

**DOKUZ EYLÜL UNIVERSITY  
GRADUATE SCHOOL OF NATURAL AND APPLIED  
SCIENCES**

**COMPARISON OF 3D SEGMENTATION  
ALGORITHMS FOR MEDICAL IMAGING**

**by  
Hakan BULU**

**December, 2006  
İZMİR**

# **COMPARISON OF 3D SEGMENTATION ALGORITHMS FOR MEDICAL IMAGING**

**A Thesis Submitted to the  
Graduate School of Natural and Applied Sciences of Dokuz Eylül University  
In Partial Fulfillment of the Requirements for the Degree of Master of Science  
in Computer Engineering, Computer Engineering Orientation Program**

**by  
Hakan BULU**

**December, 2006  
İZMİR**

## M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**COMPARISON OF 3D SEGMENTATION ALGORITHMS FOR MEDICAL IMAGING**” completed by **Hakan BULU** under supervision of **Assc.Prof.Dr. Adil ALPKOÇAK** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

.....  
**Assc.Prof.Dr. Adil ALPKOÇAK**

Supervisor

.....  
**Ins.Dr. Malik Kemal ŞİŞ**

(Jury Member)

.....  
**Prof.Dr. Cüneyt GÜZELİŞ**

(Jury Member)

.....  
Prof.Dr. Cahit HELVACI

Director

Graduate School of Natural and Applied Sciences

## **ACKNOWLEDGMENTS**

I would like to thank my supervisor "Assc.Prof.Dr. Adil ALPKOÇAK" for the encouragement, support and valuable solutions he produced.

I also appreciate the encouragement of the entire family members (the teaching staff) of Computer Engineering Dep. / Dokuz Eylül University.

Lastly, I gratefully would like to thank my fiancé, my family and my friends Tolga BERBER, Muhittin Erkut SOYHEPTEMİZ for the encouragement, support and care.

Hakan BULU

# COMPARISON OF 3D SEGMENTATION ALGORITHMS FOR MEDICAL IMAGING

## ABSTRACT

In this thesis we implemented four different 3D segmentation algorithms, and we compared their results on three different CT Data Sets. These segmentation algorithms are; Seeded Region Growing, Volumetric Segmentation Using Weibull E-SD Fields , Automatic Multilevel Thresholding by using OTSU Method and Unseeded Region Growing.

The main results gained from our application as follows; Seeded Region Growing Algorithm produced good result on unnoised datasets with suitable threshold value. Volumetric Segmentation Using Weibull E-SD Fields Algorithm produced good result on our sample dataset which has high amount of contrast difference. However, The results on medical datasets which has low amount of contrast difference. Automatic Multilevel Thresholding by using OTSU Method Algorithm which takes the segment count as an input by user interaction, produced sufficient results. And lastly, the number of segments produced by the Unseeded Region Growing Algorithm are over the expectations. But they can be considered as sufficient.

**Keywords :** OTSU , Medical Image , 3D , Weibull E-SD Fields

# TIBBİ GÖRÜNTÜLER İÇİN 3 BOYUTLU BÖLÜTLEME ALGORİTMALARININ KARŞILAŞTIRILMASI

## ÖZ

Bu tezde dört farklı 3 boyutlu bölütleme algoritması uygulamaya geçirilerek, sonuçları üç farklı bilgisayarlı tomografi görüntüsü seti üzerinde karşılaştırılmıştır. Karşılaştırılan bölütleme algoritmaları ; Seeded Region Growing, Volumetric Segmentation Using Weibull E-SD Fields , Automatic Multilevel Thresholding by using OTSU Method ve Unseeded Region Growing'dir.

Bu tezde elde edilen başlıca sonuçlar; Seeded Region Growing algoritması uygun threshold değerlerinde ve fazla gürültü içermeyen resimler üzerinde iyi sonuçlar vermiştir. Volumetric Segmentation Using Weibull E-SD Fields algoritması kontrast farkının yüksek olduğu test verileri üzerinde iyi sonuçlar vermesine karşın, yakın kontrast değerlerine sahip tıbbi görüntüler üzerinde sağlıklı sonuçlar üretmemiştir. Automatic Multilevel Thresholding by using OTSU Method algoritması kendisine verilen segment sayısı doğrultusunda başarılı denebilecek sonuçlar üretmiştir. Ve son olarak Unseeded Region Growing algoritması Seeded Region Growing algoritmasında kullanılan threshold değerlerinde beklenenden fazla sayıda segment üretmesine karşın, ürettiği sonuçlar başarılı sayılabilir.

**Anahtar Sözcükler :** OTSU , Tıbbi Görüntü , 3B , Weibull E-SD Alanları.

<b>CONTENTS</b>	<b>Page</b>
ACKNOWLEDGMENTS.....	iii
ABSTRACT.....	iv
ÖZ.....	v
CHAPTER ONE INTRODUCTION .....	1
CHAPTER TWO 3D MEDICAL IMAGE SEGMENTATION ALGORITHMS.....	3
2.1 Image Segmentation Algorithms.....	3
2.1.1 Thresholding techniques .....	3
2.1.2 Boundary-based methods .....	3
2.1.3 Region-based methods .....	4
2.1.4 Hybrid methods.....	4
2.2 Medical Image Segmentation.....	5
2.3 3D Image Segmentation Algorithms.....	6
2.3.1 Seeded Region Growing .....	6
2.3.2 Weibull E-SD Fields .....	7
2.3.3 Automatic Multilevel Thresholding by using OTSU Method .....	10
2.3.4 Unseeded Region Growing .....	13
CHAPTER THREE EXPERIMENTATION.....	14
3.1 Briefly Reviewing Of Our Application.....	15
3.2 Data Sets .....	16
3.2.1 Sample Dataset.....	17
3.2.2 Medical Datasets .....	17
3.3 Experimentations .....	20
3.3.1 Seeded Region Growing .....	20
3.3.2 Weibull E-SD Fields .....	25
3.3.3 Automatic Multilevel Thresholding by using OTSU Method .....	28
3.3.4 Unseeded Region Growing .....	32
3.4 Briefly Reviewing Of Comparison Of Algorithms.....	35
CHAPTER FOUR CONCLUSION .....	37

## CHAPTER ONE

### INTRODUCTION

Nowadays, the usage of medical imaging is not only limited by simple visualization and inspection of atomic structures, but also it can be used as in advanced surgical planning and simulation, radiotherapy planning, and patient diagnosis.

However modern visualization techniques provide extremely accurate and high quality 3D images of anatomical structures, their usage for efficient analysis is still limited, because the complex structure of animals and human with a large number of anatomical organs which are bunched together is prevented the doctors to view in different ways. Some visualization tricks, such as making an object transparent, cannot solve this problem. To solve the problem, the anatomical structure or the region of interest needs to be separated or delineated, which allows to view it individually. This technique is called *image segmentation* in medical imaging.

Image segmentation is generally considered as a very difficult problem due to very large size of datasets, complexity, and variation of the atomic organs. Noise and low contrast may be caused the boundaries of anatomical structures to be inaccurate and disconnected. Therefore the main challenge of the segmentation algorithms are to extract the boundaries of organs or region of interests and to separate it out from the remained datasets. Many approaches for the segmentation proposed in literature vary widely depending on the specific application, imaging modality (CT, MRI, etc.), and other factors. For example, the segmentation of colons has more different issues than the segmentation of lungs does. The algorithm, which gives perfect results for one application, might not even work for another. Besides these, the outcome of the segmentation algorithms can be effected by the general imaging artifacts such as noise, motion and partial volume effects. For instance, a segmentation algorithm could be robust to noise, but it might be failed in the



presence of partial volume effects. This variety creates a challenging problem for the segmentation algorithms. Currently, we do not have segmentation method, which provides acceptable results for any type of medical dataset. Generally, a segmentation algorithm approach could be consisted many segmentation algorithms, which are applied one after another. Therefore the selection of an appropriate algorithm or approach for segmentation may be caused a dilemma.

In this thesis, we compared four different segmentation algorithms and we run them on the three different CT datasets for comparing their results. These are, Seeded Region Growing, Weibull E-SD Fields, Automatic Multilevel Thresholding by using OTSU Method and Unseeded Region Growing.

In Chapter Two, fundamental image segmentation techniques and Hounsfield units are briefly reviewed. In addition to them , our 3D image segmentation algorithms are explained, in detail. In Chapter Three, first we give some information about our application and our datasets and with their screen shots. Second, we show algorithm result for all datasets with their Parameters. Finally, Chapter Four concludes the study discusses the results gained from algorithms and give a look at the future studies on the subject.

## **CHAPTER TWO**

### **3D MEDICAL IMAGE SEGMENTATION ALGORITHMS**

In this chapter, fundamental image segmentation techniques, medical image segmentation, its importance and Hounsfield units are briefly reviewed. In addition to them, our 3D image segmentation algorithms which are implemented and compared are explained in detail.

#### **2.1 Image Segmentation Algorithms**

Segmentation is a process, in which an image is divided into constituent objects or parts. It is the first and most important step in an image analysis task. Usually effective segmentation is dictated as eventual success of analysis. Therefore many segmentation techniques have been developed by researchers worldwide. The segmentation of intensity images usually involves in four main approaches; thresholding techniques, boundary-based methods, region-based techniques and hybrid methods. (Lakare, 2000)

##### ***2.1.1 Thresholding techniques***

Threshold techniques are based on the thresholds, which are usually selected from the image histograms. It is said that, the value of all pixels, which are gray, intensity, color, and others, is between two values of thresholds belong to one region. The fact that, the thresholds are derived from the histogram, which shows that, these techniques do not take into account spatial information of the image. Also they have problems to cope well with noise as well as with blurred edges on the image.

##### ***2.1.2 Boundary-based methods***

Main idea of the Boundary-based methods is assuming that pixel values change rapidly between two regions at the boundary. The basic principle of boundary-based methods is to apply some of the gradient operators convolving them

with the image. High values of the gradient magnitude are possible places of rapid transition between two different regions, what we call edges. After this step of finding edges on the image, they have to be linked to form closed boundaries of the regions. To go from image edges to the image boundaries is a very difficult task, which has been studied at a lot.

### ***2.1.3 Region-based methods***

The point is to group the pixels of the same or similar brightness or color to the regions according to the given criteria of homogeneity, in other words looking at the neighboring pixels of the given pixel and merging them into the region, if the criteria of homogeneity is satisfied. Homogeneity criteria is based on some threshold value, the choice of which is problematic, because we usually have to change a lot the right choice of the thresholds, and also thresholds are always depended on the image data, and results are sensitive to noise.

### ***2.1.4 Hybrid methods***

Hybrid methods combine one or more abovementioned criteria. This class includes the morphological watershed segmentation, variable order surface fitting, and active counter methods. The watershed method is generally applied to the gradient of the image. The gradient image can be considered as topography with boundaries between regions as ridges. Unlike the boundary-based methods, the watershed is guaranteed to produce closed boundaries even if the transition between regions have variable strength or sharpness. Although, it encounters difficulties with images, in which regions are both noisy and have blurred or indistinct boundaries. The variable order surface fitting method starts with a coarse segmentation of the image into several primitives, which are refined by iterative region growing procedure. Active contour models are based on gradient information along the boundary between regions, but they are useful only when a good initial estimate is present.

## 2.2 Medical Image Segmentation

For advanced surgical planning, simulation, radiotherapy planning and patient diagnosis medical image segmentation is an important topic. CT Scans was invented by Sir Godfrey Newbold Hounsfield. (Wikipedia , 2006). The pixel value is displayed according to the mean attenuation of the tissue that it corresponds to on a scale from -1024 to +3071 on the Hounsfield scale, given Table 2.1. By using this scale image segmentation can be performed basicly. Which means that, if you want to get bone tissue from whole volume, just getting all voxels which values are higher than 1000.

Table 2.1 The HU of common substances

<i>Substance</i>	<i>HU</i>
Air	-1000
Fat	-120
Water	0
Muscle	+40
Bone	+1000

To convert from the normal units found in CT data (a typical data set ranges from 0 to 4000 or so) you have to apply a linear transformation of the data. The equation is: (Dfanning, 2006)

$$HU = voxel\_value \times slope + intercept \quad (1)$$

Normally, these values are stored in the DICOM file itself. The tags are generally called the *Rescale Slope* and *Rescale Intercept*, and typically have values of 1 and -1024, respectively.

We used this conversion formula for testing Hounsfield Units's image segmentation performans on our medical datasets. First off all, we convert our datasets DS1 and DS2 to 12 Bit, by using DCMTK. (Dicom Office , 2006). For both of them, *slope* value is equal to 1 and *intercept* value is equal to 1024. After this transformation, we try to get bone tissue, but with bone tissue, some part of the

volume's voxels had intensity value more than 1000, too. So we couldn't get desired region by using Hounsfield Units. We get *slope* and *intercept* values from DICOM Header information and we don't know if they are true.

### 2.3 3D Image Segmentation Algorithms

Until here, we talked about segmentation algorithms for 2D images. Now in following sections, we will talk about 3D images namely *volume* and 3D image segmentations.

For 3D images, we must add depth to an image using a set of images known as a slice. These slices are made up of pixels. So, for 3D images pixel (x,y) become a voxel (x,y,z) namely volume pixel. A voxel represents a quantity of 3D data just as a pixel represents a point or cluster of points in 2D data. It is used in scientific and medical applications that process 3D images. Actually, for medical imaging 3D images consist of set 2D images namely slice. If you put set of slice in a correct order, you can get a 3D image. In following sections, our 3D segmentation algorithms are explained in detail.

#### 2.3.1 Seeded Region Growing

Seeded Region Growing is a technique to extract a connected region from a 3D volume based on some pre-defined connecting criterion. (Ulicny & Ilic , 2006) This criteria can be as simple as the voxel intensity or could be the output of any other segmentation algorithm. In the simplest form, Seeded Region Growing requires a seed point to start with. From the seed point, the algorithm grows till the connecting criteria is satisfied. We can define the connection criteria  $\delta(x)$  as follows;

$$\delta(x) = |g(x) - \text{mean}(\text{Region})| \quad (2)$$

Here  $g(x)$  is the intensity value of the current voxel. If  $\delta(x) \leq T$ , we can add current voxel to the Region. Where  $T =$  Threshold Value and set from user. ( $0 < T < 255$ )

The primary disadvantage of this algorithm is that it requires seed points which generally means manual interaction. Thus for each region to be segmented, a seed point is needed. Region growing can also be sensitive to noise and partial volume effect causing the extracted region to have holes or disconnections. Some recent work has been reported which tries to alleviate these problems. In another recent work, fuzzy analogies to region growing have also been developed. In Figure 2.1, we show pseudo code of the Seeded Region Growing Algorithm.

```

Input           : Seed Voxel, Label, 3D Volume
Output          : Region Volume

Initialization : Label the seed voxel and set the region initial mean
value as seed voxel value, push the seed voxel's unlabelled neighbors into
stack.

While stack is NOT EMPTY
  POP one voxel from stack.
  If this voxel belongs to region
    Label this voxel.
    Update region's mean value.
    PUSH this voxel's unlabelled neighbors into stack.
  End If
End While

```

Figure 2.1 Pseudo Code of The Seeded Region Growing Algorithm

### 2.3.2 Weibull E-SD Fields

This segmentation technique is purposing to make a segmentation by making coarser the volume data. For doing that,  $K \times K \times K$  cubes are generated. They are called

K-Voxel. A K-Voxel is consist of voxels. Each K-voxel is assigned two values; Expectancy and Standart Deviation (E-SD). For removing noise of image Weibull noise index is using. In this way, more precies E-SD values can be obtained. After callculating E-SD values, frequency of voxels which have the same E-SD values is plotted. Segmentation is performaing by using this plot.

Definition :

A region  $\Omega$  is called as a Spatially Distributed Object (SDO), if the expectancy and standard deviation for each K-voxel  $\Delta$  in  $\Omega$  are relatively constant, i.e.,

$$E[X_{\Delta}] \in (e_1, e_2) \text{ and } SD[X_{\Delta}] \in (d_1, d_2) \quad (3)$$

where  $e_1, e_2, d_1$  and  $d_2$  denote predefined constants with  $e_1 \leq e_2, d_1 \leq d_2$ , the random variable  $X_{\Delta}$  is defined as  $X_{\Omega}$  above.

Calculating Expectancy and Standart Deviation (E-SD) Value of K-Voxel are given as follows [4] , [5] :

$$E[X_{\Delta}] = \frac{1}{|\Delta|} \sum_{(x,y,z) \in \Delta} v(x, y, z) \quad (4)$$

and

$$SD[X_{\Delta}] = \sqrt{\frac{1}{|\Delta|} \sum_{(x,y,z) \in \Delta} v^2(x, y, z) - E^2[X_{\Delta}]} \quad (5)$$

where  $|\Delta|$  denotes the number of voxels in K-Voxel and  $v(x, y, z)$  is the intensity value of the 3D Volume at  $(x,y,z)$  point.

Removing noise or improving the signal-to-noise ratio (SNR) of a given image is an essential step in segmentation, especially in high noise situations that can disrupt the shape and lose the edge information.

Let  $v_1, v_2, \dots, v_K^3$  represent  $K^3$  image points in a given  $K$ -voxel. It is assumed that the value of a voxel is characterized by the Weibull distribution. If we use (2) and (3) to calculate the E-SD value, the results are not reliable due to noise, especially for a standard deviation. Therefore, we must find a way to distinguish whether or not the data distribution in a  $K$ -voxel is uniform. If it is not uniform, then what kind of noise is present? If few of the elements in a voxel are significantly larger and/or smaller than others, then these are called upper/lower noise. For example, in a 2-voxel, in which the set of the intensity at eight image points is  $\{ 234; 52; 64; 46; 50; 54; 62; 3 \}$ , element 234 is much larger than others, and is called an upper noise. Element 3 is significantly less than others and is called a lower noise. In order to classify the noise in a  $K$ -voxel, an auxiliary function  $g(s)$  is introduced:

$$g(s) = \frac{(\sum_{i=1}^n v_i^s)^2}{n \sum_{i=1}^n v_i^{2s}} \quad (6)$$

where  $g(s) = 0.72$ ,  $s_1$  and  $s_2$  are the roots of  $g(s)$  they are called Weibull upper and lower noise index.

- For a  $K$ -voxel, if the Weibull upper noise index  $|s_1| < 1.26$  and the lower noise index  $|s_2| > 1.26$ , then there is upper noise in it.
- For a  $K$ -voxel, if the Weibull upper noise index  $|s_1| > 1.26$  and the lower noise index  $|s_2| < 1.26$ , then there is lower noise in it.
- For a  $K$ -voxel, if the Weibull upper noise index  $|s_1| < 1.26$  and the lower noise index  $|s_2| < 1.26$ , then there is upper and lower noise in it.

Based on the theoretical analysis above, the algorithm for volume data segmentation is as follows:



*Step 1:* Given a  $K$  to determine the size of  $K$ -voxel; initialize the SDO's predefined constant in (1):  $e_2 > e_1 > 0$ ,  $d_2 > d_1 \geq 0$ , and the threshold of expectancy  $T_e > 0$ .

*Step 2:* Consider the  $j^{\text{th}}$   $K$ -voxel. Use bisection to compute its Weibull noise index  $s_1$  and  $s_2$ , which are the roots of the equation  $g(s) = 0.72$ , where  $g(s)$  is defined by (4). If there is upper noise or lower noise or both, then remove the noise directly (i.e., delete the minimum or the maximum or both). Repeat at most  $[C_K^3]$  times to execute Step 2.

*Step 3:* Calculate E-SD values using (2) and (3). If the expectancy is larger than the threshold  $T_e$ , add the  $K$ -voxel to list A. If there are  $K$ -voxels which have not been dealt with, then go to Step 2.

*Step 4:* Compute the frequency of the voxel in the list A and create the E-SD field. By choosing the suit E-SD values  $e_2 > e_1 > 0$ ,  $d_2 > d_1 \geq 0$ , select the voxel in which the expectancy  $E$  and standard deviation  $SD$  satisfy:  $e_2 \geq E \geq e_1$ ,  $d_2 \geq SD \geq d_1$ . (Hu & Razdan, 2003)

### **2.3.3 Automatic Multilevel Thresholding by using OTSU Method**

Automatic thresholding is an important technique in image segmentation. The basic idea of automatic thresholding is to automatically select an optimal gray-level threshold value for separating objects of interest in an image from the background based on their gray-level distribution.

Automatic thresholding techniques can be roughly categorized as global thresholding and local thresholding. Global thresholding selects a single threshold value from the histogram of the entire image. Local thresholding uses localized gray-level information to choose multiple threshold values; each is optimized for a small region in the image. Global thresholding is simpler and easier to implement but its result relies on good (uniform) illumination. Local thresholding methods can deal with non-uniform illumination but they are slow.

Among the global thresholding techniques, the OTSU method was one of the better threshold selection methods for general real world images. This method selects

threshold value that maximizes the between-class variances of the histogram. (Liao,Chen & Chung , 2001)

An image is a 2D grayscale intensity function, and contains  $N$  pixels with gray levels from 1 to  $L$ . The number of pixels with gray level  $i$  is denoted  $f_i$ , giving a probability of gray level  $i$  in an image of

$$P_i = f_i / N \quad (7)$$

In the case of bi-level thresholding of an image, the pixels are divided into two classes,  $C_1$  with gray levels  $[1, \dots, t]$  and  $C_2$  with gray levels  $[t+1, \dots, L]$ . Then, the gray level probability distributions for the two classes are

$$\begin{aligned} C_1 : P_1 / \omega_1(t), \dots, P_t / \omega_1(t) \text{ and} \\ C_2 : P_{t+1} / \omega_2(t), P_{t+2} / \omega_2(t), \dots, P_L / \omega_2(t), \end{aligned} \quad (8)$$

where  $\omega_1(t) = \sum_{i=1}^t P_i$

and

$$\omega_2(t) = \sum_{i=t+1}^L P_i \quad (9)$$

Also, the means for classes  $C_1$  and  $C_2$  are

$$\mu_1 = \sum_{i=1}^t iP_i / \omega_1(t) \quad (10)$$

and

$$\mu_2 = \sum_{i=t+1}^L iP_i / \omega_2(t) \quad (11)$$

Let  $\mu_T$  be the mean intensity for the whole image. It is easy to show that

$$\omega_1\mu_1(t) + \omega_2\mu_2(t) = \mu_T \quad (12)$$

$$\omega_1 + \omega_2 = 1 \quad (13)$$

Using discriminant analysis, Otsu defined the between-class variance of the thresholded image as

$$\sigma_B^2 = \omega_1(\mu_1 - \mu_T)^2 + \omega_2(\mu_2 - \mu_T)^2 \quad (14)$$

For bi-level thresholding, Otsu verified that the optimal threshold  $t^*$  is chosen so that the between-class variance  $\sigma_B^2$  is maximized; that is,

$$t^* = \arg \max_{1 \leq t < L} \{ \sigma_B^2 \} \quad (15)$$

The previous formula can be easily extended to multilevel thresholding of an image. Assuming that there are  $M-1$  thresholds,  $\{t_1, t_2, \dots, t_{M-1}\}$ , which divide the original image into  $M$  classes:  $C_1$  for  $[1, \dots, t_1]$ ,  $C_2$  for  $[t_1+1, \dots, t_2]$ ,  $\dots$ ,  $C_i$  for  $[t_{i-1}+1, \dots, t_i]$ ,  $\dots$ , and  $C_M$  for  $[t_{M-1}+1, \dots, L]$ , the optimal thresholds  $\{t_1^*, t_2^*, \dots, t_{M-1}^*\}$  are chosen by maximizing  $\sigma_B^2$  as follows:

$$\{t_1^*, t_2^*, \dots, t_{M-1}^*\} = \arg \max_{1 \leq t_1 < \dots < t_{M-1} < L} \{ \sigma_B^2(t_1, t_2, \dots, t_{M-1}) \} \quad (16)$$

where

$$\sigma_B^2 = \sum_{k=1}^M \omega_k (\mu_k - \mu_T)^2 \quad (17)$$

### 2.3.4 Unseeded Region Growing

Unseeded region growing is a fully automatic segmentation technique suitable for 3D images. The base of the Unseeded Region Growing Algorithm resembles to Seeded Region Growing Algorithm. Like Seeded Region Growing, it requires a start point and threshold value. But dissimilar Seeded Region Growing Algorithm, those parameters can be set as a first step of the algorithm, and all regions can be produced by using those values. In Figure 2.2, we show pseudo code of the Seeded Region Growing Algorithm. (Lin, Jin & Talbot , 2000)

```

Input          : Seed Voxel, Label, 3D Volume
Output         : Region Volume

Initialization: Label the seed voxel and set the region initial mean value
as seed voxel value, push the seed voxel's unlabelled neighbors into stack.

While stack is NOT EMPTY
    POP one voxel from stack.
    If this voxel belongs to region
        Label this voxel.
        Update region's mean value.
        PUSH this voxel's unlabelled neighbors into stack.
    Else
        Set this voxel as new seed point
    End If
End While

```

Figure 2.2 Pseudo Code of The Unseeded Region Growing Algorithm

## **CHAPTER THREE**

### **EXPERIMENTATION**

In this thesis, main goal of us is comparison of 3D Segmentation Algorithms. For doing that, first of all we had to display our 3D Volume (Medical Datasets) and the algorithms result in 3D. Therefore, we used Vissulation Tool Kit (VTK). VTK has lots of abilities for presetting desired 3D visualization. While doing that, it takes many parameters. For this purpose, we built an interface for setting and saving VTK's parameteres. In this way we can use this visualization Parameters again and again without setting each time.

Some of our algorithms are semi-automatic , some of them full-automatic. For semi-automatic ones, user interaction is required, so we built an one more interface. By using this interface, user can select a voxel from 3D volume. In addition to that, user can set Parameters, which is required from algorithms dynamicly. For example, threshold value.

Another subject which we consider is reducing computation time. Despite Seeded and Unseed Region Growing Algorithms do not need computation time, E-SD Fields and OTSU Method need computation time. Calculating E and SD values of each K-voxel and calculating threshold values from entered images. For reducing this computation time whenever we run our algorithm for same datasets, we constructed file formats for algorithm results. In this way, for example when we want to run E-SD Fields algorithms for DS1, just load .esd file for DS1 which contains all K-Voxels E and SD values. There is no computation time anymore.

In following sections, we will give some short information about our application and our datasets. For explanation we use set of screen shots. Each of them is captured from different point of view of 3D volume. And lastly, while showing algorithms result, we used blue color for segmented region.

### 3.1 Briefly Reviewing Of Our Application

Our application was developed on Microsoft .Net Platform by using C# programming language. For 3D representation we used Visualisation Tool Kit (VTK) for .Net.

Main Form of our application is showed in Figure 3.1. All slices are displayed in Part 1 of the form. Part 2 is containing selected slice from user. Part 3 is DICOM Header information, in XML Format. Part 4 shows us all regions which produced by selected segmentation algorithm with their Region ID which is a unique number for separating one region from the others. And lastly, Part 5 contains quick information of selected region. Region ID, Voxel List Count which means that, how many voxels is containing in the region, Voxel Edge Count means that, how many voxels are on the boundary of this region and Region Mean Value is a mean intensity value of all voxels of selected region.

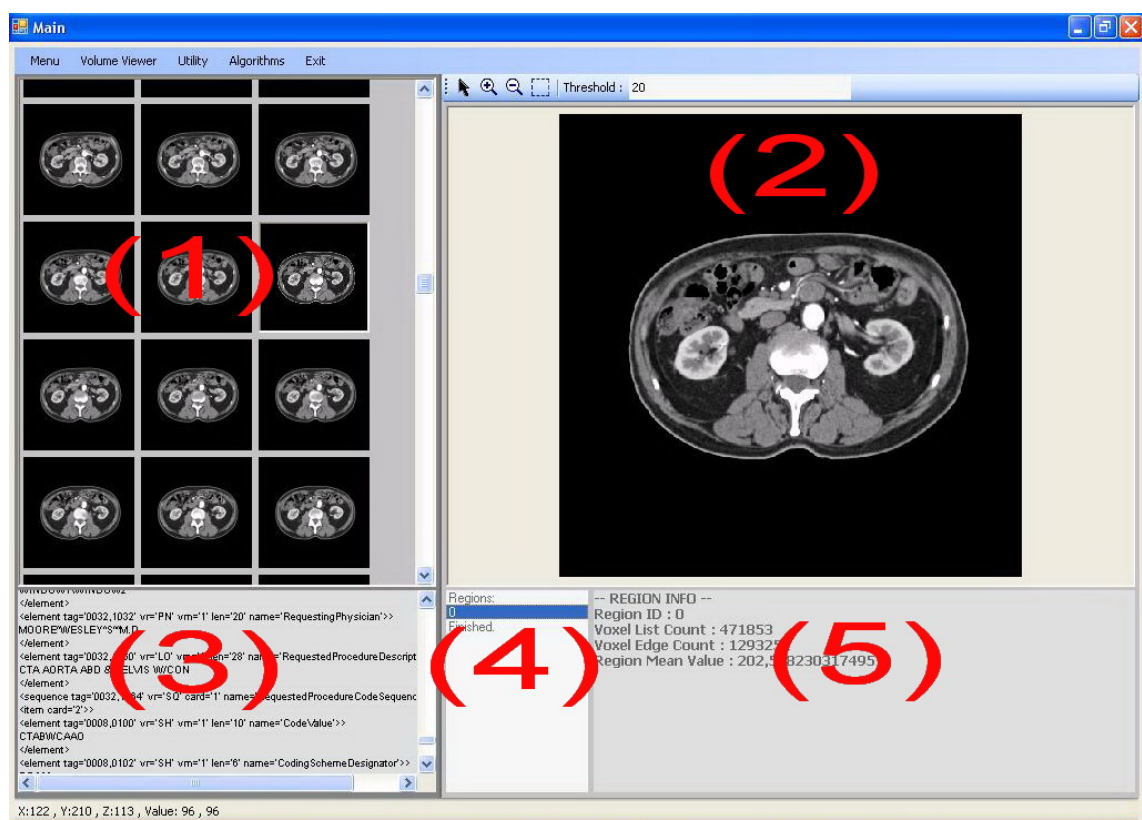


Figure 3.3 Main form of our application.

An other important form of our application is showed in Figure 3.2. By using this form you can edit transparency and color values of 3D Volume. Shortly, Part 1 of form for setting spacing value between slices. Background color of 3D volume can be set in Part 2 with Red-Green-Blue values. Transparency values can be set in Part 3 of form and lastly color values can be set Part 4 of form. In this way, for example you can adjust intensity value from 90 to 110; 0.5 transparent and light red. By using this property of our application, 3D volume can be more realistic view.

The screenshot shows the 'Volume Settings' dialog box with the following data:

**Spacing Settings:**

- X Spacing: 5 (Double)
- Y Spacing: 5 (Double)
- Z Spacing: 4 (Double)

**Background Color Settings:**

- BackGround R: 0 (0 - 255)
- BackGround G: 0 (0 - 255)
- BackGround B: 0 (0 - 255)

**Opacity Settings Table:**

Lower Bound	Upper Bound	Opacity Value	Use Linear
0	40	0.0	False
41	199	0.001	False
200	220	0.05	False
221	254	0.001	False
255	255	0	False
300	300	0.05	False

**Color Settings Table:**

Lower Bound	Upper Bound	R	G	B	Use Linear
200	220	1.0	0.0	0.0	False
255	255	0.49019...	0.49019...	0	False
300	300	0	0	1	False

Figure 3.4 Form of editing transparency and color values of 3D Volume.

### 3.2 Data Sets

We tested our algorithm on four different datasets. One of them is for testing. The other three are real Computed Tomography datasets, mainly medical datasets. Future part of the thesis, we let CT denote the Computed Tomography, SDS denotes Sample Dataset, DS1 denotes Medical Dataset 1, DS2 denotes Medical Dataset 2 and lastly DS3 denotes Medical Dataset 3. In Table 3.1 we show some property of our datasets.

### 3.2.1 *Sample Dataset*

This data set is a volume of 50 BMP images. Which size are 50×50 (width×height of an image). There are 3 objects; One cube, one sphere and one rectangle prism. The amount of contrast difference is high between all objects. Salt & Paper noise was included. Firstly, we run all algorithm on this dataset, because we can see the desired results clearly. But we know that, structure of this dataset very different from CT datasets, because CT datasets doesn't include high contrast difference between tissues. Some screen shots are given in Figure 3.3

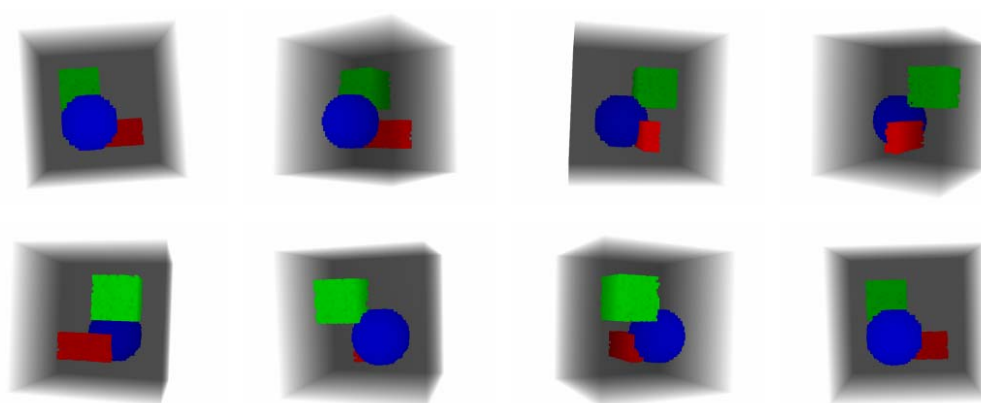


Figure 3.5 Sample screen shots of our Sample Dataset

### 3.2.2 *Medical Datasets*

DS1 is the abdomen part of the body between the pelvis and the thorax. For this dataset three different noticeable regions are kidneys , urinary bladder and bone tissue (Basin Bone & Backbone).

DS2 is the head part of the body between the sholder and the skull. For this dataset two different noticeable regions are Brain & Spinal Cord and Skull.

Like DS1, DS3 is the abdomen of the body between the pelvis and the thorax. For this dataset three different noticeable regions are kidneys , liver and bone tissue (Backbone & Ribes). Some feature of those datasets are given in Table 3.1. Figure



3.4 , Figure 3.5 and Figure 3.6 show some screen shots of in order DS1, DS2 and DS3.

Table 3.1 Some property of our Datasets.

	<i>DS1</i>	<i>DS2</i>	<i>DS3</i>
<i>Modality</i>	CT	CT	CT
<i>Manufacturer</i>	Siemens	Philips	-
<i>Number of Slice</i>	288	460	96
<i>Format of Slice</i>	DICOM	DICOM	PGM
<i>Bits Allocated</i>	16	16	8
<i>Part of Body</i>	ABDOMEN	-	-
Row	512	512	512
Column	512	512	512
Noise	No	No	Yes



Figure 3.6 Sample screen shots of our Medical CT Dataset 1

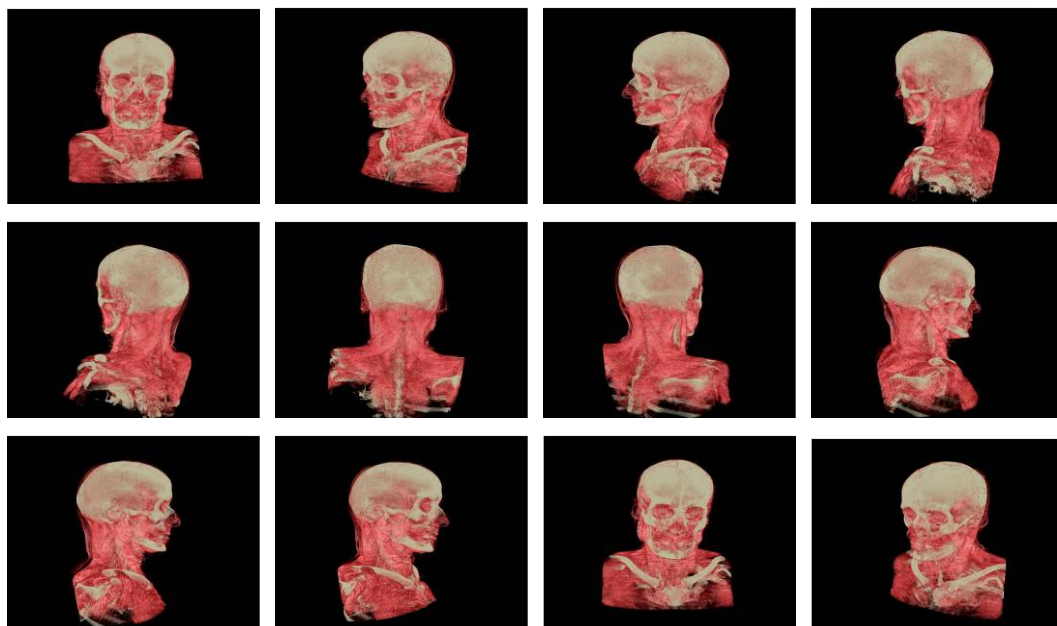


Figure 3.7 Sample screen shots of our Medical CT Dataset 2

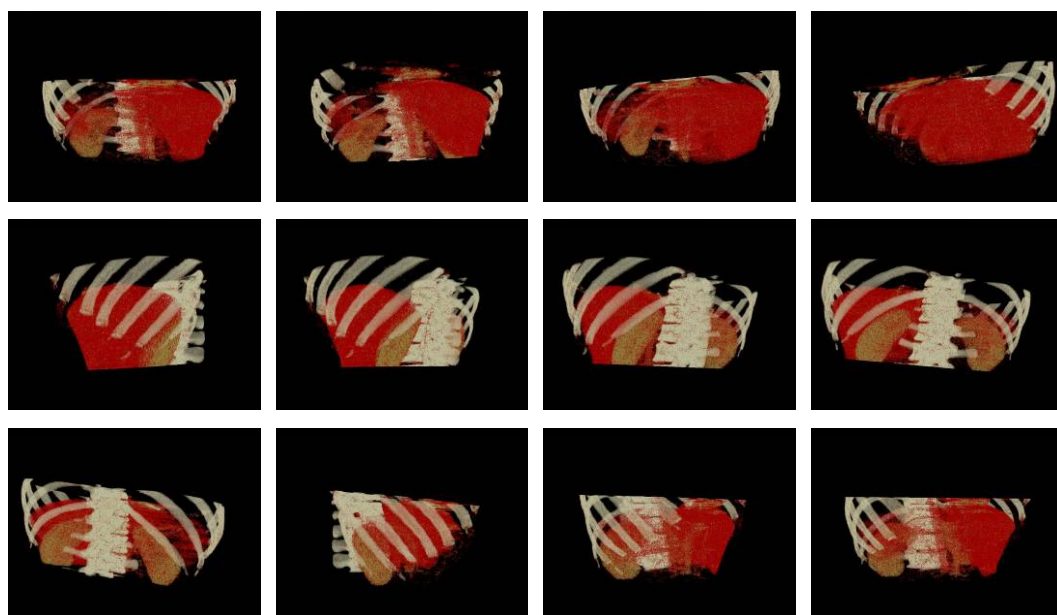


Figure 3.8 Sample screen shots of our Medical CT Dataset 3

### 3.3 Experimentations

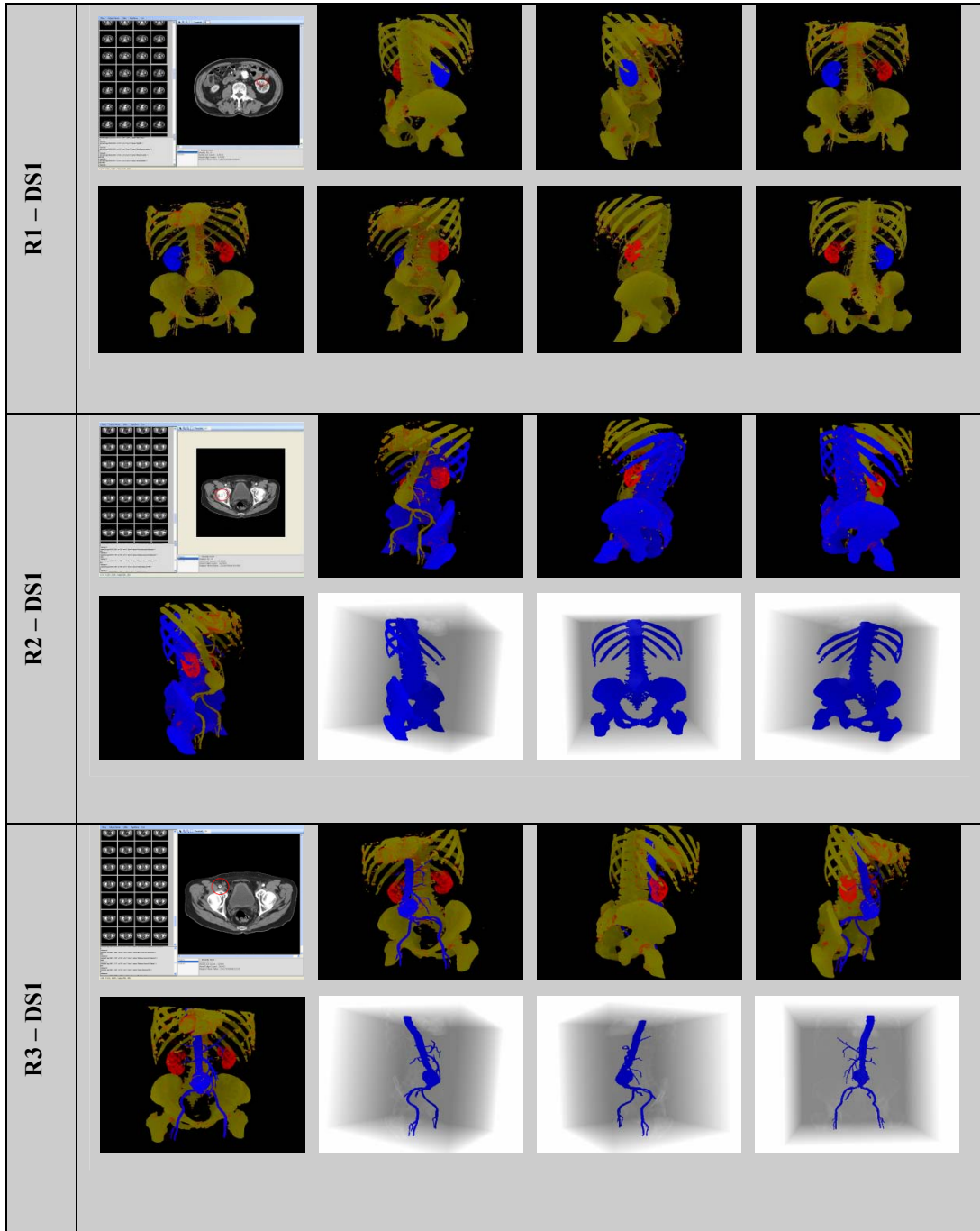
In this section, we show our algorithm's result with their Parameters by using set of screen shots, tables.

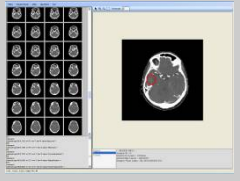
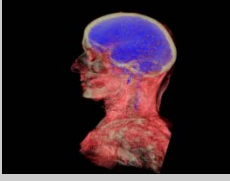
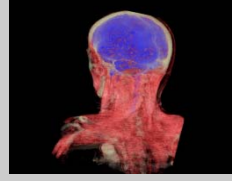
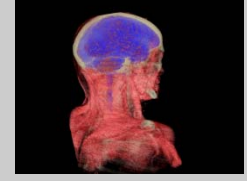
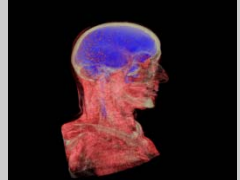



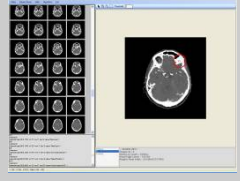
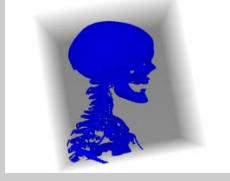
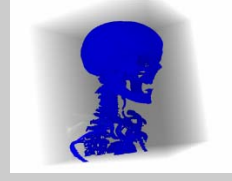


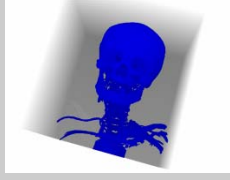

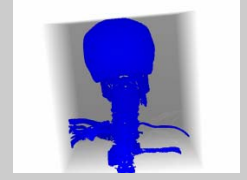
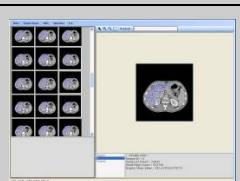
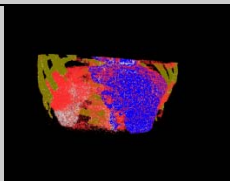
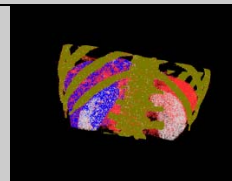
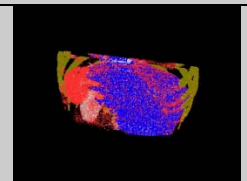
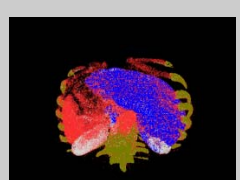
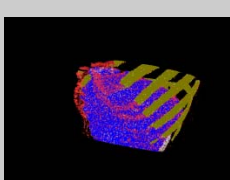
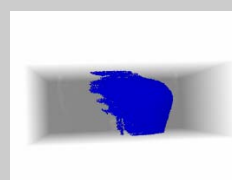
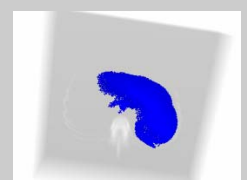
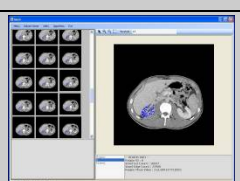
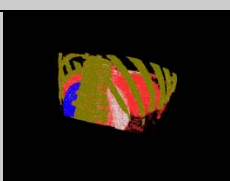
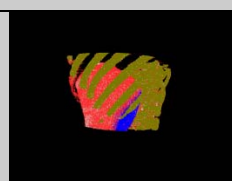

#### 3.3.1 *Seeded Region Growing*

In Table 3.2, we show Seeded Region Growing algorithm's parameters and produced region properties. Where Desired Region denotes a part of 3D volume which is interested by user. So, user select a voxel of part of volume for starting region growing. Threshold value denotes similarity value of the voxels of region ( In section 2.3.1  $T$  ). Region size means that number of voxel in the produced region. And finally, Mean denotes the mean value of the produced region's voxels.

For DS1, noticeable regions are kidneys, urinary bladder and bone tissue. This dataset is not noisy, so Seeded Region Growing Algorithm produced all desired region with suitable threshold values. For DS2, noticeable regions are skull and brain & spinal cord. This dataset is not noisy either, so Seeded Region Growing Algorithm produced all desired region with suitable threshold values. For DS3, noticeable regions are kidney, liver and bone tissue (backbone & ribs). This dataset is noisy, so Seeded Region Growing Algorithm had difficulty while producing results.

Results are shown in Figure 3.9. Parameters and produced region's properties are given in Table 3.2.



<p>R1 – DS2</p>				
				
<p>R2 – DS2</p>				
				
<p>R1 – DS3</p>				
				
<p>R2 – DS3</p>				

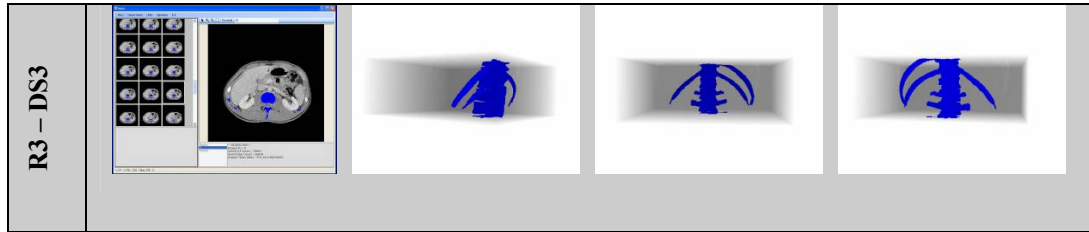


Figure 3.9 Sample screen shots of Seeded Region Growing Algorithm for DS1, DS2 and DS3. Where R1 denotes Result 1 , R2 denotes Result 2 and R3 denotes Result 3. Simply, for instance R1-DS1 denotes Result 1 for Dataset 1, and so on.

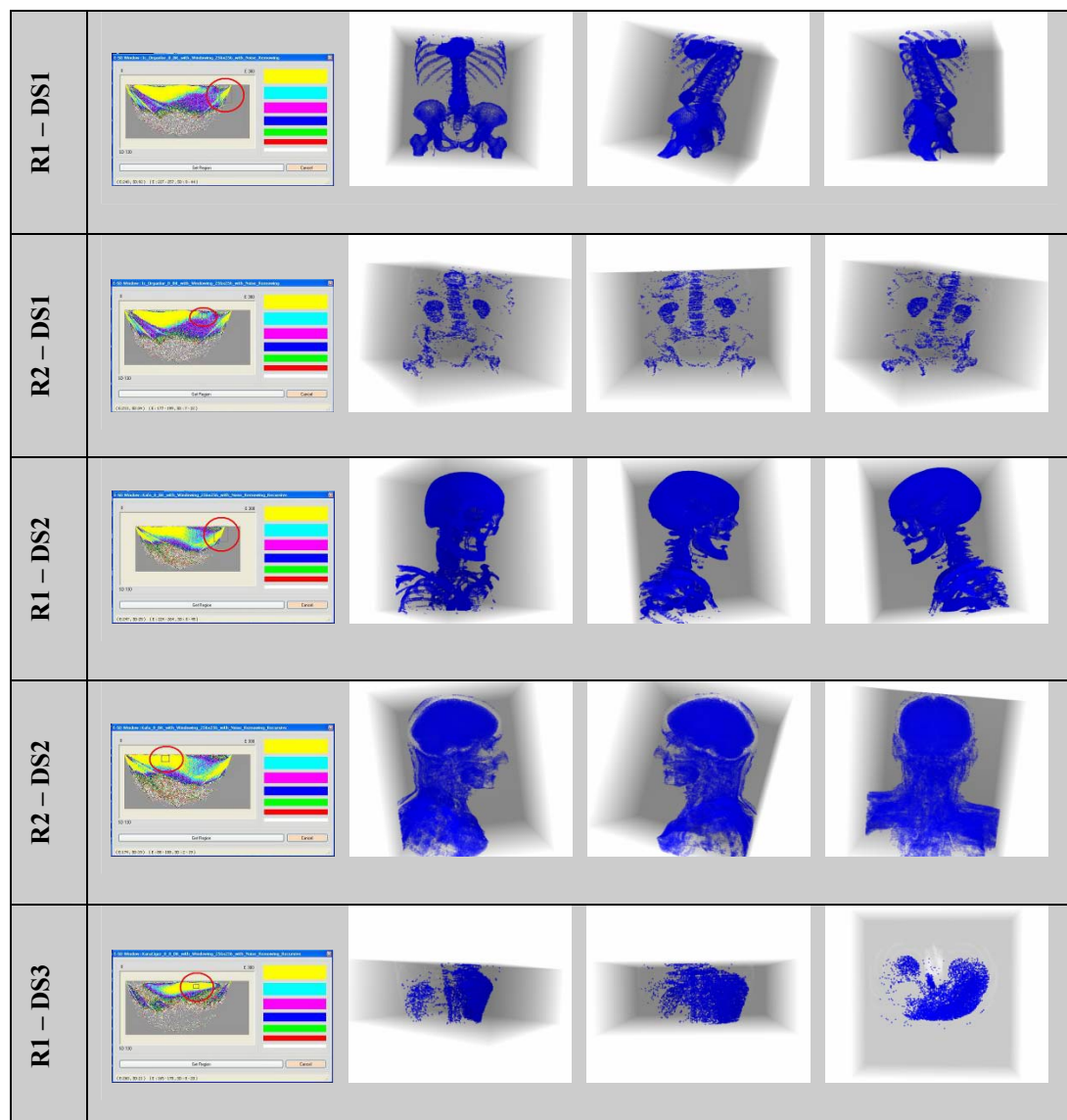
Table 3.2 Seeded Region Growing Algorithm's Parameters and Results.

	<i>DS1</i>			<i>DS2</i>		<i>DS3</i>		
	<i>Result 1</i>	<i>Result 2</i>	<i>Result 3</i>	<i>Result 1</i>	<i>Result 2</i>	<i>Result 1</i>	<i>Result 2</i>	<i>Result 3</i>
<i>Desired Region</i>	Right Kidney	Bone Tissue	Urinary Bladder	Brain & Spinal Cord	Skull	Liver	Kidney	Bone Tissue
<i>Threshold Value</i>	20	10	10	2	2	4	12	25
<i>Region Size</i>	13438	279538	52984	555466	938161	72042	16697	78894
<i>Mean Value</i>	203.72	254.64	254.75	96.4	254.98	192.2	213.5	253.4
<i>Result</i>	Satisfying	Satisfying	Satisfying	Satisfying	Satisfying	Approximatly shape of liver is shown but region is not clear.	Satisfying	Satisfying

### 3.3.2 Weibull E-SD Fields

For E-SD Fields Algorithm, E-SD values ( $e_1$ ,  $e_2$ ;  $d_1$ ,  $d_2$ ) define a movable rectangle, called a window in the E-SD field. The suit E-SDs ( $e_2 > e_1$ ;  $d_2 > d_1$ ) are determined through user interaction by moving its left-top and/or right-bottom vertices. In following sections, for all medical data sets, we show  $e_1$ ,  $e_2$ ,  $d_1$  and  $d_2$  values in Table 3.3 and their results with sample screen shots.

Results are shown in Figure 3.10 and algorithm's parameters and results are given in Table 3.3.





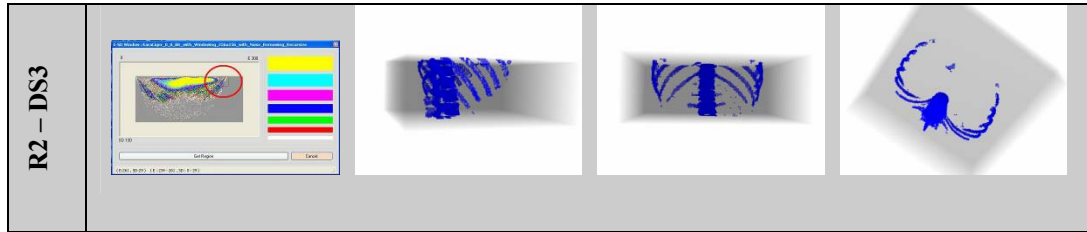


Figure 3.10 Sample screen shots of Seeded Region Growing Algorithm for DS1, DS2 and DS3. Where R1 denotes Result 1 , R2 denotes Result 2 and R3 denotes Result 3. Simply, for instance R1–DS1 denotes Result 1 for Dataset 1, and so on.

Table 3.3 Weibull E-SD Fields Algorithm's Parameters and Results.

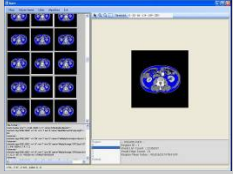
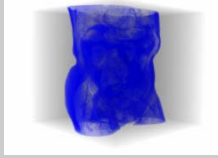
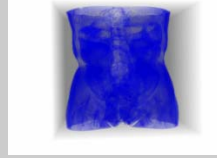
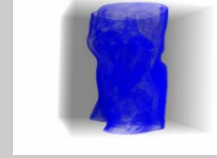
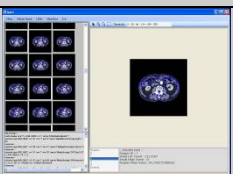
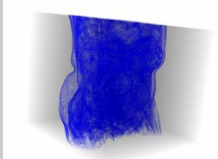
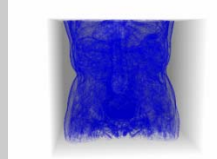
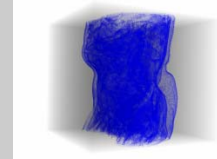
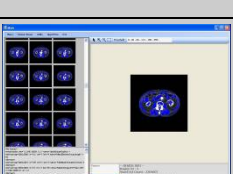
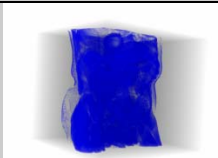
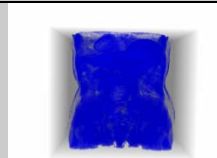
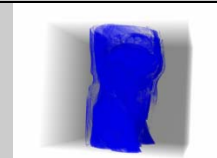
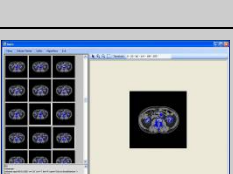

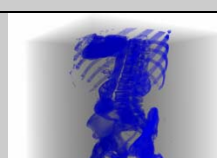
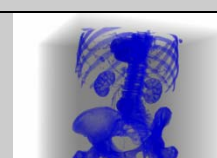
	<i>DS1</i>		<i>DS2</i>		<i>DS3</i>	
	<i>Result 1</i>	<i>Result 2</i>	<i>Result 1</i>	<i>Result 2</i>	<i>Result 1</i>	<i>Result 2</i>
$(e_1, e_2; d_1, d_2)$	(227, 257; 0, 44)	(177, 199; 7, 22)	(224, 264; 0, 45)	(88, 108; 2, 19)	(165, 178; 8, 20)	(234, 261; 0, 29)
<i>Result</i>	Bone Tissue and Urinary Bladder are in same region.	Kidneys are shown but some part of Bone Tissue too.	Bone Tissue, satisfying.	Brain is shown but some part of tissue too.	Liver is shown but some part of Kidneys too.	Bone Tissue.

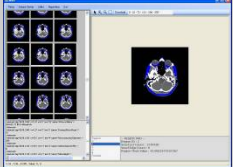
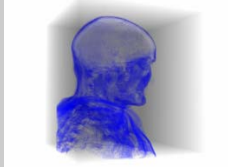
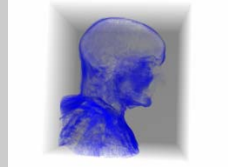
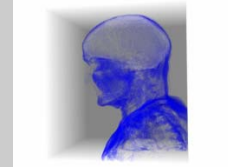
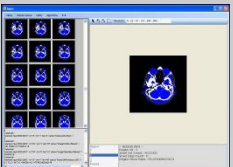
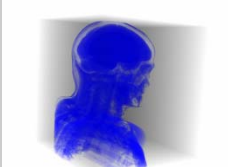
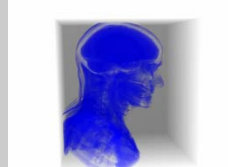
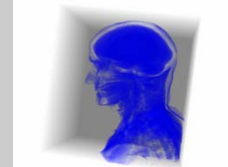
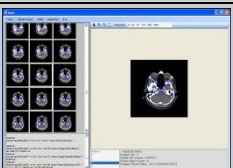
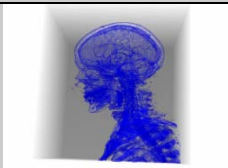
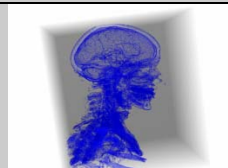
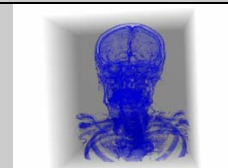
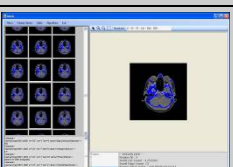

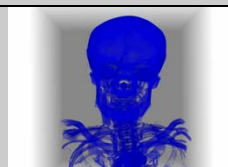
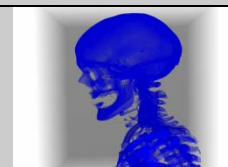
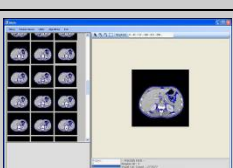

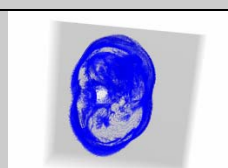
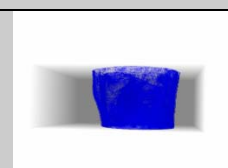
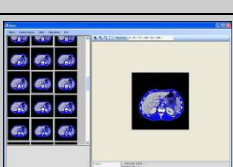
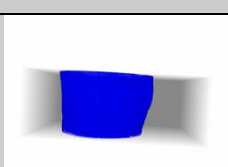
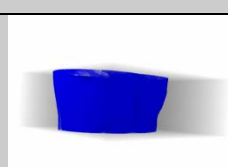
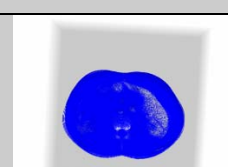
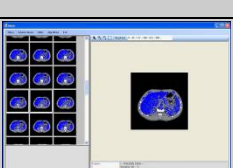
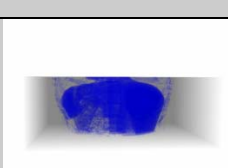
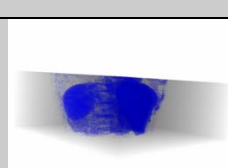
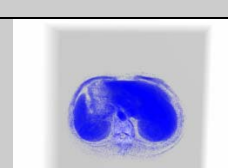
### 3.3.3 Automatic Multilevel Thresholding by using OTSU Method

For using OTSU threshold determination, firstly threshold count must be defined. OTSU Method calculates those threshold values by using entered image. After this step, threshold segmentation algorithms produced  $T + 1$  regions, where  $T$  equal to desired thresholds count. For our application  $T = 4$ .

For DS1 calculated OTSU Threshold values are 20 , 66 , 114 , 189, for DS2 calculated OTSU Threshold values are 22 , 72 , 131 , 206. And finally, for DS3 calculated OTSU Threshold values are 42 , 111 , 160 , 213.

Results are shown in Figure 3.11 and algorithm's parameters and results are given in Table 3.4.

R1 – DS1				
R2 – DS1				
R3 – DS1				
R4 – DS1				

R1 – DS2				
R2 – DS2				
R3 – DS2				
R4 – DS2				
R1 – DS3				
R2 – DS3				
R3 – DS3				

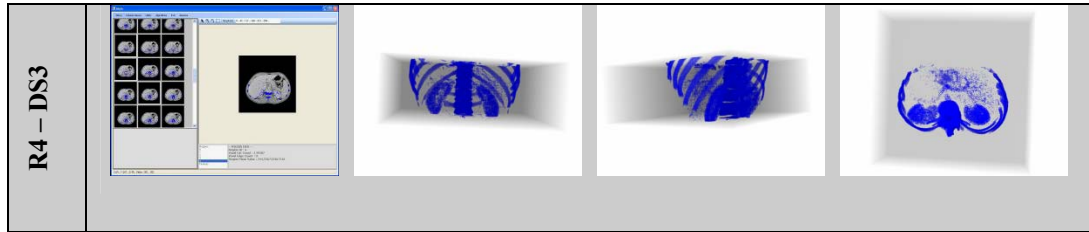


Figure 3.11 Sample screen shots of Thresholding Algorithm for DS1, DS2 and DS3. Where R1 denotes Result 1 , R2 denotes Result 2 and R3 denotes Result 3. Simply, for instance R1-DS1 denotes Result 1 for Dataset 1, and so on.

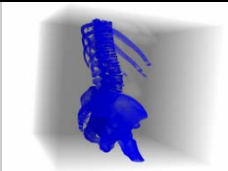
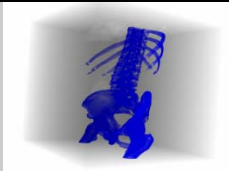
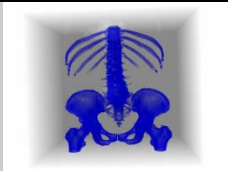
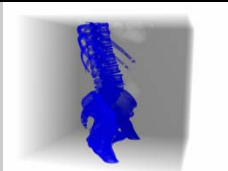
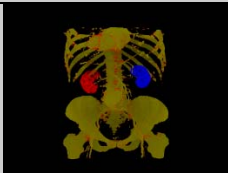
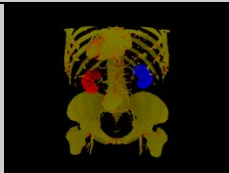

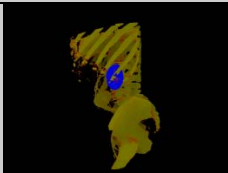
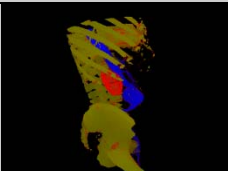
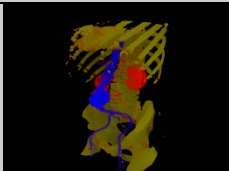
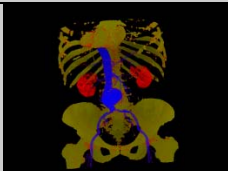
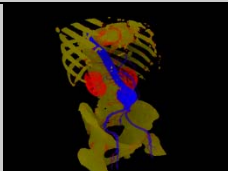
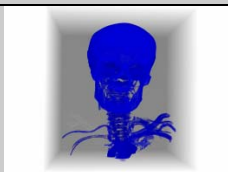
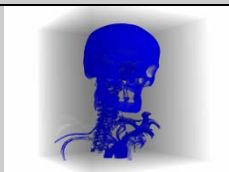
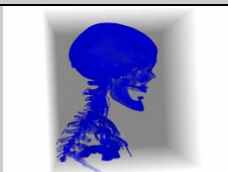
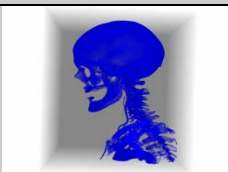
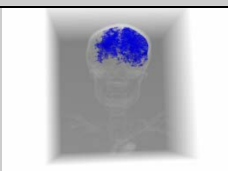
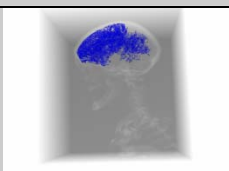
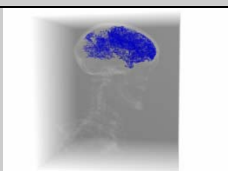
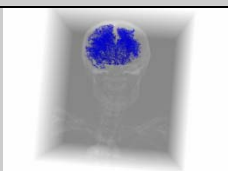
Table 3.4 Multi Level Thresholding Algorithm's Parameters and Results.

	<i>DS1</i>				<i>DS2</i>				<i>DS3</i>			
	<i>Result 1</i>	<i>Result 2</i>	<i>Result 3</i>	<i>Result 4</i>	<i>Result 1</i>	<i>Result 2</i>	<i>Result 3</i>	<i>Result 4</i>	<i>Result 1</i>	<i>Result 2</i>	<i>Result 3</i>	<i>Result 4</i>
<i>Threshold Interval</i>	20 - 66	66 - 114	114 - 189	189 - 255	22 - 72	72 - 131	131 - 206	206 - 255	42 - 111	111 - 160	160 - 213	213 - 255
<i>Region Size</i>	2236657	113167	2204607	644281	749430	4625302	649917	1253204	277677	864409	648968	139387
<i>Mean Value</i>	40.8	91.75	136.79	242.02	44.96	99.4	162.5	249.88	84.03	138.62	181.25	244.55

### 3.3.4 Unseeded Region Growing

As we mentioned in Section 2.3.4 , Unseeded Region Growing Algorithm requires a start point and threshold values. For all Datasets, we used  $(x, y, z) = (0, 0, 0)$  as a start point. In following section result of Unseeded Region Growing Algorithm is shown with threshold values and region properties.

Results are shown in Figure 3.12 and algorithm's parameters and results are given in Table 3.5. Where *Region Count* means that, number of regions which was produced by Unseeded Region Growing Algorithm.

R1 – DS1				
R2 – DS1				
R3 – DS1				
R1 – DS2				
R2 – DS2				

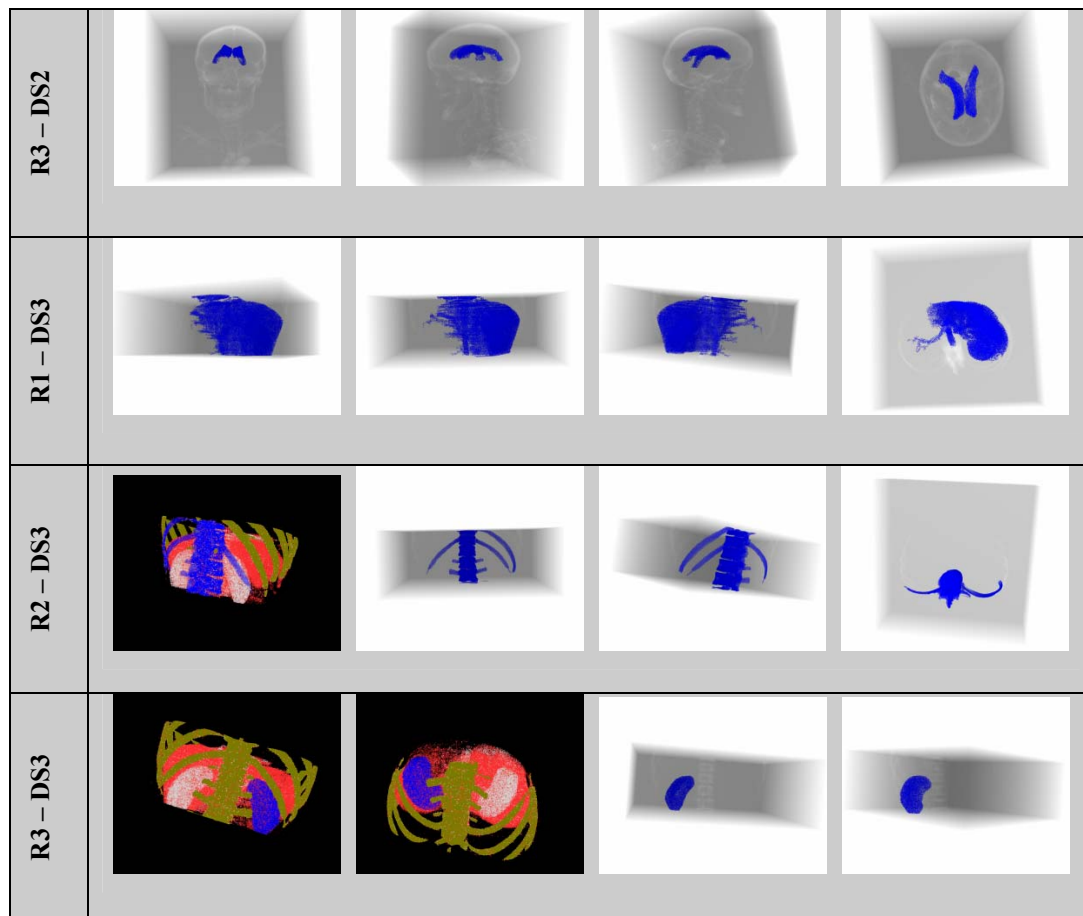


Figure 3.12 Sample screen shots of Unseeded Region Algorithm for DS1, DS2 and DS3. Where R1 denotes Result 1 , R2 denotes Result 2 and R3 denotes Result 3. Simply, for instance R1–DS1 denotes Result 1 for Dataset 1, and so on.



Table 3.5 Unseeded Region Growing Algorithm's Parameters and Results.

	<i>DS1</i>			<i>DS2</i>			<i>DS3</i>		
	<i>Result 1</i>	<i>Result 2</i>	<i>Result 3</i>	<i>Result 1</i>	<i>Result 2</i>	<i>Result 3</i>	<i>Result 1</i>	<i>Result 2</i>	<i>Result 3</i>
<i>Threshold Value</i>	20			5			10		
<i>Region Count</i>	14			24			11		
<i>Region Size</i>	301757	13836	55170	954054	81209	32096	110944	73297	16700
<i>Mean Value</i>	253.52	202.25	254.17	254.92	89.97	78.62	196.35	254.7	201.86
<i>Result</i>	Bone Tissue	Left Kidney	Urinary Bladder	Skull	Part of Brain	Part of Brain	Liver	Bone Tissue	Kidney

### 3.4 Briefly Reviewing Of Comparison Of Algorithms

In Table 3.6, we try to emphasize comparison of our algorithms, briefly. Where SRG denotes Seeded Region Growing Algorithm, OTSU denotes Automatic Multilevel Thresholding by using OTSU Method, E-SD denotes Weibull E-SD Fields Algorithm and USRG denotes Unseeded Region Growing Algorithm. We use some scales for expressing algorithm's differences. Implementation values are changing between 1 and 5, here 1 is very hard and 5 is very simple. Noise Sensitivity denotes durability of the algorithm to the noise. Values are changing between 1 to 4, here 1 is algorithm is affected from noise too much and 4 is algorithm is not affected from noise. Time performance denotes algorithms running time and changing between 1 and 5. Where 1 is very fast and 5 is very slow. Result performance denotes algorithms success, mainly relation between desired region and produced region. Values changes 1 to 10. Where 10 is perfect and 1 is very bad. And finally, General Result denotes all opinion of us about related algorithm. Here 4 denotes worst one and 1 is denotes the best one. All this scores are our opinion about related algorithms.

Table 3.6 Comparison of Segmentation Algorithms, briefly.

<i>Algorithm</i>	<i>Implementation</i>	<i>Noise Sensitivity</i>	<i>User Interaction</i>	<i>Time Performance</i>	<i>Result Performance</i>	<i>General Result</i>
<i>SRG</i>	4	1	Required	2	8	1
<i>OTSU</i>	3	2	NOT Required	5 ( for $T = 4$ )	6	3
<i>E-SD</i>	2	3	Required	4	3	4
<i>USRG</i>	4	1	NOT Required	2	7	2

## **CHAPTER FOUR**

### **CONCLUSION**

In this study, we implemented four different 3D segmentation algorithms on the three different medical datasets and provide a comparison among them. We showed all segmentation results by using sample screen shots and segmentation Parameters. As a result we can explain advantages and disadvantages of our algorithms as follows;

For Seeded Region Growing Algorithm; because of the structure of Seeded Region Growing Algorithm, although it was not produced clear results on noisy images, it produced the right segmentation on the unnoisy images with the suitable threshold values. In addition to this, another advantage of this algorithm is simple implementation. The other disadvantage of Seeded Region Growing Algorithm is the need of a seed point namely user interaction. Result of this algorithms can be improved by using suitable noise filter to the given image. For our datasets, except Medical Dataset 3 it was produced satisfactory results.

Another semi-automatic segmentation algorithm is called Weibull E-SD Fields Algorithm. In spite of Weibull E-SD Fields Algorithm produced very good results on the test datasets, it has not good results on the medical images which does not have too much contrast differences between the segments. The main reasons of this unexpected result is that, E-SD Fields Algorithm consider that, if there is two K-Voxels and they have nearly same E and SD values, they should be in the same segment. But while doing that, the algorithm do not consider the location information of K-Voxels's voxels on the 3D volume. Except from these, algorithm was not affected from noise, because of its own noise removing step. Another disadvantage of this algorithm, while selection of the region from E-SD plot, there is no striking point. So it is hard to finding the desired region. Algorithm's result can be improved, using K-Voxel's location information while drawing E-SD plot.

The base of thresholding algorithms is to calculate the right threshold values with the consideration of minimum and maximum points of the histogram of the image and to classify the voxels between those thresholds interval. We had the proper threshold values on the histogram of 3D images by using the OTSU Method. In the previous works, OTSU Method was used for 2D images segmentation, but in this thesis we applied it on 3D Volume. The segments, which we had as results, are considered successful.

And Finally, UnSeeded Region Growing Algorithm, which is the prolongation of Seeded Region Growing Algorithm, is failed on noisy images too, therefore we have used the same threshold values from Seeded Region Growing Algorithms in our tests. The region count was more than we had expected, main reason of that, even seed points are changeable, threshold value is same for all seed points. In our opinion, threshold value should be updated for every new seed point. For our datasets, however the region count was more than we had expected , results were successful.

As a future work, we plan to increase the number of segmentation algorithms to test on more medical datasets. While doing that, we will try to make our application more user friendly.

**REFERENCES**

- Converting CT Data to Hounsfield Units. (n.d.). Retrieved December 12, 2006, from [http://www.dfanning.com/fileio\\_tips/hounsfield.html](http://www.dfanning.com/fileio_tips/hounsfield.html)
- Hu, J. & Razdan, A. (2003) Volumetric Segmentation Using Weibull E-SD Fields :  
IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER  
GRAPHICS, VOL. 9, NO. 3,
- Hounsfield Scale. (n.d.). Retrieved December 12, 2006, from  
[http://en.wikipedia.org/wiki/Hounsfield\\_scale](http://en.wikipedia.org/wiki/Hounsfield_scale)
- Ilic, S. & Ulicny, B Seeded Region Growing Method for Image Segmentation.  
Retrieved April 07, 2006,  
<http://ligwww.epfl.ch/~silic/predocschool/ComputerVision/cvision.html>
- Lakare, S. (2000) 3D Segmentation Techniques for Medical Volumes : State  
University of New York
- Liao, P. & Chen, T. & Chung, P.A (2001) : JOURNAL OF INFORMATION  
SCIENCE AND ENGINEERING 17, 713-727
- Lin, Z. & Chen, T. & Jesse, J. & Talbot, H. (2001) : Unseeded region growing for 3D  
image segmentation : Australian Computer Society, Inc.
- The DICOM Standard. Retrieved December 12, 2006 , from  
<http://dicom.offis.de/dcmthk.php.en>