(13,12)	(2,10,13,12)	(1,6,2,5,8,7,10,4,9,13,12)
13	(12,3,14)	(2,10,13,12,3,14)	(1,6,2,5,8,7,10,4,9,13,12,3,14)
12	(3,14)	(2,10,13,12,3)	(1,6,2,5,8,7,10,4,9,13,12,3,14)
3	(14,11)	(10,12,3,11)	(1,6,2,5,8,7,10,4,9,13,12,3,14,11)
14	(11)	(10,12)	(1,6,2,5,8,7,10,4,9,13,12,3,14,11)
11	\emptyset	\emptyset	(1,6,2,5,8,7,10,4,9,13,12,3,14,11)

Table 4.1.1 Step by step results of the boundary detection algorithm for the organ in fig.4.1.2

The FIFO queue discipline is used to select the next node from which the traversal should continue.

In the traversal procedure, the node to be started from is arbitrary. One of the outgoing edges is followed. The other node that is neglected is remembered by queuing it. The nodes visited are marked by storing them on a list.

Looking at the digraph and the table, start with any node. In this case, it is node "1". Put node "1" node on the queue, and on the list of marked nodes and on the output list. Find the outgoing edges from the digraph on fig 4.1.3. They are "2" and "6". Since "2" and "6" are not visited before, they will be put on the list of marked nodes and the output list. Then get a node from the queue. It is "6". Find the outgoing edges. They are "5" and "8". Check them on the list of marked nodes. If they do not exist there, add them to the list of marked nodes and output list. The next node is "2", the

outgoing edges are "7" and "10". The next node is "5". The outgoing edges are "1" and "4". Since node "1" is already on the marked list, it is not put onto the output list, and also since it will be never visited again "1" is removed from the list of marked nodes. This goes on until the set of nodes to be considered is empty.

In the following sections the algorithms for two and three dimensional boundary detection are stated.

4.2 TWO DIMENSIONAL BOUNDARY DETECTION

Two dimensional boundary detection is simpler than three dimensional case. The introduction of 2D boundary detection first is that some of the intuitive ideas can be explained easier in the simpler case. Using the terminology and definitions introduced in the previous section the problem to be solved by the boundary detection algorithm is stated as follows.

The Boundary Detection Problem

Given: an interior subset Q , and a face (b,w) such that $b \in Q$ and $w \in \bar{Q}$

Find: $P(b,w)$ where B is the organ of Q containing b and W is the organ of \bar{Q} containing w .

In other words, the algorithm should find the boundary which is specified by one of its elements.

Having the definitions of vertex of an edge, outgoing vertex, and adjacent to, the boundary detection algorithm can be stated.

First we need the theorem stated below:

THEOREM 1. Let Q be an interior subset of V , B an organ of Q and w be an organ of \bar{Q} such that B and W are adjacent. Then $P(B,W)$ is a strong component of $G(Q)$

Algorithm 1.(1)

Input: an interior subset Q and an element (b,w) of $P(Q,Q)$

- 1.(Initialize) put (b,w) on the "output" list and let p be the element of $P(Q,Q)$ which is adjacent to (b,w) .
- 2.(main body) while p (b,w) do 3-4.
- 3.(output) add p to the "output list".
- 4.(Get next boundry element) replace p by the element of $P(Q,Q)$ which is adjacent to it.
- 5.(Terminate) END.

CORROLARY 1. For any interior subset Q and for any element (b,w) of $P(Q,\bar{Q})$, Algorithm 1 terminates. At the time of termination the output list contains exactly the elements of $P(B,W)$ B is the organ of Q containing b , and W is the organ of \bar{Q} containing w . (in other words algorithm solves the boundary detection problem).

Proof of Carollary 1. Theorem 1 says that $P(B,W)$ is a strong compoment of $G(Q)$. Hence starting from any node (b,w) in $P(P,W)$ we get all the other nodes in $P(B,W)$ using the "adjacent to" relationship. However for any node there is one and only one node adjacent to it, and so $P(B,W)$ must be a cycle . Obviously,algorithm 1 output all nodes of this cycle exactly once.

4.3 THREE DIMENSIONAL BOUNDARY DETECTION ALGORITHM

The three dimensional boundary detection algorithm makes use of the theorems and 3 stated below. Let's first state the algorithm.

Algorithm 2.(1)

Input: An interior subset Q and an element (b,w) of $P(Q,\bar{Q})$

Auxiliary data structures: a queue X and alist M .

1. initialize: put (b,w) on the "output list" and on the

- queue x , and put two copies of (b,w) on the list M .
2. Main body: while x is nonempty, do 3-7.
 3. Get face: remove element p from X .
 4. Left adjacency: Calculate the face q left adjacent to p .
 5. Left marking: if q is on M
 - then remove q from M
 - else put q on "output list", on X and on M .
 6. Right adjacency: calculate the face q right adjacent to p .
 7. Right marking: If q is on M ,
 - then remove q from M
 - else put q on "output list", on X and on M .
 8. Terminate: END.

THEOREM 2. Let Q be an interior subset of V , B be an organ of Q , and w be an organ of \bar{Q} such that B and w are adjacent. Then $P(B,W)$ is a strong component of $G(Q)$.

THEOREM 3. Let Q be an interior subset of V . Then every node of $G(Q)$ has indegree 2.

COROLLARY 2. For any interior subset Q and for any element (b,w) of $P(Q,\bar{Q})$, algorithm 2 terminates. At the time of termination "the output list contains exactly the elements of $P(B,W)$ where B is the organ of Q containing b and W is the organ of \bar{Q} containing w . (i.e. the algorithm solves the three dimensional boundary detection problem).

V. HIDDEN SURFACE REMOVAL

To produce a realistic representation of the surface of a three dimensional object on a screen, one has to remove the hidden parts of the surface and shade the visible parts.

The method used for hidden surface removal makes use of the properties of the surface elements. They are all squares of the same size, lie in one of three mutually perpendicular directions, and are small. The general algorithms for hidden surface removal apply to surfaces made up from more complicated elements. The algorithm, which makes use of the special properties of the surface elements, to be used in the 3-D display of CT data produces as good a result as the more general algorithms.(2)

To describe the surface elements for this application, the notion of "cuberille" is introduced.

DEFINITION. A cuberille S is a pair $\{\bar{\Phi}, C\}$ where
 (i) $\bar{\Phi}$ is an infinite collection of closed cubes called the components of $\bar{\Phi}$. These cubes cover the whole of three dimensional space and are such that if two components of intersect then the intersection is a face of both components, and (ii) C is a mapping which associates with every face the point int at the center of the face.

The motivation for the name "cuberille" is now explained "Quadrille" marking on paper is produced by two orthogonal sets of parallel lines that are equally spaced and dissect the surface of the paper into equal squares. Analogously, a cuberille is the dissection of space into equal Cubes by three orthogonal sets of equally spaced parallel planes. C is a mapping from a set of faces into the set of all points which are at the centers of the faces of the cubes. In the following paragraphs "plane of a face" means the infinite plane in which the face lies.

In this terminology, the voxels are components of a cuberille S and all the surface elements are faces of S . The following theorem greatly reduces the time for hidden surface removal.(2)

THEOREM - IF

- (i) $S = \{\Phi, C\}$ is a cuberille
- (ii) O is a point,
- (iii) α and β are two different faces of S , such that O does not lie in the plane of either α or β ,
- (iv) A is a point in the interior of α ,
- (v) B is a point in the interior of β ,
- (vi) O, A and B are collinear, with A and B on the same side of O , then
- (vii) $|OA| < |OB|$ if, and only if, $|OC(\alpha)| < |OC(\beta)|$

Following is the practical consequence of this theorem. Consider two surface elements α and β , such that O does not lie in the plane of either α or β , and point A in interior of α lies on the same line of sight from the observer O as the point B in the interior of β .

The theorem states that A hides B if, and only if, the center $C(\alpha)$ of the face α is nearer to the observer O than the center $C(\beta)$ of the face β . This theorem is not true for more complicated surface elements. It breaks down even when the surface elements are rectangles.

As a result, for a fixed observer position O , each 2D surface element needs to have only two numbers associated with it for shading and hidden surface removal:

- (i) $b(\alpha)$, the brightness of all points on the display surface. $b(\alpha)$ is calculated by the chosen shading rule, and
- (ii) $d(\alpha)$, distance of $C(\alpha)$ from the observer O .

The visible surface is calculated by the following simplified version of the z-buffer or brute force algorithm.

With each dot D on the display screen two numbers are associated: brightness $b(D)$ and distance from the observer $d(D)$. Initially, for each dot, brightness $b(D)$ is set to zero, and the distance from observer is set to a very large number (essentially. Whatever the orientation of the object is at most three of the six faces are potentially visible, which will be explained later. The visible faces making up the surface are displayed according to the following procedure.

For each face α , calculate $b(\alpha)$ and $d(\alpha)$, locate all the dots on the display screen, if that face were to be displayed alone. For each such a dot D , change $b(D)$ and $d(D)$ only if $d(\alpha)$ is less than $d(D)$. (i.e. α is closer than the depth of D). In that case, set $b(D)$ to $b(\alpha)$ and $d(D)$ to $d(\alpha)$. In this fashion, when all the dots on all the visible faces are considered, brightnesses of the visible faces will be displayed, and hidden faces will not influence the display.

VI. GEOMETRICAL TRANSFORMATIONS

The geometrical procedures applied, for the display of three dimensional data on a CRT, are discussed under this to pic. (18)

6.1 TRANSLATION

Points are translated to new positions by adding translation amounts to the coordinates of the point. The new position of a point direction, and by dify in y direction is given by. $x' = x + difx$ $y' = y + dify$ $z' = z + difz$

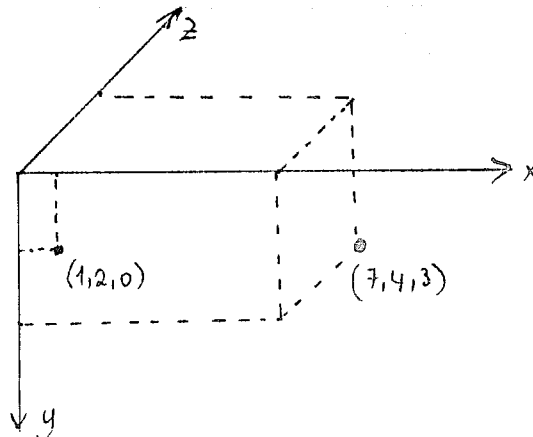


Fig.6.1.1 point (1,2,0) translated to (7,4,3)

In fig.6.1.1, a point (1,2,0) is translated by (7,4,3). The new position of the point is (8,6,3).

The translation matrix is, thus, obtained as

$$T = \begin{bmatrix} difx & dify & difz \end{bmatrix}$$

If point P is expressed as $P = \begin{bmatrix} x & y & z \end{bmatrix}$ then

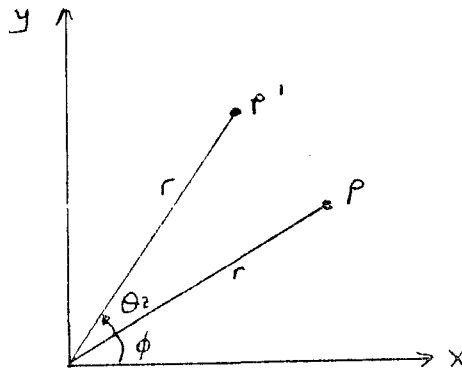
$$P' = P + T$$

6.2 SCALING

A point can be scaled by S_x in x , S_y in y and S_z in z . The scaling is done with respect to the origin, and if S_x, S_y , and S_z are not equal the proportions of the object changes. Since in medical imaging the objective is the production of the realistic pictures of organs for diagnosis and surgical planning, a uniform scaling should be done so that the proportions are not affected. Points can be rotated around the origin.

6.3 ROTATION

Suppose a rotation by θ_z around Z -axis transforms $P(x,y)$ into $P'(x',y')$. (See fig.6.3.1)



6.3.1 Point P rotated by θ_z around the origin to position P'

Because the point is rotated around the origin the distances from p to origin and from p' to origin are equal, (labeled as r). Using trigonometry

$$x' = r \cos(\theta + \phi) = r \cos \phi \cos \theta - r \sin \phi \sin \theta$$

$$y' = r \sin(\theta + \phi) = r \cos \phi \sin \theta + r \sin \phi \cos \theta$$

$$z' = z$$

and substituting $x = r \cos \phi$ and $y = r \sin \phi$ we get

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

$$z' = z$$

So the rotation matrix around z axis can be written as

$$R_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The objects to be displayed may not be rotated around the origin but around an arbitrary point P_1 . However the rotation matrices are obtained for rotations around the origin. The following steps should be carried out in order to solve the problem:

- Translate the scene, in order that P_1 coincides with the origin.
- Rotate the object around the origin.
- Translate back to the previous position.

In the implementation of 3D display algorithms of medical images two coordinate systems are defined. The coordinate system in which the voxels of the object are defined is the object space, and the coordinate system in which they are display is called the image space. Object space coordinates are integers, and the image coordinates are obtained by transformations, as rotation translation and scaling.

The object to be displayed is assumed to be made up to slices. Each slice is on the x-y plane, and each slice is added on in the z direction. The object is placed in relation to the x and y axis as shown on fig.1.3

To allow the display of any orientation of the object three angles must be specified. These are x^0 , y^0 , z^0 , and result in rotations about the three predetermined axis. The angles are chosen in such a way that they are practical for the medical user. Suppose, a patient is lying face up in a CT scanner, with feet toward the observer. The angle z^0 indicates swivel about the longitudinal axis (see fig.6.3.2) Angle x^0 indicates tilting in the x axis, while y^0 indicates vertical rotation of the person.

The transformations are applied in the order of first swivel than tilt than vertical rotation.

Given a voxel with coordinates (x,y,z) in object space, in order to obtain its coordinates in image space the following steps should be taken:

First carry the center of the object to the origin. if $(2x_c, 2y_c, 2z_c)$ are the dimensions of a bounding box enclosing the object, by subtracting (x_c, y_c, z_c) from the coordinates of each of the voxels, the object is translated, ready for rotation. After rotation the object is still at the origin, at the center of the screen. If $(2x_s, 2y_s)$ are the dimensions of the screen, and $2z_s$ are the number of gray levels, then by adding (x_s, y_s, z_s) to each pixel coordinate, the object is translated to the center of the screen.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = R \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} + \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}$$

where

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

R is the rotation matrix. See fig.6.3.2

Fig.6.3.2 Rotation Matrix

$$\begin{bmatrix} \cos y_0 + \sin z_0 \sin x_0 \sin y_0 & -\sin z_0 \cos y_0 + \cos z_0 \sin x_0 \sin y_0 & \cos x_0 \sin y_0 \\ \sin z_0 \cos x_0 & \cos z_0 \cos x_0 & -\sin x_0 \\ \sin y_0 + \sin z_0 \sin x_0 \cos y_0 & \sin z_0 \sin y_0 + \cos z_0 \sin x_0 \cos y_0 & \cos x_0 \cos y_0 \end{bmatrix}$$

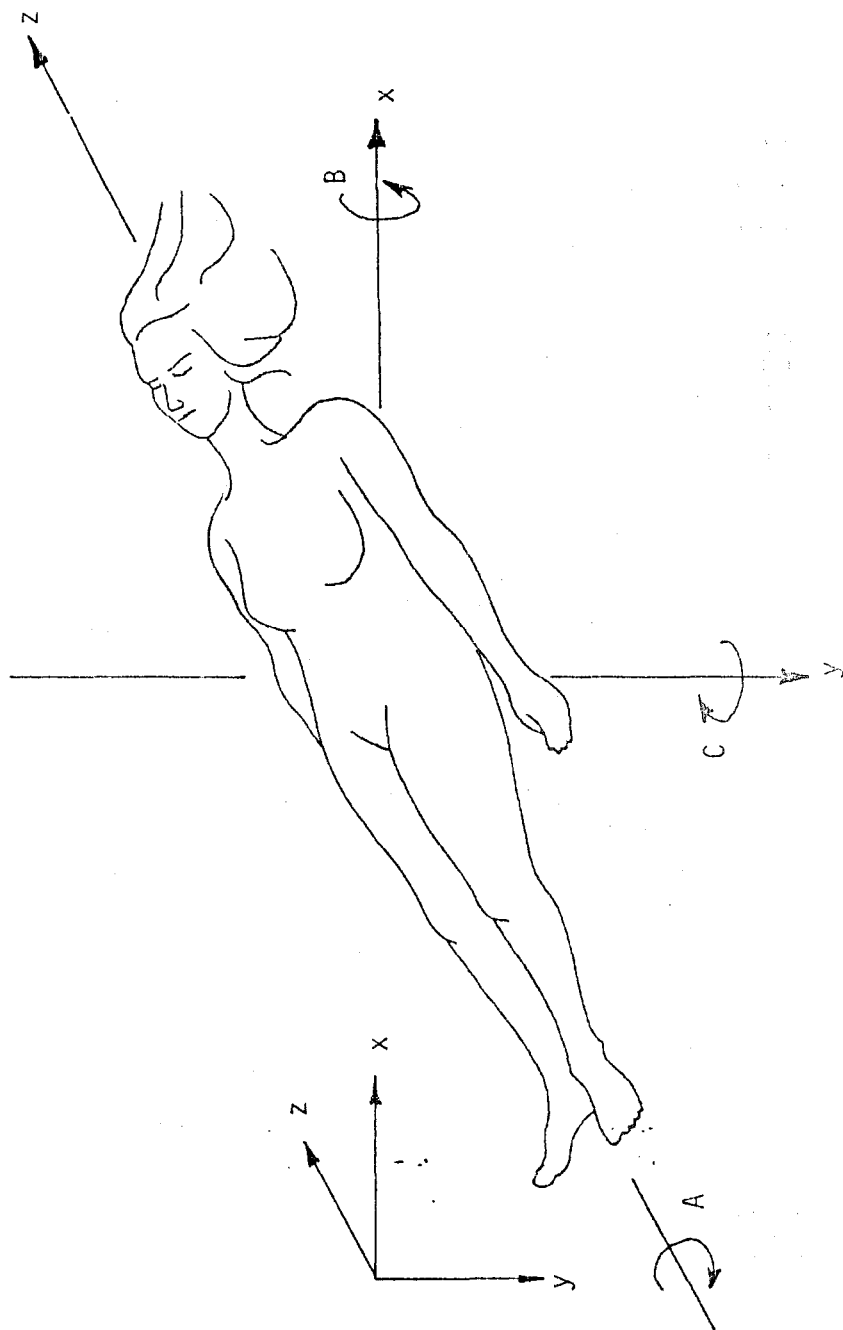


FIG.6.3.3 Angle z_0 (swivel), x_0 (tilt), y_0 (vertical rotation)

VII. PROJECTIONS

Viewing in three dimensions is not straight forward because the display device is only two dimensional. The problem is solved by projection, i.e. by transforming the three dimensional object onto a two dimensional display plane. In general projection involves a transformation from a n dimensional coordinate system to one of dimension less than n , in this case from 3D to 2D.

The projection of a 3D scene is (which is a collection of points) defined by straight projection rays which emanate from the center of projection and pass through each point of the object. These rays then intersect the projection plane to form the projection. See fig.7.1.1 (18,19)

7.1 PARALLEL AND PERSPECTIVE PROJECTIONS

Projections are divided into two classes: perspective and parallel. The position of center of projection with respect to projection plane distinguishes the two. If the distance between the two is finite, then the projection is perspective. If the distance between the two is infinite, the projection is parallel. Fig.7.1.1a and b illustrate the two projections.

If the center of projection is at infinity, the projectors are parallel. Therefore, such a projection is called parallel projection. In perspective projection while the center of projection is important, in parallel projection, the direction of projection is considered.

The perspective projection produces a picture similar to photographs, and human visual system. This is accomplished by perspective foreshortening: The size of the object to be displayed decreases by an amount proportional to the inverse of its distance from the center of projection. The images are realistic, but distances can not be taken from projections, and angles are only preserved on faces parallel to the

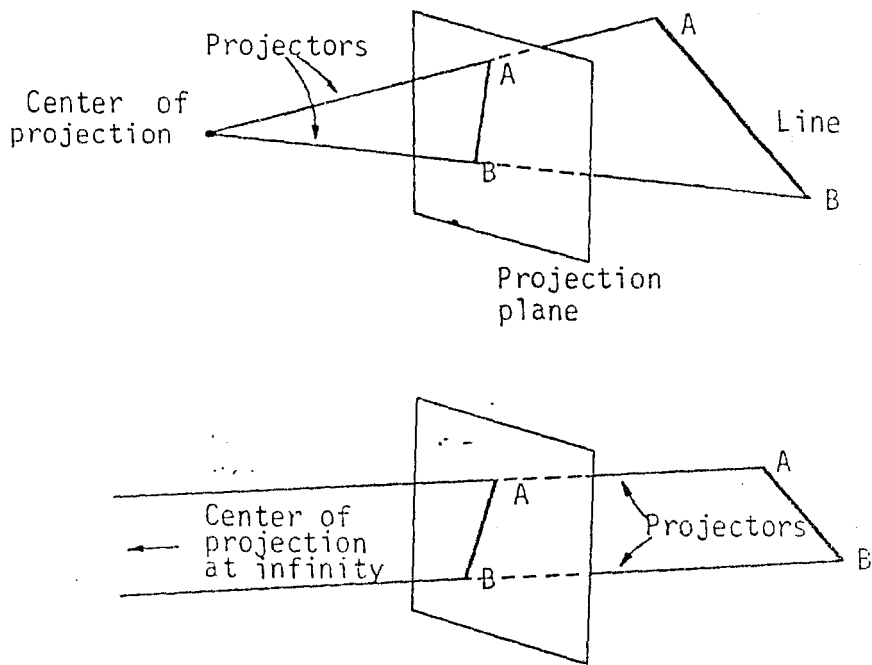


Fig.7.1.1 Parallel and perspective projections

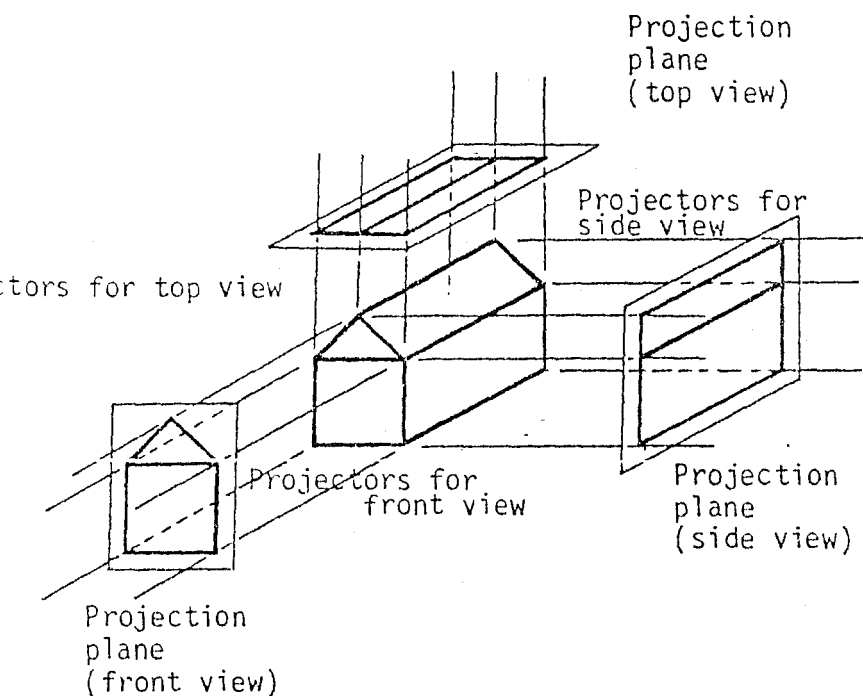


fig.7.2.1 Orthographic projection

projection plane, parallel lines do not remain parallel.

Parallel projection is not as realistic, because perspective foreshortening is not used. Parallel lines remain parallel. The projection records exact measurements.

7.2 ORTHOGRAPHIC AND AXONOMETRIC PARALLEL PROJECTIONS

There are two types of parallel projections, if the direction of projection and the normal of the projection plane are the same then the projection is called orthographic. If they are not equal then the projection is oblique.

In this application, orthographic projection are used because it preserves distances and other geometrical informations that are important.

The most common types of orthographic projections are top, front, and side views, as show in fig.7.2.1, where the projection plane is perpendicular to one of the principle axis.

In axonometric orthographic projection, the projection plane is not perpendicular to any one of the principle axis. So, several faces of the object can be shown, just as in the perspective projection, but the foreshortening is uniform instead of being related to inverse of the distance from the center of projection. Parallelism of lines are preserved but angles not. However, distance can be measured along each principle axis.

In the implementation of 3D display of organs the direction of projection is the same as the projection plane normal i.e. the z axis.
So, point p projects as

$$x_p = x, y_p = y, z_p = 0.$$

The projection matrix is

$$M_{\text{ort}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

After rotating the object, and carrying it to the center of the screen, the 3D voxel coordinates are converted to 2D coordinates.

VIII. SHADING

After hidden surface removal, the next step is to shade the visible surface of the object.

In shading, the light sources, surface characteristics, and the position of the sources and surface should be taken into account.

8.1 DIFFUSE AND SPECULAR REFLECTIONS

The light reflected off a surface can be broken down into two components: diffuse and specular. Dull matte surfaces exhibit diffuse reflection. They scatter light equally in all directions. For shading of such surfaces, Lambert's cosine law is used. The reflected light is related to the cosine of the angle between the direction of light and the normal vector of the surface, as shown on fig.8.1.1. That is, the amount of reflected light seen by the viewer is independent of the viewer's position. The diffuse illumination is:

$$I_d = I_p k_d \cos \theta.$$

I_p is the intensity of the light source. k_d is a constant from 0 to 1, and varies from one material to another.

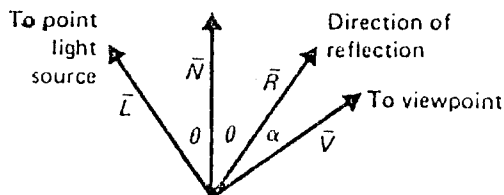


Fig.8.1.1 Direction of source of light, the normal of the surface. Diffuse and specular reflections

In the implementation used in this project, the assumption that light travels in parallel rays perpendicular to the screen from a plane source which is perpendicular to the screen is made. So, two voxels whose visible faces have the same orientation will seem to be on the same plane even though one may be further behind the other. So a depth cue is included, because light energy falls off as distance from source increase.

So the shading formula becomes (20)

$$S = \frac{I_d k_p \cos \theta}{R}$$

where R is a measure of distance from observer.

Specular reflection B not included in the shading rule used in this project. Specular surfaces reradiate light in one direction, the reflected light direction. Examples of specular surfaces are mirrors and shiny surfaces on which highlights are visible. The difference between specular and diffuse reflection physically is that in specular case, the light bounces off the surface, while in diffuse case, the light is absorbed and scattered internally before going out again. Specular component is not easy to calculate. In fig. only if angle is zero, the specular reflection is seen. The experimental approximation is introduced by Phong, and is in the form

$$W(\theta) \cos^n(\alpha)$$

$W(\theta)$ is included, because the amount of light that is specularly reflected depends on the angle of incidence, θ . The value n is the shininess factor. As n increases I_s dies off more quickly for the off axis directions. Thus a shiny surface with a concentrated highlight would have a large value of n , while a dull surface with the highlight covering a larger area on the surface will have a low value of n .

8.2 THE SHADING RULE USED IN THIS THESIS

Computationally using the formula for diffuse reflection is very appropriate for medical applications. Since the possible orientations of surface elements are only three, at most three cosine values must be calculated for the display. Also, the distance from observer need not be calculated because it is inherent in the depth array of the hidden surface removal algorithm.

8.3 SHADING USING CT NUMBERS

There are numerous different methods for shading and antialiasing. Yet a method developed for shading 3D images from CT using gray level gradient is quite interesting.

The gray value of a voxel represents the mean density within a volume element. If, for example, the density changes within a voxel from bone to tissue, the corresponding value is lower than that of bone. This effect is known as the partial volume effect. Although it deteriorates the images, K.H.Hohne and R. Bernstein (21) show that this effect can be used to gain information about the surface structure. If it is assumed that the surface is due to a border between two regions of basically uniform material, the gray level is a measure of the ratio of the materials in the voxel. No information on the surface direction can be obtained from the gray level gradients in the neighborhood of the surface voxels, as shown in fig.8.3.1

Experimentally, K.H.Hohne and R.Bernstein derived a shading rule, where the projected intensity in the image plane is computed as

$$I(i,j) = A \cos\left(\frac{g(i,j+1,k_T) - g(i,j-1,k_T)}{B}\right)$$

where $g(i, j, k_T(i, j))$ are the gray values at depth $k_T(i, j)$ at which the surface threshold is encountered. A and B are scale factors depending on the gray level range of the original images (typically $A=255$. $B \approx 100$, for 8 bit gray-level resolution). The cosine is used to achieve diffuse reflection as explained before. The pictures they produced can found in (21).

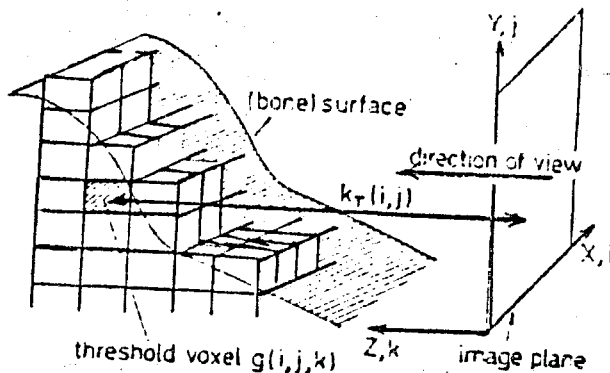


Fig.8.3.1 Voxels of a bone surface

The aliasing effect is reduced by taking 3×3 neighborhood above and below the surface voxel. Addition of depth shading improves the image. The resulting intensity is given by,

$$I(i, j) = A \cos \left(\frac{1}{9} \sum_{l=-1}^{+1} \sum_{m=-1}^{+1} \frac{(g(i+l, j+s, k_T+m) - g(i+l, j-l, k_T+m))}{B} \right) - C \cdot k_T$$

where C describes the fraction of depth shading. The computing time for this shading is comparable to depth shading. Yet, the method requires the access of full gray level data. So at least 8 or more times as much storage space is required.

8.4 SMOOTHING (2)

However, since the orientations of the detected surface are limited, the displayed image looks jagged although the organ is smooth. Either one or a combination of the techniques proposed to overcome this problem can be used. These techniques are

- i) Put the final image through a low-pass filter in order to smooth its appearance.
- ii) Make the influence of the angle between the light and surface normal unimportant as compared to the z distance.
- iii) Assume a virtual display screen of higher resolution than the actual screen and display on the actual screen and display on the actual screen by averaging the values on the virtual screen.

Smoothing reduces the artifact using little computer time. Low pass filters are used to remove noise and sharp boundaries in images. Let $G_1 \dots G_9$, denote the shading intensities in a screen dot and its eight neighbors before smoothing as shown in fig.8.4.1 below.

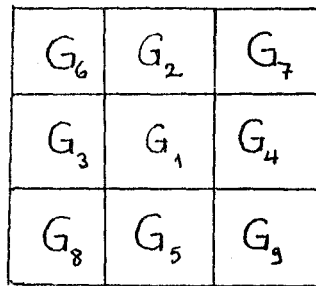


Fig.8.4.1 The eight neighbours of G_1 used in low pass filtering

After smoothing the new intensity of the center dot will be

$$G_1^{\text{new}} = \frac{(G_1 - W \sum_{i=2}^5 f_i G_i - W \sum_{i=6}^9 f_i G_i)}{1 - W \sum_{i=2}^5 f_i - W^2 \sum_{i=6}^9 f_i}$$

If a dot labeled l is on the edge of the screen and therefore does not have a neighbor labeled i , then f_i is assumed to be zero, otherwise f_i equals one. W is a used adjustable weighting factor, usually between 0 and 1. Repeated applications of such a smoothing procedure results in further improvements.

To make the effect of angle of incidence small compared to depth. The use of the cosine of half the angle of incidence is a good compromise.(22)

IX. IMPLEMENTATION OF THE 3D DISPLAY ALGORITHM

9.1 Hardware environment:

The 3D display algorithm has been implemented on an IBM PC with 8 512 KBytes of core memory, a 5 1/4" disk drive and a 40 M Bytes of hard disk. An inhouse built display terminal is connected to the IBM PC via a parallel port. The terminal can display 256x256 images with 256 gray levels. A color graphics adapter is also available for user-computer interaction.

9.2 Software environment

The operating system used is MS DOS (Version 2.) All the software has been written using Pascal Turbo (Version 3.0)

9.3 Implementation

9.3.1 User Interaction and I/O: The user should supply the program with a set of data which is then used at different stages. The following information should be supplied:

- slice/matrix size (i.e. 64x64, 128x128, or 160x160)
- number of slices
- pixel size
- slice thickness
- CT number range
- Organ bounding box
- x,y,z rotation angles.
- grid resolution.

The following sections describe both the data entry procedures and the function of each program module. The program THRESH is of COM-node (i.e. compile) and will ask the operator the necessary questions for each piece of information in an interactive way. Fig.9.1.1 shows a flow diagram of the entire software package.

It is assumed that the CT data are given as uncompressed and as a file of integer. Every CT manufacturer uses a different file structure, writing format and compression method (image compression is needed in order to decrease memory requirements).

At the present time either mathematically created phantoms can be displayed or CT numbers have to be entered manually. Other data are given as explained in the following sections:

9.3.2 PROGRAM MODULES

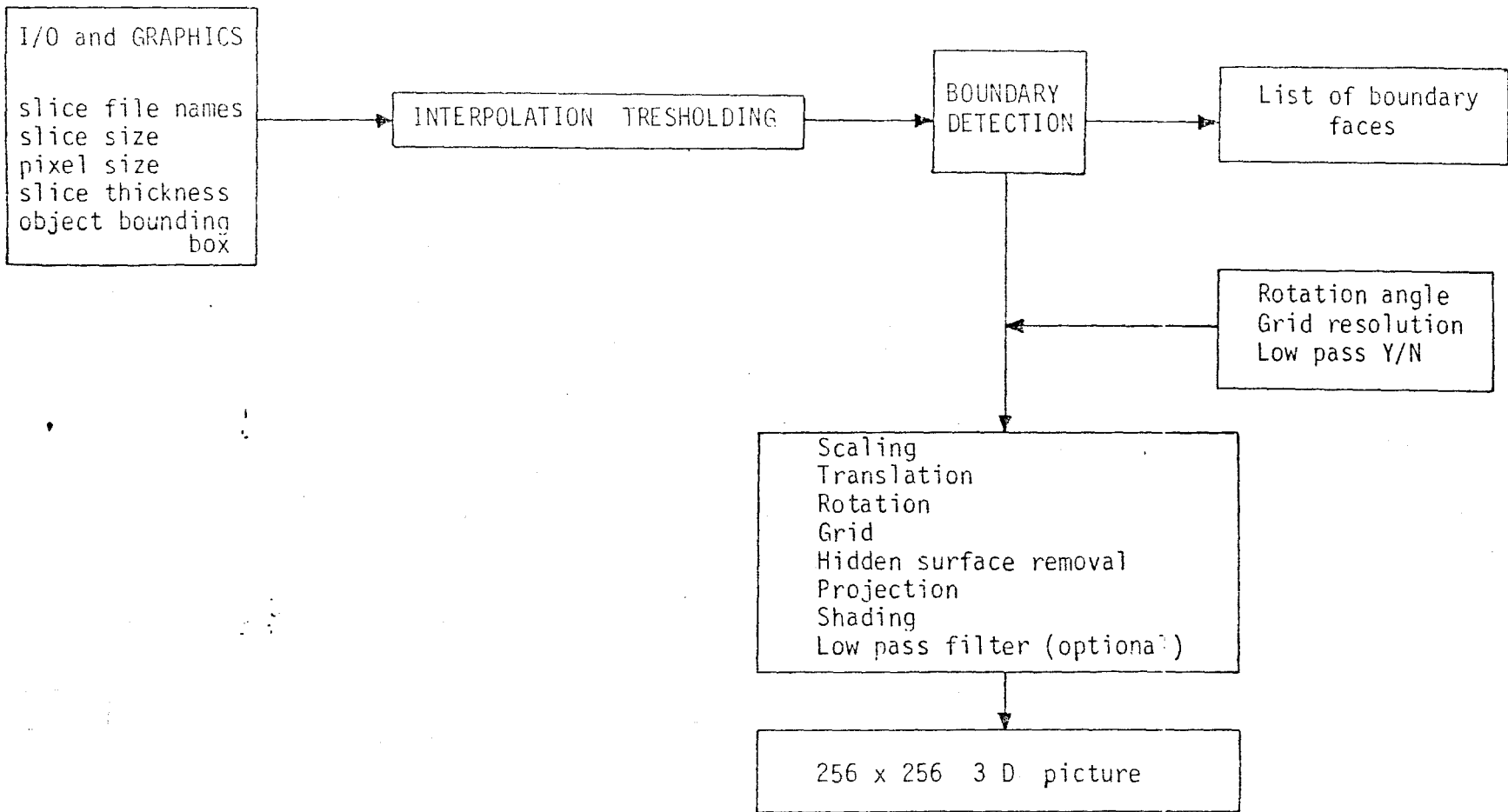
a) Identifying the object (thresholding)

The CT number range is needed for selecting the organ of interest. Since the CT number is proportional to the relative attenuation coefficient of the tissue in a particular region, certain organs can be associated with certain CT numbers.

Note: The CT disk reading procedures are considered proprietary data and no manufacturer has been able to provide us with such data yet though the author has made a few applications.

For example, a CT scanner using a CT number range between -500 and +500, water has a CT number of zero, air is equivalent to -500, soft tissue is represented by the approximate range 0-300 and high CT numbers indicate presence of bone.

FLOW DIAGRAM



FIG?9.1.1 Flow diagram of the 3D display system

ABSORBER	CT NUMBER	LINEAR ATTEN. COEF. (cm ⁻¹)*
Bone (dense)	+500	0.380
Intracranial soft tissues	+25 to +5	0.200 to 0.192
Water	0	0.190
Fat	-50	0.171
Air	-500	0

The user specifies a minimum CT number a maximum CT number. All pixels that have a CT number in between the two entered values are considered to be inside the organ, while the rest of the pixels are of no use for us. The resulting thresholded slice is displayed on the IBM PC screen using two gray levels or colors of the standard IBM graphics package. The original picture can also be seen at the 256x256 image display terminal. The CT number range can be changed interactively until the desired organ is represented by the operator.

b. Find the object bounding box

This module makes use of the IBM PC standard and turtle graphics routines. The function of this module is to determine the rectangular box containing the organ of interest. This way memory requirements are reduced and cut away views of an organ can be obtained. The user first displays a thresholded image of a desired CT slice and then interactively chooses a window that encloses displayed.

c. Interpolation

Most of the time the pixel size is smaller than the slice thickness due to design considerations of CT scanners. Since the 3D display algorithm is based on cube-shaped voxels then linear interpolation is used to estimate what the CT data would have been like if the voxels were cube shaped.

d. Bit mode storage of thresholded data

Since the thresholded slices are arrays of ones and zeros, only one bit is needed to store the value of the arrays. The heap area created by Turbo Pascal can then be used to store arrays. The program can find the byte and then the

bit location of each array element. It is clear that memory requirements are reduced by using this method of array representation.

A 128K area is reserved in the heap and consequently the program limits the input data as follows:

256 slices of 64x64 images
64 slices of 128x128 images
40 slices of 160x160 images

The source program to which the slice size is input, chains itself to program for bit storage. There are 3 chain files, each of which accepts one of three different types of input (64x64, 128x128, 160x160). After the data is stored in the heap, the program chains itself to the boundary detection algorithm.

e. Boundary detection

The boundary detection algorithm has been described in chapter IV. The auxiliary data structures of the boundary detection algorithm are a queue X and a list M. Both are linked lists created in the heap area. During the programme, the lists both grow and diminish. Dispose routines are included so that the previously allocated memory areas whose information are not needed anymore are available for reuse. As a result, the area where the information of slices is stored in the heap, does not collide with the list and queue.

The output of the boundary detection algorithm is the list of faces of voxels that are on the boundary of the object. To represent a face, the coordinates of the voxel, and a number identifying the particular face are needed. Since at most three faces of a voxel are visible from any viewing direction, the others need not be considered in the following routines where rotation, hidden surface removal projection and shading are performed. To achieve minimization of computation time, the faces are stored in six separate files corresponding to the six faces of the cube. The faces are

numbered from 1.6 according to the following convention.

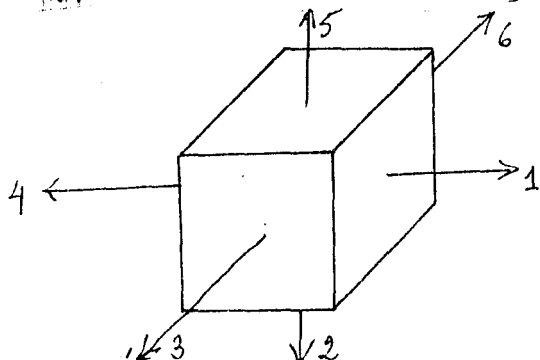


Fig.9.3.1 Numbering of faces

Each of six face files contain three numbers for each face, giving the center of the voxel having that face on the boundary.

The hidden surface removal rotation, scaling and shading routines are implemented as a separate program, because once the boundary faces are determined, the object may be displayed several times from any viewing direction.

f) Hidden surface removal

Given a viewing angle, this module removes the surface faces which are behind the visible surface.

First a decision is made on which of the six faces are visible by considering the cosine of the angle θ between the normal vector to the face and the line connecting the observer point and the face center.

If $\cos \theta > 0$, then the face is visible. Actually, the value compared is not zero, but some small number ξ , since for such small angles, none of the faces in that orientation can be seen on the screen. By choosing an appropriate ξ , one avoids calculating the gray values, and depths of faces which will never be displayed.

In hidden surface removal, the depth of the face is needed but the file contains the coordinate of the center of the voxel and not center of the face. To obtain the position

of the center of the face an offset value is added to or subtracted from that coordinate. (see fig.9.3.2)

can be seen on the screen. By choosing an appropriate , one avoids calculating the gray values, and depths of faces which will never be displayed.

In hidden surface removal, the depth of the face is needed but the file contains the coordinate of the center of the voxel and not center of the face. To obtain the position of the center of the face an offset value is added to or subtracted from that coordinate. (see fig.9.3.2)

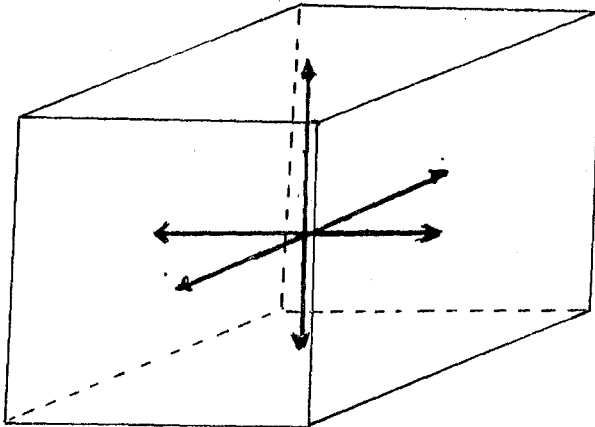


Fig.9.3.2 Offset values

h) Shading

The brightness of each face to be displayed is determined by using the following formula. (see chapter VIII).

$$B = \frac{\cos \theta}{d}$$

where θ : angle between the z direction and the normal vector of the face
 d : distance of face from observer
 B : brightness

The brightness (or gray level) should be kept within a certain range in order to avoid overflow. The minimum and maximum gray values in a desired range of values. Hence, the

shading formula becomes:

$$B = K \cdot \frac{\cos \theta - a}{b-a}$$

where $a = \frac{(\cos \theta)_{\min}}{(\text{depth})_{\max}}$ representing the minimum possible predicted brightness level

$b = \frac{(\cos \theta)_{\max}}{(\text{depth})_{\min}}$ represents the maximum possible predicted brightness value

and K is a constant that determines the gray level range between 0 and 255. $(\cos \theta)_{\min}$ and $(\cos \theta)_{\max}$ are determined by comparing the cosines of the visible faces.

$$\begin{aligned} (\text{depth})_{\max} &= (\text{depth})_c + D \\ (\text{depth})_{\min} &= (\text{depth})_c - D; \end{aligned}$$

where $(\text{depth})_c$ is the depth of the center of the object, and D is half the diagonal of the object. Whatever the size of the input, the output is converted into an 256×256 array.

i) Scan conversion

The projected faces have to be displayed on a 256×256 resolution screen. The objective is to have the most accurate display of these faces. With this in mind, one should overcome the effects of rounding errors and antialiasing effects.

The easiest method is to project the representation of the object onto an display array of the same size as the object array. This method provides low quality displays.

The scan conversion method used in our implementation is as follows:

Each face is represented by a $n \times n$ grid of smaller pixels where n is an odd number. The center pixel of the grid is considered to have a coordinate value of $(0,0,0)$. The grid is rotated for each set of visible faces.

After having calculated the brightness level of each face, it is divided into $n \times n$ pixels with coordinates that are calculated according to the relative pixel coordinates of the corresponding grid.

As the number of grid elements n is increased, the resolution is increased, the antialiasing and rounding effects are reduced and eventually eliminated.

The display array can be assumed any large array (i.e. 256×256 or 512×512). This large array can then be reduced to the actual display size by averaging. Fig.9.3.3 shows the use of grids in scan conversion.

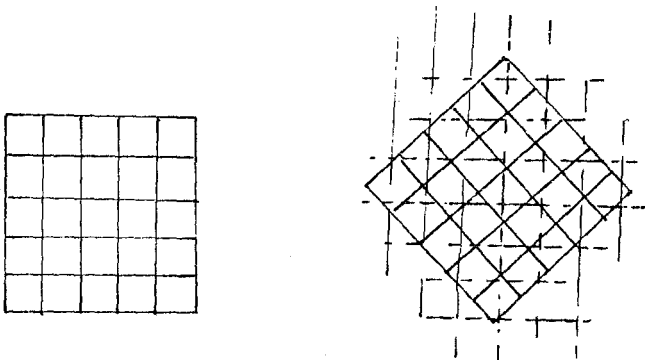


Fig.9.3.3 A 5×5 grid representing a rotated face.

X.RESULTS AND DISCUSSION

This thesis has presented a software package which can display organs in a three dimensional way using data obtained from a CT scanner and an IBM PC microcomputer.

First existing 3D display techniques and their application in medicine have been given. Then the particular organ selected for implementation has been described and considerations related to our specific implementation have been discussed.

The operating procedure of the developed system can be summarized as follows.

The CT slices are first read by the computer. Then the organ of interest is identified using interactive graphics and thresholding, the bounding box of the object is determined and the slices are interpolated in order to create CT slices that have a pixel size equal to the slice thickness. The data obtained is then processed by the boundary detection program which produces the set of faces forming the object surface.

Once the object surface is determined another program can be run in order to display the organ in a sequence of desired viewing angles. This program performs the following routines. Rotation of the organ, hidden surface removal and shading. A scan conversion technique which provides high quality displays has also been incorporated into this program.

10.1 Illustrative Results:

Mathematical phantoms have been created in order to assess the effectiveness of our 3D display package. These phantoms have been organised in slices in a similar way to real CT data.

The first mathematical phantom had a rectangular shape as shown in fig.10.1.1. The input consisted of 10 slices; each slice being an array of 64x64 CT numbers.

The boundary detection algorithm has successfully detected all the surface faces of the object and created the six face files. Then the phantom has been displayed in several directions and 3D views of the object have been obtained. (See fig.10.1.2 and 10.1.3).

These results have been obtained by using a grid of 7x7 resolution. When no grid was used considerable antialiasing effects (holes) and a jagged appearance due to low resolution and rounding errors have been noticed. (see fig.10.1.4).

G.I.Herman (2) suggests the use of a low pass filter to avoid holes and other image quality problems. We found that using the grid produces high quality displays without the need of any low pass filtering. Furthermore, the use of a grid does not introduce much extra computational time.

For certain rotation angles a single 3x3 point low pass filtering may be used to avoid occasional anti-aliasing effects. It is though evident that this type of filtering smoothes the image and eliminates noise at the expense of degrading the resolution and blurring the image. (see fig.10.1.5).

Scaling the object by reducing the size has also been reported in literature (9) to reduce the antialiasing effects. Again, using an appropriate grid seems to eliminate the need for such compression.

10.2 Discussion

The primary advantages of the 3D display implementation described in this thesis can be enumerated as follows:

- a) The system has low cost. This eliminates one of the

main disadvantages of 3D display systems.

b) The IBM PC can accommodate a large amount of core memory (512 K Bytes). The memory size can be made even larger for newer IBM PC families. The availability of core memory makes it possible to work without virtual memory for up to a certain amount of input data.

c) Turbo Pascal provides a very easy environment for program development and modification.

d) The system can be expanded easily to accommodate larger arrays such as 256x256 and 512x512 matrices by using virtual memory in an efficient way.

e) The same hardware system can be used for other image processing functions or routine IBM PC functions such as patient record tracking

f) The system can be networked to other IBM PC's or terminals using PC-SHARE.

g) Displayed images have high quality.

Some of the disadvantages and limitations of this system are:

a) CT data are usually transformed via either 8'' floppy disks or magnetic tapes. In either case, additional hardware may be necessary for efficient use.

b) The processing speed is low. This limitation can be remedied though by using faster, new IBM microcomputers. Furthermore, virtual memory minimization can decrease the processing time significantly.

c) The boundary detection algorithm is not very well suited for interactive surgical planning procedures. However the needed modification can be made with no difficulty.

d) The shading rule used is the simplest one available. More realistic images can be obtained by using more complex shading rules.

Finally, it should be stated that being able to read real CT files is a major problem due to unwillingness of CT manufacturers to provide the CT file formats and image compression methods. The author is optimistic about overcoming this problem.

10.3 Future Work

Several recommendations can be made for improving the present 3D display systems and for adding new features and new application modules. Furthermore, there is a number of medical applications which should be evaluated in order to obtain the maximum benefit from the 3D display technique.

Recommendation for future system modifications and new implementations.

Some features like transparency, moving, opening, zooming, shadowing, surface area estimation and volume estimation can be easily incorporated into the present system.

The effects of other types of shading and pseudocolor methods should be investigated in order to increase the realism of the display.

Particularly, displaying an opaque surface within a transparent object needs some future work. The objective in this case is that the operator should be able to see the spatial relationship between these two objects.

The thresholding procedure is prone to error. 3D image processing procedures that extract useful features like gradient values and texture characteristics can be developed for better identification of the object. This may be necessary for soft tissue display where the boundary cannot be as clearly defined as the bone.

The communication between a CT and the 3D display system should be established. This may require an 8" disk drive which should be connected to IBM PC. In addition CT data reading procedures should be obtained from manufactures.

A joystick or a trackball may be useful for making the user-computer interaction more practical. A video disk unit which can store large amounts of data could be connected to the IBM PC. With fast data transfer to the image processor, one can then display a series of 3D images in a movie-like fashion and create the illusion of such effects as continuous rotation and variable illumination. At the present time, the

display terminal communicates with the IBM PC via a centronix parallel port. This data transfer is limited by the IBM and a new I/O software module should be designed in order to increase the data transfer which translates into faster image display.

Computer communication techniques can be introduced, for resource sharing and distribution services. A digital communication network over telephone lines may be designed. Image compression technique should be applied in this case to fasten the data transfer speed and to reduce memory space requirement.

New application systems

A number of software/hardware modifications can be adapted to present system in order to extend its use in medicine

a) Surgical planning: New software is needed to allow a surgeon to plan his surgery on the computer. The surgeon can rehearse on the computer and decide exactly on the sites to be operated and changes that the need to perform in an interactive computer graphics environment. (See section 2.2 for further details on surgical planning)

b) Modeling. The computer can generate the code necessary for creating a plastic model of an organ by using a numerically controlled milling machine. This can be used for both surgery planning and prosthesis design.

User-computer interaction is a subject of interest here and needs investigation. Using the present 2-D interaction techniques to modify a 3-D object is the most unnatural procedure. New interaction modes should be looked for.

c) Custom prosthesis design:

To reduce the depreciation of prosthesis, new software for designing prosthesis with optimal load transfer

by maintaining maximum fit can be developed, where the user interactively with the computer makes decisions. Methods for reducing the need for interaction should be developed, while for structures, where thresholding can not be done, interaction should be called for separating the organ from the tissue that the computer can not differentiate. In CT exams of the knee, where relatively hard surface cartilage connecting the femur, tibia and patella components are difficult to separate without human intervention. When CT images cross joints user interaction is needed to select border from several that appear as a part of a single structure.

d) Radiation therapy- 3D reconstruction techniques can be used for devising a treatment plan for radiation therapy where the problem is to deliver maximum dose of radiation to a tumor while minimizing the dose to the surrounding healthy tissue. The desired objective is that the isodose surface does not intersect the sensitive structure. In practice the picture obtained can be rotated until the nonintersection of isodose surface with the surrounding sensitive structure is verified.

e) 3D MRI - 3D MRI is promising field of application. Bone marrow seems to be suitable for this application because of its high signal density. The potential role of 3D histogram analysis in quantitative bone marrow densitometry is being investigated.

f) Bone densitometry - Quantitative CT assesment of trabecular bone density in the proximal femur can not be done with the single slice strategy used for vertebral mineral determinations, due to its complex 3D geometry. Three dimensional histogram analysis overcomes this problem by introducing an integral approach to cancellous bone densitometry, over a bone volume which covers both the femoral neck and intertrochanteric regions. 3D histogram analysis of CT data may be the approach of choice for the assessment of osseous, integrity providing a noninvasive means for estimating bone strength.

g) Soft tissue imaging - Although 3D images of soft tissue can be made by increasing contrast as a result of intraarticular injection of air, the results are not very successful, the operation is time consuming and seems to provide no benefits according to the reports of the researchers at the UCSD (10)

Finally, the potential benefit of 3-D technique with different tomographic modalities should be investigated. These include Position Emission Tomography, single Photon Computed Emission Tomography ultrasound and others.

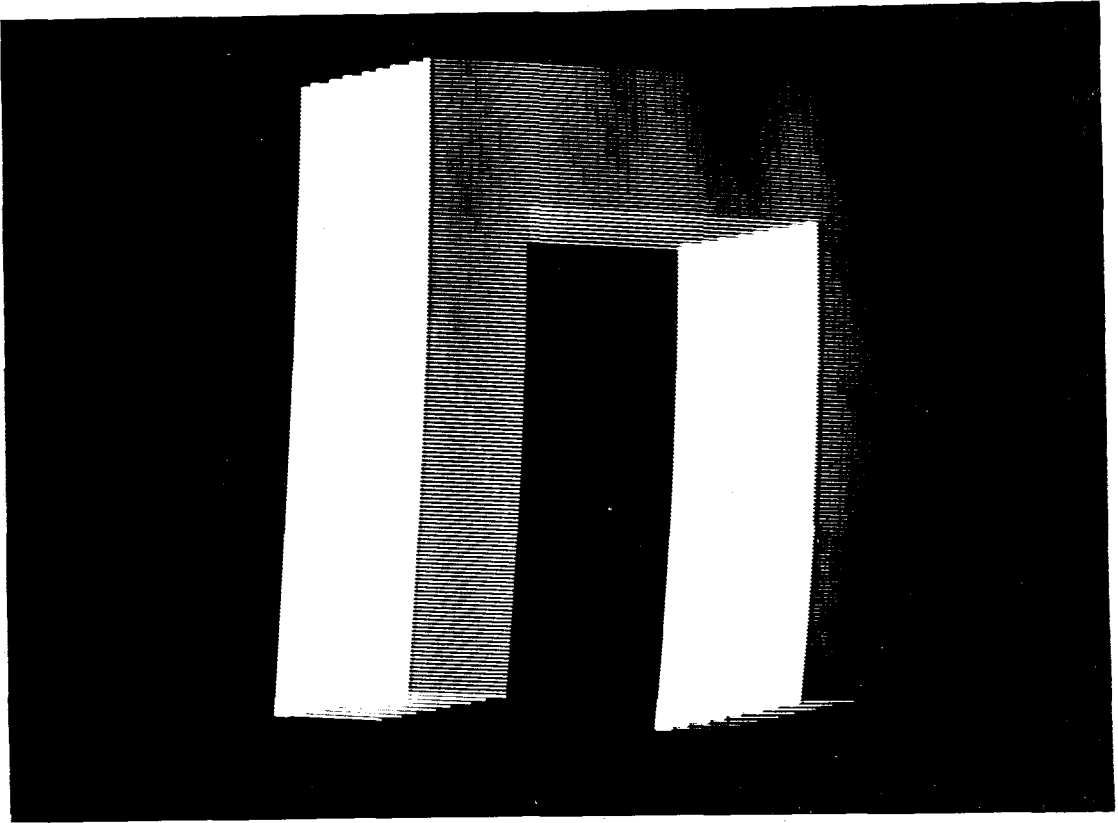


FIG.10.1.1 Rectangular mathematical phantom

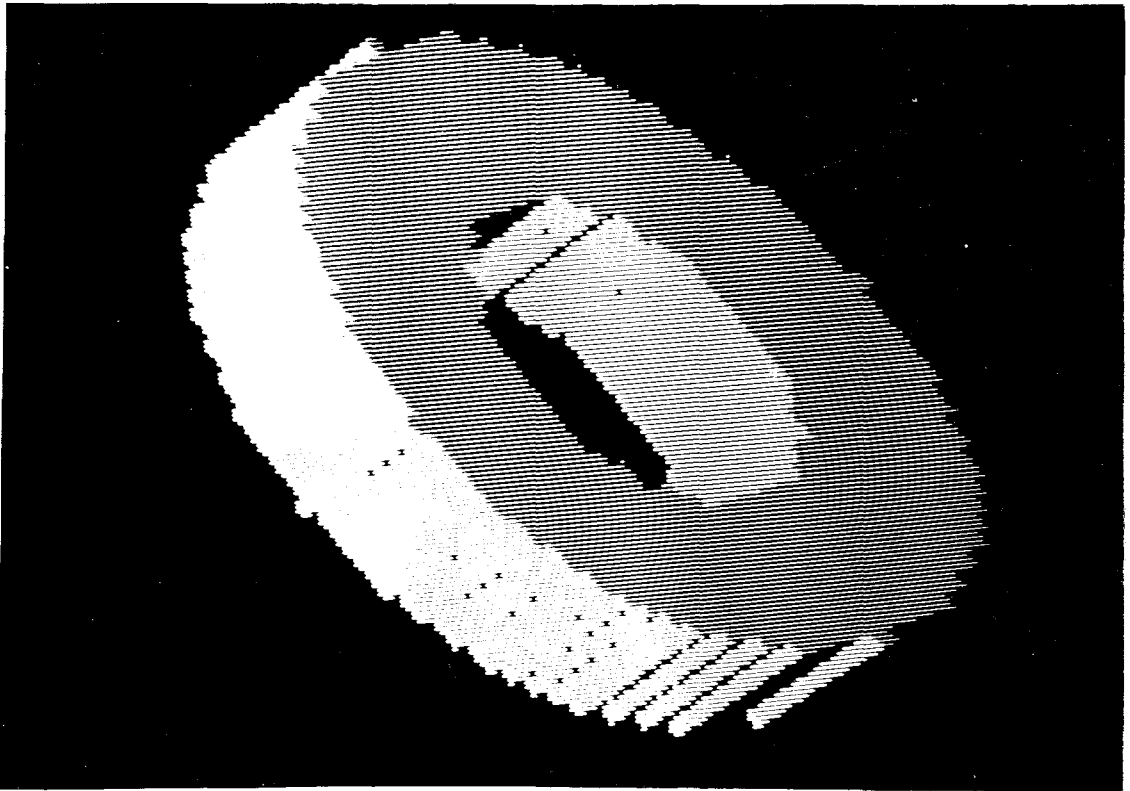


FIG.10.1,2 Object rotated

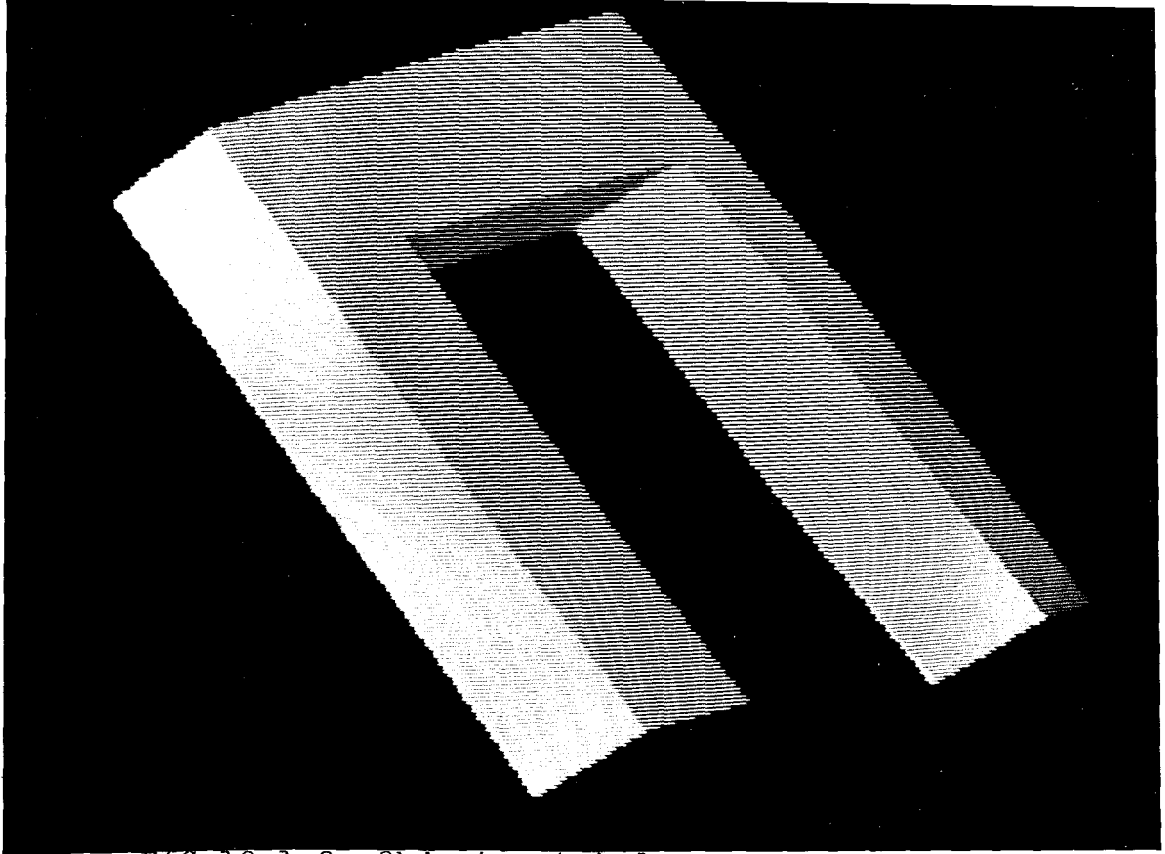


FIG.10.1.3 Object rotated

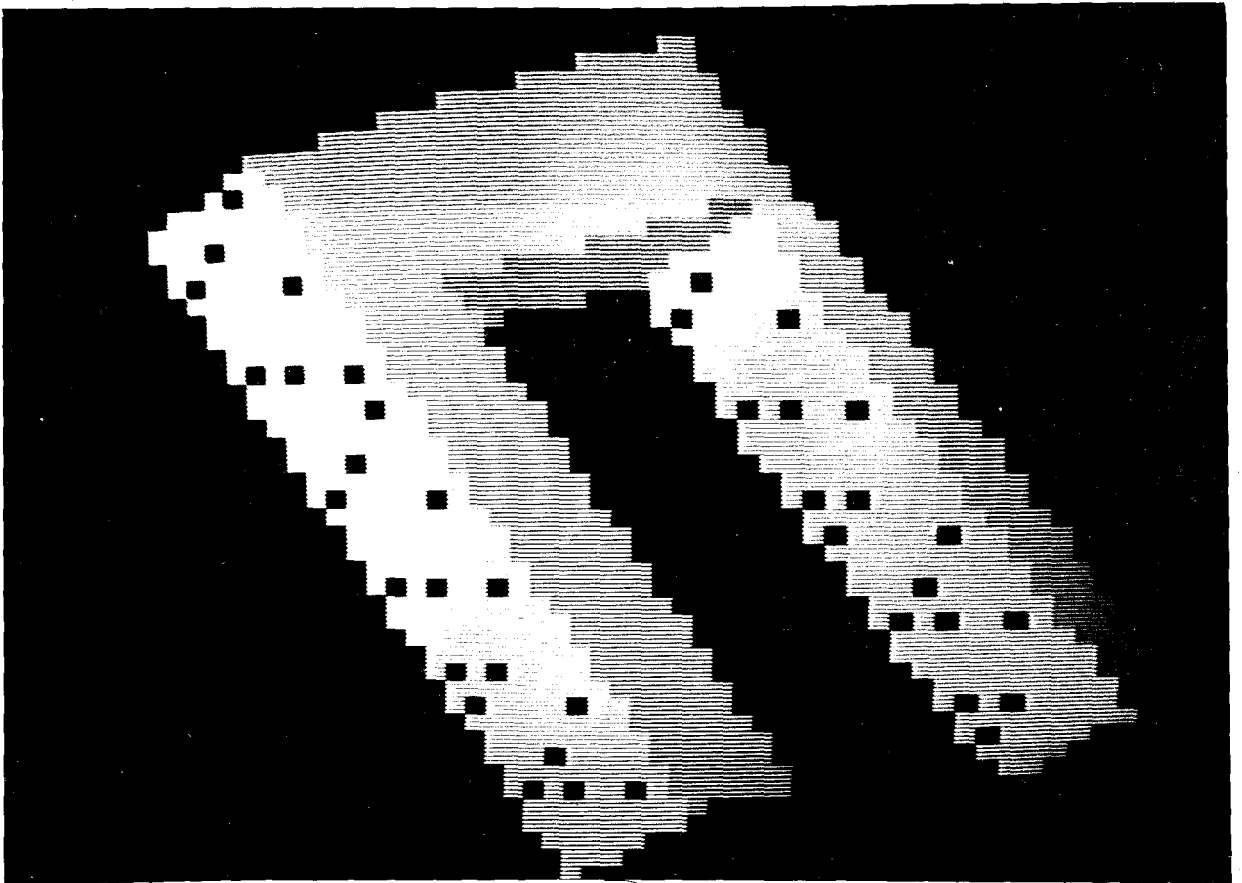


FIG.10.1.4 Image without using grids. Jagged edges and holes due to anti-aliasing are seen

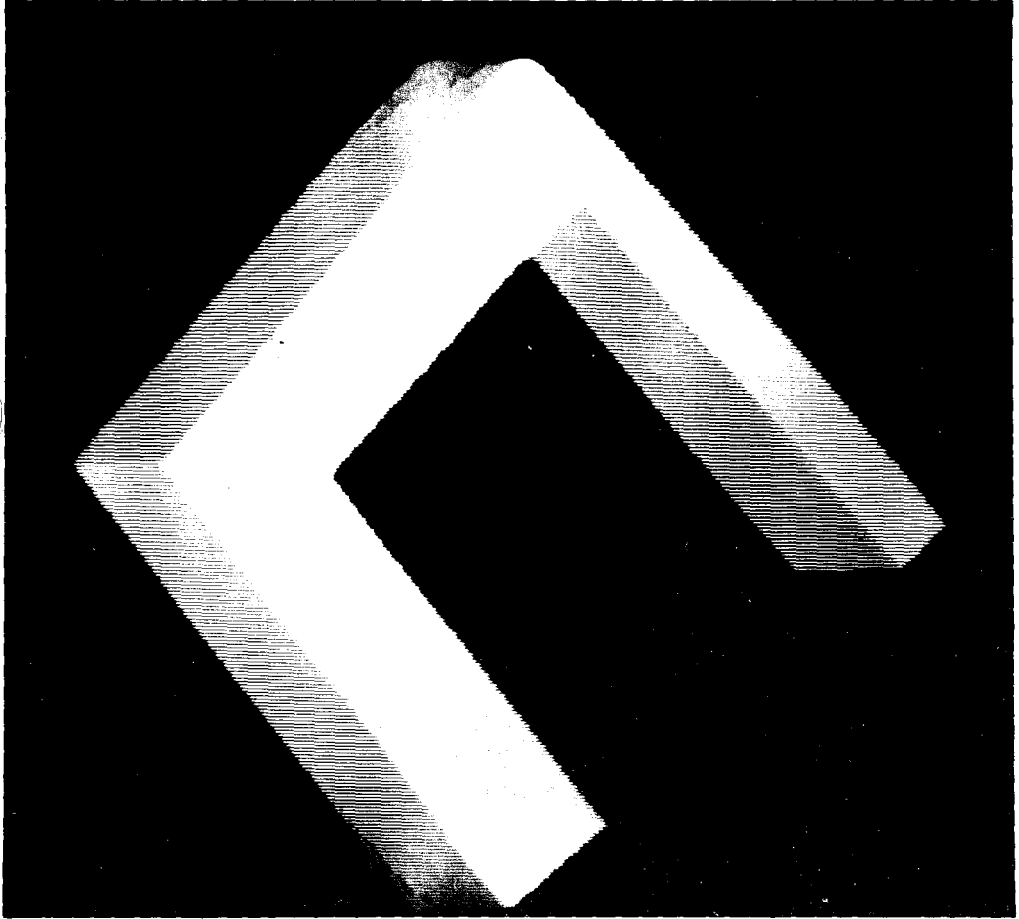


FIG.10.1.5 Image obtained by using grid and
lowpass filter

BIBLIOGRAPHY

1. Artzy, E., Fieder, G., and Herman, G.T., "The Theory Design, Implementation and Evaluation of a Three-Dimensional Surface Detection Algorithm", Computer Graphics and Image Processing, Vol.9, pp:1-24, 1981.
2. Herman, G.T., Liu, H.K. "Three Dimensional Display of Organs From Computed Tomograms", Computer Graphics and Image Processing, vol.9, pp:1-24, 1979.
3. Udupa, J.K. "Display of 3D Information in Discrete 3D Scenes Produced by Computerized Tomography", vol.71, no.3, March 1983, pp:420-431.
4. Cook, L.T., Dwyer III, S.J., Soloman, B., Lee, K.Y., "A Three Dimensional Display System for Diagnostic Imaging Applications", IEEE CG and A, August 1983, pp:13-19
5. Udupa, J.K., "Interactive Segmentation and Boundary Surface Formation For 3-D Digital Images", Computer Graphics and Image Processing, Vol.18-1981, pp:213-235.
6. Meagher, D, "Geometric Modelling Using Octree Encoding", Computer Graphics and Image Processing, Vol.19, 1982, pp:129-147
7. Doctor, L.J., Torborg, J.G., "Display Techniques for Octree Encoded Objects", IEEE CG and App, July 1981, pp:29-38
8. Gargantini, I, Walsch, T.R., Wu, O.L., "Transformations of Voxel Based Objects via Linear Octrees", IEEE CG and A, October 1986, pp:12-21
9. Frieder, G, Gordon, O, Reynolds, R.A., "Back-to-Front Display of Voxel-Based objects", IEEE CG and A, January 1985, pp:5260
10. D.J.Sartoris, "3D Display of CT data: new aid to pre op surgical planning", Diagnostic Imaging, May/June 1986, pp:2632

11. Brewster, L.J., S.T. Sushma, Tuy, H.K., Udupa, J.K., "Interactive Surgical Planning", IEEE CG and A, Vol.4., No.3, March 1984 pp:31-40
12. Rhodes, M.L., Kuo Yu-Ming, Rothman, S.L.G., "An Application of Computer Graphics and Networks to Anatomic Model and Prosthesis Manufacturing", IEEE CG and A, February 1987, pp: 12-25
13. Herman, G.T., Reynolds, R.A. Udupa, J.K., "Computer Techniques for the Representation of Three-Dimensional Data on a Two Dimensional Display", SPIE, Vol 367, pp:3-14, 1982.
14. Granholm, J.W., Robertson, D.D., Walker P.S., Nelson, P.C., "Computer Design of Custom Design Stem Prothesis", IEEE CG and A, February 1987, pp:26-35
15. R.J. Wilson, Introduction to Graph Theory, Academic Press, New York, 1972
16. A.W. Schurle, Topics in Topology, Elsevier North Holland Inc., New York, 1979
17. Rosenfeld, A., "Adjacency in digital pictures", Information and Control, vol.26, pp:24-33, 1974.
18. Newman, W.M., Sproull, R.F., Principles of interactive Computer Graphics, second edition, McGraw-Hill Book Company, 1979.
19. Harrington, S., Computer Graphics, A programming Approach, McGraw-Hill Book Company, 1983.

20. Gouraud, H., "Continuous Shading of Curved Surfaces" IEEE Transactions on Computers, Vol. C-20, no. 6, June 1971, pp: 623-629
21. Hohne, K.H., Bernstein R, "Shading 3D images from CT Using Gray Level Gradients", IEEE Transactions on Medical Imaging, Vol MI-5, No. 1, March, 1986, pp: 45-47.
22. Herman, G.T., Udupa, J.K., "Display of Three Dimensional Discrete Surfaces", Proc. of SPIE, vol. 283, pp: 90-97
23. Herman, G.T., Reynolds R.A., Udupa J.K., "Computer Techniques for the representation of three-dimensional Data on a two dimensional display", Proceedings of SPIE, vol. 367, 1982, pp: 314.