

19256.

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

MEDİKAL ÜÇ BOYUTLU GÖRÜNTÜLEME VE BİR UYGULAMA

YÜKSEK LİSANS TEZİ

Müh. Derya AKAY

Tezin Enstitüye Verildiği Tarih : 12.6.1991

Tezin Savunulduğu Tarih : 30.9.1991

Tez Danışmanı : Prof.Dr.Ertuğrul YAZGAN

Diğer Jüri Üyeleri : Prof.Dr.Ergül AKÇAKAYA

Doç.Dr.Melih PAZARCI

EYLÜL | 1991

T. G.
Yükseköğretim Kurulu
Dokümantasyon Merkezi

ÖNSÖZ

Tez çalışmalarım sırasında benden hiçbir yardımını esirgemeyen ve SUN SPARC 1+ gibi mükemmel bir sistemle çalışmama olanak sağlayan sayın hocam Prof.Dr.Ertuğrul Yazgan'a, tezin hazırlanması sırasında büyük özveri ile çalışan grup arkadaşlarım yüksek mühendis Tamer Ölmez ve mühendis Mehmet Tazebay'a, medikal verilerin sağlanmasında bize sağladığı imkanlar için Siemens firmasına ve Siemens biyomedikal bölüm satış müdürü sayın Ümit Keskin'e, Sun Sparc 1+ işistasyonunun daha etkin olarak kullanılmasında bize yardım eden STFA firması yetkililerine, bize Amerikadan gönderdiği verilerle çalışmamızı destekleyen sayın Alev Erdi'ye, tez sürecinde bize yapıcı eleştirileri ile yol gösteren ve çalışmamızın daha iyiye ulaşmasını sağlayan arkadaşlarımıza ve en son olarak ta tüm eğitim hayatım boyunca bana verdikleri destek ile daha başarılı olmamda büyük rol oynayan aileme teşekkürü bir borç bilirim.

Bu tez Devlet Planlama Teşkilatının bir projesi olarak yapılmakta ve bu kuruluş tarafından desteklenmektedir.

İÇİNDEKİLER

ÖZET.....	x
SUMMARY.....	xi
BÖLÜM 1. GİRİŞ.....	1
BÖLÜM 2. TARİHÇE.....	8
BÖLÜM 3. ÜÇ BOYUTLU GÖRÜNTÜLEMENİN ÜSTÜNLÜKLERİ VE KULLANIM ALANLARI.....	12
3.1.Üç Boyutlu Görüntülemenin Kullanım Nedenleri ve Üstünlükleri.....	12
3.2.Üç Boyutlu Görüntülemenin Tıpta Uygulama Alanları.....	16
3.2.1.Teshiste Yardımcı Olarak 3B Görüntüleme.....	16
3.2.2.Cerrahi Operasyon ve Tedavi Planlanmasında 3B Görüntüleme.....	21
3.2.3.Anatomi Alanı Başlıca Olmak Üzere Tıp Eğitiminde 3B Görüntüleme.....	25
BÖLÜM 4. ÜÇ BOYUTLU GÖRÜNTÜLEMEDE KULLANILAN YÖNTEMLER.....	27
4.1.Kesit Döndürme ile Görüntüleme Yöntemi.....	28
4.2.Yüzey Görüntüleme Yöntemleri.....	30
4.2.1.Tel Kafes (Wire Frame) Yöntemi.....	31
4.2.2.Yüzey Görüntü Yöntemi.....	32
4.2.3.Hacim Görüntü Yöntemi.....	35
4.2.3.a.Işın Tarama Yöntemi.....	38
4.2.3.b.Octree Ağaç Kodlama Yöntemi.....	39
4.2.3.c.Yüzey ve Hacim Elde Etme Yöntemlerinin Karşılaştırılması...	43
4.3.Diğer Görüntüleme Yöntemleri.....	44
4.4.3B Görüntülemede Kullanılan Yardımcı Teknikler.....	47
4.4.1.Döndürme.....	47
4.4.2.Segmentasyon.....	50
4.4.3.Gölgeleştirme.....	56
4.4.3.a.Işık Modeli.....	56

4.4.3.b.Gölgelendirme Yöntemleri.....	59
4.4.3.c.Fark Gölgelendirme.....	61
4.4.3.c.1.Z Eksenli Fark Gölgelendirme...	62
4.4.3.c.2.Gri Seviye Farkı ile Gölgelendirme.....	62
4.4.3.c.3.Uyumlu Gri Seviye Farkı Gölgelendirme.....	65
4.4.3.c.4.Kayan Küpler Segmentasyon Gölgelendirme.....	65
4.4.3.c.5.Şeffaf Gri Seviye Farkı Gölgelendirme.....	66
BÖLÜM 5. TEZDE GELİŞTİRİLEN ALGORİTMALAR VE YÖNTEMLER.....	69
5.1.Tezde Kullanılan Veriler.....	69
5.1.1.Test Modeli.....	69
5.1.2.Gerçek Veriler.....	71
5.1.3.İnterpolasyon.....	72
5.1.4.3B Görüntülemeye CT ve MRI Verilerinin Karşılaştırılması.....	75
5.2.Ekranda Ardışıl Kesitlerin Yerleştirilmesi.....	77
5.3.Döndürme ve Görünmeyen Yüzeylerin Eliminasyonu.....	78
5.4.Kesme.....	84
5.4.1.Bakış Düzleminde Kesme.....	84
5.4.2.İstenilen Düzlemde Kesme.....	87
5.5.Pencereleme Ve Segmentasyon.....	93
5.5.1.Pencereleme.....	93
5.5.2.Segmentasyon.....	96
5.5.2.a.Histogram Analizi.....	96
5.5.2.b.Gri Seviye Analizi.....	98
5.5.2.c.Eşik Seviyeli Segmentasyon.....	99
5.6.Gölgelendirme.....	102
5.6.1.Işık Kaynağı.....	102
5.6.2.Yüzey Normallerinin ve Ekran Basılacak Resimin Hesaplanması.....	103
5.6.3.26 Komşuluk Yöntemi ile Yüzey Normali Hesabı.....	105
5.7.Görüntüyü Büyütme işlemi.....	110
5.8.Yüzey Elde Etme Yöntemi ile Hızlı Döndürme ve Animasyon.....	111
BÖLÜM 6. SONUÇLAR VE AÇIKLAMALARI.....	115
BÖLÜM 7. GELECEK ÇALIŞMALAR VE 3B GÖRÜNTÜLEMENİN GELECEĞİ.....	127

EKLER.....	131
EK A. Uygulama Programları.....	131
EK A.1.Geliştirilen Temel 3B Görüntüleme Programı.....	131
EK A.2.Gri Seviye Atama Programı.....	140
EK A.3.Animasyon Programı.....	141
EK A.4.Animasyon Verilerini Oluşturan Program.....	143
EK A.5.Kesit Sayısını İnterpolasyon İle Çoğaltan ve Dosyalayan Veri Oluşturma Programı.....	144
EK A.6.Yüzey Elde Etme Yöntemi İle Hızlı Döndürme Animasyon Programı.....	148
EK A.7.Yüzey Elde Etme Yöntemi İle 3B Cisimin Yüzeylerini Oluşturan Program...	153
EK A.8.Histogram, Histogram Dengeleme İşlemlerini Gerçekleyen Program.....	157
Ek A.9.Geliştirilen 3B Görüntüleme Programında İstenilen Düzlemde Kesme Yapma Ve Kesit İnceleme Programı.....	162
EK B. SUN SPARC 1+ Hakkında Bilgiler.....	166
EK C. Konu İle İlgili Tıp Terimleri.....	172
KAYNAKLAR.....	174
ÖZGEÇMİŞ.....	176

ŞEKİL LİSTESİ

3.1	Çok kesitli ve 3B görüntü yöntemlerinin göz içi yabancı yapı örneği ele alınarak karşılaştırılması.....	14
3.2	Çok kesitli ve 3B görüntü yöntemlerinin kafa içi travma örneği ele alınarak karşılaştırılması.....	14
3.3	Kalbin erken sistol evresinde 4 odacığının 3B görüntüsü.....	18
3.4	Beyin içi tümörü görüntüleyebilmek için matkap operasyonunun uygulanması.....	20
3.5	Hastanın çene kemiği anormalliğinin operasyon ile düzeltilmeden önce ekranda operasyonun planlanması.....	22
3.6	Bilgisayar destekli cerrahinin prensibi.....	23
4.1	Kesit Döndürme Yönteminin Prensip Şeması.....	29
4.2	Kesit Döndürme Yönteminin Klinik uygulaması.....	29
4.3	Üçgen Yöntemi ile yüzey oluşturma.....	31
4.4	Üçgen yüzey yöntemi ile elde edilen 3B görüntü.....	32
4.5	Yüzey elde etme yönteminin uygulaması.....	35
4.6	Ardışıl kesit görüntülerin voxel yapısı.....	37
4.7	3B görüntü işleme operatörü.....	37
4.8	Orijinal görüntülerin segmentasyon işlemi ile binary görüntülere dönüştürülmesi.....	38
4.9	Işın tarama modeli için kamera model tekniği.....	39
4.10	Işın tarama algoritmasının prensibi.....	40
4.11	Binary kesit bilgisi üzerinde ağaç kodlama tekniği.....	41
4.12	Ağaç kodlama yönteminin akış şeması.....	42

4.13	3B görüntü için kullanılacak olan inceleme hacim birimi ve bir hacim içerisinde yerleşimi.....	42
4.14	Karma görüntü yöntemi ile elde edilen beyin görüntüsü.....	45
4.15	Yeniden Görüntüleme yönteminin prensibi.....	46
4.16	P noktasının z eksenı etrafında θ açısı kadar dönerek P' noktasını oluşturmaları.....	48
4.17	Segmentasyon ile istenilen dokunun elde edilmesi.....	51
4.18	Segmentasyon için histogram grafiğinin kullanılması.....	53
4.19	Tek ve Çift veri kaynaklı segmentasyon.....	56
4.20	Işık kaynağı ve yüzey normal vektörleri.....	57
4.21	Yansıyan ışık modeli.....	58
4.22	6 komşuluklu gri seviye gölgelendirmesi işleminde komşuluk ilişkileri.....	64
4.23	6 adet değişik gölgelendirme yönteminin karşılaştırılması.....	67
5.1	Bilgisayar programı ile üretilen test modeli.....	70
5.2	İnterpolasyon algoritmasında orijinal ve üretilen kesitler arasındaki ilişki.....	72
5.3	İnterpolasyon algoritması ile kesit üretme işleminin prensibi.....	73
5.4	Cisim ile gözlemci koordinat sistemi arasındaki ilişki.....	80
5.5	Görülmeyen yüzeylerin z' doğrultusu boyunca eliminasyonu.....	83
5.6	Bakış düzleminde kesme işlemi.....	85
5.7	Döndürme ve Kesme işleminin akış diagramı.....	86
5.8	Kesme düzleminin üstünde kalan parçanın eliminasyonu.....	91

5.9 Gri seviyelerinin tek pencereleme yöntemi.....	94
5.10 Gri seviyelerinin çift pencereleme yöntemi.....	95
5.11 Histogram analiz programının akış şeması.....	97
5.12 Eşik seviyeli segmentasyon işleminin akış şeması.....	101
5.13 26 komşuluk yöntemi için kullanılan komşuluk ilişkileri.....	106
5.14 Gölgeleştirme işleminin akış şeması.....	109
5.15 Görüntüyü büyütme işlemi.....	110
5.16 Yüzey elde etme yöntemi için kenar dedeksiyonu akış şeması.....	113
6.1 Ek A.1'deki programın sonucu. $\theta=180$, $\phi=90$	118
6.2 Ek A.1'deki programın sonucu. $\theta=180$, $\phi=135$	119
6.3 Ek A.1'deki programın sonucu. $\theta=40$, $\phi=0$	120
6.4 Animasyon Görüntüleri.....	121-123
6.5 Ek A.1'deki programın sonucu.....	124
6.6 Ek A.9'daki programda kesilen yüzeye ait 2B görüntü.....	125
6.7 Ek A.9'daki programın sonucu elde edilen istenilen düzlemde kesme yapma işlemi.....	126

TABLO LİSTESİ

4.1	3B görüntü yöntemlerinin sınıflanması.....	28
4.2	Önemli 3B yüzey görüntüleme yöntemleri.....	30
5.1	CT zayıflama değerlerinin dağılımı.....	93
5.2	2B görüntü analizi ile elde edilen ve segmentasyon için kullanılan eşik değerleri....	100



ÖZET

Bu teze konu olan üç boyutlu (3B) görüntüleme birçok mühendislik alanında uygulama bulan ve güncel olarak üzerinde araştırmalar yapılan aktif bir konudur. Bu tezde üzerinde çalışılan ; üç boyutlu görüntülemenin tıbbi olan katkısı, doktorlara ve uzmanlara gerek teşhiste gerekse tedavide sağladığı kolaylık ve doğruluktur. Giderek yaygınlaşan ve hastanelerdeki Bilgisayarlı Tomografi (Computer Tomography - CT), Magnetik Rezonans Tomografisi (Magnetic Rezonans Imaging -MRI) gibi medikal görüntüleme cihazlarında kullanılan üç boyutlu görüntüleme programları her zaman daha fazla imkanlar sunarak doktorların ve tıbbın hizmetinde bulunmaktadır.

Çalışmalarımızda CT'den alınan 2 boyutlu kesitler tarafımızdan geliştirilmiş olan bir algoritma ile birleştirilmiş ve hafızada üç boyutlu bir resim yaratılarak bunun ekranda gölgelendirme ile üç boyutlu gözükmesi sağlanmıştır. Algoritmalarımız Sun Sparc 1+ iş istasyonunda (Workstation) çalıştırılmış ve yeterli hıza ulaşılmıştır. Hızla ilgili kriterler tezde açıklanacaktır. Ayrıca segmentasyon, kesme, döndürme, histogram analizi, animasyon gibi görüntü işleme programlarında gerçekleştirilmiştir. Programlarımızda farklı gölgelendirme, döndürme ve segmentasyon methodları denenmiş ; en iyileri tesbit edilmiştir.

Bu çalışmada 1. bölümde 3B görüntülemeye genel bir giriş yapılarak problemler tanıtılmış ve tezde yapılanlardan kısaca bahsedilmiştir. 2. bölümde, üç boyutlu görüntülemenin tarihçesi anlatılmıştır. 3. bölümde ise üç boyutlu görüntülemenin yararları, kullanım alanları açıklanmıştır. 3 boyutlu görüntülemede kullanılan metodlar bölüm 4'de açıklanmıştır. Bölüm 5'de ise tezde kullanılan matematiksel yöntemler ve programların mantığı açıklanmıştır. Bölüm 6'de ise tezdeki programlardan elde edilen sonuçlar verilmiş ve sonuçlar açıklanmıştır. Programın geliştirilmesi için olan önerileri ve üç boyutlu görüntülemenin geleceği bölüm 7'dedir. Ekler ise üç bölüm halindedir. Yazılan programlar ise Ek A'da sunulmuştur. Ek B'de Sun Sparc 1+ hakkında teknik bilgi verilmiştir. Ek C'de tezde geçen yabancı terimler açıklamaları ile verilmiştir.

MEDICAL THREE DIMENSIONAL IMAGING

AND AN APPLICATION

Three Dimensional (3D) Imaging is an update subject in medicine because of it's diagnostic, therapueutic, teaching and research values. Research for 3D Imaging are being done to obtain better resolution and more speed for image processing while 3D Imaging programs are being used in hospitals and medical centers. Utility of the 3D imaging is incresing daily with the new developments in Computer Graphics and Fast Computers like workstations and several different images . Utilities can be presented to the specialists and clinicians for treatment and diagnosis by using digital image processing techniques as segmentationd, rotation around axis, clipping, artifical colors for different tissues, low pass filters, huge storing capabilities and etc.

A well trained radiologist is able to mentally visualize an entire 3D image of the object from a series of cross sections of a target volume. In actual practice, it is often diffucult to accurately visualize the 3D surface of the object, especially when the configuration of the structure is complex. 3D processing of medical data is a diagnostic technique for clearfying the 3D relationships of complex lesions and anatomical structures.

Computer Graphics allows this to be obtained as digitized information and has made it possible the full scale of pseudo 3D displays in the medical field. Clinical applications for 3D displays have been developed mainly in the following areas : Diagnostic support, simulations for surgical operations and therapy plannig. By using the 3D displays it is possible to diagnose morphological abnormalities which impossible to do accurately from individual slices of images, as well as to confirm 3D positional relationship between organs. The second area has an importance for surgical operations and therapy planning. During brain surgery this technique allows surgeons before surgical operations to understand the true location of the lesions. These utilities make the surgical operations safer than they were and this point of view, it is very easy to understand that why we need high-speed processing of the 3D images. Orthopedists and plastic surgeons also use CT images to construct 3D images of bones and soft tissues, especially in the face.

In the therapy planning it is very important to define the true location of lesion which will be applied radiation. Another important point is, to calculate the radiation dose distribution within the irradiated volume. 3D displays allow this activities during the radiation treatment. In the area of medical education, 3D images simplify the difficult anatomical explanations in the textbooks by converting explanatory diagrams into computer images.

To obtain 3D images there are two main methods : volume rendering and surface rendering. In volume rendering methods we have to calculate volumetric data and use it to show the images. In the other method we calculate only the surfaces of the objects from the individual slices. In volume rendering methods, more data have to be used to obtain the result and the process time is more than surface rendering techniques. Regardless of these problems, volume rendering methods allow us to see the all volume in clipping and segmentation process because it contains all volume data which belongs to 3D object. Even surface rendering methods are fast and need less data, they don't contain volume information.

In our programs we had to obtain segmented and clipped objects so we had to use volume rendering methods. In our programs, we only used surface information to provide fast rotating because of the short processing time of this kind of images. During our studies we realized the difference between these two methods. Even we can't clip or segment the object which is obtained by using surface rendering methods, the rotating rate is at least 10 times more than the object's rotating rate which is obtained by using volumetric data.

At the beginning of our studies we used some animation data that we created them in the computer by using programs. After we wrote our programs and tested them by using this animation data, we started to use real medical data. Our first medical data aren't enough to obtain a real medical data (25 slices), so we increased the number of the slices by using an interpolation algorithm (168 slice).

Images from the CT and MRI computers have got gray levels and we have to use this gray level to obtain our objects. In CT images gray levels represent the attenuation coefficient of the x-rays which are passing the

body and in an original image there are 4096 gray level so because of the screen limits and anatomical structure of the human eyes (human eye can't differ the difference between more than 20 gray level) we have to create a window in the gray level spread. Our window contains 256 gray level to obtain more detail and accurate results. This window can be adapted any part of 4096 gray level spread and allow us to see the information in the window.

Our screen is two dimensional so if we want to create 3D images in the screen we have to use some computer graphic algorithms like shading and hidden surfaces. We used an algorithm for shading which is called 26 neighbourhood gray level gradient shading algorithm. We used some different shading algorithms to observe the difference between the results and we decided that 26 neighbourhood gray level gradient shading was the most appropriate. This algorithm is a version of the Phong Shading Algorithm.

Two dimensional slices are located in the memory to create a 3D array. Our 3D object is located in a constant x,y,z coordinate system, when our observer coordinate system x',y',z' is rotating around the object. We preferred this rotating system because we think that rotating time would be less. The direction of the light, which is used for shading, is given on the x,y,z coordinate system. Direction of the light can be changed or adjusted automatically.

Segmentation, is a very important aspect of the 3D imaging study. If the segmentation program is good enough and the images contain the necessary data ; you can observe different tissues from images. For example it is very difficult to observe vessel information from a CT image, but it is quite easy from a MRI or an Angiography image. By using some complex algorithms, it is possible to find the cortex and ventricular structure of the brain. Gray level windowing is a kind of segmentation but it is not enough to catch desired tissues. With segmentation, all skin information can be eliminated in order to see the bone structure of the head. Another application would be to eliminate the bone and see the brain only. At this point of study, we can clip the image and see the inside of the brain. This kind of utilities are very useful for brain surgeons because they can find the true location of the foreign body and they provide accurate information for the operation.

Writing our programs, a lot of problems occurred and we tried to solve them by using 2D image processing techniques. Some of the algorithms are listed below:

Gray Point Analysis: To find the gray level of the image by using a keyboard controlled point. It travels on the image and says the gray level value where it is.

Histogram Analysis: To find the spread of the gray levels in the image. It is easy to make segmentation by using histogram analysis. Especially if we can separate the major gray levels in the images and after that it is easy to find the tissues that create these gray levels.

Histogram Equalization: When we have got a dark image we equalized the histogram to obtain a better image.

The most useful 2D image processing techniques are gray point analysis and histogram analysis and we used them to improve our 3D images successfully. The result of Segmentation process is especially perfect by using gray point analysis. Because by using this method it is easy to find the gray value of desired tissue or organ and after that segmentation process is easy to make.

One of the most important point in 3D imaging is to obtain fast images on the screen. Our system is a SUN SPARC 1+ workstation (32 MB for RAM and 1 GB harddisk, 16 mips) giving enough memory and speed for our studies. Our workstation contains following parts : a RISC architecture with 32 MB Ram, 1 GB Harddisk, a magnetic tape driver and a high resolution screen (1152 x 900). Sun Sparc 1+ system uses Sun Os 4.0 operating system which is a Unix version for Sun family. There is C compiler available in our system so we used C language to develop our programs.

In future 3D imaging programs will be more speedy will have better resolution. We wrote a high-speed 3D imaging program by using a workstation. This kind of workstations will be more common in future medical imaging techniques because their speed and high data storing capabilities so our programs will be a step for research in this subject. Our programs may be developed using a better shading algorithm and a faster rotating algorithm for volume data. Presence of a vector processor in the system will increase the speed considerably and operation speed will be very close to the commercial 3D programs. Image quality is also important for 3D images and this specification can be realized by using three dimensional workstations. In general while data processing time is increasing the image quality is decreased but the doctors and specialists want to have both of them.

At the beginning chapter of this study an overview of the 3D imaging is given. In the second chapter, the history of the 3D imaging is written. In the third chapter, the utilities of the 3D imaging and the interested areas are being explained. Three dimensional methods are explained in the chapter four. Our methods and formulas with our three dimensional rendering techniques are in the fifth chapter. In the sixth chapter, all results of our study are given and explanied. There are future adviceses for our 3D programs and an overview for the future of the 3D imaging in chapter seven. In the additional part A, you can find more technical information about Sun Sparc 1+ and some utilities for compailing C programs, writing in a vi editor. In additional part B, some of the terms in the thesis are explained. In part C, our programs are given with their explanations.



BÖLÜM 1

GİRİŞ

Üç boyutlu görüntüleme tıp alanına getirdiği yenilikler sayesinde değişik uygulama alanları bulmuş ve günümüzde üzerinde çalışmaların devam ettiği aktif bir konudur. Bilgisayarlı Tomografi , Nükleer Magnetik Rezonans gibi tekniklerde elde edilen görüntüler iki boyutludur ve bu görüntülerin teşhis amaçlı kullanılması esnasında uzmanlar ve doktorlar iki boyutlu görüntüleri beyinlerinde birleştirerek, kendilerine hayali bir üç boyut oluşturmak zorunda kalmaktadırlar. İncelenen bölgenin karmaşıklığında göz önüne alınırsa, bu işlemin nasıl zor olduğu ortaya çıkar. Bu açıdan bakıldığında Üç Boyutlu (3B) görüntüleme ihtiyaç duyulan görüntüleri hazır olarak sunmaktadır.

Üç boyutlu görüntüleme tıpta genel olarak üç temel alanda kullanılmaktadır: Teşhise yardım amacıyla, cerrahi müdahalelerin ve radyasyon tedavi işlemlerinin planlanmasında, tıp eğitiminde başlıca anatomi alanında kullanılmaktadır.

Tedavi öncesinde, klasik teşhis yöntemlerine ek olarak kullanılabilen 3B görüntüleme operasyonlardan önce

lezyonların yerinin tam tesbiti ve durumun operasyon sırasında izlenebilmesi için gerekli imkanları sunmaktadır. Bu aşamada hızlı 3B görüntüleme metodlarına olan gereksinimin nedeni ortaya çıkmış olmaktadır.

Radyoloji uzmanları , 3B görüntüleme uygulamasını radyasyon tedavisi sırasında gerekli dozun ve ışın alanının ölçülüp , tam olarak belirlenmesinde kullanılmaktadır.

Tıp öğrencilerinin eğitiminde karmaşık yapıların 3B'lu olarak gerçek dünyaya uygun olarak sunulmasında büyük önem taşımakta; eğitimi kolaylaştırıp kaliteyi artırmaktadır. Kısacası 3B görüntüleme insanın gerçek dünyasına daha uygundur ve dolayısı ile kolay anlaşılmaktadır.

Bu tezde amaç; bir üç boyutlu görüntüleme programı yazmaktır. Programlarımızda kullandığımız 3B görüntülemenin temel algoritmaları olan döndürme, kesme, gölgelendirme ve segmentasyon algoritmaları çeşitli kitap ve bilimsel makalelerden elde edilen fikirlerle kendimiz tarafından geliştirilmiştir. Tezdeki amaçlarımızdan biri her zaman görüntü kalitesini artırmak olduğundan, çeşitli 2B görüntü işleme metodları kullanılmıştır. Kesme, segmentasyon işlemlerini gerçekleştirebilmemiz için gerekli olan yöntem olarak hacim elde etme yöntemi tercih edilmiştir. Bu şekilde fazla veri işlenerek zaman kaybı olduğu halde,

kesme işlemine olanak sağlamaktadır. Programlarımızdaki bir başka amacımız ise veri işleme hızının artırılmasıdır. Veri işleme hızları gerek programlardaki etkinlik ve gerekse Sun sparc 1+'ın yüksek performansı ile tatmin edici düzeye getirilmiştir.(hacim verilerinden oluşan görüntülerin kesme, döndürme, segmentasyon ve gölgelendirme işlemleri için 3.5 dakika, yüzey verileri için aynı işlemler 10 saniye). Hız ile ilgili kriterler bölüm-6'da sonuçlarla beraber ayrıntılı olarak verilecektir.

Bir üç boyutlu görüntüleme programından; cisimlerin istenilen eksen etrafında döndürülmesi, istenilen yerden kesme yapılması, doku ayırma metodu (segmentasyon) ile istenilen yerlerin görüntülenmesi, hacim ölçümü yapılabilmesi, hızlı olması gibi özellikler istenir. Bunun yanı sıra birçok özellik daha uzmanlara programlar tarafından sağlanmaktadır. Yaptığımız program temel özelliklerin hepsini sağlamakta ve dijital görüntü işleme metodları kullanarak bazı analiz imkanları sunmaktadır.

Programın ilk yazım ve test aşamalarında, bilgisayar programı ile üretilen simülasyon verileri kullanılmıştır. Bu verilerle test edilen programlarda sonradan medikal veriler kullanılmış ve gerçek sonuçlar elde edilmiştir. Gerçek medikal verilerin Türkiyeden elde edilemedikinden dolayı zorluklar nedeniyle bu veriler Amerika'dan ve

Almanya'dan getirtilmiştir. Kullanılan medikal verilerin kesit sayısı olarak azlığından dolayı, bir interpolasyon algoritması ile kesit üretilerek bu sayı çoğaltılmıştır. Kullanılan veriler 512*512 rezolüsyonlu olup, programlarımızda hafıza sorunu yüzünden 256*256 olarak kullanılmıştır.

Bilgisayarlı Tomografi (Computer Tomography - CT) ve Magnetik Rezonans Görüntüleme (Magnetic Rezonans Imaging -MRI) cihazlarında elde edilen görüntülerde gri seviyesi esastır ve bu seviye CT'de x ışınlarının zayıflama katsayısını, MRI'da ise hidrojen atomlarına ait t_1 ve t_2 sürelerini temsil eder. Bu gri seviye aralığı orijinal görüntülerde 4096 dır ama gerek gözün fizyolojik sınırları, gerekse ekranın teknik sınırları (Sun Sparc 1 ekranı aynı anda 256 gri seviye gösterebilir.) yüzünden 256 gri seviyeli bir pencere ile sınırlanarak kullanılır. Bu aralığın değişik şekillerde kullanılması ile, görüntülerde daha iyi bir çözünürlüğe ulaşılmaktadır.

Programlarda iki boyutlu görüntüler ele alınmakta ve bunlar en başta algoritma yardımı ile RAM'da bir üç boyutlu dizi olarak tanımlanmaktadır. 3B'lu cisim için seçilen uzay koordinatları x , y , z 'dir. Gözlemci uzayımız ise x' , y' , z' olarak tanımlanmıştır. Dönme işleminde ise cisim uzayı değil, gözlemci uzayı dönmekte ve böylece

zamandan ve hızdan kazanmış olmaktadır. Dönme sırasında en önemli olaylardan biriside görünmeyen yüzeylerin basılmamasıdır. Buda program içinde bazı test koşulları yazılarak sağlanmıştır.

Programda kullanılan doku ayırma (segmentasyon) ve kesme (clipping) algoritmaları çeşitli metodlar karşılaştırarak bulunmuştur. Segmentasyon işlemi 3B görüntülenmenin en önemli bölümlerinden biridir. Bu işlem sayesinde istenilen dokular (beyin , kafa kemikleri , deri , cortex) ayırd edilebilmekte ve ekranda görüntülenmektedir. Bu işlem tüm dokuların gri seviyelerinin farklı olması esasına dayanır. Eğer elimizde bulunan verilerde bu dokuların bilgiside saklı ise elde etmek kolaydır. Bu dokuların bilgisini görüntü içinden yazdığımız gri nokta analizi ve histogram analizi gibi programlarla elde etmek mümkündür. Örneğin CT görüntülerinden beyin damarlarının görüntüsünü yakalamak hemen hemen imkansızdır çünkü bu bilgi ya yoktur yada çok az ve yetersizdir. Oysa MRI veya Anjiografi görüntüleründen bu bilgi kolayca bulunur. Kesme işlemi ise uzmana istediği yerden, istediği 3B'lu görüntüyü kesit olarak inceleme imkanı tanır. Bu işlem yardımı ile beynin tümü, cerrahlar tarafından incelenerek ameliyat öncesi lezyonların tam olarak nerede oldukları bulunur. Programımızda segmentasyon işlemleri çeşitli metodlarla gerçekleştirilmiştir.

3B görüntülemeyi 2 boyutlu bir ekranda yapmak için gerekli olan en önemli algoritmalar; gölgelendirme ve gözükmeyen arka yüzeylerin ekrana basılmamasını sağlayan algoritmalarıdır. Programımızda, gölgelendirme algoritması olarak 26 komşuluklu gri seviye farkı algoritması kullanılmış ve olumlu sonuç alınmıştır. Bu algoritma Phong gölgelendirme algoritmasının bir uygulama şeklidir.

3B görüntüleme yaparken 2 boyutlu kesitler üzerinde yapılan ön işlemlerin önemide büyüktür. Bu ön işlemleri yapan gri seviye analizi, histogram analizi ve kayan gri seviye penceresi gibi programlar tezde açıklanmıştır. Bu işlemler sayesinde daha iyi doku ayırma işlemleri yapmak mümkün olmuştur.

Programımızda 3B görüntü oluşturmak için hacim verileri hesaplanmıştır. Bir başka yöntemde yüzey elde etme işlemlerini kullanmaktadır. Hacim verileri kullanınca fazla işlem yapıldığından zaman uzamaktadır ama kesme işlemi yapıldığında hacim bilgisi olduğundan dokuları görmek mümkündür. Oysa yüzey bilgisinin kullanıldığı görüntülerde kesme yapmak mümkün değildir çünkü hacim bilgisi yoktur fakat döndürme işlemlerinde az veri işlendiğinden hız fazladır ve bu yöntem animasyonların hazırlanmasında kullanılmaktadır.

Programlar kullanılan sistemin gelişmesi ile daha yüksek hızlara ulaşacaktır. Örneğin eklenecek bir vektör işlemcisi ile sonuçların elde edilmesi daha hızlı olacaktır. Programın geliştirilmesi için olan tavsiyeler ve 3B görüntülemenin geleceği tezin son bölümünde açıklanmıştır.



BÖLÜM 2

TARİHÇE

Bu bölümde ; üç boyutlu (3B) görüntülemenin kısaca tarih içindeki gelişiminden bahsedilerek kullanıcısı olan uzman ve doktorların düşüncelerinede yer verilmiştir. [1]

3B görüntüleme çalışmaları 1970'li yıllarda başladı ve bilgisayar teknolojisindeki gelişmelerle paralel olarak ilerledi. 1980'li yılların başlarında 3B görüntüleme tıp dünyasına tanıtıldığı zaman; fazla yer kaplayan hacimsal verilerin yorumlanması ve gösterilmesi işlemini sağlayacak basitleştirilmiş bir metod olarak görüldü. Bu yeni tekniğin oluşmasını sağladığı 3. boyut ile teşhis işlemlerinin daha güvenilir olması ve büyük hacimsal yapıların gösterilmesinde süreklilik sağlanmasına rağmen 3B görüntüleme o günlerde teknolojik, klinik ve ekonomik açıdan bir karmaşa içindeydi ve bu yüzden kabul görmesi oldukça zordu. Bu karmaşa yüzünden zaten bu konuya yabancı olup isteksiz davranan uzmanların 3B görüntülemeyi kendi uygulamalarında kullanmaları beklenemezdi. Öte yandan 3B görüntülemenin maliyeti, sonuçta sağlayacağı yararları karşılamayacak durumdaydı.

Teknolojik gelişmelerin bir sonucu olarak ilerleyen bilgisayar teknolojisi de 3B görüntülemenin maliyetlerini düşürmüştür. Fakat görüntü işleme teknikleri kullanılarak dahada geliştirilen 3B görüntüleme işlemi radyolojistler tarafından teşhiste yetersiz olarak görülmekteydi. Bu konuda uzmanlar haklıydı çünkü 3B görüntüleme işlemi bazı çok zor teşhis problemlerinde ve bazı özel durumlar dışında kullanılmıyordu. 3B görüntülemenin kullanımının yaygınlaşmasını teşhiste sağladığı üstünlükler değil; tedavi ve görüntüleme arasında bir köprü vazifesi işlemini görmesi sağlamıştır.

Üç boyutlu işistasyonları 1983 yılında ticari olarak üretilmeye başlandı. İlk üretilen modeller pahalı ve yavaş olmalarının yanı sıra , klinik alanda önemli bir gelişmeye neden olmadılar. Bu sırada 3B görüntüleme üzerinde, araştırmalar ve fazla ilgi toplamamasına rağmen ticari uygulamalar için çalışmalar devam etti. Bu alanda en büyük sorunlardan birisi 3B görüntüleme hesaplarının yapılmasını sağlayacak teknolojinin çok pahalı olmasıydı. Uzmanlara göre hem yazılımda hemde donanımda maliyetin kaliteyi azaltmadan düşürülmesi gerekiyordu.

Görüntü kalitesinin ve işleme zamanının 1980 lerin ortalarında istenilen düzeye gelmesine rağmen maliyetlerin azaltılamaması yüzünden 3B görüntüleme klinik

uygulama alanında kullanılamıyordu. Bilgisayar endüstrisindeki gelişmelere bağlı olarak ilerleyen çalışmalarda, asıl kullanıcı olan uzman ve radyolojistlerin ihtiyaçları pek dikkate alınmıyordu.

1980'lerin sonlarına doğru maliyet en büyük sorun olmaya başladı. Bu aşamada işistasyonu üreten firmalar 3B görüntüleme ile ciddi olarak ilgilenmeye başladılar ve araştırma geliştirme çalışmaları sonucu 3B sistem fiyatları 120.000 \$ ile 150.000 \$ 'a kadar indirildi. Üreticiler kullanıcıların isteklerine daha fazla önem vermeye başladılar ve uzman doktorların istekleri göz önüne alınarak üretim yapılmaya başlandı.

1980'lerin sonlarında; üreticilerin bir başka sorunu ise klinik uygulamalarda kullanılacak olan 3B görüntüleme sistemlerinin karmaşıklığı idi. Çünkü bu sistemlerin sadece CT ve MRI görüntülerinden sonuç oluşturmamasından başka, tarayıcı (scanner), bilgi ağı (network) ve kameralar gibi sistemleride içermesi gerekiyordu. Bu açıdan bakıldığında üretici firmaların ürünleri, kullanıcıların isteklerinden farklıydı ve bu sorunda konuda bir boşluk yaratıyordu. Giderek geliştirilen 3B görüntüleme CT, MRI, PET (positive elektron tomography), ultrasonografi ve nükleer tıp verilerini kullanmaya başlamıştı. 3B görüntüleme sistemleri üreti-

cilerine düşen en büyük görev klinik sorunların çözülmesi idi çünkü uzmanlar teknolojik ilerlemelerin geldiği noktanın klinik uygulamaya yeterince aktarılmadığı kanaatindeydi. Bu sorun uzman doktorlar ve üretici firma mühendislerinin ortak çalışması ile çözülmüştür.

3B görüntüleme günümüzde klinik uygulamalarda halen belirli alanlarda kullanılmaktadır. Kullanılan sistemlerin maliyeti her zaman daha da aşağıya çekilmeye ve görüntü elde etme hızının giderek artırılmasına çalışılmaktadır. 3B görüntülemenin kullanım alanları, tıp alanına getirdiği yenilikler ve üstünlükleri bölüm 3'de açıklanacaktır.

BÖLÜM 3
ÜÇ BOYUTLU GÖRÜNTÜLEMENİN ÜSTÜNLÜKLERİ
VE
KULLANIM ALANLARI

Bu bölümde 3B görüntülemenin iki boyutlu görüntüleme tekniklerine olan üstünlüklerinden ve tıpta uygulama bulduğu alanlardan bahsedilmiştir. Gerektiği yerlerde 3B görüntülemenin aksaklıklarında açıklanmıştır.

3.1. ÜÇ BOYUTLU GÖRÜNTÜLEMENİN KULLANIM NEDENLERİ
VE ÜSTÜNLÜKLERİ

Klinik uygulamalarda alışlagelmiş görüntüleme teknikleri olan CT, MRI, ultrasonografi, nükleer tıp ile elde edilen görüntüler iki boyutludur. Radyolojistler için teşhiste çok kullanışlı olan bu görüntüler, tedavi planlamasında istenilen yararı vermemektedir. Uzmanlar iki boyutlu görüntülerde bir yabancı yapı (foreign body) teşhis ettikleri zaman bu yapının yayıldığı alanı, şeklini tespit etmeleri için birçok kesiti incelemeleri ve kendilerine 3 boyutlu hayali bir uzay oluşturarak düşünmek zorundadırlar. İncelenen anatomik yapının karmaşıklığında göz önüne alınırsa, bu işlemin ne kadar zor olduğu ortaya

çıkar. Oysa incelenmek istenen bölgelerin 3B görüntülerini uzmanlara hazır olarak sunan bu yeni teknik teşhis ve tedavi planlamasında büyük yararlar sağlamaktadır. Görüntü işleme tekniklerinde olan yeniliklerin 3B görüntülemeye devamlı olarak katılması ile uzmanlara çok çeşitli görüntü ve analiz imkanları sağlanmıştır. Şekil 3.1 üç boyutlu görüntü kullanarak teşhis yapmanın kolaylığını daha iyi açıklamak için kullanılmıştır. Burada kullanılan dört adet iki boyutlu görüntüde yabancı yapı işaretlenmiştir. Yabancı yapının yerinin daha iyi anlaşılmasını sağlamak için uzmanlar bu kesitleri ve hatta daha fazlasını incelemek ; sonuçları kendi oluşturdukları üç boyutlu uzayda düşünmek zorundadırlar. Oysa şekil 3.1.b'deki 3B görüntüde yabancı yapının yeri oldukça açık olarak görülmektedir. İnsanın düşünce yapısına daha yakın olduğu için 3B görüntüleme kolay anlaşılmaktadır. Şekil 3.2'de bu üstünlüğü daha iyi ortaya çıkarmak için verilmiştir. Şekil 3.2.a'daki iki boyutlu görüntülerden yabancı yapının yerini anlamak çok zordur oysa şekil 3.2.b'deki 3B görüntü ile bu teşhis zor değildir. Gerçi 3B görüntü elde etmek için kullanılan kesit sayısı teşhis için kullanılacak iki boyutlu kesit sayısından çok fazladır ama tüm kesitlerin aynı anda üç boyutlu olarak incelemeye sunulması ile bu teşhis zamanında kısalmaktadır. [2]



Şekil 3.1 Göz içi yabancı yapı. a) Çok kesitli görüntüleme
b) 3B görüntüleme. [2]



Şekil 3.2 Kafa içi travma sonrası. a) Çok kesitli
görüntüleme. b) 3B görüntüleme. [2]

3B görüntülemenin tıp alanına getirdiği en büyük yeniliklerden biriside görüntüleme ve tedavi alanları arasındaki boşluğu doldurmasıdır. Bölümün başında belirtildiği gibi iki boyutlu (2B) görüntüleme teşhiste kolaylık sağlamasına rağmen tedavi planlamasında yeterli olmamaktadır. Uzmanlar, yaklaşık 2 milyon dolar civarındaki CT, MRI gibi cihazlardan elde edilen ve gigabyte boyutlarına varan 2B görüntülerden tedavi planlamasında yararlanamamaktan şikayetçi idiler. Tedavi planlamasında sağladığı kolaylık ile 3B görüntüleme klinik alanda uygulama bulmuştur.

3B görüntüleme, anatomik yapının karmaşık olduğu omurilik ve bazı travma vakalarında, çok ayrıntılı görüntü bilgisinin gerekli olduğu durumlarda (beyin cerrahi) yararlı olmaktadır. Bu yararlarına rağmen tıp alanında belirli uygulama alanlarının dışında kullanılmamaktadır. Kullanıcıların ideal 3B görüntülemeyi istedikleri, insan vücudunun istenilen bir bölümünün sanki gerçekten ellerinde tutuyorlarmışçasına inceleme imkanı sunmasıdır. Yüksek bilgi işleme hızı ve çok iyi görüntü işleme teknikleri gerektiren bu istekleri sağlamak için araştırmalar devam etmektedir.

3.2 ÜÇ BOYUTLU GÖRÜNTÜLEMENİN TIPTA UYGULAMA ALANLARI

Bu bölümde 3B görüntülemenin tıptaki uygulama alanları ve bazı özel uygulama alanlarında üstünlükleri açıklanacaktır.

Tıp dünyasına katıldığı ilk zamanlarda 3B görüntüleme teknikleri sadece belirli uygulamalarda kullanılıyordu. Buna en büyük etken görüntü işleme tekniklerinin yetersizliği idi. Bu yetersizlik sonuçta hem 3B görüntü tekniklerinin kullanılabildiği veri elde etme tekniklerini (CT, MRI gibi) kısıtlıyor, hemde eldeki verilerin verimli olarak işlenmesine olanak sağlamıyordu. Gelişen görüntü işleme teknikleri ile bu sakıncalar ortadan kalkmış ve kullanım alanı giderek yaygınlaşmaya başlamıştır. Burada kullanım alanları 3 temel bölüme ayrılarak incelenecektir : Teşhiste yardımcı olarak, Cerrahi operasyonların ve hasta tedavi işlemlerinin planlanmasında, Anatomi alanı başlıca olmak üzere tıp eğitiminde.

3.2.1 Teşhiste Yardımcı Olarak 3B Görüntüleme

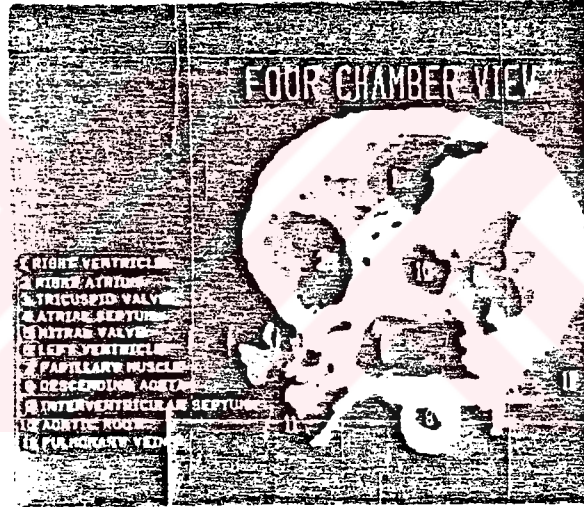
Radyolojik Teşhiste : 3B görüntüleme tekniklerinin kullanılmaya başlaması ile klasik radyoloji teşhis olumlu değişiklikler olmuştur. Uzmanlar ardışıl 2B

görüntüleri inceleyip, birleştirmek zorunda kalmadan gerçek anatomik yapıyı görebilmektedirler. Bu alanda 3B görüntüleme yeni bir konu olduğu için eldeki görüntülerin yetersizliği giderilmeye çalışılmaktadır. [3]

Kardiovasküler Dijital Anjiyografide : Dijital anjiyografi tekniği ile elde edilen bilgiler Konjetinal kalp rahatsızlıkları başta olmak üzere kronik rahatsızlıkların teşhisinde kullanılır. Dijital anjiyografi teknikleri ile kolayca ortaya çıkan damar bilgilerinin 3B görüntülenmesi damar tıkanıklarının teşhisi bakımından büyük kolaylık sağlamaktadır. [3]

Kalp Hastalıklarının Ekg Tetiklemeli MRI görüntüleri Kullanılarak Teşhisi : Karmaşık kardiyovasküler rahatsızlıkların teşhisine yeni bir boyut getiren bu teknikte EKG işaretinin R dalgası tetikleme olarak kullanılmaktadır. Bu dalgayı başlangıcı olarak alan MRI görüntüleri ile kalbin alt ucundan (apex), aort kavisine kadar 2B görüntüler elde edilmiştir. Burada nefes alma ve örtüşme hatalarını önlemek için axial düzlemde tarama yapılmıştır. Apex-aort kavisine arası yetişkenlerde yaklaşık olarak 14 cm kadardır ve bu alandan 3B görüntüleme oluşturmak için yaklaşık 28 adet kesit almak yeterlidir. Bu teknik sayesinde kalp damarlarının, kalp odacıklarının (karıncıklar ve kulakçıklar) ve kalp kapakçıklarının ayrıntılı olarak

incelenmesine olanak sağlar (Şekil 3.3). Bu yöntemin kalbi incelemek için kullanılan radyografi tekniklerine olan üstünlüğü x-ışınları kullanmaması ve aort gibi büyük damarlar ve miyokardial damarların, kalp odacıkları ile aynı zamanda görüntülenmesine olanak sağlamasıdır. 2B echo yöntemlerine olan üstünlüğü ise; büyük damarların gerçeğe yakın olarak görüntülenmesi ve kemik, hava gibi yapıların görüntüye getirdiği bozuklukların olmamasıdır. [4]



Şekil 3.3 Kalbin erken sistol evresinde 4 odacığın 3B görüntüsü.

Sperm Hücrelerinin Hareket Analizinde : Dijital anjiyografi için geliştirilen metodlarla elde edilen ve spermlerin hareket ve hızları hakkında bilgi elde etmemizi sağlayan bir tekniktir. Hız ve hareket bilgileri sayesinde insanın spermatozoa aktivitesini incelemek imkanı doğar. İncelenen

grafiklerin renklendirilmesi ile spermlerin aktivasyonlarına göre kolay incelenmesi sağlanır. [3]

Synchrotron Radyasyonu ile Angiografide : Koroner arterlerin vücut dışından (noninvasive) görüntülenmesi halen çözülmeye çalışılan bir sorundur. Bu aşamada dijital angiografi teknikleri kalbin hareketleri yüzünden etkisiz kalmaktadır. Yanlızca damarlara iyot verilerek ve sonra neredeyse eşzamanlı, monokromatik x ışınlarının kalbe uygulanması ile elde edilen görüntüler bu işlemde kullanılabilir. Elde edilen görüntüler kalbe verilen iyodun dağılımıdır. Yüksek yoğunluklu iyot ve x ışınları kullanıldığı zaman sonuçlar iyi olmaktadır. Buradaki 3B görüntüleme tekniklerinde bu iyot dağılımı kullanılmaya çalışılmaktadır. [3]

Nöroradyolojide : 3B görüntüleme tekniği ile kemik eşiği kullanılarak yapılan kemik dokunun eliminasyonu sayesinde yumuşak dokuların teşhisinde kolaylıklar sağlanmıştır. Normalde CT değeri 70 HU olan kemik değerleri segmentasyon ile ayırt edilmekte ve kafaiçi tümörleri, gözçukurundaki yabancı yapılar (foreign body), kafaiçi ventriküler bölge gibi yapılar ortaya çıkarılmaktadır. Şekil 3.4'de bir kafaiçi tümörünün, kafatası kemik yapısının segmentasyonu ile ortaya çıkarılışı görülmektedir [2]. 3B görüntüleme ile cerrahlar görsel olarak beyinde tümör, edema ve normal



Şekil 3.4 a) Beyin içi tümöre uygulanacak matkap operasyonu için segmentasyon yerinin belirlenmesi. b) matkap operasyonu gerçekleştirildikten sonra tümörün kafaiçindeki görüntüsü.

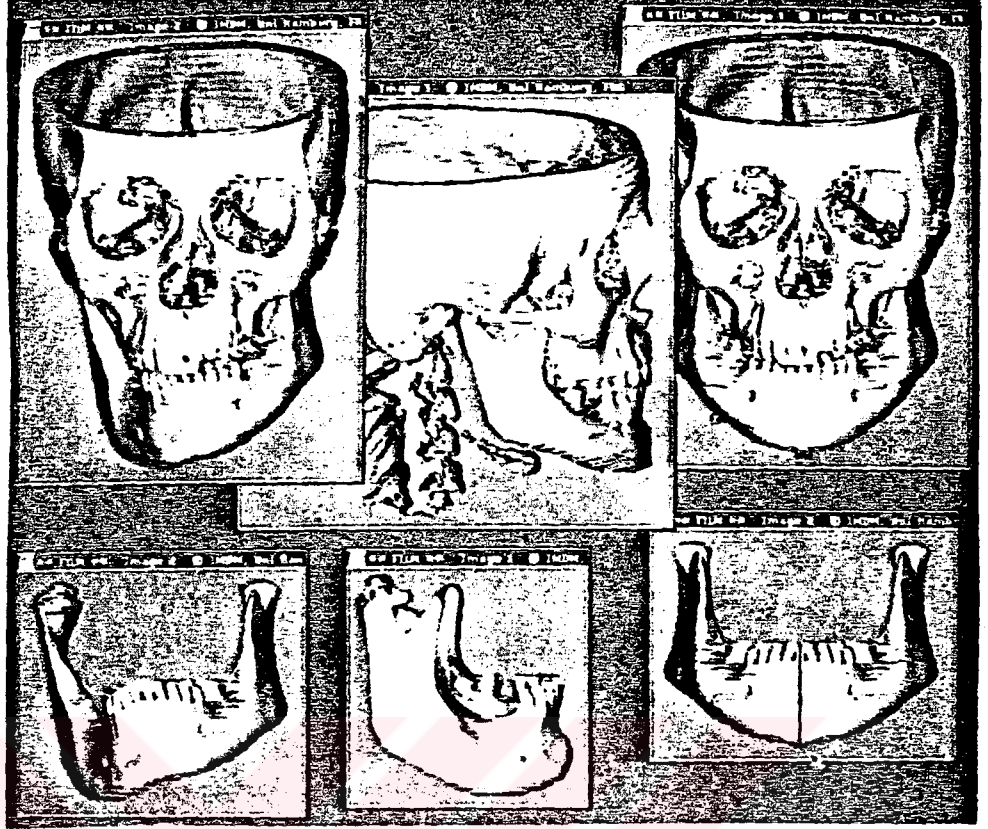
doku arasındaki farkı kolayca ayırabilmektedir. Sinir cerrahi uzmanları operasyon öncesi ve sonrası mümkün derecede fazla bilgi elde etmek ihtiyacındadırlar. MRI ve PET (Positif Emision Tomography) verilerinin aynı 3B görüntüde birleştirilerek kullanılması ile elde edilen beyinin yüzey bilgisini iki boyutlu görüntülerden incelemek mümkün değildir. Bu inceleme ile vücut hareketlerinin temeli olan motor bölgenin, duyumsal bölgenin, işitme ve duyma bölgelerinin incelenmesi sağlanmaktadır. Böylece hem teşhiste hemde tedavi planlamasında kolaylık olmakta ve uzmanlar iki boyutlu tedavi planlamasından daha kolay olduğunu söylemektedirler. [1]

Teşhiste kullanılan bazı yeni uygulamalarıda adları ile vermek gerekirse : Kalbin 3B görüntüsünün ultrasonografi bilgileri kullanılarak elde edilmesi, vücut içi sıvıların (kan, cerebrospinal sıvı gibi) 3B görüntüsünün çıkarılması işlemleri sıralanabilir. [5]

3.2.2 Cerrahi Operasyonların ve Hasta Tedavi İşlemlerinin Planlanmasında 3B Görüntüleme

Kroniofasial (craniofacial) Cerrahi Planlaması: 3B görüntülemenin cerrahi operasyon planlamasındaki en iyi kullanım alanlarından biridir. Uzmanlar operasyon yapılacak olan kemik yapının anatomisini ekranda 3B'lu olarak inceleyerek tüm işlemleri planlayabilmektedirler. Bu teknik sistem için uzun bir hesaplama zamanı ve kullanım için uzman bir operatör gerektirmektedir. Bu şekilde bir operasyonun ekran üzerinde planlanmasına örnek olarak şekil 3.5 verilebilir.[3]

Travmatoloji'de (Traumatology) Cerrahi Planlama : Bu teknikte prensip, kroniofasial cerrahi planlaması ile aynıdır ama bilgi işleme zamanı daha kısa ve acil durumlarda görüntü kalitesi oldukça düşüktür. Genelde yapılan incelemelerde araba kazaları sonucu olan yaralanmalarda hastaların pelvis, diz ve ayaklarından çoğunlukla

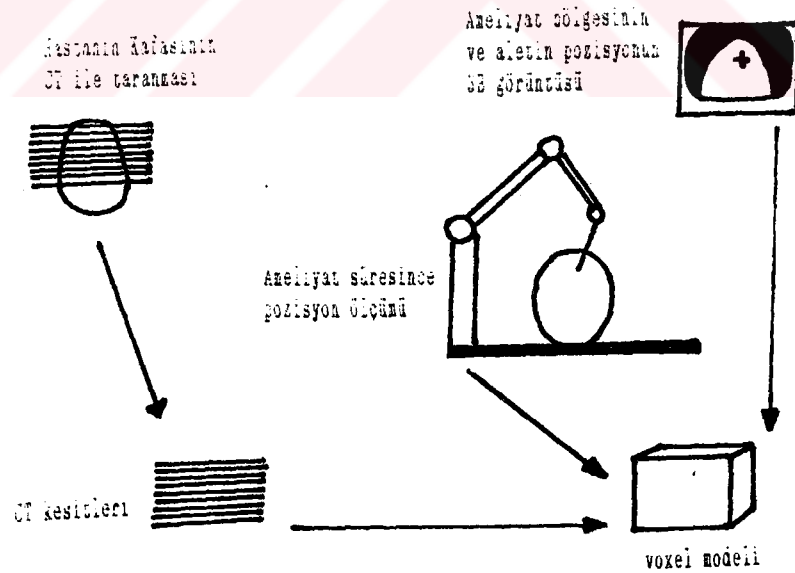


Şekil 3.5 Hastadaki çene kemiği anormalliğinin operasyon ile düzeltilmeden önce ekranda operasyonun planlanması.

belirli şikayetleri oldukları görülmüştür. Bu gibi klasik şikayetlerin incelenmesi ve operasyon planlaması için 3B görüntüleme neredeyse klasik bir yöntem olmaktadır.[3]

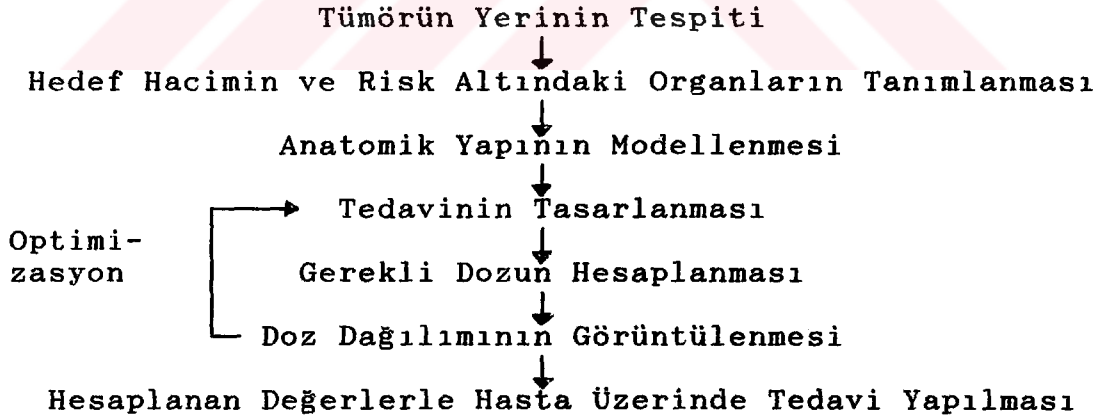
Bilgisayar Destekli Cerrahide : Bu teknik özellikle beyin cerrahları ve kafatası üzerine çalışan uzmanlar için kullanışlıdır. 3B koordinat ölçme ve görüntü teknikleri ve voxel (hacim datasının bulunduğu birim) işleme metodları sayesinde operasyon öncesi planlamada, operasyon sırasında

operasyonun gidişatını denetlemekte ve operasyon sonrası hasta kontrolünde kullanılmaktadır. Bunun için operatörler operasyon yapılacak olan bölge üzerine elle kumandalı elektromekanik bir 3B koordinat sayısallaştırıcısını uygulamaktadırlar. Bu sayısallaştırıcı ile elde edilen bilgilerden ve aynı zamanda CT aracılığı ile elde edilen kesit görüntülerinden yararlanılarak 3B görüntüler elde edilir. Bu görüntülerde hem incelenecek bölge hemde aletin pozisyon bilgisi vardır. Bu şekilde geliştirilen bir sistemin doğruluğu ± 1 mm'dir. Bu sistemle yaklaşık 60 kadar burun-boğaz ve 4 sinircerrahi ameliyatı başarılı olarak gerçekleştirilmiştir. Bu sistemin prensip şeması şekil 3.6'da verilmiştir. [6]



Şekil 3.6 Bilgisayar Destekli Cerrahinin Prensipleri

Bilgisayar Destekli Radyasyon Tedavi Planlanmasında : Radyoterapi tedavisinde tümörlü doku yok edilirken sağlıklı dokuların zarar görmesi istenmez. Bilgisayar destekli radyasyon tedavisi sayesinde her hasta için gerekli olan en uygun radyasyon ışınının dozu ve bu dozun verileceği alan tam olarak belirlenir. İki boyutlu görüntülerde sadece ardışıl kesitler değerlendirildiği için tam uygun doz ve yer bulunamayabilir. Çünkü kesitler ardışıl bile olsa tümörün veya yabancı yapının yeri her kesitte farklılaşarak uzmanları yanıltmakta ve gerçek hedef yanlış tesbit edilmektedir. Oysa 3B görüntüler sayesinde tümörün tam olarak nerede olduğu ve ne kadar yer kapladığı kolayca tespit edilmektedir. Radyoterapi tedavi planlama işlemi aşağıdaki sıraya uyularak yapılır:



Akış şemasından görüldüğü gibi optimizasyon işlemi, aslında bir iterasyon işlemidir ve en uygun dozun bulunması için deneme yanılma metodu ile belirli bir hatayı indirgeyerek sonuca ulaşmaya çalışan bir

algoritmadır. Bu işlem ne kadar çabuk olursa hasta okadar az radyasyon alır. Bu süreyi minimuma indirmek için çalışmalar devam etmektedir. Ayrıca sadece hedef yapıyı tanımlayabilmek için çok iyi segmentasyon algoritmalarına ihtiyacımız vardır aksi halde sağlıklı yapılarda zarar görebilir. [3]

3.2.3 Anatomi Alanı Başta Olmak Üzere Tıp Eğitiminde 3B Görüntüleme

3B görüntüler tıp fakültelerinde okutulan ve insan vücudundaki organik yapıların şekilsel olarak öğrencilere tanıtıldığı ders olan anatomide kullanılmaktadır. Anatomi kitaplarında bulunan karmaşık yapıların, 3B görüntüleme tekniğinin sunduğu imkanlar çerçevesinde incelenmesi ile tıp eğitimi hem kolaylaşmakta hemde kalitesini yükseltmektedir. İnsan vücudundaki yapıların hepsinin bilgisayar hafızasında bir arşiv gibi düzenlenmesi uzun bir almasına rağmen sonuçta çok yararlı olacağı kesindir. Doktor adaylarının bir kitap sayfasından sadece resimine bakarak tanıyabileceği bir yapının 3B görüntüsünü inceleyerek ve istediği yerden keserek, döndürerek yapacağı inceleme mutlaka daha olumlu sonuç verecektir. Bu alanda kullanılacak bir bilgi tabanının oluşturulmasına çalışılmaktadır. [7]

3B görüntülerimizde anatomik olarak makroskopik yapılar incelenmekte, mikroskopik yapılar hakkında fazla bilgi alınmamaktadır. İnsan vücudunun mikroskopik yapısı çok karışık olduğundan 3B görüntülemesi çok karışıktır. Elde edilen iki boyutlu görüntülerdende mikroskopik bilgileri elde etmek çok zordur. Yapılan araştırmalar ile CT kesitlerinden elde edilen 3B görüntülerle insan vücudunun mikroskopik yapısının incelenmesine çalışılmaktadır.

3B görüntüleme ilginç bir uygulama alanı olarak antropolojide'de kullanılmaktadır. Örneğin Mısır'daki mumyalarla uğraşan bilim adamları tarihteki hastalıkları ve o zamanki operasyon yöntemlerini incelemek için 3B görüntüleme yöntemleri kullanmaktadırlar. Böylece mumyalar tahrip edilmemiş olmakta ve antropologlar istedikleri bilgileri ayrıntılı olarak elde etmektedirler.

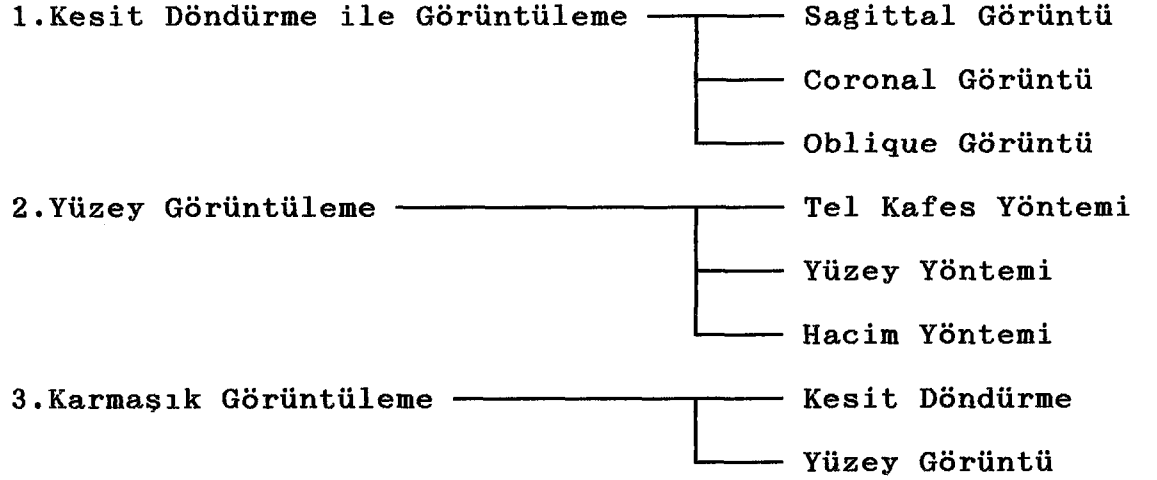
BÖLÜM 4

ÜÇ BOYUTLU GÖRÜNTÜLEMEDE KULLANILAN YÖNTEMLER

Bu bölümde 3B görüntü elde etmek için kullanılan çeşitli yöntemlerden bahsedilecektir. Öncelikle 3B görüntülerin 2B kesitlerden elde edilmesindeki yöntemler anlatılacak ve sonra segmentasyon (doku ayırma), gölgelendirme, döndürme gibi 3B görüntülemenin temel algoritmalarındaki tekniklere değinilecektir.

Son 15 yılda gelişme gösteren 3B görüntüleme yöntemlerini; Yüzey Elde Etme Yöntemleri ve Hacim Elde Etme Yöntemleri diye iki temel sınıfa ayırabiliriz. Işın Tarama ve Ağaç kodlama yöntemleride hacim verileri elde etmek için kullanılan yöntemlerdir ve ikinci sınıfa dahil olmaktadırlar. Yüzey elde etme yöntemleri içindede poligonal modelleme ile yüzey elde etme ve ayrık sınırlarla yüzey elde etme diye adlandırılan iki ayrı yöntem bulunur. Tablo 4.1'de 3B görüntüleme yöntemlerinin sınıflaması verilmiştir. Yukarıda bahsedilen hacim ve yüzey elde etme yöntemleri yüzey gösterme başlığı altında toplanmıştır. Tablo 4.2'de ise ele alınan veriler, bu verilerin girdiği işlemler ve sonucu oluşturan veri yapılarının bir şeması görülmektedir. Tablo 4.2 deki yöntemler Tablo 4.1'deki

Tablo 4.1 3B görüntüleme yöntemlerinin sınıflanması [5]



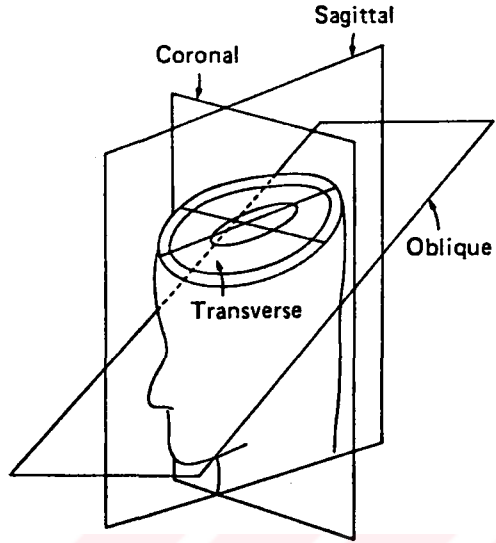
4.Yeniden Görüntüleme

(2) nolu yüzey görüntüleme yöntemlerinin daha açık bir şekilde verilmesi ile oluşturulmuştur çünkü bu yöntemler bizim için temel teşkil etmektedirler. Şimdi bu yöntemleri sırası ile inceleyelim.

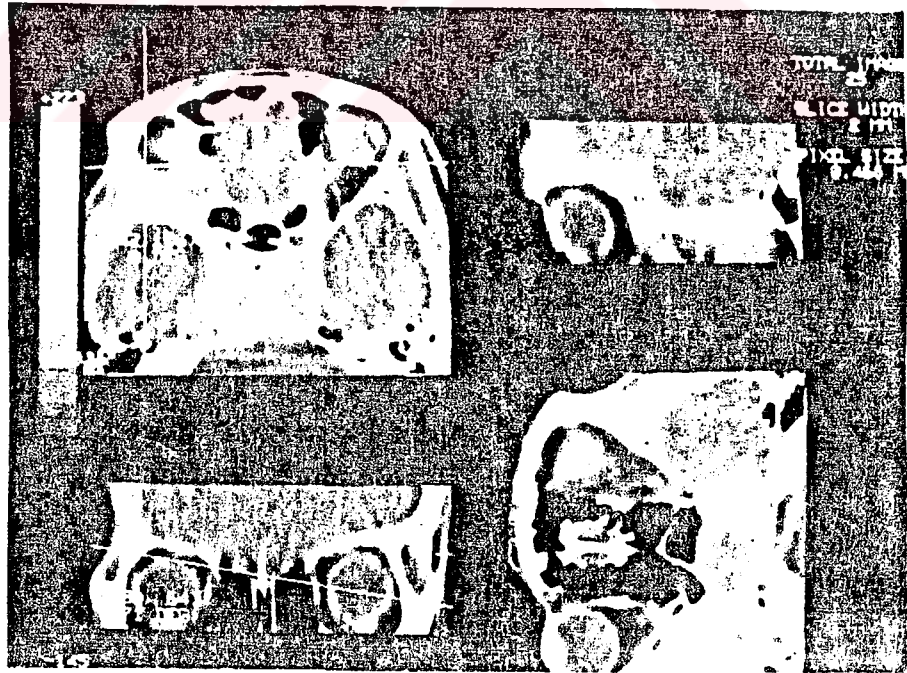
4.1 KESİT DÖNDÜRME İLE GÖRÜNTÜLEME YÖNTEMİ

Şekil 4.1'de görülen bu yöntem, bilgisayar hafızasına gerçek 3B cisimden alınmış verileri koyarak bir insan vücudu görüntüsünü hafızada oluşturmaktadır. Hafızadan alınan görüntüler istenildiği yerden kesilebilir ve ekranda iki boyutlu olarak görüntülenir. (Şekil 4.2)

Bu yöntem aslında bir 3B görüntüleme değildir ama



Şekil 4.1 Kesit Döndürme yönteminin prensip şeması



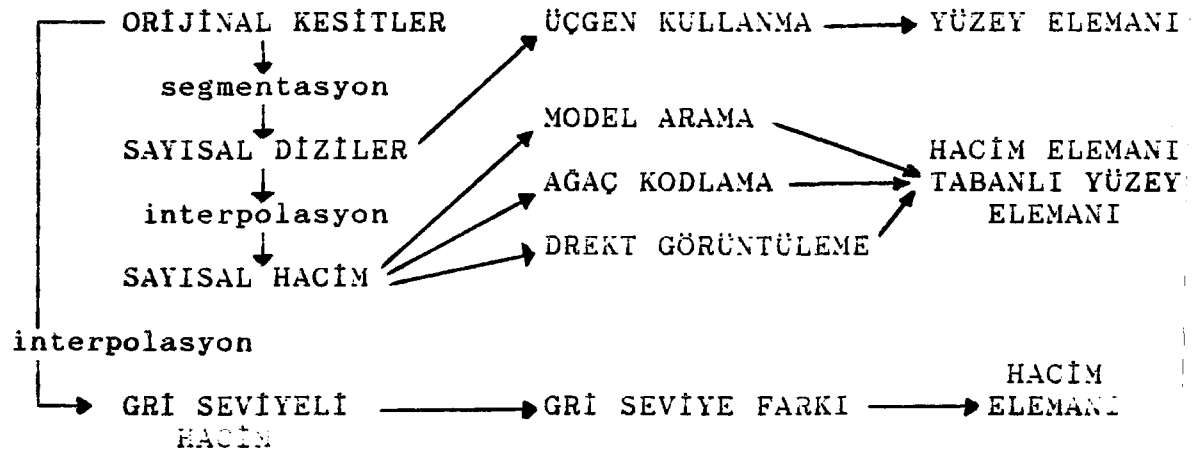
Şekil 4.2 Kesit Döndürme Yönteminin klinik uygulaması

3B görüntüleme teknikleri arasında sayılmasının nedeni görüntüler üzerinde kesme yapıp çeşitli yönlerden görüntülerin analiz imkanının sunulması ile bir çeşit 3B görüntüleme yapılmasıdır. Dört ayrı kesitte yapılan bu incelemenin gerçek 3B görüntülemenin yerini tutmayacağı kesindir ama çok hızlı analiz imkanı tanınması ve görüntülerin fazla yer tutmaması avantaj olarak ele alınabilir.

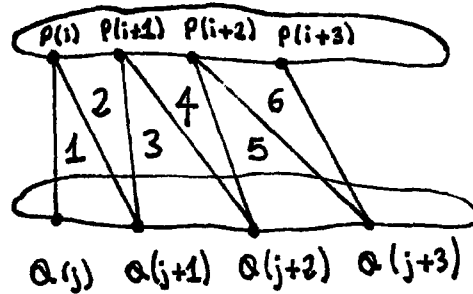
4.2 YÜZEY GÖRÜNTÜLEME YÖNTEMLERİ

Burada Tablo 4.2'deki 3 temel grubun ardından gerekli olan diğer tekniklerdende bahsedilecektir. Tablo 4.2'de önemli yöntemler şematik olarak gösterilmiştir.

Tablo 4.2 Önemli 3B yüzey görüntüleme yöntemleri [8]

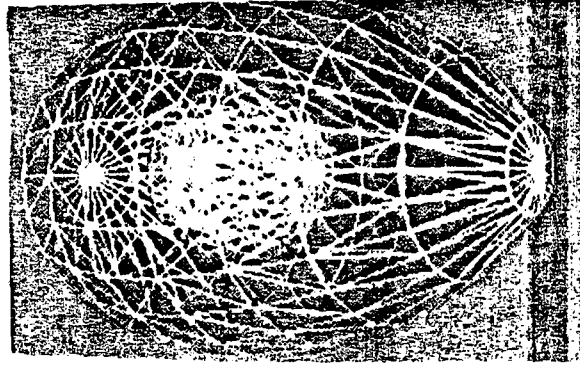


4.2.1. Tel Kafes Yöntemi : Bu yöntem Yüzey elde etme yöntemlerinin poligonal modelleme yöntemi yada özel bir uygulaması olan üçgen yöntemi olarakta bilinir. Ardışıl kesitler arasında 3B izlenimi verecek bir yüzey yoktur ama bu yöntemde kesitler arasında üçgenler hesaplanıp yüzey oluşturulmaktadır. Şekil 4.3'de bir üçgen yüzeyin nasıl oluştuğu gösterilmektedir. Üst kesitteki sınırı belirleyen ardışıl noktalar $P(i)$, $(i=1....n)$, alt kesitteki sınırı belirleyen ardışıl noktalar ise $Q(j)$, $(j=1....n)$, ile verilmiştir. Yüzey oluşturmak için $Q(j)$, $P(i)$, $Q(j+1)$ ile başlanır ve $Q(j+1)$ ile $P(i+1)$ ikinci yüzey oluşturur. $P(i+1)$ ile $Q(j+2)$ birleştirilince 3. yüzey ve $Q(j+2)$ ile $P(i+2)$ birleşince 4. yüzey oluşur ve bu işlem tüm kesitin çevresi tamamlanıncaya kadar devam eder. Bu yöntemde eğer kesitlerdeki sınırlar çok karışık ise sonuçta kabul edilebilir yüzeyler elde etmek güçleşir.



Şekil 4.3 Üçgen yöntemi ile yüzey oluşturma.

Şekil 4.4'de üçgen yöntemi ile elde edilen bir görüntü verilmiştir.



Şekil 4.4 Üçgen yüzey yöntemi ile elde edilen 3B görüntü
(yumurta ve içindeki sarısının görüntüsü).

4.2.2. Yüzey Görüntü Yöntemi : Bir önceki metodda bulunan üçgen yüzeyler uygun biçimde bir renk ile (bizim amacımız için gri seviyesi) doldurulursa görüntüler gölgeli ve daha yumuşak gözükür (Bu yöntemde önce ilgilenilen bölgenin kenar dedeksiyonu yapılmalıdır. Bu noktada işin içine kenar dedeksiyon algoritmaları girer ve 3B görüntülerin kalitesi ve güvenirliliği bu kenar dedeksiyon algoritmalarına bağlıdır). Yüzey elde etme yöntemlerinde ekrana görüntü aktarılmadan önce kenar dedeksiyonu ile gerekli kenarlar çıkarılır ve yüzey elde etme algoritmalarına hazır hale getirilir. Daha sonra üçgen yöntemi kullanılırken elde edilen yüzeylerde belirli optimizasyon kuralları uygulanır (hacim, üçgenlerin alanı, segmentlerin uzunluğu gibi). Sonuçta organa ait bir yüzey bilgi tabanı oluşturulur. Tablo 4.2'de yüzey elde etme yöntemlerinin

bazıları şemasal olarak açıklanmıştır. Yöntemlerin ilkinde CT veya MRI'dan alınan ardışıl kesitler segmentasyon (doku ayırma) işlemine tabii tutulur. Uygulanan segmentasyon işlemi ile beraber kenar dedeksiyonuda yapılır ve bu işlemlerden sonra sonuçlar binary diziler olarak hafızada bulunur. Bu binary diziler üzerinde yapılacak olan işlemlerden birisi üçgen yüzey elde etme yöntemi ile gerekli yüzeylerin hesabıdır. Bir ikinci yöntem ise hafızadaki binary verilerden oluşan kesitlerin interpolasyon algoritmasından geçirilmesidir. Sonuçta elde edilen verilere iki ayrı yöntem uygulanır : Önceden belirlenen model arama veya ağaç kodlama teknikleri. Sonuçta elde edilen görüntülerin anlaşılabilirliği, yüzeyler belirli bir renk ile doldurulduğu için üçgen yüzey elde etme yöntemleri ile elde edilen görüntülerden daha iyidir. Ekrandaki her üçgen yüzeyi gri seviyesi rengi ile doldurmak için her üçgenin yüzey normali ile ışık kaynağının normali skaler olarak çarpılır. Sonuçta elde edilen sayıya göre yüzeyin ne kadar ışık alacağı hesaplanır ve bu sayıya göre ayarlanmış olan gri seviyesi ile üçgenin içi doldurulur. Şekil 4.5'de yüzey elde etme yöntemi ile elde edilmiş olan 3B görüntüler verilmiştir. Hacim görüntü yöntemlerine geçmeden önce yüzey elde etme yöntemlerinde önem kazanan bazı teknikleri açıklayalım.

Kenar Dedeksiyonu : Yüzey elde etme metodlarında ister üçgenler ile yüzey elde etme ister diğer poligonal yüzey

elde etme yöntemlerini kullanalım kenar dedeksiyonu işleminin yapılması şarttır. Kenar dedeksiyonu yapmak için aşağıdaki teknik kullanılabilir.

$P(x,y)$ gri değerine sahip olan noktanın kenar olup olmadığını anlamak için :

$$P(x,y) \in E(k) = \left\{ P(x,y) \mid \begin{array}{l} P(x,y) \neq P(x+1,y) \quad \text{veya} \\ P(x,y) \neq P(x-1,y) \quad \text{veya} \\ P(x,y) \neq P(x,y+1) \quad \text{veya} \\ P(x,y) \neq P(x,y-1) \end{array} \right\}$$

$E(k)$: Elemanaları kenarlardan oluşan matris.

Bu yöntem kenar dedeksiyonu için en basit yöntemlerden birisidir. İyi bir kenar dedeksiyonu sağlayan 2B algoritmalarından Sobel Kenar dedeksiyonunda kullanılabilir. Bu yöntem tarafımızdan kullanılmıştır ve bölüm 5'de açıklanacaktır.

Segmentasyon : (3B görüntüleme yöntemlerinin açıklanması bittikten sonra bu bölümün sonlarında ve bölüm 5'de daha ayrıntılı açıklanacaktır.) Segmentasyon (doku ayırma) 3B veya 2B görüntülerde incelenmek istenen bölge ile, geriye kalan bölgeleri ayırd etmek için kullanılan bir tekniktir. incelenmek istenen bölgeye ait gri seviye değerleri önceden bilinir yada yakın olarak tahmin edilirse belirli kriterler konularak sadece bizim istediğimiz gri seviye aralığı incelenmeye alınır:

Örneğin CT değerleri olarak 0 - 4096 arasında değer alan 2B görüntümüzde bizim istediğimiz alanının

1500-2100 aralığı olduğunu varsayalım. Kullanılacak kriter

$$x \in P(x) = \{ x \mid 1500 < x < 2100 \} \quad x = 0,1,\dots,4095$$

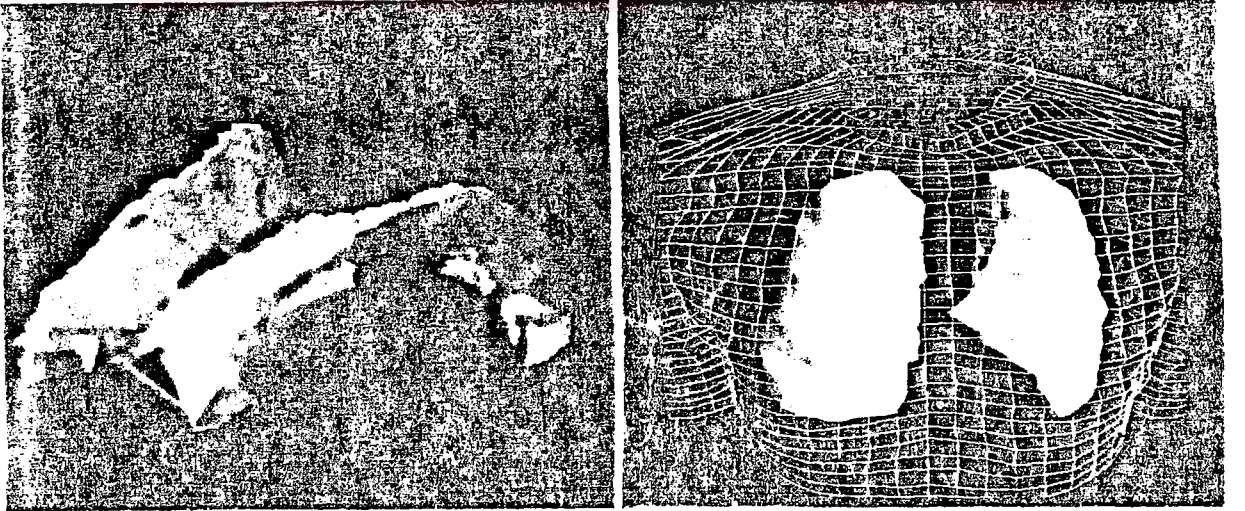
şeklinde olacaktır.

x : CT değerleri

$P(x)$: incelenecek olan bölge

Yüzey elde etme yöntemlerinde segmentasyon çok önemlidir çünkü zaten karışık olan kesit görüntülerinden bütün kenarların dedeksiyonunun yapılması, kenarlar arasında kabul edilebilir yüzeyler oluşturulması çok zordur. Bu yüzden segmentasyon ile azaltılmış verilerle çalışılarak daha iyi sonuçlar alınır.

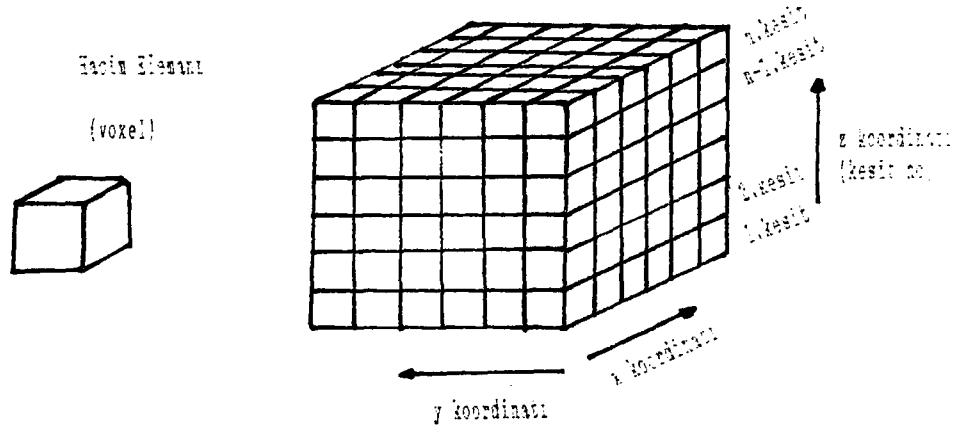
4.2.3 Hacim Görüntü Yöntemi : Bu tekniğin en önemli özelliği gri seviyeli hacim verilerinin görüntü elde etmek



Şekil 4.5 a) 40 ardışıl CT kesitinden elde edilen cerebral

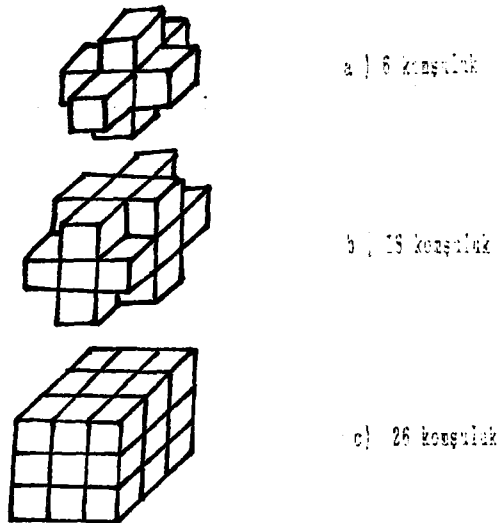
ventriküler bölgenin yüzey elde etme yöntemi ile elde edilen görüntüsü. b) Akciğerlerin yüzey görüntüleme yöntemi ile, göğüsün tel kafes yöntemi ile elde edildiği 3B görüntüler. [7]

için kullanılmasıdır. Bu özelliği ile bilgisayar grafik literatüründe katı model görüntüleme yöntemleri olarak bilinir. Bu yöntemde " voxel " adı verilen bir hacim elemanı kullanılır ve voxelin değeri olarak, kullanılan veri elde etme yöntemine uygun olan bir değişken atanır. Örneğin CT kesitlerinden elde edilen görüntülerde voxel değeri olarak x-ışınlarının zayıflatma katsayısı kullanılır. Şekil 4.6'da voxel elemanı ve bu elemanın hacim içindeki yerleşimi görülmektedir. Kesitlerin şekil 4.6'daki gibi tanımı, ancak CT'nin çözünürlüğü ile (x ve y koordinatları) kesitin çözünürlüğü (kesit kalınlığı ve z koordinatı) eşit ise problem çıkarmamaktadır. Fakat genelde koordinat çözünürlüğü, kesit çözünürlüğünden az olduğu için voxel elemanını kesit aralığını interpolate ederek hesaplamak gerekir. Voxel yada hacim yöntemi şekil 4.7'de verilen 3B görüntü işleme operatörü kullanarak organların yüzeylerinin 3B görüntüsünü çıkartan bir yöntemdir ve bu yüzden tablo 4.1'de yüzey görüntü yöntemleri arasında sayılmıştır. Şekil 4.7'de tanımlanan 3B görüntü işleme operatörleri, 3B'lu bir filtre ile bir lojik filtre içermektedirler. Segmentasyon (doku ayırma yada voxel

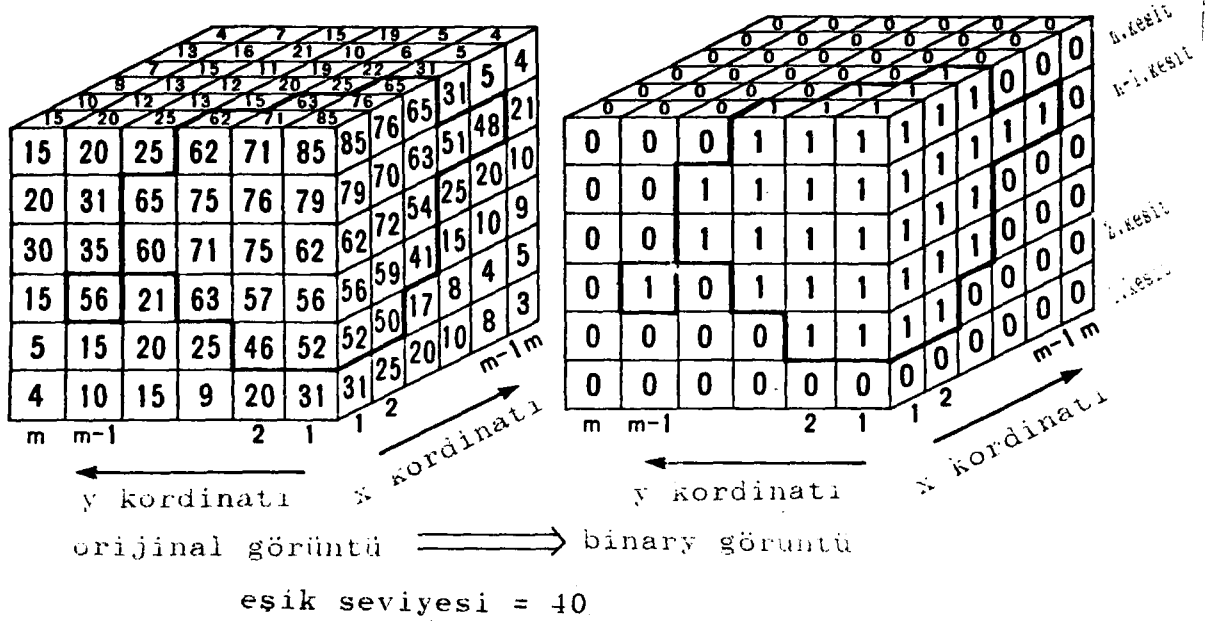


Şekil 4.6 Ardışıl kesit görüntülerin voxel yapısı.

değerlerini ayırma işlemi) işlemi ile orijinal görüntüden binary görüntülere geçme işlemide şekil 4.8'de görülmektedir.



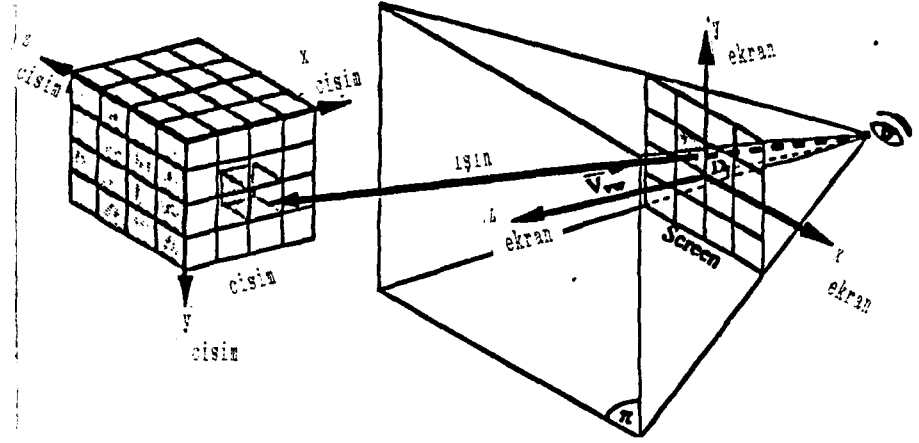
Şekil 4.7 3B görüntü işleme operatörü



Şekil 4.8 Orijinal görüntülerin segmentasyon işlemi ile binary görüntülere dönüştürülmesi.

Hacim elde etme yöntemleri içinde en çok uygulama bulan iki yöntem Işın Tarama ve Octree ağaç kodlama yöntemleridir. Bu önemli yöntemleri sırası ile açıklayalım.

4.2.3.a. Işın Tarama Yöntemi : Işın tarama yöntemi için en çok kullanılan teknik Kamera modeli tekniğidir. (Şekil 4.9). Bu teknikte önemli olan noktalar; Kaynak noktası (göz), Odak noktası (gözün baktığı yer) ve pixellerden oluşan matristir (ekran). Şekil 4.9'daki piramit "gözlem piramidi" adını alır. Yukarıda belirtilen değişkenler yardımı ile gözden gelen ışın ve ışının odaklama noktası ele alınarak bir bakış yönü tanımlanır. Işın tarama yöntemi için, ışınların 3B cisim içindeki geometrilerinin

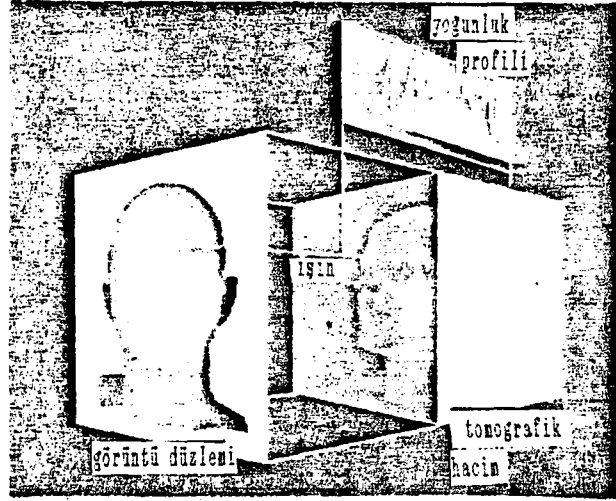


Şekil 4.9 Işın tarama yöntemi için kamera model tekniği.

[5].

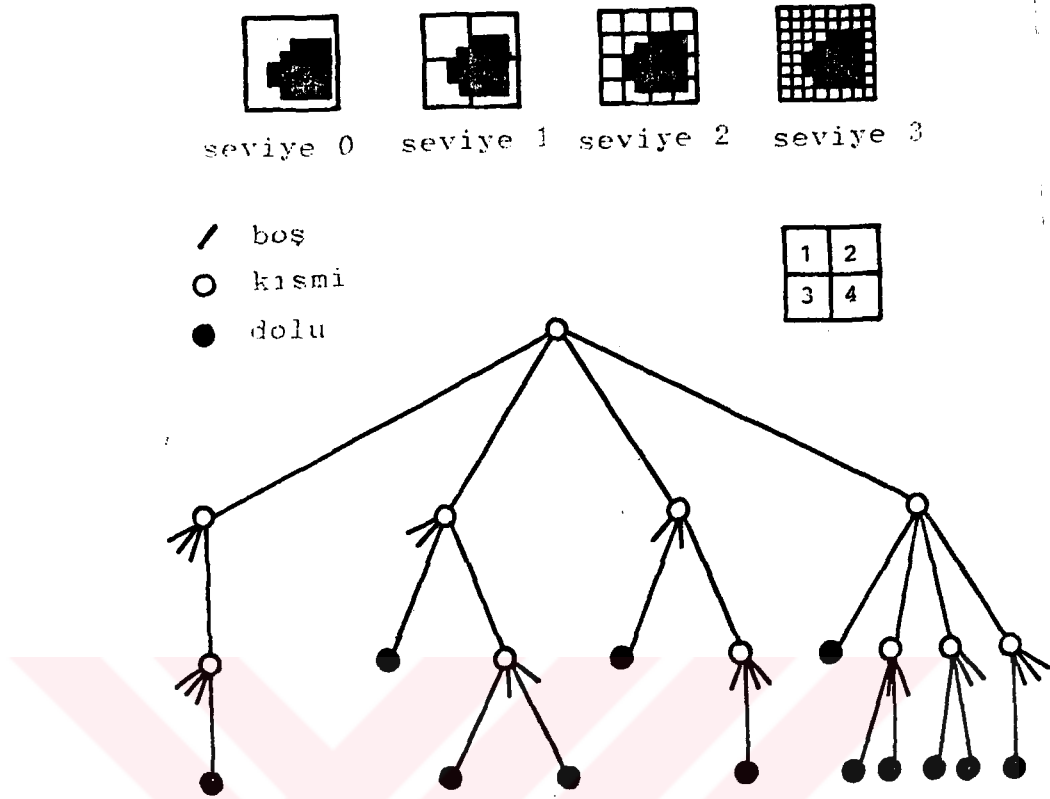
tanımlanması gereklidir. Bu yüzden kaynak noktası ile 3B cisimin pixelleri arasında ilişki kurmak için, kaynak noktasından dik olarak gelen ve pixellerin içinden geçen ışınlar tanımlanır. Ekranda görüntü oluşturmak için gerekli olan pixel değerleri, içlerinden geçen ışınlar yardımı ile bulunan bir yoğunluk fonksiyonuna göre hesaplanır (Şekil 4.10). Bu şekilde, ışının hacimin içinden geçerken oluşturduğu yoğunluk fonksiyonu ve oluşturulan görüntü düzlemi gösterilmiştir.

4.2.3.b Octree Ağaç Kodlama Modeli : Tablo 4.2'den görüldüğü gibi bu yöntemi kullanabilmek için Binary (ikili) dizilerinden interpolasyon işlemi sonucu elde edilen binary hacim verileri kullanılır. Bu yöntemin temeli $2^n \times 2^n \times 2^n$ voxel yapısının bir ağaç şeklini andıran şekilde



Şekil 4.10 Işın tarama algoritmasının prensibi. Işının yolu boyunca elde edilen yoğunluk fonksiyonundan elde edilen bir parametre görüntünün oluşturulmasında kullanılır. [9]

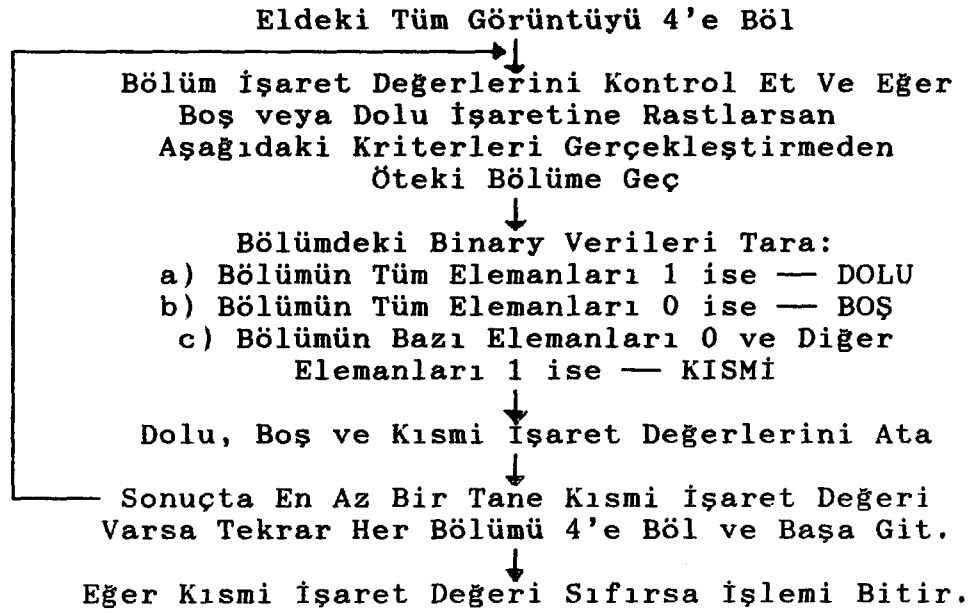
kodlanmasıdır. Bu yöntem adı verilen ağaç kodlama tekniğini 2B binary kesit üzerinden açıklayabiliriz (Şekil 4.11). 3B görüntüleme için kullanılacak octree ağaç kodlama tekniğine geçiş bu açıklamadan sonra hiçte zor değildir çünkü sadece 2B için kullanılan 4 elemanlı inceleme matrisi 3B için 8 elemanlı bir küp olarak tanımlanacaktır. Şekil 4.11'de görüldüğü gibi üç tane düğüm tanımlanır : Boş, Dolu ve Kısmi. İlk başta alınan inceleme alanı tüm görüntü düzlemidir (0. seviye). Tüm düzlem inceleme matrisi ile 4 parçaya bölünür (1. seviye) ve her parça yukarıda verilen 3 düğüm temel alınarak incelenir. Bu incelemenin akış şeması şekil 4.12'de verilmiştir. Bu akış



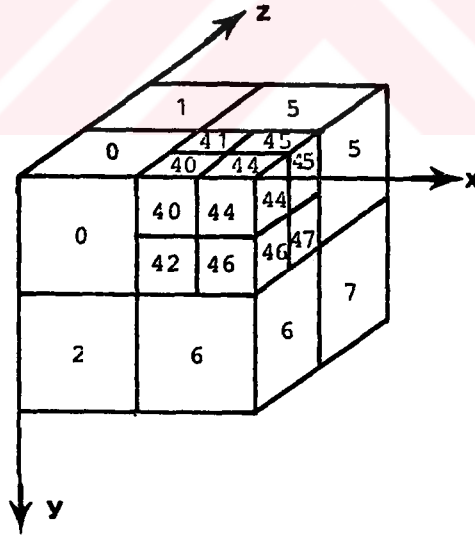
Şekil 4.11 Binary kesit bilgisi üzerinde ağaç kodlama tekniği. [10]

şemasından da anlaşıldığı gibi bu işlemler tüm görüntü boş ve dolu diye tanımlanana dek sürer. Bu işlemi üç boyutlu uygulamaya sokmak için inceleme matrisini şekil 4.13'deki gibi tanımlarsak; Şekil 4.12'de verilen mantık aynen geçerli olur. Tek Fark sadece pixellerin 2B değil 3B taranacak olmasındadır. Düğümlerde saklanacak olan bilginin önceden belirlenmesi şarttır. Bu bilgi yoğunluk değerleri olabileceği gibi, yüzey normal vektörleride olabilir.

Bu yöntemin bir yararıda gereksiz verileri kullanmadığı için işlenecek verilerin azalmasını sağlamasıdır.



Şekil 4.12 Ağaç kodlama yönteminin akış şeması.



Şekil 4.13 3B görüntü için kullanılacak olan inceleme hacim birimi ve bir hacim içerisinde yerleşimi.

Bizim için büyük önemi olan iki yöntemi ve bu yöntemler için kullanılan bazı teknikleri açıkladık. Tablo 4.1'deki 3 ve 4 numaralı yöntemleri kısaca açıklamadan önce yüzey ve hacim elde etme yöntemlerinin bir karşılaştırmasını yapalım.

4.2.3.c. Yüzey ve Hacim Elde Etme Yöntemlerinin Karşılaştırılması :

Yüzey elde etme yöntemleri hacim elde etme yöntemlerine göre daha az bilgi gerektirir ve dolayısı ile daha çabuk sonuç verirler ve görüntülerin saklanması esnasında hafızada fazla yer kaplanmaz. Fakat yüzey elde etme yöntemlerinde sadece yüzey bilgisi ekranda görüntülendiği için bu görüntüler üzerinden kesme işlemi yapıp 3B cisimin içindeki dokular görülemez. Bu işlem için baştan tekrar kesme yüzeyide dikkate alınarak yüzeylerin hesaplanması gerekir. Oysa hacim elde etme yöntemlerinde tüm görüntü hacim bilgisinide taşıdığı için ekrandaki görüntü üzerinden kesme işlemi anında yapılabilir.

Yüzey elde etme yöntemi, Hacim elde etme yöntemi ile elde edilmiş görüntülerden oluşturulabilir çünkü bu bilgi hacim yönteminde mevcuttur ama bunun tersi doğru değildir.

Nükleer tıp gibi çalışmalarda yüzey görüntüleme

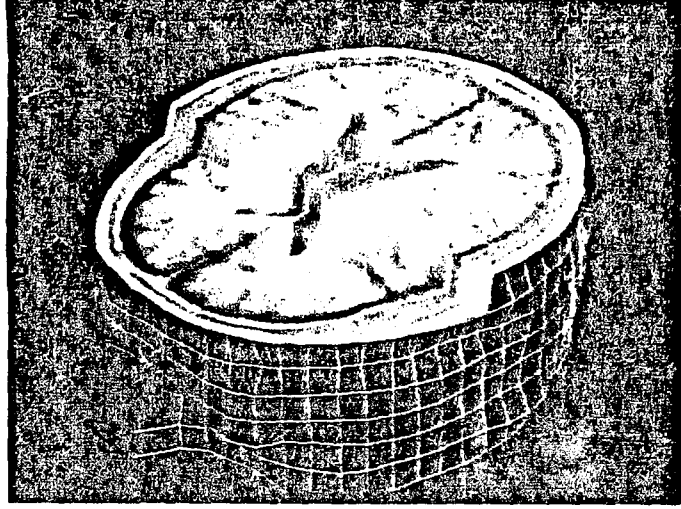
yöntemlerinin sağladığı imkanlar veri bakımından yetersiz kalmaktadır. Bu alanlardaki çalışmalarda hacim elde etme temelli görüntüler kullanılır çünkü gerekli olan hacim hesapları ve görüntüleri bu görüntülerden elde edilir.[11]

Aslında bu iki yöntemin hangisinin üstün olduğunu söylemek yanlış olur çünkü uygulama alanlarındaki üstünlükleri farklıdır ve gerektiğinde birbirlerinin eksikliklerini tamamlamaktadırlar. 5. bölümde daha ayrıntılı olarak açıklanacağı gibi bu tezdeki algoritmalarda her iki yöntemde kullanılmıştır. Yüzey elde etme yöntemlerini sadece hızlı görüntü oluşturmak istediğimiz durumlarda kullandık (animasyon gibi) ve asıl programlarımızda hacim elde etme yöntemleri tercih edilmiştir.

4.3. DİĞER GÖRÜNTÜLEME YÖNTEMLERİ

Bu bölümde tablo 4.1'de verilen 3. ve 4. yöntemler kısaca açıklanacaktır. Bu görüntüleme yöntemleri şu ana kadar açıklanan esasları temel alırlar. Karmaşık görüntü olarak adlandırılan yöntemde, yüzey döndürme yönteminde bahsedilen düzlemler (sagittal, coronal ve transverse) ile yüzey elde etme yöntemleri beraber kullanılır. Bu yöntem kullanılarak elde edilen bir görüntü şekil 4.14'de verilmiştir.

Şekil 4.14'de verilen kesim düzlemi istenilen yerde

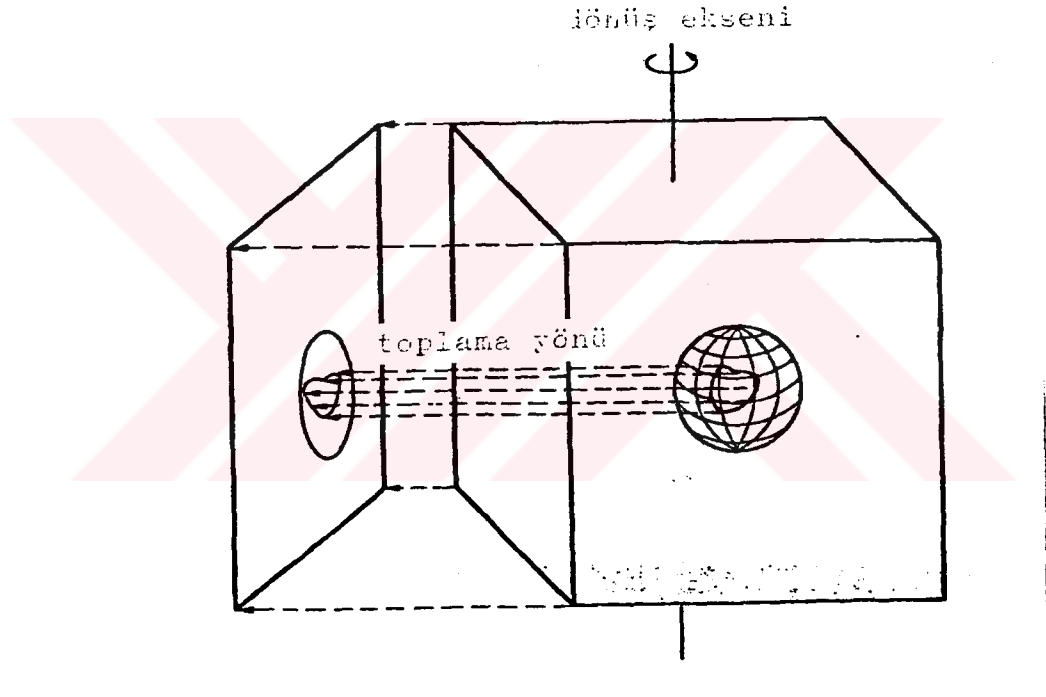


Şekil 4.14 Karma görüntü yönteminin MRI verilerine uygulanması ile elde edilen beyinin görüntüsü. Bu görüntüde beyinin görüntüsü için transverse ve oblique görüntü düzlemleri, yüz görüntüsü içinde tel kafes yöntemi (yüzeyler üçgen yerine dikdörtgen olarak tanımlanmıştır.) kullanılmıştır.

tanımlanabilir. Görüntüleme istenilen yüzey elde etme yöntemi kullanılabilir.

Yeniden görüntüleme yönteminde 3B cisimin, 2B düzlemde istenilen bir yönden görüntülenmesi sağlanır. Bu yöntemi kullanarak ekranda belirli bölgeler daha parlak (highlighted) görüntülenebilir. Şekil 4.15'de yeniden görüntüleme yönteminin prensibi verilmiştir. Hacim yapısının bir eksen etrafında döndürülmesi ile bu hacmin

2B düzleme izdüşümü eksen etrafındaki her dönüş açısı için alınır. Her dönüş açısı için elde edilenler ayrı ayrı kesitler olarak saklanır ve bilinen yöntemlerle birleştirilirse sonuçta 3B görüntüler elde edilir. Bu yöntem kenar bilgileri tam olarak belirlenemeyen cisimlerin görüntülenmesinde kullanılır.



Şekil 4.15 Yeniden görüntüleme yönteminin prensibi.

4.4. 3B GÖRÜNTÜLEMEDE KULLANILAN YARDIMCI TEKNİKLER

Bu bölümde 3B görüntü elde etmek için kullanılan diğer önemli tekniklerden bahsedilecektir. Bu teknikler sırası ile Döndürme, Segmentasyon ve Gölgeleştirme teknikleri olup; matematiksel bağıntılar verilerek ana başlıkların altında farklı yöntemlerden bahsedilecektir.

4.4.1. Döndürme

3B görüntülemede büyük önemi olan döndürme işleminin matematiksel bağıntılarının temeli uzay koordinat dönüşümleridir. Döndürme işleminde, birisi cisim ve birisi de gözlemci koordinatı olmak üzere iki adet koordinat sistemi kullanılır. Önce x, y, z eksenleri etrafında dönme matrislerinin nasıl çıkarıldığını göstermek için şekil 4.16'ya bakalım. Burada P noktasından P' noktasına z eksenini etrafında dönme yapılmaktadır. Şekil 4.16'nın geometrisine bakarak aşağıdaki denklemleri çıkarabiliriz:

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

$$\cos(\theta + \phi) = \cos(\theta)\cos(\phi) - \sin(\theta)\sin(\phi)$$

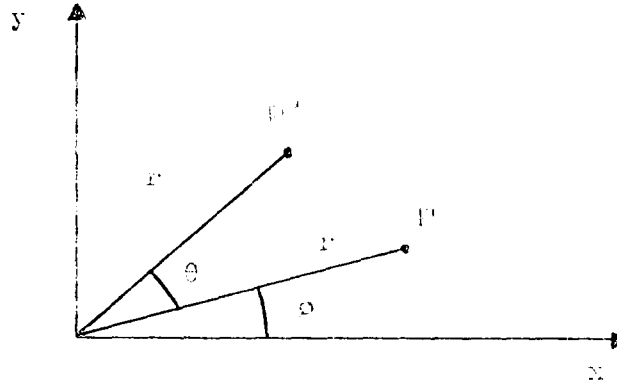
$$x' = r \cdot \cos(\theta) \cdot \cos(\phi) - r \sin(\theta) \sin(\phi) \quad (4.1)$$

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

$$\sin(\theta + \phi) = \sin(\theta)\cos(\phi) + \sin(\phi)\cos(\theta)$$

$$y' = r \cdot \sin(\theta) \cos(\phi) + r \sin(\phi) \cos(\theta) \quad (4.2)$$

$$z' = z \quad (4.3)$$



Şekil 4.16 P noktasının z eksenini etrafında θ açısı kadar dönerek P' noktasını oluşturmaları. [12]

Yukarıdaki denklemler P' noktası için yazılmıştır. P noktasının aşağıdaki denklemlerini ele alalım :

$$x = r \cdot \cos(\phi) \quad y = r \cdot \sin(\phi) \quad (4.4)$$

(4.4) denklemleri, (4.1), (4.2), (4.3) denklemlerinde yerine koyarsak :

$$x' = x \cdot \cos(\theta) - y \cdot \sin(\theta) \quad (4.5)$$

$$y' = x \cdot \sin(\theta) + y \cdot \cos(\theta) \quad (4.6)$$

$$z' = z \quad (4.7)$$

denklemleri elde edilir. Buradanda z eksenini etrafında dönme matrisi bulunur:

$$D_z = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

Z ekseninde dönme yapmak istediğimizde ve yeni

koordinatların bulunması gerektiğinde aşağıdaki işlemi yapmak gerekecektir :

$$\begin{bmatrix} x' & y' & z' \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} * Dz \quad (4.9)$$

Dz matrisinin çıkartılması işlemi x ve y içinde uygulanırsa aşağıdaki dönme matrisleri olan Dx ve Dy'yi elde ederiz:

$$Dx = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (4.10)$$

$$Dy = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (4.11)$$

Bu matrislerle dönme işlemleri şu şekilde gerçekleştirilir :

$$\begin{bmatrix} x' & y' & z' \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} * Dx \quad (4.12)$$

$$\begin{bmatrix} x' & y' & z' \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} * Dy \quad (4.13)$$

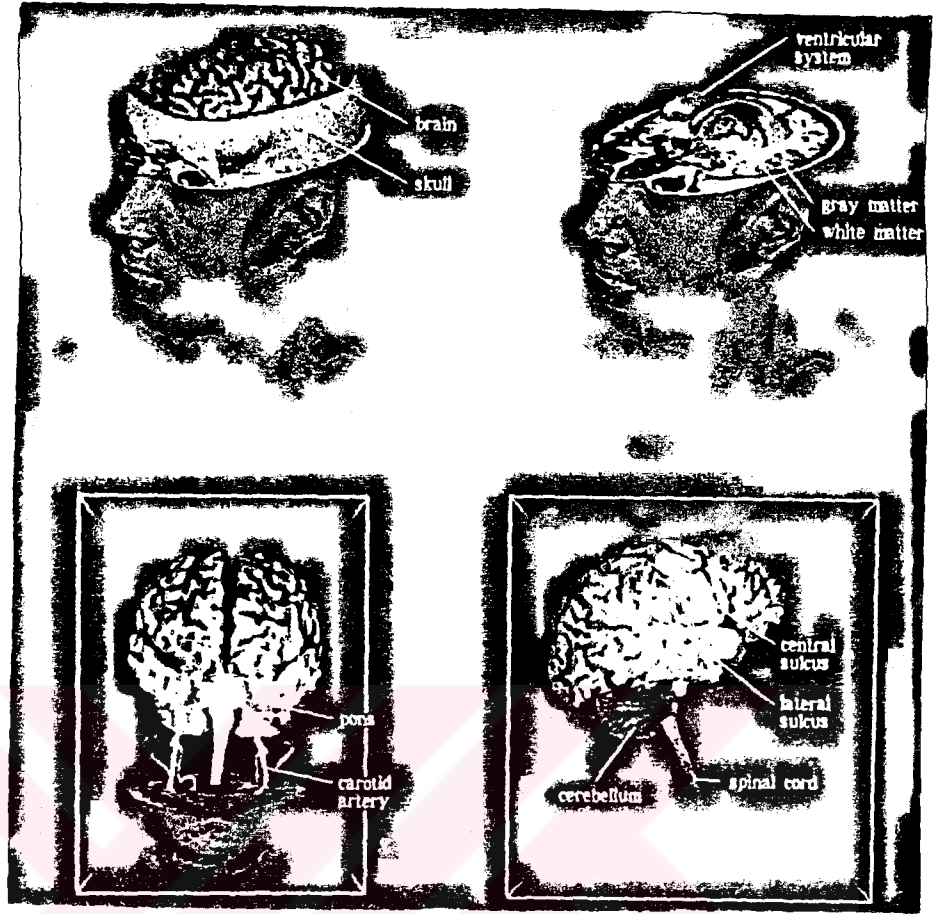
Yukarıda iki adet koordinat sisteminin kullanıldığından bahsedilmektedir. Cisim koordinat sisteminde

cisime ait olan voxeller tanımlanır. Görüntü koordinat sisteminde ise bu voxellerin görüntüleri bulunur. Cisim koordinatları sabittir ve görüntünün koordinatları yukarıda verilen dönüşüm matrisleri kullanılarak bulunur.

4.2.2. Segmentasyon

Segmentasyon (doku ayırma işlemi), 3B görüntülerin analizinde kullanılır. Bu işlemin gerçekleşebilmesi için, kullanılan CT ve MRI'dan elde edilen görüntülerde incelenmek istenen verilerin var olması gereklidir. Segmentasyon işlemi ile, verilerin içerisinde incelenmek istenen bir bölge belirlenir ve bu bölge öteki verilerden ayırılarak ekranda görüntülenir. Bu işlem medikal alanda büyük yarar sağlamaktadır. Uzmanlar sanki ellerinde gerçek organik yapı varmış gibi inceleme yapabilirler. Örnek olarak (Şekil 4.17)'de segmentasyon ile 3B görüntülerden, istenilen yapının elde edilmesi işlemi görülmektedir.

Segmentasyon işleminde önemli olan, istenilen dokuya ait kriterlerin belirlenmesidir. Bu kriterler belli olduktan sonra istenilen doku kolayca ayırd edilebilir ve ekranda görüntülenir. CT ve MRI verilerinde gri seviyesi dokuların özelliklerini temsil ettiğinden, bu özelliği kullanarak segmentasyon yapma işleminin en basit yolu "eşik yöntemi"dir. Eşik yönteminde kullanılan kriteri şu



Şekil 4.17 Segmentasyon ile istenilen dokunun elde edilmesi. a) beyin, kafatası ve yüz yapısının birarada görüntülenmesi. b) beyinde ventriküler bölge. c) carotid arterin segmentasyonu. d) cerebellum ve spinal cord'un görüntülenmesi.

şekilde yazabiliriz :

$$S(x,y,z) = \begin{cases} 1, & P(x,y,z) > \text{eşik} \\ 0, & P(x,y,z) < \text{eşik} \end{cases}$$

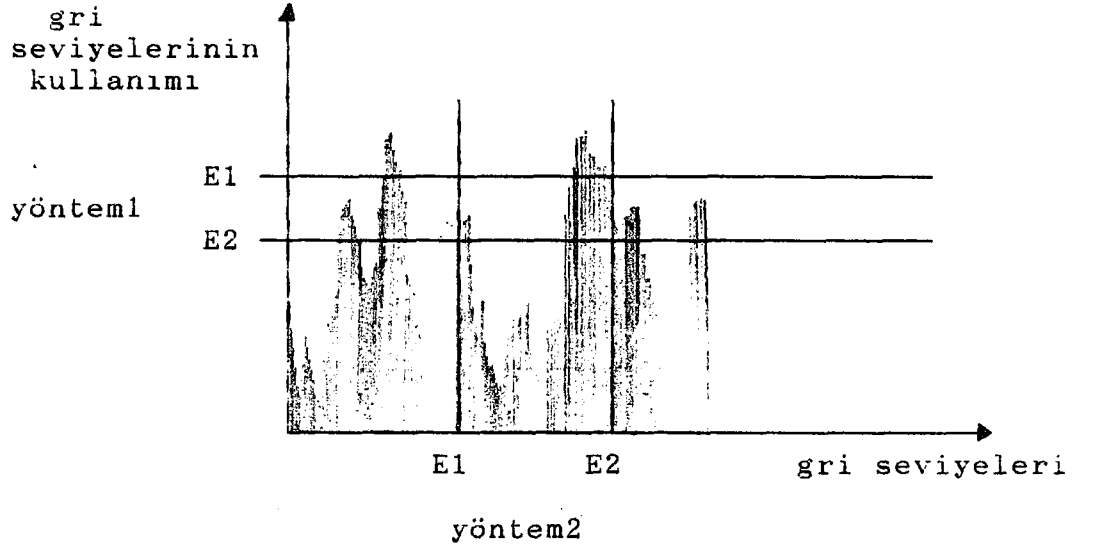
Bu eşitlikte $S(x,y,z)$ segmentasyon sonucunu, $P(x,y,z)$ 'de incelenen cisimi temsil eder. Bu yöntem iki eşik

kullanılarak dahada geliştirilebilir:

$$S(x,y,z) = \begin{cases} 1 & , \text{eşik1} < P(x,y,z) < \text{eşik2} \\ 0 & , \text{eşik1} > P(x,y,z) \text{ veya } P(x,y,z) > \text{eşik2} \end{cases}$$

Eşik seviyelerini iyi kullanabilmek için gri seviyelerinin hangi dokulara ait olduğunu belirlemek gerekmektedir. Bu aşamada histogram analizi işin içine girer. Histogram, bir görüntü içerisinde gri seviyelerinden kaç adet kullanıldığını belirleyen basit bir algoritmadır ama segmentasyon işlemlerinde çok işe yaramaktadır. Görüntüler içerisinde çoğunlukta bulunan dokular, histogram grafiklerinde tepelere neden olurlar (Şekil 4.18). Bu histogram grafiklerinin segmentasyon için kullanılması iki şekilde olur ve bunlar Şekil 4.18'de gösterilmiştir. Birinci yöntemde görüntüdeki tepeleri oluşturan dokuları segmente etmek ve ikinci yöntemde ise sadece istenilen gri seviyesine ait olan dokuyu elde etmek mümkündür. Eğer 2B görüntülerin üzerinde gri seviye analizi (tüm pixellerin gri seviyelerinin tespit edilmesini sağlayan bir algoritma.) yapılırsa segmentasyon için kullanılan eşikler daha etkin bir biçimde belirlenir.

Yukarıda bahsedilen yöntemler segmentasyon için kullanılan basit algoritmalarıdır. Segmentasyon sonuçlarının tatminkar olması (sadece istenilen dokuyu elde etmek) kullanılan yöntemin analiz kriterinin iyi olmasına



Şekil 4.18 Segmentasyon için histogram grafiğinin kullanılması.

bağlıdır. Bu yüzden daha başka birçok yöntem geliştirilmiştir. Örneğin bunlardan biriside, Marr-Hildreht operatörü kullanarak yapılan segmentasyondur [9]. Bu operatör ile yapılan işlemlerin sonucunda ortaya sadece kenar bilgileri çıkar ve bu kenar bilgilerini kullanarak segmentasyon işlemi yapılır. Bu yöntem ile elde edilen kenarların oluşturduğu yüzeyler kapalı ise görüntüleme için önemi büyüktür. Bu yüzeyler elde edildikten sonra kapalı bölgeler olarak etiketlenir. Bu etiketleme işlemi kapalı olan her bölgenin bir numara tarafından temsil edilmesi ile açıklanır. Sonra görüntüleme yapılırken sadece bu numaranın görüntülenmesi istenir. Böylece çok iyi bir segmentasyon yapılmış olur ama yüzeyler kapalı

değilse bunu bir uzmanın bilgisayarda çalışarak elle düzeltmesi gerekmektedir. Buda zaman kaybına neden olur ama sonuçlar gerçekten çok iyidir. Bu yöntem ile beyin zarı, damar yüzeyleri ve daha birçok görüntülenmesi zor doku elde edilir.

Kullanılacak bir başka uygulamada, 2B görüntülerin üzerinde istenilen dokuyu belirlemektir. İstenilen dokuya ait birkaç adet örnek alınır. Örnekler sadece dokunun üzerine mouse kontrolü ile gidip oraya bir nokta koymak ile yapılır. Algoritma alınan tüm örneklerin ortalamasını, standart sapmasını hesaplar ve sonra 3B görüntüleme yaparken alınacak segmentasyon merkezi olarak ortalama değerini, segmentasyon genişliği olarakta standart sapmayı alır. Bu işlemi matematiksel olarak şu şekilde ifade ederiz:

$$SM = \frac{\sum_{i=1}^n x_i}{n} \quad n : \text{Örnek sayısı.} \quad (4.14)$$

SM: Segmentasyon Merkezi.

x: Gri Seviyeleri. ($0 \leq x \leq 255$)

Segmentasyon genişliği ise :

$$SG = \frac{\sum_{i=1}^n (x_i - SM)^2}{n - 1} \quad (4.15)$$

3B görüntüler elde edilirken aşağıdaki kriter uygulanır :

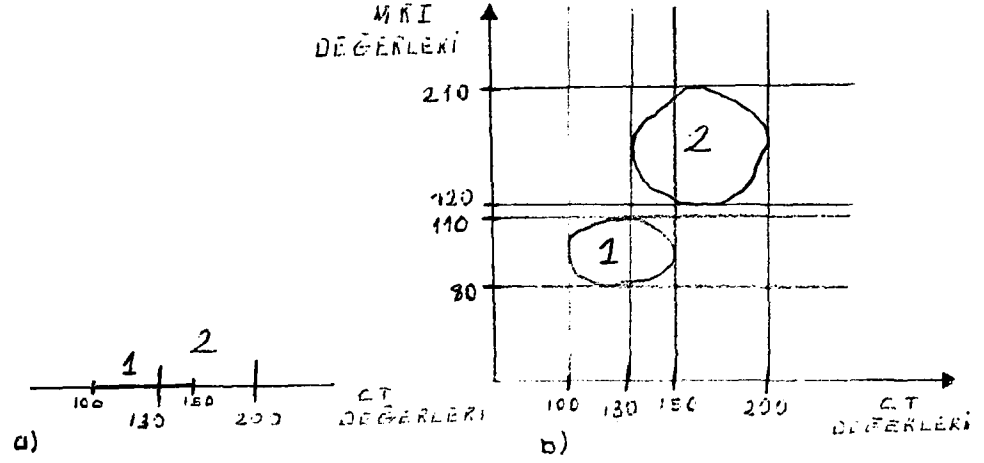
$$SO = \begin{cases} 1, & (SM - SG) < P(x,y,z) < (SM + SG) \\ 0, & (SM - SG) > P(x,y,z) \text{ veya } (SM + SG) > P(x,y,z) \end{cases}$$

SO : segmentasyon sonucu oluşacak görüntü. Bunun 1 değerini alması bu parçanın ekranda görüntüleneceğini belirtir.

$P(x,y,z)$: 3B cisimin tümünü temsil eder.

Şu ana kadar sadece tek bir kaynaktan (CT veya MRI) alınan verilerin segmentasyonuna değindik. İyi bir segmentasyon yapabilmek için bir başka yöntemde birden fazla kaynak kullanıp (CT ve MRI verileri birarada), bunları aynı görüntü üzerinde birleştirmektir. Bu işlemin yararını şu şekilde açıklayabiliriz:

Tek boyutlu verilerde dokuların gri seviye aralıkları birbirine girişim yapabilir (kenarlarda olan süreksizliklerden dolayı). Tam bu geçiş noktalarında yapılan segmentasyon işleminde istenilenin dışında dokularda görüntülenir. Oysa iki değişik veri kaynağı kullanılarak bu girişimlerin ayırd edilmesi sağlanır. Bu yöntem şekil 4.19'da açıklanmıştır.



Şekil 4.19 a)tek bir veri kaynağından alınan gri aralık sonuçları b)iki adet veri kaynağından alınan gri aralık sonuçları.

4.4.3 Gölgeleme

İki boyutlu bir ekranda görüntüleme yapıldığından 3B izlenimini vermek için, insan gözünü aldatmak gerekir. Bu işlem gölgeleme ile yapılır. Gölgeleme işlemini gerçekleştirmek için, ekranda bir ışık kaynağı (yada kaynakları) tanımlamak gereklidir. Bu ışık kaynağı ile 3B cisimin yüzeyi arasındaki ilişki ne kadar fiziksel gerçeklere uygun olarak modellenirse, gölgeleme işleminin sonucu o kadar gerçeğe yakın olur. Bu yüzden önce ışık ve yüzey arasındaki ilişkiye bakalım.

4.4.3.a. Işık Modeli

Yayılan Işık : Yüzeye gelen ışık her yöne eşit olarak dağılır ve gözlemci tarafından görülen ışığın miktarı

gözlemcinin bulunduğu yerden bağımsızdır. Işığın geldiği yüzeyde olan yansımayı bulmak için ışık kaynağının normal vektörü ile yüzey normali vektörel olarak çarpılır (Şekil 4.20). Bu işlem :

$$I = I_d * k_d * \vec{L} * \vec{N} \quad (4.16)$$

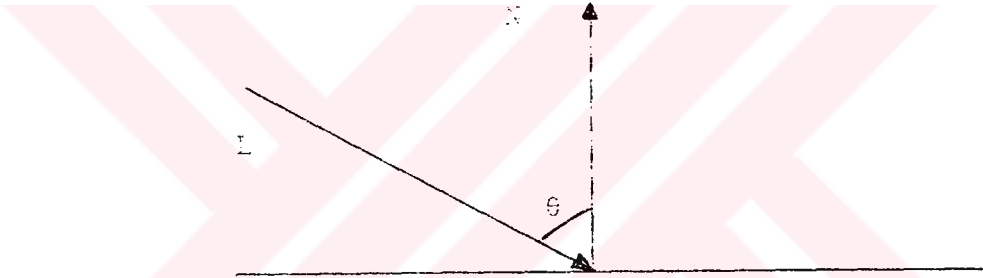
olarak yazılır.

L = ışık kaynağı vektörü

I_d = ışık kaynağının şiddeti

N = yüzey normal vektörü.

$$0 < k_d < 1$$

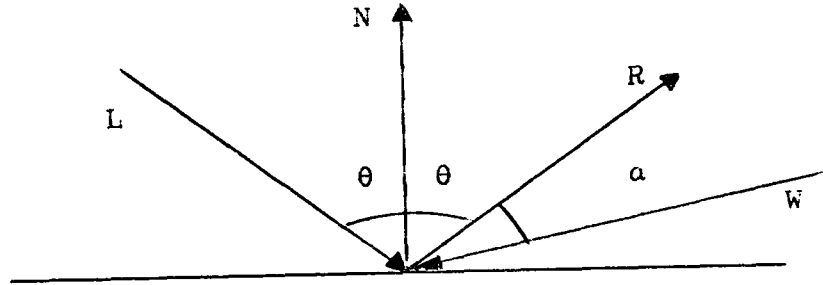


Şekil 4.20 Işık kaynağı ve yüzey normal vektörleri.

(4.16) aşağıdaki şekildedeki yazılabilir :

$$I = I_d * k_d * \cos(\theta) \quad (4.17)$$

Yansıyan Işık : Bu yansıma yüzeye bağlıdır ve ayna gibi parlak yüzeylerde görülür. Gelen ışık yüzeyden yansıyarak belirli bir yöne gider (Şekil 4.21). Gözlemcinin ışığı görmesi için α açısının 0 olması gereklidir. Fakat hiçbir yüzey mükemmel yansıtma sağlamadığı için bu yansıyan ışık belirli bir dağılımla gözlemciye ulaşır.



Şekil 4.21 Yansıyan ışık modeli.

Yansıyan ışık modeli için aşağıdaki bağıntıyı yazabiliriz :

$$I = W(\theta) * \cos^n \alpha \quad (4.18)$$

θ ve α açıları şekil 4.21'de görülmektedir. $W(\theta)$ bileşeni ise, ışığın geldiği yüzeye bağlı olarak 0 ile 1 arasında değer alır. İkinci bileşen olan $\cos^n \alpha$ ise en büyük değerine $\alpha = 0$ iken ulaşır. (n) yüzeyin parlaklık katsayısıdır. Parlak yüzeylerde 200 gibi bir sayı olurken, mat yüzeylerde 5-10 arası bir sayı olur [13].

Bu iki modelin birleştirilmesi ile Phong ışık modeli ortaya çıkmıştır :

$$I = I_k + I_d / (r + k) (k (\vec{L} * \vec{N}) + W(\theta) * \cos^n \alpha)$$

r : gözlemciden olan uzaklık.

$$0 < k < 1$$

I_a : kullanılan öteki kaynakların şiddeti.

4.4.3.b.Gölgeleme Yöntemleri

Gölgeleme yöntemleri 6 çeşittir. 1) Uzaklık gölgelendirme. 2) Sabit gölgelendirme. 3) Görüntü modeli gölgelendirme. 4) Vektör normali ile gölgelendirme. 5) Phong Gölgelendirme. 6) Fark gölgelendirme. Bu yöntemler arasında bizim için önemli olan Fark gölgelendirmedir [8]. Bu gölgelendirme yöntemlerini kısaca açıklayalım :

1) Uzaklık Gölgelendirme : Bu yöntem vektör normalini hesap etmekten kaçınmak için tasarlanmıştır. Bunu başarmak için ışık kaynağı vektörü ile yüzey vektörü arasındaki θ açısı hesaplara katılmaz. Sonuçta yumuşak (keskin hatları olmayan) bir görüntü elde edilir ve kenar bilgisi biraz kaybolur. Bunun sebebi komşu yüzeyler arasındaki mesafenin aynı olması ve kenarların nispeten arka yüzeyler tarafından örtülmesidir.

2) Sabit Gölgelendirme : Bu yöntemde yüzeyin bir noktasındaki yüzey normalini bulmak için, o noktadaki yüzey sınırlarına ait vektörler kullanılır. Bu yöntem ancak yüzey normali ve yüzey sınır normali birbirine uyarsa kullanılabilir. Yüzey sınırlarının içinde gözlemci

tarafından görülebilen noktalara konumuna göre sabit bir gri seviyesi atanarak gölgelendirme yapılır.

3) Görüntü Modeli Gölgeleme : Bu yöntemde bir yüzeye gri seviyesi atarken sadece o yüzey ve onun normali değil, o yüzeye kenardan değen 4 komşu yüzey ve onların normal-leride belirli bir katsayı ile hesaba katılır. Bu yüzeyler arasında yapılan ortalama alma işlemleri ile merkez yüzeye gri seviyesi atanır. Bu seçilen komşu modeli istenirse daha fazla komşu hesaba katılarak genişletilebilir. Bu yöntemle kenarların daha iyi belirginleştiği görülmektedir.

4) Vektör Normali ile Gölgeleme : Bu yöntem, 3. yöntemdeki bazı sorunları gidermek için geliştirilmiştir. Burada yüzey normalini hesaplamak için yüzeyin üzerinde bir koordinat düzlemi oluşturulup, ışığın yönü bu koordinat sistemi içinde hesaplanarak yüzeylere değer atanır.

5) Phong Gölgeleme : Phong gölgeleme yönteminde aşağıda verilen formül kullanılır :

$$S(x,y,z) = \frac{K_a}{d} + \left(\frac{K_d}{s} \left(\vec{L} \cdot \vec{N} \right) + \frac{K_s}{s} \left(\vec{R} \cdot \vec{V} \right)^n \right) \left(1 - K \frac{z}{z_{max}} \right) \quad (4.18)$$

K_a : 0 , K_d : 0.8 (yayılan ışık katsayısı)

K_s : 0.2 (Yansıyan ışık katsayısı)

K : Derinlik faktörü.

n : 5 (Yüzey parlaklık katsayısı)

N: x, y, z 'deki yüzey normal vektörü.

L: Işık kaynağı normal vektörü.

V: Gözlemci yönündeki normal vektör.

R: Yansıyan ışık yönündeki normal vektör.

Z_{max} : 3B cismin maksimum derinliği.

Phong gölgelendirme, bir ağ şeklinde oluşmuş olan düz yüzeylerin eğrisel olarak algılanmaları için yapılan gölgelendirme yöntemidir. Bu yöntem için yüzeylerin birleştiği yerlerdeki her normal vektör hesaplanmalıdır. Bu hesaplanan normal vektörler daha sonra yüzeyler boyunca interpolate edilir. Bu interpolasyon için iki aşamalı bir işlem uygulanır. Önce her yüzey birleşiminde interpolasyon yapılarak buradaki değerler belirlenir. Sonra bu değerler alınarak tüm yüzey işlemi yapılır. Bu interpolasyon işlemi sırasında yüzeylerin her noktasında normal vektörler bulunur. Bu yöntem ile mükemmel sonuçlar elde edilmektedir. Fakat hesaplama için gereken zaman çok uzundur.

4.4.3.c Fark Gölgeleendirme

Bu yöntemde yüzeyler üzerindeki gri seviyelerinin farkları alınarak işlem yapılır. Temel bu olmasına rağmen bu yöntemi kendi arasında 5 ayrı yöntem ayırabiliriz. 1) Z eksenini fark gölgelendirmesi 2) Gri seviye farkı gölgelendirme. 3) Uyumlu gri seviye farkı gölgelendirmesi.

4) Kayan küpler ile gölgelendirme. 5) Şeffaf gri seviye farkı gölgelendirmesi. Bu yöntemleri kısaca açıklayalım. [14]

4.4.3.c.1 Z Eksenli Fark Gölgelemlendirmesi : Bu yöntemde yüzey normalinin bileşenleri fark vektöründen yapılan yaklaşımla bulunur. Sonuçta oluşan dairesel koordinatlardaki vektör kartezyen koordinatlara çevirilir [14]. Bu algoritmanın sonuçları iyi olmasına rağmen genede yüzey açılarının çözünürlüğünün iyi olmaması nedeni ile bazı bozulmalar meydana gelmektedir. Bu sonuçlar öteki fark algoritmaları ile karşılaştırma yapıldığında ortaya çıkar (Şekil 4.23). Bu yöntemde sonuçlara dikkat edildiğinde yüzeyler arası bir süreksizlik ve yüzeylerde pürüzler görülmektedir. Fakat algoritmanın ince kenarlı yapıları gölgelendirmede başarılı olduğu söylenebilir. Bu ince kenarlı yapılara örnek olarak göz çukurunun çevresi verilebilir. Kafa görüntülerinde kafatası kemiklerinin birleşim yerlerini (suture) görmek, yüzeylerin pürüzlü olması nedeni ile mümkün değildir.

4.4.3.c.2 Gri Seviye Farkı ile Gölgelemlendirme : Yüzey normalini hesaplamak için gri seviyeli veriler kullanılırsa, daha iyi sonuçlar elde edilir. Tomografik veri elde etme tekniklerinin bir sonucu olarak, herhangi bir yüzey noktasına komşu olan öteki noktalardaki gri seviyeleri bu noktalarda bulunan dokuların (hava/deri , yumuşak doku/kemik gibi) bağıl ortalamasını temsil eder. Bu bağıl

hacimler yüzey eğimi ile ilgilidir ve bu yüzden gri seviye farkı yüzey eğiminin bir ölçüsü olarak ele alınır. Bu yöntem ile yüzey normalini belirlemek için aşağıdaki denklemleri kullanırız:

$$\begin{aligned} G_x &= g(i+1,j,k) - g(i-1,j,k) \\ G_y &= g(i,j+1,k) - g(i,j-1,k) \\ G_z &= g(i,j,k+1) - g(i,j,k-1) \end{aligned} \quad (4.19)$$

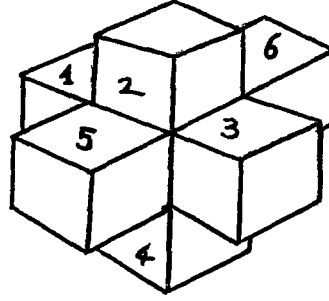
$g(i,j,k)$: i,j,k noktasındaki gri seviye değeri.

G_x, G_y, G_z : gri seviye farkları.

Yüzey normalinin bileşenleri şu şekilde normalize edilir:

$$N_u = \frac{G_u}{\sqrt{(G_x)^2 + (G_y)^2 + (G_z)^2}} \quad u = x, y, z \quad (4.20)$$

Bu yöntemde, gri seviye farkları 6 gri seviyesinin komşuluğu dikkate alınarak yapılmaktadır (Şekil 4.22). Bu komşuluk ilişkisi için 26 komşulukta Şekil 4.22' den görüldüğü gibi sadece yüzeyden değen komşular ele alınmaktadır. Oysa 26 komşuluk kullanıldığında 3 cins komşuluk söz konusu olur. Bunlar yüzeyden değenler, kenardan değenler ve nokta olarak değenlerdir. Temas yüzeyi azaldıkça gri seviye farkı etkisinin azaldığı düşünülerek, bu komşuluk farkları belirli bir katsayı ile çarpılarak



Şekil 4.22 6 komşuluklu gri seviye farkı gölgelendirmesi işleminde komşuluk ilişkileri.

kullanılır. Tezimizde bu yönteme benzer bir yöntem kullanıldığından hesaplamalar ve komşuluk sisteminin açıklaması bölüm 5'de yapılacaktır. Bu algoritmada, z eksenli fark gölgelendirmesi algoritmasına göre daha iyi sonuçlar elde edilir. Daha iyi yüzeyler elde edildildiğinden daha çok ayrıntı yakalamak mümkündür. Yanlız ince kenarlı yüzeylerde iyi sonuçlar elde edilmemektedir (Şekil 4.23). 26 komşuluk kullanmakta ince kenarlardaki görüntü bozukluklarını gidermemektedir. Bunun nedeni bu yöntemde gri seviyelerinin cisimlerin kalınlığı ile ilgili olarak değişmesidir. Bu değişim kenarı oluşturan noktaların gri seviyelerinin, komşu oldukları fakat cisime ait olmayan

noktaların gri seviyelerine bağlı olması ile oluşur. Eğer komşu gri seviyeleri daha yüksek bir gri seviyesine sahipse kenar noktalarının gri seviyeleri cisime ait olanlara göre daha yüksek olmaktadır. Komşu gri seviyesi daha düşükse, tam tersi doğrudur. Bu aksaklık aşağıdaki yöntemde giderilmiştir.

4.4.3.c.3 Uyumlu Gri Seviye Farkı Gölgelelendirmesi : İnce kenarlı cisimler için 6 komşuluk bile fazla gelebilir. Bu yüzden cisimin kalınlığına uyum sağlamak için sadece 3 ile 6 komşuluk arasında seçim yaparak yüzey normali hesaplayan bir algoritma tasarlanmıştır. Bu algoritmada G_x şu şekilde hesaplanır: (G_y ve G_z 'de aynı şekilde):
Eğer (i,j,k) noktasındaki gri seviyesinin değeri, komşuları olan $(i-1,j,k)$ ve $(i+1,j,k)$ 'dan büyükse (küçükse), fark değerini hesaplamak için (i,j,k) ile komşuları içinde daha düşük değere sahip olan (daha büyük değere sahip olan) alınır. Aksi takdirde olabilecek her durumda her iki komşunun farkı alınır. Bu yöntemin sonuçları normal katı cisimlerde aynı iken, ince kenarlara sahip olanlarda kendini gösterir.

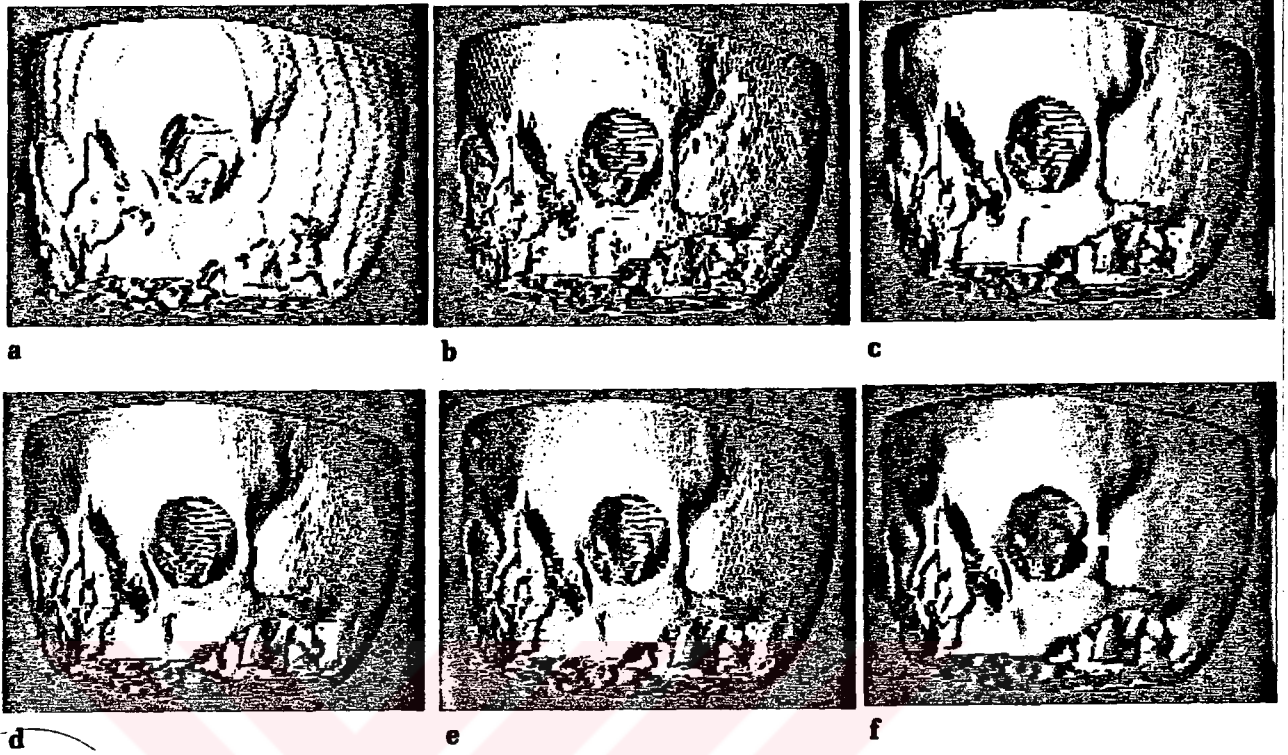
4.4.3.c.4 Kayan Küpler Segmantasyon Gölgelelendirme : Bu yöntemde bilinen yüzey elde etme yöntemleri ile gri seviye farkı yöntemi beraber kullanılır. Önce $2*2*2$ boyutlarında komşuluk oluşturan hacim birimleri ele alınarak bir küp tanımlanır. Cisime dahil olan hacim elemanları (belirlenen bir eşiğin altında veya üstündeki gri seviyeleri), ele

alınarak yukarıda tanımlanan küpün içine üçgenler yerleştirilir. Bu üçgenlerin tepelerine ait olan yüzey normal-leri, küpün kenarlarındaki gri seviye farklarının lineer interpolasyonu ile hesaplanır. Bundan sonra tanımlanan bu yüzeyler ve normal vektörleri ile bilinen bir yöntem kullanılarak gölgelendirme yapılır. Bu yöntemde oldukça iyi yüzey görüntüleri elde edilir ve ayrıca ince kenarlı cisimlerde sorunların yokolduğu görülür.

4.4.3.c.5 Şeffaf Gri Seviye Farkı Gölgelemesi : Cisimlerde daha fazla ayrıntıyı birarada görmek için (örneğin yumuşak doku) yada doğru yüzey tanımlarını otomatik olarak elde edemediğimiz zaman kullandığımız bir yöntemdir. Bir cisimin içine bakmanın en kolay yolu, her gri seviyesine bir şeffaflık değeri atamaktır. Bu gölgelendirme algoritması aşağıdaki şartları gerçekler :

- 1) Bilinen gri seviyelerine, şeffaflık değerleri atanır.
- 2) Yüzeylerdeki şeffaflık değerlerini daha iyi hesaplamak için gri seviye farkları bir ağırlık faktörü olarak kullanılır.
- 3) Bir ışın tarama algoritması kullanılarak cisimin içinde ve dışında görülmesi gereken tüm yüzeyler tespit edilerek gri seviye farkı ile gölgelendirme yapılır.

Buraya dek 3B görüntülemeye önemli olan bazı tekniklerden bahsedildi. Bundan sonraki bölümde, tezde



Şekil 4.23 46 adet tomogram kesitinden elde edilmiş, 3 aylık bir çocuğun kafatasının 6 ayrı yöntem ile gölgelendirilmesi. a) Z eksen fark gölgelendirme. b) 6 komşuluklu gri seviye farkı gölgelendirme. c) 26 komşuluklu uyumlu gri seviye farkı gölgelendirme. d) Kayan küpler segmentasyon gölgelendirme. e) Phong gölgelendirme. f) 5 iterasyon ile yapılmış Phong gölgelendirme.

yapılanlar, kullanılan matematiksel algoritmalar verilecektir. Bu algoritmaların bir kısmı bölüm 4'de zaten verilmiştir (Örneğin gölgelendirme ve segmentasyonun bazı yöntemleri gibi). Bölüm 4'de verilmeyen ama 3B

görüntüleme için önemli olan yardımcı tekniklerden biri olan "görünmeyen yüzeylerin eliminasyonu" işlemi ve "kesme" işlemi , tezde geliştirilen bir yöntem olarak verilecektir. Bölüm 4'de bahsedilen bazı segmentasyon algoritmaları bu tez çalışmaları sırasında geliştirilmiştir. Bunlar yeri geldikçe belirtilecektir.



BÖLÜM 5

TEZDE GELİŞTİRİLEN ALGORİTMALAR VE YÖNTEMLER

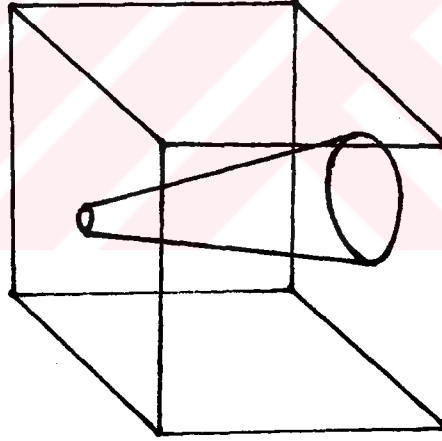
Bu bölümde, tezde kullanılan veriler, eklerde verilen programları yazmak için kullanılan ve geliştirilen algoritmalar ve yöntemler ve gerekli olan bazı 2B görüntü analizleri hakkında bilgi verilecektir.

5.1. TEZDE KULLANILAN VERİLER

5.1.1. Test Modeli

Ek A'da verilen programların ilk yazım ve test aşamalarında işlenecek gerçek veri olmadığından, bilgisayar programı ile sahte veriler (test modelleri) üretilmiş ve bu verilerle programlar test edilip denenmişlerdir. Bu test modelleri, boyutları $512 \times 512 \times 512$ olarak oluşturulmuştur. Bu şekilde oluşturulmalarının sebebi, olabildiğince çok kesit ile ideal görüntüleme yapılabilmesidir. Ama tıp alanında; hastaların 512 kesit alınırken maruz kalacağı x-ışını miktarı, zaman kaybı ve en önemlisi teknolojik kısıtlamalar (Çünkü dar bir bölgenin 512 adet kesitini almak için çok ince kesitler almak gerekir ki bu imkansızdır. Örneğin kafanın boyutu 30 cm ise $30/512 = 0.5$ mm ederki bu hassasiyete hem ulaşılması çok zordur hemde

gerek yoktur.) gibi nedenlerle bu kadar çok kesit alınmaz. Normalde 40 ile 128 kesit arasında bir sayı kullanılır. İlk başta interpolasyon ile kesit çoğaltma işlemini yapmamak için kesit sayısı fazla seçilmiştir. Test modeli görüntülenirken RAM'ın yetersiz olmasından dolayı 256×256 'ya düşürülmüştür. Şekil 5.1'de kullanılan test modeli gösterilmektedir. Bu verinin içindeki koni ile küp ayrı gri seviyelerine sahip olup (koni 50, küp 100), farklı dokuların temsil etmektedirler (kafatası içindeki beyin gibi). Bu sayede segmentasyon işlemlerini test etmek mümkün olmuştur.



Şekil 5.1 Bilgisayar programı ile üretilen sahte kesitlerden elde edilen 3B test modelinin temsili görüntüsü.

Bu verileri üretmek için aşağıdaki formüller kullanılmıştır:

$(x-128)^2+(y-128)^2 > d^2$ ise gri deęeri olarak 100 ata.

$(x-128)^2+(y-128)^2 < d^2$ ise gri deęeri olarak 50 ata.

d koninin apı olup, her kesitte artmakta ve bylece koni tepeden, tabana doęru oluřturulmaktadır. x ve y ise cismin oluřturmak iin yapılan taramanın koordinatlarıdır. ($0 < x < 120$ $0 < y < 120$ olarak taramaktadır.)

5.1.2.Gerek Veriler

Temel programların (3B ekranda grntleme, dnme, kesme, glgelendirme) tamamı test modeli ile denendikten sonra Amerika'dan gelen gerek medikal veriler programlarda kullanılmıřtır. Bu veriler, bir magnetik teyp iinde ve iki cins olarak gelmiřtir.

1) Kafa Verileri : 512*512 znrlęe sahip olan 25 adet CT kesiti. Her bir kesitin yaklařık hafızada kapladığı yer 1.6 MB ile 1.8 MB arasındadır. Bu veriler text dosyası şeklindedir. Text dosyasında veriler řu řekilde bulunmaktadır :

x , y , z

x: Grntndeki noktanın x koordinatı

y: Grntndeki noktanın y koordinatı

z: x,y koordinatlarındaki noktanın zayıflatma katsayısı.

Bu verilerin nasıl okundukları EK A.5'deki programda grlmektedir.

2)Mide Blgesi : 32 adet 512*512 znrlęe sahip CT kesitleridir. Binary olarak kaydedilmiřlerdir. Yaklařık

olarak 200 KB uzunluğundadırlar.

Gerçek verilerdeki yetersizlik dolayısı ile bu kesitlerin algoritmalara uygun olarak çoğaltılması gerekmektedir. Bu işlem interpolasyon ile yapılır.

5.1.3.İnterpolasyon

Bu işlem yukarıda açıklandığı gibi kesit çoğaltmak için yapılmaktadır ama daha sonra açıklanacağı gibi, sonuçta verileri algoritmalara uygun hale getirirken bazı sakıncalı sonuçlarada katlanmak zorunda kalınmıştır. İnterpolasyon ile 25 adet kesit önce 168 kesite çoğaltılmıştır. Bu işlem için aşağıdaki algoritma kullanılmıştır (Bu tezde geliştirilen bu algortimanın uygulama programı EK A.5'dedir).

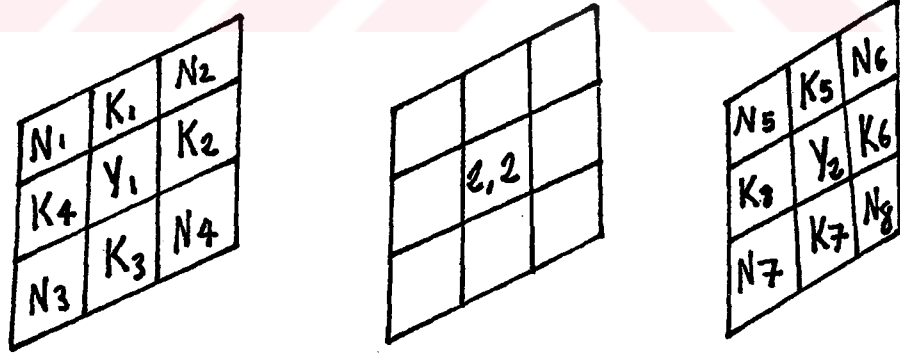


Şekil 5.2 Burada kullanılan interpolasyon algoritmasının üretilen kesitleri nasıl yerleştirdiği görülmektedir.(o : orijinal kesit , ü : üretilen kesit anlamındadır.)

Şekil 5.2'den görüldüğü gibi iki adet orijinal kesitten, 7 adet kesit üretilmektedir. Önce orijinal-kesit 1 (o1)

ile orijinal kesit-2 (o2) arasında 4 nolu kesit (ü4) üretilir. Daha sonra o1 ile ü4 arasında ü2 ve o2 ile ü4 arasında ü6 üretilir. Sırasıyla o1 ve ü2 arasında ü1, ü2 ve ü4 arasında ü3, ü4 ve ü6 arasında ü5 ve son olarak ü6 ile o2 arasındada ü7 üretilir. Böylece her iki kesit arasında üretilen 7 kesit ile 168 kesit sayısına ulaşılır.

Kesit üretme algoritması,interpolasyonun temelini oluşturmaktadır. Bu işlem için Şekil 5.3'de verilen 3*3'lük matris yapısı ele alınır. Ortadaki kesit üretilecek olan kesittir ve aşağıdaki formüller bu kesitin ortasında bulunan (2,2) noktasının değerini bulmak için verilmiştir.



Şekil 5.3 İnterpolasyon algoritmasında kesit üretmek için kullanılan yöntem. Ortadaki kesit üretilecek olan kesittir.

Şekil 5.3'de görülen kesitlerdeki harflerin anlamları şöyledir :

Y : yüzey komşuluğu.

K : kenar komşuluğu.

N : nokta komşuluğu.

Orta kesitteki (2,2) nolu elemanı ele alırsak, bu elemanın 2 adet yüzey komşusu (Y1,Y2), 8 adet kenar komşusu (K1,...,K8) ve 8 adet nokta komşusu vardır (N1,...N8). (2,2) elemanı üretilirken komşuluk derecesi daha yakın olan elemanların üretilecek değer üzerindeki etkileri fazla olmalıdır. Buda belirli bir ağırlık katsayısı ile sağlanır. (2,2) elemanının değeri aşağıdaki formülle hesap edilir :

$$A1 = (Y1 + Y2) * (0.25)$$

$$A2 = (K1 + K2 + K3 + K4 + K5 + K6 + K7 + K8)*(0.35/8)$$

$$A3 = (N1 + N2 + N3 + N4 + N5 + N6 + N7 + N8)*(0.15/8)$$

$$d(2,2) = A1+A2+A3 \quad (5.1)$$

d(2,2): (2,2) elemanının değeri.

Şimdide Y, N ve K harfleri ile verilen kısaltmalarını açık şekilde yazarak yukarıdaki denklemlerde yerine koyalım :

$$A1 = a[x,y,z1]+a[x,y,z2]$$

$$A2 = a[x,y-1,z1] + a[x,y+1,z1] + a[x-1,y,z1] + a[x+1,y,z1] \\ + a[x,y+1,z2] + a[x,y-1,z2] + a[x+1,y,z2] + a[x-1,y,z2]$$

$$A3 = a[x+1,y-1,z1] + a[x-1,y+1,z1] + a[x+1,y-1,z1] + \\ a[x+1,y+1,z1] + a[x-1,y+1,z2] + a[x-1,y-1,z2] + \\ a[x+1,y+1,z2] + a[x+1,y-1,z2]$$

Yukarıda verilen denklemler sadece bir elemanın bulunmasını sağlar ama tüm görüntü için temel aynıdır. Sadece bir çevrim içine koyularak 3*3 matrisi kaynak matrislerin (z1 ve z2) yüzeyleri boyunca tarama yapar ve kesit üretilir.

İnterpolasyon işlemi sonucu ortaya çıkan verilerde, yapılan ortalama alma işlemleri sonucu bazı kayıplar olur. Bunun zararı segmentasyon işlemi esnasında görülür çünkü veri kaybı yüzünden istenilen dokular elde edilmeyebilir. Veri kaybı aslında gri seviyeler arasındaki farkların azalması olarak tanımlanabilir. Eğer dokular orijinal kesitlerde zaten gri seviyeleri yakın olarak bulunuyor ise (CT'de damar bilgisi gibi), interpolasyon sonucu elde edilmeleri çok güçtür. İnterpolasyon işleminin bir başka sakıncasıda, eldeki kesit sayısının çoğaltılması ile ekranda oluşan görüntüdeki bozulmadır. (bakınız Şekil 6.4 ve açıklaması) Bu bozulma daha sonra interpolasyon işlemi ile sadece 3 kesit üretilmek koşulu ile giderilmiştir.

5.1.4 3B Görüntülemeye CT ve MRI verilerinin karşılaştırılması

Bu tezde kullanılan veriler CT verileridir ve bu teknikte kesit görüntülerini elde etmek için x-ışınları kullanıldığına göre elde edilen görüntülerde kemik

yapısının ağırlıklı olması gerekmektedir. Bu özelliği dolayısı ile kemik yapısı ile ilgili tıp alanlarında uygulama bulmaktadır. Fakat CT sonuçlarında yumuşak dokuların aynı kalitede elde edilemediğini kabul etmek gerekir. Yumuşak dokuların görüntülenmesi işleminde MRI mükemmel sonuçlar vermektedir. Bu özelliği dolayısı ile özellikle beyinin yumuşak dokusunda yapılacak olan incelemeler için çok elverişlidir. CT ile elde edilemeyen beyinin beyaz ve gri maddesi MRI'da ortaya çıkarılır. Ayrıca MRI verilerinden beyin damarları , ventriküler bölge, cortex gibi önemli yapılarda segmentasyon ile elde edilerek incelemeye alınabilir. Aslında bu iki yöntemin bir arada kullanılması ile daha iyi sonuçlar elde edilir. [15].

Tezde kullanılan CT görüntülerindeki yumuşak dokunun, kemik dokusu gibi etkin bir biçimde görüntülenmesi yapılamamıştır. Kemik gri seviye aralığı çeşitli gri seviye analiz programları ile kolayca tespit edilmiş ve segmentasyon yapılarak kemik yapısı ortaya çıkarılmıştır. Eğer yumuşak doku bilgisi CT'de olsaydı beyinin segmentasyonu ve daha iyi incelenmesi mümkün olacaktı.

5.2. EKRANDA ARDIŞIL KESİTLERİN YERLEŞTİRİLMESİ

3B görüntülemenin temelini oluşturan bu işlem, 3 boyutlu görüntü oluşturmak için ardışıl 2B kesitlerin ekranda uygun bir biçimde dizilmesidir. Bu işlemin gerçekleşmesi için 2B kesitler arasında yüzeylerin tanımlı olması gerekir çünkü bu yüzeyler olmadan 2B kesitler birbirinden kopuk ve bağımsız gözükür. Bölüm 4'de bu yüzeyleri oluşturmak için kullanılan yöntemler anlatılmıştır.

Bu tezde kullanılan yöntemde kesitler arasında bir yüzey ilişkisi kurulmamasına rağmen kesitler arasında bir kopukluk gözükmemektedir. Bunun sebebi, ekran çözünürlüğünün kullanılmasıdır. (1152,900) çözünürlüğüne sahip olan ekranda, arka arkaya dizilen kesitler arasındaki boşluk gözükmez çünkü kesitler ekrana çizilirken bir kesit, bir yatay (yada düşey) ekran çizgisine karşılık olacak şekilde çizilir. Ardışıl yatay (düşey) ekran çizgileri arasında boşluk olmadığından kesitler arası boşlukta gözükmez. 168 görüntü için 168 adet yatay (yada düşey) ekran çizgisi kullanılır. Dönme işlemi sonrasında kesitler arası boşluklar, yüzeylerin oluşturduğu örtüşmelerle gözükmez.

Programlarımızda kullanılan iki adet önemli matris vardır : 3B cisim matrisi ($a[x,y,z]$) ve Resim

Matrisi yada görüntü matrisi ($r[x,y]$). 3B cisim matrisi, veri dosyalarından okunan verilerin yazıldığı matristir ve sadece bu işlenmemiş verileri içinde saklar. Harddisk'e her seferinde ulaşmak çok zaman alacağı için programın (EK A.1) en başında tüm kesit bilgileri harddisk'ten RAM'da bulunan $a[x,y,z]$ alanına aktarılır ve bundan sonra çalışan her algoritma bu 3B cisim matrisini kullanır ve böylece zamandan kazanç sağlanır. Görüntü matrisinde ise, 3B cisim matrisi üzerinde yapılan işlemlerin (döndürme, segmentasyon, kesme, gölgelendirme) sonuçları saklanır ve ekrana basılan işlenmiş görüntüler bu matrisin elemanlarıdır.

5.3. DÖNDÜRME VE GÖRÜNMEYEN YÜZEYLERİN ELİMİNASYONU

Tezde kullanılan döndürme işleminin koordinat dönüşümleri bölüm 4.4.1'de verilen denklemlerle aynıdır. Tezdeki döndürme işlemleri için kullanılan iki adet koordinat sistemi vardır : Cisim koordinat sistemi ve Gözlemci koordinat sistemi. Burada dönme işlemini gözlemci yapmakta ve cisim uzayında bir dönme olmamaktadır. Bu dönme işleminin yapılması için geliştirilen algoritmanın özelliği sayesinde 3B görüntülemelerde önemli olan Görünmeyen Yüzeylerin Eliminasyonu işlemide yapılmaktadır. Bu yöntemi açıklamadan önce koordinat sistemini tanımlayalım ve koordinat sistemleri dönüşümlerini açıklayalım :

Hiçbir döndürme işlemi yapılmadan önce, cisim koordinat

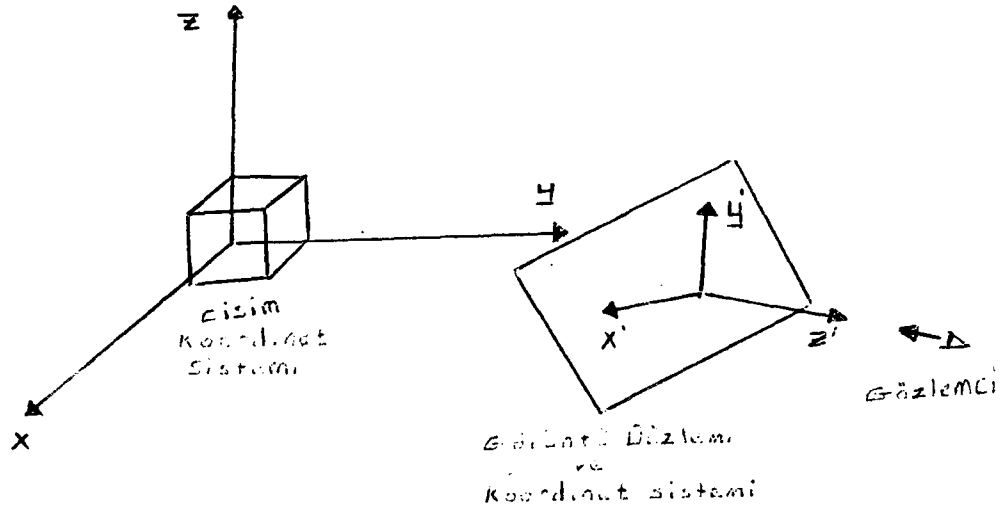
sisteminin x,y,z eksenleri ile gözlemci koordinat sisteminin x',y',z' eksenleri üst üste çakışık bulunurlar. x ve y eksenleri kesit görüntülerinin ayrıtlarını, z eksenini ise ekrandan gözlemciye doğru olan, kesitlere dik ayrıtı temsil eder. Şekil 5.4'de cisim ve gözlemci uzayları arasındaki ilişki verilmiştir. Bu iki koordinat sistemi arasındaki dönüşümler aşağıda verilen Dönüşüm Matrisi ile sağlanır.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta * \sin\phi & \sin\theta * \cos\phi \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta * \sin\phi & \cos\theta * \cos\phi \end{bmatrix} * \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}$$

(5.2)

Bu matris, gözlemci koordinat sistemi yeni bir bakış açısı için döndüğü zaman, gözlemci koordinat sistemindeki dönmenin cismin hangi koordinatlarına karşılık düştüğünün bulunması için kullanılır. Gözlemci koordinatları ile ekrana herhangi bir görüntü basılamayacağı için, cisim koordinat sistemindeki karşılıklarını bulmak gerekmektedir. Bu karşı düşen koordinat değerleri daha sonra resim matrisine atanarak ekranda görüntü oluşması sağlanır. Şimdide yukarıda verilen dönüşüm matrisinin nasıl oluşturulduğunu açıklayalım.

Önce dönmemizin y eksenini etrafında θ açısı kadar ve sonrada bu görüntünün x eksenini etrafında ϕ açısı kadar



Şekil 5.4 Cisim ve Gözlemci koordinat sistemleri arasındaki ilişki.

döndüğünü belirtmek gerekir. Bu dönmeleri aşağıdaki gibi ifade ederiz :

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X & Y & Z \end{bmatrix} * \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \quad (5.3)$$

Bu matris çarpımının sonucunda (5.4) denklem takımı elde edilir :

$$X' = X*\cos\theta - Z*\sin\theta$$

$$Y' = Y*\cos\phi + X*\sin\theta*\sin\phi + Z*\cos\theta*\sin\phi \quad (5.4)$$

$$Z' = -Y*\sin\phi + X*\sin\theta*\cos\phi + Z*\cos\theta*\cos\phi$$

Bu denklem takımından aşağıdaki matrisi yazabiliriz :

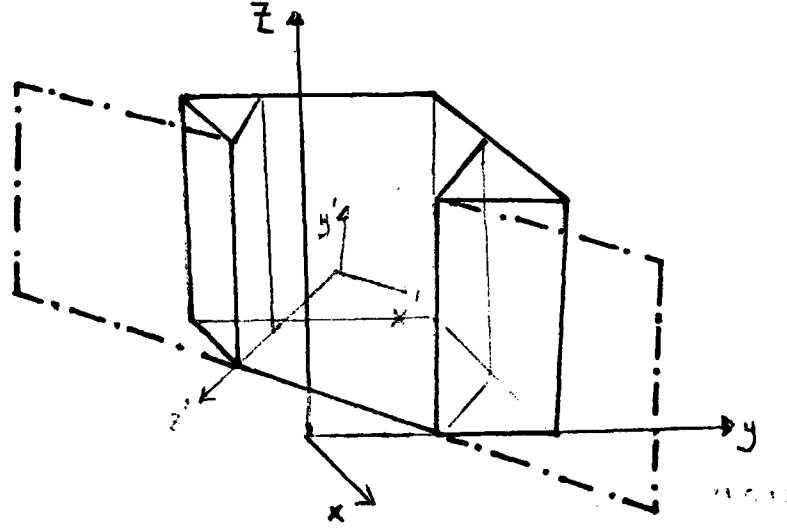
$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ \sin\theta*\sin\phi & \cos\phi & \cos\theta*\sin\phi \\ \sin\theta*\cos\phi & -\sin\phi & \cos\theta*\cos\phi \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (5.5)$$

(5.5) matrisinde X,Y ve Z'lerden oluşan matrisi yalnız bırakmak için trigonometrik fonksiyonlar içeren matrisin tersi ile X',Y',Z' matrisini çarpmak gerekir. Matrisin tersi transpozese eşittir (determinantı 1) ve sonuçta (5.2)'deki matris elde edilir.

Dönme sonucu yeni eksen takımındaki Z' doğrultusu bakış doğrultusunu gösterir. Ancak (Z') 'nün yönü bakış doğrultusunun tersidir. Hacim uzayında bakış doğrultusunda olan her Z' değeri için ortaya çıkan düzleme Görüntü Düzlemi yada Bakış Düzlemi denir. Bu düzlem, dönme esnasında sadece göze gözükecek olan yüzeyleri tespit etmek amacı ile, hacim uzayında bize yakın olan noktadan başlayarak uzak olan noktaya kadar ilerletilir. İlerleyen bakış düzleminin bir elemanı eğer sıfırdan farklı bir değer ile karşılaşır (yani bakış Düzlemi cisim noktacığınla çarparsa) bu koordinatlar bir 2B uzayda işaretlenir. (EK A.1 programında kullanılan m[x,y] matrisi) Bu uzayda "maskeleme uzayı" adı verilir ve bu uzayda saklanan bilgi derinlik bilgisidir. Bakış düzlemi cisim içinde ilerlerken

karşılaştığı noktalar için maskeleme uzayına bakar. Eğer bu noktanın değeri sıfırsa koordinat sistem dönüşümü ile cisim uzayındaki karşılığını (içerdiği veri değerini) tespit eder ve resim uzayına atar. Eğer maskeleme uzayının değeri 0'dan farklı ise bu noktaya bakılmadan atlanır. Bakış düzleminin ilerlemesi esnasında düzlemin karşılaştığı her yeni cisim noktası için, bu noktanın Z' doğrultusu (yani derinliği) boyunca arkasında kalan noktalar işaretlenir ve bunların döndürülmesi ile uğraşılmaz. Maskeleme uzayı sadece döndürmede değil, gölgelendirmede de kullanılmaktadır. Bu uzay gölgelendirme işlemi için sadece gözle görülebilecek olan yüzeyleri temsil ettiğinden aynı zamanda "görülmeyen yüzeylerin eliminasyonu" işleminide yapmaktadır. Şekil 5.5'de bu anlatılan eliminasyon işlemi gösterilmiştir.

Maskeleme uzayının kullanılması ile hem görülmeyen yüzeylerin eliminasyonu yapılmakta, hemde sadece gerekli olan yüzeyler kullanılarak hızdan kazanılmaktadır. Dönen koordinat sistemi olarak cisim koordinat sistemi ele alınırsa, 3B cisimin tüm elemanları için döndürme işlemlerini yapmak gerekir. Böylece fazla veri işlendiğinden zamandan kaybedilir. Algoritmanın daha hızlı çalışabilmesi için, dönme işlemlerinde kullanılacak olan trigonometrik fonksiyonlar programın başında katsayı olarak tanımlanır ve böylece algoritmalarda her defasında trigonometrik



Şekil 5.5 Görülmeyen yüzeylerin z' doğrultusu boyunca eliminasyonu.

işlemler için vakit kaybedilmez. Çünkü bu işlemleri yapabilmek için algoritma her seferinde C lisanının matematik arşivini bulmak ve bunu kullanmak zorundadır. Bu işlem bir kere yapılırsa zamandan kazanılmış olur ve dönme işlemlerinde eğer yön olarak bir değişiklik yapılacak ise sadece bu katsayılar değiştirilerek yeni dönmenin uygulanması sağlanır.

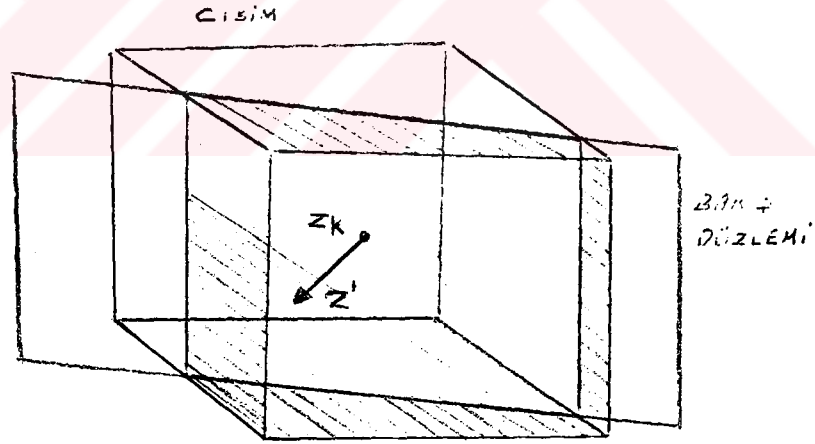
5.4. KESME

3B görüntülerde kesme işlemi uzmanlar açısından önemlidir. Çünkü ekrandaki 3B görüntünün içini görebilmesi için bu işlem şarttır. Kesme işlemi uygulanacak olan veriler hacim elde etme yöntemi ile elde edilmiş olmalıdır. Kesme işlemi ideal olarak istenilen her yönde ve açıda yapılabilmesi ve istenilen kesme yüzey şekli tanımlanabilmelidir.

5.4.1. Bakış Düzleminde Kesme

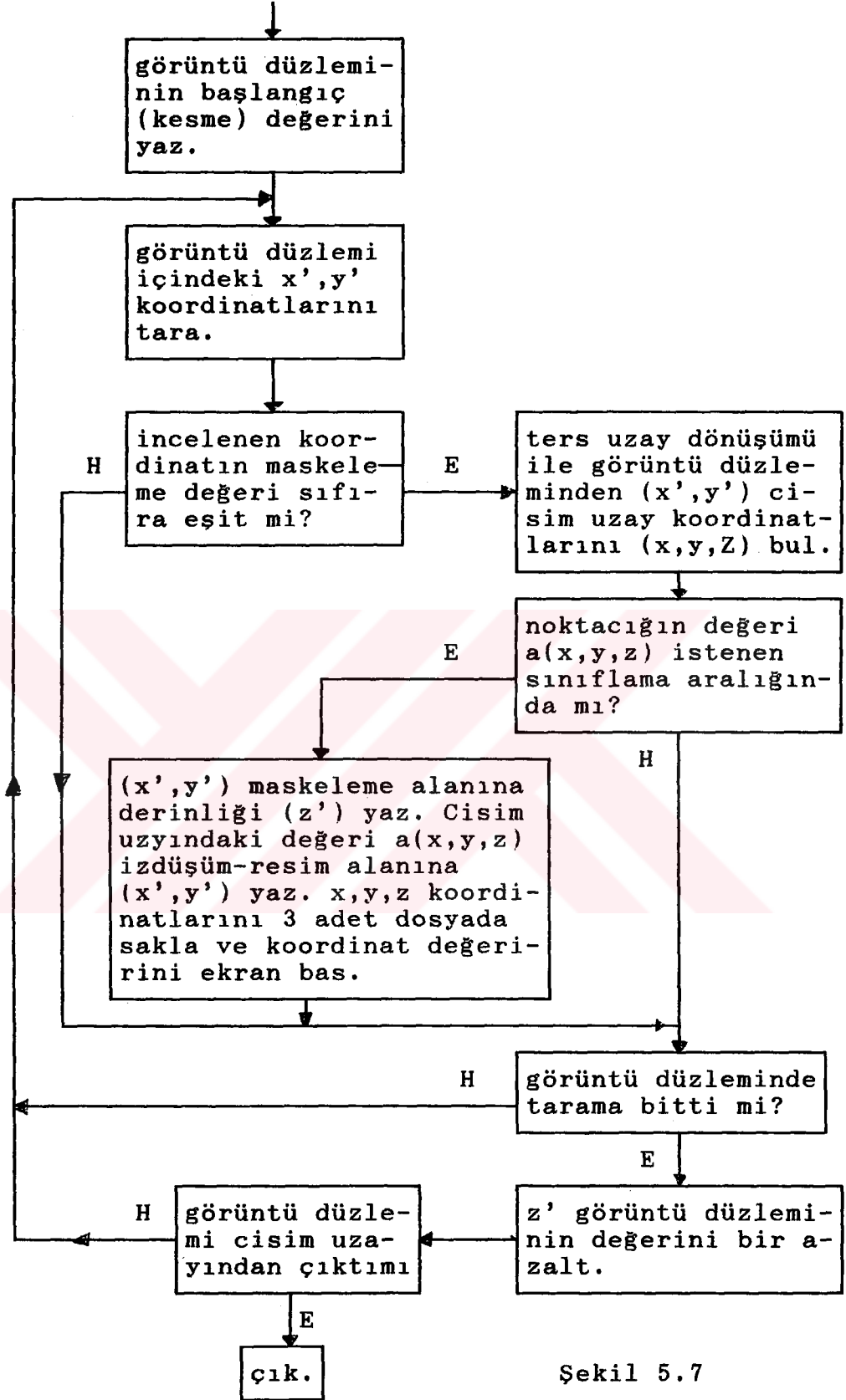
Tezde yapılan ilk kesme işlemi için bölüm 5.3'de kullanılan görüntü düzleminde yararlanılmıştır. Bu yüzden bakış düzlemimiz aynı zamanda kesme düzlemimizdir. Kesme yapmak istediğimizde sadece bakış düzleminin cisim uzayını taramaya başladığı Z' eksen değerini (Z_k : kesme yapmak için verilen değer) vermek yeterlidir. Z' eksen değeri kesme için verilen Z_k değerinden büyük olan ve cisime ait elemanları görüntülenmez ve böylece Z_k değeri ile verilen kesme düzlemi (bakış düzlemi) üzerinde cisimin kesit olarak yapısı görülür. Kesme yapılan bölgede gölgelendirme işe yaramayacağı için bu bölgenin işaretlenmesi gereklidir. Bu işlemde gölgelendirme algoritmasında kullanılan çevrimleri Z_k değerinden başlatmak ile yapılır. Kesilen yüzey gölgelendirilmez ve olduğu gibi basılır. Geri kalan cisime ait noktalar ekrana gölgelendirilmiş olarak

basılır. Kesme düzlemi olarak bakış düzlemini almak matematiksel açıdan kolaylık sağlamaktadır çünkü zaten bakış düzlemi için kullanılan yüzey ve dönüşümler ele alınmıştır ve tekrar hesap etmeye gerek yoktur. Oysa kesme düzlemini bir başka düzlem olarak tanımlarsak yeni bir koordinat sistemi tanımlamak gerekir. Bu koordinat sisteminin daha önce tanımlanan koordinat sistemleri ile ilgili dönüşümlerinin hesaplanması gerekir ve bu işlemlerle veri işleme zamanı uzar fakat istenilen yerden kesme işlemi gerçekleştirilir. Bu yöntem bölüm 5.4.2'de verilmiştir. Bakış düzleminde kesme işlemini daha iyi açıklamak için Şekil 5.6 verilmiştir.



Şekil 5.6 Bakış düzleminde kesme işlemi.

Aşağıda verilen akış şeması, Dönme ve kesme işlemlerini yapan algoritmanın nasıl çalıştığını göstermektedir. Görüldüğü gibi kesme ve döndürme işlemleri aynı algoritma içinde yapılmaktadır. Dikkat edilecek olursa



Şekil 5.7

segmentasyon işlemide bu sırada yapılır. Bu akış şeması EK A.1'deki program için geçerlidir.

5.4.2. İstenilen Düzlemde Kesme

Bakış düzleminde kesme yapmanın inceleme yapan uzmana fazla birşey ifade etmeyeceğini düşünerek, kesme işleminin, kullanıcı tarafından tanımlı bir düzlemde yapılması sağlanmıştır. Yukarıda da bahsettiğimiz gibi bu işlemi gerçekleştirmek için kesme düzleminin üzerinde tanımlı olacağı bir yeni koordinat sistemi gerekir. Bu koordinat sistemi tanımlandıktan sonra, öteki koordinat sistemleri ile arasındaki dönüşümleri oluşturacak olan matematiksel bağıntılar aşağıdaki gibi çıkartılır :

Bu yöntemde ilk ele alacağımız ve kesmenin üzerinde yapıldığını kabul ettiğimiz resim uzayıdır. Normal olarak gözlemci uzayından işe başlamadığımız içinde burada bir denklem takımının çözümü ile karşı karşıya olduğumuzu belirtelim. İlk önce kesme uzayından cisim uzayına olan dönüşümler için aşağıdaki denklemler yazılır :

$$\begin{aligned} ex &= aa1*po1 + aa2*po2 + aa3*po3 \\ ey &= bb1*po1 + bb2*po2 + bb3*po3 \\ ez &= cc1*po1 + cc2*po2 + cc3*po3 \end{aligned} \quad (5.6)$$

Bundan sonra cisim uzayından resim uzayına dönüşüm içinde aşağıdaki denklemler yazılır :

$$\begin{aligned}
dx &= ha1*ex + ha2*ey + ha3*ez \\
dy &= hb1*ex + hb2*ey + hb3*ez \\
dz &= hc1*ex + hc2*ey + hc3*ez
\end{aligned}
\tag{5.7}$$

Öncelikle yeni tanımlanan kesme uzayında yada koordinat sisteminde elde edilen koordinatların, gerçek cisim koordinat sisteminde olan yerini bulmak için (5.6) dönüşümleri kullanılır. Sonra ekrana basılacak olan resim uzayındaki koordinatları bulmak için (5.7) dönüşümleri ele alınır. Bu denklemlerde dikkat edilecek olursa 9 bilinmeyen vardır :

$$po1, po2, po3, ex, ey, ez, dx, dy, dz$$

Bu denklem takımının çözömlü olması için 3 bilinenli, 6 denklem yeter. Bilinen değeerler olarak dx,dy ve po3 değeerlerini alırız. dx ve dy değeerleri kesme düzleminin resim uzayındaki x,y koordinatları ve po3'te kesme düzleminin cisim üzerindeki yüksekliğıdir. Burada yapılan kesme işleminde en önemli ve zor olan işlem kesme düzleminin yukarısında kalan kısmın ekrana basılmamasıdır ve bu işlemi gerçekleştirmek için bir derinlik bilgisine ihtiyaç vardır. Bu derinlik bilgisi dz değışkenidir. Bu değışken yukarıdaki denklem takımı çözülrken zaten bulunacaktır. Şimdi denklem takımını çözelim.

Denklem takımı (5.6)'daki ex, ey, ez eşitliklerini (5.7)'de yerlerine koyalım :

$$\begin{aligned}
dx = & ha1 * (aa1*po1 + aa2*po2 + aa3*yük) + \\
& ha2 * (bb1*po1 + bb2*po2 + bb3*yük) + \quad (5.8) \\
& ha3 * (cc1*po1 + cc2*po2 + cc3*yük)
\end{aligned}$$

yük = po3'tür ama gerçek ifade ettiği anlam yazılmıştır.

$$\begin{aligned}
dy = & hb1 * (aa1*po1 + aa2*po2 + aa3*yük) + \\
& hb2 * (bb1*po1 + bb2*po2 + bb3*yük) + \quad (5.9) \\
& hb3 * (cc1*po1 + cc2*po2 + cc3*yük)
\end{aligned}$$

$$\begin{aligned}
dz = & hc1 * (aa1*po1 + aa2*po2 + aa3*yük) + \\
& hc2 * (bb1*po1 + bb2*po2 + bb3*yük) + \quad (5.10) \\
& hc3 * (cc1*po1 + cc2*po2 + cc3*yük)
\end{aligned}$$

denklemleri elde edilir. Bu denklemlerde po1, po2 ve yük parantezlerini kuralım :

$$\begin{aligned}
dx = & (ha1*aa1 + ha2*bb1 + ha3*cc1) * po1 + \\
& (ha1*aa2 + ha2*bb2 + ha3*cc2) * po2 + \quad (5.11) \\
& (ha1*aa3 + ha2*bb3 + ha3*cc3) * yük
\end{aligned}$$

$$\begin{aligned}
dy = & (hb1*aa1 + hb2*bb1 + hb3*cc1) * po1 + \\
& (hb1*aa2 + hb2*bb2 + hb3*cc2) * po2 + \quad (5.12) \\
& (hb1*aa3 + hb2*bb3 + hb3*cc3) * yük
\end{aligned}$$

$$\begin{aligned}
dz = & (hc1*aa1 + hc2*bb1 + hc3*cc1) * po1 + \\
& (hc1*aa2 + hc2*bb2 + hc3*cc2) * po2 + \quad (5.13) \\
& (hc1*aa3 + hc2*bb3 + hc3*cc3) * yük
\end{aligned}$$

EK A.9'daki programda görüldüğü gibi, 5.11, 5.12 ve 5.13 denklemlerindeki $po1$, $po2$ ve yük'ün çarpanları olan parantez içindeki denklemler birer sabit olarak hesaplanmaktadır :

$$\begin{aligned}
 d1 &= ha1*aa1+ha2*bb1+ha3*cc1 & d2 &= ha1*aa2+ha2*bb2+ha3*cc2 \\
 d3 &= ha1*aa3+ha2*bb3+ha3*cc3 & d4 &= hb1*aa1+hb2*bb1+hb3*cc1 \\
 d5 &= hb1*aa2+hb2*bb2+hb3*cc2 & d6 &= hb1*aa3+hb2*bb3+hb3*cc3 \\
 d7 &= hc1*aa1+hc2*bb1+hc3*cc1 & d8 &= hc1*aa2+hc2*bb2+hc3*cc2 \\
 d9 &= hc1*aa3+hc2*bb3+hc3*cc3 & & (5.14)
 \end{aligned}$$

Bu denklem takımını esas alarak 5.11 ve 5.12'yi tekrar yazalım :

$$dx = d1*po1 + d2*po2 + d3*yük \quad (5.15)$$

$$dy = d4*po1 + d5*po2 + d6*yük \quad (5.16)$$

Buradan,

$$dx-(d3*yük) = d1*po1 + d2*po2 \quad (5.17)$$

$$dy-(d6.yük) = d4*po1 + d5*po2 \quad (5.18)$$

elde edilir ve $po1$ ile $po2$ 'yi aşağıdaki denklemlerle hesaplarız :

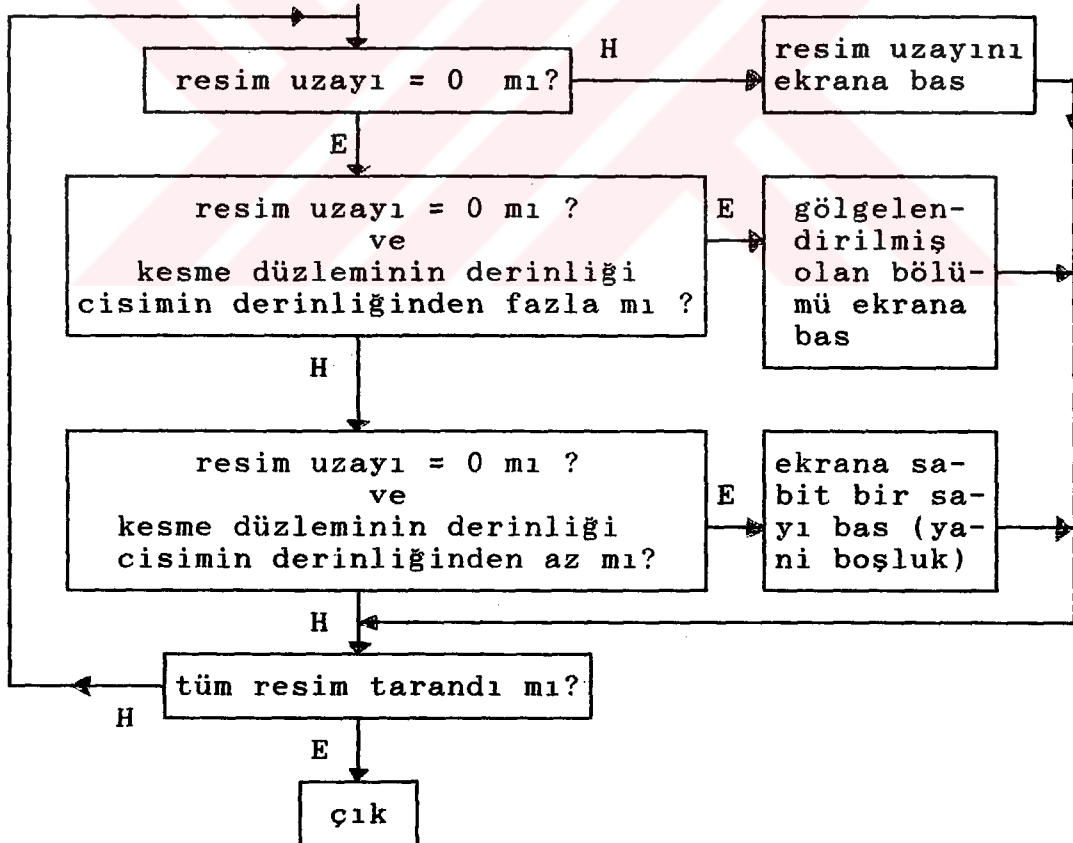
$$po1 = \frac{\begin{vmatrix} e1 & d2 \\ e2 & d5 \end{vmatrix}}{\det \begin{vmatrix} d1 & d2 \\ d4 & d5 \end{vmatrix}} \quad (5.19)$$

$$po2 = \frac{\begin{vmatrix} e1 & d1 \\ e2 & d4 \end{vmatrix}}{\det \begin{vmatrix} d1 & d2 \\ d4 & d5 \end{vmatrix}} \quad (5.20)$$

$$e1 = dx - d3*yük$$

$$e2 = dy - d6*yük$$

po1 ve po2'de bulununca 5.13'de yerlerine koyularak dz değişkeni bulunur. En son olarakta denklem 5.6'dan ex, ey ve ez cisim uzay koordinatları ortaya çıkarılır ve bu ortaya çıkan koordinatlar resim uzayına; dz derinlik bilgiside maskeleme uzayına atanır. Bundan sonra dz derinlik bilgisinide kullanarak kesme yüzeyinin altında kalan kısmın basılması, üstte kalan kısmının basılmamasını sağlayan kriterlerin tesitine gelir. Bu işlem şu şekilde yapılır.



Şekil 5.8 Kesme düzleminin üzerinde kalan bölümün basıl-mamasını sağlayan algoritmanın akış şeması.

Görüldüğü gibi bu yöntemde çok fazla matematiksel işlem kullanılmakta ve işlem uzamaktadır. Fakat istenilen yerden kesme yapmak inceleme yapan uzmana çok bilgi verir. Bu yöntemde kesilen yüzey bilgiside 2B olarak, 3B görüntünün yanında verilmekte ve böylece bu kesitinde incelenme imkanı sunulmaktadır. Bu işlemi gerçekleştiren program EK A.9'da verilmiştir. Bu program aslında geliştirilen temel 3B görüntüleme programı üzerine yazılmıştır ve bu yüzden sadece kesme işlemini gerçekleştiren bölümü verilmiştir.

Program en başta verilen döndürme değerlerine göre elde edilmiş 3B görüntü üzerinde kesmeler yapar. Eğer tekrar döndürme istenirse, başa dönülüp açılar yeniden verilmeli ve yeni dönme işleminin sonuçları beklenmelidir. Bölüm 5.8'de anlatılacağı üzere bu kesme işlemleri tezdeki hızlı çalışan programlar (yüzey elde etme yöntemi ile hızlı döndürme) çalışmaz çünkü bu programların kullandığı veriler yüzey verileridir ve hacim bilgisi dolayısı ile kesme yapılacak bilgiyi de içermemektedirler.

5.5. SEGMENTASYON

Segmentasyon ile ilgili yöntemler bölüm 4'de açıklanmıştı. Burada bahsedilen yöntemler tezde denenmiş ve başarılı sonuçlar elde edilmiştir. Geliştirilen 3B görüntü programında (EK A.1) eşik değeri kullanılarak segmentasyon yapılmaktadır. Bu yöntemleri daha ayrıntılı olarak açıklamadan önce, gerek 2B, gerekse 3B için önemli olan "pencereleme" konusu anlatılacaktır. Bu konuda segmentasyon işlemlerine dahil edilebilir.

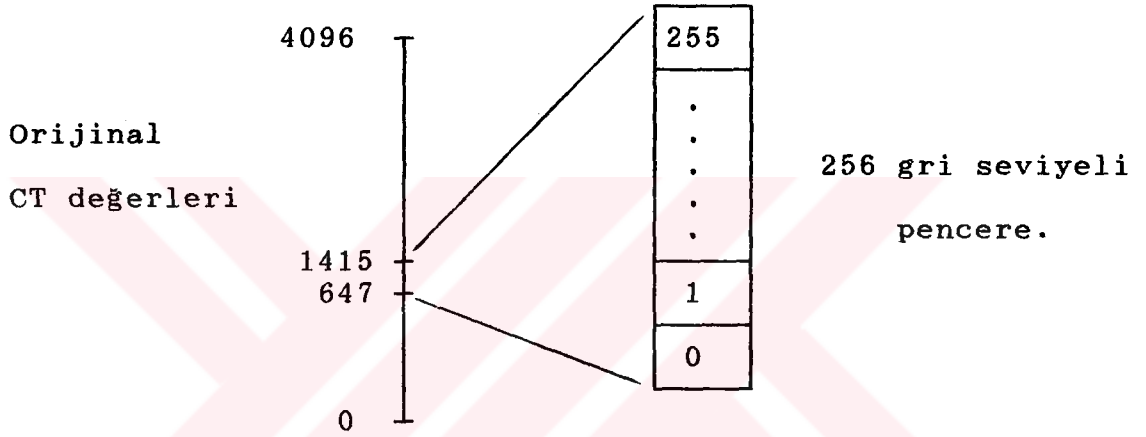
5.5.1. Pencereleme

İnsan gözü siyah-beyaz bir ekran üzerinde sadece 15 ile 20 arasında gri seviyesini seçebilir. Bu yüzden ekranda görüntüleme ile ilgili tüm detayları aynı anda vermeye gerek yoktur. İncelenmek istenilen bölgelere ait gri seviyeleri pencereleme yöntemi ile seçilip, gri seviye farkları artırılarak daha iyi görüntülenmeleri sağlanır. Tablo 5.1'de CT'de kullanılan zayıflatma katsayısının dağılımını göstermektedir.

Tablo 5.1 CT zayıflama değerlerinin dağılımı.
(HU : Hounsfield Unit, CT zayıflatma katsayısı)

HU	Maddenin Cinsi
-1000	hava
0	su
+3071	çok yoğun kemik

Tablo 5.1'deki gri seviyesi sayısı yaklaşık 4000 kadardır ve bu seviyelerin hepsini birden ekranda görüntülemenin hem teknik bakımdan imkanı yoktur (Sun Sparc 1+ aynı anda 256 gri seviyesini ekranda gösterebilir), hemde önceden açıklanan nedenlerden dolayı gereği yoktur. Bu yüzden şekil 5.9'da verilen pencereleme yöntemi kullanılır.

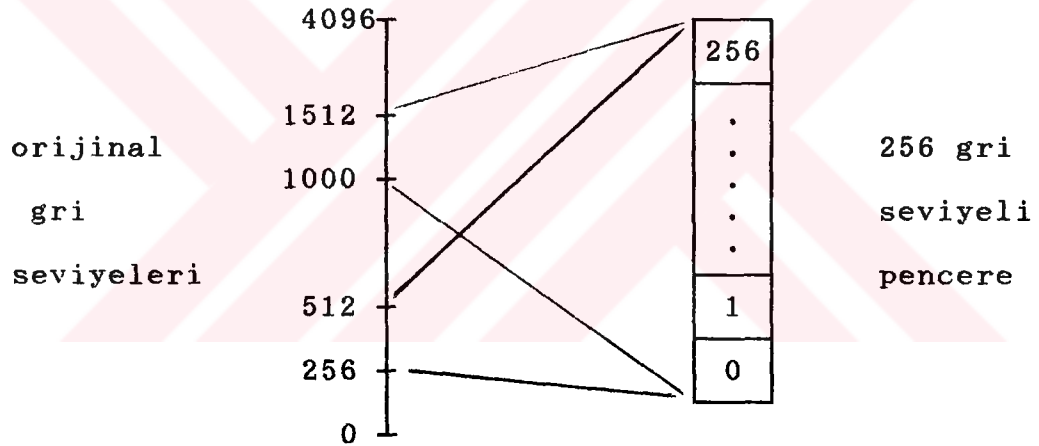


Şekil 5.9 4096 gri seviyesinin içinden bir bölümün tek pencere yöntemi ile seçilmesi.

Şekil 5.9'da görüldüğü gibi 647,1415 gri seviye aralığında 768 adet gri seviyesi vardır. Bu sayıyı 256'ya indirgemek için her 3 gri seviyesine tek gri seviyesi atanır ve bu aralıktaki gri seviyeleri 256'ya indirgendinden görüntüleme yapmak mümkün olur. Görüldüğü gibi belli bir veri kaybı oluşmaktadır. Bunun için en ideali tam 256 gri seviyesine karşılık düşen bir pencere seçmektir (1000-1255 aralığı gibi). 4096 gri seviyesinin

hepsini tek pencerede görüntülemeye kalkışırsak, her 16 gri seviyesine pencerede 1 gri seviyesi düşerki buda dokular arası gri seviye farklarının yokolarak segmentasyon işleminin sonuçlarının olumsuz etkilenmesine yol açar. Böylece etkili segmentasyon yapabilmek için pencereleme işleminin önemi anlaşılmaktadır.

Pencereleme işleminde ayrı dokuları aynı görüntü içinde göstermek için iki ayrı pencere açılabilir (Şekil 5.10).



Şekil 5.10 İki pencere yöntemi.

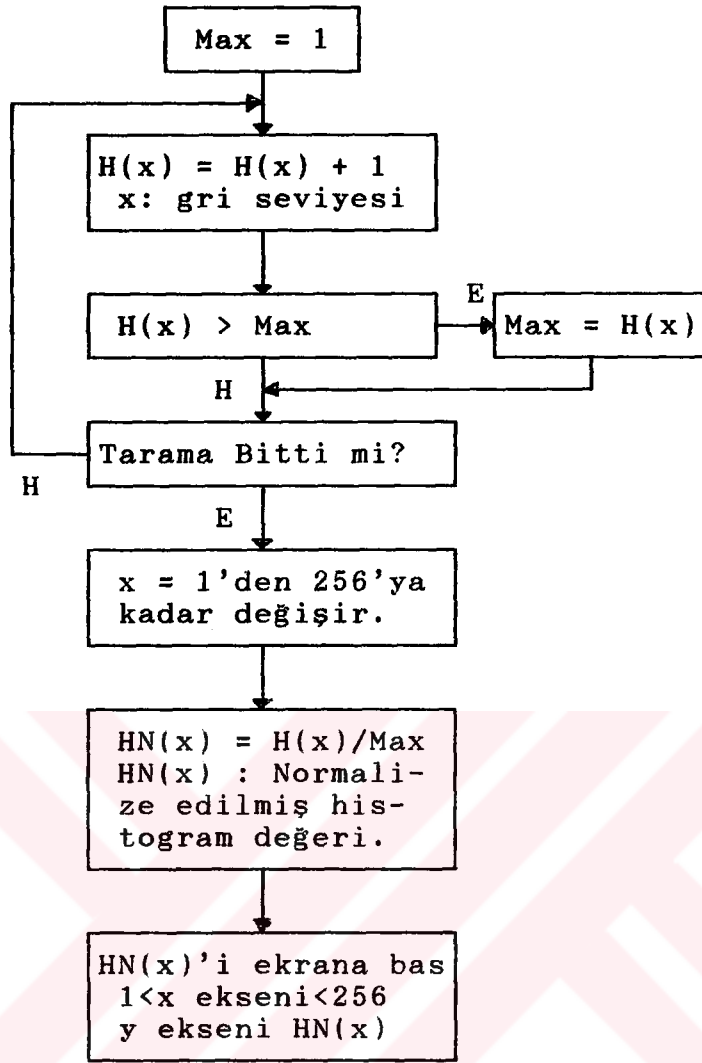
Tanımlanan bir pencere içinde bir pencere daha açmak mümkündür. Bu yöntem geniş bir aralıkta inceleme yaparken, bu aralığın daha küçük bir parçası incelenmek istenildiğinde kullanılır. Tezde kullanılan algoritmalarda segmentasyon için tek pencere yöntemi kullanılmıştır.

5.5.2. Segmentasyon

Segmentasyon işlemi algoritma yapısı bakımından basit bir işlemdir. Asıl önemli olan incelenmek istenen yapıların özelliklerini ortaya çıkarmaktır. Bu özellikleri saptamak için çeşitli yöntemler geliştirilmiştir. 2B görüntü analizi istenilen özellikleri saptamakta büyük önem taşımaktadır. Bunlardan birisi histogram analizidir. Histogram analizi bölüm 4'de açıklanmıştır. Burada bahsedilecek olan algoritmalarda bölüm 4'de anlatılanları temel almaktadır ve bölüm 4'de anlatılan algoritmalardan bazıları bu tezde geliştirilmiştir (tek ve çift eşik yöntemleri, örnek alma yöntemi - fakat bu program olarak yazılmamıştır.-)

5.5.2.a. Histogram Analizi :

Görüntünün histogramını çıkartmak için Şekil 5.11'deki akış şeması temel alınarak yazılan program kullanılır. Elimizde bulunan medikal verilerde histogram incelemesi yapıldığında verilerin 2300 - 4096 arasında sıfır değerini aldığı görülmüştür. Bunun anlamı o bölge içinde veri yok demektir. EK A.1'deki programda kullanılan segmentasyon değerleri histogram analizi ve 5.5.2.2'de açıklanacak olan gri seviye analizi kullanılarak elde edilmiştir. Histogram değerlerini ekrana bastırmak için en büyük değer bulunmalı ve bu değer alınarak bütün değerler normalize edilmelidir.



Şekil 5.11 Histogram analiz programının akış şeması.

Histogram işleminin bir kusuru, kesitlerde az yer kaplayan doku yapılarına ait olan gri seviyelerinin grafikte gözükmemesidir. Bunun sebebi baskın olan dokuların histogram tepeleri, Max değeri oluşturur ve histogram değeri az olan veriler grafikte kaybolur.

5.5.2.b. Gri Seviye Analizi :

Bu işlemde yapılan, klavye kontrollü bir noktanın 2B görüntüler üzerinde gezdirilmesi ve bu esnada noktanın bulunduğu x, y koordinatlarının ve buradaki gri seviyesinin ekranda yazılmasıdır. Bu şekilde bir analizin yararı, istenilen dokuların gri seviyelerinin aramaya gerek kalmadan bulunmasıdır. Kolayca istediğimiz dokunun üstüne noktamızı götürerek gri seviyesini öğreniriz. Bu şekilde çok küçük yer kaplayan ve histogramda gözükmeyen dokular ayırd edilebilir.

Gri seviye analizi programını geliştirmenin asıl amacı orijinal görüntülerimizdeki bazı istenmeyen verileri yoketmektir. Elimizde bulunan medikal verilerdeki hasta beyininde tümör olduğundan ışın tedavisi görmektedir. Işınları her seferinde doğru yere ulaşması içinde hastanın başı tedavi sürecinde sabit kalmalıdır. Bunun için hastanın başında bir sabitleştirme kaskı ve bu kaska ait vidalar bulunmaktadır. Bu vidalar kesitlerde bozukluk yaratmakta ve 3B görüntüler oluşturulduğunda hastanın kafasının etrafında görülmekte ve sağlıklı bir incelemeyi mümkün kılmamaktadırlar. 2B kesitlerde yapılan incelemelerde bu çubuklar yanlarda kaldıklarından hiçbir etkileri olmamaktadır.

Gri seviye analizi ile 2B görüntülerde çubuklara

ait olan gri seviyeleri yaklaşık olarak 1280 ile 1390 arasında bulunmuş fakat daha önce açıklandığı gibi çubuklara ait kenarlarda süreksizlikler olduğundan kenar gri seviyeleri 1000 gri seviyesi civarında bulunmuştur. Eğer 1000 - 1390 arasındaki gri seviyelerine ait yapıları ekranda göstermez isek çubuklarla beraber kafatası kemik yapısına ait verilerde yokolur. Buda bizim istediğimiz sonuç değildir. Gri seviye analizinin yararı burada görülmektedir. Özel dokuların gri seviyeleri kolayca tespit edilir.

5.5.2.c. Eşik Seviyeli Segmentasyon :

Bölüm 4'de iki eşik değerli segmentasyon için verilmiş olan formülü tekrar yazalım . Tezdeki programlarda bu segmentasyon algoritması kullanılmaktadır.

$$S(x,y,z) = \begin{cases} 1 , & \text{eşik1} < a(x,y,z) < \text{eşik2} \\ 0 , & \text{eşik1} > a(x,y,z) \text{ veya } a(x,y,z) > \text{eşik2} \end{cases} \quad (5.21)$$

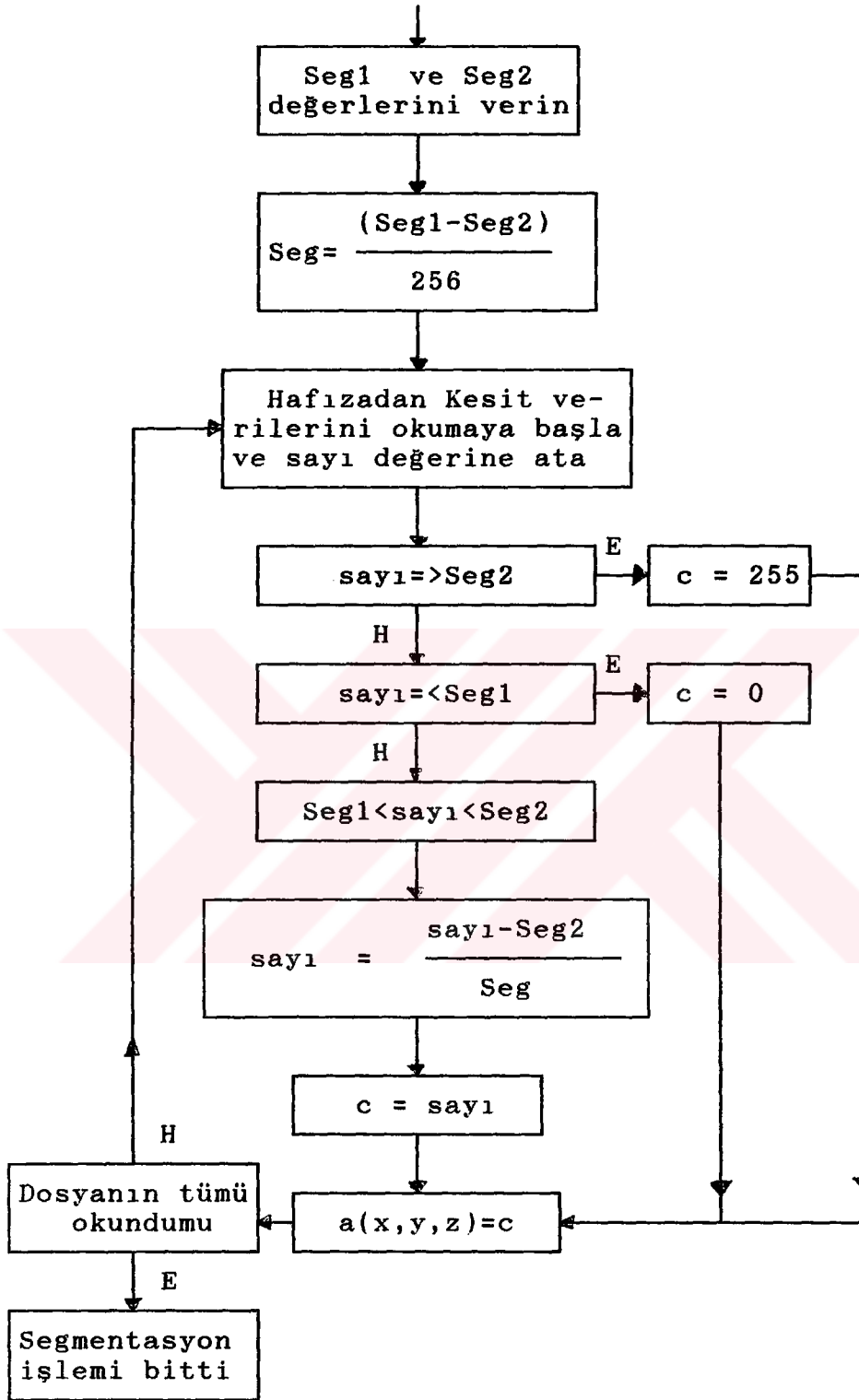
Bu formülde $S(x,y,z)$ segmentasyon ile edilen verilerin bulunduğu matristir. $a(x,y,z)$ ise tüm 3B cisim matrisini temsil eder. Sonucun 1 olması o dokuların ekrana basılacağı anlamındadır ve veriler görüntü matrisine atanırlar. Sonuç 0 ise elimine edilen verilere aittir ve algoritma işlem yapmadan bir sonraki elemanı tarar. Eşik1 ve Eşik2 değerleri programın başında sabit olarak tanımlanacağı gibi, kullanıcı tarafındanda belirlenebilmektedir. Programlarda kullanılan bazı eşik değerleri şöyle verilebilir

Tablo 5.2. 2B görüntü analizi ile elde edilen ve segmentasyonda kullanılacak eşik değerleri.

Eşik Değerleri	Ait Olan Doku	(*)
750 - 1415	yüz derisi	2.59
1500 - 2080	kemik yapısı	2.26
1000 - 1390	çubuklar	1.52

(*) : Yazılı olan sayılar, görüntüdeki 1 adet gri seviyesine, orijinal kesitlerdeki kaç adet gri seviyesinin karşılık düştüğünü belirtir.

Şekil 5.12'de Segmentasyon işleminin akış şeması verilmiştir. Bu işlem Dönme ve Kesme işlemlerinin tanımlandığı şekil 5.7'da yazılmış, ancak fazla açıklanmamıştır. Şekil 5.12'de verilen akış diagramında, kullanıcı segmentasyon aralığını (pencereyi) kendi tanımlamakta ve program bu aralığa uygun olacak şekilde gri seviyelerini 256'ya göre ayarlamaktadır.



Şekil 5.12 Eşik Seviyeli Segmentasyon işlemi.

5.6 GÖLGELENDİRME

5.6.1. Işık Kaynağı

2B ekranda, 3B görüntüler gösterebilmek için insan gözünü aldatmak gerekir. Bu işlem ekrandaki görüntü üzerine gölgelendirme ve derinlik bilgisi katılarak yapılır. Tezdeki programlarda bölüm 4.4.3.a'da açıklanan ışık modellerinden yayılan ışık modeli kullanılmıştır. Işık kaynağının sonsuzda olduğu kabul edilmiş ve bu sayede bazı matematiksel zorluklardan kurtulunmuştur. İlk yazılan programlarda ışık kaynağının uzaydaki yerini belirlemek için normal vektör değerleri kullanıcı tarafından girilecek şekilde düzenlenmiştir. Fakat cismin her dönmesinde kullanıcının bu normalleri etkin bir biçimde tespit etmesi zordur ve sonuçta istenilen ışık cisme verilemeyebilir. Bu yüzden ışık kaynağının normallerini, görüntü düzlemi normallerine eşitleyerek uyumlu bir ışık kaynağı sağlanmıştır. Böylece ışık kaynağıda gözlemci koordinat sistemi ile beraber dönmekte ve gözlemcinin gözünden çıkan bir ışık hissi vererek cisim üzerinde aydınlanma sağlamaktadır. Yönü kullanıcı tarafından ayarlanan ışık kaynağının bulunduğu koordinat sistemi, Cisim Koordinat sistemidir. Uyumlu olarak çalışan ışık kaynağı ise gözlemci koordinat sistemindedir.

5.6.2. Yüzey Normallerinin ve Ekrana Basılacak Resmin Hesaplanması

Gölgelendirme işleminde dikkate alınacak parametreler : yüzey normalleri, ışık doğrultusu ve bakış düzlemine göre derinlik bilgisidir. Program ışık açısını kullanıcıdan aldıktan (yada uyumlu olarak görüntü düzlemi açılarını aldıktan) sonra ışık vektörü ile yüzey normal vektörünü skaler olarak çarpar. Görüntü normallerini hesap etmek için görüntü koordinat sisteminden cisim koordinat sistemine dönüşüm yapılması gerekir. Döndürme işleminde görüntü koordinat sisteminden, cisim koordinat sistemine dönüşüm yapılırken elde edilen koordinatlar 3 adet iki boyutlu uzayda saklanmıştır (bakınız şekil 5.7'deki akış diagramı. EK A.1'de verilen programda bu iki boyutlu uzaylar u_1 , u_2 ve u_3 matrisleridir). Böylece yeniden koordinat dönüşümleri hesaplanmadan bu değerlerin kullanılması ile zamandan kazanç sağlanır. Dönüşüm sonucu cismin koordinatları (x,y,z) ve bu koordinatlardaki gri seviye değerleri $a(x,y,z)$ bulunur. Bölüm 4'de 4.19 ve 4.20 ile verilen denklemleri kullanarak yüzey normallerini hesaplamak mümkündür.

$$G_x = a(i+1,j,k) - a(i-1,j,k)$$

$$G_y = a(i,j+1,k) - a(i,j-1,k) \quad (5.22)$$

$$G_z = a(i,j,k+1) - a(i,j,k-1)$$

$a(i,j,k)$: i,j,k noktasındaki gri seviye değeri.

G_x, G_y, G_z : gri seviye farkları.

Yüzey normalinin bileşenleri hesaplandıktan sonra vektörel olarak yazılımı aşağıdaki gibidir:

$$\vec{G}_u = G_x \vec{i} + G_y \vec{j} + G_z \vec{k} \quad (5.23)$$

normalize edilmiş yüzey normalleri kullanmak için,

$$G = \sqrt{G_x^2 + G_y^2 + G_z^2} \quad (5.24)$$

değeri hesaplanır. Buradan normalize edilmiş yüzey normali bulunur.

$$\text{Yüzey Normali} = \vec{N}_u = \frac{\vec{G}_u}{G} \quad (5.25)$$

Işık normalini hesaplamak için kullanıcı tarafından verilen I_x, I_y, I_z değerleri (EK A.1'deki programda ol

o2 ve o3 değerleri) ele alınır. Işık normalinin vektörel yazılımı aşağıdaki gibidir :

$$\text{Işık normali} = \vec{I} = I_x \vec{i} + I_y \vec{j} + I_z \vec{k} \quad (5.26)$$

Gölgelendirme işlemi için yüzeye düşecek olan ışık miktarını hesaplamak gerekmektedir. Bunun için,

$$n = \vec{N}_u * \vec{I} \quad (5.27)$$

vektörel çarpma işlemi yapılır. Sonuçta (5.28) skaler denklemi elde edilir.

$$n = \frac{(I_x * G_x + I_y * G_y + I_z * G_z)}{G} \quad (5.28)$$

n : yüzey normali ile ışık normalinin skaler olarak çarpımıdır.

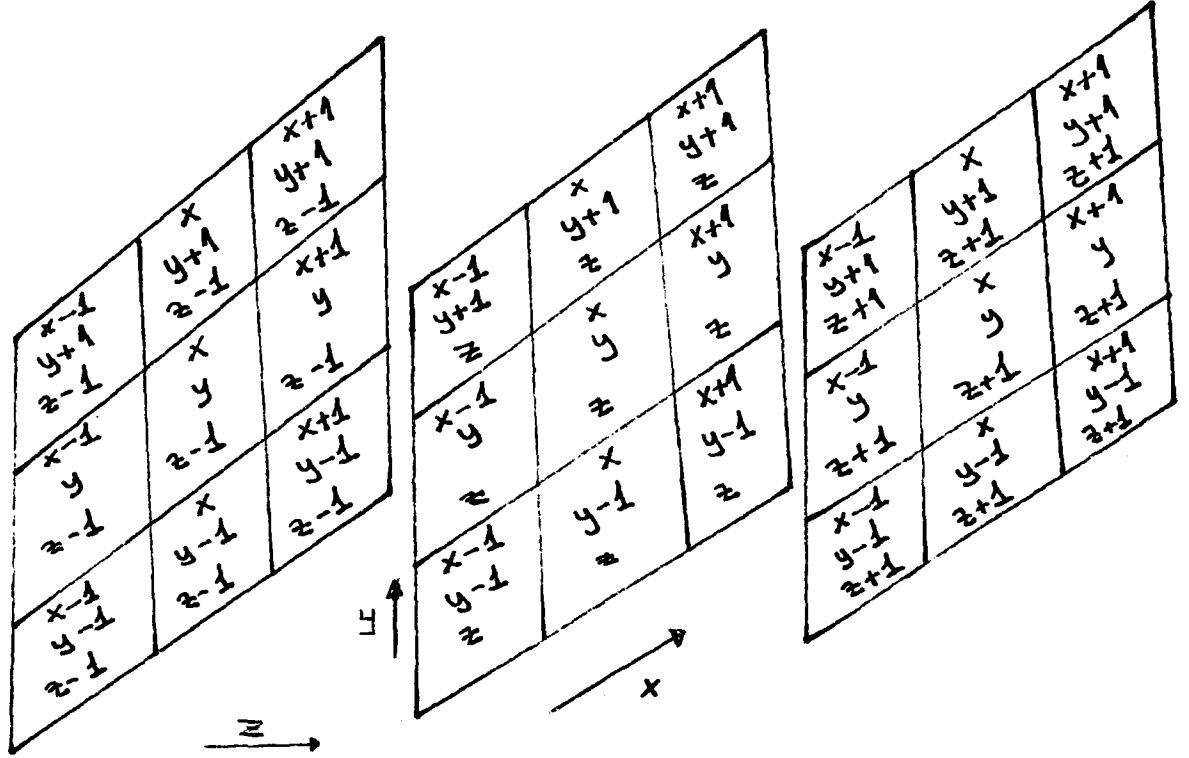
Gölgelendirme işlemi ile beraber derinlik bilgiside görüntülerde kullanılmalıdır. Derinlik bilgisi, programlarda m[x][y] matrisinde saklanır ve saklanan bilgi z' ekseninin değeridir. Bu bilgiyide gölgelendirme işleminin içine katacak olursak,

$$p = n * 0.8 + ((255 - \text{Derinlik Bilgisi}) * 0.2) \quad (5.29)$$

p : Ekrana basılacak olan resimdir. 3B, segmentasyon işlemi ve kesme işlemi uygulanmış, gölgelendirilmiş resim bilgisi tamamen bu p değerinde saklanır.

5.6.3. 26 Komşuluk Yöntemi ile Yüzey Normali Hesabı

Formül 5.22'de verilen gri seviye farkları 6 komşuluk dikkate alınarak yapılmıştır. Bu komşulukta yüzey teması önemlidir (bakınız bölüm 4, şekil 4.22). Daha iyi sonuçlar elde etmek için tezde 26 komşuluklu gölgelendirme işlemi yapılmıştır. Bu işlemde gene 6 komşulukta kullanılan 3*3*3 küp şeklindeki model kullanılır. Aşağıda verilen Şekil 5.13'de ve denklemlerde 26 komşuluğun nasıl kullanıldığı gösterilmektedir.



Şekil 5.13 26 komşuluk yöntemi için kullanılan komşuluk ilişkileri.

Şekil 5.13'de (x, y, z) için yüzey normali hesaplanacaktır. Bu işlemi gerçekleştirmek için gri seviye farkları hesaplanmalıdır :

$$g_x = (k_{20}-k_{18})+(k_{23}-k_{21})+(k_{26}-k_{24})+(k_{12}-k_{10})+(k_{14}-k_{13}) \\ +(k_{17}-k_{15})+(k_3-k_1)+(k_6-k_4)+(k_9-k_7)$$

$$g_y = (k_{26}-k_{20})+(k_{25}-k_{19})+(k_{24}-k_{18})+(k_{17}-k_{12})+(k_{16}-k_{11}) \\ +(k_{15}-k_{10})+(k_9-k_3)+(k_8-k_2)+(k_7-k_1)$$

$$g_z = (k_{18}-k_1)+(k_{19}-k_2)+(k_{20}-k_3)+(k_{21}-k_4)+(k_{22}-k_5) \\ +(k_{23}-k_6)+(k_{24}-k_7)+(k_{25}-k_8)+(k_{26}-k_9)$$

Şimdide sırası ile $k_1, k_2, \dots, k_{26}$ değerlerini yazalım :

$$k_1 = a[x-1][y-1][z-1]$$

$$k_2 = a[x][y-1][z-1]$$

$$k_3 = a[x+1][y-1][z-1]$$

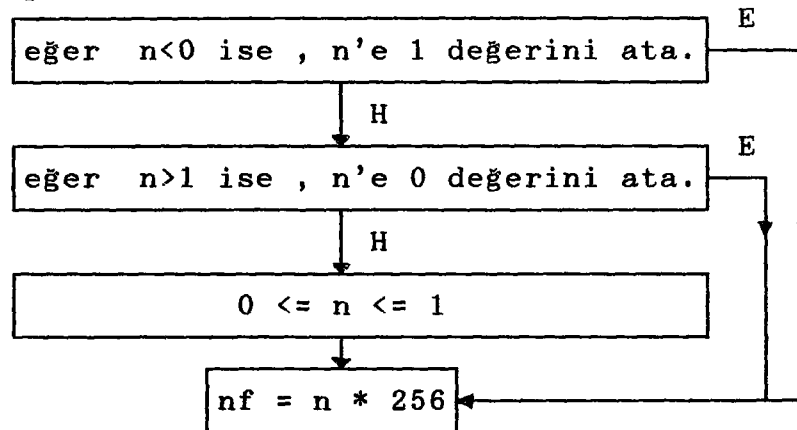
$$k_4 = a[x-1][y][z-1]$$

k5 = a[x][y][z-1]	k6 = a[x+1][y][z-1]
k7 = a[x-1][y+1][z-1]	k8 = a[x][y+1][z-1]
k9 = a[x+1][y+1][z-1]	k10 = a[x-1][y-1][z]
k11 = a[x][y-1][z]	k12 = a[x+1][y-1][z]
k13 = a[x-1][y][z]	k14 = a[x+1][y][z]
k15 = a[x-1][y+1][z]	k16 = a[x][y+1][z]
k17 = a[x+1][y+1][z]	k18 = a[x-1][y-1][z+1]
k19 = a[x][y-1][z+1]	k20 = a[x+1][y-1][z+1]
k21 = a[x-1][y][z+1]	k22 = a[x][y][z+1]
k23 = a[x+1][y][z+1]	k24 = a[x-1][y+1][z+1]
k25 = a[x][y+1][z+1]	k26 = a[x+1][y+1][z+1]

Bu değerleride verdikten sonra sıra yüzey normalinin hesabına gelir. Bu işlem,

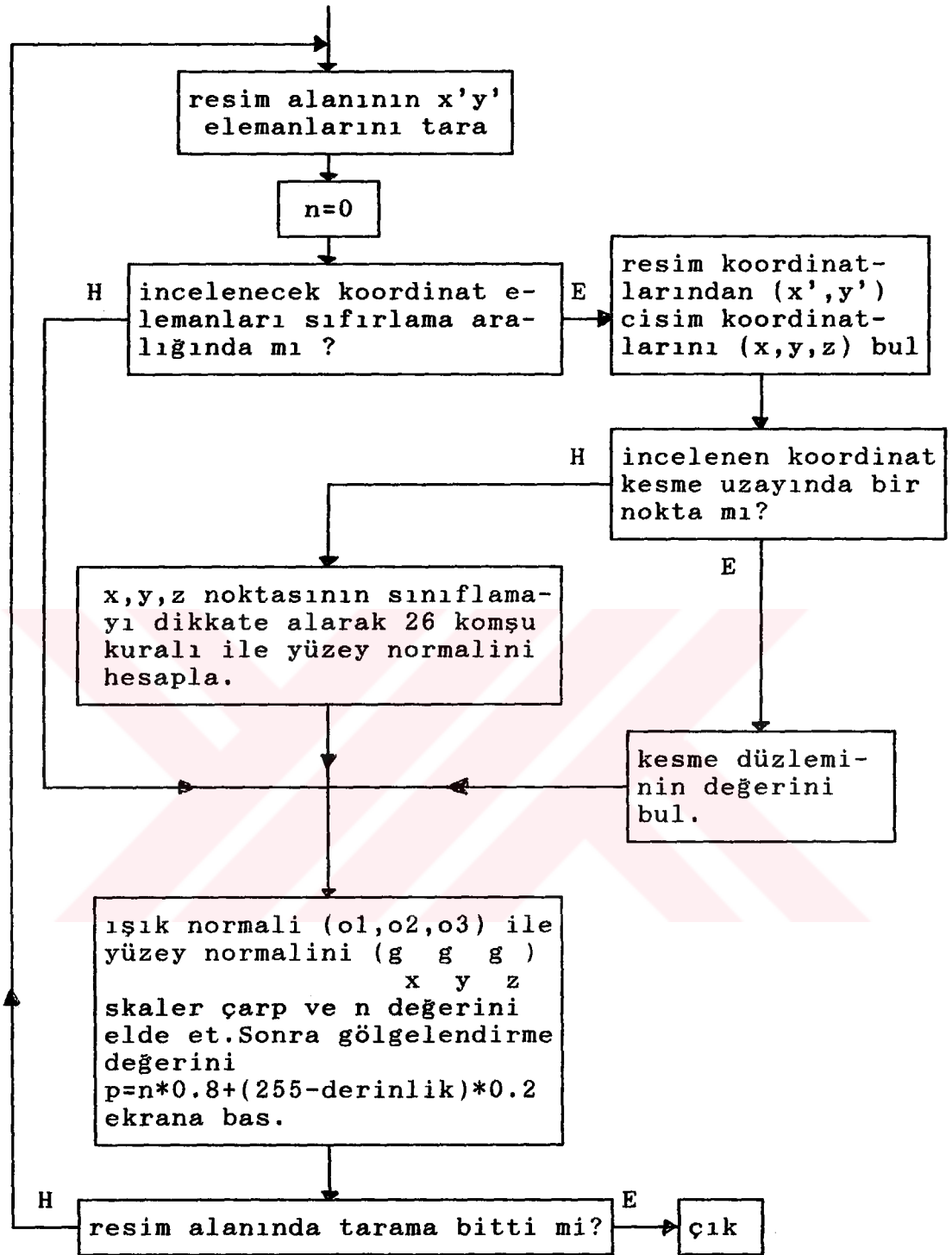
$$g = \sqrt{g_x^2 + g_y^2 + g_z^2} \quad (5.30)$$

şekilinde yazılır. Bundan sonra ışık normali ile yüzey normali (5.28) denklemi uyarınca hesap edilir. Sonuçta elde edilen değerler 1'den küçük olduğundan bu değerleri 256 değerine göre normalize etmek gerekir. Bu işlem şu şekilde yapılır :



Bu basit akış şemasında nf değeri, 256'ya göre normalize edilmiş değerlerdir. Bu denklemlerde (n) 'nin 0 olması demek, o yüzey hiç ışık almıyor demektir. Bu yüzden (n) 'nin 0 olduğu yerlere siyah (255 gri seviyesi) basılır. (n) 'nin sıfır olması için 5.28 denkleminde skaler çarpım sonucunun sıfır çıkması gerekirken bunun anlamı yüzey normali ile ışık normali arasındaki açı 90 derece veya daha fazla olması demektir. $n = 1$ değeri ise yüzey normali ile ışık normali doğrultuları aynıdır demektir ve yüzey tümü ile ışık almaktadır. Bu yüzden buraya 0 gri seviyesi yani beyaz basılır.

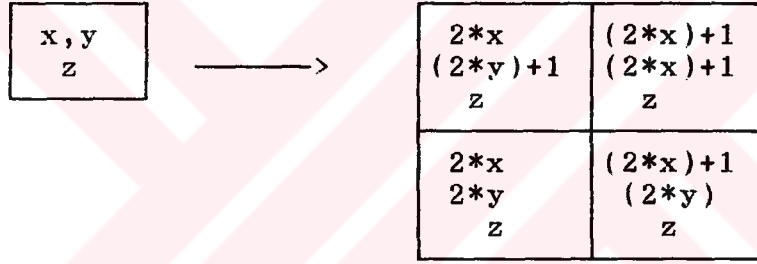
Ekrana basılacak olan asıl resmi hesaplamak için derinlik bilgisi işin içine katılır. 5.29'de verilen denklemde derinlik bilgisi 255 değerinden çıkartılmaktadır. Bunun sebebi m derinlik matrisinde derinlik bilgilerinin ters olarak saklanmış olmasıdır. En baştaki kesitte derinlik 0 olduğuna göre bu işlemi yapmak şarttır. Şekil 5.14'de gölgelendirme işleminin akış şeması verilmiştir.



Şekil 5.14 Gölgeleme işleminin akış şeması.

5.7. GÖRÜNTÜYÜ BÜYÜTME İŞLEMİ

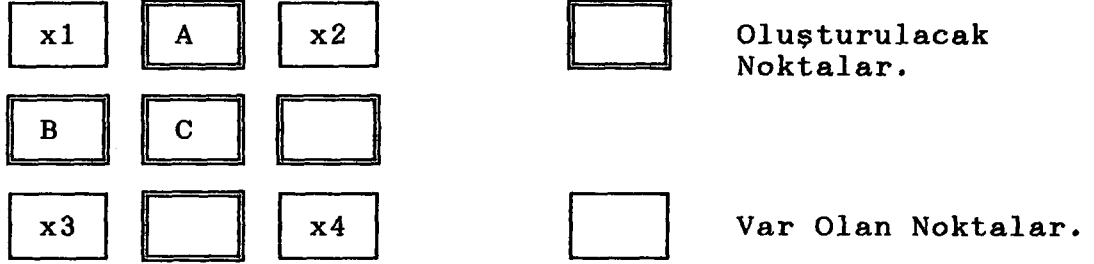
Ekrandaki görüntüyü istenilen boyutlarda büyütmek mümkündür fakat bu işlemi yaparken görüntülerde çözünürlük kaybı olur. Bu yüzden görüntüler sadece belirli bir sınıra kadar büyütülürler. Tezdeki programlarda iki ayrı büyütme işlemi yapılmıştır. Birincisinde yapılan işlem Şekil 5.15'de gösterilmiştir.



Şekil 5.15 Büyütme işlemi.

Burada kullanılan büyütme algoritmasında bir eleman 4 kere tekrarlanır ve görüntü iki kat büyümüş olur. Yanlış bir eleman tekrarlanarak 4 kere basıldığı için görüntüler iyi olmaz. Bu yüzden daha farklı büyütme işlemi uygulanmıştır. Bu büyütme işleminde aynı eleman tekrarlanmak yerine, büyütme için basılacak eleman iki komşu eleman arasında ortalama alarak belirlenir. Bu yöntem aşağıda tanımlanmıştır. :

Yeni yöntem için aşağıdaki şekli dikkate alalım :



$$A = \frac{x_1 + x_2}{2}, \quad B = \frac{x_1 + x_3}{2}, \quad C = \frac{x_1 + x_4}{2}$$

Yukarıdaki şekilde verilen büyütme işlemi ile şekil 5.14'de verilen büyütme sonuç olarak aynıdır (2 kat büyütür.) fakat ortalama alan yönteminde sonuçlar daha normal çıkmaktadır. Çünkü veriler hep tekrarlanmak yerine aralarda veri üretilmektedir. Elde bulunan görüntüleri 4 kattan daha fazla büyütmenin bir pratik yanı yoktur çünkü çözünürlük kaybı oluşmaktadır.

5.8. YÜZEY ELDE ETME YÖNTEMİ İLE HIZLI DÖNDÜRME VE ANİMASYON

Programlarımızda kullanılan görüntüler hacim elde etme yöntemleri kullanılarak elde edilmiştir. Kesme işleminin yapılabilmesi için hacim verileri kullanmak gereklidir ama fazla veri işlendiği için işlem zamanı yaklaşık 4 dakika kadar olmaktadır. Elimizdeki hacim verilerinden

sadece incelenmek istenen yapılara ait olan kenarlar elde edilerek daha az veri ile dönme işlemi daha hızlı hale getirilir. Fakat sadece kenar bilgisi olduğu için kesme işlemi yapılamaz. EK A.7'de verilen programda kenar verileri oluşturulmakta ve EK A.6'daki programda ise bu veriler ele alınarak hızlı döndürme yapılmaktadır. Kenar bilgisini 3B veriden elde etmek işlemi şu şekilde yapılır : Önce incelenmek istenen yapının segmentasyon aralığı belirlenir ve şekil 5.14'deki akış şeması ele alınarak kenar dedeksiyonu yapılır. Sonuçta elde edilen kenar dedeksiyonunda kenarlar arasında kopukluklar olabilir. Bunu gidermek için dilasyon işlemi denilen bir işlemle kenar kopuklukları giderilir. Bu işlemde şu şekilde yapılır :

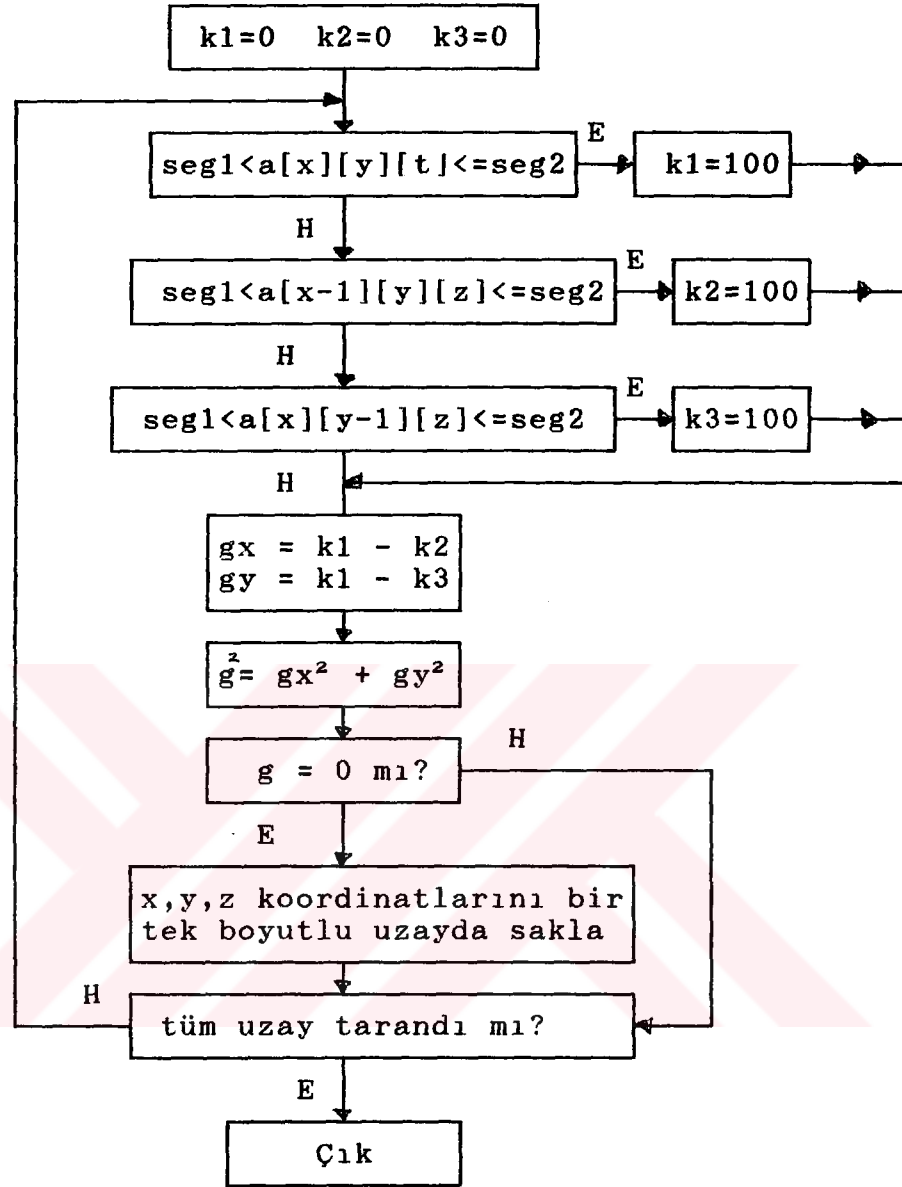
Elde bulunan kenar noktalarının (x,y,z) etrafına aynı değer (tezde kenara atanan değer 100) basılır.

x değeri için $x-1$, $x+1$ noktalarına;

y değeri için $y-1$, $y+1$ noktalarına;

z değeri için $z-1$, $z+1$ noktalarına 100 değeri basılır.

Bu işlemin sonucunda hala fazla kopukluklar varsa dilasyon işlemini bir kere daha tekrarlamak gerekir. Tezde elde edilen sonuçlarda bir kere dilasyon işlemi yapmak yeterli olmuştur. Bu şekilde elde edilen kenarlar yukarıda anlatılan dilasyon algoritması kullanılarak kopukluklar giderilir. Yüzey verileri bir dosyaya atılarak EK A.6'daki



Şekil 5.16 Yüzey elde etme yöntemi için kenar dedeksiyon algortiması.

program için veri tabanı oluşturulur.

EK A.6'daki program aslında EK A.1'de verilen programın aynı prensibi ile işlemektedir. Tek fark

retilen yzey bilgilerini kullanmasıdır. Programda dikkat edilecek olursa kesme iřlemi yukarıda aıklanan nedenlerden tr yapılmamaktadır.

Animasyon iřlemine gelince, EK A.4'deki programı ele almak gerekir. Bu programın yaptığı iř sadece sonuları arka arkaya dosyalara atmaktır. Dndrme yapılacak olan ynde aılar bir dngye sokularak her dndrme iřleminden sonra ortaya ıkan grntnn bir dosyaya atılması saęlanır. Bu dosyalar daha sonra EK A.3'de verilen program uyarınca arka arkaya ekranda gsterilir. Bu iřlem iin ekrana hızlı grnt aktarma komutları kullanılmıř ve bylece animasyon iřlemi iin gerekli hız oluřturulmuřtur.

BÖLÜM 6

SONUÇLAR VE AÇIKLAMALARI

Bu bölümde tezde EK A'da verilen programların bazılarının sonuçları açıklamaları ile beraber verilecektir. Sonuçlar Epson LQ - 400 yazıcısı ile alınmıştır.

Şekil 6.1, Şekil 6.2 ve Şekil 6.3'de görülen 3 görüntü EK A.1'de verilen program kullanılarak elde edilmiştir. Buradaki kemik görüntüsünü elde etmek için Tablo 5.2'deki 1500-2080 segmentasyon aralığı kullanılmıştır. Her bir görüntünün elde edilme süresi kesme işlemi olmadığından yaklaşık 5 dakikadır. Eğer kesme verilseydi, kesmenin derinliğine göre 3 ile 3.5 dakika arasında olacaktı. Bunun nedeni kesme olmadığı takdirde resim uzayının büyük bir alanı taraması gerektiğindendir. Kesme olduğu takdirde taranan alan azaldığından süre kısalır. Bölüm 5.5.2.b'de bahsedildiği gibi hastanın başını çevreleyen kaskın getirdiği bozukluklar şekil 6.1 ve şekil 6.3'de rahatlıkla görülebilmektedir. Özellikle şekil 6.1'de göz çukurlarının tam üzerinde beliren yarık şeklindeki bozukluk teşhis inceleme açısından büyük sakıncalar yaratmıştır. Kasktaki vidaların yapıldığı maddeden ötürü oluşan bu bozukluğu yoketmek için orijinal 3 veya 4 kesiti

yoketmek gerekir ki buda çok veri kaybına neden olur. Aslında fazla olmayan bu hata interpolasyon işlemi ile daha da artmıştır.

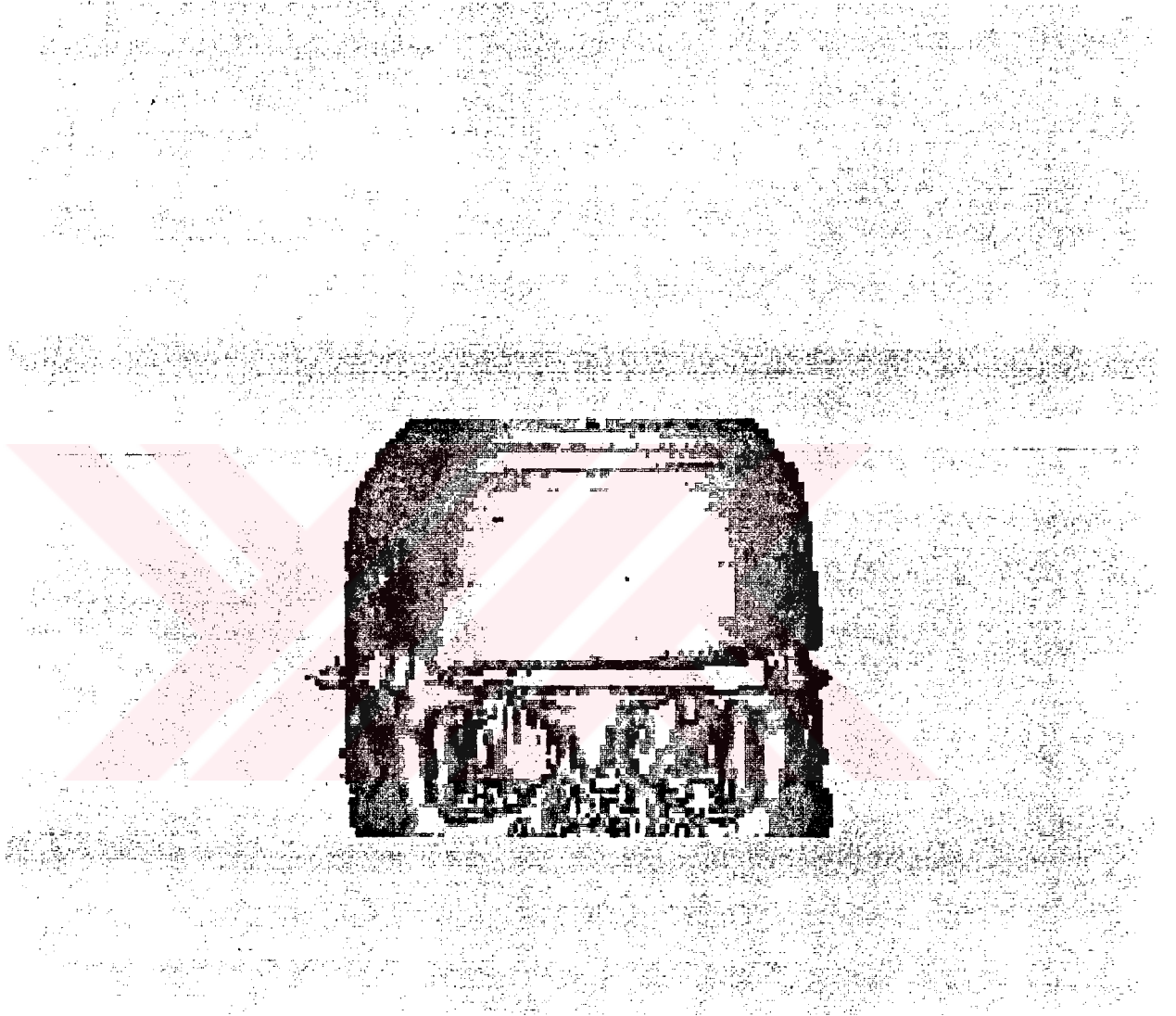
Şekil 6.1'de $\theta = 180$, $\phi = 90$ derece olarak verilmiştir. Şekil 6.2'de $\theta = 180$, $\phi = 135$ derecedir. Şekil 6.3'de ise $\theta = 40$, $\phi = 0$ derecedir.

Bölüm 5.1.3'de bahsedilen interpolasyon işleminde iki orijinal kesitten 7 adet kesit üretilmektedir. Bu şekilde ortaya çıkan görüntü Şekil 6.5'de verilmiştir. Burada görülen kafadaki uzama ise interpolasyon işleminde 7 yerine 3 kesit üretilerek giderilmiştir. Şekil 6.5'de hastanın başını çevreleyen kaska ait çubuklar ve bozuklukları oluşturan vidalar gözükmemektedir (3 kesit üretilerek yapılan işlemlere ait görüntü ise Şekil 6.7'dedir).

Şekil 6.4'deki seri halindeki 8 adet resim EK A.4 programı ile üretilmiş ve EK A.3 programı ile ekranda gösterilmiştir. Bu animasyonun elde edilmesi esnasında θ açısı 180 derece olarak alınmış ve ϕ açısı 5'er derecelik değerlerle 90 dereceden başlamak üzere değiştirilmiştir. Burada verilen şekiller arasındaki dönme açısı ise 15'er derecedir. Sırası ile ϕ 'nin değişimi 5, 15, 30, 45, 60, 90, 105, 120 derecedir. Bu görüntüler dikkat edilecek olursa yüz derisinin segmentasyon edilmemiş halini

içermektedir. Her bir resimin elde edilmesi yaklaşık 4 dakika sürmüştür. Fakat animasyon işlemini gerçekleştirirken sonuçlar 5 saniyede (25 görüntü) ekranda gösterilmektedir.

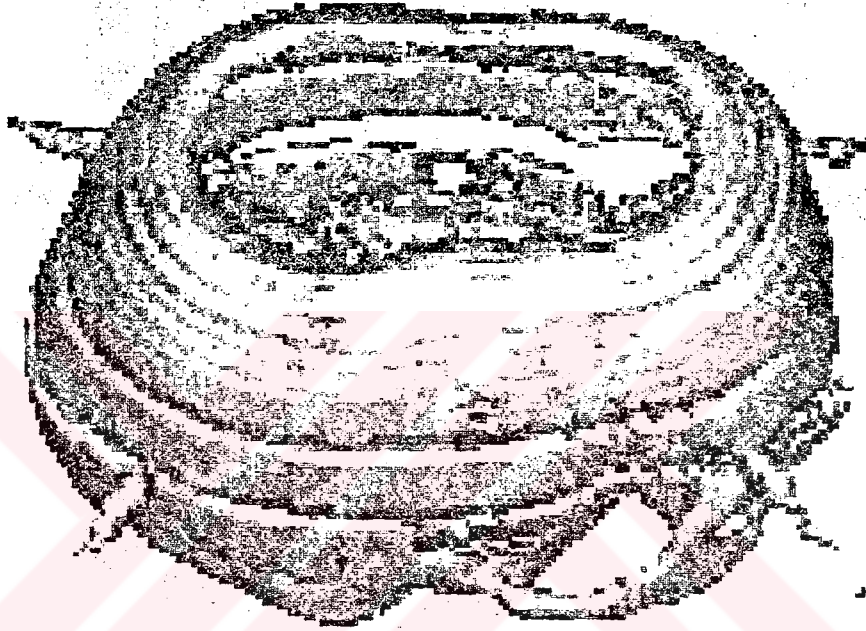
Şekil 6.6 ve Şekil 6.7'de EK A.9 ile verilen istenilen düzlemde kesme yapma programının sonuçları verilmiştir. Şekil 6.6'daki görüntü kesme yapılan düzlemde elde edilen kesitin 2B görüntüsüdür ve inceleme yapan uzmanlara kesme yapılan kesiti inceleme imkanı sunar. EK A.9'daki işleme uygun olarak yapılan kesme işlemi ise Şekil 6.7'de gösterilmiştir. Bu programda istenilen açılarla yönlendirilen kesme düzlemi yardımı ile istenilen yerden kesme yaparak, hastayı ayrıntılı inceleme imkanı vardır. Kesme için tanımlanan yeni koordinat sistemi yüzünden işlemler biraz çoğaldığından bu işlemde sonuçların elde edilmesi 5 dakika kadar sürmüştür.



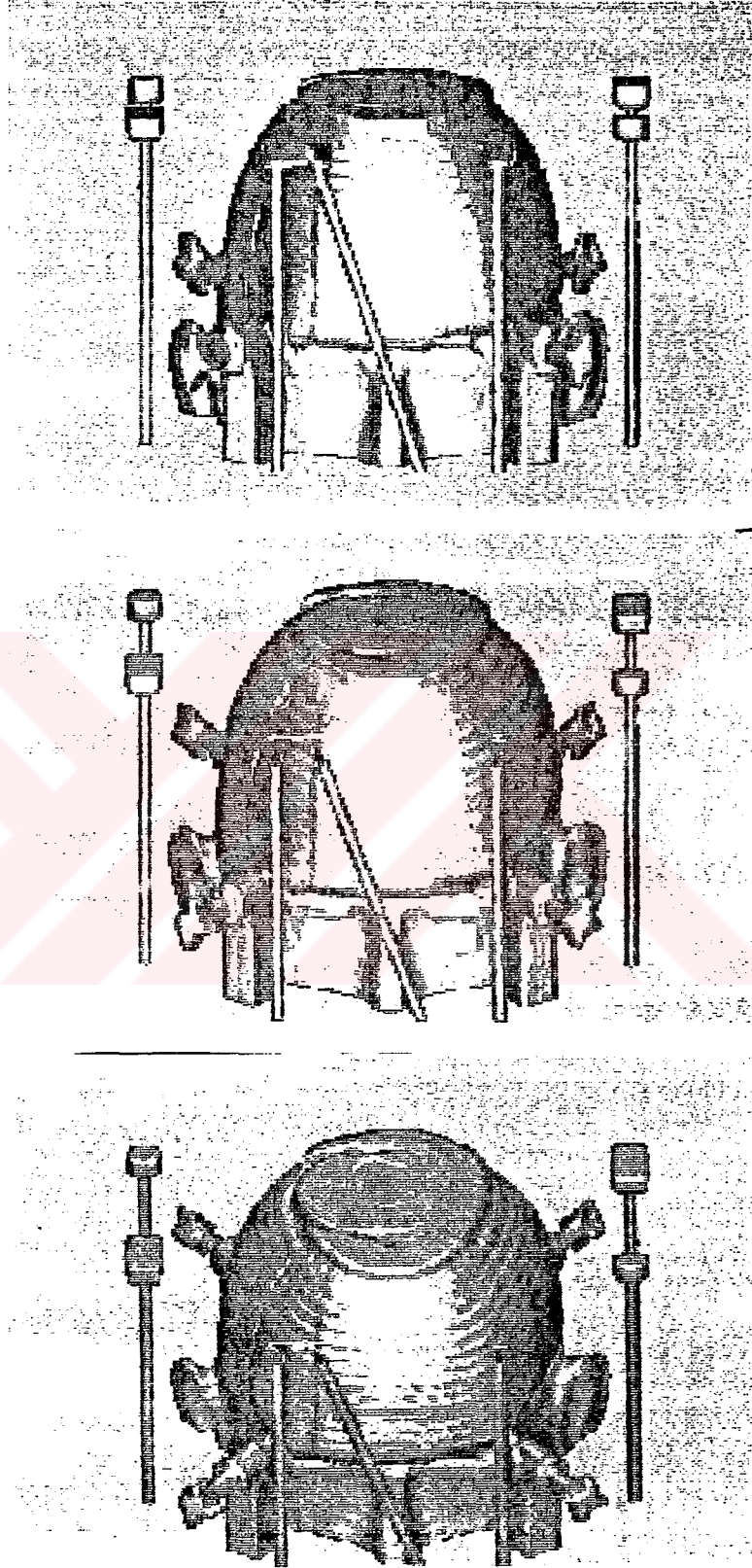
Şekil 6.1 EK A.1 programı ile elde edilen, kafanın kemik yapısının görüntüsü. $\theta = 180$, $\phi = 90$.



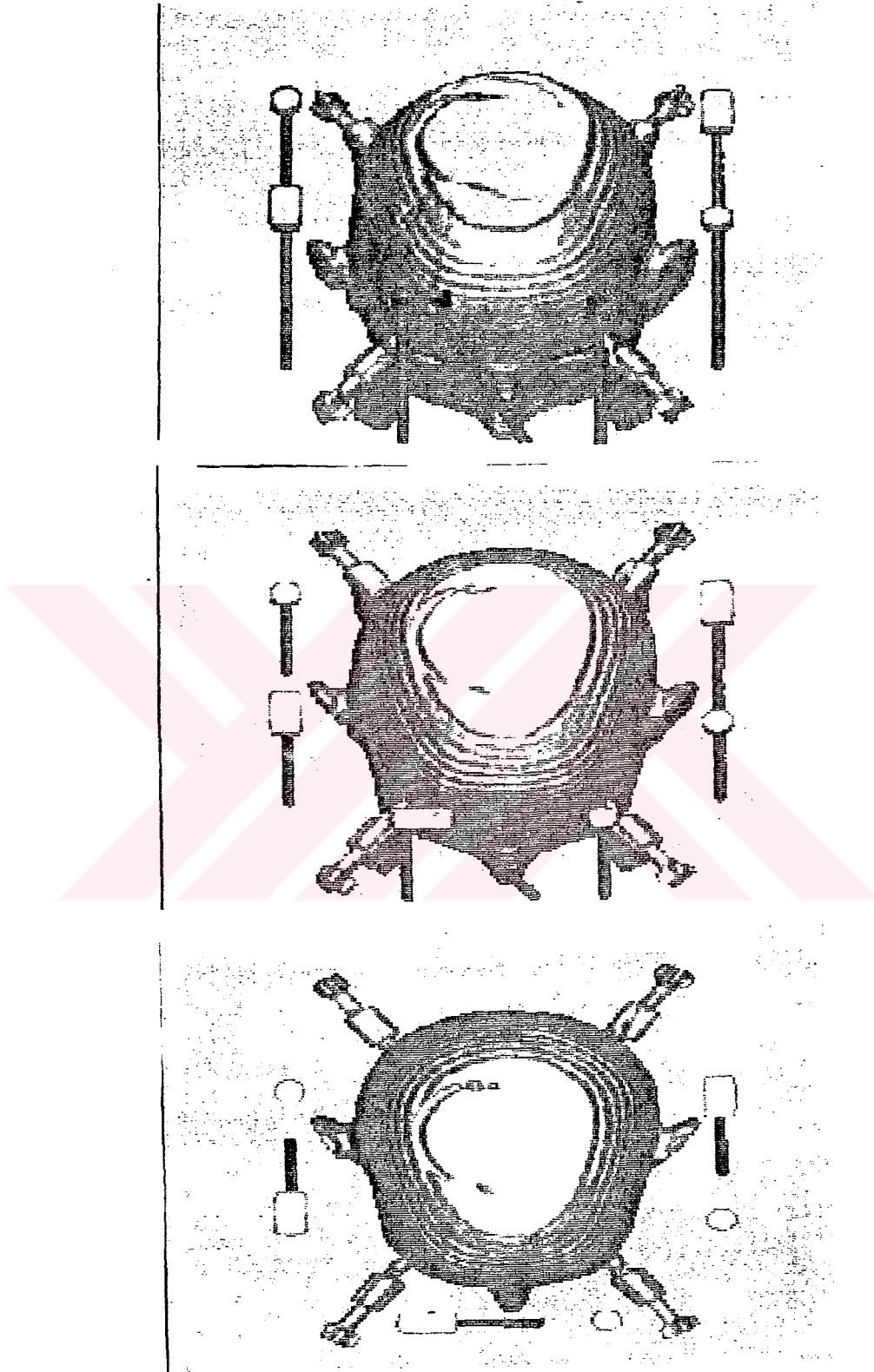
Şekil 6.2 EK A.1 programı ile elde edilen, kafanın kemik yapısının görüntüsü. $\theta = 180$, $\phi = 135$.



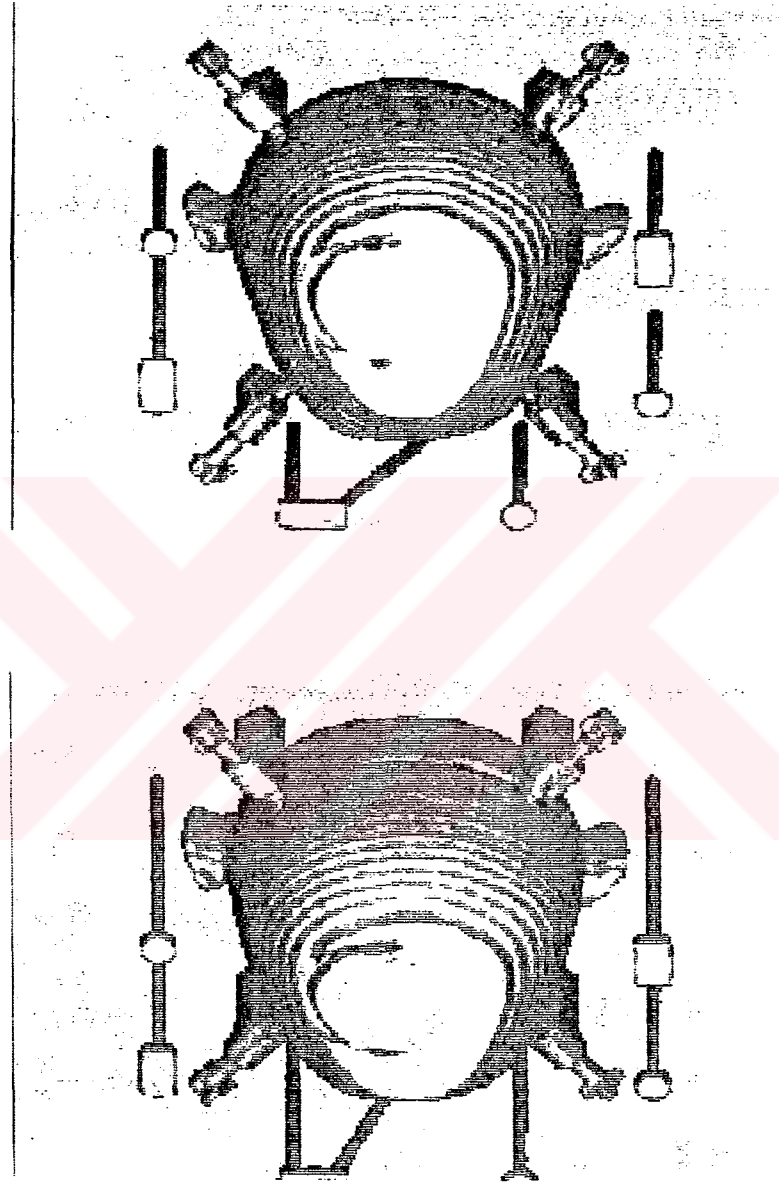
Şekil 6.3 EK A.1 programı ile elde edilen, kafanın kemik yapısının görüntüsü. $\theta = 40$, $\phi = 0$.



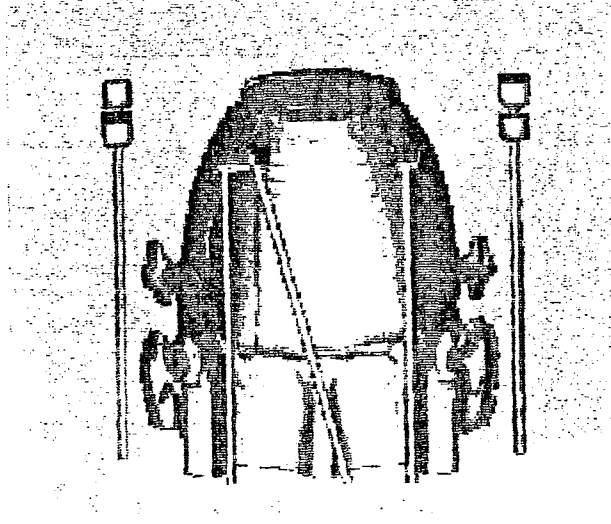
Şekil 6.4 Animasyon Görüntüleri.



Şekil 6.4 Animasyon Görüntüleri.



Şekil 6.4 Animasyon Görüntüleri.



Şekil 6.5 İnterpolasyon ile elde üretilen 7 kesit kullanılarak elde edilmiş olan verilerle yapılan görüntüleme.



Şekil 6.6 Arka sayfada verilen Şekil 6.7'dedeki kesilmiş yüzeyin tam olarak inceleme imkanı sunan 2B görüntüsü.



Şekil 6.7 EK A.9'da verilen programın sonucu. Burada kesme düzlemi için verilen yüzey açıları $\theta_1=90$, $\phi_1=40$ derecedir. Şekil 6.6 daki sonuç ile beraber bu program çok iyi analiz imkanı sağlamaktadır.

BÖLÜM 7

GELECEK ÇALIŞMALAR VE 3B GÖRÜNTÜLEMENİN GELECEĞİ

Bu bölümde, tezdeki programlar açısından gelecekte eklenebilecek olan bazı yeni yöntemlerden ve sonrada 3B görüntülemenin kısaca geleceğinden bahsedilecektir.

Tezde yazılan programlarda özellikle segmentasyon işlemi için analiz kriteri daha iyi olan algoritmalar geliştirilebilir. Bu bölümde fikir vermek açısından bazı algoritmalarından bahsedebiliriz.

Etiketleme işlemi olarak bahsedilen segmentasyon algoritması oldukça gelişmiş bir algoritmadır. Bu algoritmayı gerçekleştirmek için 2B kesitler ele alınır ve her birinin özel bir kenar dedeksiyon algoritması ile tüm kenarları dedekte edilir. Bu işlem için her kenar dedeksiyon algoritması kullanılamaz. Bu işlemi gerçekleştirmek için tezde denenen sobel kenar dedeksiyon algoritmasının sonuçları bu yöntemeye uygun değildir. Uygulanan kenar dedeksiyon algoritması sonucu elde edilen kenarlardan oluşan yapıların kapalı çevrim oluşturmaları esastır. Eğer kapalı çevrim oluşturmuyorlar ise bunları nokta nokta kullanıcının kapalı yüzey haline getirmesi gerekir. Tüm

kapalı yüzeyler elde edildikten sonra, her kapalı yüzeye bir numara verilir ve ekrana basılacak olan görüntü için sadece bu numaranın yazılması yeterlidir. Bu yöntem ile oldukça iyi sonuçlar elde edildiği çeşitli bilimsel yayınlarda açıklanmıştır. [2],[9]

Bir başka yöntemde otomatik segmantasyon yöntemi-
dir. Bu yöntem 3B cisim uzayının histogramını çıkartır ve önceden verilen tepe sayısına göre ekranda sadece bu tepelere ait olan dokuları gösterir. Örneğin başta seçilen tepe sayısı 1 ise program histogramda bulunan en büyük tepeye ait olan dokuları ekranda gösterir. Eğer tepe sayısı 3 olarak seçilmiş ise, program histogram grafiğinde en büyük 3 tepeyi seçer ve bunlara ait dokuları basar. Programın tepe seçme algoritması belirli eşik değerleri arasında çalıştırılarak daha iyi bir segmentasyon sağlanır.

Programlarda kesme düzleminin şeklide tanımlı yapılabilir. Tezde yazılan gerek bakış düzlemi kesmesi ve gerekse istenilen bir düzlemden değişik açılarla yapılan kesme işlemlerinde kesme yapan düzlem tam bir kesit almaktadır. Oysa sadece belirli bir parçanın kesilip çıkartılması mümkün değildir. Bu işlem için kesme düzleminin şekli kullanıcı tarafından değiştirilebilmelidir.

Programların bir pencereleme ortamında çalışması ile kullanıcı için büyük kolaylıklar sağlanabilir. Bu işlem için Sun Sparc 1+ 'da Sunview programı ile çalışmak gerekir. Bu programda programcılara her türlü pencereleme programı yazılması imkanı sağlanmaktadır. Böylelikle programlarda daha profesyonel hale getirilmiş olur.

Gölgelendirme işlemi içinde Phong yöntemi kullanılabilir. Ayrıca uyumlu gölgelendirme yöntemleride denerek sonuçlar bu tezde elde edilenler ve çeşitli makalelerde verilen sonuçlarla karşılaştırılabilir. Programlarda daha iyi sonuç elde etmek için hem CT hem MRI verileri aynı görüntü üzerinde kullanılabilir. Yanlız bu şekilde veri elde etmek çok zordur çünkü bir hastanın CT ile taranan bölgesi birde MRI ile taranacaktır ve bölge kesinlikle aynı olmalıdır.

3B görüntüleme tezde açıklandığı üzere tıp alanına getirdiği yeniliklerle kendine birçok uygulama alanı bulmuştur. 3B görüntüleme üzerinde bugünlerde yapılan çalışmalar; görüntü kalitesini ve veri işleme hızını arttırmak, maliyeti düşürmek ve değişik uygulama alanlarında kullanımı sağlamak gibi ana başlıklar altında sayılabilir. Görüntü kalitesini ve hızı arttırmak için iş istasyonlarının kullanılması iyi sonuç vermektedir fakat 3B görüntü programının üzerinde çalıştığı bir işistasyonu ile beraber

maliyeti pahalı olmaktadır. Her hasta, cihazda bir seans sonucunda yaklaşık 100 ile 350 dolar arası (500000 TL ile 1700000 TL arası) bir para ödemek zorundadır. Bir sistemin fiyatının 120000 dolar olduğu dikkate alınırsa günde en az 2 hasta seansı ile cihaz kendi maliyetini bir senede çıkartır. Eğer 3B görüntüleme daha fazla alanda, daha aktif olarak kullanılmaya başlarsa ve veri işleme hızı arttırılırsa maliyet sorun olmaktan çıkacaktır. Görüntü kalitesinde arttırarak, hızın artması ve kullanım alanının yaygınlaşması ile günlük seans sayısı artacak ve kendi maliyetini daha çabuk çıkartan cihazlar olarak 3B görüntüleme sistemleri ilgiyi çekerek yaygınlaşacaktır. Günümüzde yapılan çalışmalarla bu maliyetler giderek azalmaktadır. İlerde mikroişlemci teknolojisindeki gelişmelere bağlı olarak ucuzlayacak olan hızlı sistemlerle 3B görüntüleme tıp alanına daha fazla katkıda bulunacaktır.

EK A. UYGULAMA PROGRAMLARI

EK A.1 GELİŞTİRİLEN TEMEL 3B GÖRÜNTÜLEME PROGRAMI

```
/* BU PROGRAM ÜÇ BOYUTLU GÖRÜNTÜLEME İŞLEVİNİ YERİNE
GETİREN ALGORİTMALARI İÇERMEKTEDİR. BU ALGORİTMALAR
DÖNDÜRME, KESME VE GÖLGELENDİRME ALGORİTMALARIDIR.
BLOK YAPILI C LİSANI İLE YAZILMIŞ OLAN PROGRAMLARDA
HER BLOĞUN AÇIKLAMASI BLOK BAŞINDA VERİLMİŞTİR.*/
```

```
/* EKLERDE LİSTESİ VERİLMİŞ OLAN VERİ ÇOĞALTMA PROGRAMI
ÇOĞALI ; 168 ADET ÇIKIŞ DOSYASI OLUŞTURMAKTADIR. BU
DOSYALAR DIRECTORY LİSTESİNE BAKILACAK OLURSA
P000, P001, ..., P167 OLARAK GÖRÜLÜR. BURADAKİ 3B
GÖRÜNTÜLEME PROGRAMIMIZ ÖNCE BU 168 DOSYAYI
HARDDISK'TEN ALARAK RAM'A ATAR VE ORADA ÜÇ BOYUTLU
BİR DİZİ OLUŞTURUR. O YÜZDEN PROGRAM İLK ÇALIŞTIĞINDA
BİR SÜRE KADAR BU İŞLEMİ BEKLEMEK GEREKİR. PROGRAM
SONRA DÖNDÜRME AÇILARINI VE SEGMENTASYON DEĞERLERİ
İLE KESME ARALIĞININ BAŞLANGICINI SORAR. PROGRAMDA
GÖLGELENDİRME İŞLEMİ OTOMATİK OLARAK YAPILMAKTA FAKAT
İSTENİRSE KULLANICI TARAFINDAN BELİRLENMEKTEDİR.*/
```

```
/* PROGRAM IŞIK DEĞERLERİNİDE ALDIKTAN SONRA ÇOK SÜRE BİR
SÜREDE GÖLGELENDİRME İŞLEMİNİ YAPARAK SONUCU EKRANA
BASAR. BUNDAN SONRA TEKRAR KESİT BİLGİLERİNİN HARDDISK
TEN RAM'A ATILMASI İŞLEMİNİ BEKLEMeye GEREK YOKTUR.
SONUÇ YAKLAŞIK OLARAK 2.5 - 3 DAKİKA ARASINDA ORTAYA
ÇIKMAKTADIR.*/
```

```
#include <stdio.h>
#include <pixrect/pixrect_hs.h>
#include <math.h>
```

```
#define h 255
#define zzz 256
#define ccc 128
#define deg1 2.59
#define deg2 750
#define deg3 1415
unsigned char a[257][257][257];
char v[zzz][zzz];
int u1[zzz][zzz], u2[zzz][zzz], u3[zzz][zzz], m[zzz][zzz];
char isim[4];
char s[15];
unsigned char ck;
```

```

char cr,ch,c;
float a1,a2,a3,b1,b2,b3,c1,c2,c3;
float dd2,ee2,f1,f2,f3;
float dx,dy,dz,g,nf,h1,h2,pp;
float teta,fi,gx,gy,gz,deg;
float o1,o2,o3,v1,v2,v3;
unsigned int r[zzz][zzz];
int p[zzz][zzz];
int sayi,x,y,z,g1,g2,g3,yo;
int xx,yy,zz;
int n;
int l1,l2,l3;
int t,k,seg1,seg2,cli;
int dum,max,deger;
int
k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,k12,k13,k14,k15,k16,k17
,k18,k19,k20;
int k21,k22,k23,k24,k25,k26;

main()
{
    Pixrect *screen;
    FILE *fo;
    FILE *fp;
    screen=pr_open("/dev/fb");
    system("clear");
    fo=fopen("ctbak","r");

    /* KULLANILAN MATRİSİN SORUN ÇIKARTMAMASI İÇİN
       SIFIRLANMASI. EĞER BU İŞLEM YAPILMAZ İSE MATRİSLERİN
       İÇİNDE BAZI DEĞERLER KALMAKTA VE BU DEĞERLER YAPILAN
       İŞLEMLERDE VE SIFIR OLMASI GEREKEN YERLERDE BAŞKA
       DEĞERLER İÇEREREK SONUÇLARI BOZMAKTADIRLAR.*/

    for (t=0;t<=255;++t)
    {
        for (x=0;x<=255;++x)
        {
            for (y=0;y<=255;++y)
            {
                a[x][y][t]=0;
            }
        }
    }

    /* DOSYADAN DEĞERLERİN OKUNARAK İSİM ADLI DEĞİŞKEN
       DİZİSİNE ATANMASI VE BU DİZİNİNDE BİNARY DEĞER
       OKUYARAK fp POINTERINA ATANMASI */

    for (t=86;t<=170;++t)
    {
        isim[0]=isim[1]=isim[2]=isim[3]=isim[4]=0;
    }
}

```

```

        for(k=0;k<4;k++)
        {
            isim[k]=getc(fo);
        }
        cr=getc(fo);
        fp=fopen(isim,"rb");
        for (x=0;x<=255;x++)
        {
            for (y=0;y<=255;y++)
            {
                k=getw(fp);

/* EN BAŞTA BELİRLENEN DEĞERLERE GÖRE SEGMENTASYON
YAPILMASI. YUKARIDA DEFINE BÖLÜMÜNDE TANIMLANAN
DEĞ ARALIĞI 1415-750 DIR VE GRİ NOKTA ANALİZ PROGRAMI
İLE BU ARALIĞIN YUMUŞAK DOKUYU EN İYİ ŞEKİLDE TEMSİL
ETTİĞİ ORTAYA ÇIKMIŞTIR. KULLANILAN A[X][Y][Z] MATRİSİ
BİZİM ÜÇ BOYUTLU GÖRÜNTÜLEME MATRİSİMİZDİR.*/

                deg=k;
                if (deg>=deg3) ck=255;
                if (deg<=deg2) ck=0;
                if (deg>deg2 && deg<deg3)
                {
                    deg=(deg-deg2)/deg1;
                    k=deg;
                    if (k>255) k=255;
                    ck=k;
                }
                a[x][y][t]=ck;k=ck;
            }
        }
        fclose(fp);

    }
    fclose(fo);

/* KULLANILAN MATRİSLERİN SORUN ÇIKARMAMASI İÇİN
SIFIRLANMASI */

    for (x=0;x<zzz-1;++x)
    {
        for (y=0;y<zzz-1;++y)
        {
            m[x][y]=0;r[x][y]=0;p[x][y]=0;
            u1[x][y]=0;u2[x][y]=0;u3[x][y]=0;v[x][y]=0;
        }
    }

/* DÖNME , SEGMENTASYON VE KESME DEĞERLERİNİN GİRİLMESİ.
DÖNME DEĞERLERİ AÇI OLARAK ; SEGMENTASYON DEĞERLERİ GRİ
SEVİYESİ OLARAK ( 0 - 256 ARASINDA SAYILAR ) VE KESME

```

DEĞERİDE -120, 120 ARASINDA GİRİLİR. KESME DEĞERİNİN 0 OLMASI DEMEK TAM ORTADAN KESME YAPILACAK DEMEKTİR. 120 OLARAK VERİLEN DEĞER KESME OLMAYACIĞINI SÖYLER. */

Q5:

```
printf("\nTeta degerini girin: ");
gets(s);
teta=atof(s);
printf("\n Fi degerini girin: ");
gets(s);
fi=atof(s);

printf("\nSegmentasyon degerlerini girin: ");
printf("\n Birinci deger: ");
gets(s);
seg1=atoi(s);
printf("\n Ikinci deger: ");
gets(s);
seg2=atoi(s);
printf("\nKesme degerini girin (-120,120) : ");
gets(s);
cli=atoi(s);

/* UZAY DÖNÜŞÜMLERİNDE KULLANILACAK OLAN DEĞERLERİN
BAŞTAN HESAPLANMASI. BÖYLECE İLERİDE SADECE KATSAYILAR
KULLANILACAK VE DEĞERLERİN SIN, COS İŞLEMLERİNİN
YAPILMASI GEREKMEYECEKTİR. BUDA ZAMANDAN KAZANÇ
SAĞLAR.*/

teta=(teta*M_PI)/180;fi=(fi*M_PI)/180;
a1=(cos(teta));
a2=(sin(fi))*(sin(teta));
a3=(sin(teta))*(cos(fi));
b1=0;
b2=cos(fi);
b3=-sin(fi);
c1=-(sin(teta));
c2=(cos(teta))*(sin(fi));
c3=(cos(teta))*(cos(fi));
f1=-(sin(fi));
f2=(sin(teta))*cos(fi);
f3=(cos(teta))*cos(fi);

/* BU BÖLÜMDE M[X][Y] ŞEKLİNDE BİR MATRİS TANIMLANMIŞTIR
VE BU MATRİS ZAMANDAN KAZANDIRMAK AMACI İLE
KULLANILMAKTADIR. İLGİLENİLMİYEN NOKTALARA AİT OLAN
[M] MATRİSİ SIFIR OLARAK KALMAKTA VE SONRADAN
YAPILACAK OLAN ANALİZLERDE BU SIFIR DEĞERİ GÖRÜLÜNCE
ORAYLA İLGİLENİLMEMEKTEDİR.*/

zz=cli;
for (xx=-ccc+1;xx<=ccc-1;++xx)
```

```

{
  for (yy=-ccc+1;yy<=ccc-1;++yy)
  {
    dx=a1*xx+a2*yy+a3*zz;
    dy=b1*xx+b2*yy+b3*zz;
    dz=c1*xx+c2*yy+c3*zz;
    v1=dx+128.0;v2=dy+128.0;v3=dz+128.0;
    x=v1+0.5;y=v2+0.5;z=v3+0.5;v[xx+ccc][yy+ccc]
    =0;
    if (x>256 || x<0 || y>256 || y<0 || z>256 ||
    z<0 ) goto CE1;
    if (seg2>=a[x][y][z] && a[x][y][z]>seg1)
    v[xx+ccc][yy+ccc]=1;
  CE1: c=0;
  }
}

/* BU BÖLÜMDE İSE DÖNDÜRME ALGORİTMASI VE KESME
ALGORİTMASI BERABERCE UYGULANMAKTADIR. BİZİM BAKIŞ
DÜZLEMİMİZ İLE İLGİLİ OLAN DÖNÜŞÜM DEĞERLERİ DX,DY,DZ
İLE BELİRLENİR VE BAKIŞ DÜZLEMİMİZ AYNI ZAMANDA KESME
DÜZLEMİMİZDİRDE. R[X][Y] MATRİSİ İSE BİZİM EKRAM
BASILACAK OLAN RESİM UZAYIMIZDIR.*/

for (zz=cli;zz>=-ccc+1;--zz)
{
  for (xx=-ccc+1;xx<=ccc-1;++xx)
  {
    for (yy=-ccc+1;yy<=ccc-1;++yy)
    {
      if (m[xx+ccc][yy+ccc]>0) goto Q2;
      dz=a1*xx+a2*yy+a3*zz;
      dy=b1*xx+b2*yy+b3*zz;
      dx=c1*xx+c2*yy+c3*zz;
      v1=dx+128.0;v2=dy+128.0;v3=dz+128.0;
      x=v1+0.5;y=v2+0.5;z=v3+0.5;
      if (x>256 || x<0 || y>256 || y<0 || z>256
      || z<0) goto Q2;
      if (seg2>=a[x][y][z] &&
      a[x][y][z]>seg1)
      {
        m[xx+ccc][yy+ccc]=zz+ccc;
        r[xx+ccc][yy+ccc]=a[x][y][z];
        u1[xx+ccc][yy+ccc]=x;
        u2[xx+ccc][yy+ccc]=y;
        u3[xx+ccc][yy+ccc]=z;
      }
    }
  Q2: c=1;
  }
}

```

```

/* GÖLGELENDİRME ALGORİTASI BURADA YAPILMAKTADIR.
TEZDE AÇIKLANDIĞI GİBİ BURADA KULLANILAN ALGORİTMA
26 KOMŞULUKLU GRİ SEVİYE FARK ALGORİTASI DİYE
ADLANDIRILMAKTADIR. EN BAŞTA IŞIK NORMALİNİN
BİLEŞENLERİ SORULMAKTADIR. BU BİLEŞENLERİ İLGİNÇ
DÖNDÜRME AÇILARI VERİLDİĞİ ZAMAN HESAPLAMAK ZOR
OLDUĞUNDAN ; OTOMATİK OLARAK BULMAK VE AÇI NE
OLURSA OLSUN ÖNDE IŞIK GELECEK ŞEKİLDE VERMEK
GEREKİR.*/

```

```

printf("\n o1 degerini girin :");
gets(s);
o1=atof(s);
printf("\n o2 degerini girin :");
gets(s);
o2=atof(s);
printf("\n o3 degerini girin :");
gets(s);
o3=atof(s);
v1=cli;
for (xx=-ccc+1;xx<=ccc-1;++xx)
{
    for (yy=-ccc+1;yy<=ccc-1;++yy)
    {
        n=1;
        if (seg2<r[xx+ccc][yy+ccc] ||
            r[xx+ccc][yy+ccc]<=seg1) goto UU;
        x=u1[xx+ccc][yy+ccc];
        y=u2[xx+ccc][yy+ccc];
        z=u3[xx+ccc][yy+ccc];
        if (v[xx+ccc][yy+ccc]==0)
        {
            k1=0;k2=0;k3=0;k4=0;k5=0;k6;k7=0;k8=0;k9=0;k10=
            0;k11=0;k12=0;k13=0;
            k14=0;k15=0;k16=0;k17=0;k18=0;k19=0;k20=0;k21=0
            ;k22=0;k23=0;k25=0;
            k26=0;
            if (a[x-1][y-1][z-1]>=seg1 && a[x-1][y-1][z-1]<=seg2)
            k1=a[x-1][y-1][z-1];
            if (a[x-0][y-1][z-1]>=seg1 && a[x-0][y-1][z-1]<=seg2)
            k2=a[x-0][y-1][z-1];
            if (a[x+1][y-1][z-1]>=seg1 && a[x+1][y-1][z-1]<=seg2)
            k3=a[x+1][y-1][z-1];
            if (a[x-1][y-0][z-1]>=seg1 && a[x-1][y-0][z-1]<=seg2)
            k4=a[x-1][y-0][z-1];
            if (a[x-0][y-0][z-1]>=seg1 && a[x-0][y-0][z-1]<=seg2)
            k5=a[x-0][y-0][z-1];
            if (a[x+1][y-0][z-1]>=seg1 && a[x+1][y-0][z-1]<=seg2)
            k6=a[x+1][y-0][z-1];
            if (a[x-1][y+1][z-1]>=seg1 && a[x-1][y+1][z-1]<=seg2)
            k7=a[x-1][y+1][z-1];
            if (a[x-0][y+1][z-1]>=seg1 && a[x-0][y+1][z-1]<=seg2)

```

```

k8=a[x-0][y+1][z-1];
if (a[x+1][y+1][z-1]>=seg1 && a[x+1][y+1][z-1]<=seg2)
k9=a[x+1][y+1][z-1];
if (a[x-1][y-1][z-0]>=seg1 && a[x-1][y-1][z-0]<=seg2)
k10=a[x-1][y-1][z-0];
if (a[x-0][y-1][z-0]>=seg1 && a[x-0][y-1][z-0]<=seg2)
k11=a[x-0][y-1][z-0];
if (a[x+1][y-1][z-0]>=seg1 && a[x+1][y-1][z-0]<=seg2)
k12=a[x+1][y-1][z-0];
if (a[x-1][y-0][z-0]>=seg1 && a[x-1][y-0][z-0]<=seg2)
k13=a[x-1][y-0][z-0];
if (a[x+1][y-0][z-0]>=seg1 && a[x+1][y-0][z-0]<=seg2)
k14=a[x+1][y-0][z-0];
if (a[x-1][y+1][z-0]>=seg1 && a[x-1][y+1][z-0]<=seg2)
k15=a[x-1][y+1][z-0];
if (a[x-0][y+1][z-0]>=seg1 && a[x+0][y+1][z-0]<=seg2)
k16=a[x+0][y+1][z-0];
if (a[x+1][y+1][z-0]>=seg1 && a[x+1][y+1][z-0]<=seg2)
k17=a[x+1][y+1][z-0];

if (a[x-1][y-1][z+1]>=seg1 && a[x-1][y-1][z+1]<=seg2)
k18=a[x-1][y-1][z+1];
if (a[x+0][y-1][z+1]>=seg1 && a[x+0][y-1][z+1]<=seg2)
k19=a[x+0][y-1][z+1];
if (a[x+1][y-1][z+1]>=seg1 && a[x+1][y-1][z+1]<=seg2)
k20=a[x+1][y-1][z+1];
if (a[x-1][y-0][z+1]>=seg1 && a[x-1][y-0][z+1]<=seg2)
k21=a[x-1][y-0][z+1];
if (a[x-0][y-0][z+1]>=seg1 && a[x-0][y-0][z+1]<=seg2)
k22=a[x-0][y-0][z+1];
if (a[x+1][y-0][z+1]>=seg1 && a[x+1][y-0][z+1]<=seg2)
k23=a[x+1][y-0][z+1];
if (a[x-1][y+1][z+1]>=seg1 && a[x-1][y+1][z+1]<=seg2)
k24=a[x-1][y+1][z+1];
if (a[x-0][y+1][z+1]>=seg1 && a[x-0][y+1][z+1]<=seg2)
k25=a[x-0][y+1][z+1];
if (a[x+1][y+1][z+1]>=seg1 && a[x+1][y+1][z+1]<=seg2)
k26=a[x+1][y+1][z+1];

gx=k20-k18+k23-k21+k26-k24+k12-k10+k14-k13+k17-k15+k3-
k1+k6-k4+k9-k7;
gy=k26-k20+k25-k19+k24-k18+k17-k12+k16-k11+k15-k10+k9-
k3+k8-k2+k7-k1;
gz=k18-k1+k19-k2+k20-k3+k21-k4+k22-k5+k23-k6+k24-
k7+k25-k8+k26-k9;

```

/* BU BÖLÜMDE GÖLGELENDİRME İŞLEMİ İÇİN GEREKEN ; IŞIK
NORMALİ İLE YÜZEY NORMALİNİN VEKTÖREL ÇARPILMASI İŞLEMİ
YAPILMAKTADIR.*/

```

g=sqrt((gx*gx)+(gy*gy)+(gz*gz));nf=1;
if (g!=0) nf=((-a3*gx)+(-b3*gy)+(-c3*gz))/g;

```

```

        if (nf<0) nf=0;
        if (nf>1) nf=1;
        nf=nf*255;
        n=255-nf;
    }
    else
    {
        n=r[xx+ccc][yy+ccc];
    }
UU:    c=0;
        pp=(n*0.8)+((255-m[xx+ccc][yy+ccc])*0.2);
        n=pp+0.5;

/*-----
BU YAZILANLAR EK A.9'DAKİ PROGRAMLA İLGİLİDİR. EĞER İSTE-
NİLEN DÜZLEMEN KESME YAPMAK İSTENİRSE SADECE EK A.9 PROG-
RAMI BU YAZININ ALTINDAKİLER SİLİNDİKTEN SONRA OLDUĞU GİBİ
YAZILIR. SADECE BU YAZI BAŞLAMADAN ÖNCEKİ SATIRA :
z1(xx+ccc)(yy+ccc)=n İŞLEMİ YAPILIR.
-----*/

/* RESİMİN EKRANA 2 DEFA BÜYÜTÜLEREK BASILMASI.YÖNTEM-1
BU YÖNTEMDE PİXELLER ARASI BOŞLUK İKİ KATINA ÇIKARKEN
ARALARINDAKİ BOŞLUĞA GENE KENDİ DEĞERLERİ BASILIR. */

        pr_put(screen,(xx*2)+850,(yy*2)+600,n);
        pr_put(screen,(xx*2)+851,(yy*2)+600,n);
        pr_put(screen,(xx*2)+851,(yy*2)+601,n);
        pr_put(screen,(xx*2)+850,(yy*2)+601,n);
    }
}

/* RESİMİN EKRANA 2 DEFA BÜYÜTÜLEREK BASILMASI.YÖNTEM-2
YÖNTEM-2 SADECE AÇIKLAMA SATIRLARI İÇİNDEDİR VE
PROGRAMDA AKTİF DEĞİLDİR. AKTİF HALE SOKMAK İÇİN
AÇIKLAMA ÇİZGİLERİNİ KALDIRMAK YETERLİDİR. */

/* BU YÖNTEMDE PİXELLERİN ARASINDAKİ BOİLUK HER İKİ EKSEN-
DE İKİ KATINA ÇIKARKEN ARALARDAKİ BOŞLUKLARA KOMŞU
PİXELLERİN ORTALAMASI ATANMAKTADIR. */

/* for (x=0;x<512;x=x+2)
{
    for (y=0;y<512;y=y+2)
    {
        v1=x/2;v2=y/2;xx=v1;yy=v2;
        pr_put(screen,x+300,y+300,z1[xx][yy]);
        pp=(z1[xx][yy]+z1[xx+1][yy])/2;k=pp;
        pr_put(screen,x+301,y+300,k);
        pp=(z1[xx][yy]+z1[xx][yy+1])/2;k=pp;
        pr_put(screen,x+300,y+301,k);
    }
}

```

```

        pp=(z1[xx][yy]+z1[xx+1][yy+1])/2;k=pp;
        pr_put(screen,x+301,y+301,k);
    }
} */

/* PROGRAMIN AŞAĞIDA KALAN KISIMI NORMAL 3B PROGRAMINDA
KULLANILMAKTADIR. EĞER ANİMASYON İÇİN VERİ ÜRETMEK
İSTERSEK AŞAĞIDAKİ PROGRAM PARÇASI YERİNE EK A.4'DE
VERİLEN BÖLÜMÜ OLDUĞU GİBİ YAZARIZ. YANLIZCA DEĞİŞ-
KENLERİ PROGRAMIN TANIMLAMA KISIMINDA TANIMLAMAYI
UNUTMAMAK GEREKİR. */

    for (x=0;x<=zzz-1;x++)
    {
        for (y=0;y<=zzz-1;y++)
        {
            r[x][y]=0;m[x][y]=0;
            p[x][y]=0;u1[x][y]=0;u2[x][y]=0;u3[x][y]=0;
            v[x][y]=0;
        }
        ch=getchar();
        goto Q5;
        pr_close(screen);
    }

```

EK A.2 GRİ SEVİYE ATAMA PROGRAMI

```
/* SUN SPARC 1+ SİSTEMİ İLK AÇILDIĞINDA EKRANIN RENK
HARİTASI GRİ SEVİYELERİNDEN FARKLIDIR. BU YÜZDEN EKRAN
RENK HARİTASININ GRİ SEVİYELERİNE ATANMASI BU PROGRAM
İLE YAPILMAKTADIR. EKRAN AYNI ANDA 256 GRİ SEVİYESİNİ
GÖSTEREBİLMEKTEDİR. */
```

```
#include <pixrect/pixrect_hs.h>
u_char red[258],green[258],blue[258];
Pixrect *pr;
Pixrect *screen;
```

```
main ()
```

```
{
```

```
    screen=pr_open("/dev/fb");
    renk_belirle();
    pr_close(screen);
}
```

```
    renk_belirle()
```

```
    {
        int i;
        for (i=0;i<=255;i++)
        {
```

```
            /* GRİ SEVİYESİ KIRMIZI, YEŞİL VE MAVİ TEMEL
RENKLERİNİN KARIŞIMI OLDUGUNDAN HER RENGE EŞİT
DEĞERDE SAYI ATANARAK GRİ SEVİYELERİ VE TONU
SAGLANIR. */
```

```
            red[i]=green[i]=blue[i]=255-i;
        }
```

```
    /* EKRANA RENK HARİTASINI ATAMA KOMUTU */
```

```
        pr_putcolormap(screen,0,256,red,green,blue);
    }
```

EK A.3 ANİMASYON PROGRAMI

```
/* BU PROGRAM EK A.4' DE VERİLEN PROGRAM İLE OLUŞTURULAN
ARDIŞIL GÖRÜNTÜLERİ EKRANDA HIZLI OLARAK GÖSTEREREK
ANİMASYON OLUŞTURMAK İÇİN KULLANILMAKTADIR. EK A.4'DE
Kİ PROGRAMIN YAPTIĞI EK A.1' DE VERİLEN PROGRAMIN
SONUÇLARINI DOSYALARA ATARAK BU PROGRAMIN OKUYABİLECEĞİ
VERİ TABANI OLUŞTURMAKTIR. BU PROGRAMDA SUN SPARC 1+'IN
HAFİZADAN EKRANA DREKT VERİ ATAMA ÖZELLİĞİ KULLANILARAK
ANİMASYONLARIN DAHA İYİ OLMASI SAĞLANMIŞTIR.*/
```

```
#include <stdio.h>
#include <pixrect/pixrect_hs.h>

char isim[12];
char s[17];
char cr,ch,c;
unsigned char a[257][257][30];
int x,y,z;
int n;
int t,k;

main()
{
    Pixrect *pr,*tmp;
    FILE *fo;
    FILE *fp;
    pr=pr_open("/dev/fb");

    /* GÖRÜNTÜLER İÇİN HAFİZADA BİR ALAN AÇILMASINI
    SAĞLAYAN KOMUT */

    tmp=mem_create(pr->pr_size.x,pr->pr_size.y,pr-
    >pr_depth);

    /* EK A.4'DE VERİLEN PROGRAMIN ARDIŞIL VERİLERİ
    OLUŞTURUP, DOSYALADIĞI ASIL DOSYANIN ADI
    "ANDAT" 'DIR. */

    fo=fopen("andat","r");
    for (t=0;t<=25;t++)
    {
        for(k=0;k<4;k++)
        {
            isim[k]=getc(fo);
        }
        cr=getc(fo);
    }
}
```

```

        fp=fopen(isim,"r");
        for (x=0;x<=255;x++)
        {
            for (y=0;y<=255;y++)
            {
                a[x][y][t]=getc(fp);
            }
        }
        fclose(fp);
    }
    fclose(fo);
    ch='6';
    while ((ch=getchar()) != '5')
    {

```

/* GÖRÜNTÜLERİN EKRANA HIZLI OLARAK GÖNDERİLMESİ İŞLEMİ
AŞAĞIDA VERİLEN PROGRAM BÖLÜMÜ İLE VERİLMEKTEDİR. */

```

        for (t=0;t<=25 ;t++)
        {
            for (x=0;x<=255;x++)
            {
                for(y=0;y<=255;y++)
                {
                    pr_put(tmp,x,y,a[x][y][t]);
                }
            }
            pr_rop(pr,300,300,256,256,PIX_SRC,tmp,0,0);
        }
    }
    pr_close(tmp);
    pr_close(pr);
}

```

**EK A.4 ANİMASYON VERİLERİNİ OLUŞTURAN PROGRAM
(BAKINIZ EK A.1'DEKİ PROGRAM)**

```

/* BU PROGRAM EK A.1'DEKİ PROGRAMIN GÖLGELENDİRME
ALGORİTMASI BİTİMİNDE YAPILAN AÇIKLAMAYA UYGUN
OLARAK YAZILIR. TANIMLANMASI GEREKEN DEĞERLER
BURADA SADECE CİNS VE ADLARI İLE VERİLECEKTİR.
BU DEĞİŞKENLERİ EK A.1'DE TANIMLAMA BÖLÜMÜNDE
TANIMLAYINIZ.*/
/* global olarak : float fix */
/* local olarak : FILE *fy , FILE *fx */

    isim[0]=isim[1]=isim[2]=isim[3]=isim[4]=0;
    for(k=0;k<4;k++)
    {
        isim[k]=getc(fx);
    }
    cr=getc(fx);
    fy=fopen(isim,"a+");
    for(x=0;x<=255;x++)
    {
        for(y=0;y<=255;y++)
        {
            putc(r[x][y],fy);
        }
    }
    fclose(fy);
/* MATRİSLERİN SORUN ÇIKARTMAMALARI İÇİN SIFIRLANMALARI */
    for(x=0;x<=zzz-1;x++)
    {
        for(y=0;y<=zzz-1;y++)
        {
            r[x][y]=m[x][y]=0;p[x][y]=u1[x][y]=u2[x][y]=u3[x][y]=0;
            v[x][y]=0;
        }
    }

/* BURADA Fİ AÇISININ DÖNÜŞÜ DÖNGÜYE SOKULARAK HAZIRLANAN
ANİMASYON PROGRAM PARÇASI GÖRÜLMEKTEDİR. "fix" DEĞİŞKEN
HER SEFERİNDE Fİ AÇISININ BEŞ DERECE DÖNMESİNİ
SAĞLAMAKTADIR. SONUÇTA ORTAYA ÇIKAN GÖRÜNTÜLER
DOSYALARA ATANARAK SAKLANIR VE EK A.3'DE VERİLEN PROG-
RAMIN OKUYACAĞI HALE GETİRİLİR. */

fix=fix+5;
printf("cikis dosyasi=%d\n",ll);
}
    fclose(fx);
    pr_close(screen);
}

```

EK A.5 KESİT SAYISINI İNTERPOLASYON İLE ÇOĞALTAN VE DOSYALAYAN PROGRAM

```

/* BU PROGRAMDA YURTDIŞINDAN ELDE EDİLEN MEDİKAL KESİT
VERİLERİN SAYISININ AZ OLMASINDAN DOLAYI (25 KESİT)
İNTERPOLASYON İLE KESİT ÜRETİLMEKTE VE SAYI ÇOĞALTIL-
MAKTADIR (168). BU İŞLEMİN SAKINCALARI YİNE BÖLÜM 5'DE
AÇIKLANMIŞTIR. */

```

```

#include <stdio.h>
#include <pixrect/pixrect_hs.h>
#include <math.h>

```

```

void interpol();
void dosyala();
int a[513][513][10];
char b[1000];
char d[3000];
char isim[13];
char s[15];
char cr,ch,c;
int sayi,x,y,z;
int xx,yy,zz,i;
int n,z1,z2,z3,oz;
int t,k;
int dum,max,deger;
float d1,d2,d3,d4;

```

```

main()

```

```

{
    Pixrect *screen;
    FILE *fo;
    FILE *fp;
    FILE *fc;
    screen=pr_open("/dev/fb");

```

```

/* ORİJİNAL KESİTLER CTBAK ADI VERİLEN DOSYADA YAZILIDIR
VE BU DOSYALAR TEXT MODUNDADIR. DOSYALARIN BİLGİ İÇE-
RİKLERİ BÖLÜM 5'DE AÇIKLANMIŞTIR.*/

```

```

fc=fopen("ctbak","r");
i=-1;
while(fscanf(fc,"%c",&ch) !=EOF)
{
    i=i+1;
    d[i]=ch;
}
fclose(fc);

```

```

/* ORİJİNAL VERİLERDEN (KESİTLERDEN) İNTERPOLASYON İLE
   OLUŞTURULACAK OLAN YENİ VERİLER 168 ADET OLAN VE hxx
   ADI İLE VERİLEN DOSYAYA YAZILIR. hxx DOSYASININ İÇİ-
   NE BAKILACAK OLURSA P001,P002,...P167 OLARAK SADECE
   DOSYA İSİMLERİ GÖZÜKÜR. */

```

```

fo=fopen("hxx","r");
i=-1;
while(fscanf(fo,"%c",&ch) !=EOF)
{
    i=i+1;
    b[i]=ch;
}
fclose(fo);
i=0;oz=0;
for (t=0;t<=24;++t)
{
    for(k=0;k<13;k++)
    {
        isim[k]=b[i+k];
    }
    i=i+14;
    fp=fopen(isim,"r");

```

```

/* TEXT DOSYASININ OKUNMASI 1.ORİJİNAL KESİT */

```

```

while(fscanf(fp,"%d %d %x",&x,&y,&deger) != EOF)
    a[x][y][1]=deger;
fclose(fp);
for (k=0;k<13;k++)
{
    isim[k]=b[i+k];
}
fp=fopen(isim,"r");

```

```

/* TEXT DOSYASININ OKUNMASI 2.ORİJİNAL KESİT */

```

```

while(fscanf(fp,"%d %d %x",&x,&y,&deger) != EOF)
    a[x][y][9]=deger;
fclose(fp);

```

```

/* SOL TARAF İÇİN İNTERPOLASYON */

```

```

z1=1;z2=9;z3=5;interpol();
z1=1;z2=5;z3=3;interpol();
z1=1;z2=3;z3=2;interpol();
z1=3;z2=5;z3=4;interpol();

```

```

/* SAĞ TARAF İÇİN İNTERPOLASYON */

```

```

z1=5;z2=9;z3=7;interpol();
z1=5;z2=7;z3=6;interpol();

```

```

        z1=7;z2=9;z3=8;interpol();
        z1=2;dosyala();
        z1=3;dosyala();
        z1=4;dosyala();
        z1=5;dosyala();
        z1=6;dosyala();
        z1=7;dosyala();
        z1=8;dosyala();
    }
    pr_close(screen);
}

/* ÜRETİLEN KESİTLERİ DOSYALAMA ALT PROGRAMI */

void dosyala()
{
    FILE *fx;
    for (k=0;k<4;k++)
    {
        isim[k]=d[oz+k];
    }
    oz=oz+5;dum=0;

    /* CR DEN KURTULMAK İÇİN STRING SONUNA 0 EKLENDİ */

    isim[k]=0;
    fx=fopen(isim,"a+b");
    for (x=0;x<511;x=x+2)
    {
        for (y=0;y<511;y=y+2)
        {
            d1=(a[x][y][z1]+a[x+1][y][z1]+a[x][y+1][z1]+a[x+1][y+1][z1])/4;
            k=d1;
            putw(k,fx);
        }
    }
    fclose(fx);
}

/* İNTERPOLASYON İŞLEMİNİ YAPAN ALT PROGRAM. BU İŞLEMİN
MATEMATİKSLE YANI BÖLÜM 5'DE AÇIKLANMIŞTIR.*/

void interpol()
{
    for (x=1;x<511;x++)
    {
        for (y=1;y<511;y++)
        {

```

```

        d1=(a[x][y][z1]+a[x][y][z2])*(0.25);
        d2=(a[x][y-1][z1]+a[x][y+1][z2]+a[x][y-
        1][z2]+a[x][y+1][z1]+a[x+1][y][z1]+a[x+1][y][z2]+a[x
        -
        1][y][z2]+a[x+1][y][z1])*(0.35/8);
        d3=(a[x+1][y-1][z1]+a[x-1][y+1][z2]+a[x+1][y-
        1][z2]+a[x-1][y+1][z1]
        +a[x-1][y-1][z1]+a[x+1][y+1][z2]+a[x-1][y-
        1][z2]+a[x+1][y+1][z1])*(0.15/8);
        d4=d1+d2+d3;
        a[x][y][z3]=d4;
    }
}

```



EK A.6 YÜZEY ELDE ETME YÖNTEMİ İLE HIZLI DÖNDÜRME ANİMASYON PROGRAMI

/* BU PROGRAMDA EK A.7'DE VERİLEN PROGRAMIN OLUŞTURDUĞU VERİLER KULLANILMIŞTIR. BU VERİLER YÜZEY BİLGİLERİDİR VE BU PROGRAM YÜZEY BİLGİLERİNİ OKUYARAK EKRANDA GÖSTERMEKTEDİR. 3B GÖRÜNTÜLERDEN ELDE EDİLEN YÜZEY BİLGİLERİ DAHA AZ VERİYE SAHİP OLDUĞU İÇİN HIZLI DÖNDÜRME İMKANI OLMAKTADIR. FAKAT KESME İMKANI YOKTUR. BÖLÜM 5'DE AYRINTILI BİLGİ VERİLMİŞTİR. */

```
#include <stdio.h>
#include <pixrect/pixrect_hs.h>
#include <math.h>
#define h 255
#define zzz 256
#define ccc 128
#define deg1 2.59
#define deg2 750
#define deg3 1415

unsigned char a[257][257][257];
char v[zzz][zzz];
int u1[zzz][zzz],u2[zzz][zzz],u3[zzz][zzz],m[zzz][zzz];
char isim[4];
char s[15];
unsigned char ck;
char cr,ch,c;
float a1,a2,a3,b1,b2,b3,c1,c2,c3;
float dd2,ee2,f1,f2,f3;
float dx,dy,dz,g,nf,h1,h2,pp;
float teta,fi,gx,gy,gz,deg;
float o1,o2,o3,v1,v2,v3;
unsigned int r[zzz][zzz];
int p[zzz][zzz];
int sayi,x,y,z,g1,g2,g3,yo;
int xx,yy,zz;
int n;
int l1,l2,l3;
int t,k,seg1,seg2,cli;
int dum,max,deger;
int
k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,k12,k13,k14,k15,k16,k17
,k18,k19,k20;
int k21,k22,k23,k24,k25,k26;
long int kk,lk;
unsigned char b[3000000];

main()
```

```

{
    Pixrect *screen;
    FILE *fp;
    screen=pr_open("/dev/fb");
    system("clear");

    /* 3B GÖRÜNTÜ MATRİSİNİN İÇİNİN SORUN ÇIKARTMAMASI İÇİN
       SIFIRLANMASI */

    for (t=0;t<=255;++t)
    {
        for (x=0;x<=255;++x)
        {
            for (y=0;y<=255;++y)
            {
                a[x][y][t]=0;
            }
        }
    }

    fp=fopen("bakdat","r");
    kk=0;lk=2318607;
    for (kk=0;kk<lk;kk=kk+3)
    {
        x=y=z=0;
        x=getc(fp);y=getc(fp);z=getc(fp);
        a[x][y][z]=100;
        b[kk]=x;b[kk+1]=y;b[kk+2]=z;
        pr_put(screen,x+200,y+200,100);
        if (x<0 || y<0 ) printf("%d %d \n",x,y);
    }
    fclose(fp);

    /* MATRİSLERİN SORUN ÇIKARTMAMASI İÇİN SIFIRLANMASI */

    for (x=0;x<zzz-1;++x)
    {
        for (y=0;y<zzz-1;++y)
        {
            m[x][y]=0;r[x][y]=0;p[x][y]=0;
            u1[x][y]=0;u2[x][y]=0;u3[x][y]=0;v[x][y]=
            0;
        }
    }

Q5:
    printf("\nTeta degerini girin: ");
    gets(s);
    teta=atof(s);
    printf("\n Fi degerini girin: ");
    gets(s);
    fi=atof(s);

```

```

printf("\nSegmentasyon degerlerini girin: ");
printf("\n Birinci deger: ");
gets(s);
seg1=atoi(s);
printf("\n Ikinci deger: ");
gets(s);
seg2=atoi(s);
printf("\nKesme degerini girin (-81,81): ");
gets(s);
cli=atoi(s);
teta=(teta*M_PI)/180;fi=(fi*M_PI)/180;

/* UZAY DÖNÜŞÜM HESAPLARININ YAPILDIĞI BÖLÜM */

a1=(cos(teta));
a2=0;
a3=-sin(teta);
b2=cos(fi);
b1=(sin(fi))*sin(teta);
b3=(cos(teta))*sin(fi);
c2=-(sin(fi));
c1=(sin(teta))*(cos(fi));
c3=(cos(teta))*(cos(fi));
f1=-(sin(fi));
f2=(sin(teta))*cos(fi);
f3=(cos(teta))*cos(fi);
lk=kk;
kk=0;
for (kk=0;kk<lk;kk=kk+3)
{
    x=b[kk]-128;y=b[kk+1]-128;z=b[kk+2]-128;
    dx=a1*x+a2*y+a3*z+ccc;
    dy=b1*x+b2*y+b3*z+ccc;
    dz=c1*x+c2*y+c3*z+ccc;
    xx=dx+0.5;yy=dy+0.5;zz=dz+0.5;
    if (zz>m[xx][yy])
    {
        m[xx][yy]=zz;
        r[xx][yy]=a[x+ccc][y+ccc][z+ccc];
        u1[xx][yy]=x+ccc;
        u2[xx][yy]=y+ccc;
        u3[xx][yy]=z+ccc;
    }
}

/* GÖLGELENDİRMENİN YAPILDIĞI PROGRAM BÖLÜMÜ */

printf("\n o1 degerini girin :");
gets(s);
o1=atof(s);
printf("\n o2 degerini girin :");

```

```

gets(s);
o2=atof(s);
printf("\n o3 degerini girin :");
gets(s);
o3=atof(s);
v1=cli;
for (xx=-ccc+1;xx<=ccc-1;++xx)
{
    for (yy=-ccc+1;yy<=ccc-1;++yy)
    {
        n=1;
        x=u1[xx+ccc][yy+ccc];
        y=u2[xx+ccc][yy+ccc];
        z=u3[xx+ccc][yy+ccc];
        if (x==0 || y==0 || z==0) goto HAH;
        k1=0;k2=0;k3=0;k4=0;k5=0;k6;k7=0;k8=0;k9=0;k10=
        0;k11=0;k12=0;k13=0;
        k14=0;k15=0;k16=0;k17=0;k18=0;k19=0;k20=0;k21=0
        ;k22=0;k23=0;k25=0;
        k26=0;
        if (a[x-1][y-1][z-1]>=seg1 && a[x-1][y-1][z-1]<=seg2)
        k1=a[x-1][y-1][z-1];
        if (a[x-0][y-1][z-1]>=seg1 && a[x-0][y-1][z-1]<=seg2)
        k2=a[x-0][y-1][z-1];
        if (a[x+1][y-1][z-1]>=seg1 && a[x+1][y-1][z-1]<=seg2)
        k3=a[x+1][y-1][z-1];
        if (a[x-1][y-0][z-1]>=seg1 && a[x-1][y-0][z-1]<=seg2)
        k4=a[x-1][y-0][z-1];
        if (a[x-0][y-0][z-1]>=seg1 && a[x-0][y-0][z-1]<=seg2)
        k5=a[x-0][y-0][z-1];
        if (a[x+1][y-0][z-1]>=seg1 && a[x+1][y-0][z-1]<=seg2)
        k6=a[x+1][y-0][z-1];
        if (a[x-1][y+1][z-1]>=seg1 && a[x-1][y+1][z-1]<=seg2)
        k7=a[x-1][y+1][z-1];
        if (a[x-0][y+1][z-1]>=seg1 && a[x-0][y+1][z-1]<=seg2)
        k8=a[x-0][y+1][z-1];
        if (a[x+1][y+1][z-1]>=seg1 && a[x+1][y+1][z-1]<=seg2)
        k9=a[x+1][y+1][z-1];
        if (a[x-1][y-1][z-0]>=seg1 && a[x-1][y-1][z-0]<=seg2)
        k10=a[x-1][y-1][z-0];
        if (a[x-0][y-1][z-0]>=seg1 && a[x-0][y-1][z-0]<=seg2)
        k11=a[x-0][y-1][z-0];
        if (a[x+1][y-1][z-0]>=seg1 && a[x+1][y-1][z-0]<=seg2)
        k12=a[x+1][y-1][z-0];
        if (a[x-1][y-0][z-0]>=seg1 && a[x-1][y-0][z-0]<=seg2)
        k13=a[x-1][y-0][z-0];
        if (a[x+1][y-0][z-0]>=seg1 && a[x+1][y-0][z-0]<=seg2)
        k14=a[x+1][y-0][z-0];
        if (a[x-1][y+1][z-0]>=seg1 && a[x-1][y+1][z-0]<=seg2)
        k15=a[x-1][y+1][z-0];
        if (a[x-0][y+1][z-0]>=seg1 && a[x+0][y+1][z-0]<=seg2)

```

```

k16=a[x+0][y+1][z-0];
if (a[x+1][y+1][z-0]>=seg1 && a[x+1][y+1][z-0]<=seg2)
k17=a[x+1][y+1][z-0];
if (a[x-1][y-1][z+1]>=seg1 && a[x-1][y-1][z+1]<=seg2)
k18=a[x-1][y-1][z+1];
if (a[x+0][y-1][z+1]>=seg1 && a[x+0][y-1][z+1]<=seg2)
k19=a[x+0][y-1][z+1];
if (a[x+1][y-1][z+1]>=seg1 && a[x+1][y-1][z+1]<=seg2)
k20=a[x+1][y-1][z+1];
if (a[x-1][y-0][z+1]>=seg1 && a[x-1][y-0][z+1]<=seg2)
k21=a[x-1][y-0][z+1];
if (a[x-0][y-0][z+1]>=seg1 && a[x-0][y-0][z+1]<=seg2)
k22=a[x-0][y-0][z+1];
if (a[x+1][y-0][z+1]>=seg1 && a[x+1][y-0][z+1]<=seg2)
k23=a[x+1][y-0][z+1];
if (a[x-1][y+1][z+1]>=seg1 && a[x-1][y+1][z+1]<=seg2)
k24=a[x-1][y+1][z+1];
if (a[x-0][y+1][z+1]>=seg1 && a[x-0][y+1][z+1]<=seg2)
k25=a[x-0][y+1][z+1];
if (a[x+1][y+1][z+1]>=seg1 && a[x+1][y+1][z+1]<=seg2)
k26=a[x+1][y+1][z+1];
gx=k20-k18+k23-k21+k26-k24+k12-k10+k14-k13+k17-k15+k3-
k1+k6-k4+k9-k7;
gy=k26-k20+k25-k19+k24-k18+k17-k12+k16-k11+k15-k10+k9-
k3+k8-k2+k7-k1;
gz=k18-k1+k19-k2+k20-k3+k21-k4+k22-k5+k23-k6+k24-
k7+k25-k8+k26-k9;
g=sqrt((gx*gx)+(gy*gy)+(gz*gz));nf=1;
if (g!=0) nf=((o1*gx)+(o2*gy)+(o3*gz))/g;
if (nf<0) nf=0;
if (nf>1) nf=1;
nf=nf*255;
n=255-nf;
UU:  c=0;
      pp=(n*0.8)+((255-m[xx+ccc][yy+ccc])*0.2);
      n=pp+0.5;
      pr_put(screen,xx+850,yy+600,n);
HAH: c=0;
    }
  }
  for (x=0;x<=zzz-1;x++)
  {
    for (y=0;y<=zzz-1;y++)
    {
      r[x][y]=0;m[x][y]=0;
      p[x][y]=0;u1[x][y]=0;u2[x][y]=0;u3[x][y]=0;
      v[x][y]=0;
    }
  }
  ch=getchar();
  goto Q5;
  pr_close(screen);
}

```

EK A.7 YÜZEY ELDE ETME YÖNTEMİ İLE 3B CİSİMİN YÜZEYLERİNİ OLUŞTURAN PROGRAM

```

/* BU PROGRAMDA ELİMİZDEKİ 3B VERİLERDEN SEGMENTASYON
YOLU İLE KENAR DEDEKSİYONU İŞLEMİ YAPILMAKTADIR. YÜ-
ZEYİ OLUŞTURULACAK OLAN DOKU ARALIĞI SEGMENTASYON
ARALIĞI OLARAK VERİLİR VE BU ARALIĞA DÜŞEN GRİ SE-
VİYELERİNE SABİT BİR DEĞER ATANIRKEN, DİĞERLERİNE 0
ATANIR. BU BİR CİNS KENAR DEDEKSİYONUDUR. KENARLAR
ARASI KOPUKLUK DİLATION İŞLEMİ İLE GİDERİLİRKEN SO-
NUÇLAR bakdat ADLI BİR BOYUTLU DİZİYE ATANIR. Bakdat
DOSYASI EK A.6'DA VERİLEN PROGRAMIN VERİ DOSYASIDIR.
*/

```

```

#include <stdio.h>
#include <pixrect/pixrect_hs.h>
#include <math.h>
#define h 255
#define zzz 256
#define ccc 128
#define deg1 2.793
#define deg2 750
#define deg3 1465

unsigned char a[257][257][257];
char isim[4];
char s[15];
unsigned char ck;
char cr,ch,c;
float a1,a2,a3,b1,b2,b3,c1,c2,c3;
float dd2,ee2,f1,f2,f3;
float dx,dy,dz,g,nf,h1,h2,pp;
float teta,fi,gx,gy,gz,deg;
float o1,o2,o3,v1,v2,v3;
int sayi,x,y,z,g1,g2,g3,yo;
int xx,yy,zz;
int n,e1,e2,e3;
long int hh,h1;
int l1,l2,l3;
int t,k,seg1,seg2,cli;
int dum,max,deger;
int
k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,k12,k13,k14,k15,k16,k17
,k18,k19,k20;
int k21,k22,k23,k24,k25,k26;
int b[6000000];

main()
{

```

```

Pixrect *screen;
FILE *fo;
FILE *fp;
FILE *fd;
screen=pr_open("/dev/fb");
system("clear");

/* ctbak BİZİM ORJİNAL KESİTLERİN OLDUĞU DOSYADIR. */

fo=fopen("ctbak","r");

/* KENAR DEDEKSİYONUNUN YAPILACAĞI DOKUNUN SEGMENTAS-
YON DEĞERLERİNİN GİRİLMESİ. BU DEĞERLERİN NASIL
ELDE EDİLDİĞİ BÖLÜM 5'DE AÇIKLANMIŞTIR. */

printf("\nSegmentasyon degerlerini girin: ");
printf("\n Birinci deger: ");
gets(s);
seg1=atoi(s);
printf("\n Ikinci deger: ");
gets(s);
seg2=atoi(s);

/* ORJİNAL DOSYALARIN OKUNMASI VE 3B GÖRÜNTÜ MATRİSİNE
ATANMASI İŞLEMİ. DEFINE BÖLÜMÜNDE TANIMLANAN DEG1,
DEG2 VE DEG3 DEĞERLERİ TEMEL ALINARAK 4096 GRİ SEVİ-
YESİ BU ARALIĞA İNDİRGENİR. */

for (t=44;t<=211;++t)
{
    isim[0]=isim[1]=isim[2]=isim[3]=isim[4]=0;
    for(k=0;k<4;k++)
    {
        isim[k]=getc(fo);
    }
    cr=getc(fo);
    fp=fopen(isim,"rb");
    for (x=0;x<=255;x++)
    {
        for (y=0;y<=255;y++)
        {
            k=getw(fp);
            deg=k;
            if (deg>=deg3) ck=255;
            if (deg<=deg2) ck=0;
            if (deg>deg2 && deg<deg3)
            {
                deg=(deg-deg2)/deg1;
                k=deg;
                if (k>255) k=255;
                ck=k;
            }
        }
    }
}

```

```

        a[x][y][t]=ck;k=ck;
    }
}
fclose(fp);
}
fclose(fo);

/* KENAR DEDEKSİYONU YAPILARAK KENARLARIN BULUNMASI VE
BİR DİZİYE ATANMASI */

fd=fopen("bakdat","a+b");
hh=0;
for (t=44;t<=211;t++)
{
    for (x=1;x<=255;x++)
    {
        for (y=1;y<=255;y++)
        {
            k1=k2=k3=0;
            if(seg1<a[x][y][t] && a[x][y][t]<=seg2)
                k1=100;
            if(seg1<a[x-1][y][t] && a[x-1][y][t]<=seg2) k2=100;
            if(seg1<a[x][y-1][t] && a[x][y-1][t]<=seg2) k3=100;
            gx=k1-k2;
            gy=k1-k3;
            g=sqrt((gx*gx)+(gy*gy));
            if (g!=0)
            {
                b[hh]=x;hh=hh+1;b[hh]=y;hh=hh+1;b[hh]=t;hh=hh+1;
            }
        }
    }
}
hl=hh;
n=256;

/* MATRİSİN SORUN ÇIKARMAMASI İÇİN SIFIRLANMASI */

for (t=0;t<=n-1;t++)
{
    for (x=0;x<=n-1;x++)
    {
        for (y=0;y<=n-1;y++)
        {
            a[x][y][t]=0;
        }
    }
}

```

```
/* KENAR DEDEKSİYONUNUN SONUCUNDA OLAN KENAR KOPUKLUKLARI-
   NI YOKETMEK İÇİN KULLANILAN DİLATION ALGORİTMASI */
```

```
    for (hh=0;hh<hl;hh=hh+3)
    {
        x=b[hh];y=b[hh+1];z=b[hh+2];
        for (e1=x-1;e1<=x+1;e1++)
        {
            for (e2=y-1;e2<=y+1;e2++)
            {
                for (e3=z-1;e3<=z+1;e3++)
                {
                    a[e1][e2][e3]=100;
                }
            }
        }
    }
}
```

```
/* BAKDAT'IN VERİLERİNİN OLUŞTURULMASI */
```

```
    hh=0;
    for (z=0;z<=n-1;z++)
    {
        for (x=0;x<=n-1;x++)
        {
            for (y=0;y<=n-1;y++)
            {
                if (a[x][y][z]==100)
                {
                    ch=x;putc(ch,fd);ch=y;putc(ch,fd);
                    ch=z;putc(ch,fd);
                    pr_put(screen,x+500,y+300,100);
                    hh=hh+7;
                }
            }
        }
    }
    fclose(fd);
    pr_close(screen);
}
```

**EK A.8 HISTOGRAM, HISTOGRAM DENGEME İŞLEMLERİNİ
GERÇEKLEYEN PROGRAM**

```
/* BU PROGRAMDA 2B GÖRÜNTÜLER ÜZERİNDE YAPILAN HISTOGRAM
ÇIKARTMA VE İSTENİLEN ARALIKTA HISTOGRAM DENGEMESİ
İŞLEMLERİ YAPILMAKTADIR. İLK BAŞTA İSTENİLEN ARALIK
VERİLİR VE BU ARALIKTAKİ VERİLERİN HISTOGRAMI EKRANDA
ÇİZİLİR. BU ARALIK İÇİNDE TEKRAR İSTENİLEN BİR SEGMENTASYON
ARALIĞI SEÇİLEREK HISTOGRAM DENGEMESİ YAPILIR.
*/
```

```
#include <stdio.h>
#include <pixrect/pixrect_hs.h>
#define deg1 2.59
#define deg2 1415
void sifir();
void his();
void dis();
void ciz();
void ciz1();
void norm();
void grid();
unsigned char a[300][300];
float qq[300],q1[300];
long int q[300],p[300];
float lp,max;
char isim[4];
char s[17];
char o[10];
char cr,ch,c;
int x,y,z,l1,l2;
int n;
int t,k,seg1,seg2;
float k1,l3,deg;
Pixrect *screen;

main()
{
    FILE *fo;
    FILE *fp;
    printf(" birinci degeri yaz \n");
    gets(s);
    l2=atoi(s);
    printf("ikinci degeri yaz \n");
    gets(s);
    l1=atoi(s);
    k1=l1-l2;l3=k1/256;
    screen=pr_open("/dev/fb");
    fo=fopen("ctbak","r");
```

```

for (t=0;t<=167;t++)
{
    for(k=0;k<4;k++)
    {
        isim[k]=getc(fo);
    }
    cr=getc(fo);
    fp=fopen(isim,"r");
    for (x=0;x<=255;x++)
    {
        for (y=0;y<=255;y++)
        {
            k=getw(fp);
            deg=k;

/* DOSYALARI OKURKEN SEGMENTASYON ARALIĞINA GÖRE NORMALİ-
ZASYON YAPILIR VE HİSTOGRAM HESABI İÇİN MAX DEĞER BU-
LUNUR. */

            if (deg>=11) ch=255;
            if (deg<=12) ch=0;
            if (deg>12 && deg<11)
            {
                deg=(deg-12)/13;
                k=deg;
                if (k>255) k=255;
                ch=k;
            }
            k=0;k=ch;
            a[x][y]=ch;
            if (k>0) q[k]=q[k]+1;
            if (q[k]>max) max=q[k];
        }
    }
    norm(); /* Q[X] ' i MAX 'A BÖLÜP NORMALİZE EDER. */
    ciz1(); /* RESİM HİSTOGRAMINI ÇİZİYOR */
    dis(); /* RESİMİ GÖRÜNTÜLER */
    cr=getchar();
    system("clear");
    his(); /* HİSTOGRAM DENGELEME ALT PROGRAMI*/
    ciz(); /* AÇILMIŞ HİSTOGRAM ÇİZİLİYOR */
    dis(); /* AÇILMIŞ HİSTOGRAMA GÖRE RESİM
            GÖRÜNTÜLENİR */

    sifir();
    cr=getchar();
    system("clear");
    fclose(fp);
}
fclose(fo);
pr_close(screen);
}

```

```

void ciz()
{
    pr_put(screen,200,555,200);
    pr_put(screen,200,300,200);
    for (x=0;x<=255;x++)
    {
        deg=q[x];k=0;
        k=deg;
        for (y=0;y<=p[x];y++)
        {
            pr_put(screen,k+300,555-y,250);
        }
    }
    grid();
}

```

/* EKRANDA SEGMENTASYON ARALIĞININ KOLAY OKUNABİLMESİ
AMACI İLE HISTOGRAM İLE BERABER KULLANILAN ÖLÇÜM
BİRİMİ */

```

void grid()
{
    for(x=0;x<=300;x=x+50)
    {
        pr_put(screen,x+300,560,255);
        pr_put(screen,x+300,561,255);
        pr_put(screen,x+300,562,255);
    }
}

```

```

void ciz1()
{
    for (x=0;x<=255;x++)
    {
        deg=q[x];k=0;
        k=deg;
        for (y=0;y<=k;y++)
        {
            pr_put(screen,x+300,555-y,250);
        }
    }
    grid();
}

```

```

void norm()
{
    for (x=0;x<255;x++)
    {
        q[x]=q[x]*255/max;
        p[x]=q[x];
    }
}

```

```

    }
}

void dis()
{
    printf("\n Segmentasyon ilk degeri: ");
    gets(o);seg1=atoi(o);
    printf("\n Segmentasyon ikinci degeri: ");
    gets(o);seg2=atoi(o);

    for (x=0;x<255;x++)
    {
        for (y=0;y<255;y++)
        {
            k=0;k=a[x][y];
            if (k>seg1 && k<seg2)
            {
                pr_put(screen,x+600,y+300,k);
            }
        }
    }
}

/* HISTOGRAM DENGEELEME PROGRAMI */

void his()
{
    for (x=0;x<255;x++)
    {
        lp=lp+q[x];
    }
    for (x=0;x<255;x++)
    {
        qq[x]=q[x]/lp;          /* NORMALIZE PU DİZİSİ */
    }

    q1[0]=qq[0];                /* İLK ELEMAN VMIN 'DİR */
    for (x=1;x<255;x++)
    {
        for (y=0;y<=x;y++)
        {
            q1[x]=q1[x]+qq[y] ; /* TOPLAM V[X] PU DİZİSİNDEN
                                ELDE EDİLDİ */
        }
    }

    /*İSTENİLEN HISTOGRAM */

    q1[x]=(((q1[x]-q1[0])*255)/(1-q1[0]))+0.5;
}
for (x=0;x<255;x++)
{

```

```
/* İSTENİLEN HISTOGRAM DENGELMESİ SAĞLANDI */  
q[x]=q1[x];  
    }  
        for (x=0;x<255;x++)  
        {  
            for (y=0;y<255;y++)  
            {  
                k=0;k=a[x][y];ch=q[k];a[x][y]=ch;  
            }  
        }  
    }  
  
void sifir()  
{  
    for (x=0;x<256;x++)  
    {  
        q[x]=q1[x]=qq[x]=0;  
        lp=0;max=0;  
    }  
}
```

**EK A.9 GELİŞTİRİLEN 3B GÖRÜNTÜLEME PROGRAMINDA İSTENİLEN
DÜZLEMDE KESME YAPMA VE KESİT İNCELEME PROGRAMI**

```

/* BU PROGRAMDA BÖLÜM 5.4.2'DE MATEMATİKSEL BAĞINTILARI
VERİLEN İSTENİLEN DÜZLEMDE KESME YAPMA İŞLEMİNİN ALGO-
RİTMASI BULUNMAKTADIR. BU PROGRAM EK A.1'DE VERİLEN
PROGRAMIN BİTİŞİNE ÇOK YAKIN OLAN YERDEKİ İŞARETLİ YE-
RE OLDUĞU GİBİ YAZILIR. PROGRAM İÇİN GEREKLİ OLAN DE-
ĞİŞKEN TANIMLARI AŞAĞIDA VERİLMİŞTİR. */
/* DEĞİŞKEN TANIMLARI :
float tetal,fil;
int yük,zx,zy;
float aa1,aa2,aa3,bb1,bb2,bb3,cc1,cc2,cc3;
float ha1,ha2,ha3,hb1,hb2,hb3,hc1,hc2,hc3;
float d1,d2,d3,d4,d5,d6,d7,d8,d9,p01,p02;
float e1,e2,f1,f2,mat1,ex,ey,ez;
unsigned int r1[256][256],z1[256][256];
float m1[256][256]; */

/* KESME DÜZLEMİNİN DÖNME AÇILARI VE CİSİM ÜZERİNDEKİ YÜK-
SEKLİĞİ BURADA VERİLİR. DÖNME İŞLEMİ ÖNCE Y EKSENİNDE
TETA1 KADAR VE SONRA X EKSENİNDE Fİ1 KADARDIR. DÖNME
DENKLEMLERİ AYNEN NORMAL DÖNME DENKLEMLERİ ALINMIŞTIR. */

Q6 : c = 0;
system("clear");
printf("\nTetal degerini girin: ");
gets(s);
tetal=atof(s);
printf("\n Fil degerini girin: ");
gets(s);
fil=atof(s);
printf("\n yukseklik degerini girin: ");
gets(s);
yük=atoi(s);
tetal=(tetal*M_PI)/180;fil=(fil*M_PI)/180;

/* BU DÖNÜŞÜMLER KESME UZAYINDAN CİSİM UZAYINA DÖNÜŞÜM
YAPMAK İÇİN KULLANILIRLAR.*/

aa1=(cos(tetal));
aa2=(sin(tetal))*(sin(fil));
aa3=(cos(fil))*(sin(tetal));
bb1=0;
bb2=(cos(fil));
bb3=-(sin(fil));
cc1=-(sin(tetal));
cc2=(cos(tetal))*(sin(fil));
cc3=(cos(tetal))*(cos(fil));

```

```
/* BU DÖNÜŞÜMLERDE CİSİM UZAYINDAN RESİM UZAYINA GEÇİŞ
İÇİN KULLANILIRLAR */
```

```
ha1=(cos(teta));
ha2=0;
ha3=-(sin(teta));
hb1=(sin(fi))*(sin(teta));
hb2=(cos(fi));
hb3=(cos(teta))*(sin(fi));
hc1=(sin(teta))*(cos(fi));
hc2=-(sin(fi));
hc3=(cos(teta))*(cos(fi));
```

```
/* BÖLÜM 5'DE 5.14 DENKLEM TAKIMI İLE AÇIKLANAN DENK-
LEMLER. BU DENKLEMLER SABİT BİR SAYIYA ATANIP DEFA-
LARCA HESAPLAMA GEREKTİRMEYEN KULLANILIRLAR. */
```

```
d1=ha1*aa1+ha2*bb1+ha3*cc1;
d2=ha1*aa2+ha2*bb2+ha3*cc2;
d3=ha1*aa3+ha2*bb3+ha3*cc3;
d4=hb1*aa1+hb2*bb1+hb3*cc1;
d5=hb1*aa2+hb2*bb2+hb3*cc2;
d6=hb1*aa3+hb2*bb3+hb3*cc3;
d7=hc1*aa1+hc2*bb1+hc3*cc1;
d8=hc1*aa2+hc2*bb2+hc3*cc2;
d9=hc1*aa3+hc2*bb3+hc3*cc3;
```

```
/* FORMÜL 5.19'DAKİ DETERMİNANT HESABI */
```

```
f1=d3*yuk;mat1=(d1*d5)-(d2*d4);
f2=d6*yuk;
```

```
/* ASIL ÇEVİRİM BURADA BAŞLIYOR */
```

```
for (zx=-127;zx<128;zx++)
{
for (zy=-127;zy<128;zy++)
{
pol=zx;po2=zy;
e1=pol-f1;e2=po2-f2;
pol=(e1*d5)-(e2*d2);pol=pol/mat1;xx=pol;
po2=(e2*d1)-(e1*d4);po2=po2/mat1;yy=po2;
```

```
/* DERİNLİK BİLGİSİ OLAN Dz BURADA HESAPLANIR */
```

```
dz=d7*pol+d8*po2+d9*yuk;
ex=aa1*pol+aa2*po2+aa3*yuk+ccc+0.5;
ey=bb1*pol+bb2*po2+bb3*yuk+ccc+0.5;
ez=cc1*pol+cc2*po2+cc3*yuk+ccc+0.5;
x=ex;y=ey;z=ez;
```

```
/* ELDE EDİLEN KOORDİNATLARIN UZAYIN İÇİNDE OLUP OLMADIK-
LARI BURADA KONTROL EDİLİR. */
```

```
if(0<=x && x<=255 && 0<=y && y<=255 && 0<=z && z<=255)
{
    r1[zx+ccc][zy+ccc]=a[x][y][z];
    m1[zx+ccc][zy+ccc]=dz+ccc;
}
else
{
    r1[zx+ccc][zy+ccc]=0;
    m1[zx+ccc][zy+ccc]=dz+ccc;
}
}
```

```
/* KESME YÜZEYİNİN ALTINDA KALAN PARÇALARIN BASILMASI VE
ÜSTÜNDE KALANLARIN BASILMAMASI İŞLEMİ BURADA YAPILIR.
AYRICA TAM KESİLEN YÜZEYDE GÖRÜNTÜ ÜZERİNDE KESİT OLA-
RAK BASILIR. */
```

```
for (xx=0;xx<=255;xx++)
{
    for (yy=0;yy<=255;yy++)
    {
        if (r1[xx][yy]!=0 )
        {
            pr_put(screen,xx+756,yy+300,r1[xx][yy])
            ;
        }
        if (r1[xx][yy]==0 && m[xx][yy]<m1[xx][yy])
        {
            pr_put(screen,xx+756,yy+300,z1[xx][yy])
            ;
        }
        if (r1[xx][yy]==0 && m1[xx][yy]<m[xx][yy])
        {
            pr_put(screen,xx+756,yy+300,51);
        }
    }
}
```

```
/* MATRİSLER SORUN ÇIKMAMASI İÇİN SIFIRLANIR */
```

```
for (x=0;x<=zzz-1;x++)
{
    for (y=0;y<=zzz-1;y++)
    {
        r[x][y]=0;r1[x][y]=0;
        p[x][y]=0;u1[x][y]=0;u2[x][y]=0;u3[x][y]=0;
        v[x][y]=0;m1[x][y]=0;max1=0;
    }
}
```

```
    }  
  
    /* TEKRAR DÖNDÜRME İŞLEMİ YAPILMAK İSTENİRSE Y HARFİNE  
    BASILIR VE PROGRAM Q5 İLE BELİRTİLEN EN BAŞ TARAFA  
    GİDER (BAKINIZ EK A.1). DÖNME İSTENMEZ İSE KESME İŞ-  
    LEMİNE DEVAM EDİLİR.*/  
  
    printf("Dondurma istermisiniz ? y/n \n");  
    ch=getchar();  
    cr=getchar();  
    if (ch=='y') goto Q5;  
    goto Q6;  
    pr_close(screen);  
}
```

EK B

SUN SPARC 1+ HAKKINDA BİLGİLER

Bu bölümde kısaca çok kullanıcılı sistemlerden, Sun Sparc 1+ işistasyonundan ve vi editör ile C derleyiciden bahsedilecektir.

Sun Sparc 1+ işstasyonu Unix işletim sistemini temel alan bir yazılım kullanmaktadır ve bu yüzden çok kullanıcılı sistemler sınıfına girer. Çok kullanıcılı sistemlerin amaçlarını temel olarak ikiye ayırabiliriz :

- a) Görüntü makineler oluşturmak.
- b) Sistemdeki donanım kaynaklarını paylaşmak.

Tek kullanıcılı sistemlerde genel olarak kullanıcı belleği gerçek adresleri ile kullanılır. Ancak çok kullanıcılı sistemlerde, görüntü makina özelliğini sağlamak için disk yardımı ile olduğundan daha büyük bellek kullanımına izin veren işletim sistemleri gerekir. Kullanıcılar sistemdeki sınırlı kaynaklardan yararlanmaktadır. Bu nedenle sistem bir kullanıcıya, bir kaynağını süresiz olarak adayamaz. Yani kaynaklar, istekte bulunan tüm kullanıcılara belli zaman aralıkları içinde kullandırılmalıdır. Böylelikle kaynaklar zaman içinde paylaşılarak kullanılmış olur. Bu çok kullanıcılı sistemlerin en önemli görevlerindendir.

Çok kullanıcılı sistmlerin en önemli özelliklerinden biriside güvenlik sistemleridir. Bu güvenlik

önlemlerini şu şekilde sıralayabiliriz : Kullanıcı Şifresi, her kullanıcının okuma, yazma haklarının sistem yöneticisi (root) tarafından değiştirilebilmesi ve her kullanıcı alanına ne zaman girildiğinin belirlenmesi. Bu güvenlik sistemi ile birçok kullanıcının çalıştığı sistemde kullanıcı alanları birbirlerinden yalıtılmış olur.

Sun Sparc 1+ iş istasyonunda kullanılan Sun Os 4.1 işletim sistemi 4.3 BSD Unix, 4.2 BSD Unix işletim sistemlerinin geliştirilmesi ile oluşmuştur. Ayrıca AT & T ve V.3 Unix sistemlerinden alıntılar yapılarak oluşturulmuştur. Unix işletim sistemi ilk defa AT & T Bell Labaratuvarlarında geliştirilmiş ve bugüne kadar geliştirilen tüm Unix işletim sistemleri C programlama lisanında yazılmıştır. Bu sayede Sun Sparc 1+ işistasyonu programlarımızı yazdığımız C programlama lisanına tam destek vermektedir.

Sun işistasyonları mühendislik uygulamaları için (Bilgisayar destekli tasarım, Bilgisayar destekli üretim, Bilgisayar destekli yayıncılık, Program geliştirme) tasarlanmıştır. Bizim kullandığımız Sun Sparc 1+ işstasyonu SPARC (Scalable Processor ARCHitecture) teknolojisi ile imal edilmiştir. SPARC; daha basitlik ve etkinlik için tasarlanmış olan RISC (Reduced Instruction Set Cycles) yapısındadır. Bu yapıda özetle, her bir komut için minimum icra süresi sağlanmaktadır. Azaltılmış komut kümeside RISC

sistemlerinin hızlı çalışmasında etken olmaktadır.

Bu tezde kullanılan Sun Sparc 1+ istasyonunun donanım ve yazılım ile ilgili verilerini şu şekilde sıralayabiliriz :

Donanım : RISC yapısı ile beraber 32 MB RAM (en fazla 64 MB'a genişletilebilir) ve bu yapının üzerinde 1 adet 3.5' luk disket sürücü ünitesi, 1GB Harddisk, 1 adet magnetik teyp okuyucu, 1152*900 çözünürlüğü olan ve aynı anda 256 renk gösterebilen ve hızlı resim aktarma işlemi için üzerinde kendi hafızası olan bir ekran, 1 adet optik mouse. Sistemin hızı 16.5 mips (milion instruction per second) ve floating-point (kayan nokta) işlem hızı ise 2.7 megaflops' dur (mega floating-point per second).

Yazılım : Unix temelli olan ve Sun Microsystem tarafından geliştirilmiş olan SUN OS 4.1 işletim sistemini kullanılmaktadır. Sistemde SUN OS 4.1 işletim sisteminin kolay kullanımını sağlamak için pencereleme ile çalışan programlar vardır (OPENWIN ve SUNWINDOWS). Sistem satır editörü (ed) ve Ekran editörü (vi) gibi text yazılımlarına destek sağlamaktadır. Tezde yazılan programlar öncelikle vi editörde yazılmış ve sonra C derleyicide (cc) derlenmiştir. Sistem C ve Assambler lisanlarına tam destek vermektedir. Ayrıca sistemde haberleşme bağlantısını (eternet) sağlayan yazılımlar vardır.

Sistemde kullanılan vi editör ve C derleyicidende kısaca bahsedip komutlarını yazalım. Bu komutlar sayesinde Sun Sparc 1+ iş istasyonunda program yazılıp, işletilebilir:

Gerekli Olan Sun OS 4.1 Komutları :

ls -la : Dosya ve katalogları yazılım zamanları, sahipleri, uzunlukları ve izin durumları ile beraber ekranda görüntüler.

ls : Sadece dosya ve katalogların isimlerini görüntüler.

ls *.c : Sadece uzantısı c olan dosyaları görüntüler.

ls -ld : Sadece katalogları görüntüler.

cd .. : Bir üst kataloğa çıkmak için kullanılır.

cd <katalog adı> : Katalog değiştirmek için kullanılır.

rm <dosya adı> : Eğer izin hakkı varsa dosyayı silmek için kullanılır.

cp <dosya adı1> <dosya adı2> : Dosyaları kopyalamak için kullanılır. Dosya1, dosya2 adı ile kopyalanır ve dosya1 içeriğini korur.

mv <dosya adı1> <dosya adı2> : Dosya1, dosya2 olarak saklanır ve dosya1 içeriğini kaybederek yokolur.

more <dosya adı> : Dosyanın içeriğini görmek için (izin hakkı varsa)

Gerekli Olan Vi Editör Komutları :

vi <dosya adı> : "dosya adı" ile belirtilen yeni bir yazı dosyası açılır. Eğer bu ad ile aynı bir dosya var ise ekrana o dosya gelir.

esc - a: Kursörü bir sağa hareket ettirir.

esc - o: Üzerinde durduğu satırın altına boş satır açar.

esc - i: Üzerinde bulunduğu karakterin yerinden başlamak üzere yazılanları araya sokar.

esc - x: Bulunduğu konumun sağındakileri her basışta birer birer olmak üzere siler. Sonra sağdakileri silmeye başlar.

esc - r - karakter: Üzerinde bulunduğu karakteri, r harfine basıldıktan sonra basılan karakter ile değiştirir.

esc - d+w: Sağındaki kelimeyi siler.

esc - d+l: Satırı siler.

ok tuşları: Metin içinde hareketi sağlar.

shift +] : Metin içinde hızlı olarak aşağı inmek için.

shift + [: Metin içinde hızlı olarak yukarı çıkmak için.

esc - (:) : Vi editör komut bölümüne çıkmayı sağlar. Bu komut bölümünde kullanılan komutlardan bazıları:

wq - enter: Metni yazıp saklamak için.

q! - enter: Değişiklikleri saklamadan çıkmak için.

wq <dosya adı>: Dosyayı başka ad altında saklamak için.

(dosya adı yeni isimi temsil etmektedir.)

/ <kelime> - enter: "kelime" ile verilen parçayı metin

içinde bulmak için.

n - enter: Devamlı aynı kelimeyi aratırmak için.

C Derleyici Komutları :

cc -g -o <dosya adı><dosya adı.c> -<kullanılan arşivler>

-g: Programın debug (adım adım koşturulmasını sağlayan program) için gerekli olan bir komuttur. Bu parça yazılmaz ise program debug edilemez.

-o: Programın obje kodunun oluşturulması için olan komuttur.

<dosya adı>: Vi editörde yazılan programın obje kod adıdır.

<dosya adı.c>: Vi editörde yazılan programın adıdır.

<kullanılan arşivler>: Sun Os işletim sisteminde bulunan bazı yazılımların arşivleri kullanılırsa bunların derleme aşamasında mutlaka belirtilmesi gerekir. Bu arşivler programın en başında programa dahil edilmelidirler.

Örneğin :

```
#include <pixrect/pixrect.h>
```

```
#include <math.h>
```

Bu tanımlardan sonra programın derleme bölümünde:

```
cc -g -o derya derya.c -lpixrect -lm
```

olarak yazılmak zorundadır. Bu örnekte obje kod programının adı derya olmuştur. Bu program işletilmek istenildiğinde sadece obje kod programının adını yazmak yeterlidir.

EK.C KONUSU İLE İLGİLİ TIP TERİMLERİ

Anjiyografi (Angiography) : Kan damarlarının radyoopak madde zerkini takiben röntgen yoluyla incelenmesi; bu yolla damarların filme alınması.

Apex : Tepe, uç; herhangi bir organ veya oluşumun gittikçe sivrilen kısmı (konuda kalbin alt ucu).

Kardiovasküler (Cardiovascular) : Kalp ve kan damarları ile ilgili.

Cerebrospinal : Beyin ve omurilikle ilgili.

Cerebrospinal Sıvı : Beyinde ventriküler bölgede ve omurilikte dolaşan hayati sıvı.

Cerebral : Beyine ait, beyin ile ilgili.

Konjenital (Congenital) : Doğuştan; doğuştan var olan.

Kraniofasial (craniofacial) : Kafatasının ön yani yüzü oluşturan bölümü.

Cortex : Organın dış kısmı; herhangi bir organda kapsül ile medulla arasında yer alan kısım. **Cerebral cortex :** beyin korteksi.

Edema : Derialtı sıvılarında aşırı sıvı toplanması, ödem.

Lezyon (Lesion) : Patolojik değişim gösteren doku bölgesi. Herhangi bir enfekte deri hastalığında görülen sınırlı oluşumlardan her biri.

Myocardial : Kalbin kas tabakası ile ilgili, kalb kası ile ilgili.

Pelvis : Önde ve yanlarda kalça kemikleri ile sınırlanan bel bölgesindeki leğen kemiği.

Radyoterapi (Radiotherapy) : Işın tedavisi; hastalıkların ve patolojik lezyonların x ışınları veya radyoaktif ışınlarla tedavisi.

Sistol : Kalbin kanı aorta basmak için kasılma anı.

Spermatozoa : Erkek cinsiyet hücresi ile ilgili olan.

Travma (Trauma) : Herhangi bir dış etkenle vücutta oluşan harabiyet (yaralanma, incinme, zedelenme vs..)

Travmatoloji (Traumatology) : Kaza sonucu meydana gelen yaralanma ve incinmelerin tedavisini konu alan tıp dalı.

Ventriküler (Ventricular) : Karıkcık ile ilgili olan.

KAYNAKLAR

- [1] Gary.T.Read , 3D images Bridge Gap Between Diagnostic and Therapy , Diagnostic imaging, May 1991.
- [2] Tomita.K , Clinical Applications of CT 3D Surface Reconstruction Technique in Neuradiology ,Toshiba Medical Review No:16 , 1989.
- [3] Hoehne.K.H , Research Activities of the Department of Computer Science in Medicine , University of Hamburg , February 1989.
- [4] John.C.L , Three-dimensional Reconstruction of The Hearth Using EKG-triggered Magnetic Resonans Images , Journal Of Medical Imaging , April 1987.
- [5] Hoehne.K.H , 3D Imaging In Medicine , NATO ASI series , 1990.
- [6] L.Adams , Computer-Assisted Surgery , IEEE Computer Graphics And Applications , May 1990.
- [7] Yasuzo.S , Three Dimensional Displays of Medical Images Using X-ray CT and MRI , Medical Review No:16 , 1989.
- [8] Sümer.A. , 3D Medical Imaging of Internal Organs on an IBM PC , Master Thesis , Bosphorous University 1988.
- [9] Hoehne.K.H , 3D Visualization of Tomographic Volume Data Using The Generalized Voxel Model , Visual Computer , June 1990.
- [10] Herman.M , Mathematic and Computer Science in Medical Imaging , NATO series , 1989.

- [11] Herman.G.T , A Survey of 3D Medical Imaging Technologies , IEEE Engineering In Medicine and Biology , December 1990.
- [12] Newman.W.M , Principles of Interactive Computer Graphics , Mc.Graw Hill , 1981.
- [13] Alan.R.D , An Overview Of Rendering Techniques , Computer and Graphics , Vol 14 , No:1 pp:101-115 , 1990.
- [14] U.Tiede , Investigation of Medical 3D Rendering Algorithms , IEEE Computer Graphics and Applications , March 1990.
- [15] Michael.W.V , Fast Sequence Add Depth to 3D images , Diagnostic imaging, May 1991.
- [16] Sun Microsystems , Sun System introduction, Revision A , May 1988.

ÖZGEÇMİŞ

Yazar 1967 yılında İstanbul'da doğdu. Orta ve lise eğitimini sırası ile Özel Kalamış Lisesi ve Kartal Teknik Lisesi Elektrik bölümünde tamamladı. 1985 yılında İstanbul Teknik Üniversitesi Elektrik - Elektronik Fakültesi Elektronik ve Haberleşme bölümünde okumaya hak kazandı. 1989 yılında iyi derece ile mezun oldu ve aynı yıl İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik programında yüksek lisans eğitimine başladı. 1990-1991 ders yılında Prof.Dr.Ertuğrul YAZGAN yönetiminde yüksek lisans tezini üzerinde çalışmalarını sürdürdü.

