INTERACTIVE MEDICAL VOLUME VISUALIZATION FOR SURGICAL OPERATIONS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of
Dokuz Eylül University
In Partial Fulfillment of the Requirements for
The Degree of Doctor of Philosophy in Electrical and Electronics Engineering,
Electronics and Communication Program

by
Ahmet ÖZKURT

February, 2001
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We certify that we have read the thesis, entitled **"INTERACTIVE MEDICAL VOLUME VISUALIZATION FOR SURGICAL OPERATIONS"** completed by **Ahmet ÖZKURT** under supervision of **Prof. Dr. Kemal ÖZMEHMET** and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Prof.Dr. Kemal ÖZMEHMET

Supervisor

Prof Dr. Uğur HALICI

Committee Member

Yrd Doç Dr. Mehmet KUNTALP

Committee Member

Approved by the

Graduate School of Natural and Applied Sciences

Prof.Dr. Cahit Helvacı

Director

ACKNOWLEDGMENTS

I am grateful to my advisor, Prof. Dr. Kemal Özmehmet, for his invaluable guidance and continuous support during this study.

I would like to thank academic personnel of Dokuz Eylül University, Dept. of Electrical and Electronics Engineering for their sincere aid and friendship. In particular, I appreciate the help of Dr.Mehmet Kuntalp, Dr.Damla Kuntalp and Dr.Yeşim Yüksel.

The comments from the surgeons about my study were important for me. I was fortunate to have this type of help from Prof. Dr. M. Nuri Arda and Op. Dr. Mustafa Özer. I would like to thank these people for their encouragement.

Without morale support and love of my family; my mother, Şennur Özkurt and my pretty wife, Nalan; I might never have finished this long work. I'd like to thank them for their patience.

And finally, this work is dedicated to memory of my father, Celil Özkurt, who spent his life working for us.

ABSTRACT

In this thesis, "A Surgeon Assistance System", which can be used in surgery, surgical planning, and education, has been designed. The main idea lying in this approach is to equip the surgeon with a Head Mounted Display (HMD), a camera, and a powerful computer system, and overlay all information, which may required during the operation, on the real-time video signal acquired from operation area.

The system processes pre-scanned two and three-dimensional MRI (Magnetic Resonance Imaging) and CT (Computed Tomography) data and other real-time vital signs of the patients. The processing phase consists of pre-processing routines such as noise cancellation and thresholding. Processing stage consists of segmentation for tissue classification for MRI; triangulation in order to create three dimensional surfaces for both MRI and CT, and finally, three dimensional visualization routines by using OpenGL library. Also, one and two-dimensional information can be displayed instead of three-dimensional views.

By the help of this approach, it is expected that, the success rate of the surgery can be increased, concentration of the surgeon may be save during all operation, the operation time may be decreased, and more efficient surgery planning possibilities can obtained. This work is also the first step for telemedicine system, which includes remote actions in medicine.

ÖZET

Bu çalışmada, ameliyat planlamasında, eğitiminde ve ameliyat sırasında kullanılabilecek bir operatör destek sistemi geliştirilmiştir. Bu sistemde temel amaç, operatörün bir HMD (Head Mounted Display), kamera ve güçlü bir bilgisayar sistem ile donatılması ve ameliyat sırasında gerekebilecek tüm bilginin ameliyat bölgesinin gerçek video görüntüsü üzerine bindirilmesidir.

Sistem, önceden alınmış iki veya üç boyutlu MRI (Magnetic Resonance Imaging) and CT (Computed Tomography) görüntüleri ve hastanın gerçek zamanlı diğer verilerini işleyebilir. İşleme bölümü gürültü engelleme ve eşikleme gibi ön işleme algoritmalarını, MRI için, bölütleme yoluyla doku belirlemesi, CT ve MRI için, üçgenleme yoluyla üç boyutlu yüzey belirlenmesi ve son olarak OpenGL grafik kütüphanesi kullanılarak üç boyutlu görüntüleme işlemlerini kapsar. Aynı zamanda bir ve iki boyutlu işaretler de görüntülenebilir.

Bu yöntemle, ameliyat başarı oranının arttırılması, operatör konsantrasyonunun bütün işlem boyunca sağlanması, operasyon süresinin kısaltılması ve daha verimli bir ameliyat planlama sisteminin oluşturulması beklenmektedir. Çalışma aynı zamanda, tıpta uzak noktalar arasındaki işlemleri konu alan telemedicine alanındaki çalışmalar için e ilk adım niteliğindedir.

CONTENTS

Chapter Three

THEORETICAL BACKGROUND

Chapter Four

IMPLEMENTATION OF
MEDICAL VOLUME VİSUALIZATION

Chapter Five

CONCLUSION

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE
INTRODUCTION

1.1 Traditional Surgery

Surgery is a treatment method for some abnormal structures in any organ, tumor and deformations. These illnesses can be replaced or corrected in the surgeries. All surgeries have risks about the health of the patient depending on the organ and the type of operation, therefore high concentration and care are required.

In traditional procedures, a surgeon has to be prepared thoroughly for the operation. Although operation rooms are very complex areas because of the need for various high technology equipments, the success of the operation depends primarily on the performance of the surgeon. The surgeons of today, as in the past, rely primarily upon hand-eye coordination for performing various procedures. "Because light does not penetrate deeply into tissue, the surgeon's vision is limited to the exposed surface, and this restriction essentially defines the way most surgical procedures are performed" (Kikinis et al, 1991). The surgeons generally complain from restricted field of view, lack of contrast between target and surrounding tissues and inability to see beneath the surface. (Kikinis, 1994). Computer Assisted Surgery (CAS) systems can solve these problems using combination of high resolution imaging and detailed anatomical information.

During the surgery, a surgeon uses some special equipments, investigates previously obtained diagnostic images like CT data, and tries to implement the operation as much as he/she can see and touch. Some interrupts, such as expectation of biological information from another medical staff, looking at an X-Ray film on the

wall, and required but not acquired information during the operation, may cause concentration breakdowns and affect the success rate of the surgery. Such difficult conditions require a high concentration and well-organized preparation.

Education of a surgeon is a very long and tough process. The surgeon has to be an expert about biological areas, depending on the Region of Interest (ROI), sometimes called Volume of Interest (VOI). Also, each operation requires extra preparation efforts, good planning and surgery organization, and emergency procedures in case of existence.

1.2 Surgery Assistance

Integration of recent advances in computed tomography (CT), magnetic resonance imaging (MRI), nuclear medicine imaging, and image processing technology with modern therapeutic techniques (stereotactic (Burckhardt & Flury & Glauser, 1995) and microscopic surgery, radiosurgery, and radiation therapy) has dramatically affected the clinical approach to the treatment planning of most of the illnesses. It has been realized that these advances in imaging techniques can be utilized not only for diagnosis and anatomical mapping, or localization of tumors on cross-sectional planes, but also for establishing the exact three-dimensional spatial coordinates of a surgical procedure, preoperatively.

The surgeon's vision can be extended (Tang & Kwoh & Teo & Sing & ling, 2000) by the use of MRI and CT, two powerful imaging technologies which can penetrate beneath the surface and provide an image of the human body in three dimensions. Using computerized reconstruction and image processing techniques/images can be provided in a form in which the surgeon, using an interactive computer workstation, can navigate through the simulated organs without cutting, and can plan and rehearse surgical procedures. Three-dimensional reconstructions of imaging data can also be used for real-time control during the performance of surgical procedures or other therapeutical interventions (Kikinis et al, 1991).

Such systems can be used as "Pre-Operative Planning" purposes for educational aims (Sorid & Moore, 2000) and Intra-Operative Assistance to improve the success rate of the surgery. The surgeon can easily investigate the surgery area by using two dimensional "slice" views or three-dimensional reconstructed "augmented" views, which include biological content before the operation. During the operation, that information may be overlaid (Blackwell & Nikou & DiGioia & Kanade, 2000) to the video, which is acquired in real-time. In other words, vision is enhanced. This method may be helpful to surgeon while preparing and performing operations.

1.3 Main Idea

In this dissertation, a new system, named "MEDVR" (Medical Virtual Reality System) was developed and the possibility of its usage in a surgery in order to assist the surgeon with the help of virtual reality equipment (Dumay, 1996) (Rosen & Soltanian & Redeett & Laub, 1996) with real vision and data was investigated. The method is shown in Figure 1.1.

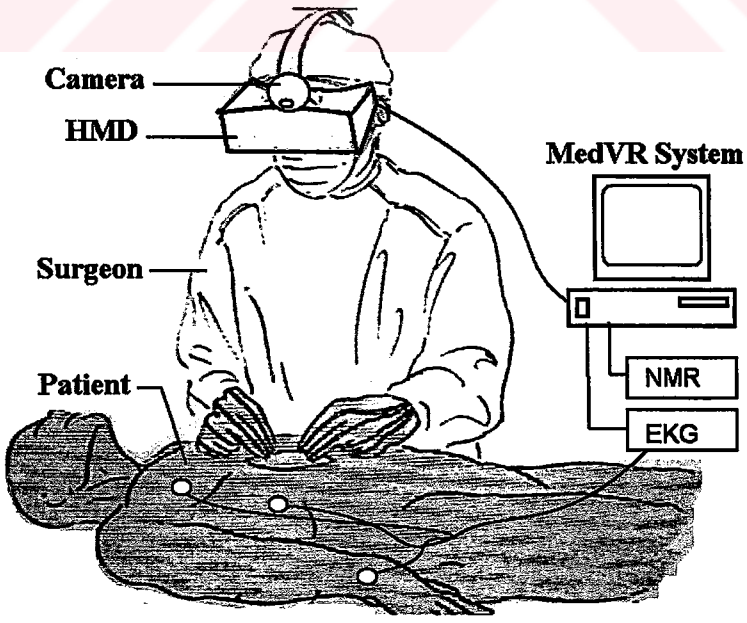


Figure1.1 Basic Augmented Reality Approach (Rosen & Soltanian & Redett & Laub, 1996, Revised)

This method is known as "Augmented Reality" (AR) (Brys, 1992) (Tang & Kwoh & Teo & Sing & Ling, 1998) or "Enhanced Reality" (ER) (Grimson et al, 1996). The purpose of our approach is to improve the surgeon's view with external imaging equipments such as high resolution cameras and to assist the surgeon with the extra anatomical information overlaid (Blackwell & Nikou & Digioida & Kanade, 2000)] (Rosen & Soltanian & Redett & Laub, 1996) on the real video. Anatomical information may be biological signals such as electrocardiogram (ECG) and oxygen concentration in real time, and previously acquired and processed medical diagnostic modalities, such as Nuclear Magnetic Resonance (NMR) and Computed Tomography (CT) images.

There are several surgeon assistance systems in use (Küpnapfel & Krumm & Kuhn & Hübner & Neisius, 1995). Most of them serve as desktop patient monitoring facilities which supports two or three-dimensional views of surgery area. This method requires that the surgeon investigates the surgery area from a different place and this may cause loss of concentration for the operator. Some of assistance units use pre and intra-operative (Chabat & Hansell & Yang, 2000) camera systems displaying operation area from inside the patient. These systems have disadvantage of camera placement. Our system combines all the methods given above and may satisfy all the requirements of surgeons during the surgery providing them with comfortable view and more details. Therefore, our approach could be an alternative to current surgery assistance methods.

1.4 The Goals

The goals of this study are as follows:

1. **Surgery Assistance**: The success rate of the surgery depends on the performance of the surgeon. Our system may improve the performance and concentration of the surgeon by using integrated computer aided vision of related tissues.

2. **Help in Surgery Planning and Education**: Such systems may be used as pre-operative planning tool and anatomical voyager by the surgeons and medical students (Sorid & Moore, 2000).

3. **Remote Surgical Actions**: The third significant goal of the study is the "possibility of remote actions for maintenance and improvement of health" (Rovetta et al, 1993). This utility is called "Telemedicine" and our approach may be extended in order to be used in "Telesurgery". This opportunity may open wider considerations in the future.

4. **Medical Segmentation**: Separation of specific medical tissues from the other ones is a very important clinical need. An automatic segmentation scheme for brain tissues is investigated, and a method, which is a mixture of spatial, statistical and neural approaches, is applied.

5. **Interactive Volume Visualization**: There is also a need for real time reconstruction and manipulation of three-dimensional view of biological tissues. This need is studied in three-dimension by using OpenGL.

1.5 The Significance of our Approach

Our approach comes with some advantages and improvements over traditional surgery. At the same time, we offer some new methods for segmentation and polygonizing, which are useful for any of other application areas. Advantages of our study are summarized belows.

Before the operation, our system can be used as a surgery planning tool. Three-dimensional reconstructions with interactive manipulation provide the optimal investigation of medical structures. Therefore, surgery preparation phase for an operator may take less time by using integrated three-dimensional views.

The success rate of the surgery can be increased. During the operation, if we support the surgeon well, it is possible to perform the surgery in a shorter period, successfully, and without any unexpected outcome.

Segmentation approach developed may be alternative to unsupervised texture classifications with the usage of statistical texture parameters such as entropy, momentum and energy. Also, our polygonizing scheme can be used as a visualization tool for any other non-medical physical variables like temperature, pressure etc; in a three dimensional interactive manner.

1.6 Meaning of Interactive Visualization

Interactive word is coming from the adjective form of the word "interact" with a meaning of "have an effect upon each other" (Oxford English Minidictionary, 1995, p271). As a technical term, interactive means machine-user interactions (Kühne et al, 1997) and usage in a way that realtime manipulation (Haigron & Berre & Coatrieux, 1996) of data and commands is possible.

The relationship between our study and this term is about the visualization content and user controlled commands (Bielser & Volker & Maiwald & Gross, 1999). Because there must be a transition and processing time, the commands such as "rotate" and "show" affect the display content in almost real-time. The display content for our study is three-dimensional view of medical areas. The user can manipulate the image by voice, keyboard, mouse or any of other machine-user interaction equipments.

In our work, the display content is segmented/not segmented, raw/polygonized three-dimensional renderings of related medical tissues. The user can change the angle of view, rotation of objects and show or hide the content with the help of an operator, keyboard, and Head Mounted Display (HMD). Head movements of the surgeon can be used as the source of rotation commands. This enables the surgeon to manipulate the display content without using his/her hands.

1.7 Region of Interest (ROI)

Our approach can be used in various areas and in many applications such as medical visualization, robotics and mechanical production. Medical area, which is

important for human health, is the chosen as application area. Region of interest of this study in medical media is predefined as surgery operations because of their critical importance. Since head and skull surgeries are difficult to operate, our work is focused investigating possible improvements of surgeon's vision in surgical operations. These improvements can be used in surgery planning, intra-operative procedures, and post-operative diagnostic studies. Also, remote actions are possible for consultance systems.

1.8 Thesis Organization

The rest of the dissertation is organized in the following way:

In the Second Chapter, all the necessary medical background is laid out. Main medical imaging modalities (NMR and CT), medical diagnostic signals and their topologic meanings are also given in this chapter.

The Third Chapter is about technical background for two and three dimensional signal processing related to medical image and volume: The data structure, preprocessing modalities, material classification (Segmentation), isosurface extraction and visualization fundamentals.

The Fourth Chapter is about the implementation of work using approaches described in the thesis. The thesis outline, segmentation efforts and results, polygonizing and surface extraction, and example applications are all presented in this chapter.

The last chapter, **Chapter Five**, is a conclusion. The results obtained from our study are discussed and their possible benefits and drawbacks are examined. The importance of the study for future scientific developments is also expressed in this part of the dissertation.

CHAPTER TWO
MEDICAL BACKGROUND

2.1 Introduction

In this chapter, the medical side of the dissertation will be summarized. In the light of the thesis focus, we will mention only the things which are used in the surgeries from the side of diagnostic evaluation. Due to the technical nature of the thesis, we will avoid the biological meanings and leave the diagnostic explanations to the medical experts. Three or two-dimensional NMR and CT imaging modalities are discussed below. The idea for this chapter is to give a glance to readers, who may face with our approach in the beginning. Since we choose brain and skull for imaging area of the human body, images and volumetric views given in this dissertation are related to only those structures.

2.2 Medical Imaging Modalities: NMR, CT and PET

In this section, Nuclear Magnetic Resonance (NMR), Computed Tomography (CT) and the Positron Emission Tomograpy (PET) methods for diagnostic tomography are summarized. However, only the methodology, will be given and deep theoretical and chemical issues will be discussed briefly.

NMR

> *"Over the past several years, magnetic resonance imaging (MRI) has developed into a very powerful and versatile diagnostic technique in medicine because of the benefits derived from low-energy radiation and the*

unprecedented level of information available from nuclear signals combine to make imaging by magnetic resonance (MR) a valuable biomedical imaging modality" (Keller, 1988, p 4).

The existence of nuclear spins, the entity essential to nuclear magnetic resonance, was known long before the first nuclear magnetic resonance (NMR) experiments were reported by Bloch and co-workers at Stanford and by Purcell, independently, at Harward in 1946. The idea to extend magnetic resonance to studies on humans was originated from Jasper Jackson, who in 1967 produced what are believed to be the first MR signals from a live animal. Lauterbur of the State University of New York at Stony Brook generated the first two-dimensional proton MR image of a water sample in 1972. Previously, Gabillard investigated one-dimensional distributions of the MR signal. Common to these experiments is the use of defined nonuniform magnetic fields, typically linear variations of the field strength along the spatial coordinates. In this manner MR signals originating from different spatial locations are discriminated. Applying these principles and Lauterbur's experiments, images were produced by other groups, and the technology rapidly advanced from imaging of small objects to, ultimately, humans.

"Unlike x-ray based modalities such as computed tomography (CT) scanning, MRI is a noninvasive technique which does not employ ionizing radiation. Further, the only tissue-specific parameter that can be determined with x-rays is electron density, which does not vary greatly from one soft tissue to another, and often necessitates the injection of contrast media". (Keller, 1988, p6)

Biologic tissue is relatively transparent to x-rays and opaque to radiation with intermediate wavelengths when proceeding from the shorter to the longer wavelengths of the electromagnetic spectrum. This is true for ultraviolet, visible, and, to some extent infrared light and microwaves. However, there is a window in tissue absorption through which radiowaves can be used to probe deep inside the human body (Wehrli, 1990).

In MRI, there is a multitude of tissue parameters that can affect MR signals. The two most significant, the relaxation times, cover a wide range of values in various normal and pathologic tissues. In addition, signal acquisition parameters can be manipulated in a variety of ways, enabling the user to control image contrast.

A rapidly alternating magnetic field at an appropriate resonant frequency in the radio frequency (RF) range, applied by a coil near the subject or specimen in the static magnetic field, changes the orientation of the nuclear spins relative to the direction of the static magnetic field as shown in Figure 2.1.

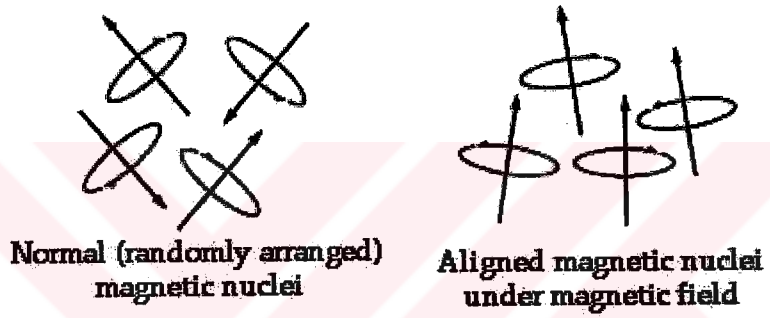


Figure 2.1 Magnetic Nuclei Normal and Under Magnetic Field

These changes are accompanied by the absorption of energy (from the alternating magnetic field) by nuclei, which undergo the transition from a lower energy state to a higher one. When the alternating field is turned off, the nuclei return to the equilibrium state, emitting energy at the same frequency as was previously absorbed. The nuclei of different elements, and even of different isotopes of the same element, have very different resonance frequencies. For a field of 0.1 T (1000 gauss), the resonance frequency of protons is 4.2 MHz and that of phosphorus is 1.7 MHz. Thus, the magnetic nuclei in the body, when placed in a static magnetic field, can be thought of as tuned receivers and transmitters of RF energy.

The principal components of the MRI machine are the core magnet, radio-frequency (rf) coils and the gradient coils. Magnet types in current use are the superconducting, resistive and permanent magnet designs ranging in strength from 0.08 to 4 T (T = 10,000 gauss). The majority of MR systems use superconducting magnets that provide fields of high strength and stability. The rf coils used to excite

the nuclei usually are quadrature coils which surround the head or body, but small (e.g. 6-10cm) flat coils placed on the surface of the head or body are also used. Besides being the essential element for spatial encoding, the gradient-coil sub-system of the MRI scanner is responsible for the encoding of specialized contrast such as flow information, diffusion information, and modulation of magnetization for spatial tagging.

The basic principle of the MRI method and the pulse sequence of a simplified MRI experiment are shown in Figure 2.2.

The spatial coding, needed for MRI, is obtained via the use of magnetic field gradients. First, a slice at height z and thickness dz of the in vivo object is selected. This is done by means of a selective 90° r.f. pulse, combined with a z-gradient. Next, the spatial coding for the x-direction is obtained by means of an x-gradient. The latter is changed with discrete steps during a fixed time tx. Finally, the spatial coding for the second direction, i.e. the y-direction, is obtained by means of a y-gradient. This gradient is not stepped, but has a fixed value during the acquisition of the FID data (in the time tx to tx+ty).

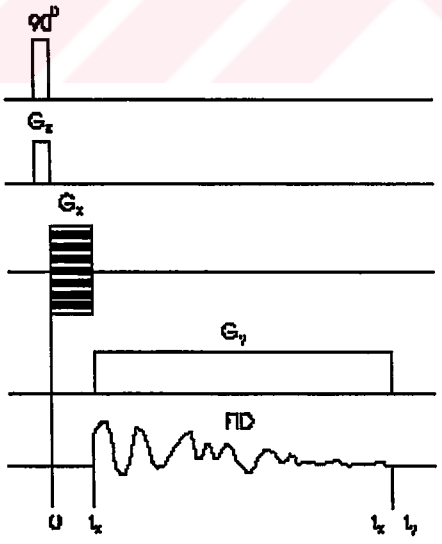


Figure 2.2: Pulse sequence of a simplified MRI experiment

The technique makes use of the fact that biological systems are composed primarily of water. Water contains hydrogen, which is just an electron orbiting a proton. Protons have associated with them a quantity called spin. Because of this, we

can think of the proton as a tiny spinning top. Protons, however, are also electrically charged. When something with an electrical charge is allowed to spin, it will have associated with it a magnetic moment. These magnetic moments can be thought of as little bar magnets. If a magnetic moment is placed into a magnetic field, it will tend to line up with the field.

The net sample magnetization is so small compared to our static field that it can't be measured while it is in the longitudinal direction. Remember that the spins are actually rotating, or processing about the static field. The angular frequency they are rotating with is called the **Armor Frequency** (Riederer, 1988), and is directly proportional to the static field strength. Imagine a field is applied perpendicular to the static field, and is oscillating at the Larmor frequency. This is called **a radio frequency (RF) pulse**. Now consider a frame of reference rotating along with this field. Our oscillating field appears stationary in this frame. Now, the only field that the spins see is the field perpendicular to the static field. So, they naturally want to line up with it, and the magnetization is thus tipped down to the plane perpendicular to the static field (the transverse plane), where we can measure it.

Once the RF pulse is turned off, the spins naturally want to go back to where they originally were. So, they will ***relax*** back to the way they were. This is **called T_1 relaxation**, where T_1 is the time it takes for the magnetization in the transverse plane to decay back to the longitudinal direction by a certain fixed amount. It may seem that the time it takes for the magnetization to disappear in the transverse plane would be the same as the time it takes for the longitudinal magnetization to reappear, but it is not. Actually, the transverse magnetization decays *before* the longitudinal magnetization grows back. This is due to another process: **T_2 relaxation**. When the spins are flipped down to the transverse plane, they begin to dephase. That means that the spins will begin to precess out of synch with one another, with some precessing faster, and some precessing slower. If there is no dephasing, the spins should be stationary in the rotating frame. If there is dephasing, the spins will precess in the rotating frame, because they are moving with angular frequencies different from the Larmor frequency. (Remember, the rotating frame is

rotating at the Larmor frequency). T_2 is the time it takes for the transverse magnetization to decay through this process by a fixed amount.

Finally, there is one more process currently used in imaging, $T1_p$ relaxation. Once the magnetization has been flipped down to the transverse plane, another RF pulse can be applied parallel to the magnetization. The spins will now begin to precess about this locking field, with a Larmor frequency dependent on the locking field strength. The locked magnetization will decay to an equilibrium value, which depends on locking field strength. The time constant for this process is T_{1p}, and it is also dependent on the locking field strength.

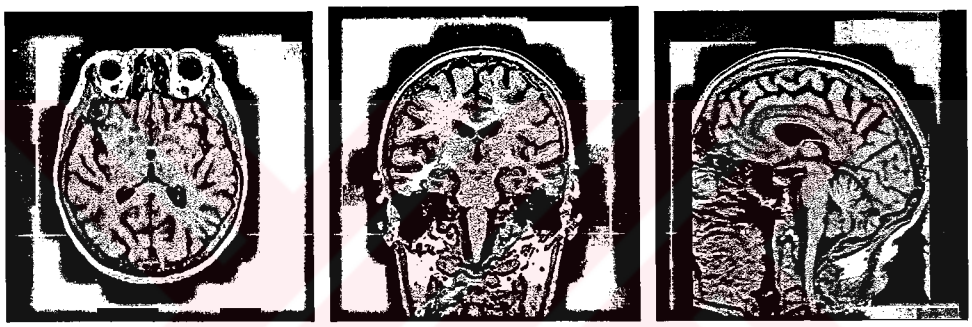


Figure 2.3 Basic Representations of MR Views: Axial, Coronal, Sagittal.

Table2.1. Magnetic resonance properties of some diagnostically relevant nuclei (Wehrli,1990).

Nucleus	Relative abundance (%)	Relative sensitivity*	Magnetogyric ratio (MHz/T)
^{1}H	99.98	1.	42.58
^{2}H	0.015	9.65×10^{-3}	6.53
^{13}C	1.11	0.016	10.71
^{19}F	100	0.830	40.05
^{23}Na	100	0.093	11.26
^{31}P	100	6.6×10^{-2}	17.23
^{39}K	93.1	5.08×10^{-4}	1.99

*At constant field for equal number of nuclei

Now that all the spins are in the transverse plane, and all the RF fields are off, a signal corresponding to the decaying magnetization can be collected. From this signal, the Larmor frequencies of the spins can be measured. The individual contributions to the signal from magnetization with different Larmor frequencies can be distinguished through mathematical techniques. (Fourier analysis of the signal). Different substances have different Larmor frequencies, and this is the basis of NMR

spectroscopy, the process by which chemists analyze the chemical composition of samples. We can apply a field gradient across our sample, making the static field different in different places, which also change the Larmor frequency. So, by analyzing signal strength for different Larmor frequencies, signal strength for different regions of the sample can be analyzed. That is how magnetic resonance images are made. Finally, different substances also have different relaxation times, and by flipping the spins around in different ways (applying pulse sequences), images with contrast based on differences in relaxation times can be created.

Table 2.2. Relaxation times for different brain tissues* (Wehrli, 1990)

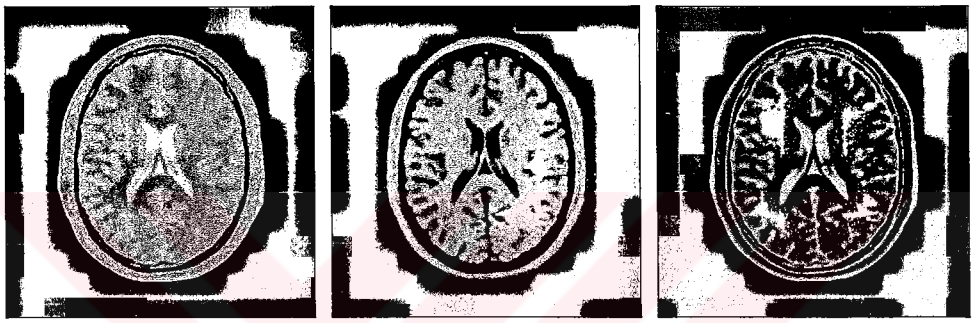
Organ	**T_1 (ms)**	**T_2 (ms)**
Putamen	747 ± 33	71 ±4
Caudate nucleus	822 ± 16	76 ±4
Thalamus	703 ± 34	75 ±4
Cortical gray matter	871 ± 73	87 ±2
Corpus callosum	509 ± 39	69 ±8
Superficial white matter	515 ± 27	74 ±5
Internal capsule	559 ±18	67 ±7
Cerebrospinal fluid (lateral ventricle)	1900 ± 353	250±3

*Determined at 1.4 T field strength from a 5 mm slice at the level of the lateral ventricles. Data are averages from three normal volunteers. (Wehrli et al, 1989)

The intensity or brightness at any given position in an MR image reflects the amplitude of the MR signal arising from that position, which, in turn is proportional to the concentration of mobile protons (proton density, abbr. N(H)) in that region (Wehrli & MacFall, 1984). If image parameters are selected which cause image intensity to reflect only variations in N(H) across the image, the image is termed a proton density image (PD). The image has a rather "flat" appearance, i.e. the intensity does not vary much from structure to structure. The reason for this appearance is that N(H) does not vary greatly in different tissues. Therefore, it is desirable to seek other sources of contrast that vary more significantly from tissue to tissue given in Table 2.2; the relaxation times T_1 and T_2 are good candidates for sources of contrast (Wehrli & MacFall, 1984).

Proton density weighted images shown in Figure 2.4 are obtained by using a long TR and a short TE $(TR = 6000ms, TE = 20ms.)$. Note the rather uniform intensity across the various brain structures.

T_2 weighting, shown in Figure 2.4, is accomplished with along TR and an intermediate TE. For both T_1 and T_2: CSF > gray matter > white matter > subcutaneous fat. For this reason CSF shows the greatest intensity and fat at least. $(TR = 4000ms, TE = 80ms.)$.



(a)PD (b) T1 (c) T2

Figure2.4. MR Images. (a) Axial, Modality=PD, Protocol=ICBM, Phantom=normal, Slice thickness=1mm, Noise=3%, INU=20%, (b) Modality=T1, Protocol=ICBM, Phantom=normal, Slice thickness=1mm, Noise=3%, INU=20%, (c) Modality=T2, Protocol=ICBM, Phantom=normal, Slice thickness=1mm, Noise=3%, INU=20% (BrainWeb Simulator).

Employment of a short TE and an intermediate TR confers T_1 weighting in Figure.2.4. $(TR = 400ms, TE = 20ms.)$. Long relaxation time structures are now hypointense. In summary, as given in Table 2.3, TR controls T_1 weighting and TE controls T_2 weighting. Short T_2 issues are dark on T_2 images, but short T_1 issues are bright on T_1 images.

Table2.3. Contrast extremes for spin echoes (Keller)

TR	TE	Contrast
$>> T_1$	$<< T_2$	PD (N(H))
$>> TR_1$	$\sim T_2$	T_2
$\sim T_1$	$<< T_2$	T_1
$\sim T_1$	$\sim T_2$	Mixed

2.3 Computed Tomography (CT)

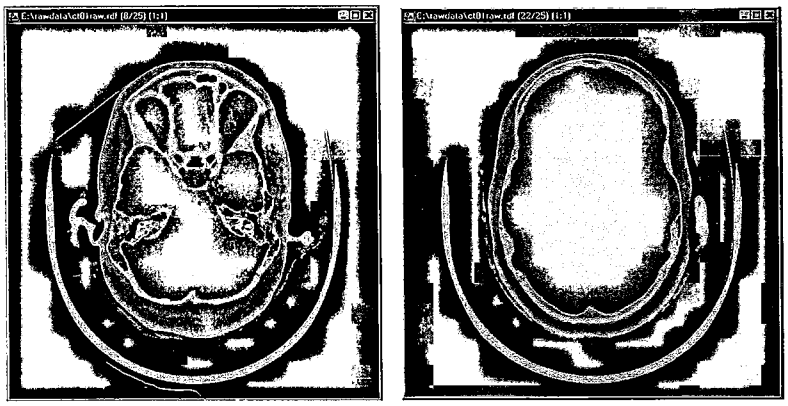


Figure 2.5. CT IMAGES 8th and 22nd slices of total 25 slices (3dctraw.rdf)

For imaging purposes, x-rays are generated by the interaction of accelerated electrons with a target material (usually tungsten). A heated cathode tube is the source of the electrons which are accelerated by applying a voltage of about 100 keV between the cathode and the tungsten anode. The x-ray tube is placed one or more meters from the patient. As the emitted x-rays pass through the patient, they are deflected by the bodily tissues and recorded by a detector on the opposite side. The most common detector is silver halide film, which is darkened by the interaction of the transmitted photons. Thus, darker areas indicate high transmission intensities while brighter areas show places of low exposure.

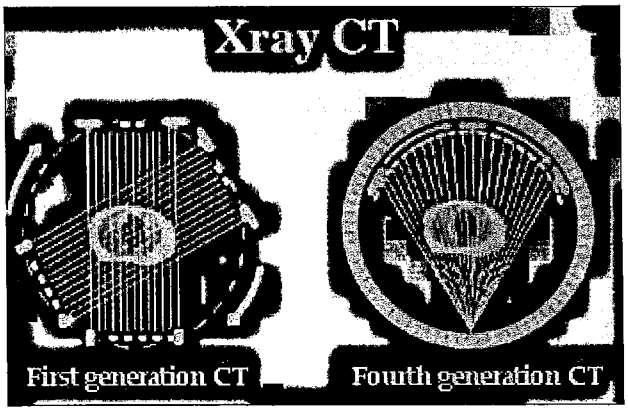


Figure 2.6 X-ray Computed Tomography

X-rays are deflected and absorbed to different degrees by the various tissues and bones in the patients' body. The amount of absorption depends on the tissue composition. For example, dense bone matter will absorb many more x-rays than soft tissues, such as muscle, fat and blood. The amount of deflection depends on the

density of electrons in the tissues. Tissues with high electron densities cause more x-ray scattering than those of lower density. Thus, since less photon reach the x-ray film after encountering bone or metal rather than tissue, the x-ray will look brighter for bone or metal.

A problem arises when using x-rays to distinguish between blood vessels and the surrounding tissues because they are very close in composition. They do not contrast enough to distinguish on the detector film. To increase image contrast, a dense fluid with elements of high atomic number ("heavy atoms" such as iodine or barium) can be injected or swallowed during the x-ray exposures. The movement through the body vasculature of a "contrast agent" such as an iodinated compound can be visualized by acquiring a sequence of x-rays. The iodine or barium agent absorbs photons more than blood and tissue because the density is higher and the elements iodine and barium have a high atomic number which gives rise to more photoelectric absorption.

Computed tomography (CT) became generally available in the mid 1970's and is considered one of the major technological advances of medical science. X-ray CT gives anatomical information on the positions of air, soft tissues, and bone. Three-dimensional imaging is achieved by rotating an x-ray emitter around the patient, and measuring the intensity of transmitted rays from different angles.

Unlike conventional x-ray transmission, x-ray CT does not use film to detect the transmitted gamma rays. Instead, an electronic device that converts the x-ray photons into an electric current collects the photons. Either a solid state device or a crystal scintillator is used to make visible light from the x-rays. Then the light photons are converted to electrons in a photoelectron multiplier tube (PMT). The result is an electric current whose magnitude is proportional to the initial energy of the x-ray photon.

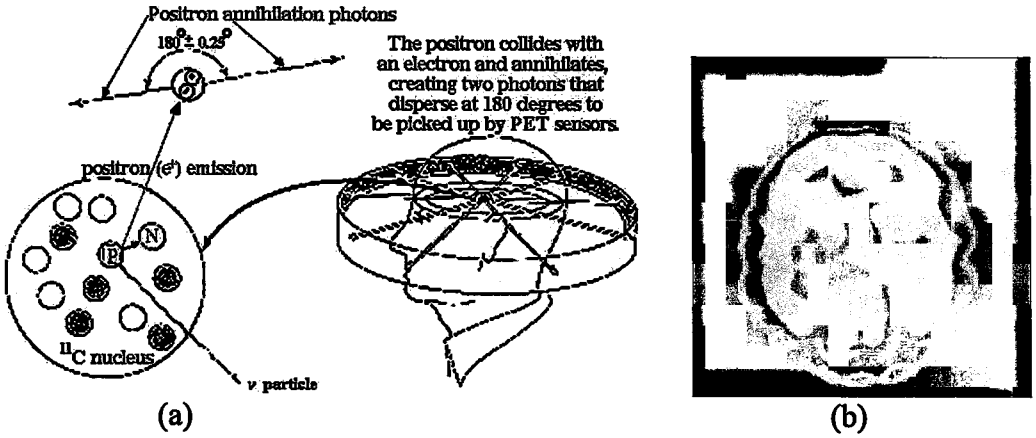
2.4 Positron Emission Tomography (PET)

The history of positron emission tomography (PET) can be traced to the early 1950's, when workers in Boston first realized the medical imaging possibilities of a

particular class of radioactive isotopes. Whereas most radioactive isotopes decay by release of a gamma ray and electrons, some decay by the release of a positron. A positron can be thought of as a positive electron. Widespread interest and an acceleration in PET technology was stimulated by development of reconstruction algorithms associated with x-ray CT and improvements in nuclear detector technologies. By the mid-1980s, PET had become a tool for medical diagnosis, for dynamic studies of human metabolism and for studies of brain activation.

PET has a million fold sensitivity advantage over other techniques used to study regional metabolism and neuroreceptor activity in the brain and other body tissues. In contrast, magnetic resonance has exquisite resolution for anatomic studies and for flow or angiographic studies. In addition, magnetic resonance spectroscopy has the unique attribute of evaluating chemical composition of tissue but in the millimolar range rather than the nanomolar range. Since the nanomolar range is the concentration range of most receptor proteins in the body, positron emission tomography is ideal for this type of imaging.

As a theory, PET imaging begins with the injection of a metabolically active tracer-a biological molecule that carries with it a positron-emitting isotope (for example, ^{11}C, ^{13}N, ^{15}O, or ^{18}F). Within minutes, the isotope accumulates in an area of the body for which the molecule has an affinity. As an example, glucose labeled with ^{11}C (half-life, 20 min), or a glucose analog labeled with ^{18}F (half-life, 1.8 hr), accumulates in the brain, where glucose is used as the primary source of energy. The radioactive nuclei then decay by positron emission. The emitted positron collides with a free electron usually within less than 1 mm from the point of emission. The interaction of the two subatomic particles results in a conversion of matter to energy in the form of two gamma rays. These high-energy gamma rays emerge from the collision point in opposite directions, and are detected by an array of detectors which surround the patient as shown in Figure 2.7.



(a) (b)

Figure 2.7. (a) Basic PET Representation and (b) Sample Image

When the two photons are recorded simultaneously by a pair of detectors, the collision that gave rise to them must have occurred somewhere along the line connecting the detectors. Of course, if one of the photons is scattered then the line of coincidence will be incorrect. After 500,000 or more annihilation events are detected, the distribution of the positron emitting tracer is calculated by tomographic reconstruction procedures. PET then reconstructs a two-dimensional image. Three-dimensional reconstructions can also be done using 2D projections from multiple angles.

The major clinical applications of PET have been in pathologic growth in brain and cancer detection of the brain, breast, heart, lung and colorectal tumors. Another application is the evaluation of coronary artery disease by imaging the metabolism of heart muscle. Contemporary instruments can scan 15cm segments of the body. Thus, with the combination of 8 full scans, a whole body image of the distribution of ^{18}F-deoxyglucose (or other common radioisotopes) can be acquired in 40 minutes; as is commonly done for evaluation of breast cancer.

Applications to the study of epilepsy, brain tumors, stroke and Alzheimer's disease have occupied the interest of neurologists for nearly 20 years. A major attribute of PET is its ability to show activity of neuroreceptors such as the dopamine, serotonin and noradrenergic receptor systems.

2.5 An Overview of Medical Image Characteristics

When we look at medical images, we see that some tissues have different contrast from the others. But, these contrast differences also depend on the image modalities such as T1 and T2. Table2.1 and Table 2.2 show an overview of these characteristics

Table 2.1 Normal Tissue Characteristics (www.med.harvard.edu/AANLIB/hms2.html).

		MR-T1[1]	MR-T2[1]	Xray-CT[2]
Normal Tissue	Dense bone	Dark	Dark	Bright
	Air	Dark	Dark	Dark
	Fat	Bright	Bright	Dark
	Water	Dark	Bright	Dark
	Brain	"Anatomic"[3]	Interm.	Interm.

1. Bright means high signal intensity, dark means low, and interm. means intermediate.
2. Bright means high density/high attenuation of x-rays, dark means low.
3. Grey matter appears gray, white matter white.

Signal in MR images is high or low (bright or dark), depending on the pulse sequence used, and the type of tissue in the image of region of interest. The following is a general guide to how tissue appears on T1- or T2- weighted images

Table 2.2 Abnormal Tissue Characteristics (www.med.harvard.edu/AANLIB/basicMR.html).

		MR-T1	MR-T2	Xray-CT	Enhancement[1]
Abnormal Tissue	Infarct	Dark	Bright	Dark	Subacute
	Bleed	Bright[2]	Bright[2]	Bright	no
	Tumor	Dark	Bright	Dark[3]	yes
	MS plaque	Dark	Bright	Dark[4]	Acute

1. Blood brain barrier leak. For MR, gadolinium; for CT, iodinated contrast material.
2. Unless very fresh or very old.
3. Unless calcified.
4. Often isodense.

Dark on T1-weighted image:

- increased water, as in edema, tumor, infarction, inflammation, infection, hemorrhage (hyperacute or chronic)
- low proton density, calcification
- flow void

Bright on T1-weighted image:

- fat
- subacute hemorrhage
- melanin

- protein-rich fluid
- slowly flowing blood
- paramagnetic substances: gadolinium, manganese, copper
- calcification (rarely)
- laminar necrosis of cerebral infarction

Bright on T2-weighted image:

- increased water, as in edema, tumor, infarction, inflammation, infection, subdural collection
- methemoglobin (extracellular) in subacute hemorrhage

Dark on T2-weighted image:

- low proton density, calcification, fibrous tissue
- paramagnetic substances: deoxyhemoglobin, methemoglobin (intracellular), iron, ferritin, hemosiderin, melanin
- protein-rich fluid
- flow void

2.6 The Surgery

Surgery is an extreme case for treatment. Some abnormal structures in any organ and any tumor or deformations can be replaced or corrected in the surgeries. All surgeries have risks depending on the organ and function. During the surgery, the patient is under different types of anesthesia. Each surgery requires a high concentration and care.

Risks start from anesthesia, causing reactions to medications, problems with breathing, continuing in the surgery with bleeding and infection. Extra risks are also possible depending on surgery area, like brain. Additional risks of brain surgery are injury to brain tissue, injury to blood vessels, nerve or muscle paralysis or weakness, loss of mental functions such as memory, speech, understanding (www.adam.com).

In this study, we are interested in brain surgeries that is the most important and most dangerous surgery type. After this point, the term "surgery" will be used for brain surgery.

Brain surgery (Craniotomy), is a surgical procedure in which the brain is accessed through the skull. The scalp is shaved, an incision is made, and a hole is drilled through the skull. A piece of the skull is removed while the brain is being operated

on, and it's replaced before the scalp is stitched closed. This operation is shown in Figure 2.8.

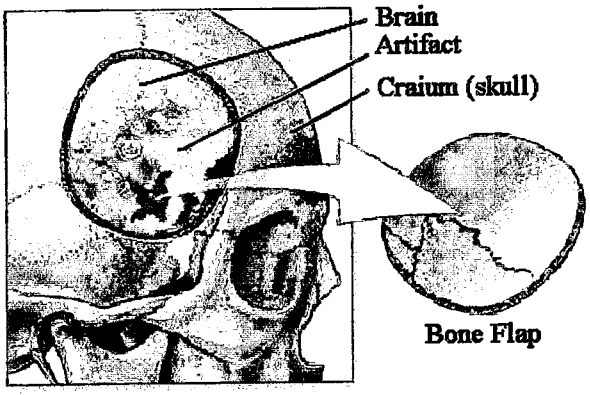


Figure 2.8 Basic Brain Surgery Procedure: Surgery to treat brain tissue through an opening (craniotomy) in the skull (cranium) (adam.com, revised).

Brain surgery may be needed to treat: Brain tumors, bleeding (hemorrhage) or blood clots (hematomas) from injuries (subdural hematoma or epidural hematomas), weaknesses in blood vessels (cerebral aneurysms), damage to tissues covering the brain (dura), pockets of infection in the brain (brain abscesses), severe nerve or facial pain (such as trigeminal neuralgia or tic douloureux), and epilepsy.

There are at least three disease processes which, if left untreated, can threaten the brain and life itself: brain tumors, aneurysms and arteriovenous malformations.

Brain tumors are abnormal growths of brain tissue that can be either benign (harmless) or malignant (possibly fatal). They often are the result of a metastasis from a cancerous disease originating from another part of the body. Symptoms of a brain tumor can include: dull, persistent headache, nausea or vomiting, problems with vision, overall physical weakness, and seizures. Not all brain tumors require surgery. Some can be treated effectively with chemotherapy and traditional modes of radiation therapy.

Maintaining adequate blood pressure to the brain requires, among other things, arteries that are not blocked and do not leak. Occasionally, the inner lining of an artery can weaken or even tear. A small pooling of blood can occur which stretches

the artery rather like a tiny balloon. This is called as an aneurysm and it can cause: headaches, visual problems, and like a balloon, it can burst.

Arteriovenous malformations (AVM) are a congenital condition, meaning that they began long before an individual's birth. They are an abnormal tangle of blood vessels in the brain which began to form when the brain was originally evolving. As an individual grows into adulthood, overall blood pressure increases, thereby weakening the walls of the abnormal vessels. As with an aneurysm, bleeding into the brain can occur. Symptoms of an AVM can include: Headaches, overall physical weakness and seizures. The malformations can be repaired through surgery. (http://www.texasneurosciences.com)

Brain

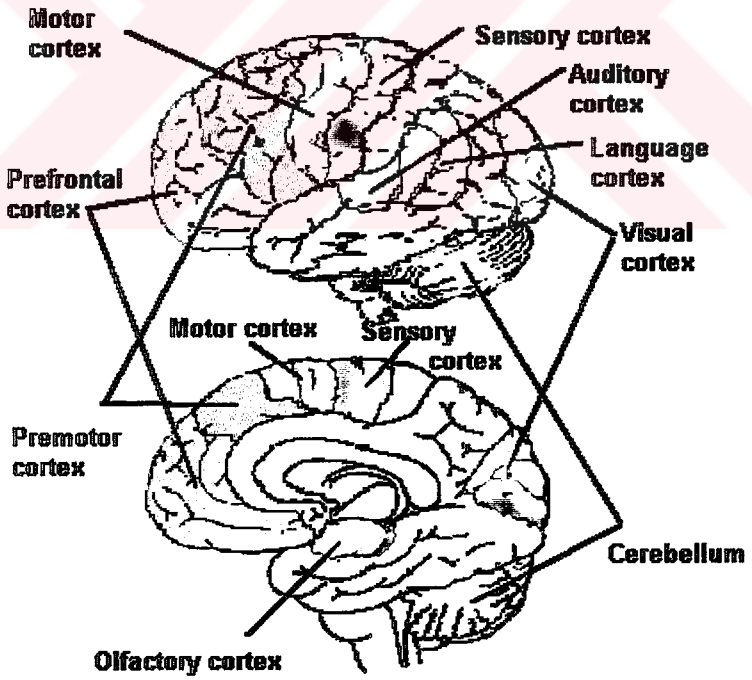


Figure 2.9 Basic Brain Structures and their Positions

Firstly, the brain is an incredibly complex organ. The brain lives apart from, and quite differently than, the rest of the body. It contains about 10 Billion (10,000,000,000) working brain cells. They are called neurons and make over 13

Trillion (13,000,000,000,000) connections with each other to form the most sophisticated organic computer on the planet, maybe even the universe. Main brain structures and their positions are shown in Figure 2.9.

Despite such complexity, most of the brain is consist of up of supporting cells. The vast majority of these are called astrocytes. These cells are the support "stuff" of the brain, and serve as a scaffold for the working brain cells and other structures. Oligodendrocytes, another type of brain cell, are much fewer in number; they are primarily responsible for making the covers (called myelin) for the vast wiring system of the brain. The ependymal cells are the fewest in number; they simply cover the inner surfaces of the brain called ventricles.

The entire brain floats in a self-contained sort of womb, and like a fetus, is surrounded by and filled with a watery fluid known as cerebrospinal fluid (CSF). These fluid spaces, when obstructed by a tumor, may enlarge and cause pressure within the closed box of the hard skull to increase dangerously. This is referred to as hydrocephalus or water-on-the-brain.

The brain has various coverings (meninges or dura), and outer pigskin shell. They hold things securely in their proper place. The cells of the meninges are unique, and some of them are capable of filtering the brain fluid (CSF) back into the bloodstream by a sort of one way valve system. They are called arachnoid cap cells.

Also, attached to the brain are a couple of hangers on. Literally, hanging beneath the brain is the Pituitary Gland, a kind of Wizard of Oz box of hormonal cells that control almost all of the body's hormonal systems. Hanging just behind the brain is a little pine cone called the Pineal Gland, the "third eye." It tells the body when it is day and when it is night via its now popular brain hormone, melatonin.

Brain Tumors

Brain tumors originate from one cell at a time and travel to other brain cells, unlike other cancers (e.g. bladder and blood cancers). So, it makes sense that the tumors of the brain occur in a frequency that corresponds directly with how many of each cell type are present in the first point of tumor.

Brain tumors can arise either from the brain itself (primary brain tumors: astrocytoma, glioblastoma, oligodendroglioma, ependymoma), or its coverings (meningiomas, pituitary tumors, pineal tumors), or the nerves at the base of the brain (acoustic neuromas, schwannomas), or even from outside the brain (metastatic brain tumors). This last case occurs when cancer cells travel through the bloodstream and lodge in the brain.

The vast majority of brain tumors are primary. Of these, the malignant astrocytoma and glioblastoma multiform are the most common, and are responsible for the bad reputation that brain tumors carry.

Important Points Regarding Primary Brain Tumors (www.brain-surgery.com):

- *Brain tumors are not cancers.*
- *Brain tumors are different!*
- *They grow only in the brain itself, and almost never travel beyond the brain.*
- *They don't metastasize. Treatment should be limited to the brain.*
- *Benign is not always "benign." Low grade gliomas, although called benign, often grow inexorably, albeit slowly.*
- *They involve the whole brain. Even though they seem to grow locally, tumor cells travel around the brain and are always found beyond the tumor margins, even on the opposite side of the brain.*
- *Benign tumors may be malignant by location -- easy tumors in tough places.*
- *True tumor margins do not exist. Total removal by local therapy (surgery, radiation, heat, cold, etc.) is not possible.*
- *The brain is immunologically isolated.*
- *The Blood Brain barrier is real.*
- *Many helpful treatments can't enter the brain via the bloodstream.*
- *Primary brain tumors are polyclonal. They are actually many tumors in one (sometimes over a thousand!)*
- *Each clone has differing sensitivity (or resistance) to anti-tumor treatments.*
- *Each clone has its own cell cycle time, doubling time, etc.*

The treatment of brain tumors is extremely difficult because of polyclonicity, the Blood Brain Barrier, the diffuse infiltrative nature of these tumors, and the perilous location of some tumors.

2.7 Computer Aided Surgery (CAS)

"Since the introduction of computer aided tomography in the 1970's, accurate visualization and measurement of the internal organs and their spatial relation to each other has been one of the main goals of medical imaging" (Shahidi, Tombropoulos, Grzeszczuk, 1998). A new trend emerging from the availability of recently developed computational resources is the increased feasibility for fast, volumetric data acquisition and reconstruction. This requires accurate high dimensional usage and analysis of the data. Additionally, some new processing algorithms such as segmentation and surface extraction methods have to be applied to raw data. These types of advances in image and volume processing are given in Chapter Three.

The developments in the technical side of the medical area easily influences the other parts such as diagnosis and operations (Toriwaki & Mori, 1999). "The usage of the high dimensional volumetric views and other progressive information in a surgery has become one of the important motivations to the start of the new field called "Computer Aided Surgery" (CAS)" (Toriwaki, Mori, 1999, p613). Interactive visualization, manipulation, and measurement of multi modality three dimensional medical images on a standard computer workstations has been developed, used, and evaluated for CAS for nearly fifteen years. These "new capabilities have provided surgeons and physicians with powerful and flexible computational support both for pre-operative surgical planning and for post-operative evaluation" (Robb, 1996).

This approach to clinical experience has fostered continual refinement of approaches and techniques, and has provided "significant information and insights related to the practical clinical usefulness of CAS and its impact on medical treatment outcome and cost" (Robb, 1996).

In CAS systems as methodology, the combination of high quality three-dimensional images and high performance but low cost computing provides efficient developments in image visualization, manipulation and measurement required for a surgery (Wieslaw & Nowinsky & Yang & Yeo, 2000).

CAS systems are also useful in minimal-access surgery and telemedicine (Green, Hill, Jensen, Shah, 1995). Minimal access surgery is also known as minimally invasive or closed surgery procedure. In this type of application, the main aim is reduction in the hospital stay and cost. But, the method has "poor eye-hand coordination afforded video display" and "the surgeon is limited by the awkwardness of maneuvering the long instruments" telemedicine (Green, Hill, Jensen, Shah, 1995). Telemedicine is other type of usage area of CAS. This term will be discussed in the section 2.10.

2.8 Dependencies for the Success Rate in the Surgery

For a good resulted surgery, some points are important. These points can be given as:

1. **Staff**: The surgeon and his team must be well educated about the surgery area and the unexpected cases.

2. **Preparation**: The surgeon must plan the operation before the surgery with the CT, MRI scans, vital signs of the patient and the possible artifacts that may occur. Our work may help in this phase.

3. **Donated Equipment**: A surgery is performed with some advanced and co-operative equipment like intra-operative guidance systems. Our work can be used as assisting equipment.

4. **Operation Time**: long operation times cause the surgeon's concentration losses and patient's vital sign difficulties. This operation time is decreased to 1-2 hours from the 10-12 hours.

5. **Diagnostic accuracy**, allows the surgeon to work precisely efficiency and reliability, each and every time. The result is that the surgical incision has been reduced to an absolute minimum. Smaller incisions, less tissue trauma,

and better surgical planning have reduced the risk of brain surgery from almost 90% in the 1940's to about 2% in the 1990's.

2.9 CAS APPLICATIONS

Sample applications and future developments in CAS are given as follows (Toriwaki & Mori, 1999).

Neurosurgical Planning

Advanced imaging techniques such as CT, MRI, SPECT, and PET have had a significant impact upon the clinical diagnosis of intra-cranial astrocytomas, and their influence upon neurosurgery and radiation therapy planning should be at least as great. Despite the demonstrated clinical utility of these modalities, there are apparent limitations in the therapeutical use of imaging data. The manner in which the spatial information is presented has hindered its full utilization in characterizing normal anatomy as well as pathological changes and their distribution. Current therapeutical approaches to intracranial tumors, such as stereotactic serial biopsy, stereotactic surgery or radiosurgery, microsurgery, and laser surgery all require a more complete delineation not only of the 3D extent of the tumor with its boundaries, but also of the relationship of the components of the complex mass (tumor, edema, necrosis, etc.) to each other and to surrounding and intervening structures such as vasculature, white matter tracts, cortical and subcoctical gray matter structures, etc. Neurosurgical treatment planning should include not only 3D display of intra-cranial anatomy but also the exact definition of coordinates and quantification of volumes of interest, and should apply various tools which allow the interactive manipulation of the data within an arbitrary reference plane or view. These tools allow interactive, control of neurosurgical procedures and are also useful for simulation and neuroanatomical or neurosurgical training.

The first computerized systems for neurosurgical planning and assistance during the actual performance of surgical procedures have already been achieved and more

complex systems are under development which provide access to information previously unavailable from cross sectional imaging data.

Laser Interventions

Laser surgery is currently used routinely in situations where the effect on tissues can be controlled by visual observation. Its use is limited when the fiber optic waveguide transmitting the laser energy is deep within tissue and the laser-induced changes are therefore invisible. To overcome this obvious restriction, we suggested the marriage of two advanced medical technologies: lasers and MRI (Jolesz & Blier & Jakab & Rucnzcl & Huttl & Jako, 1988). MRI offers several advantages; it is not only able to visualize non-invasively the target organ for laser surgical procedures deep within the tissues, but it can also demonstrate laser-induced heating and tissue changes which are secondary to the increased temperature. This technique may be applicable for ablative laser surgery in the brain and other organs, hyperthermia of tumors, or laser angioplasty. Using experimental techniques, the concept has been tested successfully in several organs, and the results have already been applied to clinical situations. This method offers the possibility to develop well-controlled laser surgery and other interventions within the MRI system, and by doing so to increase the accuracy while minimizing the invasiveness of these procedures.

Robotic Applications

Computer-guided stereotactic techniques may be improved further by the use of image-guided robotic arms or other intra-operative mechanical navigation systems. These instruments can be utilized for the positioning of biopsy needles and laser fibers. Similar principles can be used to direct focused waves or particles, such as ultrasound, microwaves, or ionizing radiation.

2.10 TECHNICAL REQUIREMENTS

Several steps are required to obtain 3D rendered data applicable for surgical planning (Kikinis et al, 1990) (Cline et al 1990):

Image Acquisition.

Cross-sectional MRI images are acquired in a two-dimensional (2D) format (slices) within a defined plane in the reference frame of the imaging *system.* To be appropriate for either surface or volume renderings to be used for surgical planning these images should fulfill the following requirements:

a) **Spatial Extent** The series of cross-sectional planes should include the entire volume of interest (usually the target plus a reasonably wide margin).

b) **Spatial Resolution.** Resolution within the imaging plane is defined by the size of the digital matrix in which the images are displayed and its smallest element (the pixel), while slice thickness defines the resolution perpendicular to the plane of the image.

c) **Tissue Contrast Resolution.** Optimization of tissue contrast is especially important in MRI in which the differentiation of various components depends upon the selection of appropriate pulse sequences.

Reformatting.

Because cross-sectional images are acquired in only a single plane, using multiple, arbitrary 2D planes intersecting at varying angles, there is better appreciation of the 3D extent of tumors and adjacent anatomic structures. The use of flexible and powerful reformatting tools provides the surgeon with some degree of interactive manipulation.

Extraction of Volumes.

Using these techniques, it is also possible to quantitate the volumes of each of the tissue components defined by the **segmentation**, automatically reconstruct the various components in 3D, and reveal their distinct spatial relationships.

Segmentation and Surface Rendering.

Segmentation, or the separation of anatomical entities from the original whole data sets, makes it possible to represent the configuration/distribution of various tissues, and to reveal the anatomical relationship of different structures. Using the

segmented 2D maps, reconstruction of the surfaces of each defined tissue element is possible.

Interactive Data Manipulation and Simulation.

Using appropriate surgical or stereotactic reference frames and various tools for data manipulation (e.g., connectivity or editing in 2D or 3D), the processed image information can be used for simulating various surgical procedures and to attempt various surgical approaches and trajectories preoperatively. Target definition for stereotactic procedures simply requires the interactive identification of the target point; therefore 2D planes are usually sufficient if they can be reformatted in arbitrary planes. For the selection of needle trajectories, the use of 3D reconstructions of anatomic and pathologic structures is necessary.

2.11 TeleMedicine and TeleSurgery

Telemedicine is a subclass of the term telepresence (Green & Hill & Jensen & Shah, 1995) (Kassel & Downs & Graves, 1997), aiming remote actions in any of application areas. If the application area is medicine, remote medicine efforts such as remote consultancy and sharing the information for diagnostic aims are called Telemedicine. Telemedicine is also divided into several subdivisions. One of these subdivisions is related to the surgery procedures. When a surgery is performed and/or affected remotely, this is called Telesurgery (Shadidi & Tombpropoulos Grzeszczuk, 1998) (Rovetta et al, 1993) or Remote Surgery.

In Telesurgery, the aim is to perform the operation in case that the surgeon and the patient are apart. The battlefield (Green, Hill, Jensen, Shah, 1995) and space may be good application areas for this purpose. By using developed robotics (Taylor et al, 1995) equipment and computer aided visualization techniques, there is no requirement for the surgeon and the patient to be together. This is also useful for the places where the surgeon cannot reach.

CHAPTER THREE
THEORETICAL BACKGROUND

3.1 Introduction

In this chapter, theoretical background related for the work done in the thesis, is given briefly. Data types and pre-processing operations for image and volume are described. Basic filtering against to noise, thresholding for extracting media from the background color are some of these operations. Then, a segmentation study, based on a spatially and statistically featured neural approach, is explained. SOM Neural Maps is used for this purpose. After segmentation, surface modeling phase is introduced. Marching Cubes Triangulation method is given for this phase.

3.2 Data Types and Volumetric Reconstruction

Two dimensional biomedical imaging modalities such as NMR and CT are acquired as "**slices**". Each slice has m x n pixels. In general, m and n are equal and take values of 256x256 or 512x512. In fact, there is no width of slice, but, since data acquisition is made in particular intervals such as 1mm and 2mm, we can assume that the width of slice is that interval in 3D condition. In 3D, the smallest volume data is called as "**voxel**" (Toriwaki & Mori, 1999).

Combination of 2D slices in 3D manner as shown in Figure 3.1, creates a volumetric representation of our data which is physically three-dimensional.

If we want to reconstruct NMR and CT slices that are acquired by the method given above, several methods are available. Direct construction is applicable but the usefulness rate is low. Instead, surface extraction from volume by using slice contouring is more acceptable. So, a surface model of tissue is obtained. This type of operation results fast, interactive visualization possibility.

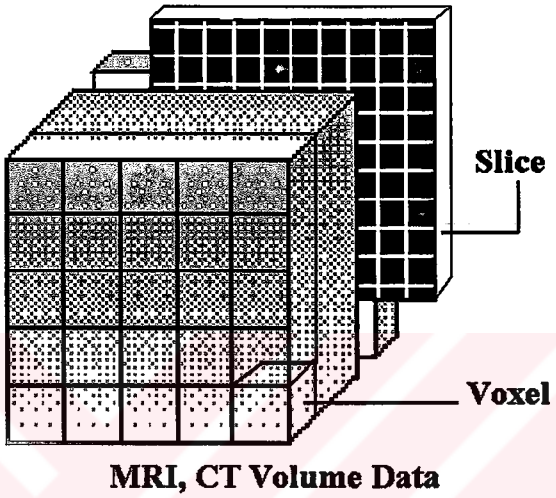


Figure 3.1. 2D and 3D data representation

3.3 Medical Data Processing Fundamentals

As a summary, a medical tomograpy image like MRI slice and one-dimensional biological signal of body like ECG must be processed after acquisition, before usage. This processing scheme consists of two main parts. The first one is "Preprocessing". In this stage, signal is filtered, thresholded and normalized, depending on the application. The second part of the processing consists of high level operations. In this stage, instead of data preparation, data is converted to a new but more understandable type, or most of the efforts are about the data analysis. Generally, the second type of operation is applied to high level medical images such as MR. For ECG and other basic biological signals, preprocessing step is enough to understand.

Basic Preprocessing algorithms in biomedical signals after acquisition are:

1. Filtering
   (a)Low Pass filtering for **noise cancellation**
   (b)High Pass filtering for **edge detection**

2. Thresholding for **noise and background elimination**
3. Tone and amplitude normalization for **making objects similar**
4. Histogram Equalization for **detection of close biological structures**

3.4 Overview of Material Classification

Biologic tissues like brain have complex structures. When we look at a MRI image of the brain, although we can recover the tissue differences with eyes, it is difficult to formulate the biologic details. The surgeon or other medical experts can differ the biologic tissues from the others by using their information and experiences. But, some of details can easily mix with the others.

The studies on the biologic materials show that every biologic tissue has its own specific properties, and some of these are close to each other. By using these biologic, statistical and spatial properties of tissues, it is possible to differ some biologic structures from the others.

Segmentation is used in conventional classification terminology. Several segmentation methods for medical tissues (Kühne& Poliwoda& Reinhart, 1997), (Talep-Ahmet & Letluc& Michel & Ayachi & Pruvo,1999), (Ma & Wang &Tsang & Tang &Xia & Tong,1998) are known. These methods are divided into three main methodologies: Automatic segmentation, (Atkins & Blair& Mackiewich,1998), Semi-automatic Segmentation, Supervised or Manual Segmentation. An anatomy expert realizes manual segmentation; generally it is a time consuming process. This approach is not applicable if the slice number of volumetric data is high, but it is the most reliable method that can be used to test other classification methods. So, semiautomatic or automatic segmentation methods are required.

Although several approaches were announced as automatic segmentation method (Ahmed & Farag, 1996), (Atkins & Mackiewich, 1998), (Shareef & Wang.& Yagel, 1999), no method is capable of fully automatic segmentation process, because every method is limited to data acquisition method, biologic structure and modality.

Segmentation algorithms use different methodologies. These are divided into five main groups: Traditional Methods, Probabilistic Methods, Fuzzy Methods and Neural Methods and combine methods. Traditional methods use region growing (Xiaohan, Yla-Jaaski, 1991,1994) (Hall et al, 1992), (Xuan & Adalı & Wang, 1995) which is based on tone information of image (Bomans & Höhne & Tiede & Riemer, 1990) and local connectivity (Joliot, Mazoyar,1992), Probabilistic methods use statistical properties of media (Caillo, Hillion, Pieczynski, 1993) (Clark et al, 1994) (Namasivaram, Hall, 1996) (Held et al, 1997) (Ma, Wang, Tsang, Tang, Xia, Tong, 1998), Fuzzy methods (Phan, Prince, 1999) use specially fuzzy C-Means and K-means algorithms. Neural methods are learning base methods (Gelenbe, Feng, 1996), (Ahmed, Farag, 1997) (Rajapakse, Giedd, Rapoport, 1997), (Cinque et al, 1998), (Shareef, Wang, Yagel, 1999). The teaching phase in the neural methods is long, but the separation of tissues by using a trained network is very short according to other methods. New approaches (Vitulano & Ruberto & Nappi, 1997), (Valova & Suganami & Kosugi, 1998) use several methods in a mixed manner at the same time (Hall et al, 1992) .These combined methods are combination of other methods such as statistical properties of probabilistic methods with neural approach. This gives maximum noticeable results. The trend goes to maximum content properties (features) and fastest classification methods. Thus, a neural approach using spatial and statistical properties of brain tissues is studied in this dissertation for the advantages of combined methods.

3.5 Neural Approach for Segmentation

3.5.1 Self organizing Maps (SOM) Neural Networks

"The network architectures and signal processes used to model nervous systems can roughly be divided into three categories, each based on a different philosophy" (Kohonen, 1990). "Feed forward networks" transform sets of input signals into sets of output signals. The desired input-output transformation is usually determined by the external, supervised adjustment of the system parameters. In "feedback networks", the input information defines the initial activity state of a feedback system, and after state transitions the asymptotic final state is identified as the

outcome of the computation. In the third category, neighboring cells in a neural network compete in their activities by means of mutual lateral interactions, and develop adaptively into specific vectors of different signal patterns. In this category learning is called competitive, unsupervised or self-organizing.

The Self-Organizing Map discussed in our study belongs to the last category of neural methods. It is a sheet-like artificial neural network, the cells of which become specifically tuned to various input signal patterns or classes of patterns through an unsupervised learning process. In the basic version, only one cell or local group of cells at a time gives the active response to the current input. The locations of the responses tend to become ordered as if some meaningful coordinate system for different input features were being created over the network. The spatial location or coordinates of a cell in the network then correspond to a particular domain of input signal patterns. Each cell or local cell group acts like a separate decoder for the same input. It is thus the presence or absence of an active response at that location, and not so much the exact input-output signal transformation or magnitude of the response, that provides an interpretation of the input information.

Self-Organizing maps or systems consisting of several map modules, have been used for tasks such as pattern recognition, robotics, process control and even processing of semantic information (Kohonen, 1990). The spatial regregation of different responses and their organization into topologically related subsets results in a high degree of efficiency in typical neural network operations.

It may be appropriate to observe here that if the maps are used for pattern recognition, their classification accuracy can be multiplied if the cells are fine-tuned using supervised learning principles (Kohonen, 1990).

It is time to demonstrate a very important phenomenon that apparently has a close relationship to the brain maps and occurs in certain spatially interacting neural networks (Kohonen, 1997). While being categorizable as a special kind of adaptation, this phenomenon is also related to regression. In regression, some simple mathematical function is usually fitted o the distribution of sample values of the input data. The "nonparametric regression" (Kohonen, 1997) considered in this

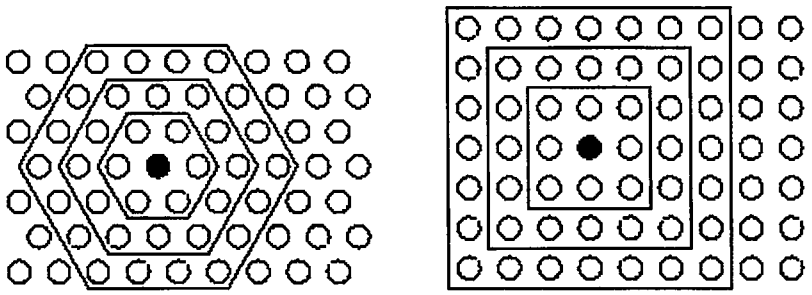
subject; however, involves fitting a number of ordered discrete reference vectors, similar to the distribution of codebook vectors to the distribution of vectoral input samples. In order to approximate continuous functions, the reference vectors are here made to define the nodes of a kind of hypothetical "elastic network" (Kohonen, 1997), whereby the topological order characteristic of this mapping, and a certain degree regularity of the neighboring reference vectors ensure from their local interactions, reflecting a kind of "elasticity". One possibility to implement such as elasticity would be to define the local interactions between the nodes in the signal space, whereas more realistic spatial interactions, from a neural modeling point of view, are definable between the neurons along the neural network. The latter approach is mainly made in this text.

Formation of a topologically ordered mapping from the signal space onto the neural network by this "regression" is already one interesting and important result, but from a practical point of view an even more intriguing result is that the various neurons develop into specific decoders or detectors of their respective signal domains in the input space (Kohonen, 1997). These decoders are formed onto the network in a meaningful order, as if some features coordinate system were defined over the network).

The Self Organizing Map (SOM) algorithm (Kohonen, 1997) that is the main subject of our neural segmentation network defines the process in which such mappings are formed. The "feature maps" realized can thereby often effectively be used for the preprocessing of patterns for their recognition, or, if the neural network is a regular two-dimensional array, to project and visualize high-dimensional signal spaces, on such a two-dimensional display. As a theoretical scheme, on the other hand, the adaptive SOM processes (Ahmed & Farag, 1997), in a general way, may explain the organizations found in various brain structures.

3.5.2 SOM Algorithm

The self-organizing phenomenon is first demonstrated in an abstract system, without reference to any biological structures or signal types.



(a) Hexagonal grid (b) Rectangular grid

Fig 3.2 Two examples of topological neighborhood *Neighborhoods of the unit marked with black dot: (a) hexagonal lattice, (b) rectangular lattice. The innermost polygon corresponds to 1-neighborhood, the second to the 2-neighborhood and so on.*

Consider figure 3.2. The SOM here defines a mapping from the input data space R^n onto a two-dimensional array of nodes. With every node i, a parametric reference vector

$$m_i = [\mu_{i1}, \mu_{i2}, \ldots, \mu_{in}]^T \in R^n \tag{3.1}$$

is associated. The lattice type of the array can be defined to be rectangular, hexagonal, or even irregular; hexagonal is effective for visual display. In the simplest case, an input vector

$$x = [\xi_1, \xi_2, \ldots, \xi_n]^T \in R^n \tag{3.2}$$

is connected to all neurons in parallel via variable scalar weights μ_{ij}, which are in general different for different neurons. In an abstract scheme it may be imagined that the input x, by means of some parallel computing mechanisms, is compared with all the m_i, and the location of the best match in some metric is defined as the location of the "response". By computer programming, of course, location of the best match is a trivial task. The exact magnitude of the response need not be determined: the input is simply mapped onto this location, like in a set of decoders.

Let $x \in R^n$ be a stochastic data vector. One might then say the SOM is a "nonlinear projection" of the probability density function $p(x)$ of the high dimensional input data vector onto the two-dimensional display. Vector x may be

compared with all the m_i in any metric; in many practical applications, the smallest of the Euclidian distances $\|x - m_i\|$ can be made to define the best-matching node, signified by the subscript c;

$$c = \arg\min_i \{\|x - m_i\|\} \tag{3.3}$$

which means the same as

$$\|x - m_c\| = \min_i \{\|x - m_i\|\} \tag{3.4}$$

During learning, or the process in which the "nonlinear projection" is formed, those nodes that are topographically close in the array up to a certain geometric distance will activate each other to learn something from the same input x. This will result in a local relaxation or smoothing effect on the weight vectors of neurons in this neighborhood, which in continued learning leads to a global ordering. Consider the eventual convergence limits of the following learning process, whereby the initial values of the $m_i(0)$ can be arbitrary, e.g. random:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \tag{3.5}$$

where $t = 0,1,2,\ldots$ is an integer, the discrete-time coordinate. In the relaxation process, the function $h_{ci}(t)$ has a very central role: it acts as the so-called neighborhood function, a smoothing kernel defined over the lattice points. For convergence it is necessary that $h_{ci}(t) \to 0$ *when* $t \to \infty$. Usually,

$$h_{ci}(t) = h(\|r_c - r_i\|, t) \tag{3.6}$$

where $r_c \in R^2$ and $r_i \in R^2$ are the location vectors of the nodes c and i, respectively, in the array. With increasing

$$\|r_c - r_i\|, h_{ci} \to 0 \tag{3.7}$$

The average width and form of h_{ci} defines the "stiffness" of the "elastic surface" to be fitted to the data points.

In the literature, two simple choices occur for $h_{ci}(t)$ frequently. The simpler of them refers to a neighborhood set of array point around the node c (fig 3.2). Let their index set be denoted by N_c (notice that we can define $N_c = N_c(t)$ as a function of time), whereby:

$$h_{ci}(t) = \alpha(t) \quad if \quad i \in N_c \text{ and } h_{ci}(t) = 0 \quad if \quad i \notin N_c . \tag{3.8}$$

The value of $\alpha(t)$ is then identified with a learning rate factor $(0 < \alpha(t) < 1)$. Both $\alpha(t)$ and $N_c(t)$ are usually decreasing monotonically in time (during the ordering process).

Another widely applied, smoother neighborhood kernel can be written in terms of Gaussian function,

$$h_{ci} = \alpha(t) \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)} \right) \tag{3.9}$$

where $\alpha(t)$ is another scalar-valued "learning-rate factor", and the parameter $\sigma(t)$ defines the width of the kernel; the latter corresponds to the radius of the $N_c(t)$ above. Both $\alpha(t)$ and $\sigma(t)$ are some monotonically decreasing function of time.

The algorithm chosen here for preliminary simulations is only representative of many alternative forms. If the SOM network is not very large (say, a few hundred nodes at most), selection process parameter is not very crucial. We can also use the simple neighborhood-set defined form of $h_{ci}(t)$.

Special caution, however, is required in the choice of the size of $N_c = N_c(t)$. If the neighborhood is too small to start with, the map will not be ordered globally. Instead various kinds of mosaic-like parcellations of the map are seen, between which the ordering direction changes discontinuously. This phenomenon can be

avoided by starting with a fairly wide $N_c = N_c(0)$ and letting it shrink with time. The initial radius of N_c can be more than a half diameter of the network! During the first 1000 steps or so, when the proper ordering takes place, and $\alpha = \alpha(t)$ is fairly large, the radius of N_c can shrink linearly to, say, one unit; during the fine-adjustment phase, N_c can still contain the nearest neighbors of cell *c*.

For very large maps, however, selection of optimal $\alpha(t)$ law may be crucial for convergence. Effective choices for these functions and their parameters have so far only been determined experimentally.

Optimizing the learning Rate

As mentioned earlier, selection of optimal learning-rate factor is important for successful operation. If we generalize the recursive learning rules (Kohonen, 1997), we have to define an individual $\alpha_i(t)$ for each m_i and write:

$$m_i(t+1) = m_i(t) + \alpha_i(t) h_{ci}(t)[x(t) - m_i(t)] \tag{3.10}$$

where h_{ci} is time-invariant, we may update $\alpha_i(t)$ whenever a correction to the m_I values is made, this correction is:

$$\alpha_i(t+1) = \frac{\alpha_i(t)}{1 + h_{ci}\alpha_i(t)} \tag{3.11}$$

also this expression was written as

$$\alpha(t) = \frac{A}{t + B} \tag{3.12}$$

by Mulier and Cherkassky, here A and B are suitable chosen constants.

3.5.3 SOM Implementation

The essence of Kohonen's SOM algorithm is that it substitutes a simple geometric computation for the more detailed properties of the Hebbian-like rule and lateral interactions (Haykin, 1999). The parameters of the algorithm are:

- *A continuous input space of activation patterns that are generated in accordance with a certain probability distribution.*
- *A topology of the network in the form of a lattice of neurons that defines a discrete output space.*
- *A time-varying neighborhood function* $h_{j,i(x)}(n)$ *that is defined around a winning neuron* $i(x)$.
- *A learning-rate parameter* $\eta(n)$ *that starts at an initial value of* η_0 *and then decreases gradually with time, n, but never goes to zero.*

For neighborhood function and learning-rate parameter, we may use equation 3.9 for the ordering phase (i.e. the first thousand iterations or so). For good statistical accuracy, $\eta(n)$ should be maintained at a small value (0.01 or less) during the convergence for a fairly long period of time, which is typically thousands of iterations. As for the neighborhood function, it should contain only the nearest neighbors of the winning neuron at the start of the convergence phase, and may eventually shrink to one or zero neighboring neurons (Haykin, 1999).

There are three basic steps involved in the application of the algorithm after initialization: sampling, similarity matching and updating. These three steps are repeated until formation of the feature map has completed. The algorithm is summarized as follows:

1. **Initialization.** Choose random values for the initial weight vectors $w_j(0)$. The only restriction here is that the $w_j(0)$ be different for $j = 1,2,..,l$ where l is the number of neurons in the lattice. It may be desirable to keep the magnitude of the weights small. Another way of initializing the algorithm is to select the weight vector $\{w_j(0)\}_{j=1}^{l}$ from the available set of input vectors $\{x_i\}_{i=1}^{N}$ n a random manner.

2. **Sampling.** Draw a sample x from the input space with a certain probability; the vector x represents the activation pattern that is applied to the lattice. The dimension of the vector x is equal to m.

3. **Similarity Matching.** Find the best-matching (winning) neuron $i(x)$ at the time step n by using the minimum distance Euclidian criterion in Equation 3.3.

4. **Updating.** Adjust the synaptic weight vectors of all neurons by using the update formula in Equation 3.5, where $\eta(n)$ is the learning rate parameter, and $h_{j,i(x)}(n)$ is the neighborhood function centered around the winning neuron $i(x)$; both $\eta(n)$ and $h_{j,i(x)}(n)$ are varied dynamically during the learning for the best results.

5. **Continuation.** Continue with step 2 until no noticeable changes in the features map are observed.

About the initialization of SOM

There are 3 types of initialization schemes in SOM: Random initialization, Usage initial samples, and linear initialization. In Random initialization, we use arbitrary values for codebook vectors $m_i(0)$. Random initialization is best and fastest policy (Kohonen, 1997), and prevent the long runs in standard initialization. Usage of initial values, the aim is to pick the $m_i(0)$ from the available input samples x(t), preferable in a random order. The third one is linear initialization, as the $m_i(0)$ can be arbitrary and identified with the array points.

Construction of a Good Maps

According to the Kohonen (Kohonen, 1997), for stable, well-oriented and least ambiguous map, we need some precautions:

1. *Be careful on forming of the array (circular, hexagonal or rectangular)*
2. *Apply a Leaning with a small number of available training samples*
3. *Enhance the rare cases*

4. *Scale the pattern components*
5. *Force representations to a wanted place on the map*
6. *Monitor of the quality of learning.*

3.6 Polygonizing a Scalar Field

A natural extension to color mapping is contouring. When we see a surface colored with data values, the eye often separates similarly colored areas into distinct regions. When we contour data, we are effectively constructing the boundary between these regions. These boundaries correspond to contour lines (2D) or surfaces (3D) of constant scalar value.

"Examples of 2D contour displays include weather maps annotated with lines of constant temperature (isotherms), or topological maps drawn with lines of constant elevation. Three-dimensional contours are called isosurfaces, and can be approximated by many polygonal primitives intersection" (Schroeder, Martin, Lorensen, 1998). Examples of isosurfaces include constant medical image intensity corresponding to body tissues such as skin, bone, or other organs. Other isosurfaces such as surfaces of constant pressure or temperature also may be created.

3.6.1 Isosurface

> *"Isosurface extraction and visualization represents one of the most effective and powerful techniques for the investigation of volumetric scalar data sets. In the same way as contour lines promote the understanding of a digital terrain model, isosurfaces improve the comprehension of the phenomenon, process or structure that the scalar field represents"* (Cignoni, Montani, Puppo, Scopigno, 1995 p31) .

The problem of extraction of isosurfaces from a volumetric data set can be formally stated as follow. A scalar volume data set is a pair (V,W), where $V = \{v_i \in R^3, i = 1,\ldots,n\}$ is a finite set of points in a domain $\Omega \subset R^3$; $W = \{w_i \in R, i = 1,\ldots,n\}$ is the corresponding set of scalar values, obtained by

measuring a tri-variate scalar field $f(x,y,z)$ in the sampling points in V, i.e. $w_i = f(v_i)$.

A digital interpolating model for volume data sets (V,W), is a pair (Σ,Φ) where Σ is a subdivision of Ω into cells $\sigma_1,\sigma_2,\ldots,\sigma_m$, and Φ is a corresponding family of real tri-variate functions $\phi_j : \sigma_j \to R, j = 1,\ldots,m$, which interpolate the values of W at all points of V. If functions of Φ are coincident at the common boundary of adjacent cells of Σ, then a continuous function ϕ is defined piecewise by Φ on Σ:

$$\phi(p) = \phi_j(p) \quad if \quad p \in \sigma_j, \quad \forall j = 1,\ldots,m \tag{3.13}$$

Function ϕ is in practice an estimate of the (otherwise unknown) measured function f over Ω.

Given $q \in R$, the set,

$$S(q) = \{p \in \Omega \,|\, \phi(p) = q\} \tag{3.14}$$

is called an **isosurface** (Cignoni, Montani, Puppo, Scopigno, 1995) of ϕ at value q. If ϕ is continuous, and q is not an extreme value of ϕ, S(q) is a 2-manifold embedded in R3, possibly with a one-dimensional boundary contained in the boundary of Ω. S(q) is defined piecewise on the cells of Σ: each cell $\sigma_j \in \Sigma$ such that $\min_{\sigma_j} \phi_j \le q \le \max_{\sigma_j} \phi_j$ is called active at q, and it contributes S(q) for a patch corresponding to the locus of points:

$$S_j(q) = \{p \in \sigma_j \,|\, \phi_j(p) = q\} \tag{3.15}$$

The isosurface extraction problem consists in finding all patches $S_j(q)$ that correspond to active cells, given (Σ,Ω) and q. Traditional methods for isosurface extraction such as the Marching Cubes (Lorensen, Cline, 1987) , and its straightforward generalization to tetrahedral models, namely the Marching

Tetrahedra- analyze in turn every cell of Σ, and for each active cell compute the corresponding isosurface patch. Hence, the computational complexity of such techniques is linear in the size of Σ.

3.6.2 Isosurface Extraction

Consider the 2D structured grid shown in Figure 3.3. Scalar values are shown next to the points that define the grid. Contouring always begins by selecting a scalar value, or contour value, that corresponds to the contour lines or surfaces generated. To generate the contours, some form of interpolation must be used. This is because we have scalar values at a finite set of points in the dataset, and our contour value may lie between the point values. Since the most common interpolation technique is linear, we generate points on the contour surface by linear interpolation along the edges. If an edge has scalar values 5 and 0 at its two end points, and if we are trying to generate a contour line of value 5, then edge interpolation computes that the contour passes through the midpoint of the edge.

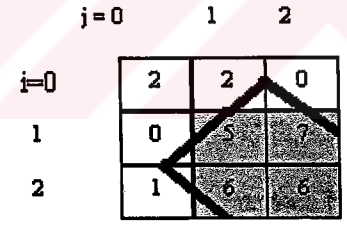


Figure 3.3: Isosurface with the isovalue=5.

Once the points on cell edges are generated, we can connect these points into contours using a few different approaches. One approach detects an edge intersection (Schroeder, Martin, Lorensen, 1998), and then "tracks" this contour as it moves across cell boundaries. We know that if a contour edge enters a cell, it must exit a cell as well.

Another approach uses a "divide and conquer" technique intersection (Schroeder, Martin, Lorensen, 1998), treating cells independently. This is the "Marching Squares" algorithm in 2D, and "Marching Cubes" (Lorensen, Cline, 1987) in 3D. The basic assumption of these techniques is that a contour can only pass through a cell in

a finite number of ways. A case table is constructed that enumerates all possible topological states of a cell, given combinations of scalar values at the cell points. The number of topological states depends on the number of cell vertices, and the number of inside/outside relationships a vertex can have with respect to the contour value. A vertex is considered inside a contour if its scalar value is larger than the scalar value of the contour line. Vertices with scalar values less than the contour value are said to be outside the contour.

For example, if a cell has four vertices and each vertex can be either inside or outside the contour, there are $2^4 = 16$ possible ways that the contour passes through the cell. In the case table we are not interested in where the contour passes through the cell (e.g., geometric intersection), just how it passes through the cell (topology of the contour in the cell).

As mentioned previously, the 3D analogy of marching squares is marching cubes. Here, there are 256 different combinations of scalar value, given that there are eight points in a cubical cell (i.e., 2^8 combinations). Figure 3.4 shows these combinations reduced to 15 cases by using arguments of symmetry. We use combinations of rotation and mirroring to produce topologically equivalent cases.

3.6.3 Triangulation Methods

Mainly, there are two types of triangulation (Held 1998) methods; the first one is based on Quickhull Algorithm for convex hulls (Barber, Dokin, huhdanpaa, 1996) and the method is called as Delaunay Triangulation (Mohan & Lee, 1993) (Guha 1994) (Toriwaki & Mori, 1999). This method uses voronoi diagrams. The second method is based marching cubes method (Lorensen, Cline, 1987) and Discretized Marching Cubes (Montani & Seateni & Scopigno, 1994) algorithms. This fast method uses look up tables to determine polygon surfaces.

3.6.4 Marching Cubes Algorithm for Polygonizing A Surface

Marching cube implementation is relatively easier than other polygonizing algorithms. Typically, two main look at tables, edge table and triangle table, are used. Edge table is for determining the edge settlement of the cube shown in

figure.4.16 after detection of the vertex value is above or below the isosurface. Triangle table is used for decision of which triangles are created for the selected cube. The general algorithm for the marching cubes is given below.

1. Select a cell.
2. Calculate the inside / outside state of each vertex of the cell.
3. Create an index by storing the binary state of each vertex in a separate bit.
4. Use the index to look up the topological state of the cell in a case table.
5. Calculate the contour location by using linear interpolation for each edge in the case table.

This procedure will construct independent geometric primitives in each cell. At the cell boundaries duplicate vertices and edges may be created. These duplicates can be eliminated by using a special coincident point-merging operation. Note that interpolation along each edge should be done in the same direction. If not, numerical round-off will likely cause points to be generated that are not precisely coincident, and will not merge properly.

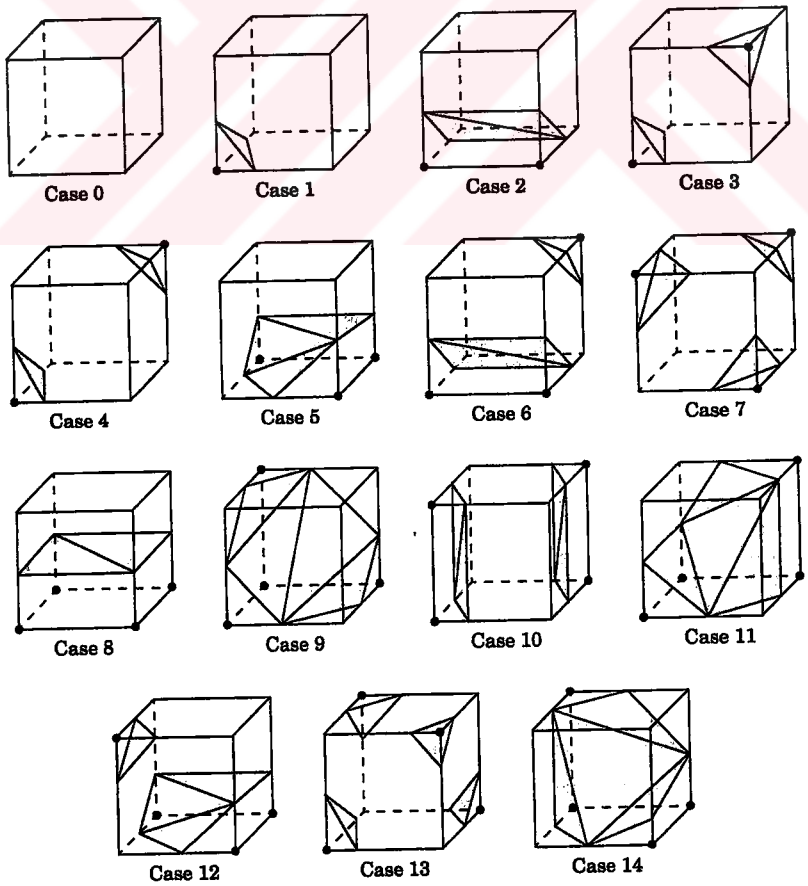


Figure 3.4: Marching Cubes Possibly 15 cases of 256 by symmetry (Shroeder, Martin, Lorensen, 1998).

There are advantages and disadvantages to both the edge-tracking and marching cubes approaches. The marching squares algorithm is easy to implement. This is particularly important when we extend the technique into three dimensions, where isosurface tracking becomes much more difficult. On the other hand, the algorithm creates disconnected line segments and points, and the required merging operation requires extra computation resources. The tracking algorithm can be implemented to generate a single polyline per contour line, avoiding the need to merge coincident points.

An important issue is contouring ambiguity (Matveyev, 1994). Careful observation of marching squares cases numbered 5 and 10 and marching cubes cases numbered 3, 6, 7, 10, 12, and 13 show that there are configurations where a cell can be contoured in more than one way (Shroeder, Martin, Lorensen, 1998). This ambiguity also exists when using an edge tracking approach to contouring. Contouring ambiguity arises on a 2D square or the face of a 3D cube when adjacent edge points are in different states, but diagonal vertices are in the same state.

In two dimensions, contour ambiguity is simple to treat: for each ambiguous case we implement one of the two possible cases. The choice for a particular case is independent of all other choices. Depending on the choice, the contour may either extend or break the current contour.. Either choice is acceptable since the resulting contour lines will be continuous and closed (or will end at the dataset boundary).

In three dimensions the problem is more complex. We cannot simply choose an ambiguous case independent of all other ambiguous cases. For example Figure 3.5 shows what happens if we carelessly implement two cases independent of one another. In this figure we have used the usual case 3 but replaced case 6 with its complementary case. Complementary cases are formed by exchanging the isovalue above, and the isovalue below the isosurface. The result of pairing these two cases is that a hole is left in the isosurface.

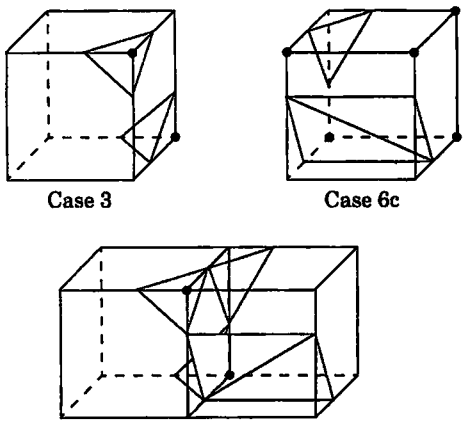


Figure 3.5 Choosing arbitrary polygons causes holes

(Shroeder, Martin, Lorensen, 1998).

Several different approaches have been taken to remedy this problem (Matveyev, 1994) (Kenmochi & Kotani & Imiya, 1999) (Montani & Scateni & Scopigno, 2000). One approach tessellates the cubes with tetrahedron, and uses a marching tetrahedral technique (Chan & Purisima, 1997). This works because the marching tetrahedra exhibit no ambiguous cases. Unfortunately, the marching tetrahedra algorithm generates isosurfaces consisting of more triangles, and the tessellation of a cube with tetrahedra requires making a choice regarding the orientation of the tetrahedra. This choice may result in artificial "bumps" in the isosurface because of interpolation along the face diagonals as shown in Figure 3.5. Another approach evaluates the asymptotic behavior of the surface, and then chooses the cases to either join or break the contour. Nielson and Hamann (Nielson, Hamann, 1991) have developed a technique based on this approach they call the asymptotic decider. It is based on an analysis of the variation of the scalar variable across an ambiguous face. The analysis determines how the edges of isosurface polygons should be connected.

A simple and effective solution extends the original 15 marching cubes cases by adding additional complementary cases. These cases are designed to be compatible with neighboring cases and prevent the creation of holes in the isosurface. There are six complementary cases required, corresponding to the marching cubes cases 3, 6, 7, 10, 12, and 13. The complementary marching cubes cases are shown in Figure 3.6.

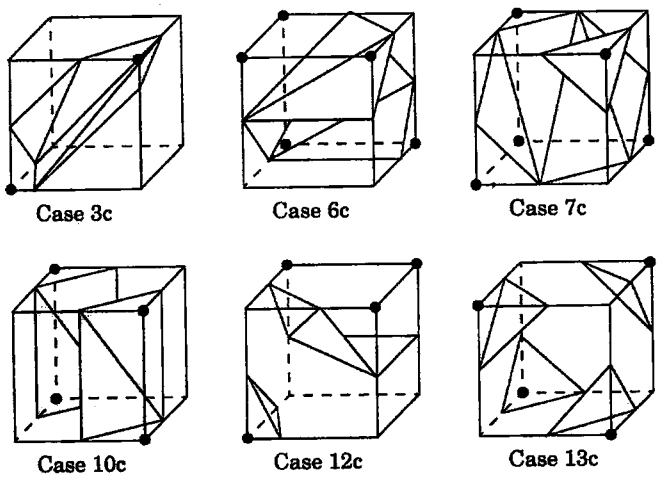


Figure 3.6: Complementary cases (Shroeder, Martin, Lorensen, 1998)

3.7 OpenGL Graphics Library

OpenGL (``GL" for ``Graphics Library") provides advanced rendering features in either immediate mode or display list mode. While OpenGL is a relatively new standard, it is very similar in both its functionality and its interface to Silicon Graphics' IRIS GL, and there are many successful 3D applications that currently use IRIS GL for their 3D rendering.

Like the graphics systems already discussed, OpenGL is a software interface to graphics hardware. The interface consists of a set of several hundred procedures and functions that allow a programmer to specify the objects and operations involved in producing high-quality graphical images, specifically color images of three-dimensional objects. Like PEX, OpenGL integrates 3D drawing into X, but can also be integrated into other window systems (e.g. Windows/NT) or can be used without a window system.

OpenGL draws *primitives* into a frame buffer subject to a number of selectable modes. Each primitive is a point, line segment, polygon, pixel rectangle, or bitmap. Each mode may be changed independently; the setting of one does not affect the settings of others (although many modes may interact to determine what eventually

ends up in the frame buffer). Modes are set, primitives specified, and other OpenGL operations described by sending *commands* in the form of function or procedure calls.

Geometric primitives (points, line segments, and polygons) are defined by a group of one or more *vertices*. A vertex defines a point, an endpoint of an edge, or a corner of a polygon where two edges meet. Data (consisting of positional coordinates, colors, normals, and texture coordinates) are associated with a vertex and each vertex is processed independently, in order, and in the same way. The only exception to this rule is if the group of vertices must be *clipped* so that the indicated primitive fits within a specified region; in this case vertex data may be modified and new vertices created. The type of clipping depends on which primitive group of vertices represents.

OpenGL provides direct control over the fundamental operations of 3D and 2D graphics. This includes specification of such parameters as transformation matrices, lighting equation coefficients, antialiasing methods, and pixel update operators. It does not provide a means for describing or modeling complex geometric objects. Another way to describe this situation is to say that OpenGL provides mechanisms to describe how complex geometric objects are to be rendered rather than mechanisms to describe the complex objects themselves.

The model for interpretation of OpenGL commands is client-server. That is, a program (the client) issues commands, and these commands are interpreted and processed by OpenGL (the server). The server may or may not operate on the same computer as the client.

OpenGL is strictly defined as "a software interface to graphics hardware." In essence, it is a 3D graphics and modeling library that is extremely portable and very fast. Using OpenGL, you can create elegant and beautiful 3D graphics with nearly the visual quality of a ray-tracer. The greatest advantage of using OpenGL is that it is orders of magnitude faster than a ray-tracer. It uses algorithms carefully developed and optimized by Silicon Graphics, Inc. (SGI), an acknowledged world leader in computer graphics and animation.

OpenGL is used for a variety of purposes, from CAD engineering and architectural applications to computer-generated dinosaurs in blockbuster movies. The introduction of an industry standard 3D API to a mass-market operating system such as Microsoft Windows has some exciting repercussions. With hardware acceleration and fast PC microprocessors becoming commonplace, 3D graphics will soon be a typical component of consumer and business applications, not just of games and scientific applications.

Let's take a look at OpenGL's origins, who's "in charge" of OpenGL, and where OpenGL is going. We'll also examine the principles of OpenGL implementation.

OpenGL is the result of SGI's efforts to improve IRIS GL's portability. The new language would offer the power of GL but would be "Open," allowing for easier adaptability to other hardware platforms and operating systems. (SGI still maintains IRIS GL, but no enhancements or features other than bug fixes are being made.)

On July 1, 1992, Version 1.0 of the OpenGL specification was introduced. Just five days later, at the very first Win32 developer's conference, SGI demonstrated OpenGL running on their IRIS Indigo hardware. Video clips from films such as *Terminator Two: Judgment Day*, and medical imaging applications were popular attractions in the vendor exhibit hall. Already, SGI and Microsoft were working together to insert OpenGL into a future version of Windows NT.

An open standard is not really open if only one vendor controls it. Thus, all enhancements to OpenGL are decided by the OpenGL Architecture Review Board (ARB), whose founding members are SGI, Digital Equipment Corporation, IBM, Intel, and Microsoft. The OpenGL ARB meets twice a year. In December 1995 the ARB ratified the final specification for Version 1.1 of OpenGL.

OpenGL is a procedural rather than a descriptive graphics language. Instead of describing the scene and how it should appear, the programmer actually describes the steps necessary to achieve a certain appearance or effect. These "steps" involve calls to a highly portable API that includes approximately 120 commands and functions. These are used to draw graphics primitives such as points, lines, and polygons in

three dimensions. In addition, OpenGL supports lighting and shading, texture mapping, animation, and other special effects.

OpenGL does not include any functions for window management, user interaction, or file I/O. Each host environment (such as Microsoft Windows) has its own functions for this purpose and is responsible for implementing some means of handing over to OpenGL the drawing control of a window or bitmap.

Direct Draw is optimized for direct access to graphics hardware. It bypasses the GDI completely and talks directly to the graphics hardware with perhaps only a thin hardware abstraction layer in between, and some software emulation for unsupported features. Direct Draw is typically used for games and allows direct manipulation of graphics memory for ultrafast 2D graphics and animation.

3.8 Augmented Reality (AR)

Augmented Reality (AR) – a technology that "presents a virtual world that enriches, rather than replaces the real world" (Brys, 1992). This is achieved by means of see-through HMD that superimposes virtual three-dimensional objects on real ones. Augmented reality refers to the combination the real and the virtual to assist the user in his environment. Applications include telemedicine, architecture, construction, devices for the disabled, and many others.

In the spirit of the first see-through head-mounted display developed by Sutherland (Sutherland, 1968), we and other researchers (e.g., Robinett, 1992; Caudell & Mizell, 1992; Bajura & Neumann, 1995) use the term *augmented reality* to refer to enrichment of the real world with a complementary virtual world. The augmented reality systems we are developing employ a see-through head-worn display that overlays graphics and sound on a person's naturally occurring sight and hearing.

By tracking users and objects in space, these systems provide users with visual information that is tied to the physical environment. Unlike most virtual realities,

whose virtual worlds replace the real world, augmented reality systems enhance the real world by superposing information onto it (From the Web page: www.cc.columbia.edu /cu/gsapp/BT/RESEARCH/ SPACE-FR/testbed.html).

Augmented Reality (AR) is a variation of *Virtual Environments* (VE), or Virtual Reality (VR) as it is more commonly called (Azuma, 1995). VE technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composite with the real world. Therefore, AR supplements reality, rather than completely replacing it. Ideally, it would appear to the user that the virtual and real objects coexisted in the same space.

Some researchers feel that AR requires the use of Head-Mounted Displays (HMDs) and other display devices. Three desired characteristics of an AR system are (Azuma, 1995): 1) Combination of real and virtual world. 2) Interactive operation in real time 3) Registering in 3-D. Fundamental application areas of AR systems are summarized as below:

Figure 3.7. Augmented Reality (AR) example: Finger Tracking (http://wearables.www.media.mit.edu/projects/wearables/index.html)

1. **Medicine:** Medical Doctors and surgeons could use Augmented Reality as visualization and training aid for surgery. It may be possible to collect 3-D datasets of a patient in real time, using non-invasive techniques like Magnetic Resonance Imaging (MRI), Computed Tomography scans (CT), or ultrasound imaging. These

datasets could then be rendered and combined in real time with a view of the real patient. In effect, this would give the surgeon "X-ray Vision" inside a patient. This would be very useful during minimally-invasive surgery, which reduces the trauma of an operation by using small incisions.

2. **Manufacturing and Repair**: Another potential application area of Augmented Reality is the assembly, maintenance, and repair of complex machineries.

3. **Annotation and Visualization:** AR could be used to annotate objects and environments with public or private information. Applications using public information assume the availability of public databases to draw upon.

4.**Robot Path Planning:** Teleoperation of a robot is often a difficult problem, especially when the robot is far away, with long delays in the communication link. Under this circumstance, instead of controlling the robot directly, it may be preferable to instead of control a virtual version of the robot. The user plans and specifies the robot's actions by manipulating the local virtual version, in real time. The results are directly displayed on the real world.

5. **Entertainment:** "At SIGGRAPH '95, several exhibitors showed "Virtual Sets" that merge real actors with virtual backgrounds, in real time and in 3-D. The actors stand in front of a large blue screen, with a computer-controlled motion camera videotaping the scene. Since the camera's location is tracked, and the actor's motions are scripted, it is possible to digitally composite the actor into a 3-D virtual background. The entertainment industry sees this as a way to reduce production costs.

6. **Military Aircraft:** For many years, military aircraft and helicopters have used Head-Up Displays (HUDs) and Helmet-Mounted Sights (HMS) to superimpose vector graphics upon the pilot's view of the real world. Besides providing basic navigation and flight information, these graphics are sometimes registered with targets in the environment, providing a way to aim the aircraft's weapons.

CHAPTER FOUR

IMPLEMENTATION OF MEDICAL VOLUME VISUALIZATION

4.1 Introduction

In this chapter, our study on medical image segmentation, surface extraction and visualization is presented. Applications of the theoretical background discussed in previous chapters are given as implementation of volumetric human head and brain data. Additionally, our interactive visualization efforts are briefly explained. The results of our work are demonstrated.

4.2 Data Properties

In this work, several of volumetric CT and NMR data, which are acquired from different equipments over the Internet. NMR volumes are 12 bit, 256x256 size 23, 55 and 74 slices depth, T1 and T2 modality grayscale raw data. CT data is 512x512 size, 50 slices depth 16 bit and grayscale. In preliminary work, slice thickness is not used, but in a real application, volume dimensions are very important.

In segmentation phase, our algorithm results are tested with the synthetically produced NMR data, which is created by BrainWeb Simulator Database of McGill University, Brain Imaging Center, Canada. These images are 8 bit long, 181x217x181 size, and T1 and T2 modality simulations. Also manually segmented data by an anatomy expert for testing our segmentation results is used. These images

are downloaded from the Internet. Table 4.1 shows an overview for various volumetric data used in this research.

Table 4.1 Overview of Volumetric Data Used in the Study.

Filename	Type	Modality	Resolution	Byte Swap	Slice Size	Slices
Mr01.rdf	Axial	MR T1	16 bit	Yes	256x256	74
Mr02.rdf	Axial	MR T1	16 bit	Yes	256x256	55
Mr04.rdf	Sagittal	MR T1	16 bit	Yes	256x256	23
Brainweb1.rdf	Axial	MR T1	8 bit	-	181x217	181
Brainweb2.rdf	Axial	MR T2	8 bit	-	181x217	60
Bwebgray.rdf	Axial	Gray*	8 bit	-	181x217	181
Bwebwhite.rdf	Axial	White*	8 bit	-	181x217	181
Bwebskull.rdf	Axial	Skull*	8 bit	-	181x217	181
BwebCSF.rdf	Axial	CSF*	8 bit	-	181x217	181
3dhead.rdf	Sagittal	MR T1	16 bit	No	256x256	109
Ctraw.rdf	Axial	CT	16 bit	Yes	512x512	25
AxialHead.rdf	Axial	MR T2	16 bit	Yes	256x256	44
788.rdf	Axial	MR T2	16 bit	Yes	256x256	60
Buchar.rdf	Axial	WGC**	8 bit	-	256cx256	59

* Segmented data for testing segmentation errors.
** White, Gray and CSF segmented volume.

4.3 The MEDVR System

In this study, a surgeon assisting and pre-operation training system, which is called as "MedVR", has been designed and realized as shown in fgure4.1. Basically, this study is the first step in surgery application of telepresence and virtual reality. The main idea is to overlay the all information related to patient's measured health data that may be needed during operation by the surgeon. This methodology is known as "Augmented Reality". For that purpose, we use a Head Mounted Display (HMD) to visualize real video and one dimensional signals, two dimensional images and real time reconstructed three-dimensional information. At the same time,

preliminary taken surgery notes by the surgeon can be used. During the operation, surgeon can review the operation and patient parameters.

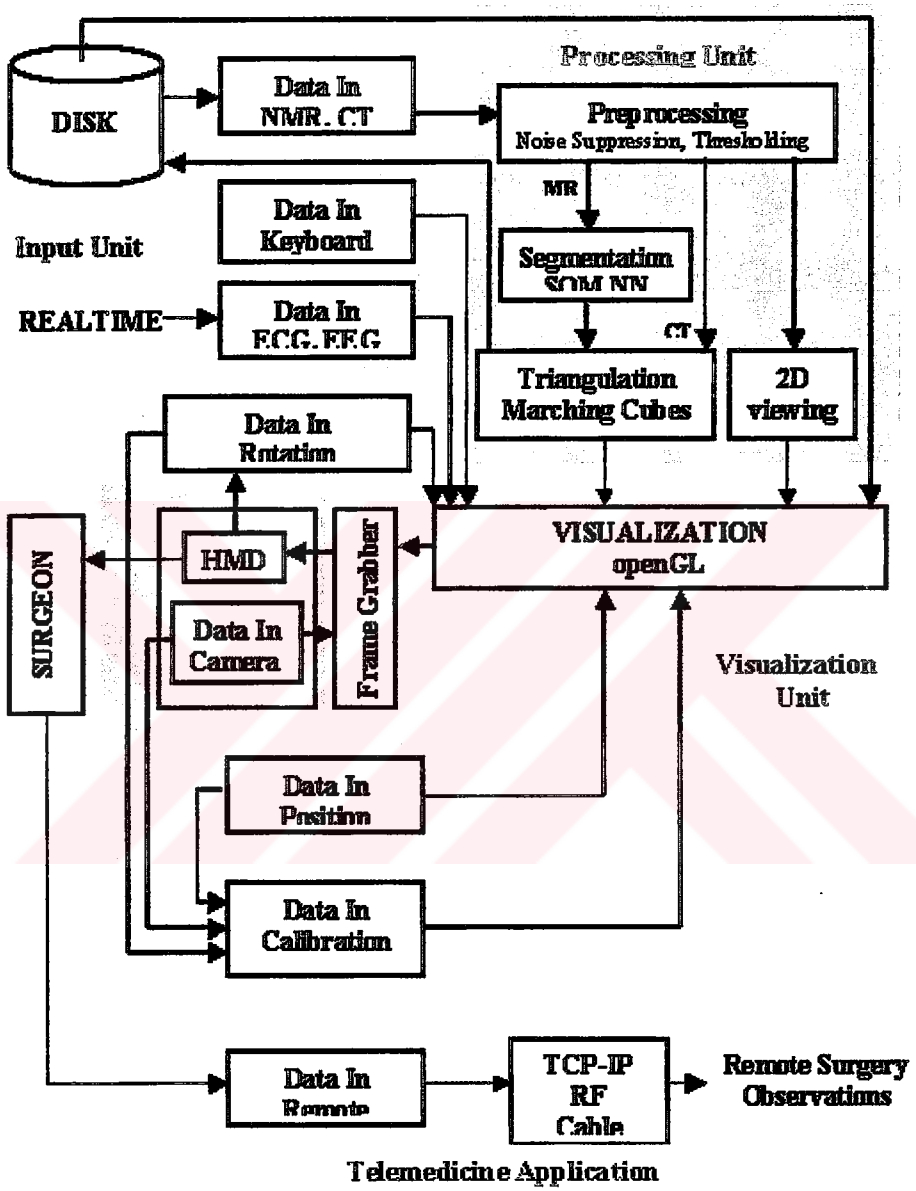


Figure 4.1 MedVR System Block Diagram

4.3.1 Outline of the system

This work consists of several main steps: Data Acquisition, Hardware, Software, and Visualization.

1. **Data acquisition**: 3D data manipulation has focused so far, but 2D slices of MR, CT images can also be acquired and displayed easily. 1D medical signals

can be acquired by using any data acquisition system. Some data acquisition systems such as PLC 818 can be used for input / output system. Several MR, and CT volume data is obtained over the Internet.

2. **Hardware**: A personnel computer equipped with a special frame grabber on a display driver and a camera are used. Also, a HMD is in the system.

3. **Software**: Software developed for our work is divided into three parts:

a) **Software Development Platform:** For that purpose, Borland C++ Builder Developing Engine for Windows environment with the support of OPENGL is used for the system MedVR.

b) **Mathematical Aspects and Computation** (Toriwaki & Mori, 1999)**:** Normalization: 12 or 16 bit values are converted to 8 bit unsigned values. Thresholding: 8bit binary images are thresholded by predefined values and filtered by a low pass filter. The threshold value is determined as 100 for 8 bit NMR data. Slice Segmentation: Thresholded medical volumes converted to slices and segmented by using self Organizing Maps (SOM network) by using visual and statistical information content parameters such as average, entropy, gray level and etc using Matlab version 5.2 with the SOM toolbox by Kohonen group. Polygonizing: Segmented data is converted to a form that consists of triangle polygonimal surfaces. This approach is more operate for polygon based visualization tools such as Opengl. Geometrical visualization: The data is converted to 3D space by coordinate transformations. Rotation and light parameters calculated. This process is known as Rendering.

c) **OpenGL**: A program is developed on a window by using an OpenGL viewport. So, OpenGL Graphics Engine and command is used to visualize all data.

4.4 Hardware

The hardware for our work consists of Head Mounted Display, Frame Grabber and a computer system, which are described below.

4.4.1 Virtual Reality Equipment: Head Mounted Display

1. **HMD**: Victormax's Cybermaxx 2.0 HMD.
2. **Tracker Interface**: 9600 Baud, 8 Data bits, 1 stop bit and no stop bit. Sequential serial tracker information as 2 bytes yaw, 2 bytes pitch and 2 bytes roll is transferred in the following form:

   FF FF (Yaw1)(Yaw2)(Pitch1)(Pitch2)(Roll1)(Roll2) (4.1)

3. **Visual Properties:** 360° yaw, ±45° pitch and roll. With the resolution up to 640x480 total, half resolution per TFT LCD. Our HMD does not require extra processing to display data. Odd lines of VGA card are shown in one LCD, Even lines are displayed by the other. So, if we do not try to get real stereoscopic vision, no extra effort is required, only connection to HMD to VGA card should be enough. Even and Odd frames difference is completed in HMD.

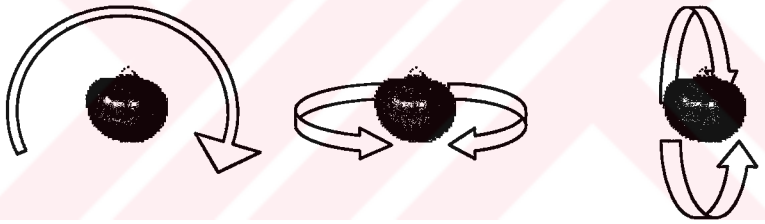


Figure 4.2. Roll, Yaw and Pitch

Roll is 0-360 degrees, yaw and pitch is -45 - +45 degrees as range of rotation as shown in figure 4.2. Our Program communicates with the tracker, open sequential data acquisition (we can poll the data whenever we wanted or tracker sends the data independently.); decode the rotation and finally passes to the visualization process.

From the data format given above yaw, roll and pitch are calculated as:

$$\begin{aligned} Yaw &= (Y1*256+Y2)*360/32768 \\ Pitch &= (P1*256+P2)*180/32768-90 \\ Roll &= (R1*256+R2)*180/32768-90 \end{aligned} \quad (4.2)$$

Figure 4. 3 shows the use of this interface in an example code.

4. **Sound:** Stereo headphones for 3d sound

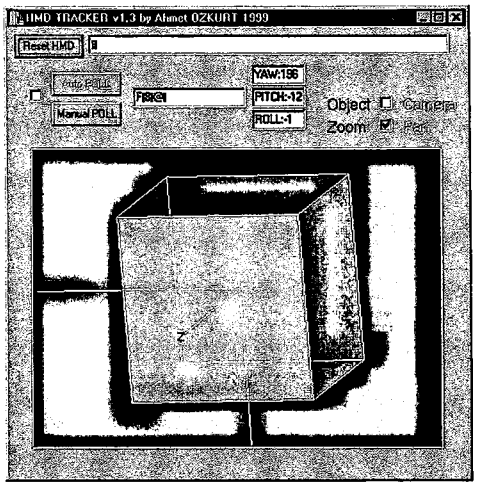


Figure 4.3. Sample Program Screen to Test HMD Tracker

4.4.2 FRAME GRABBER ISSUES

A new generation frame grabber with overlay graphics to real running video, MuTech IV-400 is selected and bought for this study. Our approach is visualization of our 3d graphics on real video in order to supply information to surgeon's HMD. This information in our case is three-dimensional MR reconstructed surface of the human head, especially brain tissues.

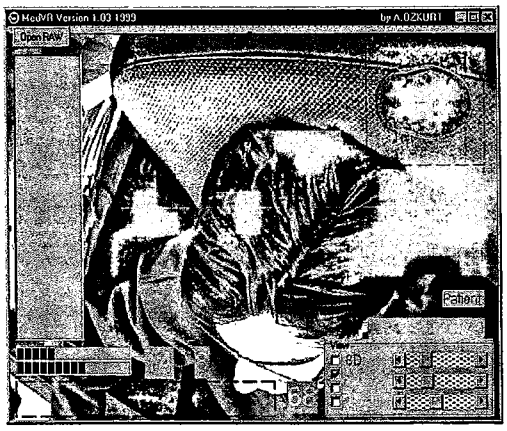


Figure 4.4 Video overlay View of Frame Grabber Mutech IV-400.

Reconstructed data is displayed depending on the target media; if the surgeon moves or rotate his head, camera replaces and the tracker senses the difference, sends the new position, then, visualization module program refreshes the 3D view position. In fact, our process is determining the 3D position and display it in a different

window. Transparency is done by selecting background color as key color, then our frame grabber automatically makes it invisible, so only the 3D volumetric view of tissue and the behind of it, real time running video appears. Figure 4.4 shows an example view obtained by key overlay method of our frame grabber.

4.4.3 Computer System

The computer system used in our prototype work consists of following components: This system is a primer configuration. It will be required to update in the future.

Table 4.2 Computer System Properties

Dual Pentium II 350Mhz CPU
Gigabyte 6XDS Mainboard
128 MB RAM
4 GB SCSI HD
10 GB IDE HD.
Mutech-IV400 Frame grabber + video Card 4M
Matrox Millenium II Video Card 8M
Tekram Video Capture Card
Philips Video Camera
VictorMaxx's CyberMaxX-II Head Mounted Display
WindowsNT 4.0 Workstation for Operating system

4.5 Preprocessing Stage

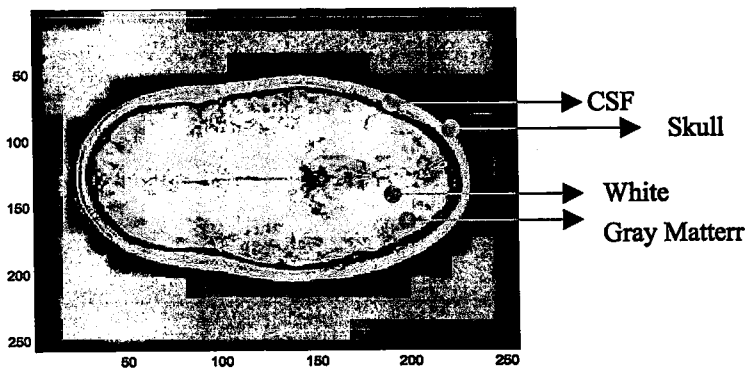


Figure 4.5. Sample T1 NMR Slice Data and preliminary tissues of interest

Primarily, most of the efforts are spent on NMR volumes in this study. CT images are easier to process. Only normalization and thresholding operation are applied to the CT image before the processing stage. NMR preprocessing stage consists of normalization, thresholding, and averaging and histogram equalization operations. The NMR slice is shown in Figure 4.5 with the main physical tissues of interest.

4.5.1 MR PreProcessing

Our NMR data is obtained from internet as 3D volumetric data or raw slices. Our slices are 256 x 256 and color depth is 12 or 16 bit gray scale. Our 3d data is arranged as sheets of slices combined together. At the same time, this system can be adapted to CT, where we have 512 x 512 CT raw data packet of 50 slices. Figure 4.5. Shows an NMR slice data of human brain.

Slices are converted to 256 gray scale to fasten segmentation and other processes. Histogram equalization is done as a preprocessing step for our data. In early phases, we use 64 x 64 and 128 x 128 data to see the result of algorithm implemented. Basic Tone Segmentation techniques are applied to our test data. This test program screen is shown in Figure 4.6.

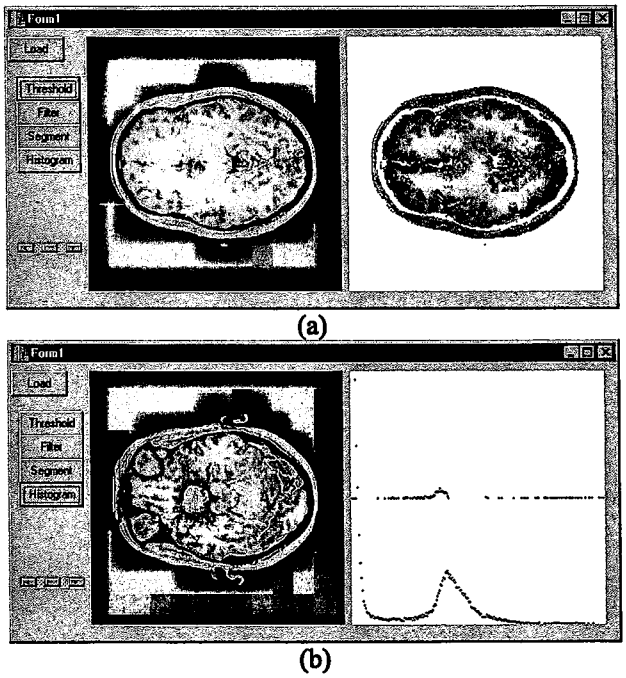


Figure 4.6. Basic Applications for Preprocessing on Sample Data Sets:
(a)Threshold, (b) Histogram

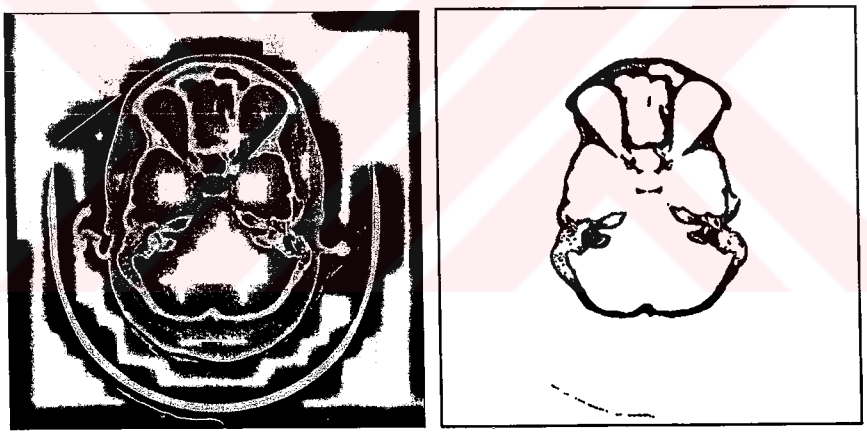
Normalization is realized by basic digitizing aspect given below:

$$x_{normalize} = \frac{x_{original}}{2^k} \cdot 2^8 \tag{4.3}$$

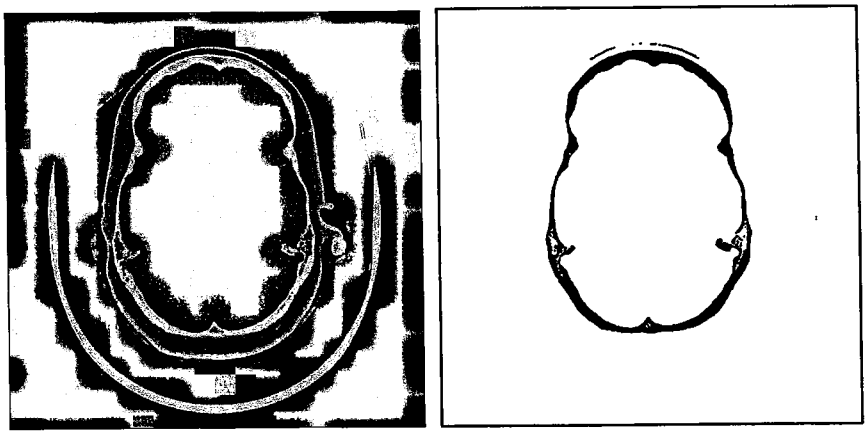
Where k is the input data resolution (12 or 16), $x_{normalize}$ is the output tone in 8 bit, $x_{original}$ is the input tone in k-bit resolution.

4.5.2 CT PreProcessing

Only normalizing and thresholding are applied to CT data, because of the CT content. In CT slice shown in Figure 4.7, only the skull is the dominant information relative to soft tissues such as brain and muscle. So, it is very easy to separate skull from a CT image.



(a) Slice7



(b) Slice 16

Figure 4.7: CT preprocessing results for extracting skull. Threshold Value=105.

4.6 NMR Volume Processing

Major Processing of the study is done on NMR volumes. Direct volume visualization and rendering techniques might have been applied to volumetric data, but this would require many resources and not contain valuable information. Instead, if biological tissues of brain such as White and Gray matter from NMR volume are segmented slice by slice, it may be useful for a good visualization. After segmentation, poligonizing is applied in order to extract surfaces of each segmented region.

4.6.1 Texture Analysis Method

There exist many approaches to the modeling of the textures. With stochastic patterns such as cloud images in which no natural contours occur, the co-occurrence matrix approach has been used with success. To illustrate what a co-occurrence matrix (Parker 1997) (Kohonen, 1997) is, we shall use the following very simple example. Consider Figure4.8.(a) that delineates a pixel group with shade values 0,1 and 2. These pixels lie on horizontal rows. Consider for simplicity only pairs of pixels with relative distance of $d = 1$ from left to right. There are two such pair with values (0,0) and (0,1), and on with (2,1) and (2,2), respectively; no other combinations are found. The values pairs (0,0), (0,1) etc identify the locations of the element matrix $M_d(d=1)$ in Figure 4.9(b), whereby the value of the left pixel in the pair is the row index of the matrix running left to the right, and the value of the right pixel is the column index running from top to bottom, respectively. The elements of M_d represent the co-occurrences of the said pairs of the values in the pixel groups.

The co-occurrence matrices M_d are in general constructed of the several relative displacements of d of the pixel pairs; in two-dimensional pixel groups they may be picked up in any direction.

The elements of M_d represent certain statistics of the shade values. For further compression of the information, statistical indicators of the elements $M_d(i,j)$ themselves can be used as texture features such as:

$$\text{"Energy"*} = \sum_i \sum_j M_d^2(i,j), \tag{4.4}$$

$$\text{"Momentum"*} = \sum_i \sum_j (i-j)^2 M_d(i,j) \tag{4.5}$$

$$\text{"Entropy"*} = \sum_i \sum_j M_d(i,j) \ln M_d(i,j) \tag{4.6}$$

Original image
(Consisting of horizontal lines)

0 0 1
0 0 1
2 2 1

Cooccurrence matrix $M_1(i,j)$

	j = 0	1	2
i = 0	2	2	0
1	0	0	0
2	0	1	1

Figure 4.8: Cooccurance Matrix Calculation method (*Kohonen, 1997).

These "statistical" (Parker, 1997) (Valova & Suganami & Kosugi, 1998) and Spatial (Brink 1996) properties of brain tissues are used as fundamental features of texture for segmentation.

4.6.2 Segmentation

Segmentation algorithm used in this work is based on Self Organizing Map (SOM) neural network explained in section 3.4.1. This unsupervised learning neural network is proposed by Kohonen. The outline of our approach is given as:

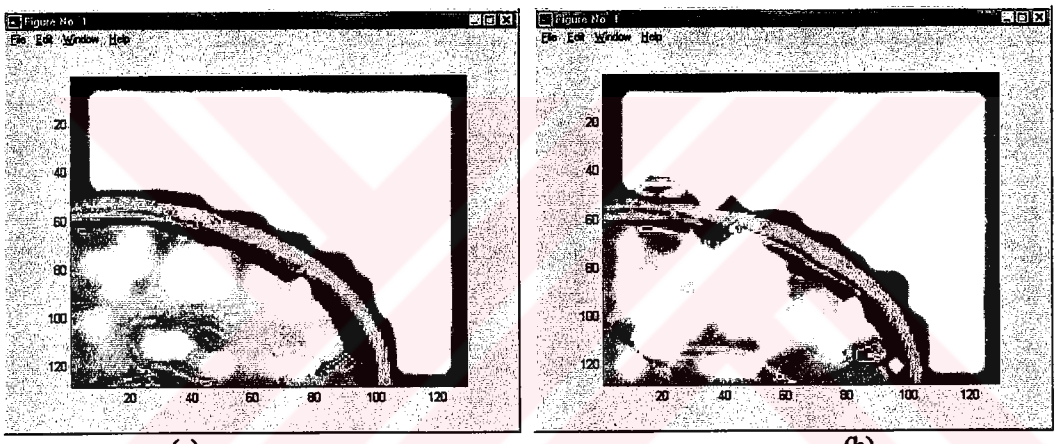
1. Different map sizes, 5x5, 7x7 and 9x9 are used.
2. The features for classification brain images are:

Table 4.3 Features for Neural Segmentation

Selected Pixel Tone
Selected Pixel Neighbor tones (total 24)
Selected Pixel Area Average Tone
Selected Pixel Area Similarity Coefficient (Gradient)
Selected Pixel Area Energy Values for 1st and 2nd Neighboring
Selected Pixel Area Momentum Values for 1st and 2nd Neighboring
Selected Pixel Area Entropy Values for 1st and 2nd Neighboring

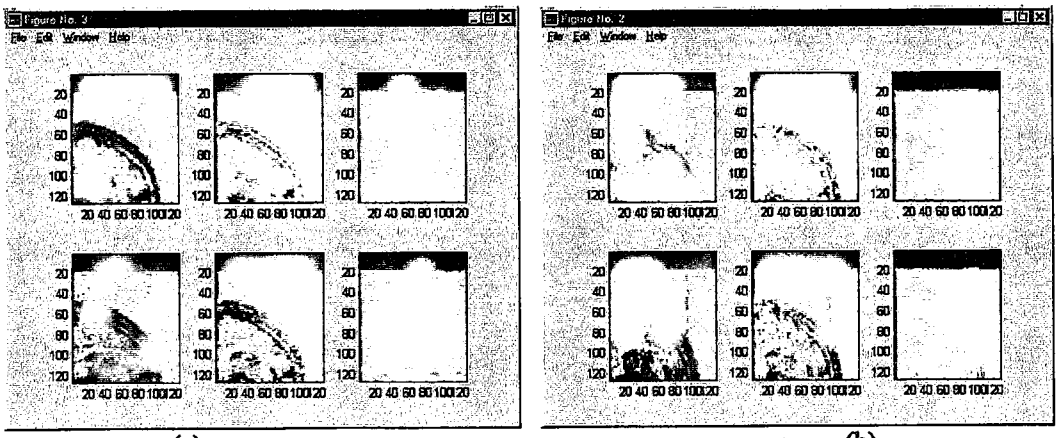
3. A SOM neural network is taught with a learning slice, by different epoch numbers to find winner neurons for each feature set per each pixel.
4. It is then tested to find which neuron is matched to which biological tissue part on the slice.
5. Related neurons with white or gray tissues are defined.

Figure 4.8 shows some example slices for segmentation. We use the same type of MRI images for learning and testing phases. Important Features, Entropy, Momentum and Energy of an example image are given in the Figure 4.9 as images.



(a) (b)

Figure 4.9: Data for segmentation,(a) Learning Data, (b) Test data



(a) (b)

Figure 4.10: Example features, Energy, Momentum, Entropy. (a) x axis, (b) y axis

It is seen from the feature image shown in Figure 4.9 that, energy gives high tone intensity, momentum gives general tone differences and finally entropy gives

information about the content for texture. Highly changing areas responses high entropic values.

Figure 4.11 and Figure 4.12 show segmentation examples. In this figure, example raw image is segmented as white and gray matter and displayed.

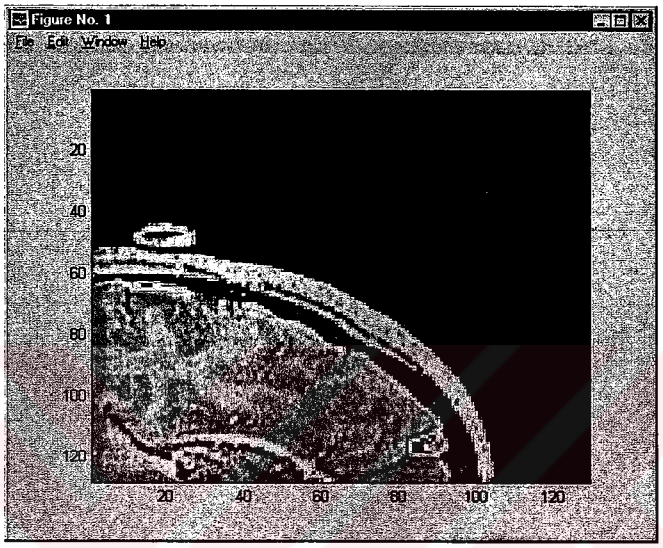


Figure 4.11: Test Data for Segmentation. Right top of T1 MRI slice 66 of 77.

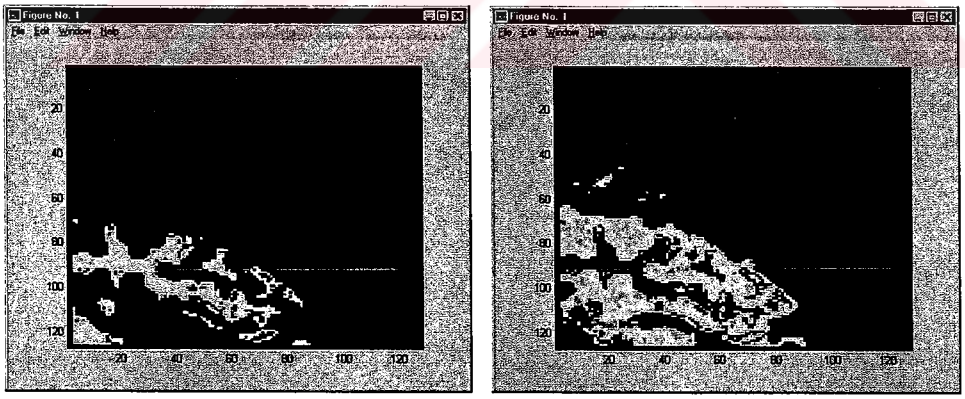


Figure 4.12: Segmented (a) White Matter (b) Gray Matter from the slice shown in figure 4.10.

Segmentation process in a neural network includes two phases: Learning and Testing. Also each phase is divided into two subphases: Feature extraction and operation. Table 4.43 gives required process times for several example slices.

4.6.3 Segmentation Error

For highly accurate segmentation, we need to determinate segmentation errors. As expressed before, there is a segmentation error even if manual segmentation is applied by a medical expert. Determination of errors is realized in this work by finding difference points between pre-segmented data taken from BrainWeb simulator and our segmentation results. The original slice is taught to our SOM network and our result is compared with its originally segmented image. This error determination results are given in the Table 4.4. As seen from that table, our Error rate is under 10%.

Table 4.4 Segmentation Error Rate

Image	Modality	Map Size	Slice Size	Pixel Error	Error Rate %
BrainWeb1	T1	7x7	180x216	1854	4.76
BrainWeb1*	T1	9x9	180x216	2341	6.02

*Given below as Figures 4.12.and 4.13.

Brainweb T1 slice100 slice is shown in Figure 4.13 with the segmented white matter. This segmentation is realized by the neuron hits of 1, 2, 3, 4, 5, 6, 10, 11, 12, 14, 15, and 16. Figure 4.14 shows the comparison our result.

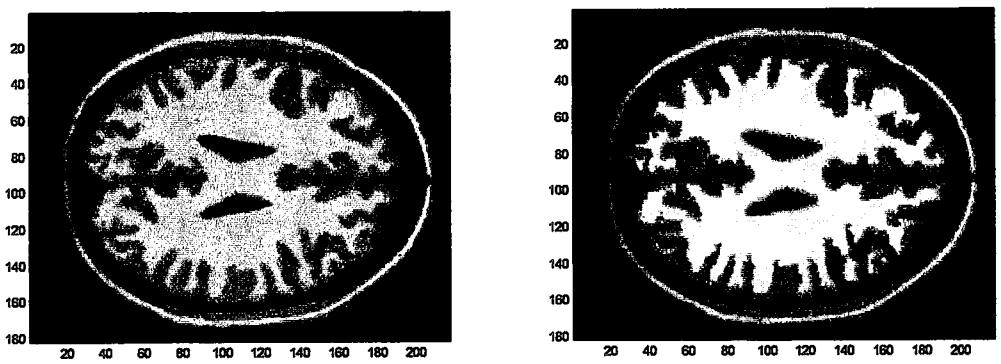


Figure 4.13 Segmented White Matter for Error Determination

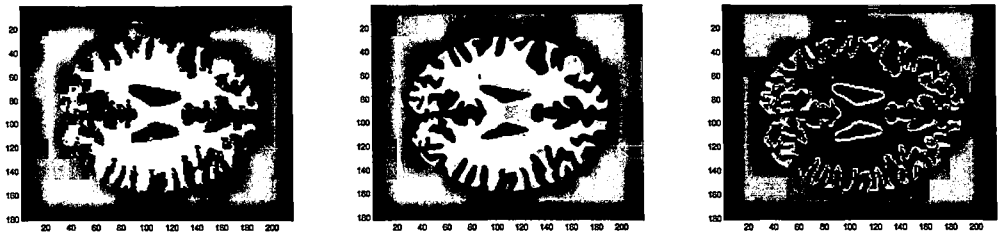


Figure 4.14 comparison of our work with the simulated White Matter

Table 4.5 Segmentation Time (sec)

Image	Learning Phase		Testing Phase	
	Feature	Learning	Feature	Learning
BrainWeb T2 181x217 8 bit 7x7	551.43	1979.7	731.82	194.60
1slice70 T1 256x256 16 bit 7x7	870.14	9597.7	884.93	360.15
1slice70 T1 128x128 16 bit 7x7	206.73	773.01	209.78	88.76
1slice 70 T1 128x128 16 bit 9x9	212.75	1296.4	209.93	118.92

4.6.4 Volume Reconstruction

Segmented slice data is combined together as different files to be retrieved whenever they are required. In this work, white, gray volumes from MRI data and skull from CT slices are reconstructed as volumetric surfaces. Reconstruction may be applied as direct rendering and polygonizing (Shahidi & Tombpropoulos & Grzesczczuk, 1998), but direct rendering is not valuable too much and used for only creating two-dimensional volumetric renderings. Polygonizing is used for determining volume surfaces and this is more meaningful anatomically.

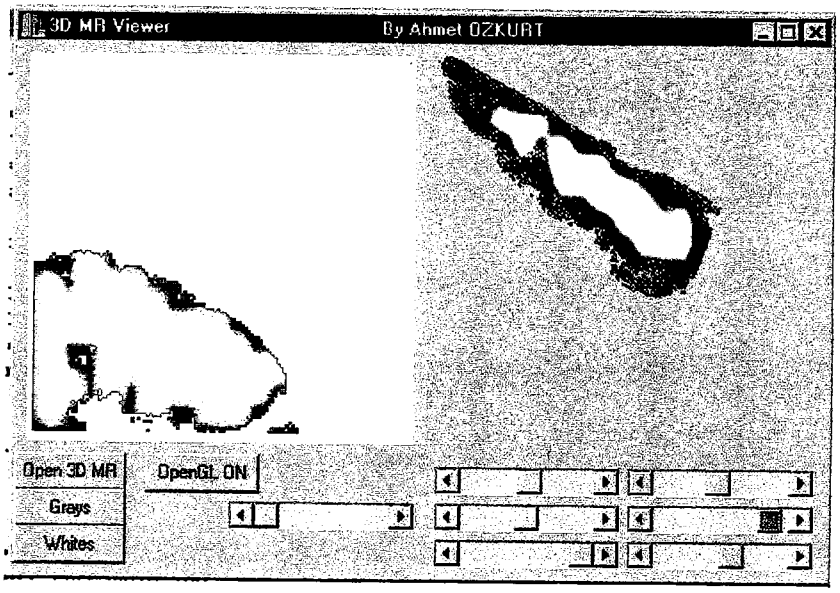


Figure 4.15: Direct Rendering of Segmented Right Top of White Matter

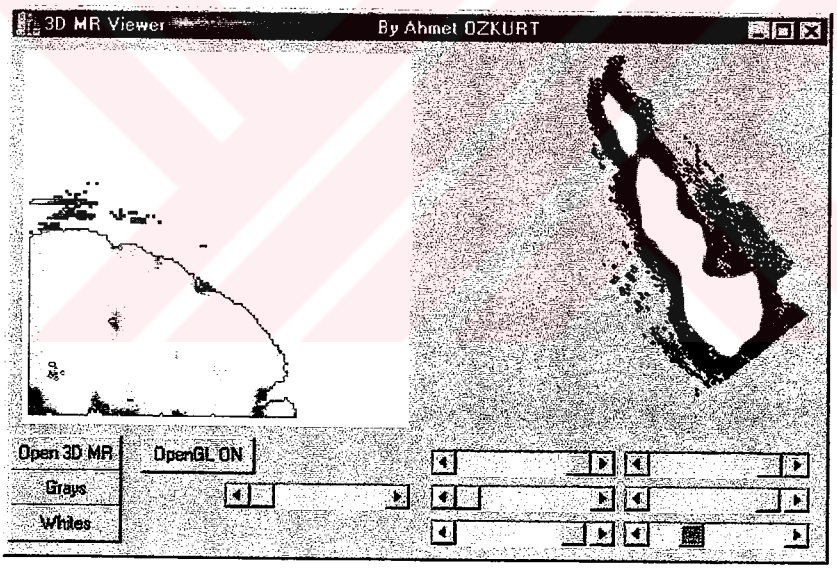


Figure 4.16: Direct Rendering of Segmented Right Top of Gray Matter

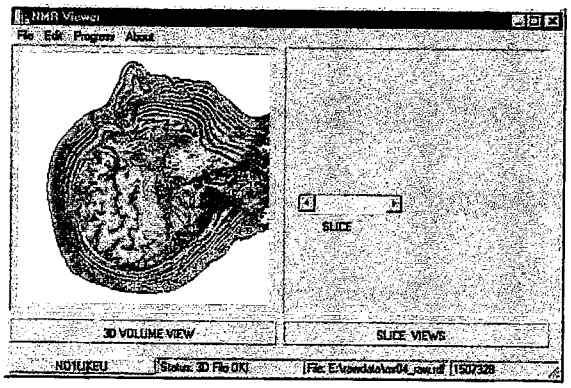


Figure 4.17 Direct Rendering

4.6.5 Edge Detection

Figure 4.18. Edge Detection Result for segmented Gray quarter slice.

Edge Detection algorithm is based on gradient detection of right, down, right-down directions.

4.6.6 Triangulation

Triangulation is a process to extract a surface model of any scalar volumetric data. The sampled discrete values of spatial voxels are converted to polygonimal patches. Combined patches coat a closed surface. This process is known as reconstruction. Marching Cubes algorithm (Lorensen & Cline, 1987), given in section3.5.4, is used for surface reconstruction in this study.

Implementation of Marching Cubes Algorithm is given as follows: Marching Cubes is an algorithm for rendering isosurfaces in volumetric data. The basic notion is that we can define a cube by the pixel values at the eight corners of the cube as shown in Figure 4.19 below. If one or more pixels of a cube have values less than the user-specified isovalue, and one or more have values greater than this value, we know the cube must contribute some components of the isosurface. In our case, segmented regions have boolean isovalues. By determining which edges of the cube are intersected by the isosurface, we can create triangular patches that divide the cube between regions within the isosurface and regions outside. By connecting the patches from all cubes on the isosurface boundary, we get a surface representation.

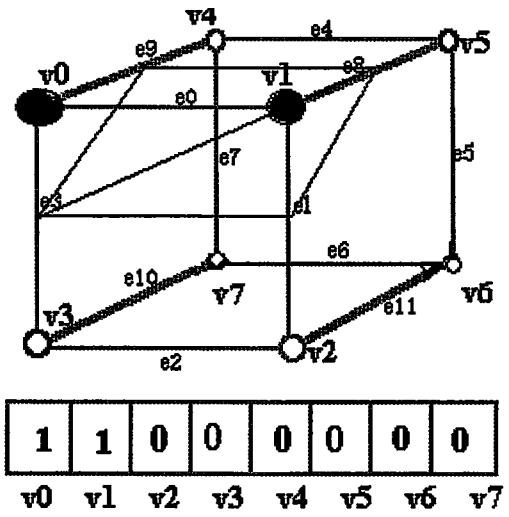


Figure 4.19. A cube settlement for marching cubes algorithm (li, Agathoklis, 1993)

There are two major components of this algorithm. The first is deciding how to define the section or sections of surface that chop up an individual cube. If we classify each corner as either being below or above the isovalue, there are 256 possible configurations of corner classifications. Two of these are trivial; whether all the points are inside or outside the cube does not contribute to the isosurface. For all other configurations we need to determine where, along each cube edge, the isosurface crosses, and use these edge intersection points to create one or more triangular patches for the isosurface.

If we assume symmetries, there are really only 16 unique configurations in the remaining 256 possibilities of the vertex in the cube. When there is only one corner less than the isovalue, this forms a single triangle that intersects the edges that meet at this corner, with the patch normal facing away from the corner. Obviously there are 8 related configurations of this sort. By reversing the normal we get 8 configurations that have 7 corners less than the isovalue. These are not really unique, however. For configurations with 2 corners less than the isovalue, there are 3 unique configurations (e.g. for configuration 12), depending on whether the corners belong to the same edge, belong to the same face of the cube, or are diagonally positioned relative to each other. For configurations with 3 corners less than the isovalue, there are again 3 unique configurations (e.g. for configuration 14), depending on whether there are 0, 1, or 2 shared edges (2 shared edges gives you an 'L' shape). There are 7

unique configurations when you have 4 corners less than the isovalue, depending on whether there are 0, 2, 3 (3 variants on this one), or 4 shared edges (e.g. for configuration 30 - again you may need to tweak the colors to see the triangle for the isolated (far) inside sphere/pixel).

For determining the edges for selected vertexes, an edge table is used. Table 4.6 shows the example edge table used in the study. Each element in the edge table is an index for one of the 256 different vertex combinations.

Table 4.6 Example Edge Table in hexadecimal form.

```
int edgeTable[256]={
0x000, 0x109, 0x203, 0x30a, 0x406, 0x50f, 0x605, 0x70c,
0x80c, 0x905, 0xa0f, 0xb06, 0xc0a, 0xd03, 0xe09, 0xf00,
0x190, 0x99 , 0x393, 0x29a, 0x596, 0x49f, 0x795, 0x69c,
0x99c, 0x895, 0xb9f, 0xa96, 0xd9a, 0xc93, 0xf99, 0xe90,
0x230, 0x339, 0x33 , 0x13a, 0x636, 0x73f, 0x435, 0x53c,
.......
}
```

After determination of vertex combination number from edge table, the next step is looking up the triangles from triangle table showing which three edges are constructing a triangle. At the same time, triangle table pointed how many triangles will be created. A triangle table is given in the Table 4.7.

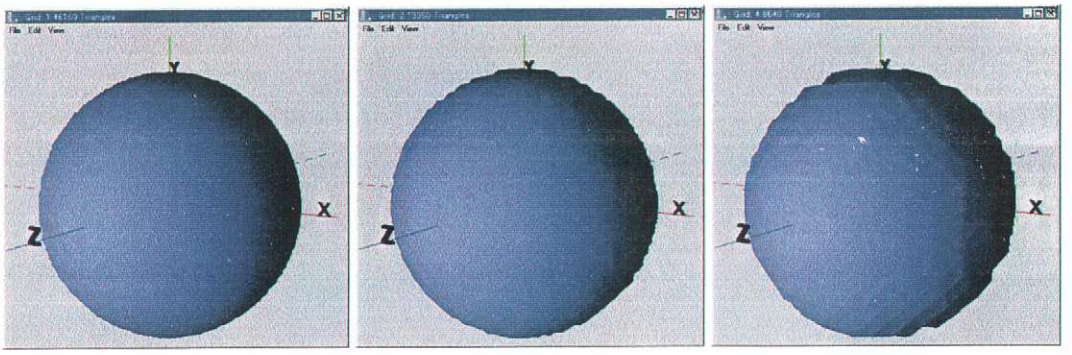
Table 4.7 Triangle Table showing triangles will be created.

```
Int triTable[256][16] ={
{-1,-1,-1,-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{ 0, 8, 3,-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{ 0, 1, 9,-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{ 1, 8, 3, 9, 8, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{ 1, 2, 10,-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{ 0, 8, 3, 1, 2, 10, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{ 9, 2, 10, 0, 2, 9, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
...}
```

The selected part of the table above shows that:

1. 2 triangles will be created because 6 non -1 elements exist,
2. 1, 8 and 3 edges shows first triangle and 9, 8, and 1 edges points second triangle.

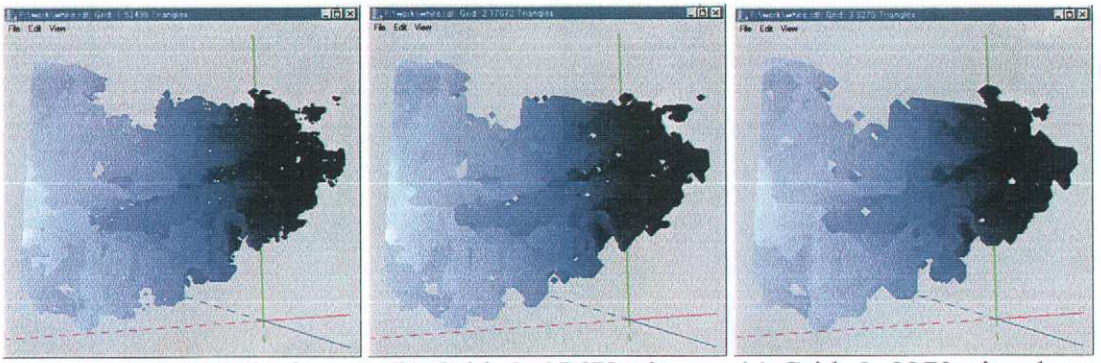
Each of the non-trivial configurations results in between 1 and 5 triangles being added to the isosurface. The actual vertices themselves can be computed by interpolation along edges, or, their default location to the middle of the edge. The interpolated locations will obviously give you better shading calculations and smoother surfaces.



(a) Grid=1, 46106 triangles (b) Grid=2, 19360 triangles (c) Grid=4, 8640 triangles

Figure 4.20 Polygonizing Example of Sphere (80x80x80)

Now that we can create surface patches for a single voxel, we can apply this process to the entire volume. We can process the volume in slabs, where each slab is comprised of 2 slices of pixels. We can either treat each cube independently, or we can propagate edge intersections between cubes which share the edges. This sharing can also be done between adjacent slabs, which increases storage and complexity a bit, but saves computation time. The sharing of edge/vertex information also results in a more compact model, which is more amenable to interpolated shading. Example triangulation is given in Figure 4.20 for a synthetic sphere for the grid sizes 1,2, and 4 with the triangle numbers 46106, 19360 and 8640 respectively. Figure 4.21 shows the white matter triangulation results for grid sizes 1,2 and 4.



(a) Grid=1, 53498 triang. (b) Grid=2, 17672 triang. (c) Grid=3, 9270 triangles

Figure 4.21 White Matter Triangulation Example

Numbers of triangles produced by our algorithm is given in Table 4.8. In fact the grid size 1 is the most applicable and give more anatomical detail. But, when the input volume data size is increased, the number of triangle becomes very high. While acceptable detail level is saved, higher grid sizes can be applied. The detail level is depending on the application. In a brain and eye operation, as much as high anatomical detail should be required, but in a knee replacement, high grid size and lower detail level can be used.

Table 4.8 Numbers of triangles produced for different volume data

Grids	Test Sphere 80x80x80	Gray Matter 128x128x23	White Matter 128x128x23	BrainWeb Skull 181x217x181	BrainWeb White 181x217x181
1	46160	90462	53498	727310	905700
2	19360	30984	17672	290844*	356420*
3	11956	16270	9270	169844	207624
4	8640	10382	5850	113712	139226
5	6616	7278	4074	81934	100546
6	5620	5240	3026	63052	76074
7	4712	4082	2278	49322	59232

*Given in Figure 4.28 as OpenGL Three Dimensional Rendering

4.7 Video Overlay Method

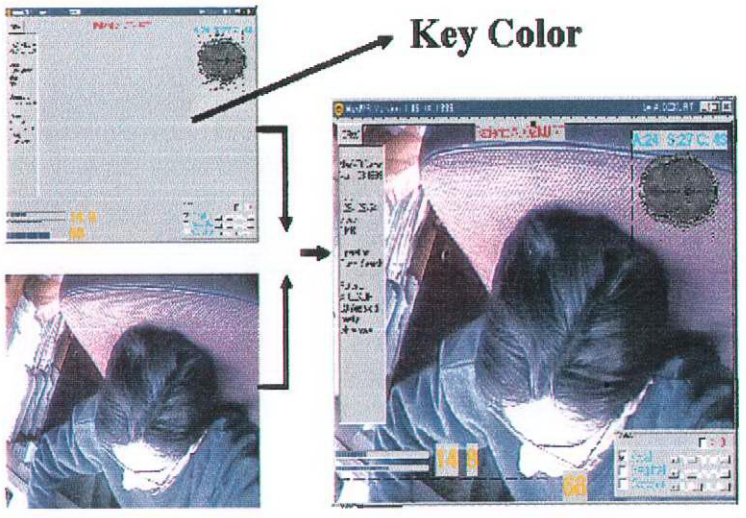


Figure 4.22 Video Overlay Method

Our approach requires that, computer aided reconstructions of medical volumes, two dimensional slice views and one dimensional vital signs of the patient like ECG have to be overlaid on live video acquired by a camera system mounted on the head of the surgeon. This process is made in our work by hardware. This hardware consists of a smart frame grabber Mutech-IV400 with key overlay function. Figure.4.22 shows this operation. Hardware implementation of this function eliminates the processing power drawbacks when video is processed inside the software. All processing power is delivered to visualization actions.

4.8 Example Results

4.8.1 Segmentation

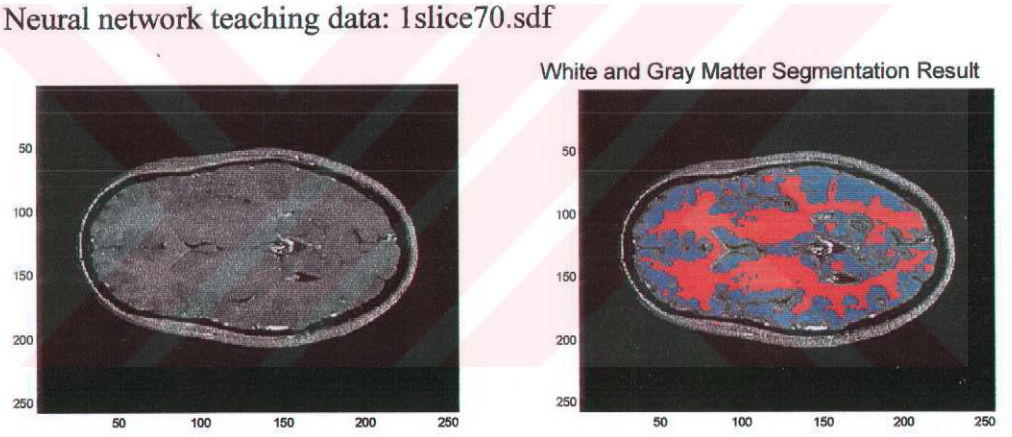


Figure 4.23 Slice70 Axial T1 Slice (256x 256, 16 bit, 70th of 74), white and gray

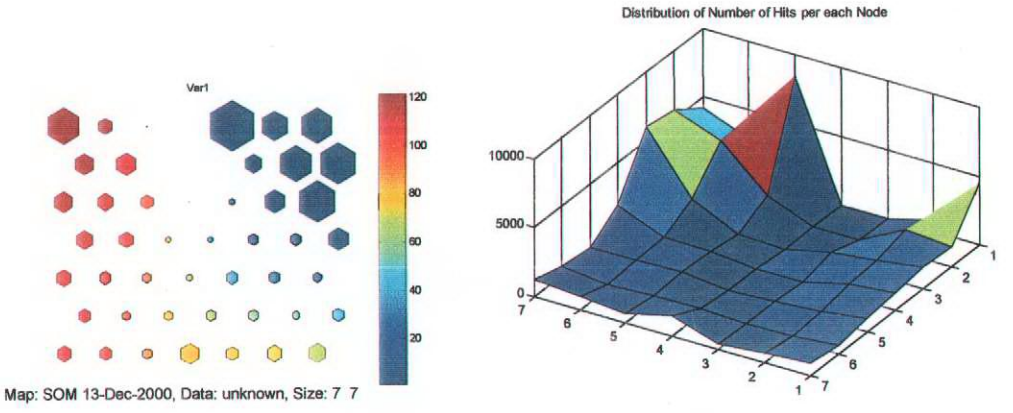
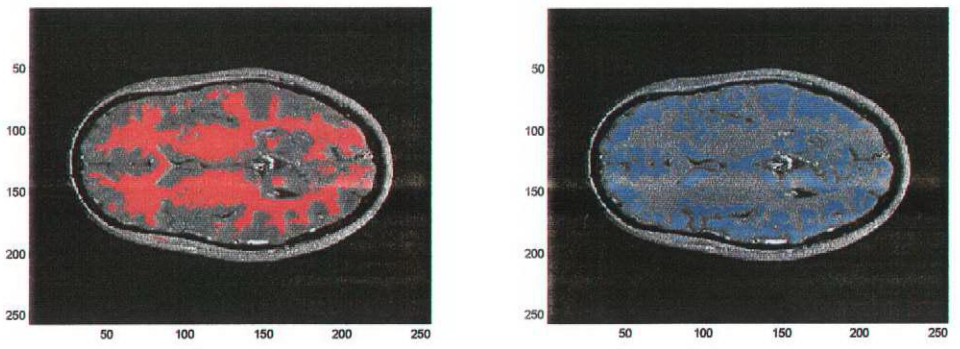


Figure 4.24. Number of hits per each Neuron

Whites winners 1,2,3 and Grays winners 4,8,9,10,17, 18 are given in Figure 4.25. Figure 4.26 also shows gray and whites together on related slice.



(a) White Matter (b) Gray Matter

Figure 4.25 (a) White (segmented from the neurons 1,2 and 3) Matter, (b) Gray (segmented from the neurons 4,8,9,10,17 and 18) Matter

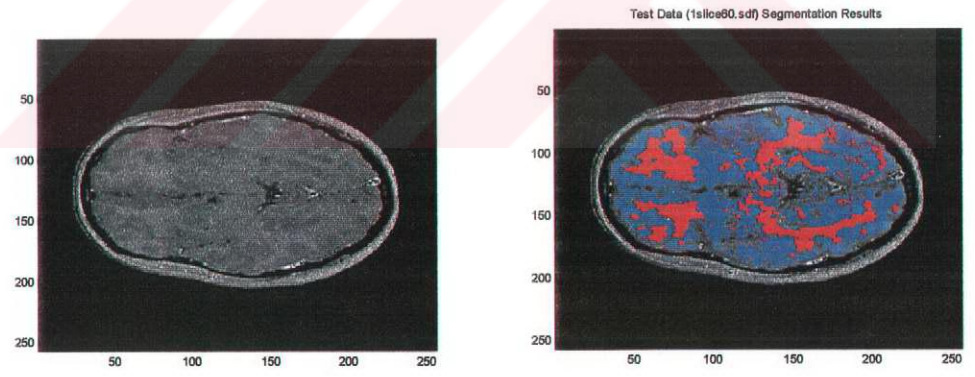


Figure 4.26 (a) Neural network TEST data 1slice 60.sdf and (b) White and Gray

4.8.2 Triangulation

Figure 4.27 shows the Marching Cubes Algorithm results for different unit cube combinations of 256 cases.

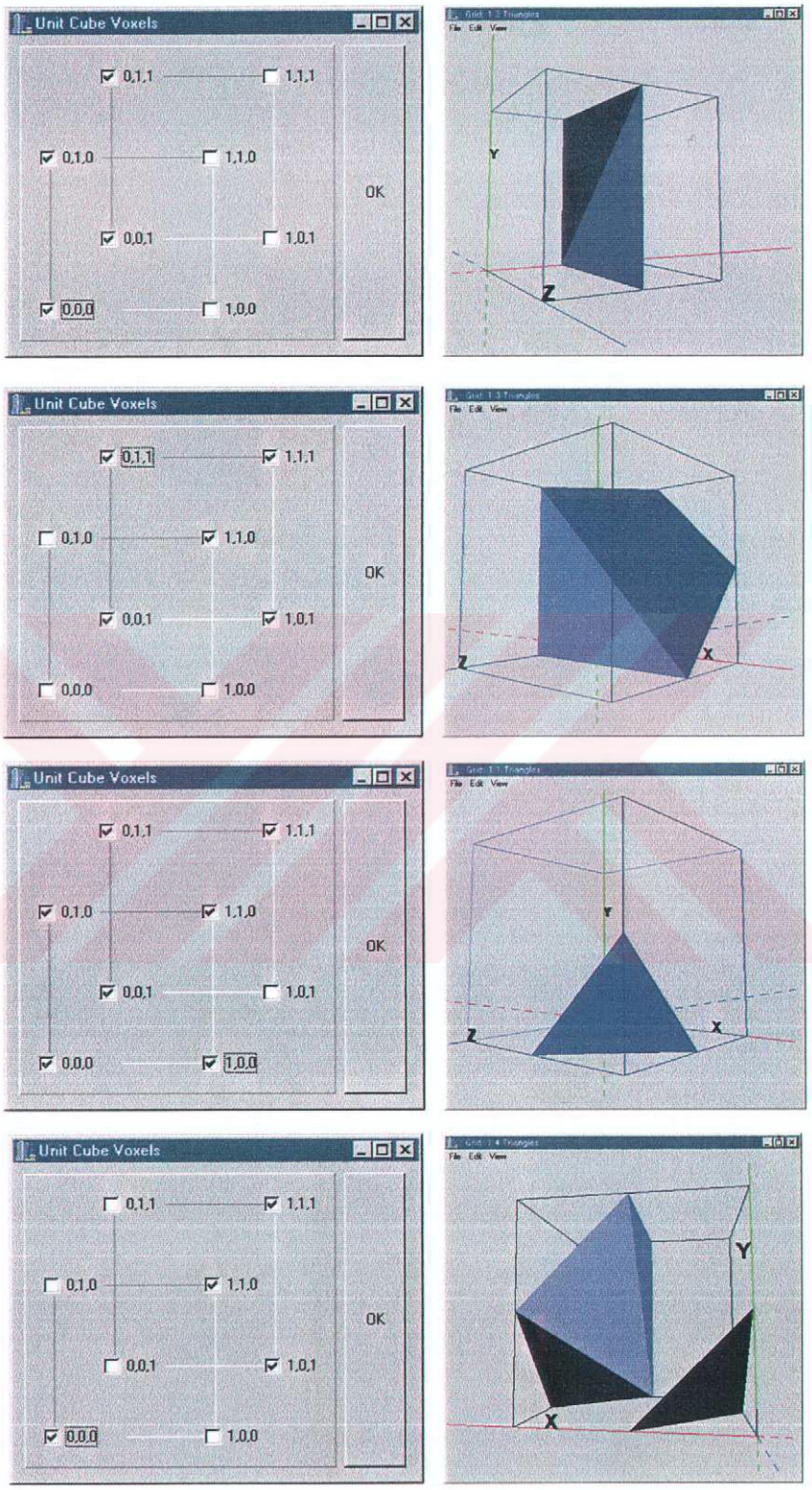


Figure 4.27 Example Unit Cube combinations and Related Triangles

Example Triangulation of quarter human skull data is given in Figure 4.28 below.

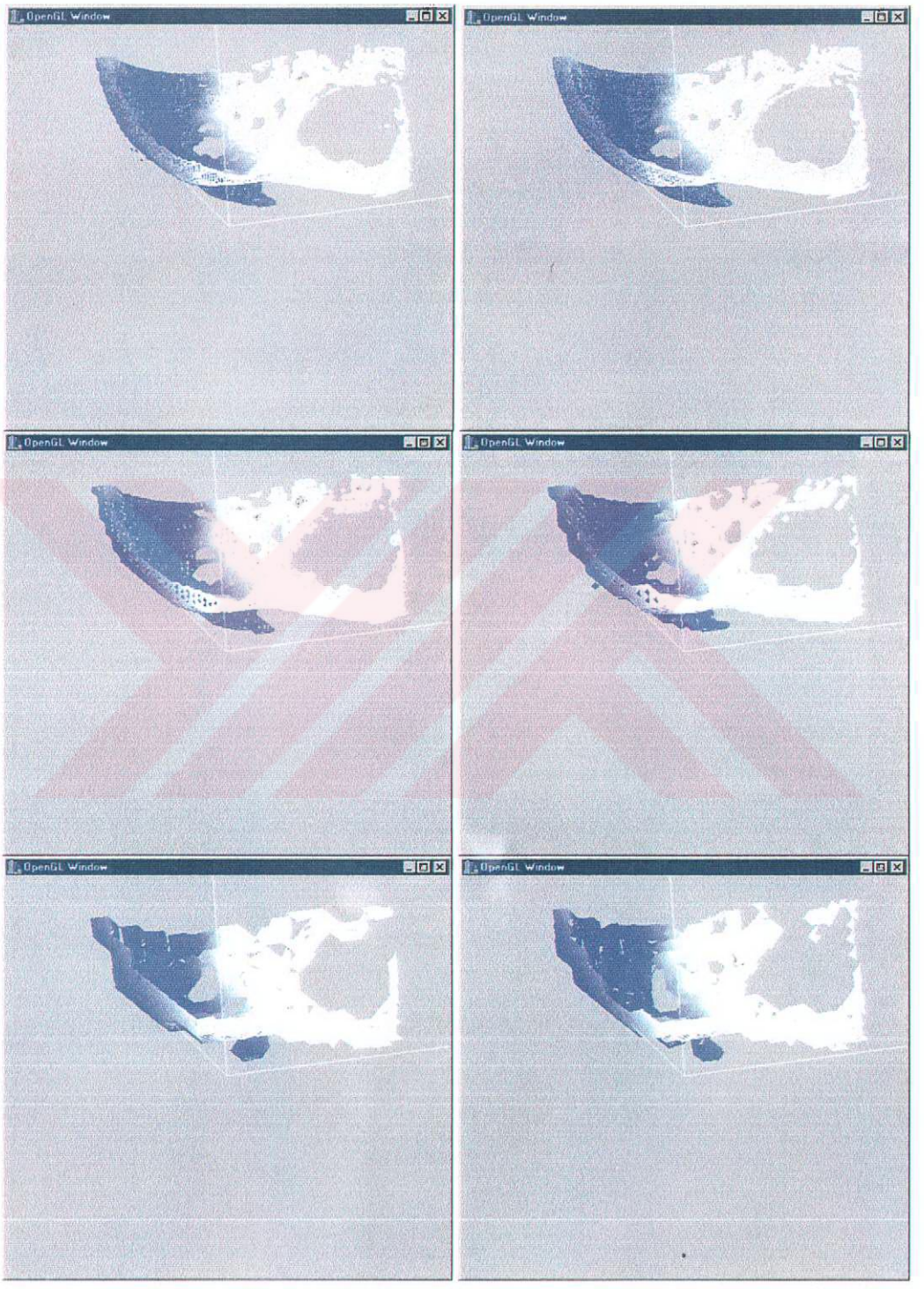


Figure 4.28 Triangulation of Quarter of CT Skull for grids 1,2,4,6,8,12.

4.8.3 Visualization

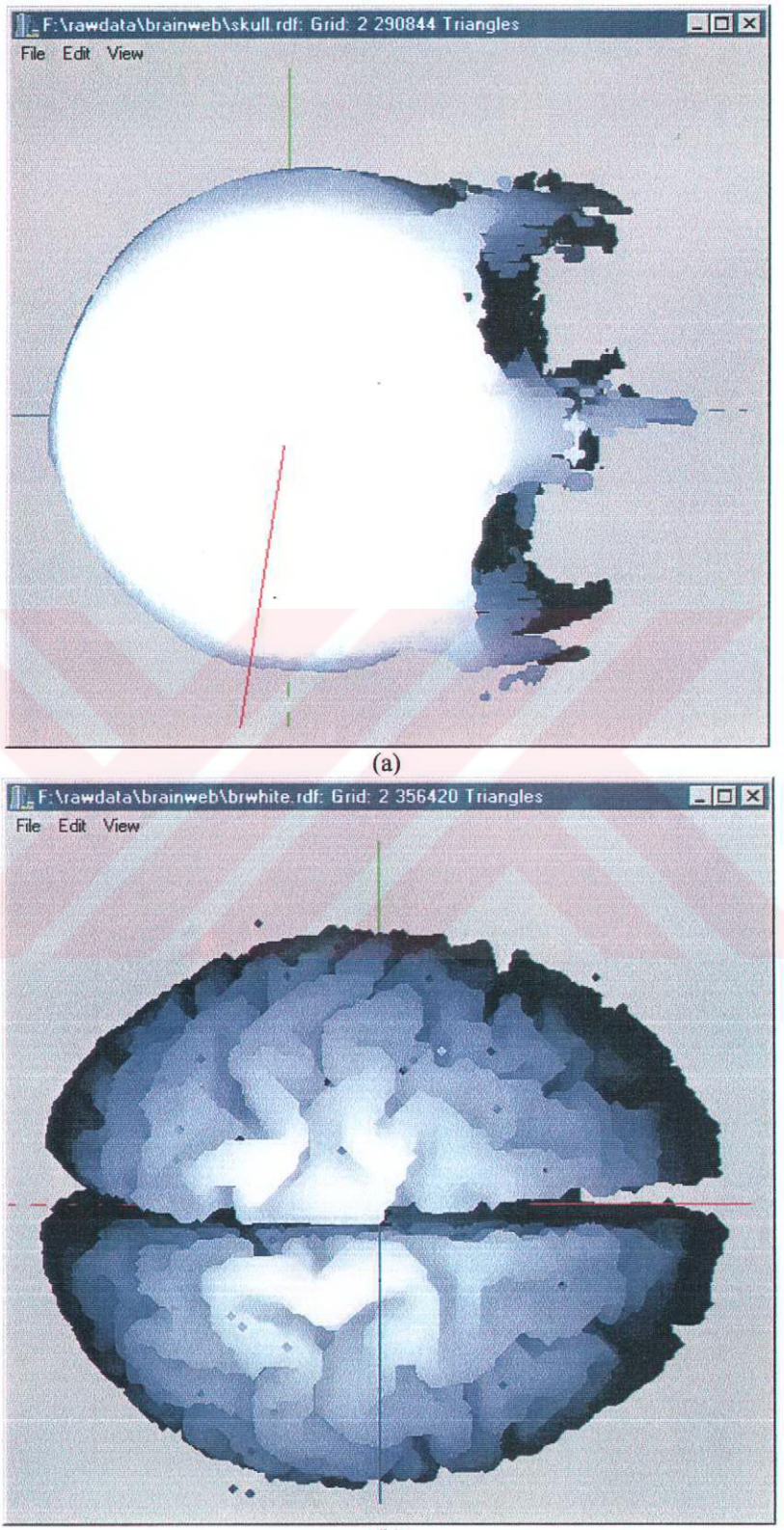
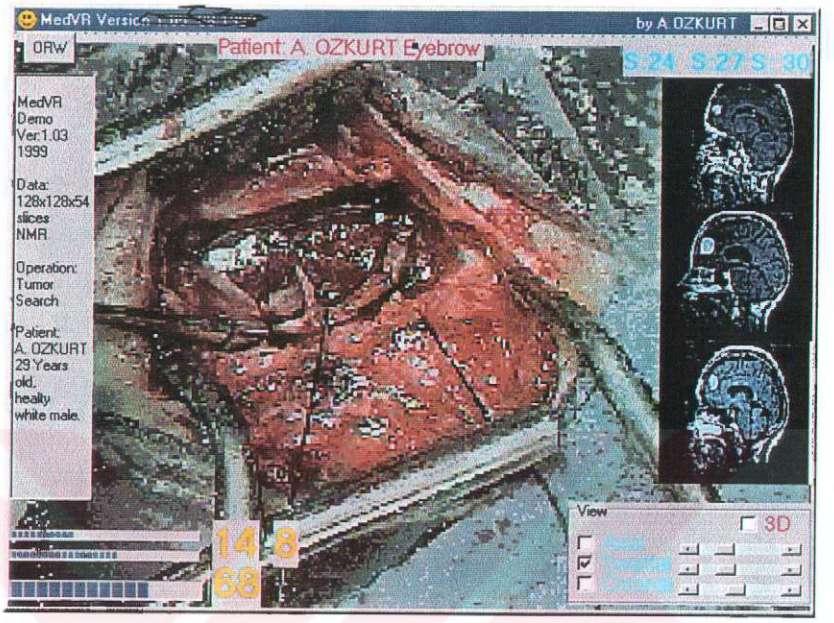
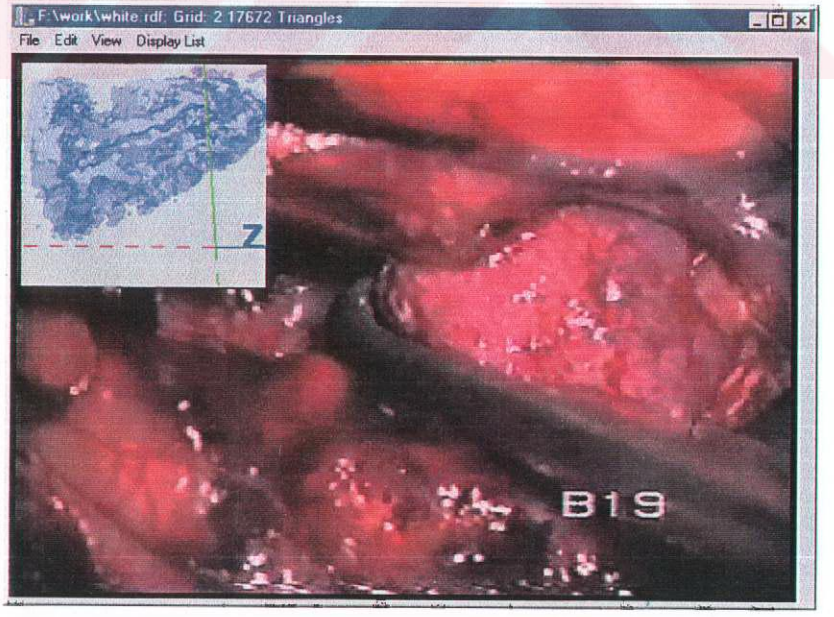


Figure 4.29 OpenGL Renderings of a) BrainWeb Skull and b) White Matter Data.

Example screen of the system, when it is in use, is shown in Figure 4.30 (a) and (b). These are not real application snapshot and it is used to give an idea to the reader. Background image is taken from a brain tumor surgery around the eyebrow of the patient.



(a)



(b)

Figure 4.30 Expecting Snapshot of MEDVR system when the system in use. (a) One and two- dimensional Information Support, (b) Three-dimensional overlay.

4.9 Common Restrictions and Limitations

In our work, we have several problems and limitations, which are described below.

1. In triangulation, the ambiguity problem, which originates from arching Cubes algorithm, exists.
2. The processing phase takes too long for segmentation.
3. OpenGL Library cannot be implemented as much as required for high-speed interactive manipulation of volumetric data.
4. Fully controllable MRI and CT equipments are required. For high success rate, we need the same type of images.
5. An operator is required to accomplish commands given by the surgeon.

CHAPTER FIVE
CONCLUSION

In this thesis, a surgeon assistance system named MEDVR have been implemented and possibilities and requirements of this system as a useful system for a surgery are investigated. The method for surgery assistance is to reconstruct three-dimensional surface models of the medical tissues and display these renderings on an HMD, which is mounted on the surgeon's head. This possibility introduces the maximum benefits of a surgery assistance system by using real-time interactive information support.

Method in this study is to develop a surgery planning, education and surgery assistance system. The approach is Augment Reality (AR), donated by a HMD (Head Mounted display) and a camera, and the combination of real world and real data. The system can be used for any surgery in any area of human body. The other types of patient monitoring and surgeon assistance systems have their own advantages and disadvantages that are discussed in Section 1.3. The medical staff will determine the usability of this system after a proper usage period.

Several comments given on this system by the surgeons indicate that this system may be useful in surgical actions. Some of the operators believe that this system is a new patient monitor and a surgery assistance tool.

This system will be our first step to telemedicine. In the future, this system may be used to share surgical data, including patient vital signs and surgeon's treatment; therefore it could be possible to realize the surgeries with the help of consultance of remote experts.

5.1 Advantages of Our System

This system brings some advantages for a surgeon, directly for patient's health. Those advantages given below help the surgeon pre, intra and after the operations. At the same time, some parts and data processing algorithms are important trials of applications. When this system is used, we expect the following benefits:

1. Our system can be used as a surgery planning and education tool before the operation. Three-dimensional reconstructions with interactive manipulation provide the optimal investigament for medical structures. The surgeon can prepare to the operation with more powerful and realistic material than conventional procedures.

2. Preoperative planning of surgeons using such systems may prevent possible surgery replications; therefore reduce the cost of surgical repeats.

3. For an operator, preparation phase to a surgery with volumetric fly-through takes shorter amount of time relative to conventional methods. This provides the surgeon with time-efficient usage.

4. The system is supposed to increase the success rate of the surgery; before and during the operation, with a well supported surgeon, it is possible to perform a surgery in a shorter time, successfully.

5. Since the success rate of the surgery is depending on the surgeon's performance, saving the concentration of the surgeon in whole operation may protect the surgeon and the patient from the unexpected failures.

6. The system can be used as a good local or remote patient monitor.

7. In the future, the system can be used for a teleconsultance system, before, during, and after the operation. This improves the safety of the patient.

8. With the help of a highly developed medical robotics system, a surgery can be performed remotely (Taylor et al, 1995), in such cases where the surgeon is

far away from the surgery site. Examples of such cases are encountered in war and space applications.

9. Our approach can be used in other application areas such as robot control and mechanical modeling. Since this work is an example of the Augmented Reality or Enhanced Reality approach, it may be a pioneer for other application areas.

10. Our approach to segmentation may be an alternative to unsupervised texture classification with a neural approach using spatial and statistical texture parameters such as entropy and energy.

11. Our polygonizing scheme can be used as a visualization tool for any other physical variables, e.g., temperature, in a three dimensional interactive manner. For a scalar valued data, fast and efficient visualization studies may be derived.

5.2 Disadvantages and Limitations

This study, since it is only a prototype, has some limitations and unsolved details. These limitations can be overcome by an inter-disciplinary work with the help of ideas that may come from further studies of medical experts and our group.

1. In triangulation, the ambiguity problem, originated from Marching Cubes algorithm, occurs in some high grid number selections. To solve this problem, some additional methods like co-cases and alternative look up tables were tried, but holes on the surface could not be eliminated completely.

2. The processing phase takes too long for segmentation. For a 50 sliced volumetric data, segmentation takes approximately 12 hours. This problem can be overcome by using more powerful workstations. Our prototype system has not supplied that such processing power.

3. OpenGL Library cannot be implemented as much as required. We need extra help for this tool. Meanwhile, experienced staff on three-dimensional graphics is needed.

4. Fully controllable MRI and CT equipment are required. For high success rate, we need the same type of images. The small differences in the image tone and contrast, cause wrong segmented biological tissues, so, we need to guarantee image reliability.

5. An operator is required to accomplish commands given by the surgeon. The commands about interactive visualization are now carried out by extra staff. The surgeons prefer controlling the vision themselves by using voice or other interfacing tools such as head movements and mouth.

6. Our vision system, consisting a HMD and camera, offers relatively low resolution. The HMD works maximum 640x480 graphic input and the resolution per eye is 320x240. This resolution may not satisfy the required vision requirements. However, using different high resolution HMD can solve this limitation.

7. Our HMD system is not comfortable as much as required. A small noise and temperature is felt during usage. Long employments may cause a little headache. Moreover, lighter and more comfortable HMDs are present in the market.

5.3 Comments About Our System from Surgeons

In the beginning of the study, surgeons were suspicions about our method. These concerns are indeterminate conditions of the usage of extra vision system that is different from the surgeon's eyes. However, since microscopes and extra lenses for some surgeries are in use, surgeons accept our system as a new visualization system when it reaches to comfortable visual level. Dr. Muharrem Akguner, a plastic-reconstructive surgeon of Dokuz Eylül University Medical School, found our system

as "utopia" in the beginning. But, recent developments and surveys (Cinguin et al, 1995), (Rosen & Soltanian & Redett & Laub, 1996), (Toriwaki & Mori, 1999), (Nowinsky & Yang & Yeo, 2000) (Sorid & Moore, 2000) in the medical area proved that, this type of visual assistance systems will help the surgeons more than conventional diagnostic tools.

Additionally, surgeons warn that too much information on the vision causes distractions for operators. Instead, Dr. Mustafa Özer from Eşrefpaşa Hospital, suggests using overlay data whenever the surgeon needs. This suggestion is adequate and not difficult to implement in our system.

Surgeons think that the usage of three-dimensional views in the assistance system may be helpful. However, overlay of reconstructed volume on the real video, showing surgery area, may cover the operation area, this may cause loss of anatomic detail. Instead, they suggest that, it may be more convenient to use this overlay temporarily, before or after the surgery, or sometimes during the surgery.

A necessity of surgeons is being able to display the location of the surgery tools on the reconstructed three-dimensional model of the tissues. For this purpose, highly efficient three-dimensional positioners (Cinquin et al, 1995) and their proper montage to the surgery equipment are required. This study should be investigated in the future.

Finally, surgeons' evaluations on our system indicate that our system could be applicable in the surgeries as an assistance tool after the limitations of the system are resolved.

5.4 Future Plans

The next step of the study is to turn the prototype into a working system. Therefore, we require an inter-disciplinary teamwork among the medical staff and hardware and software experts. The main method in these efforts is to solve problems and limitations given in section 5.3 and redesign the system according to

suggestions from the surgeons as expressed in the section 5.4. Financial support must also be supplied before this phase.

Another possible application area of the system is remote consultancy between two medical points. Another benefit of remote consultancy is the education of medical students. For this purpose, Internet, satellite links and wireless communication systems can be used.

Another approach for this system could be remote surgery performance. Utilizing advanced robotics equipment, it may be possible to realize a remote surgery between distant points, even in space.

REFERENCES

Ahmed M.N., Farag A.A., (1996). 3D Segmentation and Labeling Using Unsupervised Clustering for Volumetric Measurements on Brain CT Imaging. ANNIE'96, Artificial Neural Networks in Engineering, 1-15.

Ahmed M.N., Farag A.A., (1997). Two-Stage Neural Network for Volume Segmentation of Medical Images. Pattern Recognition Letters, 18, 11-13, 1141-1151.

Ahmed M.N., Farag A.A., (1997). 3D Segmentation and Labeling using Self-Organizing Kohonen Network for Volumetric Measurements on Brain CT Images. Proc. of 18th Annual Conf. on IEEE Engineering. in Medicine and Biology, 738-739.

Ahmed M.N., Farag A.A., (1997). Volume Segmentation of CT/MRI Images using Multiscale Features, Self-Organizing Principal Components Analysis (SOPCA), and Self-Organizing Feature Map (SOFM). Int. Conf on artificial Neural Networks ICNN'97.

Akay M., (1996). The VRT Revolution: Applications Range from Telesurgery to Monitoring Early Human Development. IEEE Engineering in Medicine and Biology, 31-33.

Atkins M.S., Mackiewich T., (1998). Fully Automatic Segmentation of the Brain in MRI. IEEE Trans. on Medical Imaging, 17, 1, 98-107.

Azuma, R. T., (1995). A survey of augmented reality. Computer Graphics. SIGGRAPH '95 Proceedings, Course Notes #9: Developing Advanced Virtual Reality Applications, 1-38.

Bajura, M., Fuchs, H., and Ohbuchi, R., (1992). Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery within the Patient. Computer Graphics Proceedings of SIGGRAPH'92, 26, 2, 203-210.

Bajura, M. and Neumann, U. (1995). Dynamic Registration and Correction in Augmented Reality Systems. Proc. VRAIS '95 Virtual Reality Annual International Symp., 189-196.

Barber C.B., Dopkin D.P., Huhdanpaa H., (1996). The Quickhull Algorithm for Convex Hulls. ACM Trans. on Mathematical Sofware, 22, 4, 469-483.

Berg M.D. , Kreveld M.V., Overmars M., Scwarzkopt O., (1997). Computational Geometry : Algorithms and Applications. Springer Verlag.

Bielser D., Maiwald V.A., Gross M.H., (1999). Interactive Cuts through 3-Dimensional Soft Tissue. Proc. of EuroGraphics'99, 18, 3, C31-C38.

Blackwell M., Nikou C., DiGioia A.M., Kanade T., (2000). An Image Overlay System for Medical Data Visualization. Elsevier Medical Image Analysis, 4, 67-72.

Bomans M., Höhne K., Tiede U., Riemer M., (1990). 3-D Segmentation of MR Images of the Head for 3-D Display. IEEE Trans. on Medical Imaging, 9, 2, 177-183.

Brink A.D., (1996). Using Spatial Information as an Aid to Maximum Entropy Image Threshold Selection. Elsevier Pattern Recognition Letters, 17, 29-36.

Brown M. A.,. Semelka R.C., (1999). MRI: Basic Principles and Applications.

Burckhardt C.W., Flury P., Glauser D., (1995). Stereotactic Brain Surgery. IEEE Engineering Medicine and Biology, May/June, 314-316.

Caudell, T. and Mizell, D. (1992). Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes. Proc. Hawaii International Conf. on Systems Science, 2, 659-669.

Chabat F., Hansell D.V., Yang G., (2000). Computorized Decision Support in Medical Imaging. IEEE Engineering in Medicine and Biology, September/October, 89- 96.

Chan S.L., Purisima E.O., (1997).A new Tethedral Tessellation Scheme for Isosurface Generation. Comput. & Graphics, 22, 1,83-90.

Cignoni, P., Montani,C., Puppo, E., Scopigno, R., (1996). Optimal Isosurface Extraction from Irregular Volume Data. Proceedings of IEEE Symposium on Volume Visualization Symposium. 31-38.

Cinguin P. et al, (995). Computer Assisted Medical Interventions, IEEE Engineering in Medicine and Biology. 1995, 254-262.

Cinque L., Romagnoli R., Levialdi S., Nguyen P.T.A., Guan L., (1998). Self-Organizing Map for Segmenting 3D Biological Images. 14th Inter. Conf. on pattern Recognition. 471-473.

Clark M., Hall L.O., Goldgot D.B., Clarle L.P., Velthuizen R.P., Silbiger M.S., (1994). MRI Segmentation Using Fuzzy Clustering Techniques. IEEE Engineering in Medicine and Biology, November/December. 730-733.

Cline HE, Lorensen WE.,Kikinis, R.,et al.(1990). 3-D segmentation of MR images of the head using probability and connectivity. Journal of Computer Asisted Tomography. 14, 6, 1037.1045.

Dumay A.C.M., (1996). Beyond Medicine. IEEE Engineering Medicine and Biology, March/April, 34-40.

Duta N. and Sonka M.,(1997). Segmentation and interpretation of MR brain images using an improved knowledge-based active shape models. Information Processing in Medical Imaging, 375-380.

Ehricke H. H., (1991). Automated 3D MR image analysis with a rule-based segmentation system. CAR '91 Computer Assisted Radiology, 543-548.

Faulkner G., Krauss M., (1996). Guidelines for Establishing a Virtua Reality Lab. IEEE Engineering in Medicine and Biology, March/April, 86-93.

Feiner S., MacIntyre B., Seligmann D., (1993). Knowledge-based Augmented Reality. Communications of the ACM, 36, 7, 52-62.

Gelenbe E., Feng Y., Ranga K., Krishman R., (1996). Neural Networks for Volumetric MR Imaging of the Brain. Int. Workshop on Neural Networks for Identification, Control, Robotics and Signal/Image Proc. (NICROSP'96), 194-202.

Green P.s., Hill J.W., Jensen J.F., Shah A., (1995). Telepresence Surgery. IEEE Engineering in Medicine and Biology, May/June, 324-329.

Greenleaf W.J., (1996). Developing the Tools for Practical VR Applications. IEEE Engineering in Medicine and Biology, 23-30.

Grimson, W., Ettinger G., White S.J., Lozano-Perez T., Wells W.M., Kikinis R., (1996). An Automatic Registration Method for Frameless Stereotaxy, Image Guided Surgery, and Enhanced Reality Visualization. IEEE Trans. on Medical Imaging, 15, 2, 129-140.

Grimson W., Ettinger G., Kapur T., Leventon M., Wells W., and Kikinis R., (1998). Utilizing Segmented MRI Data in Image-Guided Surgery. International Journal of Pattern Recognition and Artificial Intelligence, 11, 8, 1367-1397.

Guha S., (1994). An Optimal Mesh Computer Algorithm for Constrained Delaunay Triangulation. 8th Inter. Parallel Procesing Symp. IPPS'94, 102-109.

Haigron P., Berre G.L., Coatrieux J.L., (1996). 3D Navigation in Medicine. IEEE Engineering in Medicine and Biology, March/April, 70-78.

Hall L.O. et. al., (1992). A Comparison of Neural Network and Fuzzy Clustering Techniques in Segmenting Magnetic Resonance Images of the Brain. IEEE Trans. on Nural Networks, 3, 5, 672-682.

Haykin S., (1999). Neural Networks: A Comprehensive Foundation.

Hege H.C., Seebaß M., Stalling D., Zockler A., (1997). Generalized Marching Cubes Algorithm Based on Non-Binary Classifications, ZIB (Zentrum für Informationstechnik) Preprint SC 07-05.

Held M., (1998). Efficient and Reliable Triangulation of Polygons. Proc. Of International Conf. on Comp. Graphics, 633-643.

Held K., Kops E.R., Krause B., Wells W.M., Kikinis R., Muller-Gartner H., (1997). Markov Random Field Segmentation of Brain Images. IEEE Trans. on Medical Imaging, 16, 6, 878-886.

Hoppe H., DeRose T, Duchamp T., McDonald J., and Stuetzle W., (1992). Surface Reconstruction from Unorganized Points. Computer Graphics (SIGGRAPH '92 Proceedings), 26, 2, 71-78.

Ishikawa M., Kamiura N., Hata Y., (1999). Thresholding Based Image Segmentation Aided by Kleene Algebra. IEICE Trans. Inf. & Syst., E82-D, 5, 962-967.

Janin A.L., Mizell D.W., Caudell T.P., (1993). Calibration of Head-Mounted Displays for Augmented Reality Applications. Proceedings of IEEE Virtual Reality Annual International Symposium (VRAIS'93), 246-255.

Kalawsky, R. S., (1993). Science and Engineering Issues of Virtual Environment Systems. Proceedings of 10th Anniversary Eurographics UK Conference, 137-146.

Kassell N., and Graves B., (1997). Telepresence in Neurosurgery: The Integrated Remote Neurosurgical System. Proceedings of Medicine Meets Virtual Reality 5.

Keller P.J., (1988). Basic Principles of Magnetic Resonance Imaging. GE Medical Systems.

Kenmochi y., Kotani K., Imiya A., (1999). Marching Cubes Method with Connectivity. ICIP 99, Intern. Conf. on Image Proces. , 4, 361-365.

Kikinis R, Jolesz FA, Gerig G, et al. (1990). 3D mofphometric and morphologic information derived from clinical brain MR images. 3D Imaging in Medicine. Nato ASI Series F: Computer and Systems Sciences, 60, 441-454.

Kikinis R., Jolesz F.A., Cline H. A., Lorensen W.E., Gerig G. (1991). The Potential use of MRI Guidance for Computerized Surgical Procedures. Annual International Conference of the IEEE Engineering and Medicine and Biology Society. 13, 1, 0303-304.

Kikinis R. Cline H., Altobelli D., Halle M., Lorensen W., Rhodes B., Jolesz F., (1992). Computer Assisted Planning of surgical Procedures. IEEE-EMBS Conf., 2065-2066.

Kikinis R., (1994). Virtual Reality in MRI. Inter. Conf. Proc. Combined Volumes, 629-631.

Kohonen T., (1990). The Self-organizing Map. Proceedings of the IEEE, 78, 9, 1464-1480.

Kohonen T., (1997). Self-organizing Maps. Springer.

Kühnapfel U., Krumm H.G., Kuhn Ch., Hübner M., Neisius B., (1995). Endosurgery Simulations with KISMET: A flexible tool for Surgical Instrument Design, Operation Room Planning and VR Technology based Abdominal Surgery Training. Proc. VR World95.

Kuhne G., Poliwoda C., Reinhart C., Gunther T., Hesser J., and Manner R., (1997). Interactive Segmentation and Visualization of Volume Data Sets. Proc. Visualization '97.

Li, J., Agathoklis P., (1993). A Case Study of Isosurface Generation in 3D Visualization. IEEE Pac Rim'93.

Lorensen W.E., Cline H.E., (1987). Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm. Proc. of SIGGRAPH'87 Computer Graphics, 21, 4, 163-169.

Lorensen W. E., (1995). Marching through the Visible Man. Proceedings of IEEE Visualization '95, 368--373.

Ma F., Wang W., Tsang W.W., Tang Z, Xia S., and Tong X., (1998). Probabilistic segmentation of volume data for visualization using SOM-PNN classifier. Proceedings of the 1998 IEEE Symposium on Volume visualization, 71-78

MacIntyre B., Feiner S.,.(1996). Language-Level Support for Exploratory Programming of Distributed Virtual Environments. Proc. UIST '96 ACM Symp. on User Interface Software and Technology, 83-95.

Matveyev S.V.,(1994). Approximation of Isosurface in the Marching Cubes: Ambiguity Problem. Visualization '94, 288-292.

Mohan S., Lee J., (1993). Three Dimensional finite Element Mesh Generation Using Delaunay Triangulation. IEEE AP-S/ URSI Int. Symp., 312-315.

Montani C., Scateni R., and Scopigno R., (1994). Discretized Marching Cubes. IEEE Visualization '94 Proceedings, 281-287.

Montani C., Scateni R., and Scopigno R., (2000). Decreasing Isosurface Complexity via Discrete Fitting. Elsevier Comp. Aided Geometric Design, 17, 207-232.

Namasivayam A., Hall L.O., (1996). Integrating Fuzzy Rules into Fast, Robust Segmentation of Magnetic Resonance Images. NAFIPS'96, 23-27.

Nielson G.M., Hamann B., (1991).The Asymtotic Decider: Resolving the Ambiguity in Marching Cubes. Procedings of Visualization'91, 83-91.

Niessen W.J., Vincken K.L., Weickert J., Viergever M.A., (1998). Three-dimensional MR Brain Segmentation. 6th Inter. Conf. on Computer Vision, Bombay, 53-58.

Nowinsky W.L., Yang, G.L., Yeo T. T., (2000). Computer Aided Stereotactic Functional Surgery Enhanced by the Use of Multiple Brain Atlasses. IEEE Transactions on Medical Imaging, 19, 1, 62-69.

Parker, J.R., (1997). Algorithms for Image Processing and Computer Vision, John Wiley & Sons, Inc.

Rajapakse J., Giedd J.N., Rapoport J.L., (1997). Statistical Approach to Segmentation of Single-Channel Cerebral MR Images. IEEE Trans. on Medical Imaging, 16, 2, 176-186.

Riederer S. J., (1988). Recent Advances in Magnetic Resonance Imaging. Proc. of IEEE, 76, 9, 1095-1105.

Robb, R. A..(1996). VR Assisted Surgery Planning. IEEE Engineering in Medicine and Biology, 60-69.

Robinett, W. (1992). Synthetic Experience: A Taxonomy Presence. Teleoperators and Virtual Environments, 1,2.

Rosen J. M., Soltanian H., Redett R. J., Laub D.R., (1996). Evolution of Virtual Reality: From Planning to Performing Surgery, IEEE Engineering in Medicine and Biology, March/April, 16-22.

Rovetta A. et. al., (1993). The First Experiment in the World of Robotic Telesurgery for Laparoscopy Carried Out by Means of Satellites Networks and Optical Fibres Networks on 7th July 1993, Proc. of Inter. Conf. on Industrial Electronics, Control and Instrumentation IECON'93, 51-56.

Shadidi R., Tombropoulos R., Grzeszczuk R.P., (1998). Clinical Applications of Three-Dimensional Rendering of Medical Data Sets. Invited Paper. Proc. of the IEEE, 86, 3, 555-568.

Shareef N., Wang D.L., Yagel, R., (1999). Segmentation of Medical Images Using LEGION. IEEE Trans. on Medical Imaging, 18, 1, 74-91.

Shroeder W., Martin K., Lorensen B., (1998). The Visualization toolkit: An object Oriented approach to 3D graphics. 2nd edition, Prentice Hall, ISBN 0-13-954694-4, 159-164

Sorid D., Moore S.K., (2000). The Virtual Surgeon. IEEE Spectrum, 26-31.

Taleb-Ahmed A., Thuc V. Le,. Michel T. S., Ayachi M., and Pruvo J.P., (1999). Automatic Segmentation of Multiple Sclerosis Lesions from MRI Slices. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing NSIP'99.

Tang S.,. Kwoh C, Teo M., Sing N.W., Ling, K. (1998). Augmented Reality Systems for Medical Applications. IEEE Engineering Medicine and Biology, 49-58

Taylor R. H. et al, (1995). A Telerobotic Assistant for Laparoscopic Surgery. IEEE Engineering in Medicine and Biology, May/June, 279-288.

Toriwaki J., Mori K., (1999). Recent Progress in Medical Image Processing-Virtualized Human Body and Computer-Aided Surgery. IEICE Trans. Inf. & Syst., E82-D, 3, 611-628.

Valova I., Suganami Y., Kosugi Y., (1998). Conditional-Class-Entropy-Based Segmentation of Brain MR Images on a Neural Tree Classifier. IEICE Trans. Inf. & Syst., E81-D, 4, 382-390.

Vitulano S., Rubrto C, Di Nappi M., (1997). Different Methods to Segment Biological Images. Elsevier Pattern Recognition Letters, 18, 1125-1131.

Wang X., Feng D., Shi J., (1997). A New Algorithm for Multiresolution Isosurface Extraction. IEEE Conference on Information Visualization IV '97, 330-336

Webster A., Feiner S., MacIntyre B., Massie W., Krueger T., (1996). Augmented Reality in Architectural Construction, Inspection and Renovation. Proc. ASCE Third Congress on Computing in Civil Engineering, 913-919.

Wehrli, F.W., MacFall, J.R., (1984). Mechanism of Contrast in NMR Imaging. Journal of . Compt. Assisted Tomography. 8, 369.

Wehrli F.W.,(Unknown). Principles of Magnetic Resonance.

Wiederhold M.D., (1996). From Virtual Worlds to the Therapist's Office. IEEE Engineering in Medicine and Biology, March/April, 44-46.

Xiaohan Y., Yla-Yaaski J., (1991). A New Algorithm for Image Segmentation Based on Region Growing and Edge Detection. Proc.of IEEE Inter. Symp. on Circuits and Systems, 1, 516-519.

Xiaohan Y., Yla-Yaaski J., (1992). Image Segmentation Combining Region Growing and Edge Detection. Proc. of Inter. Conf. on Pattern Recognition, 3, 481-484.

Xuan J., Adalı T., Wang Y., (1995). Segmentation of Magnetic Resonance Image Integrating Region Growing and Edge Detection. Proc. of IEEE Int. Conf. on ımage Processing ICIP'95, 3, 544-547.

APPENDIX A

MATLAB CODES

OpenFL.m for Feature Calculations

```
fid=fopen('1slice70.sdf','r');
F = fread(fid,'uint16');
data=reshape(F,256,256);
%data=fix(data(1:1:128,1:1:128)/256);
data=fix(data/256);
colormap(gray);
imagesc(data);
pause
tic
[m,n]=size(data);
maskSize=5;
offset=fix(maskSize/2);

energyx1=zeros(m-1,n-1);
momentumx1=zeros(m-1,n-1);
entropyx1=zeros(m-1,n-1);
energyx2=zeros(m-1,n-1);
momentumx2=zeros(m-1,n-1);
entropyx2=zeros(m-1,n-1);

energyy1=zeros(m-1,n-1);
momentumy1=zeros(m-1,n-1);
entropyy1=zeros(m-1,n-1);
energyy2=zeros(m-1,n-1);
momentumy2=zeros(m-1,n-1);
entropyy2=zeros(m-1,n-1);

correlx1=zeros(m-1,n-1);
correly1=zeros(m-1,n-1);
correlxy1=zeros(m-1,n-1);

k=0;
for i=offset+1:m-offset
```

```
  for j=offset+1:n-offset
    k=k+1;
pixe=[data(i-offset:i+offset,j-offset:j+offset)];
    pixels.x1=[data(i-offset:i+offset,j-offset:j+offset) data(i-offset:i+offset,j+offset)];
    pixels.x2=[data(i-offset:i+offset,j-offset) data(i-offset:i+offset,j-offset:j+offset) ];
pixels.y1=[data(i-offset:i+offset,j-offset:j+offset) ;
            data(i+offset,j-offset:j+offset)];
    pixels.y2=[data(i-offset,j-offset:j+offset) ;
            data(i-offset:i+offset,j-offset:j+offset) ];
pixels.x3=[data(i-offset:i+offset,j-offset:j+offset) data(i-offset:i+offset,j+offset) data(i-offset:i+offset,j+offset)];
    pixels.x4=[data(i-offset:i+offset,j-offset) data(i-offset:i+offset,j-offset) data(i-offset:i+offset,j-offset:j+offset) ];
pixels.y3=[data(i-offset:i+offset,j-offset:j+offset) ;
            data(i+offset,j-offset:j+offset);
            data(i+offset,j-offset:j+offset)];
    pixels.y4=[data(i-offset,j-offset:j+offset) ;
            data(i-offset,j-offset:j+offset) ;
            data(i-offset:i+offset,j-offset:j+offset) ];
deltax1=abs(pixels.x1-pixels.x2);
    deltax1=deltax1(:,2:end-1)+1;
    table=tabulate(reshape(deltax1,1,maskSize*(maskSize-1)));
    entropyx1(i,j)=sum(table(:,2).*log(table(:,2)+realmin));
    energyx1(i,j)=sum(table(:,2).^2);
    momentumx1(i,j)=sum((table(:,1)-1).*table(:,2));
deltax2=abs(pixels.x3-pixels.x4);
    deltax2=deltax2(:,3:end-2)+1;
    table=tabulate(reshape(deltax2,1,maskSize*(maskSize-2)));
    entropyx2(i,j)=sum(table(:,2).*log(table(:,2)+realmin));
    energyx2(i,j)=sum(table(:,2).^2);
    momentumx2(i,j)=sum((table(:,1)-1).*table(:,2));
deltay1=abs(pixels.y1-pixels.y2);
    deltay1=deltay1(2:end-1,:)+1;
    table=tabulate(reshape(deltay1,1,maskSize*(maskSize-1)));
    entropyy1(i,j)=sum(table(:,2).*log(table(:,2)+realmin));
    energyy1(i,j)=sum(table(:,2).^2);
    momentumy1(i,j)=sum((table(:,1)-1).*table(:,2));

    deltay2=abs(pixels.y3-pixels.y4);
    deltay2=deltay2(3:end-2,:)+1;
    table=tabulate(reshape(deltay2,1,maskSize*(maskSize-2)));
    entropyy2(i,j)=sum(table(:,2).*log(table(:,2)+realmin));
    energyy2(i,j)=sum(table(:,2).^2);
    momentumy2(i,j)=sum((table(:,1)-1).*table(:,2));
correlx1(i,j)=corr2(pixe,pixels.x1(1:5,1:5));
  correly1(i,j)=corr2(pixe,pixels.y1(1:5,1:5));
  correlxy1(i,j)=corr2(pixels.x1(1:5,1:5),pixels.y1(1:5,1:5));
```

```
end
end
toc
figure
subplot(2,3,1),imagesc(20*log10(energyx1))
subplot(2,3,2),imagesc(20*log10(entropyx1))
subplot(2,3,3),imagesc(20*log10(momentumx1))
subplot(2,3,4),imagesc(20*log10(energyx2))
subplot(2,3,5),imagesc(20*log10(entropyx2))
subplot(2,3,6),imagesc(20*log10(momentumx2))
figure
subplot(2,3,1),imagesc(20*log10(energyy1))
subplot(2,3,2),imagesc(20*log10(entropyy1))
subplot(2,3,3),imagesc(20*log10(momentumy1))
subplot(2,3,4),imagesc(20*log10(energyy2))
subplot(2,3,5),imagesc(20*log10(entropyy2))
subplot(2,3,6),imagesc(20*log10(momentumy2))
MaskSize=5;
NumberOfFeatures=37
[m,n]=size(data);
dataMtx=zeros((m-1)*(n-1),NumberOfFeatures+2);
k=0;
Index=[];
offset=(MaskSize-1)/2;
```

Devam.m for Learning Phase

```
k=0;
for i=1+offset:m-1-offset,
  for j=1+offset:n-1-offset,
    k=k+1;
    pixels=(data(i-offset:i+offset,j-offset:j+offset));
    pixels2=reshape(pixels,1,25);

dataMtx(k,:)=[pixels2,mean2(pixels),correlx1(i,j),correly1(i,j),correlxy1(i,j),momentumx1(i,j),momentumx2(i,j),momentumy1(i,j),m
omentumy2(i,j),entropyx1(i,j),entropyx2(i,j),entropyy1(i,j),entropyy2(i,j),i,j];
  end
end
figure
tic
MapSize=[9 9];
D=som_data_struct(dataMtx(:,1:NumberOfFeatures));
sMap=som_init(D,MapSize,'linear');
sMap=som_trainops(sMap,'batch');
```

```
sMap=som_train(sMap,D,100,NaN,NaN,NaN,NaN);
som_show(sMap,0);
toc
figure
Hits=som_hits(sMap,D)
som_show(sMap,1,'',[],Hits);
resultMtx=zeros(m-1,n-1,prod(MapSize));
for i=1:k,
  bmu=som_bmus(sMap,dataMtx(i,1:NumberOfFeatures));
  resultMtx(dataMtx(i,NumberOfFeatures+1),dataMtx(i,NumberOfFeatures+2),bmu)=dataMtx(i,5);
end
figure
colormap(gray);
for i=1:49, image(resultMtx(:,:,i)),i,pause,end
```

TestEt.m for Testing Phase

```
resultMtx=zeros(m-1,n-1,prod(MapSize));
for i=1:k,
  bmu=som_bmus(sMap,dataMtx(i,1:NumberOfFeatures));
  resultMtx(dataMtx(i,NumberOfFeatures+1),dataMtx(i,NumberOfFeatures+2),bmu)=dataMtx(i,5);
end
toc
figure
colormap(gray);
for i=1:49, image(resultMtx(:,:,i)),i,pause,end
```

Cooc.m for Co-Occurance Matrix

```
function [X1,Y1,X2,Y2]=cooc(M,SIZE)
[m,n]=size(M);
X1=zeros(SIZE); %sparse(SIZE);
Y1=X1;
X2=zeros(SIZE); %sparse(SIZE);
Y2=X2;
for i=1:m,
  for j=1:n-2
    X1(M(i,j)+1,M(i,j+1)+1)=X1(M(i,j)+1,M(i,j+1)+1)+1;
   X2(M(i,j)+1,M(i,j+2)+1)=X2(M(i,j)+1,M(i,j+2)+1)+1;
  end
end
for i=1:m-2,
  for j=1:n
```

```
    Y1(M(i,j)+1,M(i+1,j)+1)=Y1(M(i,j)+1,M(i+1,j)+1)+1;
    Y2(M(i,j)+1,M(i+2,j)+1)=Y2(M(i,j)+1,M(i+2,j)+1)+1;
  end
end
```

Energy.m for Energy Feature

```
function energy=Energy(M)
energy.x=sum(sum(M.x.^2));
energy.y=sum(sum(M.y.^2));
```

Entropy.m for Entropy Feature

```
function entropy=Entropy(M)
entropy.x=sum(sum(Mx.*log(Mx+realmin)));
entropy.y=sum(sum(My.*log(My+realmin)));
```

Moment.m for Moment Feature

```
function moment=Momentum(M)
[X,Y]=meshgrid(0:255,0:255);
moment.x=sum(sum( ((X-Y).^2).*Mx));
moment.y=sum(sum( ((X-Y).^2).*My));
```

Tektek.m for Manual Decision on Neuron's Hit

```
imagesc(data);
D=data;
s=0;
for i=1:49,

  for x=1:255,
    for y=1:127,
      if resultMtx(x,y,i)>0 D(x,y)=255; s=s+1;
      end
    end
  end
  i
  s
  imagesc(D);
  pause;
  D=data;
  imagesc(data);
  s=0;
end
```

APPENDIX B

MARCHING CUBES ALGORITHM TABLES

Edge Table by Cory Gene Bloyd*

```
int edgeTable[256]={
0x0  , 0x109, 0x203, 0x30a, 0x406, 0x50f, 0x605, 0x70c,
0x80c, 0x905, 0xa0f, 0xb06, 0xc0a, 0xd03, 0xe09, 0xf00,
0x190, 0x99 , 0x393, 0x29a, 0x596, 0x49f, 0x795, 0x69c,
0x99c, 0x895, 0xb9f, 0xa96, 0xd9a, 0xc93, 0xf99, 0xe90,
0x230, 0x339, 0x33 , 0x13a, 0x636, 0x73f, 0x435, 0x53c,
0xa3c, 0xb35, 0x83f, 0x936, 0xe3a, 0xf33, 0xc39, 0xd30,
0x3a0, 0x2a9, 0x1a3, 0xaa , 0x7a6, 0x6af, 0x5a5, 0x4ac,
0xbac, 0xaa5, 0x9af, 0x8a6, 0xfaa, 0xea3, 0xda9, 0xca0,
0x460, 0x569, 0x663, 0x76a, 0x66 , 0x16f, 0x265, 0x36c,
0xc6c, 0xd65, 0xe6f, 0xf66, 0x86a, 0x963, 0xa69, 0xb60,
0x5f0, 0x4f9, 0x7f3, 0x6fa, 0x1f6, 0xff , 0x3f5, 0x2fc,
0xdfc, 0xcf5, 0xfff, 0xef6, 0x9fa, 0x8f3, 0xbf9, 0xaf0,
0x650, 0x759, 0x453, 0x55a, 0x256, 0x35f, 0x55 , 0x15c,
0xe5c, 0xf55, 0xc5f, 0xd56, 0xa5a, 0xb53, 0x859, 0x950,
0x7c0, 0x6c9, 0x5c3, 0x4ca, 0x3c6, 0x2cf, 0x1c5, 0xcc ,
0xfcc, 0xec5, 0xdcf, 0xcc6, 0xbca, 0xac3, 0x9c9, 0x8c0,
0x8c0, 0x9c9, 0xac3, 0xbca, 0xcc6, 0xdcf, 0xec5, 0xfcc,
0xcc , 0x1c5, 0x2cf, 0x3c6, 0x4ca, 0x5c3, 0x6c9, 0x7c0,
0x950, 0x859, 0xb53, 0xa5a, 0xd56, 0xc5f, 0xf55, 0xe5c,
0x15c, 0x55 , 0x35f, 0x256, 0x55a, 0x453, 0x759, 0x650,
0xaf0, 0xbf9, 0x8f3, 0x9fa, 0xef6, 0xfff, 0xcf5, 0xdfc,
0x2fc, 0x3f5, 0xff , 0x1f6, 0x6fa, 0x7f3, 0x4f9, 0x5f0,
0xb60, 0xa69, 0x963, 0x86a, 0xf66, 0xe6f, 0xd65, 0xc6c,
0x36c, 0x265, 0x16f, 0x66 , 0x76a, 0x663, 0x569, 0x460,
0xca0, 0xda9, 0xea3, 0xfaa, 0x8a6, 0x9af, 0xaa5, 0xbac,
0x4ac, 0x5a5, 0x6af, 0x7a6, 0xaa , 0x1a3, 0x2a9, 0x3a0,
0xd30, 0xc39, 0xf33, 0xe3a, 0x936, 0x83f, 0xb35, 0xa3c,
0x53c, 0x435, 0x73f, 0x636, 0x13a, 0x33 , 0x339, 0x230,
0xe90, 0xf99, 0xc93, 0xd9a, 0xa96, 0xb9f, 0x895, 0x99c,
0x69c, 0x795, 0x49f, 0x596, 0x29a, 0x393, 0x99 , 0x190,
0xf00, 0xe09, 0xd03, 0xc0a, 0xb06, 0xa0f, 0x905, 0x80c,
0x70c, 0x605, 0x50f, 0x406, 0x30a, 0x203, 0x109, 0x0   };
```

Triangle Table by Cory Gene Bloyd*

```
int triTable[256][16] =
{{-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 8, 3, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 1, 9, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 8, 3, 9, 8, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 2, 10, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 8, 3, 1, 2, 10, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{9, 2, 10, 0, 2, 9, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{2, 8, 3, 2, 10, 8, 10, 9, 8, -1, -1, -1, -1, -1, -1, -1},
{3, 11, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 11, 2, 8, 11, 0, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 9, 0, 2, 3, 11, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 11, 2, 1, 9, 11, 9, 8, 11, -1, -1, -1, -1, -1, -1, -1},
{3, 10, 1, 11, 10, 3, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 10, 1, 0, 8, 10, 8, 11, 10, -1, -1, -1, -1, -1, -1, -1},
{3, 9, 0, 3, 11, 9, 11, 10, 9, -1, -1, -1, -1, -1, -1, -1},
{9, 8, 10, 10, 8, 11, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{4, 7, 8, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{4, 3, 0, 7, 3, 4, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 1, 9, 8, 4, 7, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{4, 1, 9, 4, 7, 1, 7, 3, 1, -1, -1, -1, -1, -1, -1, -1},
{1, 2, 10, 8, 4, 7, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{3, 4, 7, 3, 0, 4, 1, 2, 10, -1, -1, -1, -1, -1, -1, -1},
{9, 2, 10, 9, 0, 2, 8, 4, 7, -1, -1, -1, -1, -1, -1, -1},
{2, 10, 9, 2, 9, 7, 2, 7, 3, 7, 9, 4, -1, -1, -1, -1},
{8, 4, 7, 3, 11, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{11, 4, 7, 11, 2, 4, 2, 0, 4, -1, -1, -1, -1, -1, -1, -1},
{9, 0, 1, 8, 4, 7, 2, 3, 11, -1, -1, -1, -1, -1, -1, -1},
{4, 7, 11, 9, 4, 11, 9, 11, 2, 9, 2, 1, -1, -1, -1, -1},
{3, 10, 1, 3, 11, 10, 7, 8, 4, -1, -1, -1, -1, -1, -1, -1},
{1, 11, 10, 1, 4, 11, 1, 0, 4, 7, 11, 4, -1, -1, -1, -1},
{4, 7, 8, 9, 0, 11, 9, 11, 10, 11, 0, 3, -1, -1, -1, -1},
{4, 7, 11, 4, 11, 9, 9, 11, 10, -1, -1, -1, -1, -1, -1, -1},
{9, 5, 4, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{9, 5, 4, 0, 8, 3, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 5, 4, 1, 5, 0, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{8, 5, 4, 8, 3, 5, 3, 1, 5, -1, -1, -1, -1, -1, -1, -1},
{1, 2, 10, 9, 5, 4, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{3, 0, 8, 1, 2, 10, 4, 9, 5, -1, -1, -1, -1, -1, -1, -1},
{5, 2, 10, 5, 4, 2, 4, 0, 2, -1, -1, -1, -1, -1, -1, -1},
{2, 10, 5, 3, 2, 5, 3, 5, 4, 3, 4, 8, -1, -1, -1, -1},
{9, 5, 4, 2, 3, 11, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
```

Alternate Triangle Table by Heller*

```
int triTable[256][13] =
{   {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    { 8, 3, 0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    { 9, 0, 1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    { 8, 3, 1, 8, 1, 9,-1,-1,-1,-1,-1,-1,-1},
    {10, 1, 2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    { 8, 3, 0, 1, 2,10,-1,-1,-1,-1,-1,-1,-1},
    { 9, 0, 2, 9, 2,10,-1,-1,-1,-1,-1,-1,-1},
    { 3, 2, 8, 2,10, 8, 8,10, 9,-1,-1,-1,-1},
    {11, 2, 3,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    {11, 2, 0,11, 0, 8,-1,-1,-1,-1,-1,-1,-1},
    {11, 2, 3, 0, 1, 9,-1,-1,-1,-1,-1,-1,-1},
    { 2, 1,11, 1, 9,11,11, 9, 8,-1,-1,-1,-1},
    {10, 1, 3,10, 3,11,-1,-1,-1,-1,-1,-1,-1},
    { 1, 0,10, 0, 8,10,10, 8,11,-1,-1,-1,-1},
    { 0, 3, 9, 3,11, 9, 9,11,10,-1,-1,-1,-1},
    { 8,10, 9, 8,11,10,-1,-1,-1,-1,-1,-1,-1},
    { 8, 4, 7,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    { 3, 0, 4, 3, 4, 7,-1,-1,-1,-1,-1,-1,-1},
    { 1, 9, 0, 8, 4, 7,-1,-1,-1,-1,-1,-1,-1},
    { 9, 4, 1, 4, 7, 1, 1, 7, 3,-1,-1,-1,-1},
    {10, 1, 2, 8, 4, 7,-1,-1,-1,-1,-1,-1,-1},
    { 2,10, 1, 0, 4, 7, 0, 7, 3,-1,-1,-1,-1},
    { 4, 7, 8, 0, 2,10, 0,10, 9,-1,-1,-1,-1},
    { 2, 7, 3, 2, 9, 7, 7, 9, 4, 2,10, 9,-1},
    { 2, 3,11, 7, 8, 4,-1,-1,-1,-1,-1,-1,-1},
    { 7,11, 4,11, 2, 4, 4, 2, 0,-1,-1,-1,-1},
    { 3,11, 2, 4, 7, 8, 9, 0, 1,-1,-1,-1,-1},
    { 2, 7,11, 2, 1, 7, 1, 4, 7, 1, 9, 4,-1},
    { 8, 4, 7,11,10, 1,11, 1, 3,-1,-1,-1,-1},
    {11, 4, 7, 1, 4,11, 1,11,10, 1, 0, 4,-1},
    { 3, 8, 0, 7,11, 4,11, 9, 4,11,10, 9,-1},
    { 7,11, 4, 4,11, 9,11,10, 9,-1,-1,-1,-1},
    { 9, 5, 4,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    { 3, 0, 8, 4, 9, 5,-1,-1,-1,-1,-1,-1,-1},
    { 5, 4, 0, 5, 0, 1,-1,-1,-1,-1,-1,-1,-1},
    { 4, 8, 5, 8, 3, 5, 5, 3, 1,-1,-1,-1,-1},
    { 2,10, 1, 9, 5, 4,-1,-1,-1,-1,-1,-1,-1},
    { 0, 8, 3, 5, 4, 9,10, 1, 2,-1,-1,-1,-1},
    {10, 5, 2, 5, 4, 2, 2, 4, 0,-1,-1,-1,-1},
    { 3, 4, 8, 3, 2, 4, 2, 5, 4, 2,10, 5,-1},
    {11, 2, 3, 9, 5, 4,-1,-1,-1,-1,-1,-1,-1},
```

{0, 11, 2, 0, 8, 11, 4, 9, 5, -1, -1, -1, -1, -1, -1, -1},
{0, 5, 4, 0, 1, 5, 2, 3, 11, -1, -1, -1, -1, -1, -1, -1},
{2, 1, 5, 2, 5, 8, 2, 8, 11, 4, 8, 5, -1, -1, -1, -1},
{10, 3, 11, 10, 1, 3, 9, 5, 4, -1, -1, -1, -1, -1, -1, -1},
{4, 9, 5, 0, 8, 1, 8, 10, 1, 8, 11, 10, -1, -1, -1, -1},
{5, 4, 0, 5, 0, 11, 5, 11, 10, 11, 0, 3, -1, -1, -1, -1},
{5, 4, 8, 5, 8, 10, 10, 8, 11, -1, -1, -1, -1, -1, -1, -1},
{9, 7, 8, 5, 7, 9, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{9, 3, 0, 9, 5, 3, 5, 7, 3, -1, -1, -1, -1, -1, -1, -1},
{0, 7, 8, 0, 1, 7, 1, 5, 7, -1, -1, -1, -1, -1, -1, -1},
{1, 5, 3, 3, 5, 7, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{9, 7, 8, 9, 5, 7, 10, 1, 2, -1, -1, -1, -1, -1, -1, -1},
{10, 1, 2, 9, 5, 0, 5, 3, 0, 5, 7, 3, -1, -1, -1, -1},
{8, 0, 2, 8, 2, 5, 8, 5, 7, 10, 5, 2, -1, -1, -1, -1},
{2, 10, 5, 2, 5, 3, 3, 5, 7, -1, -1, -1, -1, -1, -1, -1},
{7, 9, 5, 7, 8, 9, 3, 11, 2, -1, -1, -1, -1, -1, -1, -1},
{9, 5, 7, 9, 7, 2, 9, 2, 0, 2, 7, 11, -1, -1, -1, -1},
{2, 3, 11, 0, 1, 8, 1, 7, 8, 1, 5, 7, -1, -1, -1, -1},
{11, 2, 1, 11, 1, 7, 7, 1, 5, -1, -1, -1, -1, -1, -1, -1},
{9, 5, 8, 8, 5, 7, 10, 1, 3, 10, 3, 11, -1, -1, -1, -1},
{5, 7, 0, 5, 0, 9, 7, 11, 0, 1, 0, 10, 11, 10, 0, -1},
{11, 10, 0, 11, 0, 3, 10, 5, 0, 8, 0, 7, 5, 7, 0, -1},
{11, 10, 5, 7, 11, 5, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{10, 6, 5, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 8, 3, 5, 10, 6, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{9, 0, 1, 5, 10, 6, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 8, 3, 1, 9, 8, 5, 10, 6, -1, -1, -1, -1, -1, -1, -1},
{1, 6, 5, 2, 6, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 6, 5, 1, 2, 6, 3, 0, 8, -1, -1, -1, -1, -1, -1, -1},
{9, 6, 5, 9, 0, 6, 0, 2, 6, -1, -1, -1, -1, -1, -1, -1},
{5, 9, 8, 5, 8, 2, 5, 2, 6, 3, 2, 8, -1, -1, -1, -1},
{2, 3, 11, 10, 6, 5, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{11, 0, 8, 11, 2, 0, 10, 6, 5, -1, -1, -1, -1, -1, -1, -1},
{0, 1, 9, 2, 3, 11, 5, 10, 6, -1, -1, -1, -1, -1, -1, -1},
{5, 10, 6, 1, 9, 2, 9, 11, 2, 9, 8, 11, -1, -1, -1, -1},
{6, 3, 11, 6, 5, 3, 5, 1, 3, -1, -1, -1, -1, -1, -1, -1},
{0, 8, 11, 0, 11, 5, 0, 5, 1, 5, 11, 6, -1, -1, -1, -1},
{3, 11, 6, 0, 3, 6, 0, 6, 5, 0, 5, 9, -1, -1, -1, -1},
{6, 5, 9, 6, 9, 11, 11, 9, 8, -1, -1, -1, -1, -1, -1, -1},
{5, 10, 6, 4, 7, 8, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{4, 3, 0, 4, 7, 3, 6, 5, 10, -1, -1, -1, -1, -1, -1, -1},
{1, 9, 0, 5, 10, 6, 8, 4, 7, -1, -1, -1, -1, -1, -1, -1},
{10, 6, 5, 1, 9, 7, 1, 7, 3, 7, 9, 4, -1, -1, -1, -1},
{6, 1, 2, 6, 5, 1, 4, 7, 8, -1, -1, -1, -1, -1, -1, -1},
{1, 2, 5, 5, 2, 6, 3, 0, 4, 3, 4, 7, -1, -1, -1, -1},

{ 9, 5, 4, 8,11, 2, 8, 2, 0,-1,-1,-1,-1},
{ 3,11, 2, 1, 5, 4, 1, 4, 0,-1,-1,-1,-1},
{ 8, 5, 4, 2, 5, 8, 2, 8,11, 2, 1, 5,-1},
{ 5, 4, 9, 1, 3,11, 1,11,10,-1,-1,-1,-1},
{ 0, 9, 1, 4, 8, 5, 8,10, 5, 8,11,10,-1},
{ 3, 4, 0, 3,10, 4, 4,10, 5, 3,11,10,-1},
{ 4, 8, 5, 5, 8,10, 8,11,10,-1,-1,-1,-1},
{ 9, 5, 7, 9, 7, 8,-1,-1,-1,-1,-1,-1,-1},
{ 0, 9, 3, 9, 5, 3, 3, 5, 7,-1,-1,-1,-1},
{ 8, 0, 7, 0, 1, 7, 7, 1, 5,-1,-1,-1,-1},
{ 1, 7, 3, 1, 5, 7,-1,-1,-1,-1,-1,-1,-1},
{ 1, 2,10, 5, 7, 8, 5, 8, 9,-1,-1,-1,-1},
{ 9, 1, 0,10, 5, 2, 5, 3, 2, 5, 7, 3,-1},
{ 5, 2,10, 8, 2, 5, 8, 5, 7, 8, 0, 2,-1},
{10, 5, 2, 2, 5, 3, 5, 7, 3,-1,-1,-1,-1},
{11, 2, 3, 8, 9, 5, 8, 5, 7,-1,-1,-1,-1},
{ 9, 2, 0, 9, 7, 2, 2, 7,11, 9, 5, 7,-1},
{ 0, 3, 8, 2, 1,11, 1, 7,11, 1, 5, 7,-1},
{ 2, 1,11,11, 1, 7, 1, 5, 7,-1,-1,-1,-1},
{ 3, 9, 1, 3, 8, 9, 7,11,10, 7,10, 5,-1},
{ 9, 1, 0,10, 7,11,10, 5, 7,-1,-1,-1,-1},
{ 3, 8, 0, 7,10, 5, 7,11,10,-1,-1,-1,-1},
{11, 5, 7,11,10, 5,-1,-1,-1,-1,-1,-1,-1},
{10, 6, 5,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{ 8, 3, 0,10, 6, 5,-1,-1,-1,-1,-1,-1,-1},
{ 0, 1, 9, 5,10, 6,-1,-1,-1,-1,-1,-1,-1},
{10, 6, 5, 9, 8, 3, 9, 3, 1,-1,-1,-1,-1},
{ 1, 2, 6, 1, 6, 5,-1,-1,-1,-1,-1,-1,-1},
{ 0, 8, 3, 2, 6, 5, 2, 5, 1,-1,-1,-1,-1},
{ 5, 9, 6, 9, 0, 6, 6, 0, 2,-1,-1,-1,-1},
{ 9, 6, 5, 3, 6, 9, 3, 9, 8, 3, 2, 6,-1},
{ 3,11, 2,10, 6, 5,-1,-1,-1,-1,-1,-1,-1},
{ 6, 5,10, 2, 0, 8, 2, 8,11,-1,-1,-1,-1},
{ 1, 9, 0, 6, 5,10,11, 2, 3,-1,-1,-1,-1},
{ 1,10, 2, 5, 9, 6, 9,11, 6, 9, 8,11,-1},
{11, 6, 3, 6, 5, 3, 3, 5, 1,-1,-1,-1,-1},
{ 0, 5, 1, 0,11, 5, 5,11, 6, 0, 8,11,-1},
{ 0, 5, 9, 0, 3, 5, 3, 6, 5, 3,11, 6,-1},
{ 5, 9, 6, 6, 9,11, 9, 8,11,-1,-1,-1,-1},
{10, 6, 5, 4, 7, 8,-1,-1,-1,-1,-1,-1,-1},
{ 5,10, 6, 7, 3, 0, 7, 0, 4,-1,-1,-1,-1},
{ 5,10, 6, 0, 1, 9, 8, 4, 7,-1,-1,-1,-1},
{ 4, 5, 9, 6, 7,10, 7, 1,10, 7, 3, 1,-1},
{ 7, 8, 4, 5, 1, 2, 5, 2, 6,-1,-1,-1,-1},
{ 4, 1, 0, 4, 5, 1, 6, 7, 3, 6, 3, 2,-1},

{8, 4, 7, 9, 0, 5, 0, 6, 5, 0, 2, 6, -1, -1, -1, -1},
{7, 3, 9, 7, 9, 4, 3, 2, 9, 5, 9, 6, 2, 6, 9, -1},
{3, 11, 2, 7, 8, 4, 10, 6, 5, -1, -1, -1, -1, -1, -1, -1},
{5, 10, 6, 4, 7, 2, 4, 2, 0, 2, 7, 11, -1, -1, -1, -1},
{0, 1, 9, 4, 7, 8, 2, 3, 11, 5, 10, 6, -1, -1, -1, -1},
{9, 2, 1, 9, 11, 2, 9, 4, 11, 7, 11, 4, 5, 10, 6, -1},
{8, 4, 7, 3, 11, 5, 3, 5, 1, 5, 11, 6, -1, -1, -1, -1},
{5, 1, 11, 5, 11, 6, 1, 0, 11, 7, 11, 4, 0, 4, 11, -1},
{0, 5, 9, 0, 6, 5, 0, 3, 6, 11, 6, 3, 8, 4, 7, -1},
{6, 5, 9, 6, 9, 11, 4, 7, 9, 7, 11, 9, -1, -1, -1, -1},
{10, 4, 9, 6, 4, 10, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{4, 10, 6, 4, 9, 10, 0, 8, 3, -1, -1, -1, -1, -1, -1, -1},
{10, 0, 1, 10, 6, 0, 6, 4, 0, -1, -1, -1, -1, -1, -1, -1},
{8, 3, 1, 8, 1, 6, 8, 6, 4, 6, 1, 10, -1, -1, -1, -1},
{1, 4, 9, 1, 2, 4, 2, 6, 4, -1, -1, -1, -1, -1, -1, -1},
{3, 0, 8, 1, 2, 9, 2, 4, 9, 2, 6, 4, -1, -1, -1, -1},
{0, 2, 4, 4, 2, 6, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{8, 3, 2, 8, 2, 4, 4, 2, 6, -1, -1, -1, -1, -1, -1, -1},
{10, 4, 9, 10, 6, 4, 11, 2, 3, -1, -1, -1, -1, -1, -1, -1},
{0, 8, 2, 2, 8, 11, 4, 9, 10, 4, 10, 6, -1, -1, -1, -1},
{3, 11, 2, 0, 1, 6, 0, 6, 4, 6, 1, 10, -1, -1, -1, -1},
{6, 4, 1, 6, 1, 10, 4, 8, 1, 2, 1, 11, 8, 11, 1, -1},
{9, 6, 4, 9, 3, 6, 9, 1, 3, 11, 6, 3, -1, -1, -1, -1},
{8, 11, 1, 8, 1, 0, 11, 6, 1, 9, 1, 4, 6, 4, 1, -1},
{3, 11, 6, 3, 6, 0, 0, 6, 4, -1, -1, -1, -1, -1, -1, -1},
{6, 4, 8, 11, 6, 8, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{7, 10, 6, 7, 8, 10, 8, 9, 10, -1, -1, -1, -1, -1, -1, -1},
{0, 7, 3, 0, 10, 7, 0, 9, 10, 6, 7, 10, -1, -1, -1, -1},
{10, 6, 7, 1, 10, 7, 1, 7, 8, 1, 8, 0, -1, -1, -1, -1},
{10, 6, 7, 10, 7, 1, 1, 7, 3, -1, -1, -1, -1, -1, -1, -1},
{1, 2, 6, 1, 6, 8, 1, 8, 9, 8, 6, 7, -1, -1, -1, -1},
{2, 6, 9, 2, 9, 1, 6, 7, 9, 0, 9, 3, 7, 3, 9, -1},
{7, 8, 0, 7, 0, 6, 6, 0, 2, -1, -1, -1, -1, -1, -1, -1},
{7, 3, 2, 6, 7, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{2, 3, 11, 10, 6, 8, 10, 8, 9, 8, 6, 7, -1, -1, -1, -1},
{2, 0, 7, 2, 7, 11, 0, 9, 7, 6, 7, 10, 9, 10, 7, -1},
{1, 8, 0, 1, 7, 8, 1, 10, 7, 6, 7, 10, 2, 3, 11, -1},
{11, 2, 1, 11, 1, 7, 10, 6, 1, 6, 7, 1, -1, -1, -1, -1},
{8, 9, 6, 8, 6, 7, 9, 1, 6, 11, 6, 3, 1, 3, 6, -1},
{0, 9, 1, 11, 6, 7, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{7, 8, 0, 7, 0, 6, 3, 11, 0, 11, 6, 0, -1, -1, -1, -1},
{7, 11, 6, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{7, 6, 11, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{3, 0, 8, 11, 7, 6, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 1, 9, 11, 7, 6, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},

{ 9, 4, 5, 8, 0, 7, 0, 6, 7, 0, 2, 6,-1},
{ 4, 5, 9, 6, 3, 2, 6, 7, 3,-1,-1,-1,-1},
{ 7, 8, 4, 2, 3,11,10, 6, 5,-1,-1,-1,-1},
{11, 6, 7,10, 2, 5, 2, 4, 5, 2, 0, 4,-1},
{11, 6, 7, 8, 0, 3, 1,10, 2, 9, 4, 5,-1},
{ 6, 7,11, 1,10, 2, 9, 4, 5,-1,-1,-1,-1},
{ 6, 7,11, 4, 5, 8, 5, 3, 8, 5, 1, 3,-1},
{ 6, 7,11, 4, 1, 0, 4, 5, 1,-1,-1,-1,-1},
{ 4, 5, 9, 3, 8, 0,11, 6, 7,-1,-1,-1,-1},
{ 9, 4, 5, 7,11, 6,-1,-1,-1,-1,-1,-1,-1},
{10, 6, 4,10, 4, 9,-1,-1,-1,-1,-1,-1,-1},
{ 8, 3, 0, 9,10, 6, 9, 6, 4,-1,-1,-1,-1},
{ 1,10, 0,10, 6, 0, 0, 6, 4,-1,-1,-1,-1},
{ 8, 6, 4, 8, 1, 6, 6, 1,10, 8, 3, 1,-1},
{ 9, 1, 4, 1, 2, 4, 4, 2, 6,-1,-1,-1,-1},
{ 1, 0, 9, 3, 2, 8, 2, 4, 8, 2, 6, 4,-1},
{ 2, 4, 0, 2, 6, 4,-1,-1,-1,-1,-1,-1,-1},
{ 3, 2, 8, 8, 2, 4, 2, 6, 4,-1,-1,-1,-1},
{ 2, 3,11, 6, 4, 9, 6, 9,10,-1,-1,-1,-1},
{ 0,10, 2, 0, 9,10, 4, 8,11, 4,11, 6,-1},
{10, 2, 1,11, 6, 3, 6, 0, 3, 6, 4, 0,-1},
{10, 2, 1,11, 4, 8,11, 6, 4,-1,-1,-1,-1},
{ 1, 4, 9,11, 4, 1,11, 1, 3,11, 6, 4,-1},
{ 0, 9, 1, 4,11, 6, 4, 8,11,-1,-1,-1,-1},
{11, 6, 3, 3, 6, 0, 6, 4, 0,-1,-1,-1,-1},
{ 8, 6, 4, 8,11, 6,-1,-1,-1,-1,-1,-1,-1},
{ 6, 7,10, 7, 8,10,10, 8, 9,-1,-1,-1,-1},
{ 9, 3, 0, 6, 3, 9, 6, 9,10, 6, 7, 3,-1},
{ 6, 1,10, 6, 7, 1, 7, 0, 1, 7, 8, 0,-1},
{ 6, 7,10,10, 7, 1, 7, 3, 1,-1,-1,-1,-1},
{ 7, 2, 6, 7, 9, 2, 2, 9, 1, 7, 8, 9,-1},
{ 1, 0, 9, 3, 6, 7, 3, 2, 6,-1,-1,-1,-1},
{ 8, 0, 7, 7, 0, 6, 0, 2, 6,-1,-1,-1,-1},
{ 2, 7, 3, 2, 6, 7,-1,-1,-1,-1,-1,-1,-1},
{ 7,11, 6, 3, 8, 2, 8,10, 2, 8, 9,10,-1},
{11, 6, 7,10, 0, 9,10, 2, 0,-1,-1,-1,-1},
{ 2, 1,10, 7,11, 6, 8, 0, 3,-1,-1,-1,-1},
{ 1,10, 2, 6, 7,11,-1,-1,-1,-1,-1,-1,-1},
{ 7,11, 6, 3, 9, 1, 3, 8, 9,-1,-1,-1,-1},
{ 9, 1, 0,11, 6, 7,-1,-1,-1,-1,-1,-1,-1},
{ 0, 3, 8,11, 6, 7,-1,-1,-1,-1,-1,-1,-1},
{11, 6, 7,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{11, 7, 6,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{ 0, 8, 3,11, 7, 6,-1,-1,-1,-1,-1,-1,-1},
{ 9, 0, 1,11, 7, 6,-1,-1,-1,-1,-1,-1,-1},

{8, 1, 9, 8, 3, 1, 11, 7, 6, -1, -1, -1, -1, -1, -1, -1},
{10, 1, 2, 6, 11, 7, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 2, 10, 3, 0, 8, 6, 11, 7, -1, -1, -1, -1, -1, -1, -1},
{2, 9, 0, 2, 10, 9, 6, 11, 7, -1, -1, -1, -1, -1, -1, -1},
{6, 11, 7, 2, 10, 3, 10, 8, 3, 10, 9, 8, -1, -1, -1, -1},
{7, 2, 3, 6, 2, 7, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{7, 0, 8, 7, 6, 0, 6, 2, 0, -1, -1, -1, -1, -1, -1, -1},
{2, 7, 6, 2, 3, 7, 0, 1, 9, -1, -1, -1, -1, -1, -1, -1},
{1, 6, 2, 1, 8, 6, 1, 9, 8, 8, 7, 6, -1, -1, -1, -1},
{10, 7, 6, 10, 1, 7, 1, 3, 7, -1, -1, -1, -1, -1, -1, -1},
{10, 7, 6, 1, 7, 10, 1, 8, 7, 1, 0, 8, -1, -1, -1, -1},
{0, 3, 7, 0, 7, 10, 0, 10, 9, 6, 10, 7, -1, -1, -1, -1},
{7, 6, 10, 7, 10, 8, 8, 10, 9, -1, -1, -1, -1, -1, -1, -1},
{6, 8, 4, 11, 8, 6, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{3, 6, 11, 3, 0, 6, 0, 4, 6, -1, -1, -1, -1, -1, -1, -1},
{8, 6, 11, 8, 4, 6, 9, 0, 1, -1, -1, -1, -1, -1, -1, -1},
{9, 4, 6, 9, 6, 3, 9, 3, 1, 11, 3, 6, -1, -1, -1, -1},
{6, 8, 4, 6, 11, 8, 2, 10, 1, -1, -1, -1, -1, -1, -1, -1},
{1, 2, 10, 3, 0, 11, 0, 6, 11, 0, 4, 6, -1, -1, -1, -1},
{4, 11, 8, 4, 6, 11, 0, 2, 9, 2, 10, 9, -1, -1, -1, -1},
{10, 9, 3, 10, 3, 2, 9, 4, 3, 11, 3, 6, 4, 6, 3, -1},
{8, 2, 3, 8, 4, 2, 4, 6, 2, -1, -1, -1, -1, -1, -1, -1},
{0, 4, 2, 4, 6, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 9, 0, 2, 3, 4, 2, 4, 6, 4, 3, 8, -1, -1, -1, -1},
{1, 9, 4, 1, 4, 2, 2, 4, 6, -1, -1, -1, -1, -1, -1, -1},
{8, 1, 3, 8, 6, 1, 8, 4, 6, 6, 10, 1, -1, -1, -1, -1},
{10, 1, 0, 10, 0, 6, 6, 0, 4, -1, -1, -1, -1, -1, -1, -1},
{4, 6, 3, 4, 3, 8, 6, 10, 3, 0, 3, 9, 10, 9, 3, -1},
{10, 9, 4, 6, 10, 4, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{4, 9, 5, 7, 6, 11, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 8, 3, 4, 9, 5, 11, 7, 6, -1, -1, -1, -1, -1, -1, -1},
{5, 0, 1, 5, 4, 0, 7, 6, 11, -1, -1, -1, -1, -1, -1, -1},
{11, 7, 6, 8, 3, 4, 3, 5, 4, 3, 1, 5, -1, -1, -1, -1},
{9, 5, 4, 10, 1, 2, 7, 6, 11, -1, -1, -1, -1, -1, -1, -1},
{6, 11, 7, 1, 2, 10, 0, 8, 3, 4, 9, 5, -1, -1, -1, -1},
{7, 6, 11, 5, 4, 10, 4, 2, 10, 4, 0, 2, -1, -1, -1, -1},
{3, 4, 8, 3, 5, 4, 3, 2, 5, 10, 5, 2, 11, 7, 6, -1},
{7, 2, 3, 7, 6, 2, 5, 4, 9, -1, -1, -1, -1, -1, -1, -1},
{9, 5, 4, 0, 8, 6, 0, 6, 2, 6, 8, 7, -1, -1, -1, -1},
{3, 6, 2, 3, 7, 6, 1, 5, 0, 5, 4, 0, -1, -1, -1, -1},
{6, 2, 8, 6, 8, 7, 2, 1, 8, 4, 8, 5, 1, 5, 8, -1},
{9, 5, 4, 10, 1, 6, 1, 7, 6, 1, 3, 7, -1, -1, -1, -1},
{1, 6, 10, 1, 7, 6, 1, 0, 7, 8, 7, 0, 9, 5, 4, -1},
{4, 0, 10, 4, 10, 5, 0, 3, 10, 6, 10, 7, 3, 7, 10, -1},
{7, 6, 10, 7, 10, 8, 5, 4, 10, 4, 8, 10, -1, -1, -1, -1},

{ 7, 6,11, 3, 1, 9, 3, 9, 8,-1,-1,-1,-1},
{ 1, 2,10, 6,11, 7,-1,-1,-1,-1,-1,-1,-1},
{ 2,10, 1, 7, 6,11, 8, 3, 0,-1,-1,-1,-1},
{11, 7, 6,10, 9, 0,10, 0, 2,-1,-1,-1,-1},
{ 7, 6,11, 3, 2, 8, 8, 2,10, 8,10, 9,-1},
{ 2, 3, 7, 2, 7, 6,-1,-1,-1,-1,-1,-1,-1},
{ 8, 7, 0, 7, 6, 0, 0, 6, 2,-1,-1,-1,-1},
{ 1, 9, 0, 3, 7, 6, 3, 6, 2,-1,-1,-1,-1},
{ 7, 6, 2, 7, 2, 9, 2, 1, 9, 7, 9, 8,-1},
{ 6,10, 7,10, 1, 7, 7, 1, 3,-1,-1,-1,-1},
{ 6,10, 1, 6, 1, 7, 7, 1, 0, 7, 0, 8,-1},
{ 9, 0, 3, 6, 9, 3, 6,10, 9, 6, 3, 7,-1},
{ 6,10, 7, 7,10, 8,10, 9, 8,-1,-1,-1,-1},
{ 8, 4, 6, 8, 6,11,-1,-1,-1,-1,-1,-1,-1},
{11, 3, 6, 3, 0, 6, 6, 0, 4,-1,-1,-1,-1},
{ 0, 1, 9, 4, 6,11, 4,11, 8,-1,-1,-1,-1},
{ 1, 9, 4,11, 1, 4,11, 3, 1,11, 4, 6,-1},
{10, 1, 2,11, 8, 4,11, 4, 6,-1,-1,-1,-1},
{10, 1, 2,11, 3, 6, 6, 3, 0, 6, 0, 4,-1},
{ 0, 2,10, 0,10, 9, 4,11, 8, 4, 6,11,-1},
{ 2,11, 3, 6, 9, 4, 6,10, 9,-1,-1,-1,-1},
{ 3, 8, 2, 8, 4, 2, 2, 4, 6,-1,-1,-1,-1},
{ 2, 0, 4, 2, 4, 6,-1,-1,-1,-1,-1,-1,-1},
{ 1, 9, 0, 3, 8, 2, 2, 8, 4, 2, 4, 6,-1},
{ 9, 4, 1, 1, 4, 2, 4, 6, 2,-1,-1,-1,-1},
{ 8, 4, 6, 8, 6, 1, 6,10, 1, 8, 1, 3,-1},
{ 1, 0,10,10, 0, 6, 0, 4, 6,-1,-1,-1,-1},
{ 8, 0, 3, 9, 6,10, 9, 4, 6,-1,-1,-1,-1},
{10, 4, 6,10, 9, 4,-1,-1,-1,-1,-1,-1,-1},
{ 9, 5, 4, 7, 6,11,-1,-1,-1,-1,-1,-1,-1},
{ 4, 9, 5, 3, 0, 8,11, 7, 6,-1,-1,-1,-1},
{ 6,11, 7, 4, 0, 1, 4, 1, 5,-1,-1,-1,-1},
{ 6,11, 7, 4, 8, 5, 5, 8, 3, 5, 3, 1,-1},
{ 6,11, 7, 1, 2,10, 9, 5, 4,-1,-1,-1,-1},
{11, 7, 6, 8, 3, 0, 1, 2,10, 9, 5, 4,-1},
{11, 7, 6,10, 5, 2, 2, 5, 4, 2, 4, 0,-1},
{ 7, 4, 8, 2,11, 3,10, 5, 6,-1,-1,-1,-1},
{ 4, 9, 5, 6, 2, 3, 6, 3, 7,-1,-1,-1,-1},
{ 9, 5, 4, 8, 7, 0, 0, 7, 6, 0, 6, 2,-1},
{ 4, 0, 1, 4, 1, 5, 6, 3, 7, 6, 2, 3,-1},
{ 7, 4, 8, 5, 2, 1, 5, 6, 2,-1,-1,-1,-1},
{ 4, 9, 5, 6,10, 7, 7,10, 1, 7, 1, 3,-1},
{ 5, 6,10, 0, 9, 1, 8, 7, 4,-1,-1,-1,-1},
{ 5, 6,10, 7, 0, 3, 7, 4, 0,-1,-1,-1,-1},
{10, 5, 6, 4, 8, 7,-1,-1,-1,-1,-1,-1,-1},

{6, 9, 5, 6, 11, 9, 11, 8, 9, -1, -1, -1, -1, -1, -1, -1},
{3, 6, 11, 0, 6, 3, 0, 5, 6, 0, 9, 5, -1, -1, -1, -1},
{0, 11, 8, 0, 5, 11, 0, 1, 5, 5, 6, 11, -1, -1, -1, -1},
{6, 11, 3, 6, 3, 5, 5, 3, 1, -1, -1, -1, -1, -1, -1, -1},
{1, 2, 10, 9, 5, 11, 9, 11, 8, 11, 5, 6, -1, -1, -1, -1},
{0, 11, 3, 0, 6, 11, 0, 9, 6, 5, 6, 9, 1, 2, 10, -1},
{11, 8, 5, 11, 5, 6, 8, 0, 5, 10, 5, 2, 0, 2, 5, -1},
{6, 11, 3, 6, 3, 5, 2, 10, 3, 10, 5, 3, -1, -1, -1, -1},
{5, 8, 9, 5, 2, 8, 5, 6, 2, 3, 8, 2, -1, -1, -1, -1},
{9, 5, 6, 9, 6, 0, 0, 6, 2, -1, -1, -1, -1, -1, -1, -1},
{1, 5, 8, 1, 8, 0, 5, 6, 8, 3, 8, 2, 6, 2, 8, -1},
{1, 5, 6, 2, 1, 6, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 3, 6, 1, 6, 10, 3, 8, 6, 5, 6, 9, 8, 9, 6, -1},
{10, 1, 0, 10, 0, 6, 9, 5, 0, 5, 6, 0, -1, -1, -1, -1},
{0, 3, 8, 5, 6, 10, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{10, 5, 6, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{11, 5, 10, 7, 5, 11, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{11, 5, 10, 11, 7, 5, 8, 3, 0, -1, -1, -1, -1, -1, -1, -1},
{5, 11, 7, 5, 10, 11, 1, 9, 0, -1, -1, -1, -1, -1, -1, -1},
{10, 7, 5, 10, 11, 7, 9, 8, 1, 8, 3, 1, -1, -1, -1, -1},
{11, 1, 2, 11, 7, 1, 7, 5, 1, -1, -1, -1, -1, -1, -1, -1},
{0, 8, 3, 1, 2, 7, 1, 7, 5, 7, 2, 11, -1, -1, -1, -1},
{9, 7, 5, 9, 2, 7, 9, 0, 2, 2, 11, 7, -1, -1, -1, -1},
{7, 5, 2, 7, 2, 11, 5, 9, 2, 3, 2, 8, 9, 8, 2, -1},
{2, 5, 10, 2, 3, 5, 3, 7, 5, -1, -1, -1, -1, -1, -1, -1},
{8, 2, 0, 8, 5, 2, 8, 7, 5, 10, 2, 5, -1, -1, -1, -1},
{9, 0, 1, 5, 10, 3, 5, 3, 7, 3, 10, 2, -1, -1, -1, -1},
{9, 8, 2, 9, 2, 1, 8, 7, 2, 10, 2, 5, 7, 5, 2, -1},
{1, 3, 5, 3, 7, 5, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 8, 7, 0, 7, 1, 1, 7, 5, -1, -1, -1, -1, -1, -1, -1},
{9, 0, 3, 9, 3, 5, 5, 3, 7, -1, -1, -1, -1, -1, -1, -1},
{9, 8, 7, 5, 9, 7, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{5, 8, 4, 5, 10, 8, 10, 11, 8, -1, -1, -1, -1, -1, -1, -1},
{5, 0, 4, 5, 11, 0, 5, 10, 11, 11, 3, 0, -1, -1, -1, -1},
{0, 1, 9, 8, 4, 10, 8, 10, 11, 10, 4, 5, -1, -1, -1, -1},
{10, 11, 4, 10, 4, 5, 11, 3, 4, 9, 4, 1, 3, 1, 4, -1},
{2, 5, 1, 2, 8, 5, 2, 11, 8, 4, 5, 8, -1, -1, -1, -1},
{0, 4, 11, 0, 11, 3, 4, 5, 11, 2, 11, 1, 5, 1, 11, -1},
{0, 2, 5, 0, 5, 9, 2, 11, 5, 4, 5, 8, 11, 8, 5, -1},
{9, 4, 5, 2, 11, 3, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{2, 5, 10, 3, 5, 2, 3, 4, 5, 3, 8, 4, -1, -1, -1, -1},
{5, 10, 2, 5, 2, 4, 4, 2, 0, -1, -1, -1, -1, -1, -1, -1},
{3, 10, 2, 3, 5, 10, 3, 8, 5, 4, 5, 8, 0, 1, 9, -1},
{5, 10, 2, 5, 2, 4, 1, 9, 2, 9, 4, 2, -1, -1, -1, -1},
{8, 4, 5, 8, 5, 3, 3, 5, 1, -1, -1, -1, -1, -1, -1, -1},

{ 5, 6, 9, 6,11, 9, 9,11, 8,-1,-1,-1,-1},
{ 0, 9, 5, 0, 5, 3, 3, 5, 6, 3, 6,11,-1},
{ 0, 1, 5, 0, 5,11, 5, 6,11, 0,11, 8,-1},
{11, 3, 6, 6, 3, 5, 3, 1, 5,-1,-1,-1,-1},
{ 1, 2,10, 5, 6, 9, 9, 6,11, 9,11, 8,-1},
{ 1, 0, 9, 6,10, 5,11, 3, 2,-1,-1,-1,-1},
{ 6,10, 5, 2, 8, 0, 2,11, 8,-1,-1,-1,-1},
{ 3, 2,11,10, 5, 6,-1,-1,-1,-1,-1,-1,-1},
{ 9, 5, 6, 3, 9, 6, 3, 8, 9, 3, 6, 2,-1},
{ 5, 6, 9, 9, 6, 0, 6, 2, 0,-1,-1,-1,-1},
{ 0, 3, 8, 2, 5, 6, 2, 1, 5,-1,-1,-1,-1},
{ 1, 6, 2, 1, 5, 6,-1,-1,-1,-1,-1,-1,-1},
{10, 5, 6, 9, 3, 8, 9, 1, 3,-1,-1,-1,-1},
{ 0, 9, 1, 5, 6,10,-1,-1,-1,-1,-1,-1,-1},
{ 8, 0, 3,10, 5, 6,-1,-1,-1,-1,-1,-1,-1},
{10, 5, 6,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{11, 7, 5,11, 5,10,-1,-1,-1,-1,-1,-1,-1},
{ 3, 0, 8, 7, 5,10, 7,10,11,-1,-1,-1,-1},
{ 9, 0, 1,10,11, 7,10, 7, 5,-1,-1,-1,-1},
{ 3, 1, 9, 3, 9, 8, 7,10,11, 7, 5,10,-1},
{ 2,11, 1,11, 7, 1, 1, 7, 5,-1,-1,-1,-1},
{ 0, 8, 3, 2,11, 1, 1,11, 7, 1, 7, 5,-1},
{ 9, 0, 2, 9, 2, 7, 2,11, 7, 9, 7, 5,-1},
{11, 3, 2, 8, 5, 9, 8, 7, 5,-1,-1,-1,-1},
{10, 2, 5, 2, 3, 5, 5, 3, 7,-1,-1,-1,-1},
{ 5,10, 2, 8, 5, 2, 8, 7, 5, 8, 2, 0,-1},
{ 9, 0, 1,10, 2, 5, 5, 2, 3, 5, 3, 7,-1},
{ 1,10, 2, 5, 8, 7, 5, 9, 8,-1,-1,-1,-1},
{ 1, 3, 7, 1, 7, 5,-1,-1,-1,-1,-1,-1,-1},
{ 8, 7, 0, 0, 7, 1, 7, 5, 1,-1,-1,-1,-1},
{ 0, 3, 9, 9, 3, 5, 3, 7, 5,-1,-1,-1,-1},
{ 9, 7, 5, 9, 8, 7,-1,-1,-1,-1,-1,-1,-1},
{ 4, 5, 8, 5,10, 8, 8,10,11,-1,-1,-1,-1},
{ 3, 0, 4, 3, 4,10, 4, 5,10, 3,10,11,-1},
{ 0, 1, 9, 4, 5, 8, 8, 5,10, 8,10,11,-1},
{ 5, 9, 4, 1,11, 3, 1,10,11,-1,-1,-1,-1},
{ 8, 4, 5, 2, 8, 5, 2,11, 8, 2, 5, 1,-1},
{ 3, 2,11, 1, 4, 5, 1, 0, 4,-1,-1,-1,-1},
{ 9, 4, 5, 8, 2,11, 8, 0, 2,-1,-1,-1,-1},
{11, 3, 2, 9, 4, 5,-1,-1,-1,-1,-1,-1,-1},
{ 3, 8, 4, 3, 4, 2, 2, 4, 5, 2, 5,10,-1},
{10, 2, 5, 5, 2, 4, 2, 0, 4,-1,-1,-1,-1},
{ 0, 3, 8, 5, 9, 4,10, 2, 1,-1,-1,-1,-1},
{ 2, 1,10, 9, 4, 5,-1,-1,-1,-1,-1,-1,-1},
{ 4, 5, 8, 8, 5, 3, 5, 1, 3,-1,-1,-1,-1},

```
{0, 4, 5, 1, 0, 5, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{8, 4, 5, 8, 5, 3, 9, 0, 5, 0, 3, 5, -1, -1, -1, -1},
{9, 4, 5, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{4, 11, 7, 4, 9, 11, 9, 10, 11, -1, -1, -1, -1, -1, -1, -1},
{0, 8, 3, 4, 9, 7, 9, 11, 7, 9, 10, 11, -1, -1, -1, -1},
{1, 10, 11, 1, 11, 4, 1, 4, 0, 7, 4, 11, -1, -1, -1, -1},
{3, 1, 4, 3, 4, 8, 1, 10, 4, 7, 4, 11, 10, 11, 4, -1},
{4, 11, 7, 9, 11, 4, 9, 2, 11, 9, 1, 2, -1, -1, -1, -1},
{9, 7, 4, 9, 11, 7, 9, 1, 11, 2, 11, 1, 0, 8, 3, -1},
{11, 7, 4, 11, 4, 2, 2, 4, 0, -1, -1, -1, -1, -1, -1, -1},
{11, 7, 4, 11, 4, 2, 8, 3, 4, 3, 2, 4, -1, -1, -1, -1},
{2, 9, 10, 2, 7, 9, 2, 3, 7, 7, 4, 9, -1, -1, -1, -1},
{9, 10, 7, 9, 7, 4, 10, 2, 7, 8, 7, 0, 2, 0, 7, -1},
{3, 7, 10, 3, 10, 2, 7, 4, 10, 1, 10, 0, 4, 0, 10, -1},
{1, 10, 2, 8, 7, 4, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{4, 9, 1, 4, 1, 7, 7, 1, 3, -1, -1, -1, -1, -1, -1, -1},
{4, 9, 1, 4, 1, 7, 0, 8, 1, 8, 7, 1, -1, -1, -1, -1},
{4, 0, 3, 7, 4, 3, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{4, 8, 7, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{9, 10, 8, 10, 11, 8, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{3, 0, 9, 3, 9, 11, 11, 9, 10, -1, -1, -1, -1, -1, -1, -1},
{0, 1, 10, 0, 10, 8, 8, 10, 11, -1, -1, -1, -1, -1, -1, -1},
{3, 1, 10, 11, 3, 10, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 2, 11, 1, 11, 9, 9, 11, 8, -1, -1, -1, -1, -1, -1, -1},
{3, 0, 9, 3, 9, 11, 1, 2, 9, 2, 11, 9, -1, -1, -1, -1},
{0, 2, 11, 8, 0, 11, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{3, 2, 11, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{2, 3, 8, 2, 8, 10, 10, 8, 9, -1, -1, -1, -1, -1, -1, -1},
{9, 10, 2, 0, 9, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{2, 3, 8, 2, 8, 10, 0, 1, 8, 1, 10, 8, -1, -1, -1, -1},
{1, 10, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{1, 3, 8, 9, 1, 8, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 9, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{0, 3, 8, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1},
{-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1}};
```

```
{ 5, 0, 4, 5, 1, 0,-1,-1,-1,-1,-1,-1,-1},
{ 3, 8, 0, 4, 5, 9,-1,-1,-1,-1,-1,-1,-1},
{ 9, 4, 5,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{ 7, 4,11, 4, 9,11,11, 9,10,-1,-1,-1,-1},
{ 3, 0, 8, 7, 4,11,11, 4, 9,11, 9,10,-1},
{11, 7, 4, 1,11, 4, 1,10,11, 1, 4, 0,-1},
{ 8, 7, 4,11, 1,10,11, 3, 1,-1,-1,-1,-1},
{ 2,11, 7, 2, 7, 1, 1, 7, 4, 1, 4, 9,-1},
{ 3, 2,11, 4, 8, 7, 9, 1, 0,-1,-1,-1,-1},
{ 7, 4,11,11, 4, 2, 4, 0, 2,-1,-1,-1,-1},
{ 2,11, 3, 7, 4, 8,-1,-1,-1,-1,-1,-1,-1},
{ 2, 3, 7, 2, 7, 9, 7, 4, 9, 2, 9,10,-1},
{ 4, 8, 7, 0,10, 2, 0, 9,10,-1,-1,-1,-1},
{ 2, 1,10, 0, 7, 4, 0, 3, 7,-1,-1,-1,-1},
{10, 2, 1, 8, 7, 4,-1,-1,-1,-1,-1,-1,-1},
{ 9, 1, 4, 4, 1, 7, 1, 3, 7,-1,-1,-1,-1},
{ 1, 0, 9, 8, 7, 4,-1,-1,-1,-1,-1,-1,-1},
{ 3, 4, 0, 3, 7, 4,-1,-1,-1,-1,-1,-1,-1},
{ 8, 7, 4,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{ 8, 9,10, 8,10,11,-1,-1,-1,-1,-1,-1,-1},
{ 0, 9, 3, 3, 9,11, 9,10,11,-1,-1,-1,-1},
{ 1,10, 0, 0,10, 8,10,11, 8,-1,-1,-1,-1},
{10, 3, 1,10,11, 3,-1,-1,-1,-1,-1,-1,-1},
{ 2,11, 1, 1,11, 9,11, 8, 9,-1,-1,-1,-1},
{11, 3, 2, 0, 9, 1,-1,-1,-1,-1,-1,-1,-1},
{11, 0, 2,11, 8, 0,-1,-1,-1,-1,-1,-1,-1},
{11, 3, 2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{ 3, 8, 2, 2, 8,10, 8, 9,10,-1,-1,-1,-1},
{ 9, 2, 0, 9,10, 2,-1,-1,-1,-1,-1,-1,-1},
{ 8, 0, 3, 1,10, 2,-1,-1,-1,-1,-1,-1,-1},
{10, 2, 1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{ 8, 1, 3, 8, 9, 1,-1,-1,-1,-1,-1,-1,-1},
{ 9, 1, 0,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{ 8, 0, 3,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1}};
```

- Tables are from the Web Page:
  http://www.swin.edu.au/astronomy/pbourke/modelling/polygonize/

NOTES