DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ABDOMINAL IMAGE SEGMENTATION AND VISUALIZATION USING HIERARCHICAL NEURAL NETWORKS

by
M. Alper SELVER

March, 2010
İZMİR

ABDOMINAL IMAGE SEGMENTATION AND VISUALIZATION USING HIERARCHICAL NEURAL NETWORKS

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

by
M. Alper SELVER

March, 2010
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**ABDOMINAL IMAGE SEGMENTATION AND VISUALIZATION USING HIERARCHICAL NEURAL NETWORKS**" completed by **M. ALPER SELVER** under supervision of **PROF.DR. CÜNEYT GÜZELİŞ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Cüneyt GÜZELİŞ

Supervisor

Prof. Dr. Oğuz DİCLE

Thesis Committee Member

Asst. Prof. Dr. Güleser K. DEMİR

Thesis Committee Member

Examining Committee Member

Examining Committee Member

Prof. Dr. Cahit HELVACI
Director

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to Prof. Dr. Cüneyt GÜZELİŞ for his supervision and guidance not only for my PhD study but for my whole graduate years from the very first day I entered Dokuz Eylül University.

I am very grateful for the kind assistance and guidance of Prof. Dr. Oğuz DİCLE and Assist. Prof. Dr. Güleser K. DEMİR through my thesis study and our project.

My deepest appreciation to Prof. Dr. Walter HILLEN and Dr. Felix FISCHER for their collaboration and friendship during our studies in FH-Juelich, Germany. Speaking of Germany, I would like to thank Dr. Hatice DOĞAN for her patience and friendship during our studies and stay in Juelich.

Many thanks to Pınar PAYZİN and Özgür ÖZDEMİR for continuously helping me collecting information and data from Radiology.

I also would like to thank to Prof. Dr. Süleyman MEN, Prof. Dr. Mustafa SEÇİL, Prof. Dr. Yiğit GÖKTAY, Dr. Emel ONUR, Dr. Sinem DİLŞEN, Dr. Ömür GENCEL for their time and help in the evaluation of clinical studies.

My sincere thanks to Aykut KOCAOĞLU for his endless efforts and studies which mean a lot in the completion of this thesis.

My special thanks to Assoc. Prof. Dr. Yeşim ZORAL and Assist. Prof. Dr. Serkan GÜNEL in helping me broaden my view and perspective in many ways.

I would like to thank Yalçın İŞLER, Mehmet ÖLMEZ, Yakup KUTLU, and my other colleagues for their encouragement and support.

To Ayhan YAZICI, for his persistent friendship.

My sincere thanks to my parents-in-law, Hale and Osman, and my sister-in-law Öykü, for their patience, understanding and encouragement.

My heartiest gratitude to my parents, Yusuf and Sacide, and my sister Aslı, for their constant love and everlasting care in me.

Last but not least, my deepest thanks to my wife, Özlem, for her love, trust, support and patience in all those years of hard work.

M. Alper Selver

ABDOMINAL IMAGE SEGMENTATION AND VISUALIZATION USING HIERARCHICAL NEURAL NETWORKS

ABSTRACT

Medical imaging modalities can provide very detailed and informative mappings of the anatomy of a subject. Therefore, diagnostic imaging has become an important tool in medicine by increasing knowledge of normal and pathological anatomy, so helping diagnosis and planning treatment. These detailed and informative mappings can be processed to extract the information of interest instead of dealing with whole data. Development of tools and techniques to accomplish information extraction and rendering that information can be grouped under the fields of image segmentation and visualization. These two fields are strongly related with each other and they play a vital role in numerous radiological imaging applications such as the quantification of tissue volumes, diagnosis, localization of pathologies, study of anatomical structures, treatment planning, computer aided surgery and medical education.

Segmentation depends highly on the specific application, imaging modality, and other factors such as artifacts, motion, partial volume effects and noise. Imaging of human abdomen is one of the challenging application areas of segmentation due to the highly overlapping intensity ranges of organs of interest. Therefore, selection and development of an appropriate segmentation method depends on the requirements of the problem and organ of interest.

On the other hand, the goal of medical visualization is to produce clear and informative pictures of the important structures in a data set but simple approaches have limited performance on visualization of abdomen. Volume visualization can be used either directly with the whole volume data or after a segmentation algorithm. For both cases, volume rendering is an important technique since it displays 3-Dimensional images directly from the original data set and provides "on-the-fly" combinations of the selected image transformations such as opacity and color. The only interactive part during the generation of the volume rendered medical images is the Transfer Function

specification, therefore it is important to design effective tools for handling this parameter.

For segmentation and visualization tasks discussed above, developing new methods, algorithms, and applications that can be used in medical image segmentation is necessary to use 3-D volume visualization more effectively in diagnosis, treatment planning etc. During the development of these methods, robust and stable query and retrieve from different storage media, ability of manipulating 2-D/3-D images and proper visualization of the results are necessary. Flexible tools and libraries are needed to revisit already-solved problems, to re-develop existing programs, or to rapidly implement and test new algorithms which can save these researchers' time and effort.

In this thesis, novel studies on segmentation, interactive visualization of medical images and studies on their implementation are presented. First of all, a robust and patient oriented segmentation algorithm is developed for pre-evaluation of liver transplantation donor candidates. For the, enhancement of the visualization of abdominal organs, a new domain and a technique for multi-stage approximation to this domain, which is then used for transfer function specification for volume rendering, are introduced. Finally, the developed liver segmentation algorithm is implemented as an application of a more general framework on object based medical image segmentation and representation.

Keywords: Abdominal imaging, segmentation, Volume rendering, Transfer function, Java, Neural networks.

HİYERARŞİK SİNİR AĞLARI İLE ABDOMİNAL GÖRÜNTÜ BÖLÜTLEME VE ÜÇ BOYUTLU GÖRÜNTÜLEME

ÖZ

Tıbbi görüntüleme ile anatomi hakkında detaylı bilgiler elde edinilebildiğinden, tanı amaçlı görüntüleme bir çok açıdan önemli hale gelmiştir. Görüntüleme cihazları tarafından sunulan veriler, tüm veri yerine ilgilenilen dokunun görüntülerde belirlenerek ayrılması suretiyle işlenebilir. Sayısal görüntü gösterimi ve işlenmesi alanındaki gelişmelerin yardımı ile de bu görüntülerin incelenmesinde yeni tekniklerin kullanılması da mümkün olmaktadır. Ayrıca bu sayısal çoklu veri dilimleri, çeşitli görüntüleme teknikleri kullanılarak üç boyutlu görüntülerin oluşturulmasında, tanı, ameliyat benzetimi ve tedavi planlama gibi alanlarda da kullanılabilmektedir. Bu işlemleri gerçekleştirecek yöntem ve araçların geliştirilmesi ve elde edilen verilen sunulması, bölütleme ve üç boyutlu görüntüleme başlıkları altında incelenmektedir. Bölütleme ve görüntüleme birbirleriyle yakın ilişkili iki alan olup, bir çok radyolojik uygulamada kullanılmaktadırlar.

Bölütlemede kullanılacak teknik, tıbbi uygulama alanına, görüntüleme cihazına ve gürültü gibi dış etkenlerden kaynaklanan bir çok faktöre bağlıdır. Örtüşen organ ve doku yoğunlukları nedeniyle de, abdominal organ görüntüleme pek çok zorluk içeren bir bölütleme uygulama alanıdır. Bu nedenle, uygun bir bölütleme yönteminin seçimi ve geliştirilmesi, bölütlenecek organın özelliklerine bağlıdır. Üç boyutlu görüntülemede ise amaç önemli doku ve organların en net ve açık olarak gösterilebilmesidir ancak benzer nedenlerden ötürü abdominal üç boyutlu görüntülemede temel teknikler yetersiz kalmaktadır. Transfer fonksiyonları, üç boyutlu görüntülerle etkileşimde renk ve opaklık gibi önemli parametrelerin belirlenmesini etkileşimli olarak sağladıklarından, bu parametrelerin belirlenmesinde etkili ve kullanışlı tekniklerin geliştirilmesi önemlidir. Hem bölütleme, hem de üç boyutlu görüntüleme için, yeni tekniklerin geliştirilmesi kadar, bu tekniklerin uygun ve kullanışlı araçlar haline getirilmesi de önemlidir. Bu tez ile, abdominal görüntülerde bölütleme, üç boyutlu görüntüleme ve bunların etkin şekilde gerçeklenmesi üzerine yeni çalışmalar sunulmaktadır.

Tez kapsamında öncelikle karaciğer donor adaylarının karaciğer hacimlerini ölçmek üzere kullanılan bir karaciğer bölütleme yöntemi geliştirilmiştir. Geliştirilen yöntem, çok katmanlı yapay sinir ağlarının adım adım eğitimi ve kullanımı ile karaciğer bölütlemedeki problemlerle baş edebilen otomatik ve uyarlamalı bir karaciğer bölütleme yöntemidir.

İkinci olarak, abdominal organların üç boyutlu görüntülenme başarımını artırmak amacıyla yeni bir teknik önerilmiş ve hacim görüntülemede transfer fonksiyonu saptanmasında kullanılmıştır. Geliştirilen yöntemde, abdominal organların, görüntüler üzerindeki özelliklerinden yararlanılarak yeni bir fonksiyon tanımlanmış ve bu fonksiyona yakınsamada hiyerarşik yapay sinir ağları kullanılmıştır.

Son olarak, geliştirilen karaciğer bölütleme yöntemi, daha genel bir çalışma olan nesne tabanlı bölütleme kapsamında gerçeklenmiştir. Eklenti bir program halinde kodlanarak bir tıbbi görüntüleme yazılımına tümleştirilen karaciğer bölütleme yöntemi, karaciğerin tüm analizinde yararlı olacak örnek eklenti programlar ile beraber kullanılarak, nesne tabanlı bölütleme ve üç boyutlu görüntülemenin sağladığı kazanımlar incelenmiştir. Bu yaklaşım, genel amaçlı bölütleme üzerine hazırlanan başka eklenti programlar ile farklı abdominal görüntülere de uygulanarak sonuçları sunulmuştur.

Anahtar sözcükler: Abdominal görüntüleme, Bölütleme, Hacim görüntüleme, Transfer fonksiyonu, Java, Sinir Ağları

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

1.1 General Scope of the Thesis

Due to the technological developments in medical imaging technology, Computed Tomography (CT), Magnetic Resonance (MR) imaging, digital mammography, and other imaging modalities can provide very detailed and informative mappings of the anatomy of a subject. According to these developments, diagnostic imaging became an important tool in medicine by increasing knowledge of normal and pathological anatomy, so helping diagnosis and planning treatment. These detailed and informative mappings, which are provided by emerging modalities, results in larger data that have increased size and number of 2-Dimensional (2-D) images for each study. This necessitates the use of computers and algorithms for processing and analysis of these data. For assisting and automating specific tasks in radiology, delineation and displaying of anatomical structures and other regions of interest are important. The tools and techniques to accomplish those tasks are grouped under image segmentation and visualization fields. These two fields are strongly related with each other and they play a vital role in numerous radiological imaging applications such as the quantification of tissue volumes, diagnosis, localization of pathologies, study of anatomical structures, treatment planning, computer aided surgery and medical education.

Segmentation depends highly on the specific application, imaging modality, and other factors such as artifacts, motion, partial volume effects and noise. For example, the segmentation of brain tissue has different requirements from the segmentation of the liver and is also different in CT and MR images. Thus, there is currently no method that provides acceptable results for all cases of medical images. Methods that work in a wider sense can be applied to a variety of data but the methods that are specialized to particular applications can achieve more accurate performances by taking prior knowledge into account. Selection of an appropriate approach to a segmentation problem, therefore, depends on the requirements of the problem. Abdominal image processing is a challenging application area of segmentation due to overlapping characteristics of organs and tissues.

On the other hand, the goal of medical visualization is to produce clear and informative pictures of the important structures in a data set. Several research activities addressing the limitations of current visualization systems are aiming to come up with new techniques which will carry on volume visualization from research and teaching hospitals to routine clinical work. Volume visualization can be used either directly with the whole volume data or after a segmentation algorithm, which eliminates the redundant data and present the data of interest only. When the whole volume data is used, volume rendering (Porter, 1984, Levoy, 1988) is an important technique since it displays 3-Dimensional (3-D) images directly from the original data set and provides "on-the-fly" combinations of the selected image transformations such as opacity and color. The only interactive part during the generation of the volume rendered medical images is the Transfer Function (TF) specification, therefore it is important to design effective tools for handling this parameter (Pfister et al., 2000). Unfortunately, finding good TFs is a very difficult task because of the availability of various possibilities and since this flexibility can not be kept in strict bounds, finding an appropriate TF for a meaningful and intelligible volume rendering is an active research field.

For segmentation and visualization tasks discussed above, developing new methods, algorithms, and applications that can be used in medical image segmentation is necessary to use 3-D volume visualization more effectively in diagnosis, treatment planning etc. During the development of these methods, robust and stable query and retrieve from different storage media, ability of manipulating 2-D/3-D images and proper visualization of the results are necessary but they can take a significant amount of time. Moreover, implementation of these tools are out of scope for the researchers dealing with segmentation and/or visualization techniques who need to focus on proving the reliability and robustness of their algorithms. Thus, flexible tools and libraries are needed to revisit already-solved problems, to re-develop existing programs, or to rapidly implement and test new algorithms which can save these researchers' time and effort.

In this thesis, novel studies on segmentation, interactive visualization of medical images and studies on their implementation are presented. The developed methods are focused on abdominal image processing and cover three main topics:

- The first topic is the development of a robust and patient oriented segmentation algorithm for pre-evaluation of liver transplantation donor candidates.
- The second topic is the enhancement of the visualization of abdominal organs by introducing a new domain and a technique for multi-stage approximation to this domain which is then used for transfer function specification for volume rendering.
- Finally, the third topic is the implementation of the developed liver segmentation algorithm as an application of a more general framework on medical image segmentation and representation.

1.2 Specific Aims

The specific aims of this thesis, which covers the studies to improve the segmentation, visualization and analysis of abdominal image processing problems, are introduced in the following sections.

1.2.1 Pre-evaluation of Liver Transplantation Donors

The first subject of this thesis consists of the study about the development of a method for automatic segmentation of liver in contrast enhanced CT images. This is a very important procedure since the results are used for the measurement of the liver volume and analysis of the liver vasculature that are important stages to decide whether a candidate for transplantation is suitable or not.

Routine preoperative evaluation of donors requires both CT (Flohr et al., 2000) and CT with contrast medium injection, namely CT-Angiography (CTA), which are currently the most widely used radiographic techniques for the rendering of liver parenchyma, vessels and tumors in living liver transplantation donors. However, due to gray level similarity of adjacent organs, injection of contrast media and partial volume effects; robust segmentation of the liver is a very difficult task. Moreover, high variations in liver position, different image characteristics of different CT modalities and atypical liver shapes make the segmentation process even harder. The strategy of this study for overcoming these difficulties involves a segmentation method which does not

utilize a common parameter set found from all patient datasets. Instead, the method is capable of adapting the parameter set to each patient. The main reason for this approach is that the ranges of the parameter values differ significantly from patient to patient, and these wide ranges decrease the efficiency of the method when one utilizes a common parameter set for all patients. Thus, a method, which examines and adapts its parameters according to each patient, is proposed and the approach is named as patient-oriented segmentation. For qualifying 'patient oriented', the algorithm learns data set characteristics in parallel to segmentation process, and adapts its parameters to these characteristics.

The developed iterative segmentation algorithm combines classification of pixels (using an unsupervised clustering method i.e. K-means) with adjacent slice information (obtained by skeletonization) via morphological reconstruction. A more complex classifier (Multi Layer Perceptron network - MLP) is used for the datasets where the K-means clustering gives insufficient results. Here, the neural network is designed to classify features extracted from the current and adjacent (previously segmented) slices and therefore intrinsically robust to gray level and shape variations. The decision between using either K-Means or MLP is also done automatically by the algorithm.

The developed algorithm gives sufficient performance for different modalities, varying contrast, dissected liver regions and atypical liver shapes. Results indicate that challenging difficulties explained before can be handled properly using the developed method and it is also clinically feasible in terms of processing time.

1.2.2 Transfer Function Specification for Abdominal Visualization

The medical visualization, which aims to produce clear and informative pictures of the important structures in a data set, requires extensive user interaction. One of the important advantages of volume rendering (Drebin, Carpenter, & Hanrahan, 1988) is that combinations of selected parameters, such as opacity and color, can be determined during the rendering pipeline. During the generation of volume rendered medical images, TF specification is the step where these adjustments can be done. Therefore, it is crucial and important to design accurate TFs to produce meaningful and intelligible 3-D images. However TF design is a very difficult task because of the availability of

various possibilities. Since this flexibility can not be kept in strict bounds, specification of an appropriate TF is a challenging problem especially when there is no initial TF design prior to the optimization process.

Volume rendering would be used more often in clinical practice if the complexity of interaction (i.e. setting a TF for volume rendering) becomes less. To reduce the complexity of TF design, a semi-automatic method for TF initialization and a new, effective and interactive domain for TF optimization is introduced in this thesis. The proposed method is based on a Volume Histogram Stack, i.e. VHS, instead of conventional volume histogram and handles TF specification as a (vector-valued) function approximation problem where the domain is the 2-D input space of Hounsfield value and slice number and the range variables are opacity and color. The method automates and simplifies the optimization of a TF.

The newly introduced VHS data model allows the detection of tissues both in calibrated (i.e. CT) and uncalibrated (i.e. MR) medical datasets. As a consequence of the fact that each slice histogram is represented separately, VHS preserves inter-slice spatial domain knowledge, so it exploits more priori information. It also demonstrates changes in the gray values through the series of slices, thus including information on local histogram distributions of the tissues. In other words, VHS can represent the intensity values of the tissues as well as their spatial information and local distributions which are not available in conventional volume histograms.

1.2.3 Integrating Developed Methods to a Medical Image Viewer

Developing new techniques, algorithms, and applications that can be used in medical image segmentation is necessary to be able to use 3-D volume visualization in diagnosis, treatment planning etc. The development of the complete package, which required robust and stable query and retrieve from different storage media, manipulate 2-D/3-D images, convert images, and effectively visualize them can take a significant amount of time. Moreover, it is out of scope for the researchers dealing with segmentation algorithms, who need to focus on proving the reliability and robustness of their algorithms. Flexible tools and libraries are needed to revisit already-solved problems, to

re-develop existing programs, or to rapidly implement and test new algorithms which can save these researchers' time and effort.

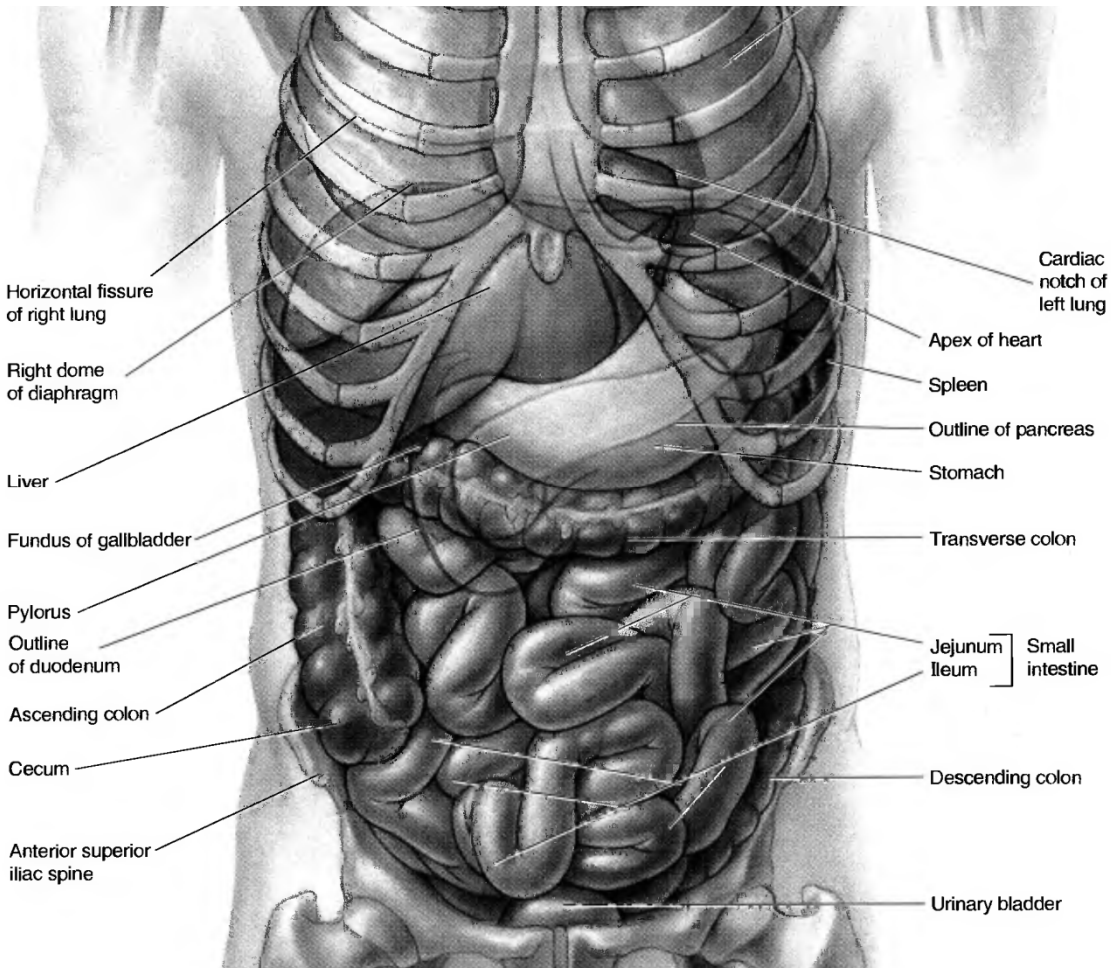
The goal of this part of the thesis is to present developments and refinements of segmentation algorithms in particular applications to abdominal images. The method for achieving this is to provide access to the 3-D rendering (i.e. Volume Rendering (VR), Surface Rendering (SR) and Maximum Intensity Projection (MIP) (Robb, 1995)) capabilities that can be used to visualize the results of new segmentation algorithms. It benefits practitioners by allowing them to make use of their advanced algorithms developed by different tools (i.e. MatLab, Java, Insight Registration and Segmentation Toolkit (ITK) (Ibanez, & Schroeder, 2005; Martin, Ibanez, Avila, Barre, & Kaspersen, 2005)) with a low learning curve and it can assist algorithm developers by proving a simple application. Thus, the developers are enabled to easily and routinely make use of their algorithms with little to no learning curve from within a Digital Imaging and Communications in Medicine (DICOM) (American College of Radiology, National Electrical Manufacturers Association, 2005) application. As opposed to direct use of the ITK and Java, researchers do not need to deal or spend time to gain programming experience on loading data, displaying images or showing the results in a proper way which requires a high experience on Visualization Toolkit (VTK) (Schroeder, Martin, & Lorensen, 1998) and Java due to the due to various cases of DICOM format and different medical applications.

The proposed architecture of implementation mechanisms are used to develop plug-ins for segmentation. These plug-ins consist of general purpose segmentation plug-ins, task specific ones and interactive visualization plug-ins. The advantages of using these different types of plug-ins are compared using abdominal image processing applications and development and programming issues are discussed.

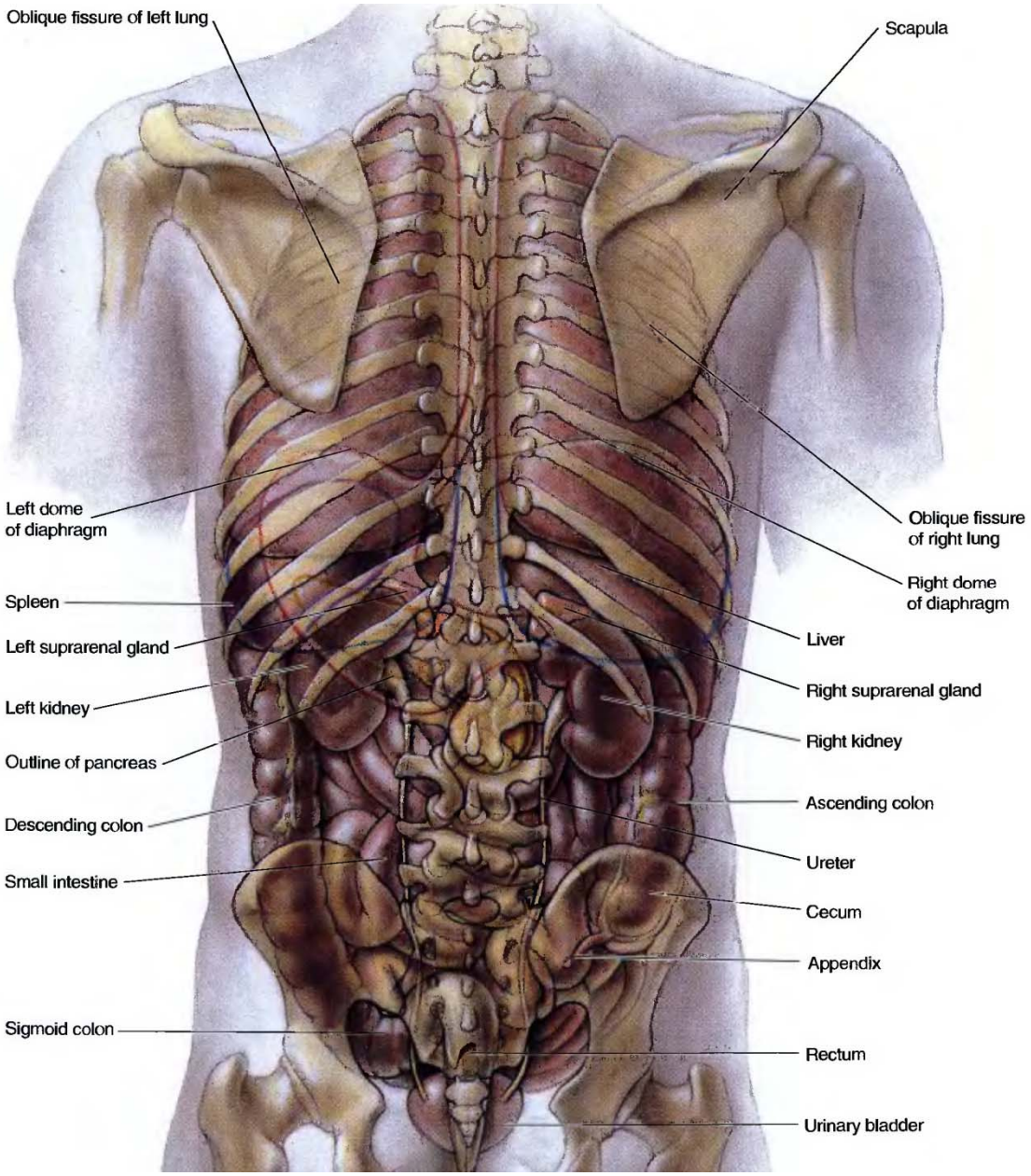
CHAPTER TWO
BACKGROUND

2.1 Anatomy of Abdomen

The human abdomen is the part of the body between the pelvis and the thorax. It stretches from the thorax at the thoracic diaphragm to the pelvis at the pelvic brim (Tortora & Anagnostakos, 1984). The pelvic brim stretches from the lumbosacral angle (the intervertebral disk between L5 and S1) to the pubic symphysis and is the edge of the pelvic inlet. The space above this inlet and under the thoracic diaphragm is termed the abdominal cavity. The boundary of the abdominal cavity is the abdominal wall in the front and the peritoneal surface at the rear.



(a)



(b)

Figure 2.1 Human Abdomen (a) anterior view (b) posterior view (Moore, & Dalley, 1999).

Functionally, the human abdomen is where most of the alimentary tract is placed and so most of the absorption and digestion of food occurs here. The alimentary tract in the abdomen consists of the lower esophagus, the stomach, the duodenum, the jejunum, ileum, the cecum and the appendix, the ascending, transverse and descending colons, the sigmoid colon and the rectum. Other vital organs inside the abdomen include the liver,

the kidneys, the pancreas and the spleen. The abdominal wall is split into the posterior (back), lateral (sides) and anterior (front) walls.

The abdomen contains most of the tube like organs of the digestive tract, as well as several solid organs. Hollow abdominal organs include the stomach, the small intestine, and the colon with its attached appendix. Organs such as the liver, its attached gallbladder, and the pancreas function in close association with the digestive tract and communicate with it via ducts. The spleen, kidneys, and adrenal glands also lie within the abdomen, along with many blood vessels including the aorta and inferior vena cava. Anatomists may consider the urinary bladder, uterus, fallopian tubes, and ovaries as either abdominal organs or as pelvic organs. Finally, the abdomen contains an extensive membrane called the peritoneum. A fold of peritoneum may completely cover certain organs, whereas it may cover only one side of organs that usually lie closer to the abdominal wall. Anatomists call the latter type of organs retroperitoneal.

2.1.1 Liver

The liver is the largest glandular organ, is located on the right side of the abdominal cavity, has a reddish brown color and has a weight of about 1.5 kg (Anthea et al., 1993). The liver has lobes of unequal size and shape and it is in connected with two large blood vessels. The first one (i.e. hepatic artery) carries blood from the aorta and the second one (i.e. portal vein) carries blood containing digested food from the small intestine. These blood vessels subdivide into capillaries which then lead to a lobe.

The liver is necessary for survival and there is currently no technique or machinery that can to compensate the absence of liver. It has a wide range of functions, including detoxification, protein synthesis, and production of biochemicals necessary for digestion. It produces bile, an alkaline compound which aids in digestion, via the emulsification of lipids. It also performs and regulates a wide variety of high-volume biochemical reactions requiring highly specialized tissues, including the synthesis and breakdown of small and complex molecules, many of which are necessary for normal vital functions.

2.1.1.1 Vasculature

As mentioned previously, the liver receives a dual blood supply from the hepatic portal vein and hepatic arteries (Figure 2.2). Seventy five percent of the liver's blood supply is venous blood carried from the spleen, gastrointestinal tract, and its associated organs by hepatic portal vein. The rest of the blood is supplied by the hepatic arteries which carry arterial blood to the liver. Oxygen is provided from both sources almost in equal amount.

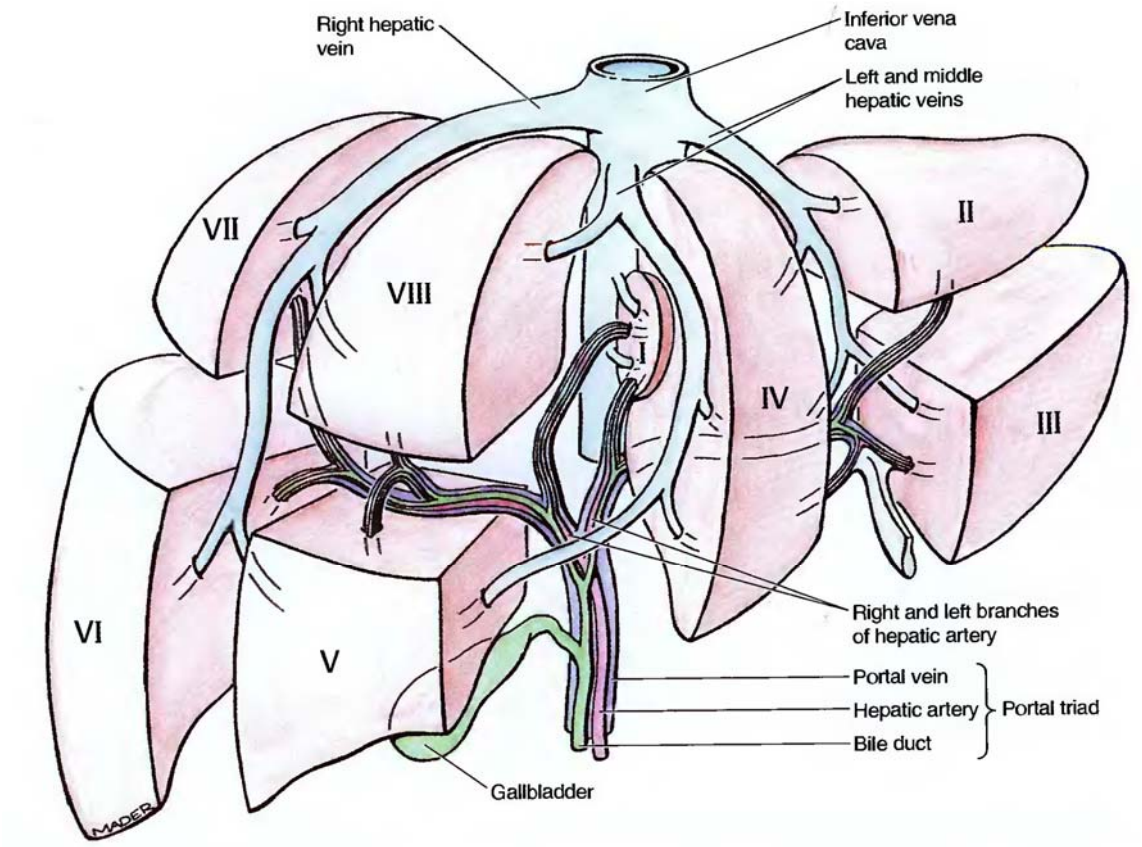


Figure 2.2 Internal anatomy and vasculature of the liver (Moore, & Dalley, 1999).

2.1.1.1 Liver Transplantation

Liver transplantation is an operation that is applied to people with irreversible liver failure (i.e. chronic liver diseases such as cirrhosis, chronic hepatitis C, alcoholism, autoimmune hepatitis, rarely fulminant hepatic failure etc.). Living Donor Liver Transplantation (LDLT) is a technique in which a portion of a living person's liver is removed and used to replace the entire liver of the recipient. Although LDLT is first

applied as adult-to-child, today adult-to-adult liver transplantation has been done using the donor's right hepatic lobe which amounts to 60% of the liver. Due to the ability of the liver to regenerate, both the donor and recipient can have normal liver function (Anthea et al., 1993).

With the recent advances of non-invasive imaging, living liver donors can be examined and evaluated prior to surgery to decide if the anatomy is feasible for donation. The evaluation is usually performed by Multi Detector row CT (MDCT) (Flohr et al., 2005) and MR. MDCT is good in vascular anatomy and volumetry. MR is used for biliary tree anatomy. Donors with smaller liver volumes than necessary or very unusual vascular anatomy, which makes them unsuitable for donation, could be screened out to avoid unnecessary operations.

2.1.2 Right and Left Kidneys

The kidneys are paired organs, one on each side of the spine, located behind the abdominal cavity at the vertebral level T12 to L3 (Tortora & Anagnostakos, 1984). The right kidney sits just below the diaphragm and posterior to the liver, the left below the diaphragm and posterior to the spleen. Resting on top of each kidney is an adrenal gland (also called the suprarenal gland). The asymmetry within the abdominal cavity caused by the liver typically results in the right kidney being slightly lower than the left, and left kidney being located slightly more medial than the right. Each adult kidney weighs between 125 and 170 g in males and between 115 and 155 g in females (Boron, 2002).

They are an essential part of the urinary system, but have several secondary functions concerned with homeostatic functions. These include the regulation of electrolytes, acid-base balance, and blood pressure. In producing urine, the kidneys excrete wastes such as urea and ammonium; the kidneys also are responsible for the reabsorption of glucose and amino acids. Finally, the kidneys are important in the production of hormones including calcitriol, renin and erythropoietin.

The kidneys receive blood from the paired renal arteries, and drain into the paired renal veins. Each kidney excretes urine into a ureter, itself a paired structure that empties into the urinary bladder. Despite their relatively small size, the kidneys receive

approximately 20% of the cardiac output (Boron, 2002). Each renal artery branches into segmental arteries, dividing further into interlobar arteries which penetrate the renal capsule and extend through the renal columns between the renal pyramids. The interlobar arteries then supply blood to the arcuate arteries that run through the boundary of the cortex and the medulla. Each arcuate artery supplies several interlobular arteries that feed into the afferent arterioles that supply the glomeruli.

After filtration occurs the blood moves through a small network of venules that converge into interlobular veins. As with the arteriole distribution the veins follow the same pattern, the interlobular provide blood to the arcuate veins then back to the interlobar veins which come to form the renal vein exiting the kidney for transfusion for blood.

2.1.3 Spleen

The spleen is located in the left upper quadrant of the abdomen beneath the 9th to the 12th thoracic (Tortora & Anagnostakos, 1984), is approximately 11 centimeters in length and weighs 150 grams (Spielmann, DeLong, & Kliewer, 2005). It removes old red blood cells and holds a reserve in case of hemorrhagic shock, especially in animals like horses (not in humans), while recycling iron (Mebius & Kraal, 2005). It synthesizes antibodies in its white pulp and removes, from blood and lymph node circulation, antibody-coated bacteria along with antibody-coated blood cells (Mebius & Kraal, 2005) Recently, it has been found to contain, in its reserve, half of the body's monocytes, within the red pulp, that, upon moving to injured tissue (such as the heart), turns into dendritic cells and macrophages while aiding "wound healing", or the healing of lacerations. It is one of the centers of activity of the reticuloendothelial system and can be considered analogous to a large lymph node as its absence leads to a predisposition toward certain infections.

2.1.4 Stomach

The stomach is a muscular organ of the digestive tract and located between the esophagus and the small intestine. It is on the left upper part of the abdominal cavity. The top of the stomach lies against the diaphragm. It is involved in the second phase of

digestion, following mastication (chewing). The stomach churns food before it moves on to the rest of the digestive system.

In a healthy humans, the stomach has a relaxed, near empty volume of about 45 ml. It is a distensible organ. It normally expands to hold about 1 litre of food, but will hold as much as 2-3 litres (whereas a newborn baby will only be able to retain 30ml).

2.1.5 Abdominal Aorta

The abdominal aorta is the largest artery in the abdominal cavity. As part of the aorta, it is a direct continuation of the descending aorta (of the thorax). It begins at the level of the diaphragm, crossing it via the aortic hiatus, technically behind the diaphragm, at the vertebral level of T12. It travels down the posterior wall of the abdomen in front of the vertebral column. It thus follows the curvature of the lumbar vertebrae, that is, convex anteriorly. The peak of this convexity is at the level of the third lumbar vertebra (L3).

It runs parallel to the inferior vena cava, which is located just to the right of the abdominal aorta, and becomes smaller in diameter as it gives off branches. This is thought to be due to the large size of its principal branches. At the 11th rib, the diameter is about 25 mm; above the origin of the renal arteries, 22 mm; below the renals, 20 mm; and at the bifurcation, 19 mm.

The abominal aorta's venous counterpart, the Inferior Vena Cava (IVC), travels parallel to it on its right side. Above the level of the umbilicus, the aorta is somewhat posterior to the IVC, sending the right renal artery travelling behind it. The IVC likewise sends its opposite side counterpart, the left renal vein, crossing in front of the aorta. Below the level of the umbilicus, the situation is generally reversed, with the aorta sending its right common iliac artery to cross its opposite side counterpart (the left common iliac vein) anteriorly.

2.1.6 Pancreas, Gall Bladder and Other Organs

The pancreas is a gland organ in the digestive and endocrine system of vertebrates. It is both an endocrine gland producing several important hormones, including insulin, glucagon, and somatostatin, as well as an exocrine gland, secreting pancreatic juice containing digestive enzymes that pass to the small intestine. These enzymes help in the further breakdown of the carbohydrates, protein, and fat in the chyme.

The gallbladder is a hollow organ that sits in a concavity of the liver known as the gallbladder fossa. In adults, the gallbladder measures approximately 8 cm in length and 4 cm in diameter when fully distended. It is divided into three sections: fundus, body, and neck. The neck tapers and connects to the biliary tree via the cystic duct, which then joins the common hepatic duct to become the common bile duct. The adult human gallbladder stores about 50 millilitres of bile, which is released when food containing fat enters the digestive tract, stimulating the secretion of cholecystokinin (CCK). The bile, produced in the liver, emulsifies fats in partly digested food. After being stored in the gallbladder, the bile becomes more concentrated than when it left the liver, increasing its potency and intensifying its effect on fats.

2.2 Abdominal Imaging, Acquisition and Display

Modern imaging modalities and techniques allow acquisition of anatomical and physiological information from human body in detail. In radiology, images are primarily acquired with these modalities and then processed to enhance the information of interest among others. For more advanced analysis, digital image processing techniques can be used to extract necessary information or to make measurements which can be used for planning treatments, surgeries and other operations. This section focuses on the tools used for acquiring and displaying abdominal images. Then, the next section covers fundamentals of abdominal image processing.

2.2.1 Image Acquisition

Several different modalities are in clinical use for imaging of abdominal anatomy and physiology (Bidaut, 2000). These modalities use different properties of human body

to acquire data and different techniques to reconstruct them into images. CT uses attenuation measure of X-rays to provide information about the absolute density of tissues, MR uses proton density and relaxation mechanisms, MR spectroscopy (MRS) uses chemical contents of the tissues, single photon emitted computed tomography (SPECT), and positron emission tomography (PET) uses the varying type of tracers that are injected through a vessel prior to the acquisition. Moreover, different information can also be acquired using the same modality through different techniques such as injection of contrast agents in CT or MR, different tracers in SPECT and PET, dynamic acquisitions, etc.

2.2.1.1 Computer Tomography

In radiology, CT scanner is an essential tool because of its useful and fast applicability in a wide range of clinical situations. CT scanners measure the attenuation of X-rays, which are transmitted by rapid rotation of the X-ray tube 360° around the patient, by a ring detectors located on the gantry around the patient (Flohr et al., 2005). The cross-sectional, two-dimensional images are then generated from these measurements using mathematical techniques that can reconstruct 2-D data from multiple 1-D projections.

Due to the advancements in related technology, various techniques have been developed after first CT scanners which acquire single slice at a time (sequential scanning). By enabling the X-ray tube to rotate continuously in one direction around the patient, helical or spiral CTs are implemented. During the continuous rotation of tube, the table on which the patient is lying is also moves through the X-ray beam. With this technique, information can be acquired as a continuous volume (Flohr et al., 2005). The benefit of this technique is mainly on acquisition speed that allows acquisition of volume data without mis-registration that is caused by breath hold time of the patient and other patient movements. The most recent CT scanners have the same principles of the spiral scanner but in addition to that, they consist of multiple rows of detector rings which provide the possibility of multiple slice acquisition for each rotation of the X-Ray tube (Flohr et al., 2005).

These improvements in CT image acquisition have drastic effects on what can be done through volumetric applications such as CTA for vascular analysis, 3-D imaging and image processing techniques (i.e. Multi-planar reconstruction, maximum intensity projection, surface rendering, and volume-rendering).

2.2.1.2 Magnetic Resonance Imaging

Unlike CT that uses radiation, MR uses a powerful magnetic field to align the hydrogen atoms in the body which is largely composed of water thus contain hydrogen. Inside the magnetic field of the MR scanner, the magnetic moments of hydrogen atoms align with the direction of the field (Suetens, 2002). Radio frequency (RF) fields are used to alter this alignment which causes the hydrogen nuclei to produce a rotating magnetic field when returning to the original magnetization alignment. This field is then received by the antennas on scanner and the incoming signal can be used to reconstruct cross-sectional images or volume data.

Although soft tissues are represented in a very narrow scale (i.e. Hounsfield value range) in CT, the technique behind MR imaging provides high contrast between different soft tissues of the body since it depends on the fact that tissues with different amount of hydrogen return to their equilibrium state at different rates. Therefore, it is especially useful in neurological, musculoskeletal, cardiovascular, and oncological imaging although every part of the body can be imaged. The parameters of the MR scanner (i.e. application time and strength of fields etc.) can be changed to create contrast between different types of body tissue. Similar to CT imaging, contrast agents may be injected intravenously to enhance the appearance of blood vessels, tumors etc. Although, development MR compatible versions of implants and pacemakers is an emerging field, currently patient with those devices are generally prevented from having an MR scan due to effects of the magnetic field.

2.2.1.3 Ultrasound

Ultrasound is cyclic sound pressure with a frequency greater than (approximately) 20 kHz. Ultrasound technology is used in many applications based on penetrating a medium and measuring the reflection signal which can reveal information about the

inner structure of the medium. In medicine, ultrasonography is one of the most widely used and ultrasound-based diagnostic medical imaging technique which is used for imaging of many organs, tissues and especially fetuses in real time. Compared to CT and MR, ultrasound is relatively inexpensive and portable.

2.2.2 Image Display

Digital images are composed from a grid of 2-D elements (i.e. pixels) or 3-D elements (i.e. voxels). Although different techniques and pipelines are required for generation and display of 2-D and 3-D data, many 2-D processing techniques can be extended to 3-D data sets where 2-D pixels expand into 3-D voxels. Depending on the modality in use, images are processed to enhance the most important part of the dynamic range to emphasize the information of interest prior to display. Some of these processing functions, which are presented in the following sub-sections, can significantly alter the presentation of the information which can be misleading without proper understanding of the technique applied. For this reason, image display should always be handled very carefully in clinical practice.

2.2.2.1 Image Windowing

Typical digital image types are usually 8 bit images that have 256 (i.e. 2^8) gray levels typically from 0 to 255. Although, this dynamic range is more than the noticeable dynamic range of the human eye, clinical imaging equipment mostly produce 12 or 16 bit images for more accurate representations. This requires a conversion from 12-bit (2^{12} → values from 0 to 4095 = 2^{12} - 1) or 16 bit data (2^{16} → values from 0 to 65535 = 2^{16} - 1) to 8-bit representation to be fed to the display hardware. It is possible to apply a proportional scaling (i.e. [0, 4095] → [0, 255] or [0, 65535] → [0, 255]), however, because of the under-sampling it becomes very difficult to assess the density variations of interest. Since all of the dynamic range contain more than the information of interest, windowing can be used to reduce the spectrum of interest prior to scaling. By this way, only some part of the input dynamic range is scaled and thus less under-sampled. For CT and MR, windows are defined by Window Center (WC) and Window Width (WW) which are also the standard tags in DICOM format. Preset windows, which are defined

by WC and WW, are commonly used to increase the contrast of specific tissue types, as the examples in Fig. 2.3.

2.2.2.2 Look Up Tables

In current display systems, the conversion between a digital image and a display screen is done through a Look-Up Table (LUT) (Lutz, Pun, & Pellegrini, 1991). An LUT is a function that simply converts a value derived from the input data to an output value based on its shape. This output value is then used to produce a point, whose brightness is proportional to the output of the LUT, on the display (Fig. 2.3).

By adjusting the shape of the corresponding function, LUTs can be used to enhance some part of the dynamic range as it is done in windowing. These shapes can be linear and non-linear depending on the enhancement.

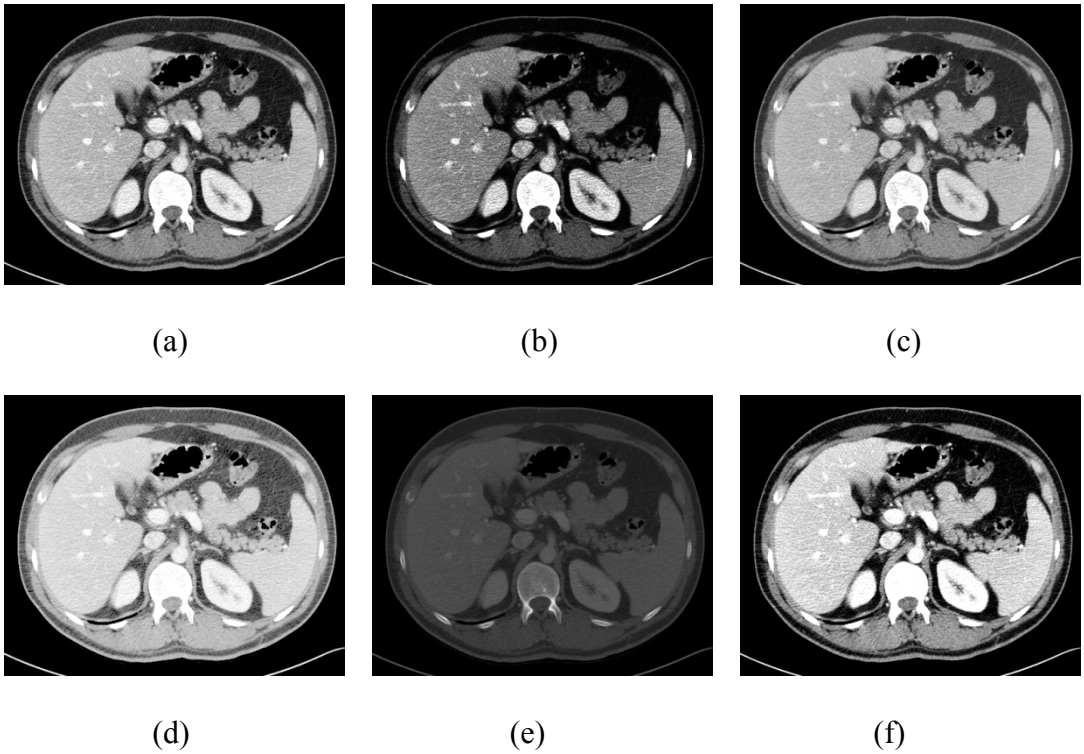


Figure 2.3 Different Window Level and window width adjustments for an abdominal CT image (a) original (b) square LUT (c) logarithmic LUT (d) square root LUT (e) windowing presets for bones (f) windowing presets for mediastinum.

2.2.2.3 Interpolation and Re-sampling

Tomographic images acquired by CT/MR usually have same size (e.g. in millimeters) for a pixel in both dimensions (i.e. x and y). However, in volumetric data, the third dimension (i.e. z axis), which represents slice thickness; mostly have different size than the other two axes. The slice thickness is usually greater than the sizes of other axes and these different sizes require interpolations to produce images that adequately represent true anatomic proportions and relationships in all three axes. Several interpolation techniques exist from simple ones, which use the values of neighbor pixels/voxels, to complex ones, which use polynomial or surface approximations, shape modeling, and splines. Naturally, as the technique becomes more complex, it requires more computation but represents the missing data better (Robb, 1995).

2.3 Abdominal Image Processing

Digital image processing techniques can be used to extract any necessary and/or important information from complete image to make measurements or other analysis which can be used for planning treatments, surgeries and other operations. As mentioned in image display, digital image processing techniques, which are presented in the following sub-sections, can change the original image data which should carefully be handled to prevent misleading results. Therefore, although, very sophisticated techniques are available, most of them are still seldom implemented in current clinical practice and on commercially available systems. Although some of these techniques, which are related with the content of this thesis, are given in the following sections of this chapter, many other techniques are available (Gonzalez & Woods, 1992).

2.3.1 Image Enhancement

Redistribution of the pixel values in an image is an alternative way of using windowing or LUTs to enhance structures of interest. One of the common and simplest ways to accomplish this redistribution is using image histogram which shows the sum of pixels at the same value for each value in an image. Redistributing the pixel values to create a histogram that is uniform or linear is one of the most useful techniques for global enhancement and called histogram equalization (Fig. 2.4). This technique can be

extended to volumes by summing up the histograms of all images to create a cumulative histogram, called volumetric histogram.

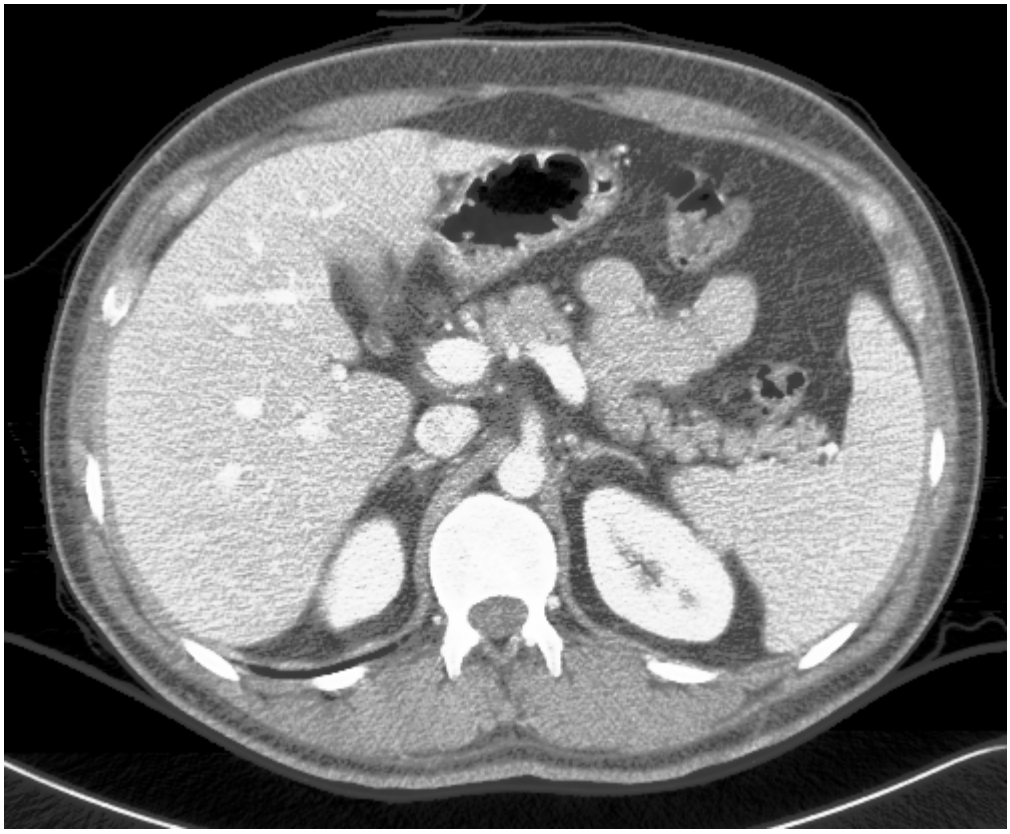


Figure 2.4 Result of histogram equalization applied to Fig. 2.3 (a)

2.3.2 Filtering

Tomographic images can be processed through filters to modifications or enhancements. The application of the filters can be either in the original domain of the image (i.e. spatial domain that consists of pixels) or in the frequency domain that covers a spectrum of frequency components after 2-D Fourier Transform (FT) (Gonzalez & Woods, 1992). Similar to 1D signals, a kernel (i.e. filter mask) is convolved with the image where each pixel block that is equal to the kernel size is mathematically combined with the kernel to produce the filtered image. The kernel size is mostly an odd number (i.e. 3x3, 5x5 etc.) so that there is a pixel in the center and usually only the value of this center pixel changes at each step of the process. Thus, one of the main parameters of a filter is the size of the kernel that determines the size information to be used at each

effect. On frequency domain filtering, the representation of the filter function in frequency domain is multiplied with the frequency spectrum, which is calculated via FT, of the original image. Then, Inverse FT of the resulting spectrum is taken for obtaining the filtered image on spatial domain.

2.3.2.1 Low-Pass Filtering

Low-pass filtering (Gonzalez & Woods, 1992) can be used for removing high-frequency components of an image such as noise. Depending on the strength and type of the low-pass filter in use, edges, small details, some kinds of texture can also be eliminated. Thus, low-pass filter preserves large structures and enhance homogeneous regions in an image. For abdominal images, these homogeneous regions usually correspond to parenchyma of the organs or the inner side of the tissues (Fig. 2.5).

Smoothing is most commonly performed by taking the average a pixel and its neighbors. Another technique weighs the filter coefficients according to their distance from the center of the filter which is then called a weighted mean filter. The filter coefficients can also alter based on a Gaussian shape centered on the middle of the kernel. On the other hand, median filtering determines and uses the middle value when all the values inside the kernel are ordered from lowest to highest. In spite of the others, median filter uses only values from the image and preserves edges better than averaging.

2.3.2.2 High-Pass Filtering

In contrast with low-pass filtering, high pass filtering (Gonzalez & Woods, 1992) can be used to enhance or extract the detail information in an image. This high frequency information can be small features, edges or other sharp/instant changes. To detect these changes, high-pass filters are generally based on detecting differences and discontinuities that are characterized drastic changes between neighboring pixels in spatial domain and by high frequencies in the frequency domain. Since, enhancing details can also increase noise, the parameters of filters should carefully be determined depending on the specific application (Fig. 2.5).

Gradient (i.e. Prewitt, Sobel, etc.), Laplacian and Laplacian of Gaussian (LoG) (i.e. Mexican hat) filters are some typical examples of these filters. Gradient filters are based on a first-order derivative over a neighborhood of the pixel and produce a vector in the spatial domain in which, the largest values correspond to the largest local changes (e.g. edge). Laplacian filters are second-order derivatives and they produce null values for gradients' maxima and minima. To decrease the sensitivity of Laplacian filters to noise, the result of the filter can be added to the original image (i.e. unsharp masking) to enhance high frequency components. In LoG filters, image is first smoothed by a Gaussian filter before enhancing edge detection with a Laplacian filter.

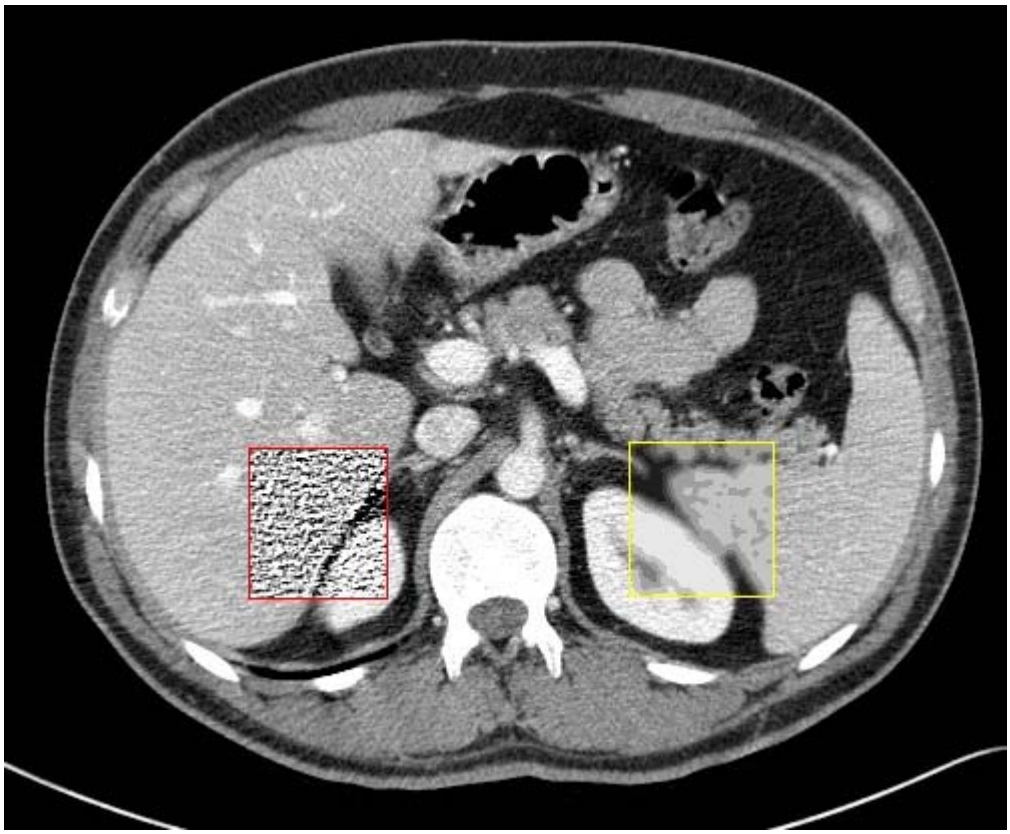


Figure 2.5 Effects of high pass filtering (red rectangle) and low pass filtering (yellow rectangle)

2.3.2.3 Morphological (Structural) Filtering

Morphologic filtering is a process based on the shapes of the objects on an image and structuring element of the filter (Watt, 1993). Since shape information can not be represented in frequency domain via FT, these filters can only be applied on the spatial

domain. The shape of the structuring element is defined in a kernel and the filtered image is calculated by convolving this kernel with the original image. Although, its application is most effective on binary image processing, morphological filtering can also be used for gray-level images. Mainly there are two morphological filters (i.e. erosion and dilation).

Erosion tends to shrink objects, open holes or gaps in binary images (0 for the background and 1 for the objects), and takes the minimum value of the image over the convoluted shape in gray level images. On the other hand, dilation tends to expand objects and close holes and gaps in binary images, takes the maximum value of the image over the convoluted shape in gray level images. By combining erosion and dilation, two more filters can be derived. Opening filter is the combination of erosion followed by dilation and closing filter is the combination of dilation followed by erosion. By changing the shape and size of the structuring element, morphologic filters can be used to suppress artifacts, select objects with a specific shape, remove small objects, connect separated objects and split unwanted connections etc.

2.3.2.4 Anisotropic Filtering

As mentioned in previous sub-sections, low-pass filters are necessary to remove the noise from digital images; however, they also cause blurring which is not wanted because of unclear edges/borders and removal of potentially important high frequency data. To reduce image noise without removing significant parts of the image content, (i.e. edges, lines or other details), anisotropic diffusion can be used to remove noise from digital images without blurring edges (Perona & Malik, 1987). The anisotropic diffusion equations are equal to Gaussian blurring when diffusion coefficient is chosen to be constant. When the diffusion coefficient is chosen as an edge seeking function (Perona & Malik, 1987), the resulting equations encourage diffusion (hence smoothing) within homogeneous regions and prohibit it across strong edges. Hence the edges can be preserved while removing noise from the image. By running the diffusion with an edge seeking diffusion coefficient for a certain number of iterations, the image can be evolved towards a piecewise constant image with the boundaries between the constant components being detected as edges.

In the formulation presented in (Perona & Malik, 1987), the filter depends on the image content such that it approximates an impulse function close to edges and other structures that should be preserved in the image. As a consequence, the resulting images preserve linear structures while at the same time smoothing is made along these structures. Consequently, anisotropic diffusion is an iterative process where a relatively simple set of computation are used to compute each successive image in the family and this process is continued until a sufficient degree of smoothing is obtained.

2.3.3 Multi-dimensional Image Processing

Displaying and processing only 2-D slices allow limited analysis on tomographic data sets which are in fact series of 2-D images (i.e. slices) that are discrete cuts through 3-D volume of acquisition. Taking advantage of this fact, advanced image processing techniques can be used for further analysis and visualization of the volume data which can be used to extract and visualize information or objects in a more realistic way. The following subsections cover information about some of these techniques including Multi planar reconstruction for detailed analysis in 2-D, and visualization techniques such as surface and volume rendering for 3-D.

2.3.3.1 Multi Planar Reconstruction (MPR)

Since they are simply some cuts from the actual volume, 2-D images can be stacked together to reconstruct the volume at the time of the acquisition. Multi Planar Reconstruction (MPR) is a technique that can display non-acquired orthogonal orientations of the volume by readdressing the order of pixels and create images of orthogonal planes (Gonzalez, & Wintz, 1987) from 2-D slices. For example, when the original plane of acquisition was the transaxial plane, orthogonally reformatted coronal (from back to front or vice versa) and sagittal (from left to right or vice versa) sections can be obtained by MPR. The reconstruction can be done with orthogonal planes (standard MPR), with oblique plane extraction which is a cut of the reconstructed volume along any arbitrary plane (Robb, 1995), or with curved plane extraction, which is a non-planar cut along 3-D curved path inside the reconstructed volume (Robb, 1995).

MPR is a very useful tool can for displaying structures that are not “well oriented” with regard to the original acquisition’s main axis. In current systems, the user can determine an arbitrarily oriented plane at some oblique angle to the axes of the volume image which produces corresponding cross section images at the other two planes. Reconstructed MPR images can also be process through image processing techniques.

2.3.3.2 Surface Rendering

Surface analysis is useful when there is a need to visualize 3-D surfaces of a tissue or organ. To represent objects as surfaces, their defining voxels can first be found and then joined together by smaller triangular/polygonal surfaces. These surfaces are then merged together to represent the outer shell of the structure of interest (Watt, 1993). For an informative rendering, the number of small surfaces is very high and finding correct representation type for each of these surfaces requires high computational power that necessitates usage of efficient algorithms. One of the most efficient algorithms for creating surfaces is Marching Cubes (MC) (Lorensen, & Cline, 1987).

Once surfaces have been created using MC or other algorithms, post-processing operations, such as smoothing, can be applied for refinement. Also, for faster display or for decreasing storage requirements, surfaces can be simplified by merging tiles or facets or suppressing redundant ones (Schroeder, Zarge, & Lorensen, 1992).

For displaying an extracted surface in a proper way, various properties should be adjusted and determined regarding its representation, color, transparency (or opacity), and response to external lighting (Watt, 1993). Light and shading models have extremely important effects on the visualization of the surface. Several complex models are developed for clear and realistic representation of surfaces. As an alternative, surfaces can also be textured by projecting images or data on the elementary polygonal tiles and this technique is called texture mapping. Due to computational complexity of surface based rendering techniques, current systems are supported by appropriate hardware (i.e. graphics card) that are responsible for some steps of the rendering pipeline.

2.3.3.3 Volume Rendering

Compared with surface rendering, volume rendering (Watt, 1993) does not require the segmentation of structures but uses all voxels from the volume. In this type of rendering, a color and an opacity/transparency is assigned to each voxel through transfer functions that link voxel values with an LUT like curve. The volume is then represented through ray-tracing (or casting) paradigm (Watt, 1993). In ray-tracing, rays extend from infinity to the observer while some of the rays pass through empty space, whereas others intercept the object(s). Ray tracing implies that all rays originate at infinity (beyond the object) and travel toward the observer. Ray casting goes the other way, from the observer to infinity (and beyond). The individual opacity/transparency parameters can be selected to best tune the representation to the observer's needs. A special (simplified) mode of volume rendering is MIP, where only the highest value on a ray is projected to the observer. This technique is used mainly for rendering isolated and highly contrasted objects such as bones or vascular structures in CT or MR angiography.

2.3.4 Image Segmentation

Segmentation in biomedical image processing refers to isolation of objects that usually needs to be further analyzed, measured or visualized. Segmentation can be divided into three types as manual, semi-automatic and automatic.

Manual methods should be the most reliable techniques since they involve interactive delineation of boundaries by an expert physician. However, they are also time consuming, error prone, subjective and not reproducible because they are based on operator's trace using a device for labeling border pixels (i.e. mostly a mouse and a computer program). This operation does not only depend on the capabilities and interaction mechanisms of the software in use but also affected by external factors such as physical conditions of the environment and operator (i.e. room and display lighting, time of the day etc.). Within manual techniques, the simplest segmentation method is the selection of a Region of Interest (ROI) in 2-D or Volume of Interest (VOI) in 3-D depending on the needs of the user (Robb, 1995) but this can only have minor benefit. A very common manual method is drawing contours using interactive software tools around a structure (e.g. an organ) of interest on each slice, which shows some part of the

structure. At the end of the process, these contours can be displayed together on a workstation as 3-D representation of the corresponding structure. Depending on the capabilities of the software in use, this process can be very time consuming and tedious since a single tomographic series usually consist of tens or hundreds of images.

Semi-automatic methods combine image processing techniques with expert intervention. In most of the cases, the expert initializes the process by inserting some initial information or boundary. Region growing and active contour techniques are widespreadly used examples of semi-automated techniques. In both of these techniques, initial seed points and boundaries are inserted. Starting from this initialization, region growing technique checks for connectivity of seed pixels based on some criteria (i.e. thresholding). Similarly, active contours try to minimize a cost function that is formed by the initial boundary.

Fully automatic techniques are mostly available for a single organ or for a specific aim because it is almost impossible to develop an automatic algorithm that can handle several different applications. Automatic techniques require determination of several parameters without user intervention, however the range of these parameters can not be kept in strict bounds due to the high variations in human anatomy, image characteristics etc. This high number of parameters and large variations in parameter space prevents the usage of automatic methods in a broad sense.

The segmentation techniques in the following paragraphs can be used both as semi-automatic and automatic depending on the way of their implementation. These techniques include thresholding approaches, classifiers, clustering approaches, Markov random field models, artificial neural networks, and atlas guided approaches. Often supported and used together with some pre-processing and/or post-processing operations (i.e. edge or contrast enhancement), these techniques can be used solely or in a combination.

Thresholding techniques extract only the voxels whose value falls within lower and upper threshold value range. A thresholding procedure attempts to determine an intensity value, called the threshold, that separates a group of pixels, which are between lower and upper threshold values, from the others. Determination of more than one

threshold value is a process called multi-thresholding (Sahoo, Soltani, & Wong, 1988). Thresholding is a simple but sometimes very effective technique for obtaining a good segmentation result when structures of interest have different intensity (or feature value in general) range than other tissues. Unfortunately, this is usually not the case for medical images, especially for soft tissues.

Classifier methods are pattern recognition techniques that partitions a feature space derived from the image using several feature extraction techniques (i.e. texture, spectral etc.) (Schalkoff, 1992). All features in use constitute feature space. A feature space can be as simple as image intensities themselves or a histogram, which is an example of a 1D feature space. Classifiers for medical image segmentation are usually supervised, which use a training data that are used as references for adjusting parameters of a classifier. Some examples of the classifiers that can be used in image segmentation are nearest-neighbor classifiers, where each pixel or voxel is classified in the same class as the training datum with the closest intensity, The K -nearest-neighbor (kNN) (Duda, Hart, & Stork, 2000) classifier which is a generalization of the nearest neighbor approach. Another classifier is the Parzen window (Duda, Hart, & Stork, 2000), where the classification is made according to the majority vote within a predefined window of the feature space centered at the unlabeled pixel intensity. A commonly-used parametric classifier is the maximum likelihood (ML) (Duda, Hart, & Stork, 2000) or Bayes classifier that assumes the pixel intensities are independent samples from a mixture of probability distributions, usually Gaussian.

Similar to classifiers, clustering algorithms also perform partitioning using a feature space but they do not require training data. Therefore, they are called unsupervised methods. To partition the space without any training data, clustering methods work in an iterative way in which some parameter (i.e. any kind of cost function) from current partitioning is calculated and then used (i.e. tried to be minimized) for changing partitions. Some of the well known and commonly used clustering algorithms are the K -means (Coleman & Andrews, 1979), the fuzzy C-means algorithm (Dunn, 1973), and the expectation-maximization (EM) algorithm (Duda, Hart, & Stork, 2000). The K -means clustering algorithm clusters data by iteratively computing a mean intensity for each class and segmenting the image by classifying each pixel in the class with the

closest mean (Jain, & Dubes, 1988). The number of classes should be given prior to application. The initial centers of the clusters can be initialized randomly however a better initialization would increase the performance of the clustering process. The fuzzy C-means algorithm generalizes the K-means algorithm, allowing for soft segmentations based on fuzzy set theory (Zadeh, 1965). The EM algorithm applies the same clustering principles with the underlying assumption that the data follows a Gaussian mixture model. It iterates between computing the posterior probabilities and computing maximum likelihood estimates of the means, covariances, and mixing coefficients of the mixture model.

Markov random field (MRF) modeling itself is not a segmentation method but a statistical model which can be used within segmentation methods. MRFs model spatial interactions between neighboring or nearby pixels. These local correlations provide a mechanism for modeling a variety of image properties (Li, 1995). In medical imaging, they are typically used to take into account the fact that most pixels belong to the same class as their neighboring pixels. In physical terms, this implies that any anatomical structure that consists of only one pixel has a very low probability of occurring under a MRF assumption.

Artificial Neural Networks (ANNs) are parallel networks of processing elements that simulate biological learning by performing elementary computations. Learning is achieved through the adaptation of weights assigned to the connections between these elements. A survey on neural networks can be found in (Clark, 1991) and a more detailed information can be accessed in (Haykin, 1999). ANNs can be used in a variety of ways for image segmentation such as classification (Hall et al., 1992) and clustering. In case of classification, the weights of ANN are determined using training data, and then adjusted weights are used to segment new data. Because of the many interconnections used in a neural network, spatial information can easily be incorporated into its classification procedures. In case of clustering, ANNs are used in an unsupervised fashion as a clustering method, as well as for deformable models (Vilarino, Brea, Cabello, & Pardo. 1998).

When a standard atlas or template is available, atlas based approaches can be used that are generated by compiling information on the anatomy that requires segmenting. Using atlas as the reference, atlas-guided approaches perform a classification task in the spatial domain and they treat segmentation as a registration problem (Maintz, & Viergever, 1998) such as finding a one-to-one transformation that maps a pre-segmented atlas image to the target image that requires segmenting is found.

Besides the technique to use, the modality of acquisition has also very important affects on the segmentation method selection because different modalities provide images with very different characteristics.

Segmentation of abdominal organs, such as the liver, kidneys, and spleen, from CT is very important due to the wide use of CT but it also has many challenges. Many artifacts can arise in CT scans, such as beam-hardening artifacts, partial-volume artifacts, and streak artifacts (Slone, Fisher, Pickhardt, Gutierrez, & Balfe, 2000). These artifacts are due low attenuation adjacent to bones; spatial averaging of disparate tissues in close proximity and patient motion, respectively. Also, different organs and tissues have overlapping Hounsfield ranges which prevent the usage of thresholding. Moreover, tissue intensities can change drastically for each instance of operation. Further difficulties arise due to lack of organ tissue homogeneity within and among different image slices, both in shape and texture.

In MR images, major difficulty is the intensity inhomogeneity artifact (Condon, Patterson, Wyper, Jenkins, & Hadley, 1989, Simmons, Tofts, Barker, & Arridge, 1994), which causes a shading effect to appear over the image. This artifact can significantly degrade the performance of methods that assume that the intensity value of a tissue class is constant over the image. Numerous approaches have been proposed in the literature for performing tissue classification in the presence of intensity inhomogeneity artifacts. Such as usage of a filter prior to other operations (Meyer, Peyton, & Pipe, 1995).

Segmentation algorithms have had fairly limited application in ultrasound imaging especially due to high levels of speckles Furthermore, the real-time acquisition in ultrasound makes it better suited for motion estimation tasks (Mikic, Krucinski, & Thomas, 1998) where active contours are usually more suitable because of their

dynamic nature. Ultrasound is also often employed in detecting pathology using textural classifiers (Mojsilovic, Popovic, Neskovic, & Popovic, 1997) but regions of interest are typically obtained through manual interaction.

2.3.5 Measurements

Making measurement (i.e. angle, distance, area) in 2-D is a straightforward process using pixel values. The extension of these measurements to 3-D is still possible but sometimes requires more complicated calculations. Distances and angles can be computed within 2-D and 3-D data for providing more accurate measurements of complex structures. Distance measurements require two points while angle measurements can be calculated with three with a straightforward calculation (Robb, 1995).

On the other hand, measurement of an area or a volume can be done in many ways. Once a structure has been segmented out, its contents can be analyzed to provide its value range or other relevant information such as its total volume. The simplest way of calculating a segmented structure's volume is by multiplying the number of voxels that have been identified as belonging to the structure by the volume of a single voxel calculated from the acquisition and reconstruction parameters. For DICOM images, these parameters are stored in "sliceThickness", "pixelSpacingX", and "pixelSpacingY" tags. However, neither this technique nor any other can provide the exact value of the volume.

2.4 Neural Networks

A biological neural network is composed of groups of biologically connected neurons, each of which are connected to many other neurons. These connections are called synapses and they are formed from axons to dendrites. In used in artificial intelligence, simplified models of these neural networks are used for computations as in neural processing and ANNs have been applied successfully to speech recognition, image analysis and adaptive control etc. The tasks that ANNs can be applied might broadly be divided into three categories:

i) Function approximation, regression analysis, time series prediction and modelling.
ii) Classification, pattern and sequence recognition, novelty detection and sequential decision making.
iii) Data processing, filtering, clustering, blind signal separation and compression.

2.4.1 Learning Processes

The learning process for ANN (Haykin, 1999) can be defined as adjustment of the free parameters of an ANN based on some measure and the main aim is to improve the performance of ANN for a better fitting to its field of application. There exist two major learning paradigms each of which corresponds to a particular learning task. These are supervised learning and unsupervised learning (Haykin, 1999). In supervised learning, which can be expressed also as learning with a teacher, a set of example pairs (i.e. training data) are used to adjust the free parameters of the network. This adjustment is done not only under the influence of training data but also an error signal that is defined as the difference between the desired response and the actual response of the network. This adjustment is carried out iteratively in a step-by-step fashion. When the adjustment phase (i.e. training) is finished, the performance of the network can be evaluated with unseen data. In unsupervised learning, free parameters are adjusted in such a sense that a cost function is tried to be minimized (Haykin, 1999). The idea behind the unsupervised learning is to tune the network parameters to the statistical regularities of the input data and thus developing the ability create a new class automatically. The cost function is determined by the task formulation and can be any function of the input data and the network's output (Haykin, 1999).

There are many algorithms for training neural networks; most of them can be viewed as an application of optimization theory and statistical estimation (i.e. Back propagation by gradient descent (Haykin, 1999)). Evolutionary computation methods, simulated annealing, expectation maximization and non-parametric methods are among other commonly used methods for training neural networks (Haykin, 1999).

2.4.2 Single and Multi Layer Perceptrons

The perceptron is the simplest form of an ANN that consists of a single neuron with adjustable synaptic weights and bias. It is proven with the perceptron convergence theorem that the perceptron algorithm converges after a finite number of iterations if the data set is linearly separable (Rosenblatt, 1958). Learning occurs in the perceptron by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result (Haykin, 1999). This is an example of supervised learning, and is carried out through backpropagation, a generalization of the least mean squares algorithm in the linear perceptron.

A multilayer perceptron is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate output. It is a modification of the standard linear perceptron in that it uses three or more layers of neurons (nodes) with nonlinear activation functions, and is more powerful than the perceptron in that it can distinguish data that is not linearly separable. The multilayer perceptron consists of an input and an output layer with one or more hidden layers of nonlinearly-activating nodes. Each node in one layer connects with a certain weight to every node in the following layer (Haykin, 1999).

2.4.3 Radial Basis Function Networks

A radial basis function network is an artificial neural network that uses radial basis functions as activation functions. It is a linear combination of radial basis functions. They are used in function approximation, time series prediction, and control. In an RBF network there are three types of parameters that need to be chosen to adapt the network for a particular task:

i) the center vectors ,
ii) the output weights and
iii) the RBF width parameters

In its simplest form, construction of an RBF network involves three layers. The first layer is the input layer which is responsible for receiving input data. The hidden layer

comes after the first layer and it applies a non-linear transformation from the input space to the hidden space which is usually higher dimensional due to increased likelihood of linear separability in higher dimension. The third layer constructs the output by applying a weighted sum.

CHAPTER THREE
LIVER SEGMENTATION FOR PRE-EVALUATION OF LIVER TRANSPLANTATION

Identifying liver region from abdominal CTA datasets is one of the essential steps in evaluation of transplantation donors prior to the hepatic surgery. However, due to gray level similarity of adjacent organs, injection of contrast media and partial volume effects; robust segmentation of the liver is a very difficult task. Moreover, high variations in liver position, different image characteristics of different CT modalities and atypical liver shapes make the segmentation process harder. To overcome these challenges, a three stage (i.e. pre-processing, classification, post-processing) automatic liver segmentation algorithm, which adapts its parameters according to each patient by learning the data set characteristics in parallel to segmentation process to address all the challenging aspects mentioned above, is proposed. The efficiency in terms of the time requirement and the overall segmentation performance is achieved by introducing a novel modular classification system consisting of a simple classification system (i.e. K-Means based) and a complex one (i.e. MLP based) which are combined with a data-dependent and automated switching mechanism that decides to apply one of them. Proposed approach also makes the design of the overall classification system fully unsupervised that depends on the given CTA series only without requiring any given training set of CTA series. The segmentation results are evaluated by using area error rate and volume calculations and the success rate is calculated as %94.91 over a data set of diverse CTA series of 20 patients according to the evaluation of the expert radiologist. The results show that, the proposed algorithm gives better results especially for atypical liver shapes and low contrast studies where several algorithms fail.

3.1 Introduction and Related Work

Living donated liver transplantation is a procedure where a healthy voluntary donor gives a part of his or her liver to another person. Measurement of the liver volume and analysis of the liver vasculature are important stages to decide whether a candidate for transplantation is suitable or not. Generally, liver volume information is used to avoid

size incompatibility between donor and patient, and vasculature analysis in 3-D is used for pre-evaluation of surgery. Thus, the success of liver transplantation depends on the patency of liver volume and its supplying vessels and accurate knowledge of the hepatic and portal vascular anatomy of donors for living-related transplantation would reduce the incidence of vascular complications during and after transplantation.

Routine preoperative evaluation of donors requires both CT (Flohr et al., 2005) and CT with contrast medium injection, namely CTA, which are currently the most widely used radiographic techniques for the rendering of liver parenchyma, vessels and tumors in living liver transplantation donors. Instead of conventional angiography, CTA offers several advantages: it is minimally invasive and has diminished patient morbidity, cost, and radiation exposure to patients and staff. Moreover, CTA provides detailed information on vasculature due to the injection of contrast media. Before 3-D rendering (Ney, Fishman, Magid, & Drebin, 1990) of the vasculature and the measurement of liver volume, accurate segmentation of the liver from surrounding tissues and organs is necessary. Since the number of image slices used for 3-D rendering is very large, manual segmentation of the liver on each slice is time consuming and tedious. Also the results highly depend on the skill of the operator. Therefore an automatic segmentation procedure to segment the liver in all slices is needed.

Besides its several advantages over manual segmentation, automatic segmentation of the liver is very challenging. These challenges arise from the following difficulties: First of all, the gray level values of adjacent organs of the liver are similar to each other (Fig. 3.1 (a)-3.1 (b)). This similarity reduces the performance of thresholding techniques dramatically. Secondly, due to the injection of contrast media and/or different modality settings, the liver (and all other tissues) may have different gray-level values for different patient datasets, or even in different slices of the same data set (Fig. 3.1 (a)-3.1 (c)). These effects prevent the usage of the gray level dependent segmentation techniques. Finally, the anatomical structure of the liver in different image slices is different and its shape can vary significantly from patient to patient (Fig. 3.1). Even two or three separate regions can be seen in the same slice (Fig. 3.1 (b)). Moreover, it is reported in (Soler et al., 2001) that around %15 of the patients have atypical liver shapes (i.e. unusual size or orientation of the liver, liver shape after segmentectomy) (Fig. 3.1

(d)). Thus, traditional shape based segmentation techniques are not enough to segment the liver efficiently.

The strategy for overcoming these difficulties involves a segmentation method which does not utilize a common parameter set found from all patient datasets. Instead, the method is capable of adapting the parameter set to each patient. The main reason for this approach is that the ranges of the parameter values differ significantly from patient to patient, and these wide ranges decrease the efficiency of the method when one utilizes a common parameter set for all patients.

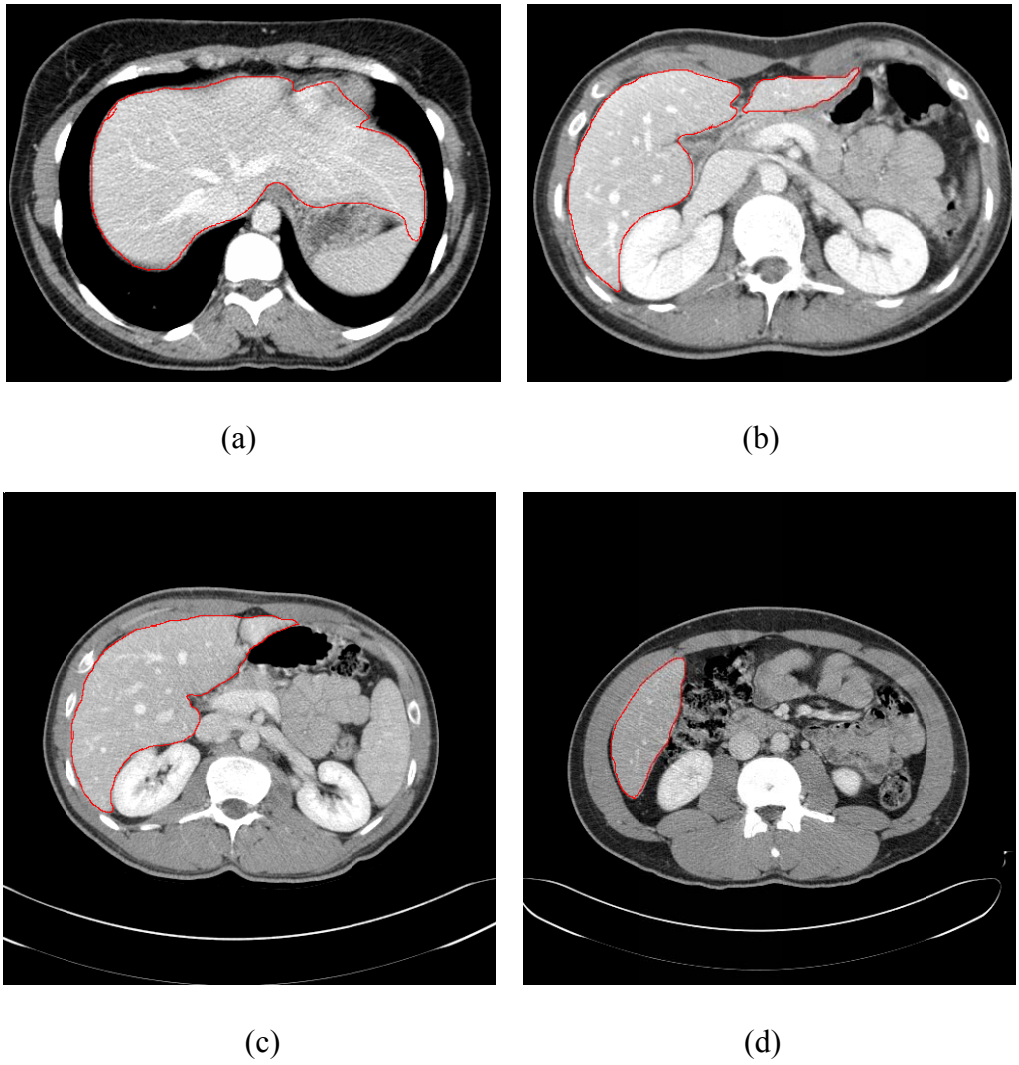
Thus, a method, which examines and adapts its parameters according to each patient, is proposed. We call this approach as patient-oriented segmentation. For qualifying 'patient oriented', the algorithm learns data set characteristics in parallel to segmentation process, and adapts its parameters to these characteristics.

In the literature, different automatic and semi-automatic methods have been developed and performed for the segmentation of the liver from CTA series. These methods include but not limited to morphological techniques (Bae, Giger, Chen, & Kahn, 1993, Gao, Heath, Kuszyk, & Fishman, 1996, Lim, Jeong, & Ho, 2006, Masumoto et al., 2003), deformable models (Montagnat & Delingette, 1996, Chou et al., 1995, Soler et al., 2001, Gao, Kosaka, & Kak, 2000, Heimann, Wolf, & Meinzer, 2006, Heimann, Meinzer, & Wolf 2007), and neural networks (Tsai, 1994, Husain, & Shigeru, 2000, Lee, Chung, & Tsai, 2003). However, neither in semi-automatic, nor in automatic algorithms, the problems of atypical liver shapes, different modality characteristics and datasets with low contrast adjacent tissues is handled together. Therefore they do not deal with the all variations in CTA images at the same time.

Morphological techniques combined with gray level thresholding are used in (Bae et al., 1993) while in (Gao et al., 1996) and (Lim et al., 2006) these are combined with a parametrically deformable contour model which is used for boundary refinement. Although the method proposed in (Gao et al., 1996) is reported to be successful in most of the cases, a mean gray level value assumption is made for the liver at the intermediate levels of the algorithm. This assumption limits its use when the liver is more attenuating (brighter) due to the contrast media. Deformable contour models are also used by

(Montagnat & Delingette, 1996) and (Chou et al., 1995). However the need for setting some seed points and parameters such as maximum gradient or time threshold makes it hard to use for radiologists. In (Soler et al., 2001), an automatic algorithm is proposed using deformable models; however this method does not provide correct results for atypical liver shapes. Another automatic technique is proposed in (Heimann, Wolf, & Meinzer, 2006), in which a 3-D active shape model is built from 32 samples using an optimization approach based on the minimum description length. The combination of deformable models and statistical priors (Heimann, Meinzer, & Wolf 2007) seems to be effective for fully automatic techniques where initial parameters for the Statistical Shape Model (SSM) are determined with an evolutionary algorithm and a modified active shape method is used to refine the detected parameters. As in (Heimann, Wolf, & Meinzer, 2006), the method of (Heimann, Meinzer, & Wolf 2007) also requires the training of the SSM with a data set to model the expected shape and appearance of the liver so resulting in a dependency on the set of CTA series used in the training.

Artificial neural networks are used for gray level classification in (Tsai, 1994) and for feature based recognition in (Husain, & Shigeru, 2000) which are discussed in detail in this study. The technique proposed in (Tsai, 1994) is semi-automatic and require more than one manually segmented image as training data prior to the automated process. The method in (Husain, & Shigeru, 2000) is not patient oriented and training is done with a limited set of images. Due to the high variation of image characteristics, a larger and more diverse database is recommended to generalize this system for reliable performance. A contextual neural network with a high segmentation performance is proposed in (Lee, Chung, & Tsai, 2003), but the results show that it fails where the gray level of the desired region is too close to the adjacent tissues. In (Koss, Newman, Johnson, & Kirch, 1999), texture of the abdominal organs is used for segmentation. Although this approach is successful in general for abdominal organs, it fails in the segmentation of liver and spleen, especially in atypical liver case, since their texture is similar in CTA datasets. Recently, Seo et al. (Seo, Ludeman, Park, & Park 2004) proposed a fully automatic algorithm by determining the spine first and then by using it as a reference point for segmenting the liver using morphological operators, multimodal thresholding and a decision rule. However, this approach is tested with a very limited set of CT series.



(a) (b)

(c) (d)

Figure 3.1 Examples of abdominal CTA images (a) Very low contrast between liver and vena cava. (b) Low contrast between liver and muscle tissue (c) High contrast between liver and adjacent tissues (i.e. right kidney, muscle, vena cava) (d) Atypical liver shape that causes unclear boundary with spleen.

In this thesis study, a robust and efficient method, which can automatically segment the liver of transplantation donor candidates in CTA series, is proposed. The success rate is calculated as %94.91 over a data set of diverse CTA series of 20 patients according to the evaluation of the expert radiologist experienced on pre-evaluation of transplantation donors for more than 100 cases. The robustness of the method follows from its capability of dealing with the contrast variations and atypical liver shapes. These capabilities are provided by the patient oriented structure which learns the characteristics of a patient data set for each slice in parallel to the segmentation process and adapts its parameters according to these characteristics. The iterative segmentation

algorithm uses classification of pixels (using an unsupervised clustering method i.e. K-means) together with adjacent slice information. A more complex classifier (MLP) is developed for the datasets where the K-means clustering gives insufficient results. The efficiency in terms of the time requirement and the overall segmentation performance is achieved by introducing a novel modular classification system consisting of a simple classification system (i.e. K-Means based) and a complex one (i.e. MLP based) which are combined with a data-dependent and automated switching mechanism that decides to apply one of them. The switching is based on the detection of "low contrast" data set or atypical liver shape. If none of these is detected then K-means based classification system is applied on a single feature (i.e. the gray level value of each pixel), otherwise MLP based classification system is utilized with three features (i.e. mean, standard deviation and distance transform). The developed method gives sufficient performance for different modalities, varying contrast, dissected liver regions and atypical liver shapes. Results indicate that we have effectively overcome the challenging difficulties explained before. This performance is achieved with the proposed modular classification system as well as introducing the distance transform as a feature for each slice and then using this information in the succeeding slice to reveal three dimensional properties of the liver which can not be obtained by the set of slices processed individually. In other words, the approach in the paper provides the ability of dealing with the contrast variations and atypical liver shapes first by recognizing the existence of these problems, by choosing appropriate classification method, and then by solving the segmentation problem using inter-slice information provided by the distance transform.

The rest of this chapter is organized as follows. The properties of the patient datasets are presented in Section 3.2. The following steps of the three step segmentation system are explained in Section 3.3:

1) Preprocessing, covers the removal of adjacent tissues to the liver (i.e. fat tissue, right kidney, spine and ribs)
2) The classification of the liver with modular classification system by using either k-means or a neural network structure depending on the data set properties (i.e. contrast, atypical liver shape)

3) The features used for classification are established in Section 4 as the second step of the segmentation system.
4) Post-processing that consists of the removal of the misclassified objects and identification of disjointed parts of the liver

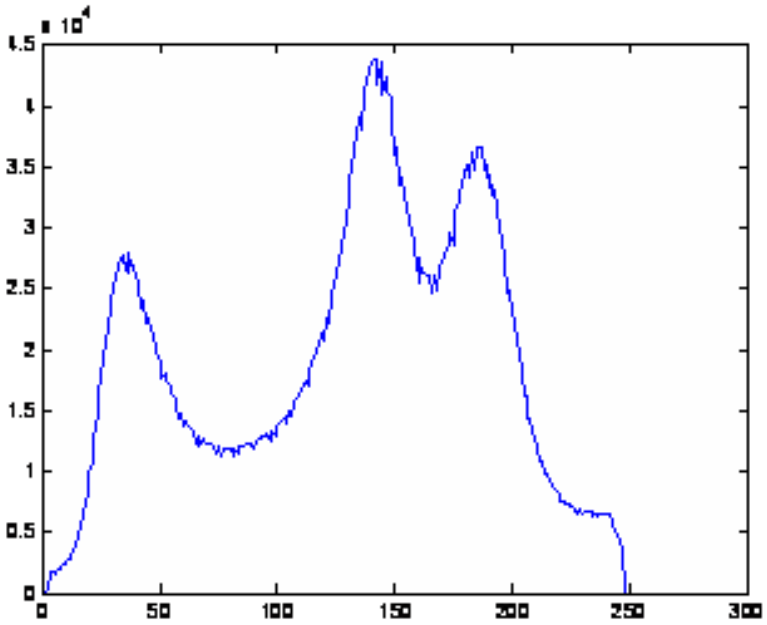
are explained in Section 3.3. The evaluation of the proposed system is given in Section 3.4.

3.2 Patient Datasets

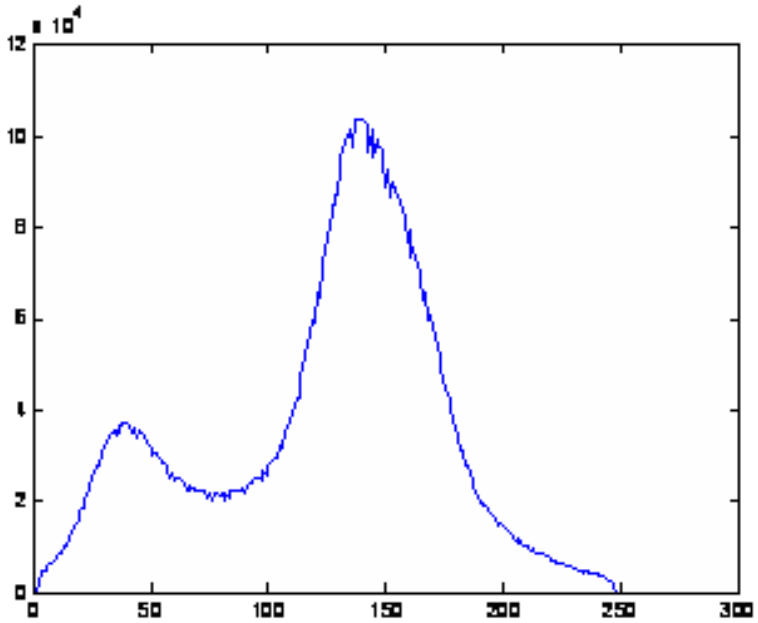
The datasets were acquired after contrast agent injection at portal phase using a Philips Secura CT with 2 detectors and a Philips Mx8000 CTA with 16 detectors, both equipped with the spiral CTA option and located in Dokuz Eylül University Radiology Department. Spiral CTA acquires data continuously, in a spiral path, as the patient is transported at a constant speed through the gantry. This technique scans the entire liver in 15 to 30 seconds and offers several advantages for both liver tumor detection and 3-D visualization. Its speed also reduces or eliminates respiratory mis-registration between slices.

20 datasets (CTA series), which were obtained by these scanners, consist of 12 bit DICOM images with a resolution of 512 x 512. The datasets were chosen randomly from the Picture Archiving and Communication System (PACS). All of the 20 CTA series have 3 to 3.2mm slice thickness and this corresponds to a slice number around 90 (minimum 77, maximum 105 slices).

The segmentation system is designed to work with 8 bit images to support all image types. Therefore 12 bit DICOM images are reduced to 8 bit using window center and window width information, which are stored in Meta information header of the original DICOM images. Although a simple proportional scaling ([0, 4095] → [0, 255]) would be the most obvious way of such a conversion, windowing is used to reduce the undersampling effect. By using windowing, the full contrast of the output display range is expanded over actually useful part of the input density range (Bidaut, 2000).



(a)



(b)

Figure 3.2 Two examples of volumetric histograms (a) Volumetric histogram of a 'high contrast' CTA series one image of which is shown in Fig. 3.1 (c) (three lobes), (b) volumetric histogram of a 'low contrast' CTA series one image of which is shown in Fig. 3.1 (b) (two lobes) (For illustration purposes 0 and 255 are not drawn in the histograms).

20 patient datasets are divided into two groups based on their volumetric histograms, namely ‘high contrast’ and ‘low contrast’. It is observed that some datasets have three lobes in their volumetric histograms (Fig. 3.2 (a)) where these lobes correspond to the fat tissue, darker soft tissues (i.e. muscles, stomach, intestines) and brighter soft tissues (i.e. heart, kidney, spleen) from left to right, respectively. The liver belongs to both second and third lobes with varying ratios due to contrast media it absorbs and modality settings. These datasets are called ‘high contrast’ because the gray level value of the liver is different than the adjacent tissues and organs (Fig. 3.1 (c)). In ‘low contrast’ datasets, dark and bright soft tissues form only one lobe, thus the volumetric histograms have two lobes in total (Fig. 3.2 (b)). In these datasets, it is harder to segment the liver because of the gray level similarity with adjacent organs (Fig. 3.1 (a)). In 20 patient datasets, it is found that 15 datasets belong to ‘high contrast’ group while 5 datasets belong to ‘low contrast group. The developed algorithm first determines the group of the data set and applies different classification and post-processing methods based on this decision.

3.3 Segmentation of the Liver

The developed segmentation algorithm is designed to have three stages. The first stage is preprocessing which consists of the removal of the irrelevant tissues (the fat tissue, the spine, the right kidney and the ribs) from the original images and finding the smallest possible ROI, where the liver tissue is known to exist. The second step of the segmentation procedure is the segmentation of the liver. This step consists of two parts: **1.** Automatic selection and segmentation of an ‘initial image’ **2.** The segmentation of the remaining slices one by one starting from the ‘initial image’. The third step, post-processing, includes necessary operations to remove small mis-segmented objects and to smooth boundaries, Moreover, identification of all components of the liver when the liver dissects into two or more regions is also done at this post-processing stage.

Before starting the process, the user selects a slice which is called ‘initial kidney image’. The ‘initial kidney image’ is the slice where the liver and the right kidney exist together for the last time in the data set. Starting from this image, the algorithm for

removing the right kidney runs through the data set until the right kidney disappears in all slices.

In the proposed algorithm, the default selection of ‘initial kidney image’ is the last slice of the CTA series. Although this assumption is mostly true, an interface with the user is also provided for the cases in which the right kidney does not exist at the last slice. This selection can also be done with no need to user interface in an automatically yet complicated fashion. For simplicity, it is prefered to use this particular one-touch user interface, that is, selection of ‘initial kidney image’. Needless to say, this is a rather simple task for the user.

3.3.1 Pre-processing

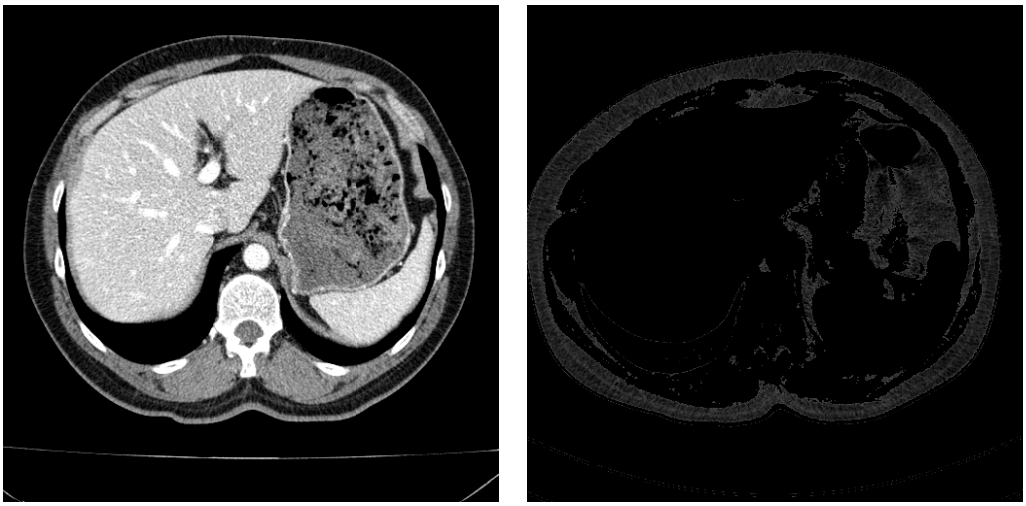
After the selection of the ‘initial kidney image’, the pre-processing starts by removing irrelevant tissues and organs including the fat tissue, the spine, the ribs and the right kidney. It is worth to point that all steps of the preprocessing stage are applied to the original images and the result of each step is removed at the end of the stage.

3.3.1.1 Removing the Fat Tissue

To remove the fat tissue from a patient data set that consists of several CTA images (Fig. 3.3 (a)), an adaptive thresholding method has been applied. In the volumetric histogram of a CTA series, the first lobe of the histogram corresponds to the fat tissue if there is enough fat tissue in the patient. To locate this lobe global minimum/minima of the histogram have to be found. For this reason, an averaging filter is applied to eliminate high frequency components of the histogram. Then the gradient of the smoothed histogram is calculated and global minimum/minima are found where the gradient histogram changes from negative to positive. Finally, a proper threshold value is found as the gray level value of the first global minimum. The tissues, which are removed with the application of the determined threshold value, are shown in Fig. 3.3 (b).

In young or/and fit patients, the fat tissue might be so less that it does not correspond to a lobe in the volumetric histogram. In other words, the first lobes in Fig. 3.2 (a) and

3.2 (b) might be suppressed due to this affect. To prevent wrong determination of the threshold value in these datasets, the maximum gray level value for the fat tissue threshold is limited to 100.



(a) (b)

Figure 3.3 Automatic determination of the threshold to remove fat tissue (a) original abdominal CTA image (b) thresholded fat tissue.

3.3.1.2 Removing the Spine and the Ribs

In the literature, the common method to remove the spine and the ribs is thresholding because of their high Hounsfield values. However this method can also remove some vessels inside the liver or some other enhanced tissues (i.e. the kidneys) due to the brightening affect of the contrast media. These unexpected removals might cause the segmentation process to fail or the loss of important information from the liver. Therefore, thresholding is combined with anatomy knowledge to remove the spine and the ribs in the developed method.

First, an empirically determined threshold value (T=245) has been applied to the image (Fig. 3.4 (a)) to remove fat, skin and muscle tissues. The resulting image consists of the spine, the ribs and contrast enhanced tissues (i.e. liver vessels, kidneys) (Fig. 3.4 (b)). Then, the spine and the ribs must be removed from the thresholded images without losing any information of contrast enhanced tissues. Since it is known that the spine and the ribs surround the contrast enhanced tissues, a topology based method is used. First,

the row and column projections are calculated from the thresholded image. The column projection is used to find the columns where the ribs start (from left and right) by using the first and last non-zero values on it. The row projection is used to determine the row where the spine and ribs start (from the bottom) by using the first non-zero value on it. The middle point of the corresponding row and the values determined as the starting point of the ribs are then used to make a frame through the spine direction (Fig. 3.4 (b)). By dilating this frame, the ribs and the spine are excluded (Fig. 3.4 (c)). To segment the spine and the ribs by combining these two images, Binary Morphological Image Reconstruction (BIMIR) (Vincent, 1993) is used.

BIMIR is based on two images, a marker and a mask. Processing is based on the concept of the connectivity of these images. BIMIR processes the marker image, based on the characteristics of the mask image. The high gray level values in the marker image specify where the processing begins. The processing continues until the gray level values stop changing. If g is the mask and f is the marker, the reconstruction of g from f is defined by the following iterative procedure:

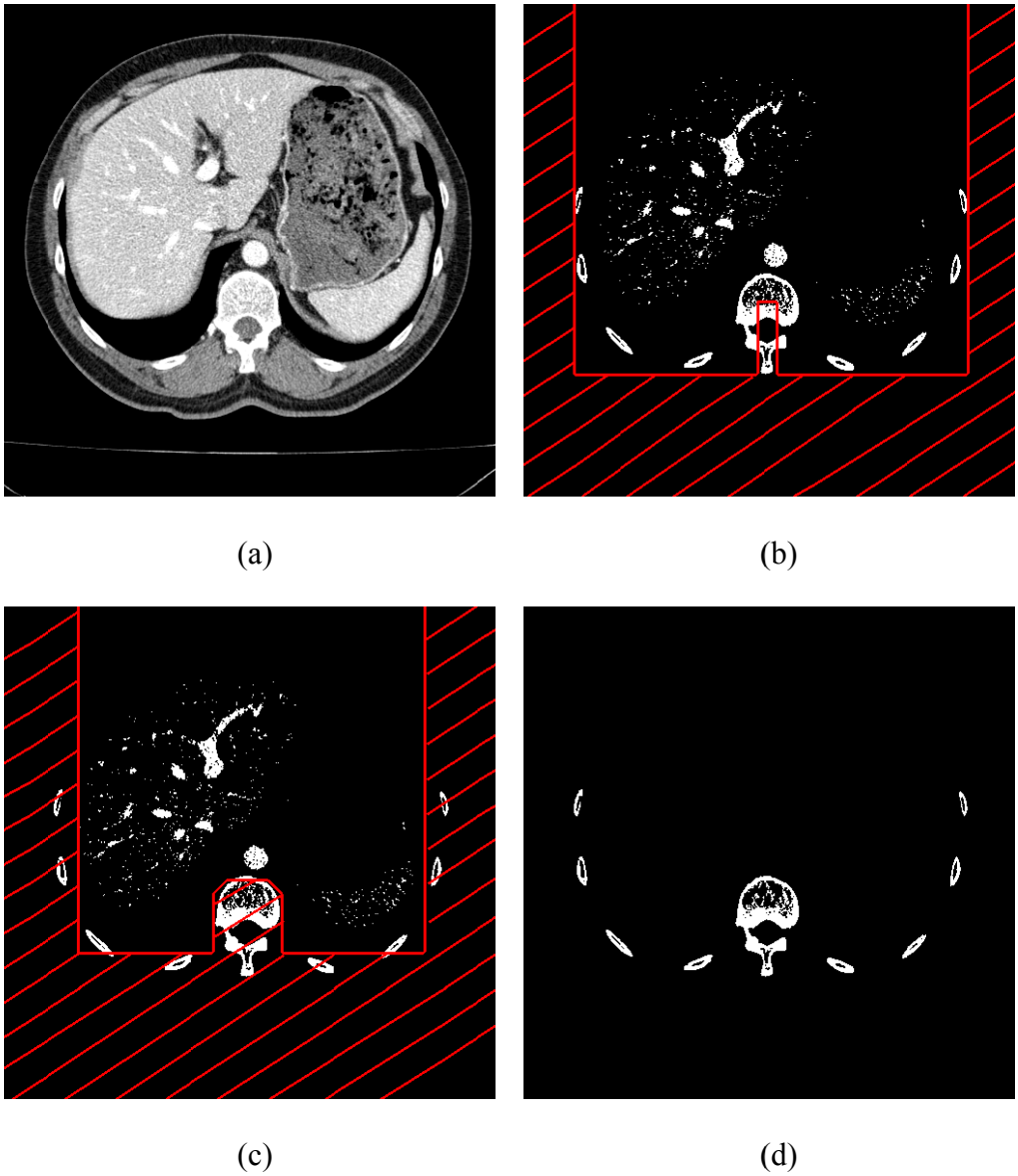
1. Initialization of h_1 to be the marker image f,
2. Creation of the Structuring Element, S,
3. Repeat: $h_{k+1} = (h_k \Theta S) \cap g$ until $h_{k+1}=h_k$,

where the dilation operation, Θ, is defined as

$$h_k \Theta S=\{z \mid (S)_z \cap A \neq \emptyset\}$$

Conceptually, BIMIR can be thought as repeated dilations of the marker image until the contour of the marker image fits under the mask image. In this way, the peaks in the marker image "spread out", or dilate.

By intersecting framed image and the thresholded image using ‘AND’ operation, the marker image is generated. Using the thresholded image as the mask in the BIMIR, the spine and the ribs are obtained (Fig. 3.4 (d)). The advantage of using BIMIR is that it is possible to reconstruct a rib correctly even if some parts of it remain outside the frame as in Fig. 3.4 (c).



(a) (b)

(c) (d)

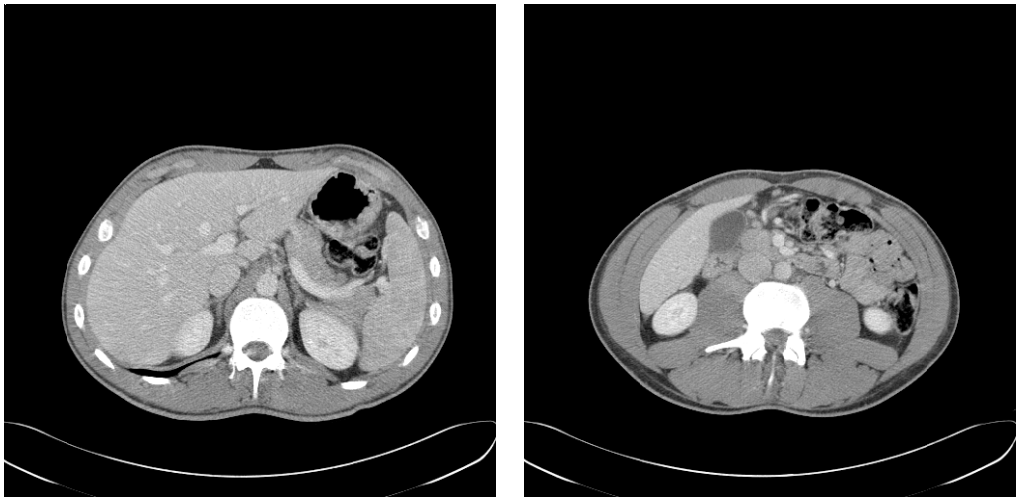
Figure 3.4 Removing the spine and the ribs (a) Original image, (b) frame drawn by using the coordinates of the spine and the ribs calculated from the projections (c) Dilation of the frame and the exclusion of the ribs and the spine (d) Combining thresholded image with the image obtained from the outer side of the frame in (c) using morphological image reconstruction.

3.3.1.3 Removing the Right Kidney

The right kidney is an adjacent organ to the liver which can have different gray level values due to the contrast media (Fig. 3.1 (b) - 3.1 (d)). Moreover, its shape and location differs from patient to patient. Also the shape of the kidney varies through the slices of a CTA series (Fig. 3.1 (c), 3.1 (d)). Because of the partial volume affects, sometimes the border between the right kidney and the liver almost vanishes or in other words,

becomes blurry because of the partial volume effect (Fig. 3.5 (a)). Therefore the elimination of the kidney should be done prior to the segmentation of the liver to increase segmentation performance.

After removing the ribs and the spine, the detection of the right kidney starts from the 'initial kidney image', which is a slice where the border between the liver and the right kidney is clear (i.e. Fig. 3.5 (b)). This user selected slice is mostly the last slice for a CTA series, however sometimes the kidney ends before the liver. Kidney detection process starts from 'initial kidney image' and continues through the beginning of the series until the right kidney is removed from the data set. The reason for the direction of the process is the gray level value similarity and unclear boundary between the liver and the right kidney at the slices, where the right kidney begins to appear (Fig. 3.5 (a)). In such images, the liver and the right kidney could not be separated even by complicated methods. The only way to segment the kidney in those conditions is to use the information obtained from a slice where the liver and the right kidney can be separated easier (Fig. 3.5 (b)).



(a) (b)

Figure 3.5 The right kidney and the liver (a) at the preceding slices where they can not be separated easily, (b) at the succeeding slices where the border between them is more clear.

The removal of the right kidney starts with the classification of the 'initial kidney image' into 5 clusters by using the K-means algorithm. K-means algorithm (MacQueen, 1967) partitions the pixels in the image into n clusters by using an iterative procedure.

The aim is to minimize the sum, over all clusters, of the within-cluster sums of gray level value-to-cluster centers:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{3.1}$$

where $\| x_i^{(j)} - c_j \|^2$ which is a chosen distance measure between a gray level value $x_i^{(j)}$ and the cluster center c_j is an indicator of the distance of the n data points from their respective cluster centers. In the proposed system, Euclidean distance metric and batch update method (Haykin, 1999) are used where every iteration consists of re-assigning gray level values to their nearest cluster centers, all at once, followed by recalculation of cluster centers.

The initial centers of the clusters are chosen to be 30 (for the background), 255 (for the spine and the ribs), threshold found for the fat tissue (generally around 80) and two equidistant gray level values between 255 and the threshold found for the fat tissue. After the application of the K-means method, it is observed that the kidneys are always assigned to the brightest cluster together with the spine and other very bright tissues (Fig. 3.6. top right).

Then, a seed region is generated at the right of the spine, which is found during the removal of the spine and the ribs (Fig. 3.6. top left). The seed region is determined at a location, where at least some part of the kidney is most likely to exist, by using anatomical information.

To segment the kidney from this result, BIMIR is used. In the ‘initial kidney image’ the marker is the seed region and the mask is the image that consists of pixels which belong to the brightest cluster The largest connected component after the BIMIR operation forms the kidney (Fig. 3.6-Output Image).

The kidneys at the other slices are then detected iteratively. For each preceding slice, the skeleton of the previously segmented kidney is used as the new marker image. And the image of the pixels that belong to the brightest cluster of the current slice is used as

the mask image. Using BIMIR algorithm gives the kidney structure of the current slice. The overall procedure is illustrated in Fig. 3.6.

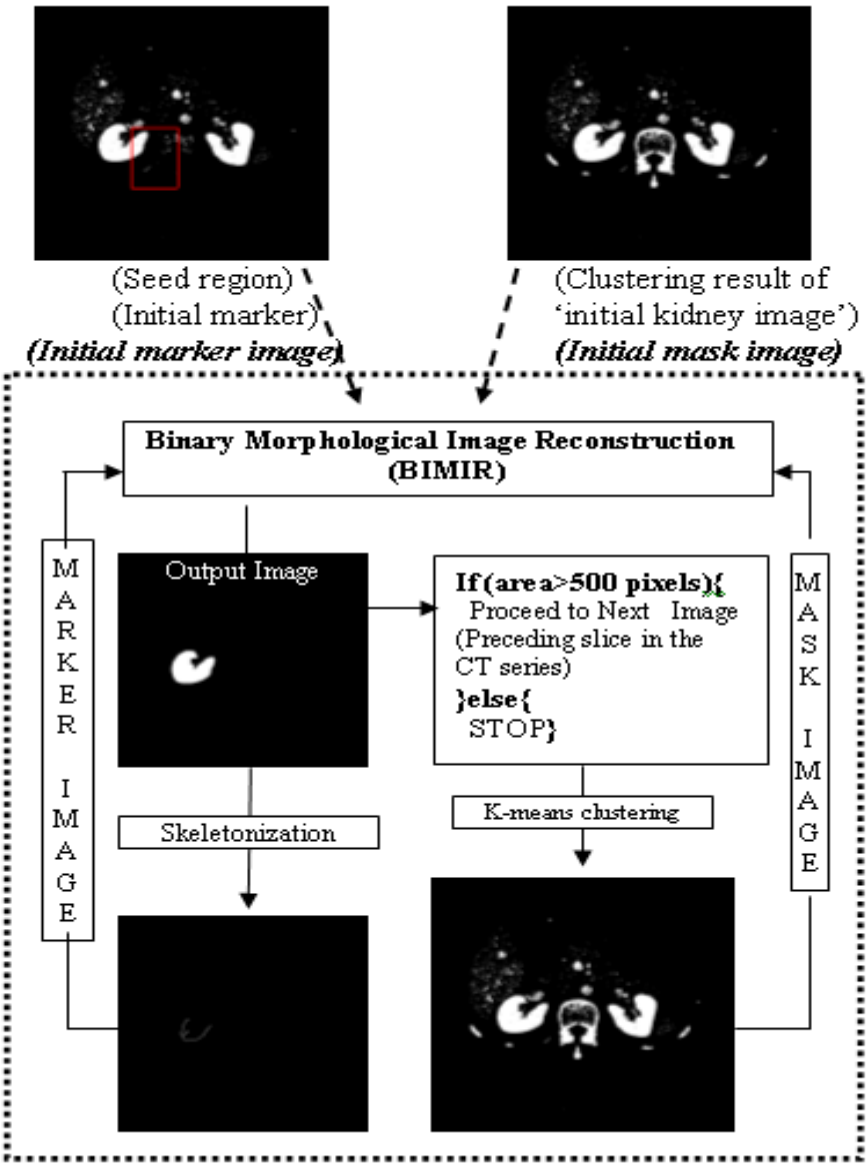


Figure 3.6 Determination of the kidney: Initially, seed region inside the red frame is used as the marker image and the image after K-means clustering (brightest cluster) is used as the mask image. Application of BIMIR using these images gives the right kidney. Then, skeleton of the detected kidney (marker image) and clustering result of the next image (mask image) is used to detect the kidney in the next image.

The skeleton of a kidney is calculated using iterative skeletonization (thinning), which is a method to reduce all objects in an image to lines, without changing the essential structure of the image (Blum, 1967). It removes the pixels on the boundaries of the objects but does not allow objects to break apart. The pixels remaining make up the image skeleton.

Since the skeleton of the previously segmented kidney can be thought as an iteratively eroded image, BIMIR restores exactly the shape of the kidney in the current slice without depending on the similarity between the shapes and the structuring element. By using this method, kidneys are detected automatically until the kidney area drops to a user defined value (default value is equal to 500 pixels).

This method works efficiently for varying gray level values, shapes and positions of the kidney. It also gives sufficient results invariant of contrast and even when the kidney has lesions. This method; skeletonization combined with BIMIR, is also used in the iterative segmentation of the liver.

3.3.1.4 ROI Selection

For reducing the computation complexity and for increasing the performance of the segmentation algorithm, as much irrelevant information as possible should be removed from the image at the preprocessing. From the anatomy knowledge, it is known that the liver is surrounded by the ribs from the left, right and bottom, thus the pixels at the outer side of the ribs can be removed. Also the unnecessary parts from the top (starting from the first non zero pixel) can be removed. The remaining ROI decreases the image size by %40 in average and reduces the computational complexity significantly.

These 4 steps of the preprocessing are done for the complete series of CTA images. At the end of the preprocessing, the fat tissue, the bones, the ribs, and the right kidney are removed from the original images and these images are resized using the ROI mentioned above. An example of a preprocessed image is shown in Fig. 3.9 (a). In what follows, the segmentation algorithm is applied to these preprocessed images.

3.3.2 Classification of the Liver

The second step of the segmentation procedure is the segmentation of the liver. This step consists of two parts: 1.Automatic selection and segmentation of an ‘initial image’ 2.The segmentation of the remaining slices one by one starting from the ‘initial image’.

The ‘initial image’ is a slice where the liver boundary does not overlap with any adjacent organ boundaries, especially the heart and the right kidney. It is selected automatically by choosing the slice that comes just before the first appearance of the right kidney which is determined during the kidney removal stage. After the preprocessing stage, the segmentation algorithm starts from this ‘initial image’ and then runs through the end of the data set. Then starting from the ‘initial image’ again, it runs through the beginning of the data set to complete the segmentation process. After the segmentation of the initial image, liver structures at other slices are segmented iteratively.

For this purpose, a novel modular classification system is introduced consisting of a simple classification system (i.e. K-Means based) and a complex one (i.e. MLP based) which are combined with a data-dependent and automated switching mechanism that decides to apply one of them. The introduced modular classification system with data-dependent and automated switching mechanism constitutes a new kind of system of classifiers (Fig. 3.7), some of which are simple and therefore efficient in time-memory requirements with good generalization ability and the others are complex providing a high classification performance, such that depending on the data set, herein the CTA series, one of the classifiers become active.

The switching is based on the detection of “low contrast” data set or atypical liver shape. The switching mechanism does indeed perform a classification task that assigns the CTA series, based on the histogram evaluation and in some cases also according to intermediate results of the K-Means based classifier, into one of the following three categories: a) Low contrast (MLP is employed for this category), b) High contrast (K-means is employed for this category), and c) High contrast- atypical liver shape (MLP is employed for this category). In other words, the switching mechanism selects MLP classifier instead of the K-Means classifier where the K-means clustering method does

not give sufficient results (i.e. if the liver has atypical shape (i.e. Fig. 3.1 (a)) or the gray level difference of adjacent tissues is very low (i.e. Fig. 3.1 (d)). For this reason a process for automatic detection of an atypical liver shape and a low contrast image characteristic is developed (Fig. 3.8).

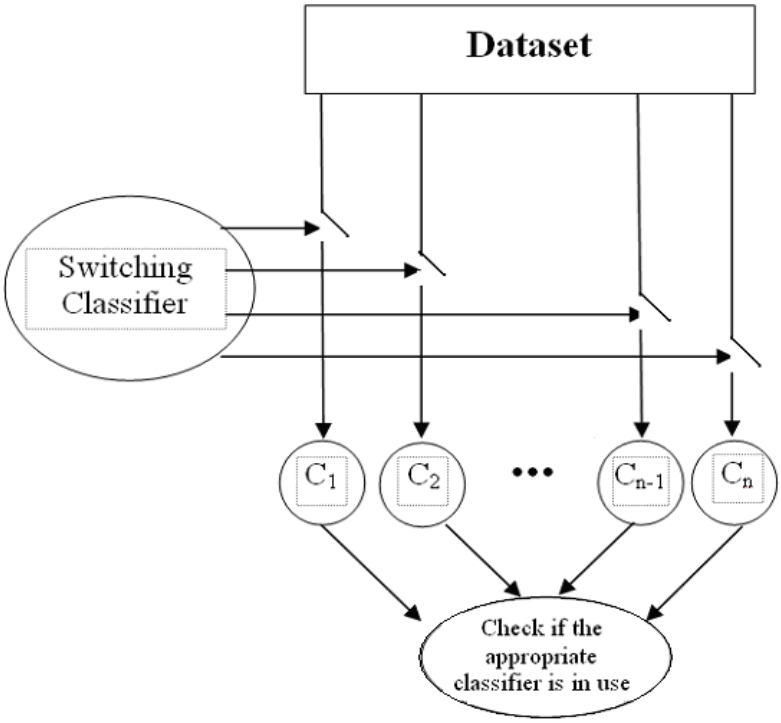


Figure 3.7 Proposed modular classification system which constitutes a new kind of system of classifiers. In our case, "n" is equal to two since only K-Means and MLP classifiers are used.

In this manner low contrast image characteristics are detected by determining the number of lobes in the volume histogram (discussed in Section 2 and 3.1). If the histogram has three lobes, the algorithm proceeds with K-Means. Otherwise (if it has two lobes) the algorithm switches to MLP.

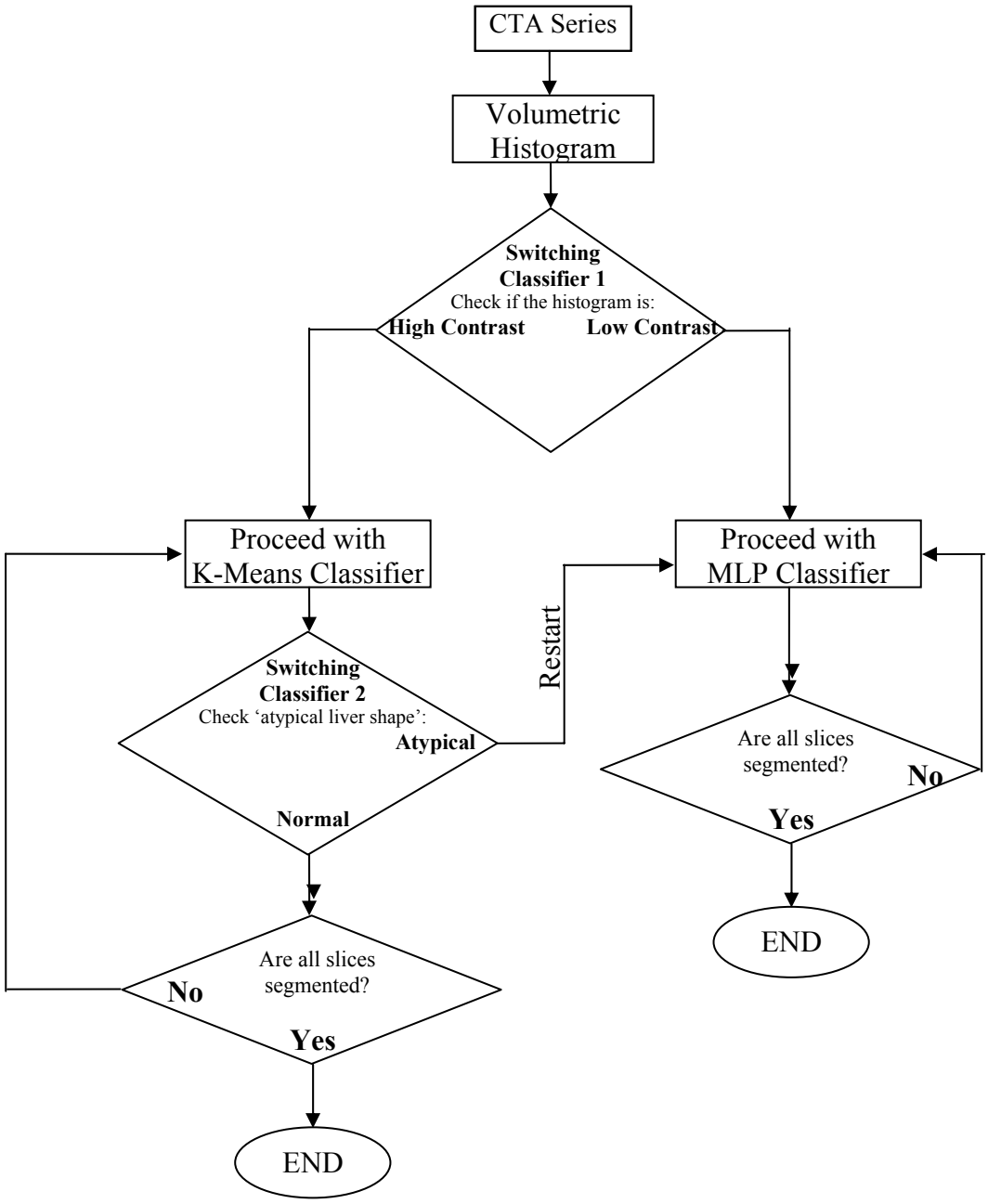


Figure 3.8 Switching mechanism and selection of the appropriate classifier for automatic detection of an atypical liver shape and a low contrast image characteristic.

The other criterion in the selection of the classifier to use is the atypical liver case. Existence of atypical liver shape means that the liver elongates (stretches) to the left side of the abdomen and becomes adjacent to the spleen which has mostly the same gray value range as the liver (Fig. 3.1 (d)). In those cases, the boundary between the spleen

and the liver is uncertain even for human eye. Therefore the algorithm may not detect the boundary between the spleen and the liver and segments them together. This causes a dramatic increase in the area of the segmented object. To detect atypical liver shape, the area of the segmented object for each slice is found. Since the change in the area should be smooth for successive slices the algorithm switches to MLP if there is a sudden increase in the area.

It is worth to point that, the result of both classifiers (i.e. K-Means and MLP) gives rough results and these results are refined at the post-processing stage (i.e. removal of the small mis-segmented objects, identification of all components of the liver when the liver dissects into two or more regions). However, it is worth to point that when MLP is used as the classifier, the necessary post-processing operations is significantly less than the operations used after K-Means classifier.‘

3.3.2.1 Initial Image Segmentation

As previously mentioned, ‘initial image’ is a slice that should satisfy three constraints. First of all, the liver should have a relatively big area in the slice but it does not have to be the biggest organ. Second, it should consist of one connected component and third, the liver boundary should not overlap with any adjacent organs, (i.e. the heart and the right kidney). An example of an initial image is given in Fig. 3.4 (a). After preprocessing (Fig. 3.9 (a)), the image can be classified into three clusters: background, bright organs and dark organs. Excluding the background pixels from the process, the two clusters are found using the Otsu’s method (Otsu, 1979) by finding the optimal threshold to separate dark organs (stomach and muscle) and brighter organs (liver, spleen, and heart).

Otsu’s method chooses the threshold value k that maximizes the between-class variance σ^2_B which is defined as:

$$\sigma^2_B = w_0.(\mu_0-\mu_T)^2 - w_1.(\mu_1-\mu_T)^2 \qquad (3.2)$$

where

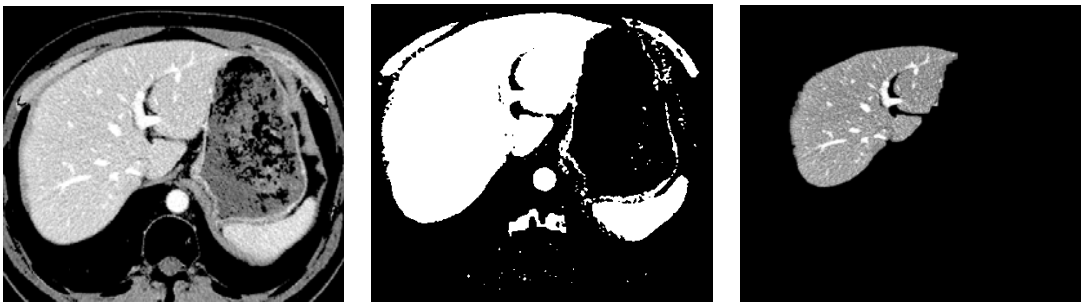
$$w_0 = \sum_{q=0}^{k-1} p_q(r_q), w_1 = \sum_{q=k}^{255} p_q(r_q),\ \mu_T = \sum_{q=0}^{255} q.p_q(r_q)\ ,$$

$$\mu_0 = \sum_{q=0}^{k-1} q.p_q(r_q)/w_0\ \text{, and}\ \mu_1 = \sum_{q=k}^{255} q.p_q(r_q)/w_1$$

Here $p_q(r_q)$ is the discrete probability density function, as in $p_q(r_q)=n_q/n$ where n is the total number of pixels in the image, and n_q is the number of pixels that have intensity level r_q (q=0,1,,255).

Selecting the cluster which has the foreground organs (i.e. the right side of the threshold) (Fig. 3.9 (b)) and then taking the biggest connected component in the slice results with the segmented liver structure (Fig. 3.9 (c)).

It is important to point that this method gives sufficient results only under the constraints given for the 'initial image'. Considering the datasets used in this study, the slices, which are located around the one third of the series (i.e. 30th slice of 90 slice CTA series), are determined to be suitable for being 'initial image'. However, it is also possible to select any slice in the series, which satisfies the necessary conditions, as 'initial image'.



(a) (b) (c)

Figure 3.9 Initial image segmentation (a) The preprocessed 'initial image' (b) Clustering result (c) The biggest component in the slice.

3.3.2.2 Segmentation with K-Means

After segmenting the initial image, the algorithm first runs downwards to the last slice and then upwards to the first slice starting from the initial image (Fig. 3.10 (a)).

It is clear that the preprocessed images contain the liver, the tissues and the organs that have similar or darker gray level values than those of the liver because the brighter tissues and organs are removed at the preprocessing phase (except the bright tissues inside the liver).

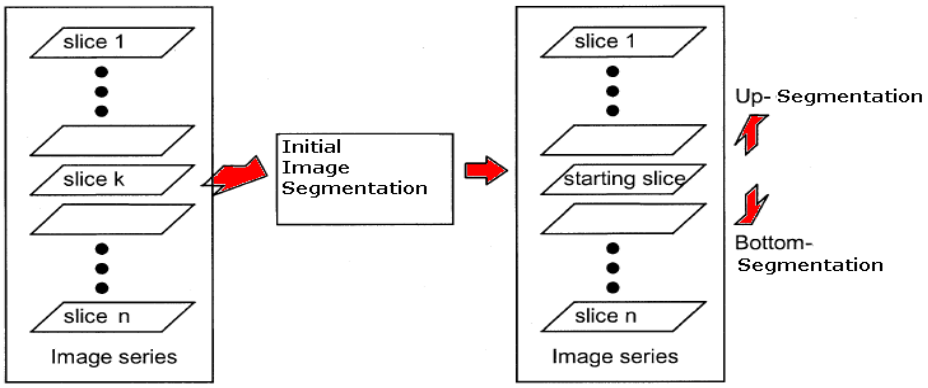
By using the K-means method an unsupervised clustering is applied to these preprocessed images for classifying the organs into two clusters. At this step, the initial cluster centers are determined as follows: For the 'initial image', the first cluster center is given as the minimum gray level value of that image (excluding the background). The second cluster center is determined as the mean value of the segmented initial image. Then for the other slices, the centers found in the preceding slice are used as the initial centers.

The brighter cluster at the clustering result is preserved since it always consists of the liver. The identification of the liver including its dissected parts and removal of the incorrectly segmented objects is done at the post processing stage.

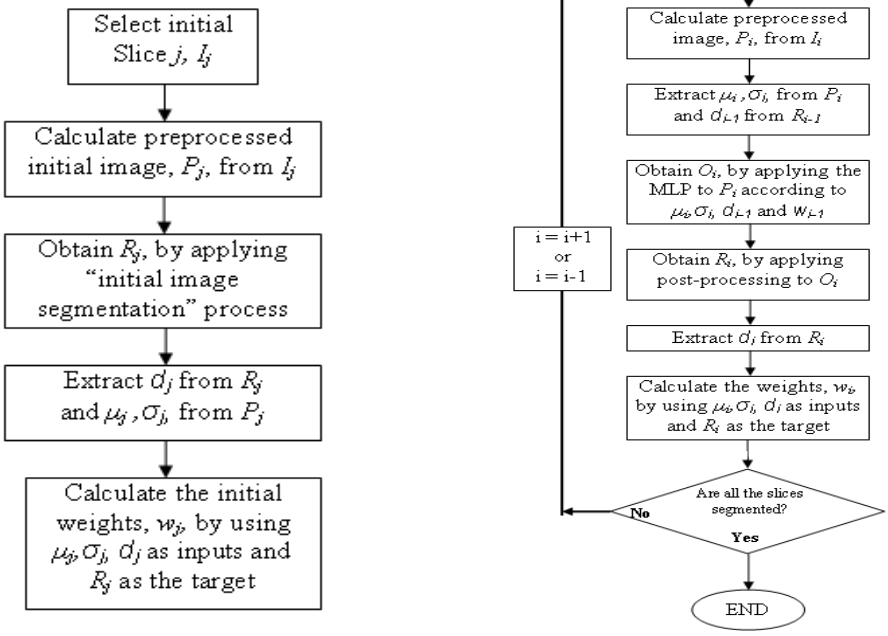
3.3.2.3 Segmentation with MLP

Although segmentation of the liver using the K-means clustering method generally gives sufficient results, it fails when the liver has atypical shape (Fig. 3.1 (d)) or the gray level value difference between the liver and the adjacent tissues are very low (Fig. 3.1 (a), 3.1 (d)). To obtain acceptable results also for these cases, a feature based segmentation process is developed. In feature based segmentation, first, K-Means is tested as the classifier. However, the segmentation results were not adequate hence a more complex classifier, Multi Layer Perceptron (MLP) is used instead. The overall structure of the segmentation process using MLP is shown in Fig. 3.10.

In segmentation with the MLP, the preprocessed 'initial image' and the segmented 'initial image' are used for initial training of the network. From the preprocessed 'initial image' two features (Mean and standard deviation) are calculated. Then, the distance transform is applied to the segmented 'initial image' and the pixel values after the transform are taken as an additional (third) feature.



(a)



(b) (c)

Figure 3.10 (a) Segmentation process by using MLP. (b) Preprocessing of all images is followed by the selection and segmentation of the initial image. The initial training is done by using the segmented initial image as the desired output and the feature vectors obtained from initial and segmented initial images as training data, (b) Then, the algorithms proceeds to next slices and at each slice previously found weights are used for classification with MLP. The weights are updated using the current segmented image as the desired output and the feature vectors obtained from the current slice as training data. In the figure μi,σi, and di corresponds to mean, standard deviation and the distance transform features obtained from slice I, respectively where i = j-1, j-2, , 2, 1 for up-segmentation and i = j+1, j+2,, N-1, N for bottom segmentation with N is the number of slices.

The features are calculated in a window of size 9x9 centered for a given pixel. Although some sudden changes in the image (i.e. edges) can be identified more accurately with a smaller window size, the optimal size for identifying the liver region is decided to be 9x9 after extensive experimentation by concerning to prevent finding details inside the liver area and to get better information about the orientation of the liver. Also in (Tsai, 1994, Husain, & Shigeru, 2000), 5x5 window size is presented to be optimal for 256x256 and 240x320 images, therefore using 9x9 for 512x512 images is appropriate.

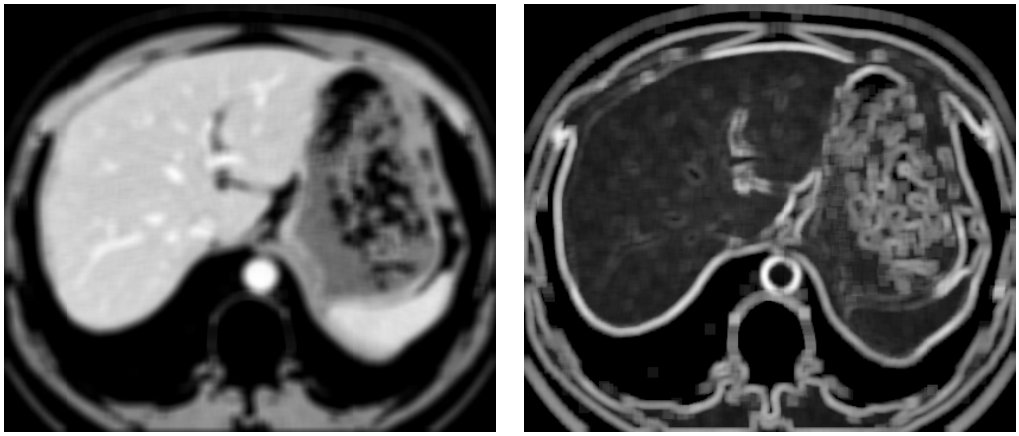
To represent the homogeneous regions (i.e. Liver parenchyma) in the current slice, the mean feature is used (Fig. 3.11 (a)). For a pixel the mean gray level value is calculated by

$$\bar{x}_{ij} = \frac{1}{N} \times \left(\sum_{i-4}^{i+4} \sum_{j-4}^{j+4} x_{ij} \right) \tag{3.3}$$

To represent the edges (i.e. liver boundary) the standard deviation feature is used (Fig. 3.11 (b)) which is calculated as:

$$\sigma_{ij} = \frac{1}{N} \times \left(\sum_{i-4}^{i+4} \sum_{j-4}^{j+4} \left(x_{ij} - \bar{x} \right) \right) \tag{3.4}$$

where $\bar{x}_{ij}$ is the mean value of the pixel x located at the position i, j and σ_{ij} is the standard deviation of the pixel located at the same position. N is the total number of pixels in the window which is equal to 81 in the proposed approach.



(a) (b)

Figure 3.11 Feature images (a) mean (b) standard deviation

Finally, the distance transform feature is used to represent the previously segmented image. The distance transform provides a metric that measures the separation of the pixels in the image. The metric is calculated to measure the total Euclidean distance along the horizontal, vertical, and diagonal directions (Fig. 3.12 (a)).

In the proposed algorithm, the distance transform gives information about the liver location at the adjacent (preceding/succeeding) slice. Since the liver size and location does not change dramatically between adjacent slices, the distance transform of a segmented liver (Fig. 3.12 (b)) gives quite important information about the liver location at the adjacent (preceding/succeeding) slice.

2.8	2.2	2	2.2	2.8
2.2	1.4	1	1.4	2.2
2	1	0	1	2
2.2	1.4	1	1.4	2.2
2.8	2.2	2	2.2	2.8

(a)

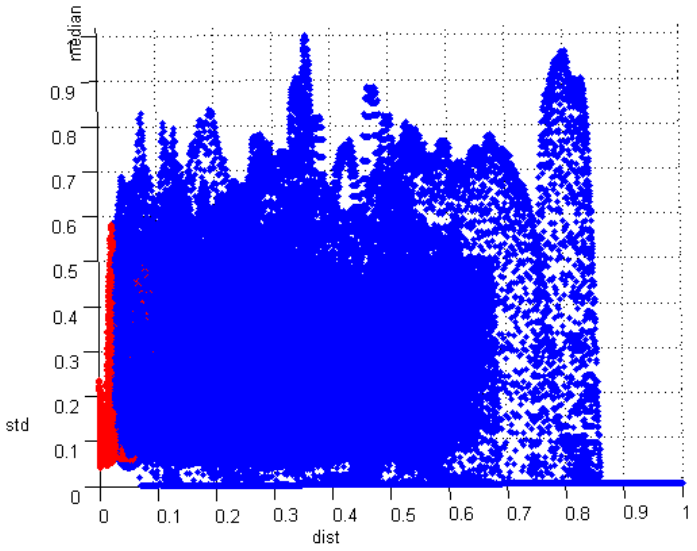
(b)

Figure 3.12 Distance transform process (a) Distance transform of a binary image in which only the pixel at the center has value 1 (b) Distance transform of a segmented slice.

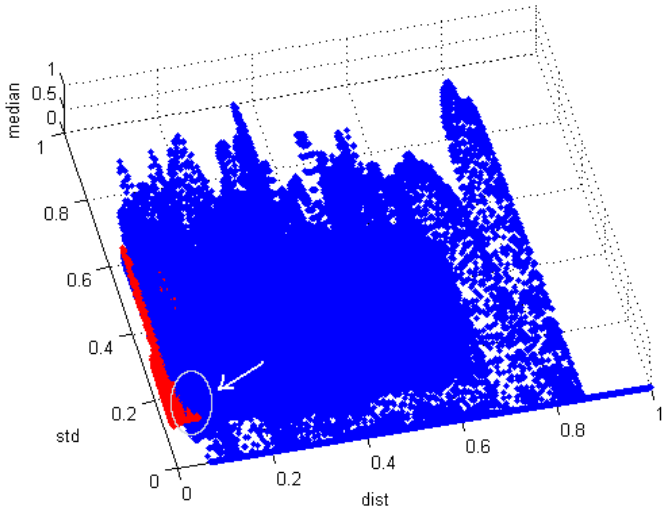
The distribution of the features in the feature space is shown in Fig. 3.13. The lighter data points (x: [0 0.1], y: [0 1]) correspond to the pixels that belong to the liver in the figure. From Fig. 3.13 (a), it is observed that the most discriminative feature is the distance transform. However the distance transform is not sufficient to discriminate the data inside the circle in Fig. 3.13 (b). It is observed that the data inside the circle becomes separable by using the mean and standard deviation features.

After the extraction of features, first, these three features, distance transform, mean and standard deviation, are used as the inputs of the K-Means clustering algorithm. Although, there is a slight increase in the performance of classification, it is observed that atypical liver cases and 'low contrast' CTA series can not be segmented properly. Therefore, a more complex classifier is needed and MLP is chosen as the classifier.

The network is trained initially by using the segmented initial image as the desired output and the calculated three features as the training data. At the network output, each input pixel is classified as belonging to the liver region or lying outside the liver region on the basis of these features.



(a)



(b)

Figure 3.13 Feature space from two different views, (a) The most discriminative feature is the distance transform since most of the data is separable along the distance transform space. (b) However, it is not sufficient to discriminate the data inside the circle and additional features (i.e. mean, standard deviation) are required.

After this initial training, weights are updated and iteration proceeds to the next slice. Mean and standard deviation features are calculated for the preprocessed next slice

(which is the current slice to be segmented) and the distance transform is calculated from the previously segmented image. By using these features and the weights (obtained from the previous slice), the current slice is segmented. After the segmentation of the current slice, the network is trained again by using the features (mean, standard deviation and the distance transform calculated for the segmented images of current slice) as the input and the new segmented image as the desired output. After the training and calculation of the weights, the algorithm proceeds with the next slice and this iterative procedure continues until all images are processed. Using the previously adjusted weights as the initial weights of the next training phase, the training time is reduced significantly.

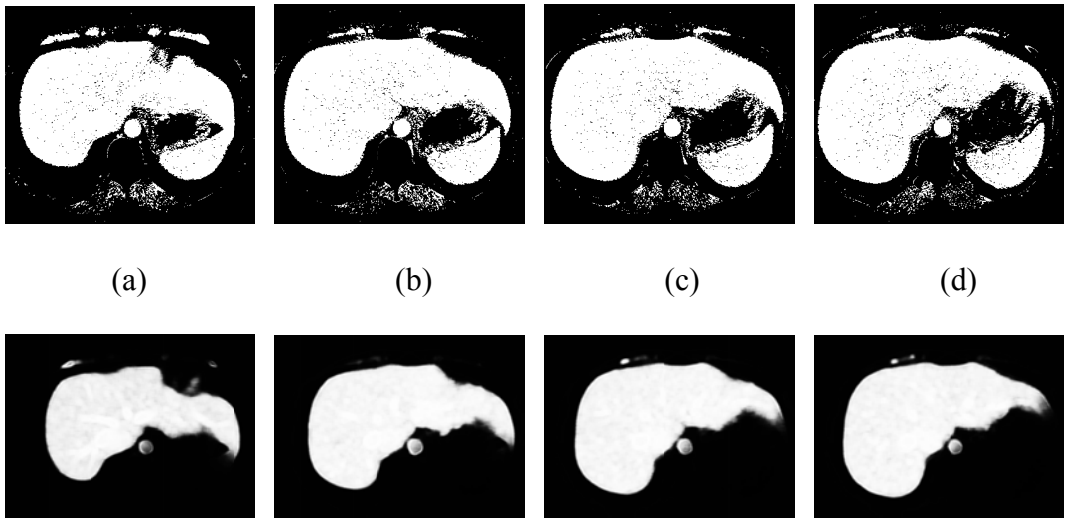
The MLP structure used for the segmentation consists of 3 neurons at the input layer, which corresponds to the number of input features. There are 8 neurons at the hidden layer, each of which has a bias input that ranges between +/- 1. The biases are updated along with weights during error backpropagation (Haykin, 1999). The output layer consists of l neuron. The output of the network lies between 0 and 1 for each pixel and it is thresholded by 0.5. Then, for an input region belonging to the liver class, the output is designed to be unity whereas for all other the output is designed to be zero. This network structure is determined to be the optimum after extensive experimentation and due to a compromise between efficiency and reliability.

3.3.3 Analysis and Classification of Features and Classifiers

In (Husain, & Shigeru, 2000), five optimum features are reported as follows: mean gray level, standard deviation, skewness, entropy, homogeneity. However, these statistical descriptors are not sufficient to differentiate two organs/tissues that have similar texture and/or statistical properties. This drawback limits the usage of these features in atypical liver shapes where the border between the spleen and the liver vanishes. Since the spleen has almost the same texture and statistical properties as liver, it becomes very difficult to segment the liver without the spleen. Moreover, in ‘low contrast’ CTA series, the statistical properties of muscle tissues and vasculature (i.e. aorta, inferior vena cava) get closer to the liver which also hardens the correct segmentation of the liver by only using these features without any spatial information.

To overcome these problems, the distance transform is selected as an additional feature. The approach in the selection of the distance transform is to simulate the decision process of a radiologist. For instance, in the case of atypical liver shapes, the unclear border between the liver and the spleen is determined by the radiologist by following the slices (especially the ones just before and after) where the border is more visible. Similarly, the distance transform feature provides a metric to represent the liver at the previously segmented slice which gives information about the liver location at the adjacent (preceding/succeeding) slice. Although this feature is affected by the slice spacing of CTA data, the Slice Thickness (ST) of the datasets used in this study is 3.2 mm, which is a rather big value considering the emerging technology of CT modalities. Even with this thick ST, the information provided by the distance transform is enough to handle atypical liver shapes and 'low contrast' datasets.

In the following figure, an example case is presented to show the advantages of using distance transform for detection of the liver. Fig. 3.14 shows a set of images that are selected from a CTA series where the liver has atypical shape (Fig. 3.1 (a)). Fig. 3.14 (a)- (d) shows the outputs of the K-Means classifier while Fig. 3.14 (e)- (h) represents the outputs of the MLP classifier. The slice numbers are 33, 31, 28, and 24 from left to right where the slice 33 is the initial image to be segmented. As shown in Figures 3.14 (a), 3.14 (e); both MLP and K-means are successful in classifying the liver without any connectivity with other objects. However, as the segmentation process continues through the beginning of the CTA series, the liver gets closer to the spleen (Fig. 3.14 (b), 14 (c)) and they finally they merge (Fig. 3.14 (d)). Since the texture and gray level variation of the liver and the spleen are very similar, it is necessary to use the information obtained from the previously segmented slices. The distance transform allows the usage of this information as it limits the search area to a region that is slightly bigger than the previously segmented liver region. As shown in Fig. 3.14 (f) - 3.14 (h), the distance transform prevents the spleen to appear at the output of the classifier.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.14 The effect of the distance transform in segmentation of atypical liver. Each figure shows the output of the classification results with K-Means or MLP without any post-processing (a) The 'initial image' can be segmented by K-Means (Slice 33), (b), As the slice by slice segmentation proceeds, the spleen, which has almost the same texture and statistical properties as liver, becomes closer to the liver (Slice 31), (c) As spleen and liver gets closer, their border can not be identified clearly since they belong to the same cluster due to the similar characteristics (Slice 28), (d) When the borders of the liver and the spleen intersects, the clustering is not enough to classify these two organs (Slice 24) (e) The 'initial image' can also be segmented by MLP (Slice 33), (f), (g) As the slice by slice segmentation proceeds, the distance transform limits the classifier output to a region close to the location of previously segmented liver (Slices 31-28) (h) The effect of distance transform prevents the misclassification of the spleen even when the border between the liver and the spleen vanishes (Slice 24).

Here, it is worth to point that the distance transform is combined with the features proposed in (Husain, & Shigeru, 2000) as another experiment. However, after extensive simulation studies no noticeable increase is observed in the performance of segmentation when skewness, entropy, and homogeneity are included in the feature set. But these three features increase the computational burden and slow down the process dramatically. Therefore, skewness, entropy, and homogeneity features are not used during classification.

Another important issue is the training of the neural network since it has significant effect on the performance and computation time. The technique proposed in (Tsai, 1994) requires more than one manually segmented image as training data which is undesired

for automated processes. In (Husain, & Shigeru, 2000), training is done with a limited set of images and due to the high variation of image characteristics, a larger and more diverse database is recommended to generalize this system for reliable performance. After the adjustment of the network weights using the training set, these fixed weights are used for segmentation of other datasets. However, this approach is error prone and needs new training sets for the datasets with new image characteristics such as different modalities and modality settings. Moreover, simulations show that using a fixed hyper-plane (network weights) to segment all images in a CTA series, decreases the segmentation performance. This means that a new set of weights for each image increases the segmentation performance and the weights should be adjusted during the segmentation of different slices of the same CTA series. To provide this adjustment, the weights of the proposed network are updated at each slice. This is done by using the original form of a slice as the input and the segmentation result as the desired output of the training. Initialized by the "initial image", the previously adjusted weights are used in the segmentation of the next slice. The same weights are also used as the initial weights of the next training phase which reduce the training time significantly. A similar training methodology is proposed in (Lee, Chung, & Tsai, 2003), where a contextual neural network with a high segmentation performance is proposed. But the results show that it fails where the gray level of the desired region is too close to the adjacent tissues since the proposed method is designed for the segmentation of all abdominal organs.

3.3.4 Post-processing

The results obtained from the segmentation algorithms are roughly segmented liver structures (Fig. 3.15 (a)). To remove small mis-segmented objects and for boundary smoothing, a post-processing is needed. Moreover, identification of all components of the liver when the liver dissects into two or more regions is also done at the post-processing stage.

Post processing is handled differently for the slices before and after the "initial image" slice because of the different image characteristics.

In the post processing phase of the slices after the 'initial image', a series of nonlinear filtering and morphological operations is applied to separate weakly connected

components in the clustered binary image. First, a median filter is applied to remove the white spots appear at the background and black spots appear inside the liver (Fig. 3.15 (b)). Then, erosion operation is applied to eliminate the small unconnected objects that do not sit within the structuring element (Fig. 3.15 (c)). Next the skeleton of the previously segmented liver is obtained using skeletonization (Fig. 3.15 (d)). By using BIMIR, in which the skeleton is used as the marker image and K-means or MLP result as the mask image, the liver is identified (Fig. 3.15 (e)). Usage of skeletonization also provides the important information about the separated parts of the liver when it dissects into two or more regions (Fig. 3.15 (d)). The algorithm searches for the second and the third components automatically for the slices after the 'initial image'. After that, dilation operation is applied to restore the contours that have not been completely removed by the erosion (Fig. 3.15 (f)). Finally a Gaussian filter is applied to the edges to smooth the contours.

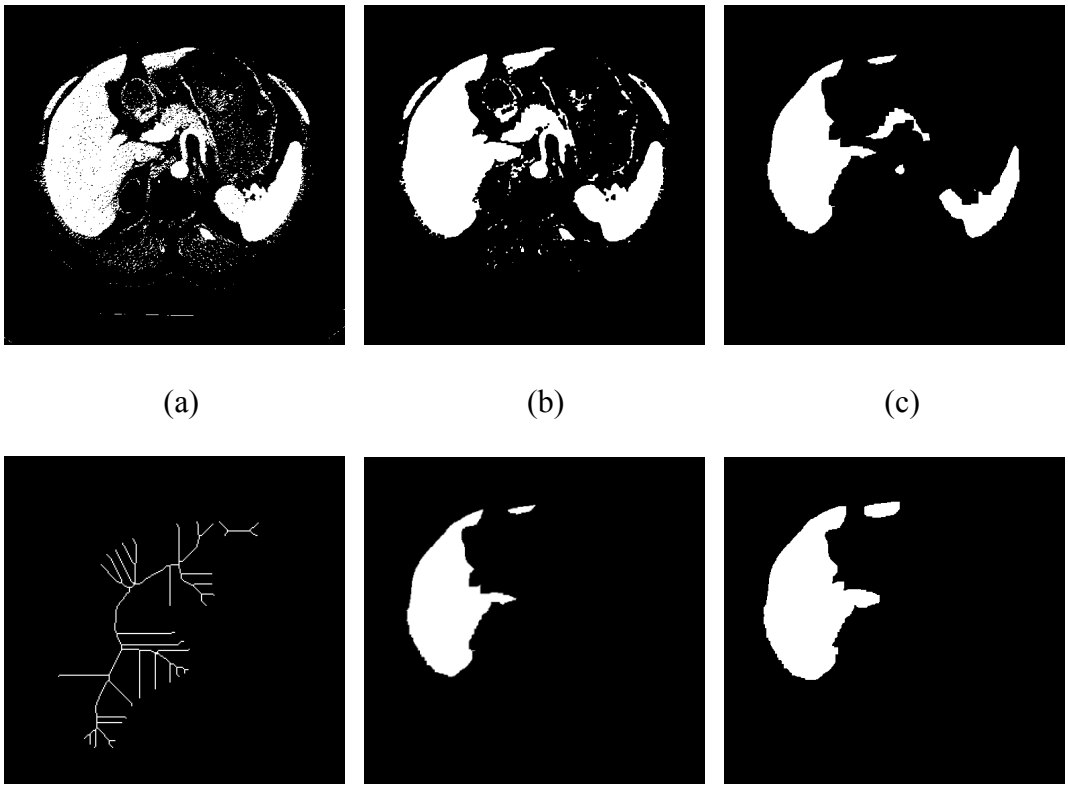
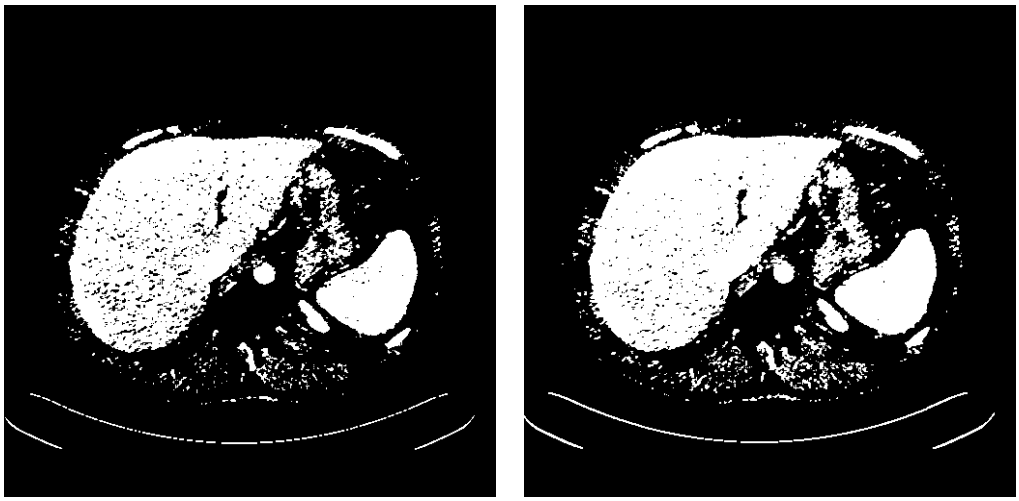


Figure 3.15 Post-processing stage for a slice: Results of (a) K-means clustering (b) Median filtering (c) Erosion (d) Skeletonization of the previously segmented liver (e) BIMIR using median

filtering result as the marker and skeleton as the mask (f) dilation to restore the contours that have not been completely removed by the erosion.

The difference between post processing of the slices before and after the initial image is due to the existence of heart tissue in the slices before the initial image. For the slices before the initial image, the liver area from the previously segmented slice is not dilated because the heart can also be included due to gray level similarity of it with the liver. Another difference is that the algorithm looks for only one connected component for the first slices.

Post processing is handled differently also for the low and high contrast datasets. In low contrast datasets; nonlinear filtering is applied in a different way such that the value after the median is taken instead of the median value. This kind of filtering increases the connectivity of the liver by eliminating more dark spots than median filtering. (Fig. 3.16)



(a) (b)

Figure 3.16 (a) Affect of median filtering (b) affect of modified nonlinear filtering.

3.4 Evaluation

The segmentation results are evaluated by using the Area Error Rate (AER) (Seo et al., 2004). AER is defined as the area difference between the Region Segmented by the

Algorithm (RSA) and the Region Segmented Manually (RSM). Defining a union region RU as RSA U RSM and an intersection region RI as RSA ∩ RSM, AER is equal to:

$$AER = \frac{RU - RI}{RM} \times 100\% \qquad (3.5)$$

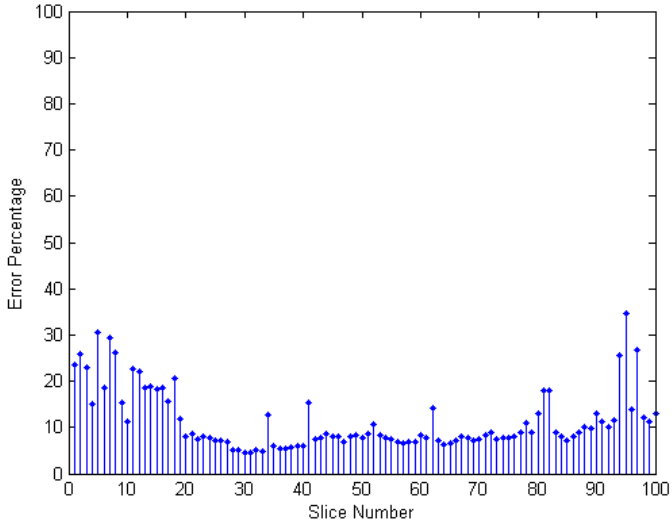
AER is similar to the criteria Volumetric Overlap Error (VOE) which is used in (Van Ginneken, Heimann, & Styner, 2007). VOE is defined as:

$$VOE = \frac{RU - RI}{RU} \times 100\% \qquad (3.6)$$

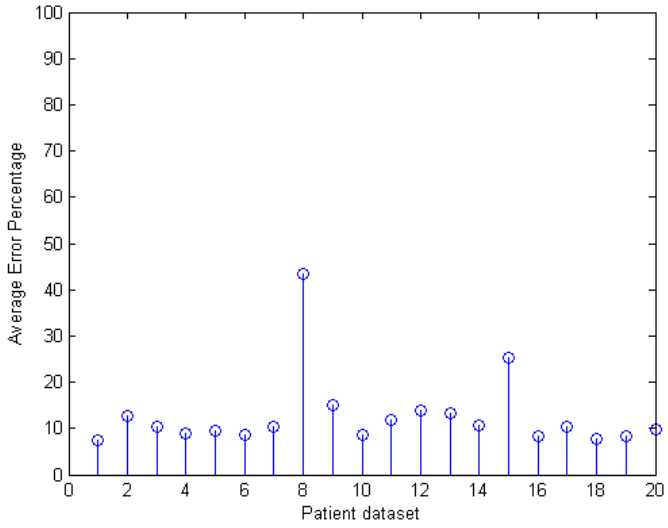
VOE is equal to 0 for a perfect segmentation and has 100 as the lowest possible value, when there is no overlap at all between segmentation and reference. AER also takes 0 values for a perfect segmentation however the lowest possible value is not limited to 100. For evaluation, AER is calculated directly (without any boundary modification) between the manually and automatically segmented images. The manually segmented images are segmented by an expert radiologist from Dokuz Eylül University Radiology department.

The slice by slice average AER for 20 datasets (‘5’ low contrast and 15 ‘high contrast’, 3 of which have atypical liver shape) with the K-means algorithm is shown in Fig. 3.17 (a). It is observed that the algorithm shows better performance for the slices at the middle and at the end of the datasets. The high AER values for the slices at the beginning of the datasets are due to unclear boundary between the heart and the liver. The average AER for the complete data set is calculated as % 12.15 by using the K-Means algorithm.

As explained in Section 3.2 the overall algorithm switches automatically from K-Means to MLP when it is needed. To illustrate the effect of this switch on the segmentation performance several experiments are performed. Fig. 3.17 (b) shows the average AER for each patient data set when the system just uses the K-Means. The high AER values in datasets 8, 9 and 15 are due to atypical liver shapes.



(a)



(b)

Figure 3.17 (a) AER for 20 datasets (b) Average AER calculated for each data set if just K-Means algorithm is used.

As mentioned above, the main points where the K-means algorithm fail are the first slices where the heart and the liver can hardly be segmented even with the human eye, the patient datasets with atypical liver shapes (Fig. 3.18 (a)) and the separation of the tissues in 'low contrast' data set where the gray level value of the adjacent tissue (organ or vessel) is very close to the liver (Fig. 3.18 (c), 18.e). The algorithm with the neural

network, MLP, classifier solves these problems and performs better segmentation in those cases (Fig. 3.18 (b), 3.18 (d), 3.18.f). These high AER values are reduced from %41.20 to %12.73 in data set 8, %18.20 to % 9.95 in data set 9, and %29.8 to %11.30 in data set 15 by using MLP. The high AER values in 2, 12 and 13 are due to low contrast between the liver and its adjacent tissues, organs. These high AER values are reduced from %14.7 to %10.16 in data set 2, %17.40 to % 11.6 in data set 12, and %17.1 to %10.24 in data set 13 by using MLP.

However, the time required for the algorithm with K-Means classifier is less than MLP. Therefore, it is necessary to use the overall system, which takes the advantage of both classifiers, to obtain the optimum results.

Moreover, it is clear that, AER is very sensitive to the pixel differences between automatically and manually segmented images. Therefore even 1 or 2 pixel difference between these images increases the error rate significantly especially at the boundaries even when no modification is needed. Therefore, a qualitative evaluation is also made by an expert radiologist. The evaluation of the expert is based on his idea if a slice needs modification or not. Then the AER is calculated for the slices that need modification. Fig. 3.19 shows the average AER calculated for the slices that need modification for each patient data set by using the overall system. The results show that the average AER is reduced significantly to %5.09 with a minimum of %2.4 and a maximum of 7.63.

As a last analysis we calculated volume measurement of the liver for comparing the segmentation performances of the algorithms (K-Means, MLP and overall) and the success rate of the proposed approach is (Fig. 3.20 (a)). To measure the volume of the segmented parts, pixel spacing and slice thickness values are used from DICOM Meta information. The error range of the volumes obtained with the K-Means algorithm is between 20 mm3 to 210 mm3. The percentage errors of these measurements are found between 0.7% and 16.26% with a mean error of %4.7. The same analysis for MLP show that the error range of the volumes obtained with the K-Means algorithm is from 40 mm3 to 160 mm3 and from 2% to 12.8% with a mean of %7.5. The results again show that the MLP shows better performance at the datasets with a typical liver shape (8, 9,

and 15) and low contrast adjacent tissues (2, 12, and 13) although the K-Means is better in overall average error.

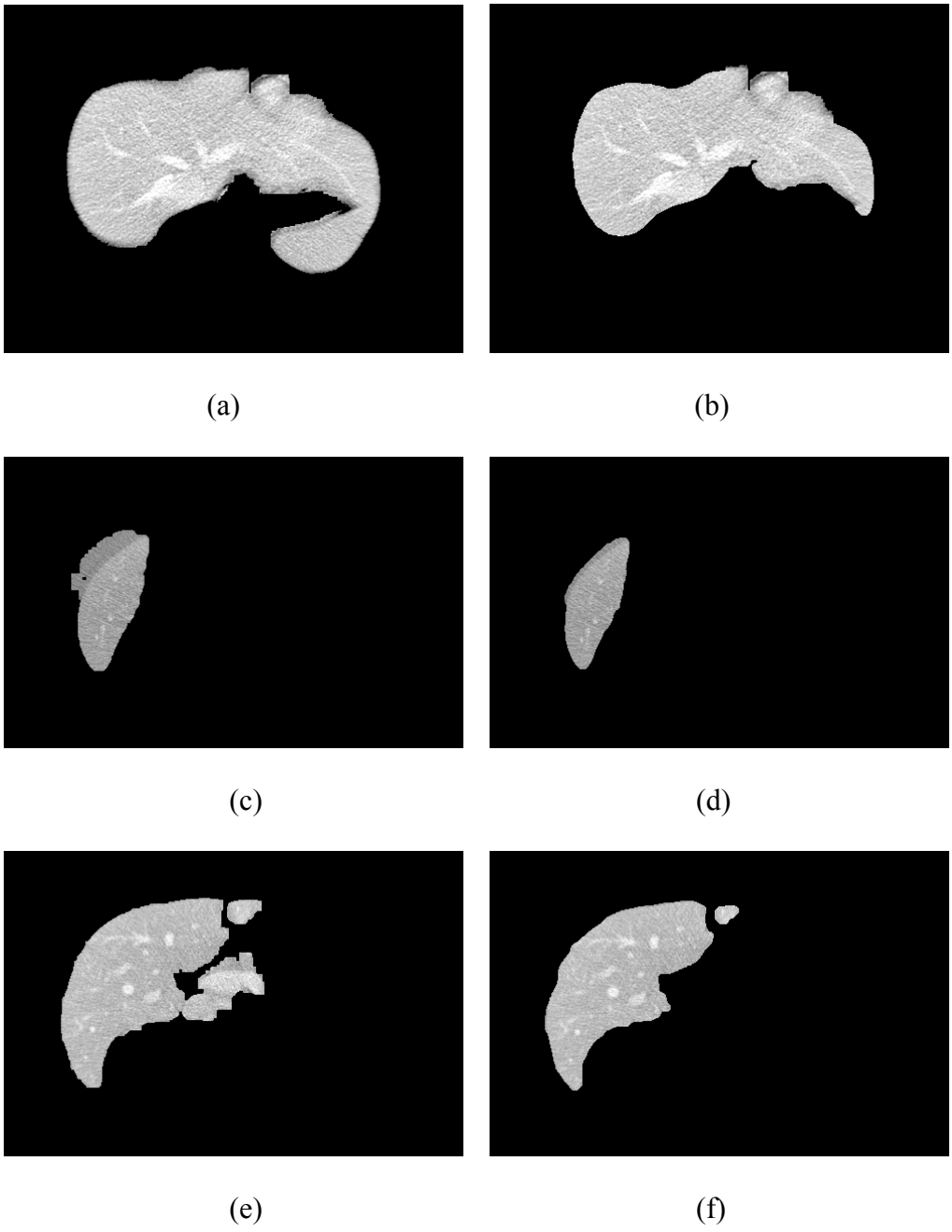


Figure 3.18 a) Segmentation results for Figure 3.1 (a): Algorithm result with K-Means (b) Algorithm result with MLP (c) Segmentation results for Figure 3.1 (d): Algorithm result with K-Means (d) Algorithm result with MLP (e) Segmentation results for Figure 3.1 (c): Algorithm result with K-Means (f) Algorithm result with MLP.

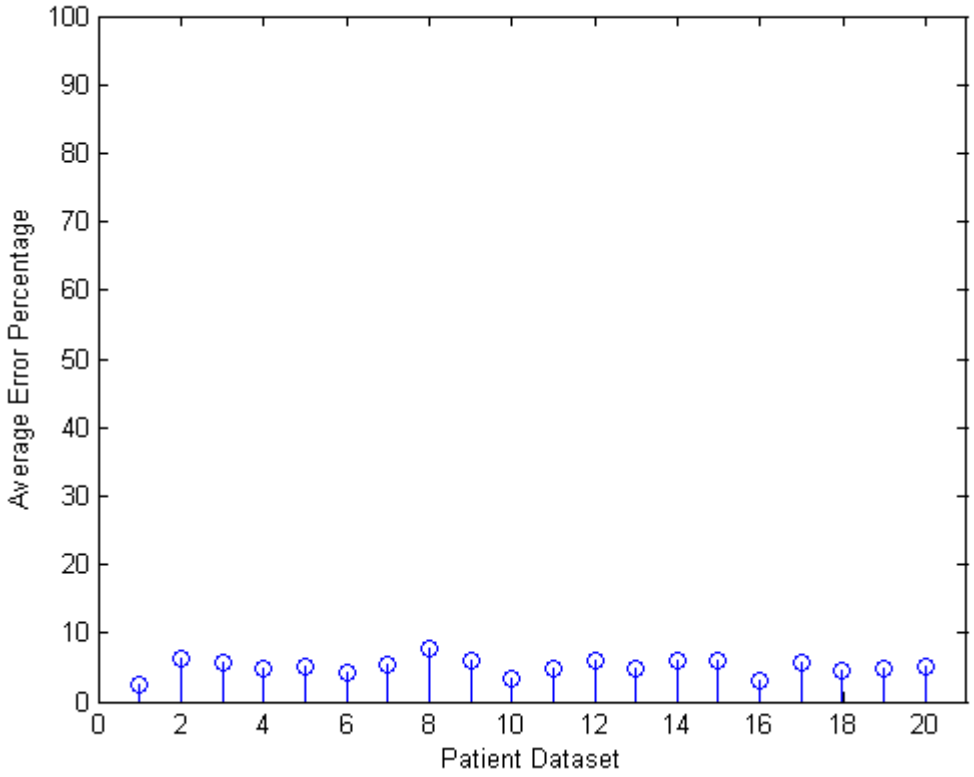
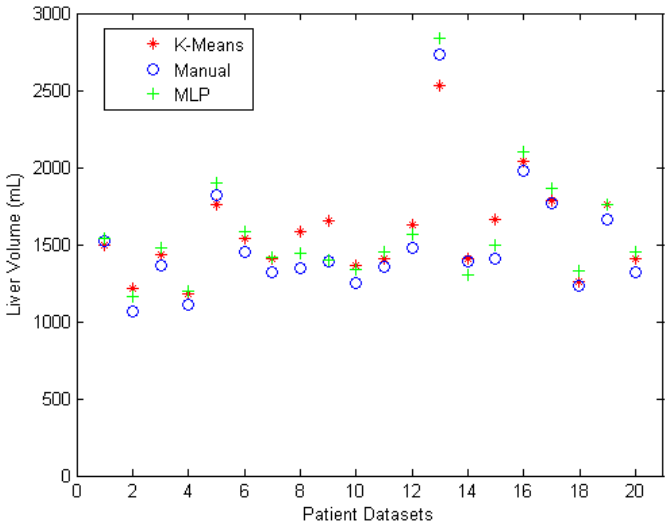


Figure 3.19 AER calculated for the slices that are determined by the expert as further modification is needed.

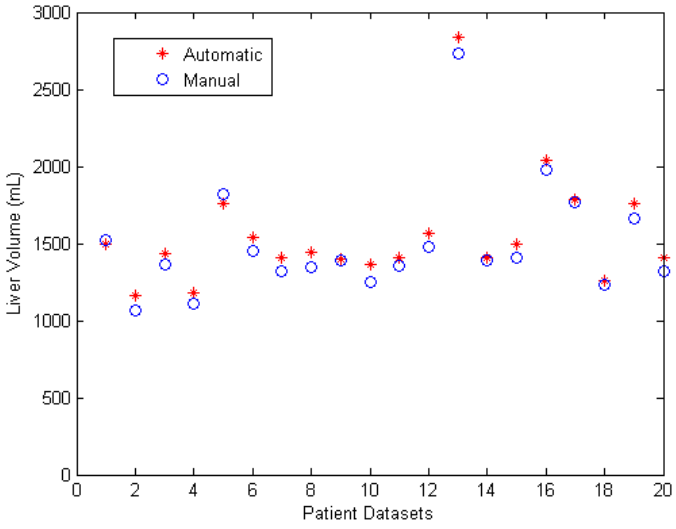
The overall system performance is shown in (Fig. 3.20 (b)). The error range of the volumes obtained with the algorithm is from 20 mm3 to 140 mm3. The percentage errors of these measurements are found between 0.7% and 12.8% with a mean error of %6.4.

The Java version of the program with the K-means algorithm runs for 5 to 7 minutes in a standard PC with 2GB Ram and 3GHz processor and requires 750 MB of memory. The Matlab® version runs more slowly and takes around half an hour with K-means classifier. The algorithm with the MLP classifier ends approximately in 45 minutes both in Matlab® and in Java. On the other hand it takes around 60 to 90 minutes for an experienced user to segment liver from 100 slices manually and it requires user experience both on the liver and the software which should consist of the necessary tools for manual segmentation of the liver. In comparison with manual segmentation tool that is currently in use, the proposed algorithm is clinically feasible and much more efficient

in terms of time and efficiency. Moreover, it produces the output data in a form that is suitable for volume rendering and thus for further analysis in 3-D (Fig. 3.21).

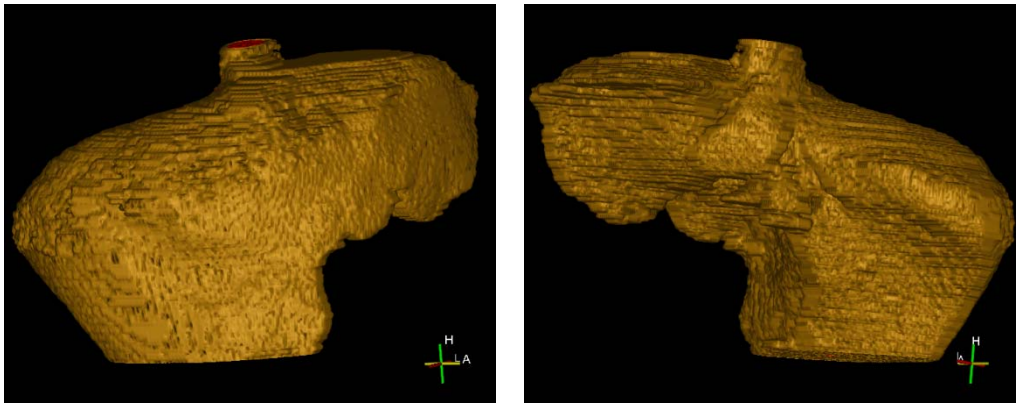


(a)

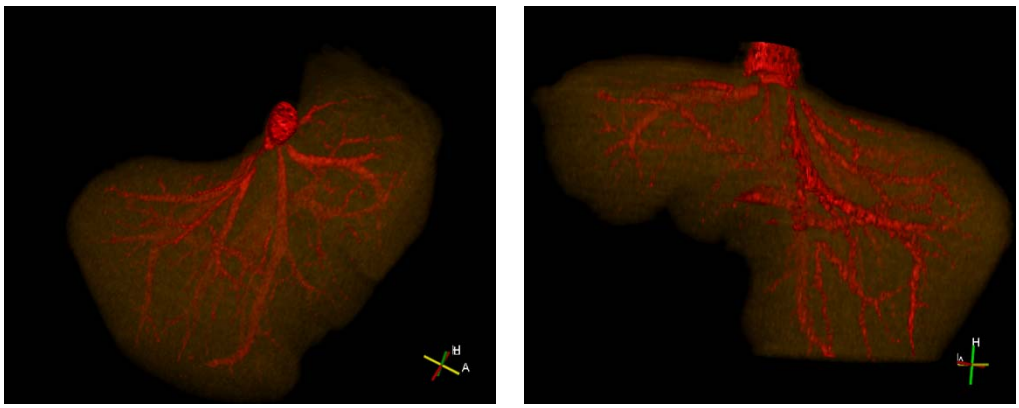


(b)

Figure 3.20 Volume measurement results of segmented parts (a) Comparison of volumes obtained by MLP and K-Means with manually segmented. (b) Overall system performance.



(a) (b)



(c) (d)

Figure 3.21 (a, b) 3-D visualization of segmentation results (c, d) visualization of inner liver using transfer functions in volume rendering (transparent parenchyma in yellow color, opaque vessels in red color)

CHAPTER FOUR
TRANSFER FUNCTION SPECIFICATION FOR ABDOMINAL VISUALIZATION

As being a tool that assigns optical parameters, used in interactive visualization, TFs have very important effects on the quality of volume rendered medical images. Unfortunately, finding accurate TFs is a tedious and time consuming task because of the tradeoff between using extensive search spaces and fulfilling the physician's expectations with interactive data exploration tools and interfaces. Therefore, it is necessary to integrate different features into the TF without losing user interaction. By addressing this problem, a semi-automatic method for initial generation of TFs is introduced. The proposed method uses a fully Self Generating Hierarchical Radial Basis Function Network (SEG-HRBFN) to determine the lobes of a Volume Histogram Stack (VHS) which is introduced as a new domain by aligning the histograms of the image slices of a CT/MR series. The new self generating hierarchical design strategy applied on RBFN allows for recognizing suppressed lobes corresponding to suppressed tissues in VHS and also for representing the overlapping regions which are parts of the lobes but can not be represented by the Gaussian bases associated to the lobes due to the overlapping. Approximation with a minimum set of basis functions using SEG-HRBFN provides the possibility of selecting and adjusting suitable units to optimize the TF. The proposed method allows the integration of spatial knowledge, local distribution of the tissues and their intensity information into the TF while preserving the user control. Its applications on different CT and MR data sets show enhanced rendering quality in abdominal studies which are presented in Chapter 5.

4.1 Introduction and Related Work

The medical visualization, which aims to produce clear and informative pictures of the important structures in a data set, requires extensive user interaction. One of the important advantages of volume rendering (Drebin, Carpenter & Hanrahan, 1988) is that combinations of selected parameters, such as opacity and color, can be determined during the rendering pipeline. During the generation of volume rendered medical

images, TF specification is the step where these adjustments can be done. Therefore, it is crucial and important to design accurate TFs to produce meaningful and intelligible 3-D images. However TF design is a very difficult task because of the availability of various possibilities. Since this flexibility can not be kept in strict bounds, specification of an appropriate TF is a challenging problem especially when there is no initial TF design prior to the optimization process.

To overcome this difficulty generally a number of predefined TF presets are used as starting point (so called initial TF design). The main idea behind this approach is that certain types of volume data are standardized in the range of data values and special sub-ranges are assigned to the same type of tissue (Thus, predefined TFs are adjusted due to these ranges). However, depending on different modality settings, injection of a contrast media or environmental circumstances, these sub-ranges may vary significantly. For these reasons, a limited number of TF presets can not be enough to cover all possible cases and to provide useful initial TFs. In order to create a useful initial TF that provides a good basis prior to optimization, an automatic tissue detection method that finds the intensity range for each tissue of interest is needed.

As clearly stated in (Lundström, Ljung, & Ynnerman, 2006), the usual TF design procedure for a user, i.e. physician, is tissue based: The user first defines the tissues, determines their locations and then assigns visual properties (i.e. opacity, color) to them. However, this design procedure is a very time consuming, tedious and operator dependent task. The first difficulty is to detect tissues in an unknown data set which is a problem far from being trivial. Another major problem is to separate tissues with overlapping intensity distributions.

Current approaches for TF specification can be divided into three groups as: manual, data centric and image centric. The manual approach addresses the need for expert intervention alone in generating the final image (Pfister, 2000) while the data-centric approaches are based on measuring the data set properties. Bajaj et al. (Bajaj, Pascucci, & Schikore, 1997) have used iso-value determination to find the contours that are hidden behind another. Kindlmann et al. (Kindlmann & Durkin, 1998) have introduced multi-dimensional TFs and have used edge detection concept from computer vision.

Other data centric techniques use topology analysis (Fujishiro, Azuma, & Takeshima, 1998), stochastic properties of datasets (He, Hong, Kaufman, & Pfister, 1996), and multidimensional data analysis (Kniss, Kindlmann, & Hansen, 2002). The image-centric approaches, on the other hand, are based on evaluating TFs on the basis of images they produce where the physician can select one of these rendered images (Shiaofen, Tom, & Mihran, 1998, Kniss, Kindlmann, & Hansen 2001, Konig, & Gröller, 2001, Marks, Andalman, Beardsley, & Pfister, 1997).

Recently, local properties are used in some successful applications for TF generation. In (Lundström, Ljung, & Ynnerman, 2006), histogram contents for local neighborhoods are used to detect and separate tissues with similar intensities. In (Lundström, Ynnerman, Ljung, Persson, & Knutsson, 2006), an enhancement that amplifies ranges corresponding to spatially coherent materials by using alpha-histograms, which are individually retrieved by dividing the data set into local regions, is implemented. In (Roettger, Bauer, & Stamminger, 2005), spatialized transfer functions are introduced as one- or more-dimensional transfer functions, where spatial information has been used to derive the color, whereas statistical (and/or spatial information) is used to setup the opacity. These studies show the importance of local information in solving major problems in TF generation such as the classification of overlapping tissues. In (Sato et al., 2000), 3-D filters, based on gradient vector and Hessian matrix, are used to enhance specific 3-D local intensity structures. Filter banks are utilized for considering local texture and other characteristics in (Lum, Shearer, & Ma, 2006).

Machine learning is introduced for transfer function specification in (Tzeng, Lum, & Ma, 2005). In that system, the physician works in the volume data space by directly painting on sample slices where the painted voxels are used in an iterative training process and the trained system can then classify the entire volume. All process is hardware accelerated, thus providing immediate visual feedback. A fuzzy classification based system is used by (Kniss et al. 2005) which provide the user access to the quantitative information computed during fuzzy segmentation. The decision making step of classification is deferred until render time, allowing the user finer control of the importance of each class. To avoid the artifacts of binary classification, semantic values

are defined (Rautek, Bruckner, & Groller, 2007) using fuzzy sets, enabling a linguistic specification of renderings. A high level semantic model with a simple user interface to improve the usability of direct volume rendering applications is introduced in (Salama, Keller, & Kohlmann, 2006).

For time-varying medical data, several methods are pro-posed and compared in (Fang, Möller, Hamarneh, & Celler, 2007) while an interface is de-signed based on cascading the histograms of all time steps in (Akiba, Fout, & Ma 2006).

All of these techniques produce informative images and offer more degrees of freedom; however generated TF models are also more difficult to setup by the user. Since data exploration is an essential element of creating a TF to fulfill the physicians' expectations, this reduction should be done effectively. Therefore, a physician interaction mechanism that is informative, easily understandable and highly interactive should be provided after the initialization of a TF using a (semi) automatic method.

Using the similarity of the lobes to the Gaussian functions in the volume histogram; a 2-step TF specification method for 1-D volume histogram is introduced in (Selver, Fischer, Kuntalp, & Hillen, 2007). That method is capable of determining the hidden lobes by removing the main lobes which are already recognized in the first step.

In this paper, a semi-automatic method is developed to shorten the design process by creating an initial TF for the tissues of interest. This initialization is realized with a fully Self Generating Hierarchical Radial Basis Function Network (SEG-HRBFN) that approximates to a Volume Histogram Stack (VHS). A VHS is a two dimensional (2-D) histogram which is constructed by aligning the histograms of each slice in a data set (Fig. 4.1). Thanks to its properties, VHS incorporates spatial domain knowledge with local distributions of the tissues and their intensities.

The TF specification is then, posed as a two-stage procedure: The lobes corresponding to different tissues in VHS data is characterized by a Gaussian based function approximation (or say Gaussian expansion) in the first stage. Then, in the second stage a physician exploits Gaussian spectrum, i.e. the centers and widths of deter-mined Gaussian basis, to assign different opacity and color for each point in the 2-

D input space of Hounsfield value and slice number. So, the overall TF specification is considered as a (vector-valued) function approximation problem where the domain is the 2-D input space of Hounsfield value and slice number and the range variables are opacity and color. Thus, the method presented in the paper changes the search space from 1-D to 2-D (i.e. from 1-D gray/Hounsfield value domain to 2-D gray/Hounsfield value and spatial location domain, i.e. the slice number) which contains more priori information and domain knowledge for representation of the tissues.

The SEG-HRBFN, which is designed by a hierarchical learning strategy, is employed to approximate to 2-D VHS. The SEG-HRBFN provides a multi-step procedure; i) for capturing all suppressed lobes of importance in a successive manner by associating the lobes with the Gaussian bases and also ii) for representing the overlapping regions, which can not be represented by the Gaussian bases associated to the lobes, by assigning additional Gaussians for them in the residual VHS, that is the remainder after removing the already obtained approximation from the original VHS. SEG-HRBFN allows an approximation with a minimum set of basis function that can be further adjusted by the physician to optimize the TF in an interactive way. Due to the requirement of small number of neurons, the construction of SEG-HRBFN is done in quasi real time.

At each layer, the SEG-HRBFN contains a number of Gaussian units that are used to approximate to the lobes in VHS. The number of these units and their structural parameters (i.e. centers, widths) and linear weights are determined automatically. Finally, a combination of the user selected units is used to create an accurate initial TF. The application of the proposed method to several medical datasets shows its effectiveness, especially in visualization of abdominal organs. The proposed method allows the integration of spatial knowledge, local distribution of the tissues and their intensity information into the TF while providing an effective and user friendly interaction mechanism which preserves the possibility of using it in clinical applications.

According to the organization of this chapter, the VHS data and its properties are explained in Section 4.2. Approximation with SEG-HRBFN is established in Section 4.3 and its comparison with RBFN and HRBFN algorithms are presented in Section 4.4.

Representation of the results of initial TF generation algorithm is given in Section 4.5. The application of the proposed method to several medical datasets and comparison of the results of the proposed method with different techniques are presented in the next Chapter.

4.2 Volume Histogram Stack (VHS) Data

In CT datasets, the same type of tissues may have different gray value distributions and may appear at different locations in a histogram depending on the patient, environmental circumstances, injection of a contrast media, and certain modality parameters although there is a calibrated intensity scale, i.e. Hounsfield Values (HV). Moreover, volume rendering is not commonly used for MR data sets since there is no calibrated intensity scale. In conventional approach, both for CT and MR data sets, the volume histogram is the main guide to find the tissues of interest. However, the tissues do not always correspond to visible peaks since they might be suppressed by dominant peaks of unimportant tissues. Especially in abdominal CT/MR datasets, soft tissues and organs (i.e. liver, kidney, spleen, aorta, muscle tissue etc) exist in a very narrow range of HV/gray values. This overlap hardens the usage of transfer functions in the visualization of abdominal organs, hence complex and time-consuming segmentation methods are needed to be employed prior to the visualization.

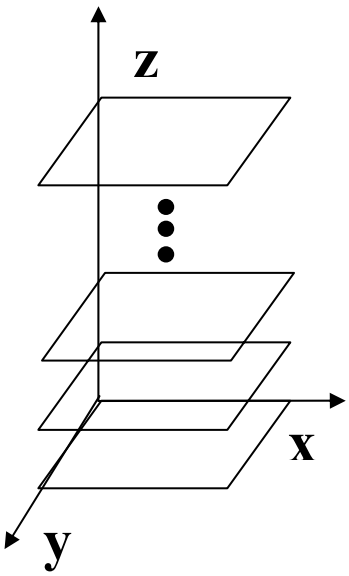
One of the attractive advantages of the medical datasets is that some of the tissues/organs have small size in the initial slices where they begin to appear, slightly expand in the successive ones and finally disappear. This causes a lobe like histogram distribution for a tissue/organ which has usually a shape similar to a radially asymmetric (elliptical) Gaussian. However, the effect of having a lobe like distribution is visible for most studies and cases but not for all. One of the frequently used CT/MR studies showing this characteristic is the abdominal study. The organs of interest in these datasets (i.e. liver, kidneys, spleen etc.) are quite large so that they dominate a number of slices. Therefore, it is possible to get additional and important information by using the z-dimension (orthogonal to the slices) as exploited in the 2-D VHS introduced by this paper. Organs that are spatially separated in the z-dimension get separate lobes whereas they have intersecting lobes if they are spatially non-separated.

This approach can also be extended to x- and y- dimensions. For example, if the major slicing axis is z-dimension, the histograms can be generated for axial images (x-y dimensions) and aligned through z-dimension to construct VHS. Furthermore, VHS can also be generated by using the histograms of coronal images (y-z dimensions) and aligning them through x-dimension or the histograms of sagittal images (x-z dimensions) aligning them through y-dimension. Thus, the VHS can be generated based on the organ to be visualized and independent of slicing axis without an additional scanning procedure. With this opportunity, VHS can distinguish structures which are separated in any of x-, y- and z-dimensions. Traditional volume histograms can not take this advantage as they represent cumulative gray level distribution over all data set. Therefore, a data format is necessary to represent the spatial range of the tissues, their local properties as well as gray level distributions.

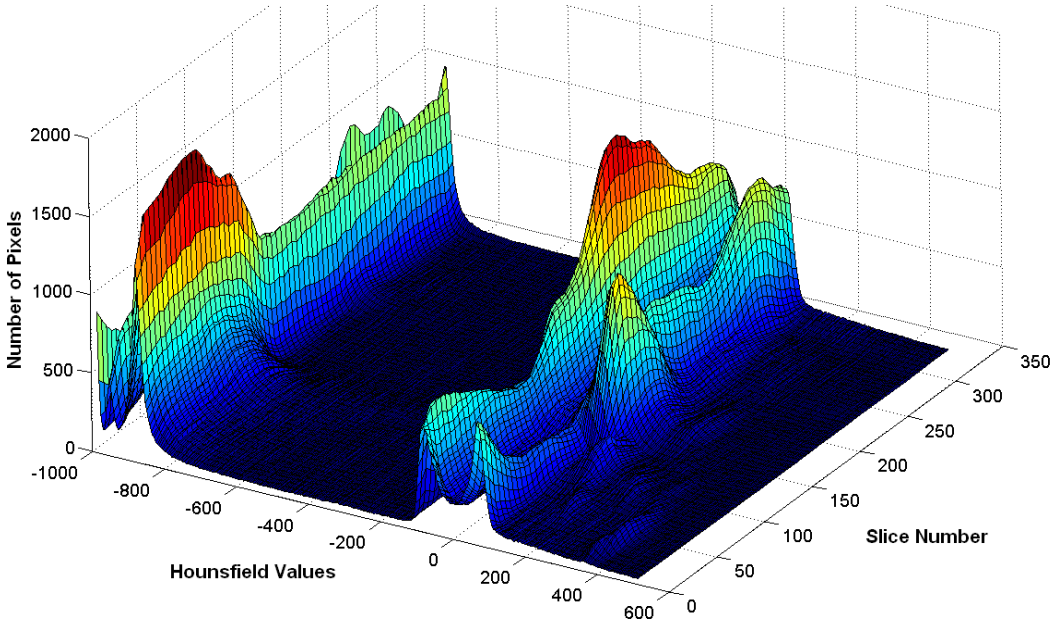
For this purpose, the newly proposed data model, namely VHS, is generated by aligning histograms of all slices (images) (Fig. 4.1 (a)) in a data set (Fig. 4.1 (b) and Fig. 4.2 (b)). The VHS data exploits more priori information as saving inter-slice spatial domain knowledge since each slice histogram is represented separately. It demonstrates changes in the gray values through the series of slices, thus includes information on local histogram distributions of tissues. For example, when a tissue appears larger in an image, the number of pixels representing this tissue also increases and vice versa. The VHS demonstrates these changes much better than the volume histogram since the data distribution is shown in a continuous way through the series. Thus, it can represent the intensity values of the tissues as well as their spatial information and local distributions which are not available in conventional volume histograms. The tissues which are at different slices but with similar gray level distributions can clearly be distinguished by using this spatial information.

When two or more tissues have overlapping gray values and share a set of slices, where some parts of them appear, they construct suppressed lobes and overlapping regions (intersecting lobes). The suppressed lobes are defined as the hardly recognizable minor lobes due to the domination by the major ones. The overlapping regions indeed do not correspond to the lobes but they are parts of these lobes and can not be

represented by the Gaussians associated to these lobes as a consequence of overlapping (Fig. 4.2 (a) and Fig. 4.6).

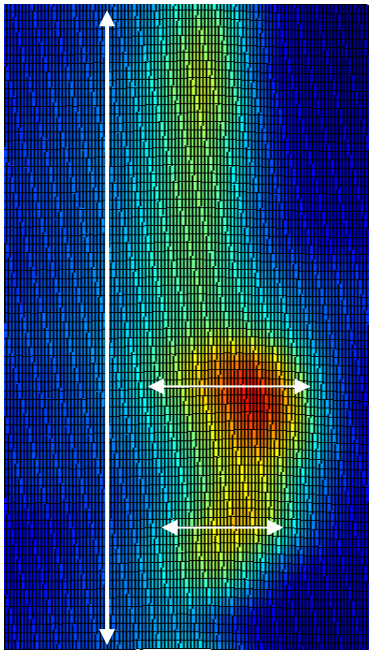


(a)

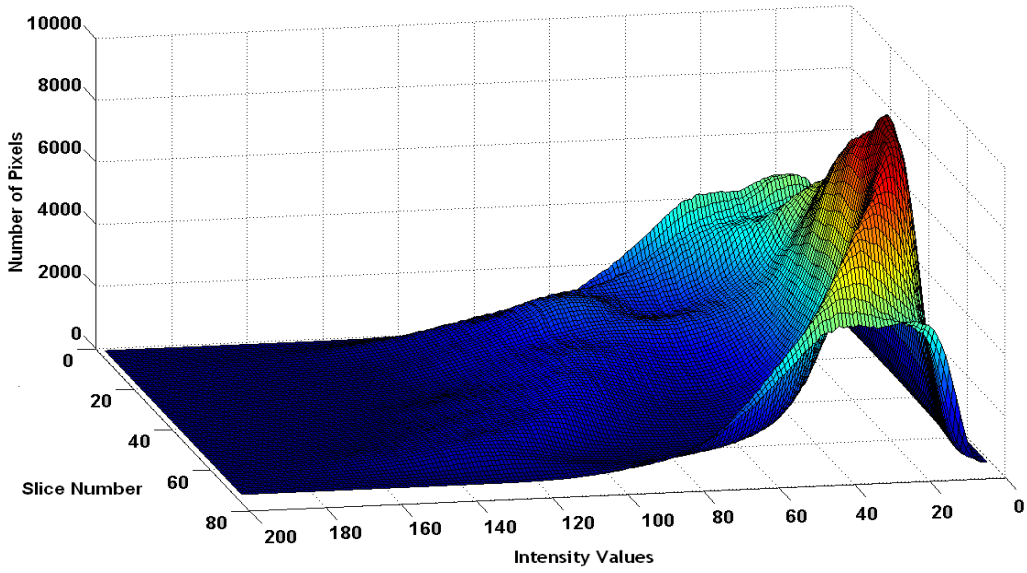


(b)

Figure 4.1 (a) Planes used in this paper for constructing VHS which are orthogonal to a particular axis (z in this figure) (b) Volume Histogram Stack (VHS) data for a CT Angiography data set of 326 slices.



(a)



(b)

Figure 4.2 (a) 0-100 range of VHS data obtained from the PANORAMIX data set and plotted from the top view. Muscle tissue that goes through the whole volume with almost the same amount of pixels in each slice is shown by the vertical line. Two tissues that overlap with this tissue but have wider range of Hounsfield values, which are shown by the horizontal lines, can be recognized by SEG-HRBFN using local distribution information obtained from the VHS data. (b) VHS data obtained from the BEAUFIX data set. (Figure is mirrored on "Intensity Value" axis for better illustration of the lobes.)

Even in such cases, the proposed method can detect suppressed lobes and overlapping regions and an efficient classification can be achieved by assigning a Gaussian basis function of SEG-HRBFN to each of these partially intersecting lobes where the dominant lobes are first approximated by the associated Gaussians and then the minor lobes which remain in the residual VHS, i.e. the difference between the former approximation and the original, are approximated by newly assigned Gaussians. As explained in the previous paragraphs, VHS can be generated not only for the slicing axis but also for an arbitrary axis (i.e. x-, y- or z- dimensions) depending on the organ to be visualized.

To design an effective initial TF by using VHS in the shortest amount of time, an automatic and lobe detection based TF design method is needed. SEG-HRBFN introduced in this paper provides such a method.

4.3 Self Generating Hierarchical Radial Basis Function Network

Among all artificial neural networks, RBFN seems to be a good choice for approximating VHS data due to the almost perfect matching of the Gaussian basis to the lobes. Choosing Gaussian basis for representing each lobe not only provides a good approximation but also, indeed more importantly, gives the possibility of optimizing the TF by the physician via adjusting the Gaussian parameters in an interactive way. During simulations, it is observed that determining the Gaussian parameters (i.e. centers, widths and heights) in an appropriate way is quite important because capturing all suppressed lobes of importance and representing the overlapping regions are critical in the quality of rendering result.

The basic strategy of using fixed centers and/or the same width for all units in RBFN design can not guarantee the determination of suppressed lobes and overlapping, and therefore the proper determination of Gaussian centers and a different width for each Gaussian unit is necessary. Even if an appropriate number for hidden neurons is chosen together with appropriate center locations and widths, simulations show that the RBFN approximation (Section 4.3.1) tends to fit only major lobes and skip the suppressed information carried by the minor lobes. To overcome this drawback, the HRBFN (Ha, 1998, Cerveri, Forlani, Borghese, & Ferrigno, 2002, Ferrari, Frosio, Piuri, & Borghese,

2005), would be an appropriate choice since it assigns Gaussians in all scales to represent all details of the function to be approximated. However, simulations show that (Section 4.3.2) the original HRBFN produces a huge number of hidden neurons when it is applied to approximate VHS. This is not a desired property for TF design since the physician should deal with a small number of units to obtain an efficient interaction mechanism.

Considering the above reasons, HRBFN is used as the network for approximating to VHS data but with a new learning strategy, called as self generating hierarchical learning strategy. The developed SEG-HRBFN (Section 4.3.3) provides a procedure for capturing all suppressed lobes of importance in a successive manner by associating the lobes with a minimum number of Gaussian bases.

4.3.1 Radial Basis Function Networks (RBFN)

As shown in Fig. 4.3, single-output RBFN consists of three different types of neurons: Input neurons which are used just for feeding the input data to the hidden neurons, nonlinear neurons having a Gaussian transfer function at the hidden layer and a linear neuron performing a weighted sum at the output layer. Such an RBFN defines the following function:

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} w_j . g(\boldsymbol{x} - \boldsymbol{c}_j ; \boldsymbol{\sigma}_j) \tag{4.1}$$

Where $\{\boldsymbol{c}_j \in R^D\}$ corresponds to the center of j[th] Gaussian unit, $\{\boldsymbol{\sigma}_j \in R\}$ to the width, and $\{w_j \in R\}$ to the j[th] linear weight.

4.3.2 Hierarchical Radial Basis Function Networks

As observed in the simulations done at the beginning phase of the research of which the results are presented in this paper, the back-propagation method and also the hybrid methods suffer from the following drawback in approximating to VHS data: Even if an appropriate number for hidden neurons is chosen, the RBFN approximation tends to fit only major lobes skipping the suppressed information carried by the minor lobes which

also correspond to the whole or some part of the other tissues desired to be rendered. To overcome this drawback, one can exploit the idea of (Selver, Fischer, Kuntalp, & Hillen, 2007) where the minor lobes are determined after removing the main lobes which are already recognized in a first step. In this sense, the Hierarchical Radial Basis Function Network (HRBFN) (Ha, 1998, Cerveri, Forlani, Borghese, & Ferrigno, 2002, Ferrari, Frosio, Piuri, & Borghese, 2005), which is originally proposed for noise elimination in face reconstruction, would be an appropriate choice since it assigns Gaussians in all scales to represent all details of the function to be approximated. It constructs the approximation layer by layer in a hierarchical way beginning with the coarsest information (i.e. low frequency components which correspond to major lobes in VHS) and continues with the more detailed information (i.e. high frequency components which correspond to minor lobes in VHS).

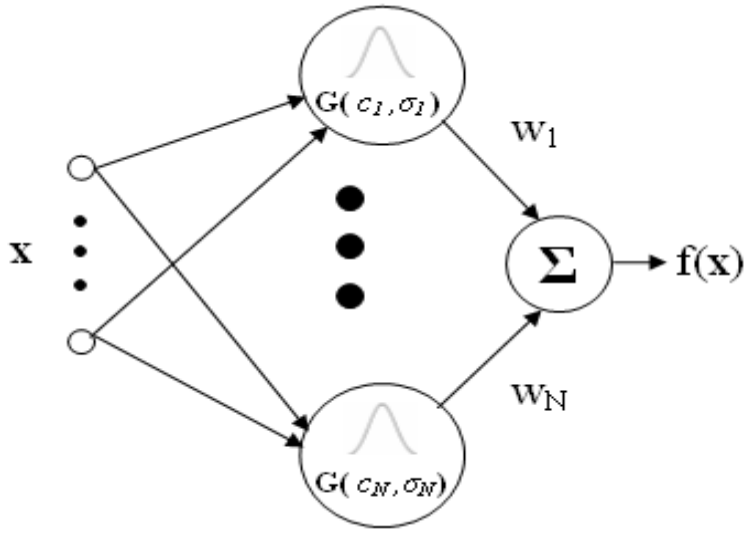


Figure 4.3 Multi-input single-output RBFN with N hidden neurons

4.3.3 Implementation of Self-Generating Hierarchical Radial Basis Function Networks

The original HRBFN produces a huge number of hidden neurons when it is applied to approximate VHS. This is not a desired property for TF design since the physician has to deal with small number of units to obtain an efficient interaction mechanism. Considering all of the cited reasons, HRBFN is used as the network for approximating to VHS data but a new learning strategy, called as self generating hierarchical learning

strategy, is developed and used for the design of RBFN in this paper. The developed Self Generating HRBFN (SEG-HRBFN) provides a multi-step procedure for capturing all suppressed lobes of importance in a successive manner by associating the lobes with a minimum number of Gaussian bases. The HRBFN and the newly developed self generating hierarchical learning strategy are explained in the sequel.

SEG-HRBFN performs a mapping *f(·)*: $R^D \rightarrow R$, as the sum of K approximations $\{l_i(\cdot)\}_{i=1,2,...,K}$:

$$f(\boldsymbol{x}) = \sum_{i=1}^{K} l_i(\boldsymbol{x}) \tag{4.2}$$

Herein, approximation layers, $l_i(\cdot)$'s, are RBF sub-networks; they are indeed not structural layers but just functional layers, i.e. approximation layers, constructed in a successive manner along the training phase by self generating hierarchical learning strategy. Hence, each $l_i(\cdot)$ is a linear combination of Gaussian units. The j^{th} Gaussian unit at i^{th} layer is defined by, $g_{ij}(\cdot) = \exp\left(-(\boldsymbol{x} - \boldsymbol{c}_{i,j})^T . \left(1/\boldsymbol{\sigma}_{i,j}^{\,2}\right) . (\boldsymbol{x} - \boldsymbol{c}_{i,j})\right)$ which is capable of producing radially asymmetric Gaussians.

Each $l_i(\cdot)$ is composed of M_i Gaussian units found,

$$l_i(\boldsymbol{x}) = \sum_{j=1}^{M_i} w_{i,j} . g(\boldsymbol{x} - \boldsymbol{c}_{i,j}; \boldsymbol{\sigma}_{i,j}) \tag{4.3}$$

So the network has totally $M = \sum_{i=1}^{K} M_i$ units.

The complete output of the HRBFN is the combination of all approximation layers. At each layer, SEG-HRBFN searches the required number of Gaussian units and calculates all necessary parameters in an automatic way. Starting from the VHS data *f(•)* (Fig. 4.4 (a)), first a peak detection algorithm is used to determine M_i and the locations of the centers for each layer as explained in Section 4.3.4. Then, for each Gaussian unit, a width is calculated by considering the topology of the other units (Section 4.3.5). Finally, the linear weights are calculated (Section 4.3.6).

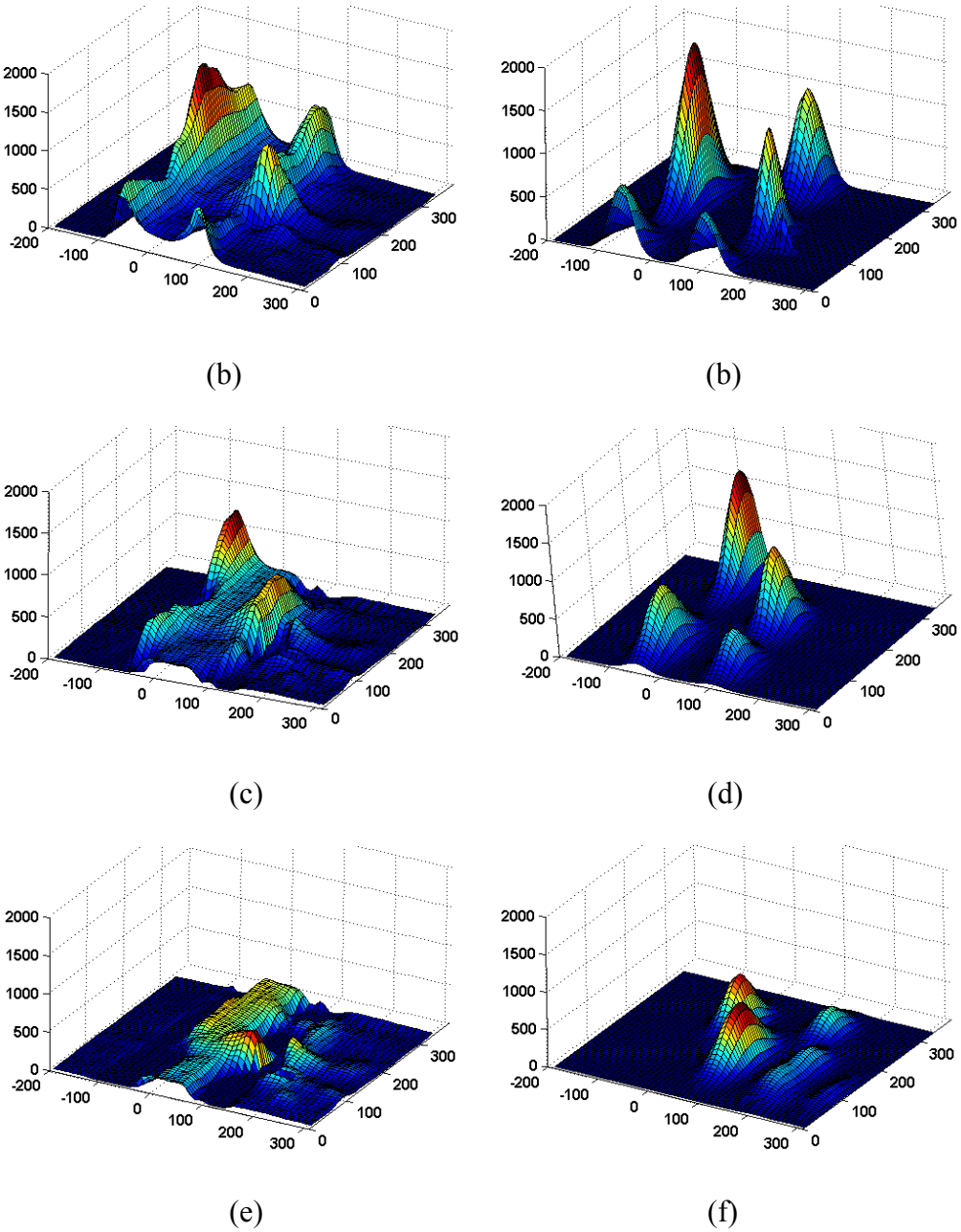


Figure 4.4 Finding the lobes of VHS data: (a) Original VHS data in Figure 2 is reduced to range (-200)-(+300) HV for illustration purposes. (To be approximated by the first layer.) (b) Lobes found for the first layer. (c) Residual VHS data obtained from the difference of (a) and (b). (To be approximated by the second layer.) (d) Lobes found for the second layer. (e) Residual VHS data obtained from the difference of (c) and (d). (To be approximated by the third layer.) (f) Lobes found for the third layer. (Illustration Notes: 1)Very small lobes found by the SO-HRBN are left out for clarity 2) For best representation of the 3-D nature of the VHS data, all illustrations are prepared with the same camera position, [x y z] = [186.0 -361.72 11380.64] by using Matlab® 7.0 R14)

After reconstructing the surface by using the Gaussian units found for a layer (Fig. 4.4 (b)), a residual is calculated point-wise, i.e. for each point x^n used in the training (Fig. 4.4 (c) for an illustration of the process.):

$$r_1(\boldsymbol{x}^n) = f(\boldsymbol{x}^n) - l_1(\boldsymbol{x}^n) \tag{4.4}$$

The next approximation layer considers the residual VHS found in the previous layer as the new function to be approximated (Fig. 4.4 (d)) and the number of Gaussians and their parameters are calculated now for that residual (Fig. 4.4 (e)-(f)). The general expression of a residual VHS to be approximated at ith layer is given as:

$$r_{i-1}(\boldsymbol{x}^n) = f(\boldsymbol{x}^n) - \sum_{j=1}^{i-1} l_j(\boldsymbol{x}^n) = r_{i-2}(\boldsymbol{x}^n) - l_{i-2}(\boldsymbol{x}^n) \tag{4.5}$$

To approximate VHS data, this procedure continues for four layers by default since it is observed in the simulations that four layers are found sufficient in most cases (Fig. 4.5). Of course, it can be further iterated if the generated Gaussian units are not found enough to construct an appropriate TF by the user and until satisfactory number of units are obtained.

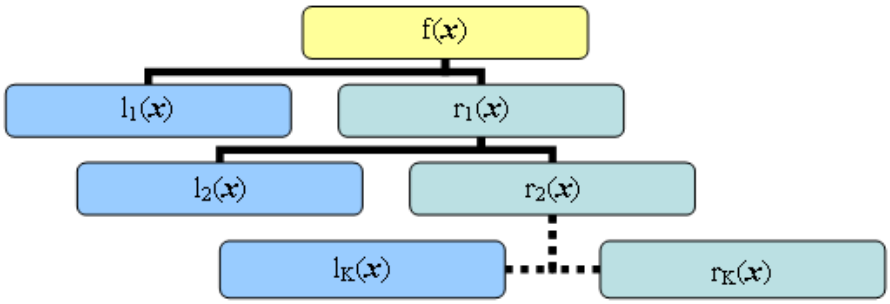


Figure 4.5 HRBFN construction chart

4.3.4 Determination of M_i and Centers of a Layer

As cited before, a common method to determine the centers of RBF networks is using a clustering algorithm (Hartigan, 1975) (i.e. K-Means (Duda, & Hart, 1973)) which is applied only to input training sample vectors. However, in our case, such an IC method produces incorrect results since the VHS data is uniformly sampled at all

dimensions. To overcome this difficulty an IOC method (Uykan, Güzeliş, Çelebi, & Koivo, 2000) or a method for non-uniform re-sampling of the data prior to clustering process can be used, however such methods have high computational complexity and can prevent the realization of the network in a short time. Since then, a peak detection algorithm is used for the determination of Mi and the centers. In this manner, first, the VHS data is smoothed with a 7x7 average filter to remove high frequency components and then its gradient is calculated in both directions of x, say x_1 and x_2. The size of 7x7 for the filter is found to be optimum after extensive experimentation with different scales.

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial}{\partial x_1} f(\boldsymbol{x}), \frac{\partial}{\partial x_2} f(\boldsymbol{x}) \right) \tag{4.6}$$

If a data point at the gradient of the *f(•)* is a crossing point (from positive to negative) for both directions, then that point is selected as a peak. Finally, the projection of that data point to x (x_1-x_2 coordinates) determines the center location for the corresponding lobe. (The peak detection procedure may also be realized by any other peak detection method. The above method is found to be sufficient in the simulations.)

4.3.5 Determination of the Widths of the Gaussian Units

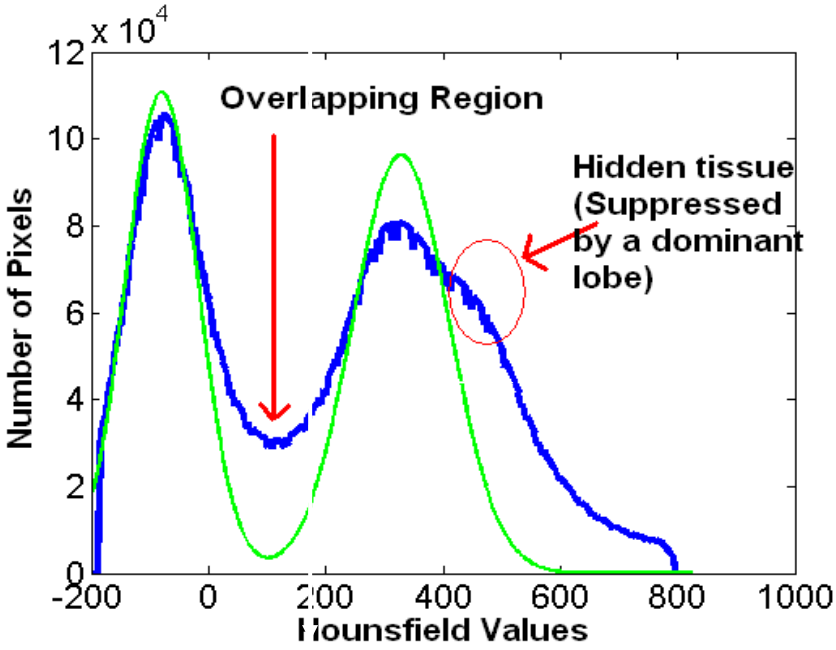
In medical visualization, one of the hardest problems is to distinguish the overlapping tissues, and therefore representing an overlapping region (between two intersecting lobes of VHS) in a clear way is very important. Identifying minor lobes are also highly important in the representation of suppressed tissues as cited. To handle both of these challenges effectively, determining the widths of the Gaussian units in an appropriate way is as important as correct determination of unit locations. Although the use of the same width for all units has been proved to be sufficient to obtain a universal approximator (Park, & Sandberg, 1991), it can produce inefficient results due to the insufficient number of neurons or else learning difficulties in TF specification problem since the distribution of a tissue (lobe) is discarded. Since assigning a different width for each Gaussian improves the model performance (Musavi, Ahmed, Chan, Faris, & Hummels, 1992, Rojas et al., 2000), the width values for each Gaussian have been

calculated using a modified version of the closest RBF heuristic (Park, & Sandberg, 1991) as explained in the next paragraph.

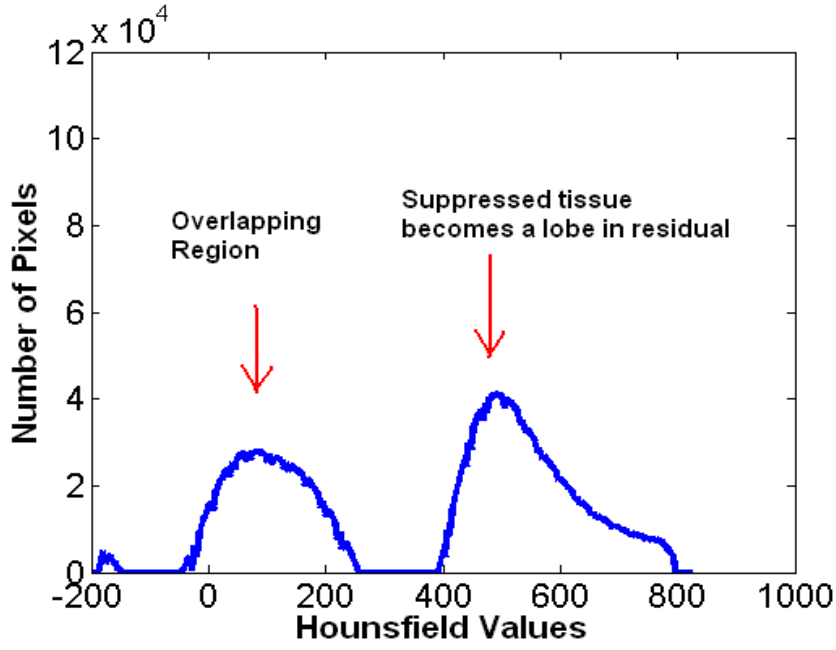
An overlapping region generally occurs between the closest lobes, so it can be represented by a Gaussian unit at a succeeding layer coming after the one where these lobes are already approximated by the SEG-HRBFN (Fig. 4.6). To leave the overlapping region to be approximated in next layer, the width of a Gaussian unit is determined as the half of the distance between its center and the closest center(s) to it. Since the Gaussians used in SEG-HRBFN are not restricted to be radially symmetric, the width search is capable of finding different values for dimensions x_1 and x_2. One constraint to be considered in the selection of widths is: If a point $x^{i,j}$ at which VHS is zero or almost zero is closer to the considered center than the closest center to that center, then the distance to that data point is chosen as the width. After the selection of the 'initial simple task for the user.

$$\sigma_{i,j} = \min\left\{\begin{array}{l}\left(\min\left(\left\|c_{i,k} - c_{i,l}\right\|_2\right)/2, \forall i,k,l\right), \\ \left(\min\left(\left\|c_{i,k} - x^{i,j}\right\|_2\right)/2, \forall i,k,j \ \textit{where}\ f(x^{i,j}) = 0\right)\end{array}\right\} \qquad (4.7)$$

This constraint is used to prevent the construction of wide Gaussian units that can possibly suppress the detection of the minor lobes. Extensive experimentation on different types of datasets has shown that, similar to overlapping regions, this method is useful in representing suppressed lobes (Fig. 4.6) such that these suppressed lobes are fitted by Gaussian units after the approximation to the major lobes.



(a)



(b)

Figure 4.6 Representation of the overlapping region and suppressed lobes using SEG-HRBFN: (a) two lobes are determined at the first layer and SEG-HRBFN generates two units for these lobes, (b) at the second layer (residual signal), the overlapping region and suppressed tissue appear as new lobes which will be represented by new Gaussian units at that layer (A 2-D representation is used instead of 3-D for illustration purposes)

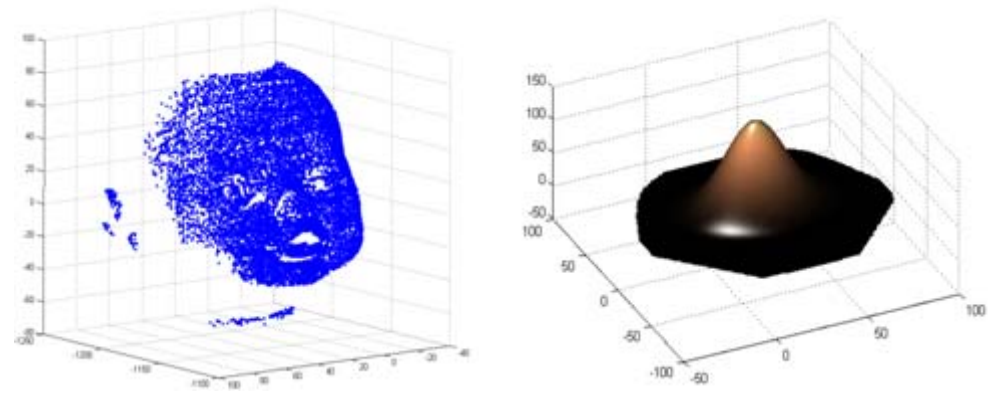
4.3.6 Determination of the Linear Weights

After determining all the parameters concerning the Gaussian units, the network becomes a linear model and the linear weights can then be calculated by using pseudo-inverse methods, e.g. singular value decomposition (Chen, Billings, & Luo, 1989) and orthogonal least squares (Kanjilal, & Banerjee, 1995), or the Least Mean Squares (LMS) rule or any other gradient descent method (Haykin, 1999). LMS rule is chosen in the paper because of its speed in convergence process and relatively less computational complexity.

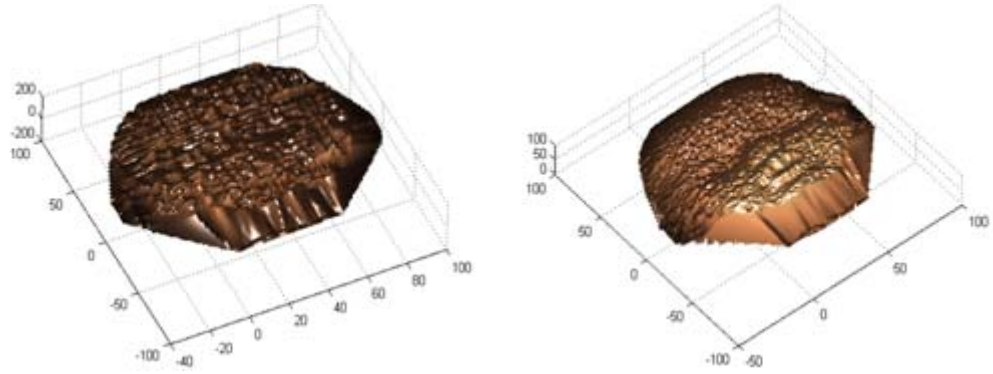
Here, it is worth to point that, the height of the Gaussian units is a minor concern since not the height but the position and width of a unit are used for TF construction. (Note that the heights of Gaussian units in an approximation layer are actually needed to be calculated for the determination of the optimal values of the centers and widths of the units in the succeeding layer.) In other words, the main concern is to find the lobes of *f(•)* data rather than reconstructing them exactly. Therefore, negative parts of residual signals are set to zero to prevent the production of negative lobes resulting in confusion and to prevent the complexity at the interactive optimization level.

4.4 Comparison of SEG-HRBFN with HRBFN

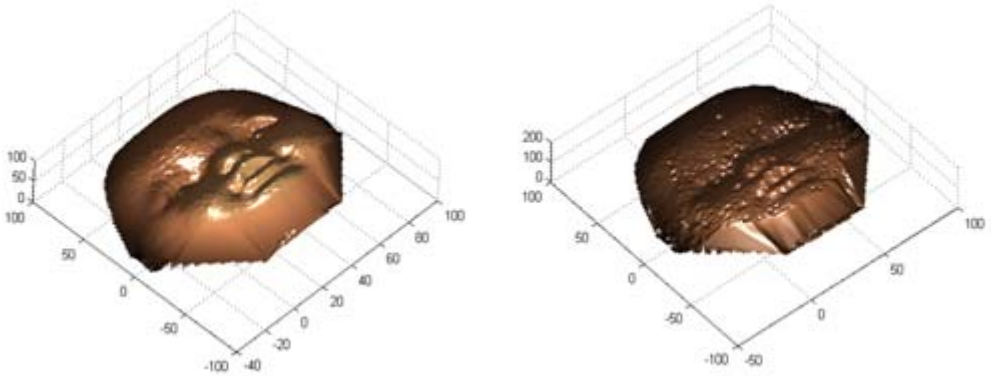
The main advantage of SEG-HRBFN compare to HRBFN in TF design is its capability of capturing all suppressed lobes and overlapping regions of importance in a successive manner as associating the lobes with a minimum number of Gaussian bases. This subsection is devoted to demonstrate the effectiveness of SEG-HRBFN in approximation with a minimum number of units and also without losing any details, as comparing it with HRBFN (Ferrari, Maggioni, & Borghese, 2004) and also with Fast-HRBFN (Ferrariet al., 2005) in 3-D reconstruction of range data presented in (Ferrari, Maggioni, & Borghese, 2004) which is known to be a successful application of HRBFN.



(a) (b)



(c) (d)



(e) (f)

Figure 4.7 Reconstruction of the baby doll face by using multilayer SEG-HRBFN reconstruction. (a) Auto-scan range data that consist of over 61.000 data points to represent the doll face, (b) First layer alone, (c) Reconstruction up to layer 8, (d) Reconstruction up to layer 14, (e) Reconstruction up to layer 22, (f) Reconstruction up to layer 29.

HRBFN model of (Ferrari, Maggioni, & Borghese, 2004) is of course not claimed to be optimal since it creates an increasing number of Gaussians with fixed width (i.e. arbitrarily doubled) for each layer without considering the signal characteristics at that layer. It checks for the necessity of several Gaussians most of which are indeed not used in the reconstruction of the original data. Nevertheless, it is a good starting point for optimization and it allows building an effective network in a very short time. The price to be paid for a high speed in the configuration is the huge number of units. Therefore, such an approach is not suitable for the construction of a TF from VHS data since merging a large number of Gaussian units generated by the physician to reach an optimal TF is a very hard (almost impossible) task to accomplish. If the number of Gaussian units is high, the physician's control over the parameter space would be lost which hardens the optimization of the TF by the physician.

SEG-HRBFN differs from HRBFN as preventing the above mentioned redundancy by choosing small numbers of appropriate basis functions. SEG-HRBFN is able to obtain a reconstruction of the function to be approximated in a similar quality but with fewer units. In order to compare SEG-HRBFN with HRBFN (Ferrari, Maggioni, & Borghese, 2004) and Fast-HRBFN (Sherstinsky, & Picard, 1996) in terms of the numbers of units required to construct a signal, a typical ensemble of range data points obtained through the auto-scan system (Ferrari et al., 2005) from a baby doll face is used (Fig. 4.7 (a)). In the reconstruction of this data, connecting the points to form a triangular mesh produces an undesirable wavy mesh and traditional linear filtering cannot be applied to clean the surface since data are not equally spaced. Moreover, the highly variable spatial frequency content of a face requires an adaptive approach. This requirements make the HRBFN based approaches suitable for that problem since the quality of the network output increases with the number of layers by adding details mainly in the most difficult regions like the nose, the eyes, and the lips (Fig. 4.7). These details are obtained by means of Gaussian clusters at smaller scales in the higher layers. In the HRBFN and Fast HRBFN approaches, these clusters are created by the network itself at the configuration period by inserting a Gaussian only where the local reconstruction error (Ferrari, Maggioni, & Borghese, 2004) is larger than the measurement error. However, these local operations produce a huge number of units.

The results in Table 1 show the effectiveness of SEG-HRBFN compared to HRBFN and Fast HRBFN in the reconstruction of a 3-D signal with a similar quality but using less number of units which is an important issue in TF initialization problem. The disadvantage of SEG-HRBFN comes from iterative machinery which is much slower than the method proposed in (Ferrari, Maggioni, & Borghese, 2004). The required time to obtain the results in Table 4.1, are 5.26 seconds for HRBFN, 1.78 seconds for Fast-HRBFN and 142.14 seconds for SEG-HRBFN. Although, there is a significant difference between approximation times due to the need for high number of Gaussian units for suitable representation of the range data; in the problem of TF initialization, the number of generated Gaussian units is not high and using minimization algorithms (i.e. Gradient Descent) for such a small number of units still allows building a network in a reasonable time (Discussed in Chapter 7).

Table 4.1 Comparison of HRBFN, FAST-HRBFN and SEG-HRBFN NETWORKS in terms of, the number of units and layers required to construct the AUTO-SCAN baby doll range data.

	HRBFN			**FAST HRBFN**			**SEG-HRBFN**		
Layer	**Grid Size (# Units)**	**ε**	**RMS Error**	**Grid Size (# Units)**	**ε**	**RMS Error**	**# Units**	**ε**	**RMS Error**
1	**14x15 (175)**	4.66	5.91	**14x15 (175)**	4.61	5.76	**1**	49.16	76.24
2	**27x29 (635)**	1.89	2.73	**27x29 (635)**	1.77	2.56	**14**	41.78	66.94
3	**53x57 (2133)**	0.796	0.32	**53x57 (2133)**	0.756	1.26	**36**	32.11	51.38
4	**105x113 (4962)**	0.397	0.76	**105x113 (4962)**	0.411	0.748	**38**	20.45	39.67
10							**142**	7.56	9.51
16							**410**	0.402	0.81
29									
Total Units	**7205**			**8087**			**2044**		

4.5 Representation and Adjustment of the Units Produced by SEG-HRBFN

After the determination of Gaussian units by SEG-HRBFN, the user selects the useful ones among all units to construct the initial TF design. For this selection process, the Gaussian units found by the network are listed together with the information on their centers, widths and the slice range they exist. User selected components (Gaussian units) from this list are then merged for making N groups, each of which will represent a tissue (Fig. 4.8). In the reconstruction phase of a group, the user can add any units to any groups and can change the structural parameters (i.e. widths, centers) of the units of a group.

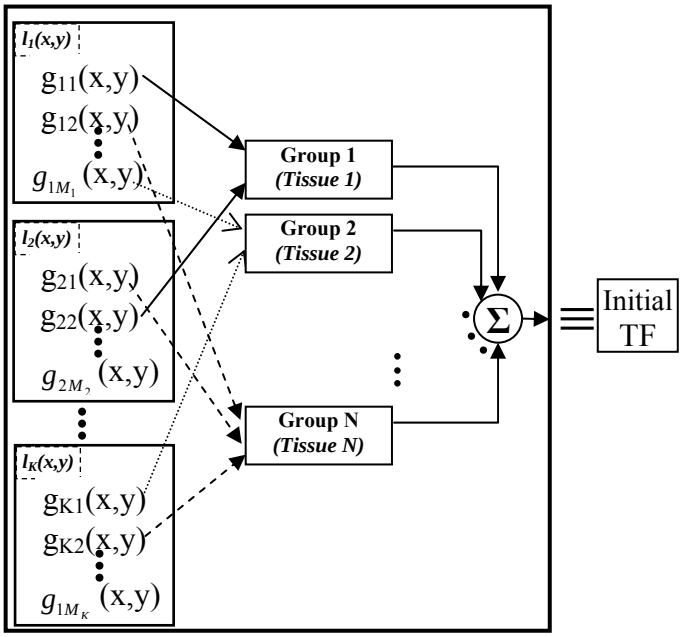


Figure 4.8 Building an appropriate TF by using the user selected Gaussian units. First the physician groups the suitable units together to represent a tissue. Then, these groups (tissues of interest) are combined to build an initial TF.HRBFN construction chart

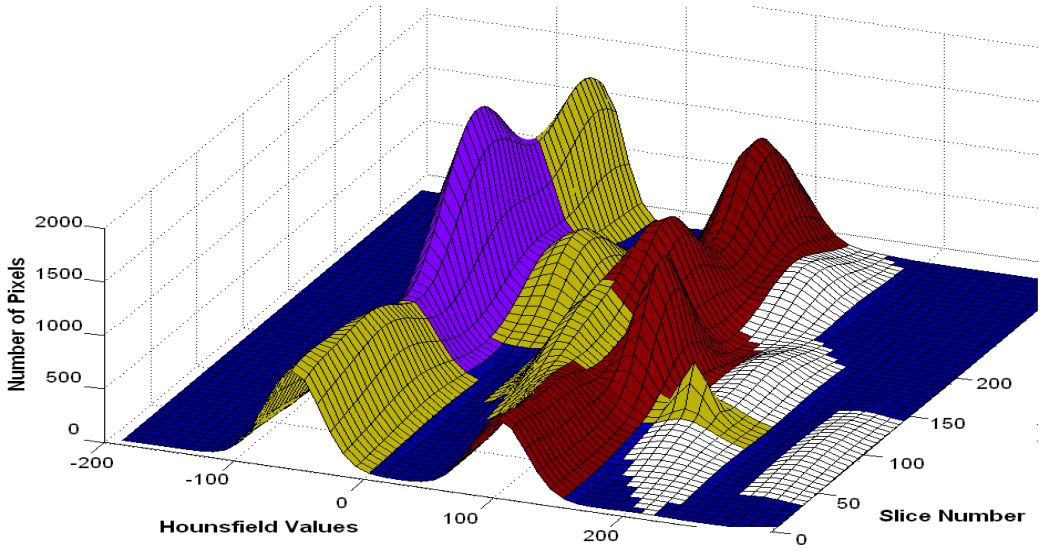
Supposing that the user wants to visualize *N* tissues, let a TF be specified by a combination of *S* Gaussian units that are selected among $M = \sum_{i=1}^{K} M_i$ units where each M_i units are generated by i^{th} approximation layer, i.e. $l_i(\cdot)$, of SEG-HRBFN. In other words, the suitable ones among *M* Gaussian units, which are generated by SEG-

HRBFN, are selected by the user and then assigned to one of the *N* groups (i.e. color and opacity) each of which represents a tissue of interest. Thus each group is constructed by Si units which might be chosen from *K* different layers. $S = \sum_{i=1}^{K} S_i$ Gaussians for *N* groups can be defined with different center and width ranges and they define the TF when combined in an appropriate way. So, the actual shape of the TF depends on the number *S* of selected Gaussian units, their centers $\left\{ \boldsymbol{c}_{i,j} \middle| \boldsymbol{c}_{i,j} \in R^D \right\}$, their widths $\left\{ \boldsymbol{\sigma}_{i,j} \middle| \boldsymbol{\sigma}_{i,j} \in R \right\}$ and the linear weights $\left\{ w_{i,j} \middle| w_{i,j} \in R \right\}$ where i, j denotes the j^{th} Gaussian unit at i^{th} layer.

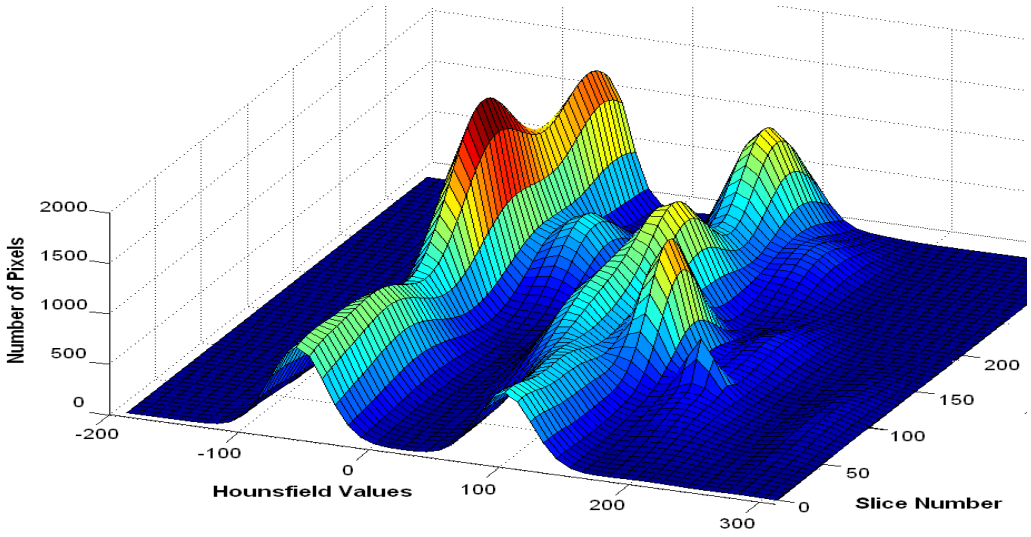
A group, which is composed of user selected Gaussian units, has two main properties: opacity and color. Thus, the voxels belonging to a group are rendered with the group's opacity and color values. Fig. 4.9 (a) shows the selected Gaussian units for the VHS data in Fig. 4.4. This approximation consists of the algebraic addition of physician selected Gaussian units among the ones found by SEG-HRBFN up to the third layer of reconstruction. As clearly seen from the figure, no grouping has been done yet and no color or opacity values have been assigned to any units.

After the initialization is finished, the optimization process controlled by the physician begins. The optimization process includes the change of free parameters of the TF (i.e. centers and widths) and the grouping of Gaussian units. By adjusting these free parameters, the user can change the location or spread of a unit, thus control the number and appearance of voxels represented with that unit. An example of grouping the Gaussian units by the physician is given in Fig. 4.9 (b).

Each group represents a tissue of interest and thus each unit takes the opacity and color value of the group it belongs to. Finally, a combination of the groups (tissues) constructs an initial TF design (Fig. 4.9 (b)). One of the important advantages of the proposed TF initialization process over 1-D TF design is also shown in Fig. 4.9 (b) where it is clear to see that the projection of the VHS data to the Hounsfield Value axis would result in an overlapping of separate tissues as in volume histogram. However, these tissues can be individually classified using the VHS data.



(a)



(b)

Figure 4.9 Building an appropriate TF by using the user selected Gaussian units. First the physician groups the suitable units together to represent a tissue (Figure 4.8). Then, these groups (tissues of interest) are combined to build an initial TF, (a) Selected Gaussian units for the VHS data in Fig. 4.4, (b) Each lobe (Gaussian unit) is assigned to a group, which represents a tissue, and takes the opacity and color of that group.

Unlike typical TFs having many parameters that have to be set by the user by hand or through by an interactive exploration of the volume data, Gaussian units have simple expressions relying on a limited number of free parameters, i.e. centers and widths. Thus, using 3-D Gaussian units as the components of the initial TF design does not

increase the memory requirements exponentially as most of the other types of components do and this makes Gaussian units practically useful and realizable in quasi real time.

This method can easily be extended to produce an initial TF where each Gaussian unit has its own opacity and color values. However, this would make the physician's intuition process more tedious and time consuming due to the need for determining color, opacity and structural parameters for each Gaussian unit. Since the data exploration is an essential element of creating a TF in medical visualization and the images have to fulfill the physician's expectations, the grouping of Gaussian units provides a physician's intuition that is both robust and requires less parameter adjustments to get a result.

CHAPTER FIVE
APPLICATION OF SEG-HRBFN BASED TRANSFER FUNCTION INITIALIZATION TO ABDOMINAL DATASETS

The method, which is introduced in Chapter 4, is applied to four challenging abdominal datasets. The first application is a CTA study acquired after the operation of stent implantation in the treatment of Abdominal Aortic Aneurysms (AAA). The second application is a contrast-enhanced renal MR Angiography (MRA) series acquired on a 3T scanner, namely the BEAUFIX data set from OsiriX database (Osirix). The third application is the PANORAMIX data-set (Osirix) which also belongs to OsiriX database and is again an abdominal CTA acquired on a 16 detector scanner for a patient with AAA. The fourth application is another abdominal CT series taken at the venous phase for the evaluation of a liver transplantation donor (one of the patient series that is used in Chapter 3). Different challenging visualization problems and tasks are handled using these datasets and presented in the following sections.

5.1 Application to Abdominal Aortic Aneurysms

AAA is a chronic degenerative disease with life threatening implications. It is thought to arise through a localized form of arterial wall injury superimposed on various predisposing factors which result in progressive aortic dilatation accompanied by alterations in vessel geometry, redistribution of hemodynamic wall stresses, and diminished tensile strength.

About 5–6% of the male population above 65 years is diagnosed with AAA each year. A common treatment for AAA consists of the insertion of a vascular graft (stent) that creates a barrier between the blood flow and the weakened vascular wall of the aneurysms. Imaging the patient before and after the graft (stent) insertion is fundamental for controlling the efficacy of the procedure.

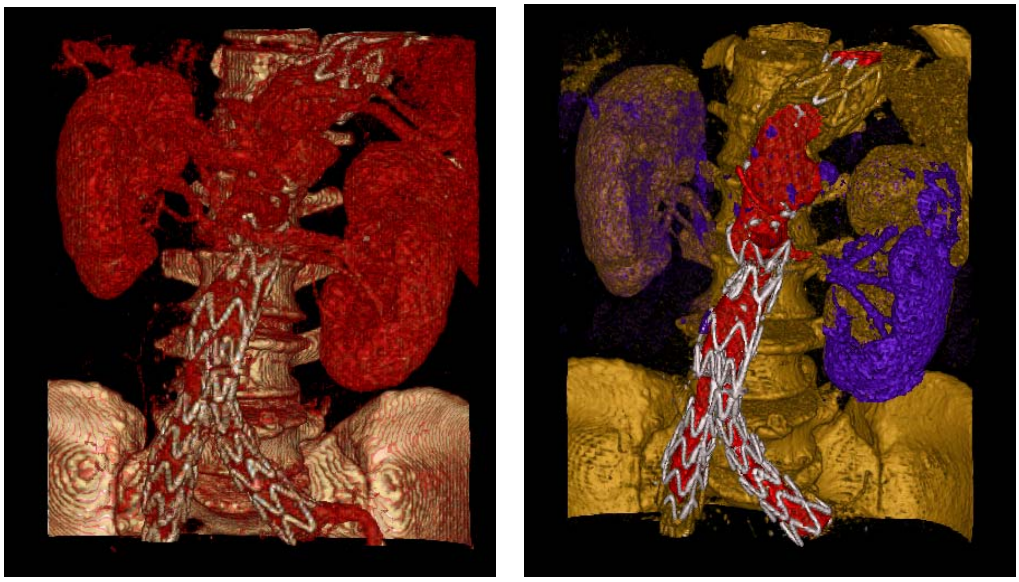
The images considered in this study correspond to CTA scans acquired with a 16 detector scanner after the graft has been implanted where the network-like structure of the graft should be visible. There are 326 images each of which is 12 bit. The main

objective of the 3-D rendering is to represent the stent and vessels clearly. However, vessels with contrast agent have significantly overlapping HV range with the bones and the stent material. Fig. 5.1 (a) shows the rendering result of a conventional 1-D TF prepared by the software in (Selver et al., 2007) as a combination of trapezoids optimized by a radiologist depending on the volume histogram.

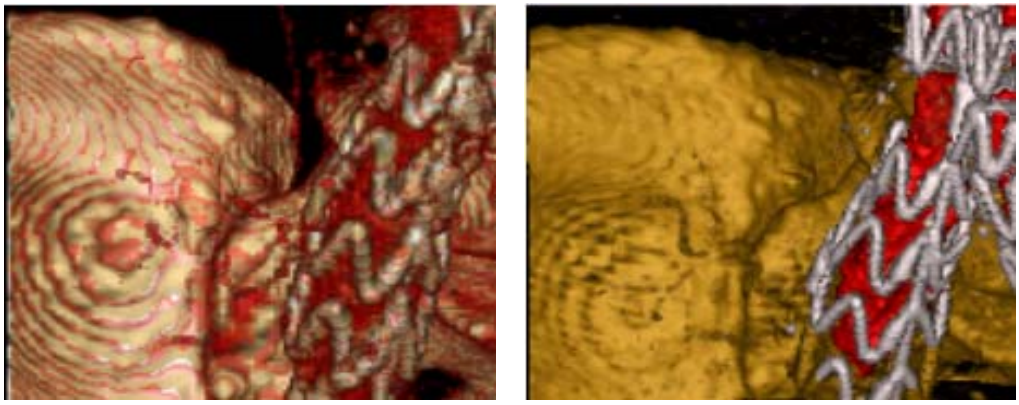
The colors chosen for rendering are white for the stent material, yellow for bones and red for the vessels. With a 1-D TF, some red voxels are included in bones and stent material. Moreover, a lot of yellow is included in stent material. These variations and mixtures in the structures of interest are evaluated as higher than the acceptable level. Fig. 5.1 (b) shows the rendering result of a TF prepared by SEG-HRBFN from VHS data. VHS data exhibits more distinctly separated lobes corresponding to the tissues of interest although these tissues cannot be segregated in a volume histogram.

The 28 Gaussian units produced by the SEG-HRBFN are adjusted through the centers and widths and are grouped. Then, a color is assigned to each group by a radiologist. The proposed method makes the vessel stand out from the bones and makes the bones stand out from the graft (stent). Also an additional group with purple color is found to represent outer surface of kidneys which improves the rendering quality. This result is also illustrated in Fig. 5.1 (c) and 5.1 (d). A small part of the volume in Fig. 5.1 (a) where the colors assigned to the bones, stent material and vessels are overlapping is shown in Figure 5.1 (c). The smearing effect due to the overlapping HV range is removed by the usage of VHS together with the SEG-HRBFN and the texture of these three tissues are rendered much smoother in Fig. 5.1 (d).

Another important criterion for evaluating the visualization result is the objectivity of the rendered images. When a physician is extracting the tissues of interest manually, the rendered images become operator-dependent and the risk of overlooking important information (i.e. suppressed features) significantly increases. Since the SEG-HRBFN automatically finds and represents all potential lobes like data with Gaussian units, the physician only needs to select and groups the useful ones, thus more objective classification is realized.



(b) (b)



(c) (d)

Figure 5.1 Rendering results of optimized TF designs (a) using volume histogram and 1-D TF (b) using VHS data and SEG-HRBFN (c) Separation of graft and spongy bone from aorta. With a 1-D TF, spongy bone and graft material can be visualized only with a smearing effect coming from the aorta, (d) the usage of VHS data and SEG-HRBFN make the aorta stand out from the spongy bone and graft material.

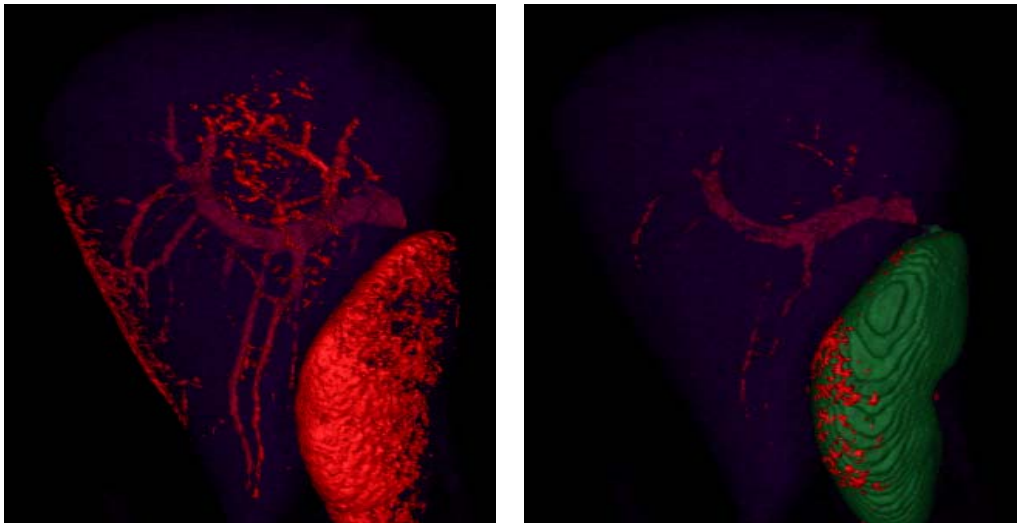
Finally, application of HRBFN instead of SEG-HRBFN for approximation of VHS data in this data set produces 198 units for the same error criterion. Due to the high number of units generated by HRBFN, an efficient user interaction is not possible in terms of assigning color and opacity to the units produced by the network. This result shows the need for SEG-HRBFN which provides an approximation with a minimum number of units.

5.2 Application to Magnetic Resonance Angiography

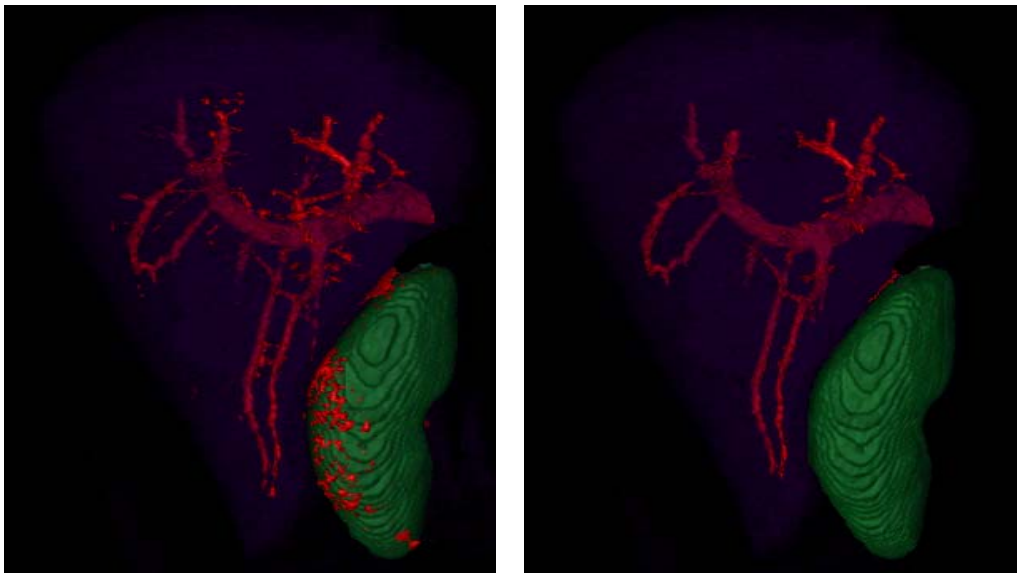
The second case of application is the BEAUFIX data set (Osirix), which is a contrast-enhanced renal MRA series acquired on a 3T scanner. One of the main difficulties in visualization of this data set is presented in Fig. 5.2. When the tissues of interest are right kidney, liver and liver vasculature, it is not possible to obtain a clear rendering using 1-D TFs due to the overlapping of intensity value ranges of these tissues. In Fig. 5.2 (a), a 1-D TF is adjusted to render the liver with purple (semi-transparent), right kidney with green (semi-transparent) and the liver vasculature with red (opaque). The overlapping of these tissues is clearly visible in 3-D rendering. The surface of the liver is rendered with a mixture of purple and green while the vasculature significantly overlaps with kidney. Next, Fig. 5.2 (b) shows the results for the rendering where the 1-D TF used to obtain the rendering in Fig. 5.2 (a) is further adjusted by narrowing the liver vasculature range and by making the kidney opaque. Now, the kidney is rendered in a clear way, however the vasculature has missing branches. So, the vasculature is not clearly visible due to the overlapping with the right kidney and liver parenchyma.

Fig. 4.2 (b) shows the VHS data that is generated for the BEAUFIX data set. SEG-HRBFN approximation results in 19 Gaussian units. When these units are combined in a suitable way by an expert radiologist, the final rendering result becomes as shown in Fig. 5.2 (c). As seen from the figure, the overlapping effects are minimized and all the three tissues (liver parenchyma, liver vasculature and right kidney) can be visualized clearly at the same time. This application also shows that the proposed method is applicable to the datasets not only acquired by CT modalities, which are standardized in the range of data values, but also to the datasets acquired by MR or any other modalities.

Other domains using intensity and gradient magnitude also yield good results for the datasets used so far (Fig. 5.2 (d)). In these methods, material boundaries can be visualized with 2-D opacity maps due to the inclusion of the gradient magnitude. Angiographic datasets are appropriate for such an operation since the boundaries are intensified due to the contrast media.



(b) (b)



(c) (d)

Figure 5.2 The results of optimized 1-D TF designs aiming to render (a) liver (purple, semi-transparent) and the liver vasculature (red, opaque), (b) liver (purple, semi-transparent), right kidney (green, opaque) and liver vasculature (red, opaque), (c) rendering results of optimized TF design using the SEG-HRBFN approximation and the VHS data, (d) rendering results of optimized TF design using the intensity/gradient magnitude domain based TF, see the text for details.

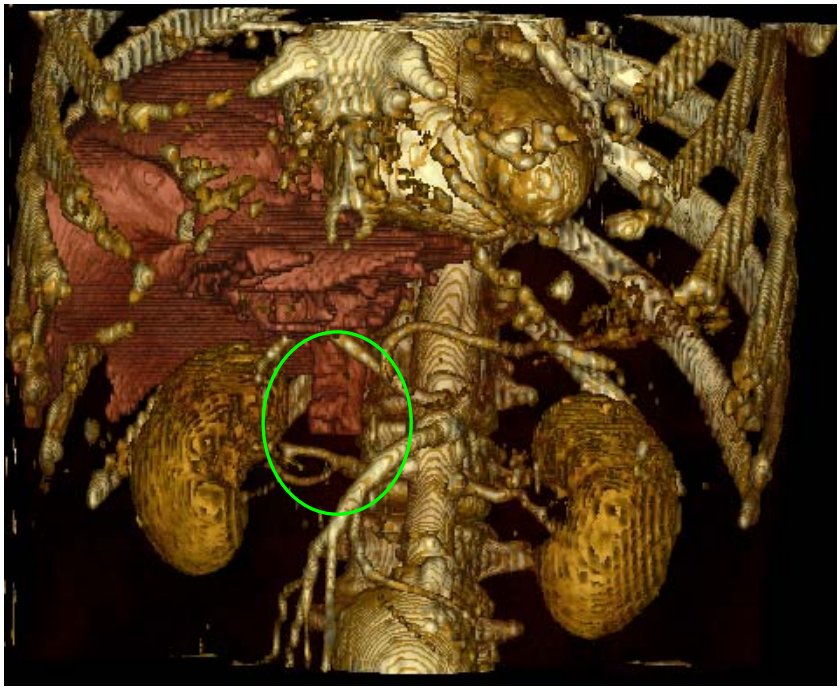
The material boundaries correspond to a scatter plots (Kindlmann, & Durkin, 1998, Kniss, Kindlmann, & Hansen, 2002) or arcs (Roettger, Bauer, & Stamminger, 2005)

which connect the footprints of the materials. The application of this 2-D domain results in with a clear rendering of kidney and liver vasculature (Fig. 5.2 (d)). There exist less misclassified voxels both for liver vasculature and kidney. However, these footprints can be indistinguishable depending on the imaging quality of the scanner, reconstruction and quantization artifacts of the scanner and the density distribution of the scanned object. In certain types of abdominal studies, the density of the boundaries between soft tissues (i.e. liver, spleen, kidneys, muscle tissues and bottom half of the heart) usually overlap significantly or even become inseparable which hardens the usage of gradient based methods. In the third and fourth medical applications, this limitation has been demonstrated and discussed.

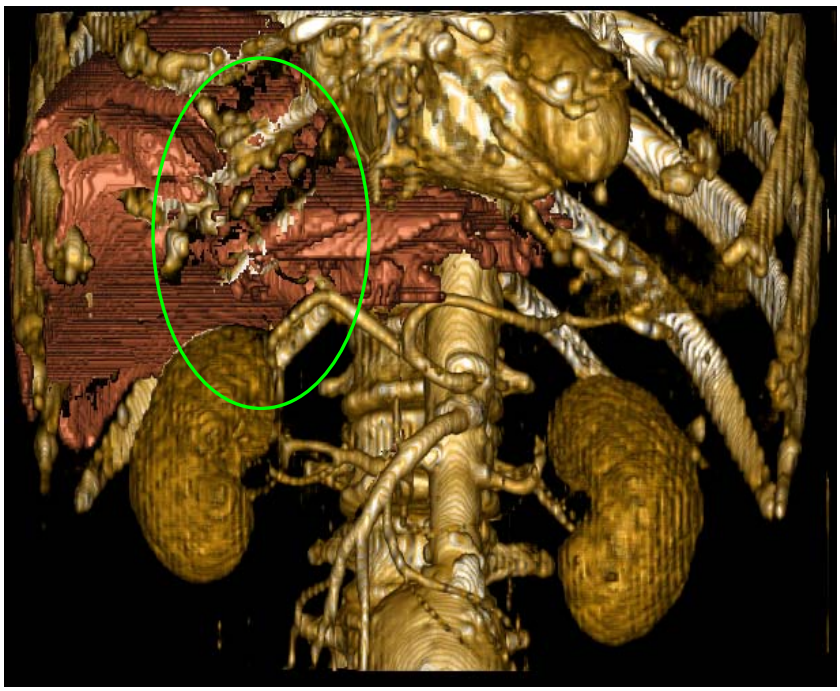
5.3 Application to Computer Tomography Angiography

The third case of application is the PANORAMIX data-set (Osirix) which is an abdominal CTA acquired on a 16 detector scanner for a patient with AAA. Although, a very high rendering quality can be obtained in visualizing aorta together with kidneys and bones, it is not possible to render the liver with these tissues using 1-D TF approach.

This data set is very suitable to show the performance of the VHS data since the kidney and liver are separated in z-direction (orthogonal to the slices). This means that they are spatially separated in the z-dimension and they get separate lobes when the VHS is generated thorough z-dimension. (Note that it is not the case in conventional volume histograms.). This gives the possibility of separating the kidney and liver by assigning different Gaussian units (using SEG-HRBFN) with different colors to the lobes that correspond to these organs. The VHS data constructed for PANORAMIX data set is similar to Fig. 4.1 (b) and Fig. 4.2 (a) shows the 0-100 range of this VHS data and plotted from the top view. Fig. 5.3 (a) shows the rendering result obtained after the optimization of the units found by SEG-HRBFN. Although the liver can not be classified alone (i.e. it is classified together with inferior vena cava and gall bladder) the smearing effect of the muscle tissue is prevented and the overlapping with the right kidney is removed which provides rendering with higher quality.



(a)



(b)

Figure 5.3 (a) Rendering results of optimized TF designs for rendering liver together with aorta, bones and kidneys using VHS data and SEG-HRBFN (b) using intensity/gradient based TF domain.

The rendering result obtained with the usage of intensity/gradient based TF domain is presented in Fig. 5.3 (b). The advantage of using gradient information over VHS is that the visualization result excludes inferior vena cava (See the green circle in Fig. 5.3 (a)) due to the density difference at the boundary between the liver and vena cava which can be determined by gradient information mainly because of the thin white outer border of the vena cava.

On the other hand, the disadvantage of using intensity/gradient based TF domain is the density overlapping between the boundary of muscle tissue and the liver results with local visualization artifacts where the muscle tissue is rendered together with the liver. (See the green circle in Fig. 5.3 (b).) This effect also blocks the rendering of the ribs and can result with incorrect determination of liver volume and size. This effect is further discussed and demonstrated in the fourth application example.

This application shows that, important information that is provided by the VHS data is the representation of the local distribution of the tissues. In this application, the advantage of using local distribution of the tissues is used to differentiate muscle tissue from the liver. The muscle tissue goes through the whole volume with almost the same amount of pixels in each slice. The distribution of the muscle tissue is shown in Fig. 4.2 (a) by the vertical line. In addition to the separation of the organs (i.e. liver and kidney), the VHS data also represents the local distribution of the tissues. The liver does not exist in all slices hence its HV range dominates only a local region in VHS data (i.e. horizontal lines). Therefore, these overlapping tissues can be classified in an optimal manner using the self-generating hierarchical design strategy of the SEG-HRBFN where the lobe that goes through the whole volume is approximated with units different than the units that constructs the lobes corresponding to the liver.

5.4 Application to Computer Tomography Angiography

The fourth application is another abdominal CT series taken at the venous phase for the evaluation of a liver transplantation donor. This application demonstrates that the VHS can be generated independent of major slicing axis which is z-dimension (axial) for this data set. The aim of the visualization is the rendering of the liver and spleen with different colors and without rendering the muscle tissue. The liver and spleen appears in

the same slices for axial and coronal images (Fig 5.4 (a)-c). Therefore, it is not possible to visualize them individually using VHS data. However, in sagittal images, they appear in different slices (Fig. 5.4 (d), (e)) thus they get separate lobes in VHS data generated for sagittal view. Fig. 5.5 shows the rendering results obtained with VHS and intensity/gradient based TF domains.

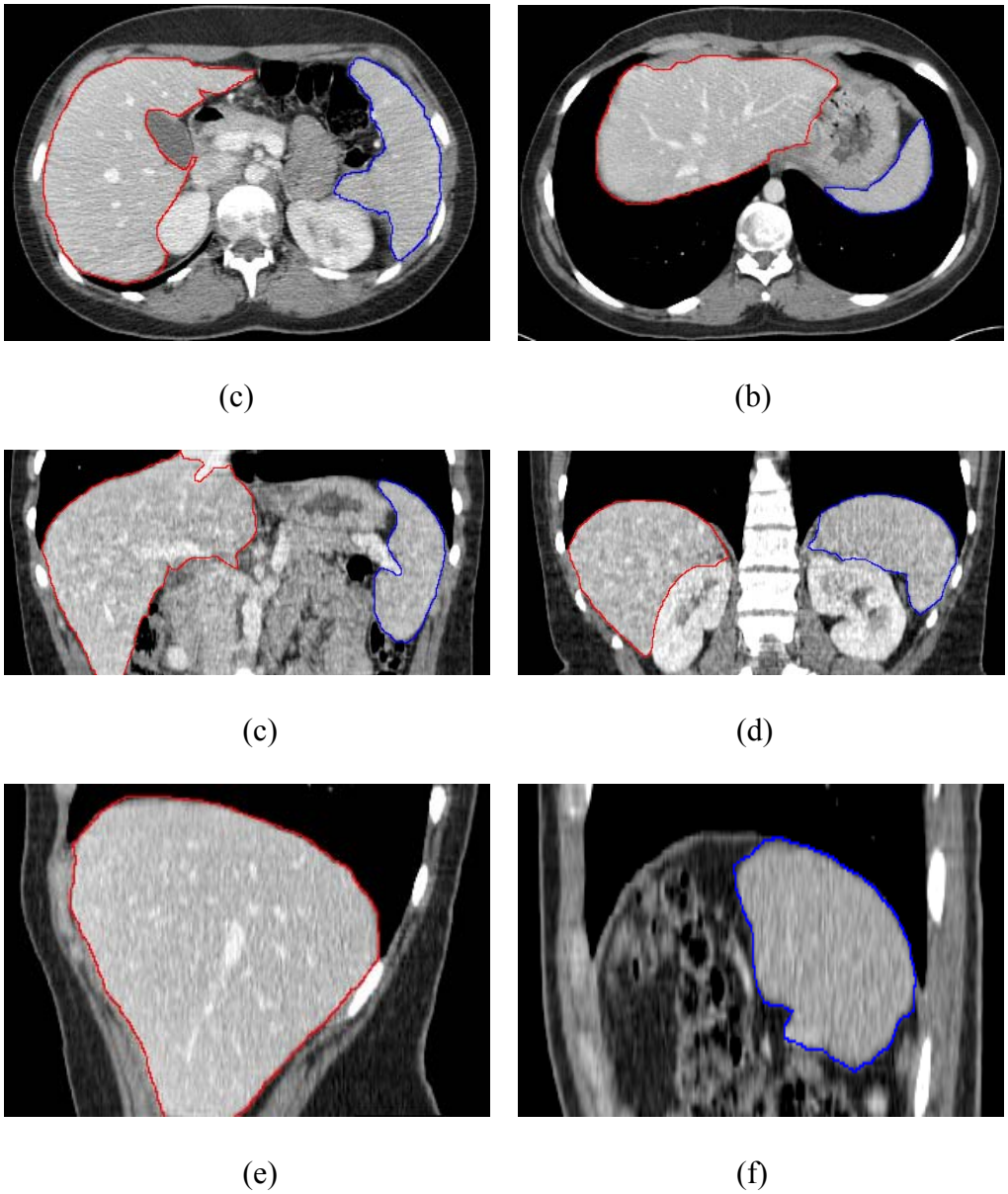
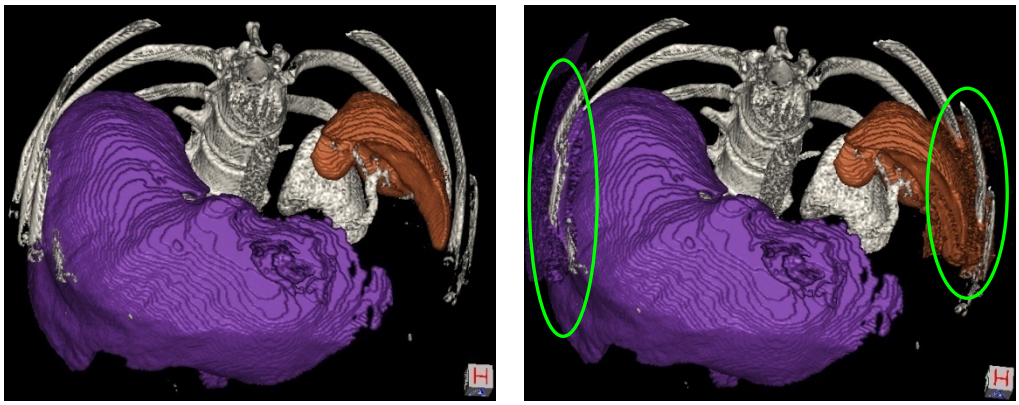


Figure 5.4 CT image examples acquired for the evaluation of a liver transplantation donor (liver: red, spleen: blue) (a) Axial image (original slicing axis), it is clear that the liver and spleen have the same density distribution and texture (b), (c) Reconstructed coronal images where liver and spleen appear together (d), (e) Reconstructed sagittal images where liver appears and ends before the spleen.

Similar to the previous application, the usage of intensity/gradient based TF domain results with a rendering in which artifacts from the muscle tissue are rendered attached to liver and spleen especially where the boundary between the liver, spleen and the muscle tissue overlap significantly and prevent a high gradient magnitude. The same effect may also arise between the boundary of liver and spleen if their boundaries intersect as in the case of atypical liver shapes (Selver et al., 2008). On the other hand, sagittal VHS data allows the determination of liver and spleen since they get separate lobes. The muscle tissue has also been eliminated from the rendering result by using the local distribution effect of liver and spleen which is a similar distribution to Fig. 4.2 (a).



(a) (b)

Figure 5.5 Rendering results of optimized TF designs using (a) SEG-HRBFN and VHS data (b) intensity/gradient based TF domain result with artifacts coming from the muscle tissue where the boundary between liver/spleen and muscle tissue overlaps significantly and do not produce a high gradient.

CHAPTER SIX
INTEGRATING DEVELOPED PROGRAMS INTO A VISUALIZATION SOFTWARE USING AN OBJECT BASED PLUG-IN INTERFACE

Developments in computer technology, high speed networking and the widespread acceptance of DICOM standard have enabled many improvements in the way hospitals view their images. In this manner, volume visualization (Kaufman, Sobierajski, 1995) and analysis is a key tool in a variety of health care applications. It is widely used in radiation oncology, surgical planning, and education. Thus, there are several commercial (Amira, Analyze, Advanced Visual Systems AVS, Upson et al., 1989, The Data Visualization & Analysis Platform, IRIS Explorer, MatLab, VisiQuest Visual Framework, VolView: A Volume Visualization System, MeVisLab: A Development Environment for Medical Image Processing and Visualization, Mevis) and open-source (Image Processing and Analysis in Java, ImageJ, Medical Image Processing, Analysis and Visualization, MIPAV, A Free Scientific Software Package, Scilab, Vis5d, VisAD Java Visualization, Osirix, Tian, Xue, Dai, Chen & Zheng 2008) frameworks with medical image processing and visualization capabilities. In most of these applications, proper segmentation of medical data is critical for producing informative results (Pham, Chenyiang, & Jerry, 2000). Since segmentation is an important step prior to rendering, there also exist several descriptive (Hahn, Link, & Peitgen, 2003, Brodlie et al., 1991, Hansen, 2005) and comparative (Bitter, Van Uitert, Wolf, Ibanez, & Kuhnigk, 2007) reviews of visualization frameworks which provide segmentation tools and algorithms. However, it is also a difficult procedure because of the restrictions imposed by variations in image characteristics, human anatomy, and pathology. Moreover, what is interesting from clinical point of view is usually not only an organ or tissue itself but its properties together with adjacent organs or related vessel systems that are going in and coming out (i.e. volume measurements, vasculature analysis and lobe determination in pre-evaluation of liver transplantation donors). For an informative rendering, these necessitate the usage of different segmentation methods in a single application, and combining/representing the results together in a proper way. One way to achieve this goal is using the following mechanism:

i) A host platform that is capable of loading, representing and visualizing medical image (i.e. DICOM) series and flexible enough to allow integrating different segmentation methods for advanced data processing.

ii) A plug-in system and data transfer protocol to establish the communication between the host and plug-in that provides the possibility of using different segmentation tasks by integrating them to the host.

According to this approach, this chapter introduces:

iii) Implementation of such an interface which can be used to plug-in and then to apply a segmentation method to a DICOM series.

iv) Developed plug-ins, which are semi-automatic (general purpose medical image segmentation) and/or automatic (specifically designed for an application), and their integration to the host.

The design of the interface is based on handling each segmentation procedure as an object where all parameters of each object can be specified individually. Therefore, it is possible to use different plug-ins with different interfaces and parameters on the segmentation of different tissues in the same data set while rendering all of the results together is still possible. The design allows access to ITK, Java and MatLab functionality together, eases sharing and comparing segmentation techniques, and serves as a visual debugger for algorithm developers.

The plug-ins for the proposed segmentation and transfer function initialization methods, which are introduced in Chapter 3 and 4, are implemented according to the programming system that is described in this chapter. During this chapter, first, the properties of the host program, the idea behind the object based segmentation approach is described. Then, integration of a plug-in to the host is introduced by using the plug-in that is developed for liver segmentation as an example.

6.1 Introduction

There are mainly three issues that should be handled properly in the field of medical image segmentation:

1) Segmentation is generally considered to be the most challenging step prior to 3-D rendering because of the restrictions imposed by variations in image characteristics, human anatomy, and pathology. Due to these large variations, the design of a segmentation routine is extremely challenging in the medical context. Even if a routine works efficiently in normal subjects, they typically fail in pathologic cases which often are more interesting from clinical point of view. Hence, there is a strong need for a high number of segmentation methods that can be used in different clinical studies and cases. Moreover, in most of the cases, what is interesting from clinical point of view is usually not only an organ or tissue itself but its properties together with adjacent organs or related vessel systems that are coming into and going out of it. For an informative rendering, these necessitate the usage of different segmentation methods in a single application, and combining/representing the results together in a proper way.

2) In these segmentation methods, operator interaction should always be possible although it introduces a subjective element to image processing and analysis (Olabarriaga, & Smeulders, 2001). Providing user interactions in the early stages of the design process may considerably decrease failure rates. On the other hand, a high degree of automation is required in order to achieve high accuracy and precision. Thus, a user-friendly implementation of interaction tools should combine intuitive and easy handling together with acceptable performance.

3) In segmentation of medical volumes, another important criterion is the representation of segmentation results. A slice by slice representation, which is often more accurate and complementary with 3-D rendering, can easily be provided. On the other hand, volumetric visualization of segmentation results gives more information to the user for further analysis (Caban, Joshi, & Nagy, 2007). However, 3-D visualization is another area of research considering that the image quality, memory usage and interaction mechanisms should be provided in an advanced level. This area of study is mostly out of scope and time consuming for the researchers who are focused and studying on segmentation.

Considering these three issues, in this chapter, two main topics are introduced:

i) the implementation of a medical volume visualization software that can be used as a host program to integrate plug-ins that might be written using several different applications such as ITK (Ibanez, & Schroeder, 2005)., MatLab (MatLab) and Java.

ii) Programming and development of a plug-in that is based on the liver segmentation method introduced at chapter 3.

By developing a plug-in interface to the host medical visualization software, researchers can access their new image processing algorithms from within a full-featured visualization application and allow algorithm developers to quickly verify and improve their new processing techniques.

The design of the plug-in interface is based on handling each segmentation procedure as an object where pre-processing, post-processing and rendering parameters of each object can be specified individually as well as the method and parameters of segmentation. In a more general sense, the goal of the approach presented in this study is to drive the segmentation from a visualization viewpoint. Each segmentation process follows the typical visualization pipeline, consisting of the modules 'Pre-processing', 'Segmentation', 'Post-processing' and 'Rendering' where the module 'Segmentation' can be provided by the user with a plug-in. Appearance of each visualized object (or, in other words, result of each segmentation procedure) can be modified by adjusting the parameters of the above mentioned modules Thus, the proposed strategy creates a direct link between the object manager used for visualization and the segmentation interface used to obtain a specific object.

This approach aims and allows the application of different methods (and/or same method with different parameters) to the same data set by assigning an object to each. Rendering all of the objects (segmentation results) together or individually and handling the appearance of each is possible using the object manager. Although similar studies are reported in the literature (Martin, Ibanez, Avila, Barre, & Kaspersen, 2005), using more than one plug-in in order to visualize different tissues with different segmentation methods at the same application has not been discussed. Moreover, no handling mechanism (such as the object manager in this study) or interface has been proposed or

designed for handling multiple organs/tissues to visualize. During this study, different plug-ins developed by using ITK, MatLab and Java are integrated into the visualization software and applied to various medical cases to show the effectiveness of the proposed method and design.

The rest of this chapter is organized as follows. Section 6.2 gives information about the host medical image viewer in general. Then, Section 6.3 explains the plug-in architecture and life-cycle while Section 6.4 introduces different plug-in applications that are developed based on the proposed architecture. The comparison of the developed liver segmentation plug-in with the general purpose plug-ins is also given.

6.2 Features of the Host Visualization Software

6.2.1 Software Capabilities in General

Even it supposes to work as a main application for segmentation plug-ins, as a medical image viewer, the host software should provide some basic functions (Hoffmann, 2000). Therefore, the following functionalities are provided by the host:

1) Image query and receive from any image source (i.e. PACS, local hard drive, CD or any other storage medium) is possible using the host software. It can decode and display most common image formats (i.e. jpg, jpeg, png, bmp, tiff etc.) including DICOM. DICOM images can be received via the DICOM protocol or using a DICOMDIR file. If not available, the host software creates a DICOMDIR file by searching through the files at the selected folder. DICOM services for image display and PACS connection are implemented in detail and tested with images of several vendors including Siemens, Sectra, Agfa, and GE.

2) Being Java based, it is platform independent and usable via web which is important due to the heterogeneous environments of hospitals and wide acceptance of different operating systems (i.e. Windows, Linux etc.) that require the software programs to be able to run on these platforms. Contrary to expectations, the performance issues in Java are comparable to C++ as shown by several studies (Vivanco, & Pizzi,

2004, Prechelt, 2000), where the performance of the virtual machine optimization is similar to C++, especially in loops.

3) Several basic image manipulation and processing tools (i.e. filters, LUT operations, rotation, measurements etc.) are implemented. A user friendly navigation method and interface for displaying different datasets of the same or different patient(s) have been designed. The GUI (Fig. 6.1 and 6.2) is designed to look like readily learned and used (e.g. require a manual as less as possible), yet it is flexible enough to meet the varied needs of different healthcare professionals.

4) Advanced imaging techniques including MPR, Oblique Sectioning, Curved MPR, (Robb, 1999) volume and surface rendering capabilities have been implemented.

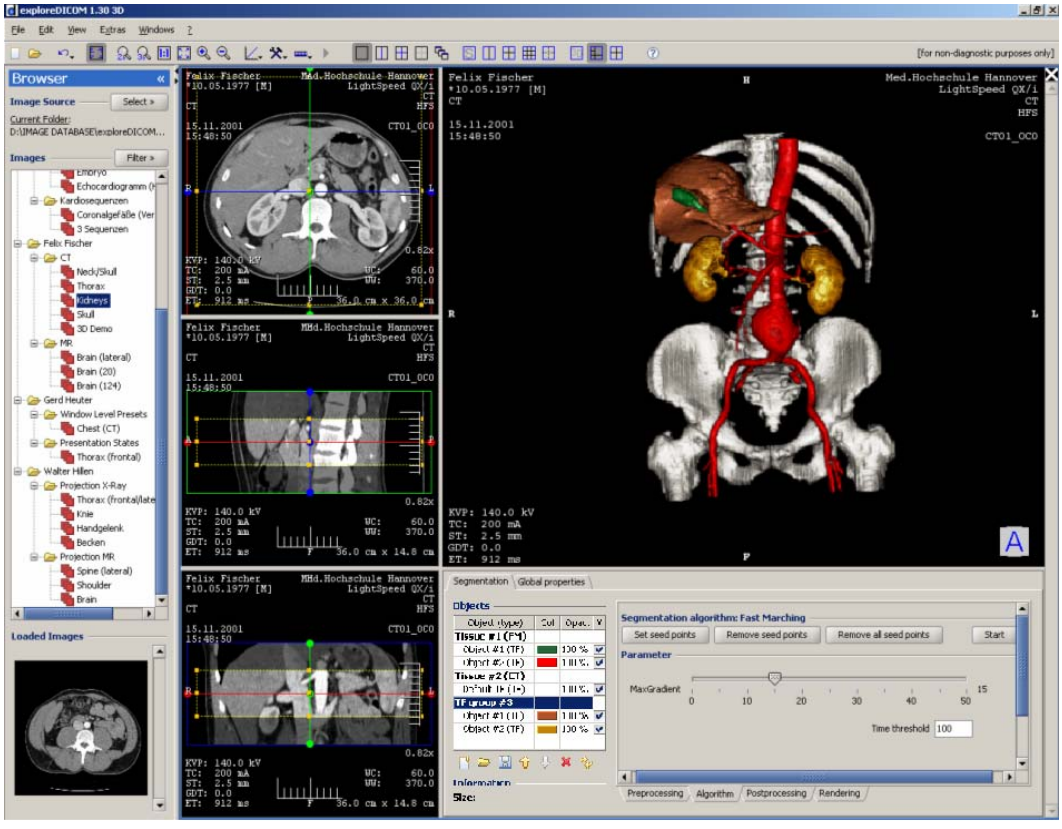


Figure 6.1 User Interface of the host software. The GUI layout is given in Fig. 6.2. The software is capable of displaying single images, image series, or other sequences. The 3-D rendering based visualization of DICOM series is integrated and supported by VOI selection, seed point insertion and segmentation plug-ins.

6.2.2 Software Capabilities in 3-D Visualization

A 3-D imaging software should have functions to provide the ability to gain desired information from 3-D images (Tory, & Torsten, 2004). First requirement is speed which should be fast enough to avoid user frustration. However, this performance should not only be provided by the hardware, which would drive the costs to impractical levels, but also with effective software design and programming. The host software provides high performance rendering on standard PC hardware and works minimum with a standard Pentium III processor, 100MB free hard disk space and for 3-D visualization a graphics card that supports OpenGL. To make the software run efficiently on different hardware, the most important parameters that determine the time and memory needs of visualization (i.e. interpolation type, rendering method etc.) can also be determined by user. For example, if an advanced graphics card is available, texture mapping can be used instead of ray-casting. For slower computers, shading and high quality interpolation can be turned on after adjusting other parameters.

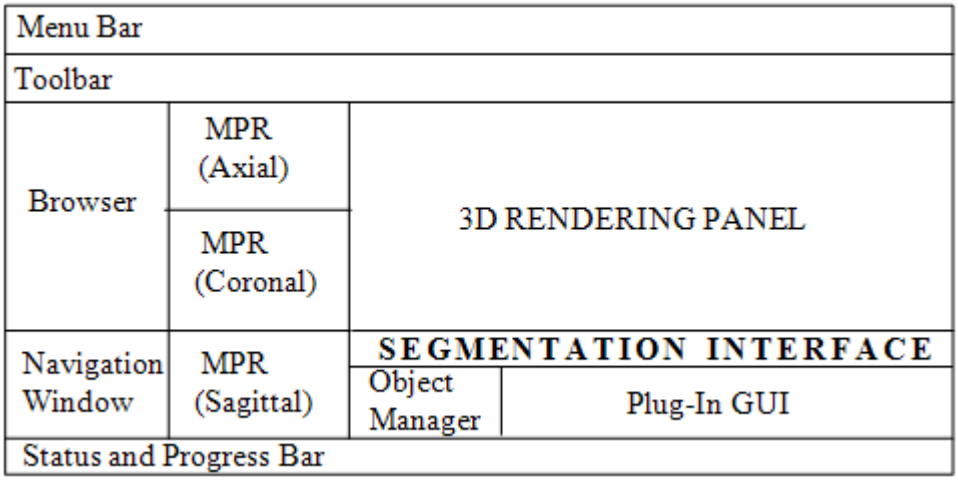


Figure 6.2 Layout of the user interface of the host software in 3-D mode (MPR: Multi Planar Reconstruction).

Second, it is important to have the opportunity to visualize more than one image series at the same time for comparison of different studies which can be done with the host viewer. As the capabilities provided by imaging modalities (CT, MR etc.) are increasing, software tools, which can visualize this data independent of modality type, are needed. Different rendering techniques should also be supported and be selectable

by the user since each can be used in different cases. In host software, SR, VR and MIP are integrated and can be used independent of the modality type. In the implementation of these techniques, VTK has been used in addition to Java. Tools for adjustment of important parameters (shading, interpolation, TF etc.) are integrated to enhance structural understanding.

During development, different formats were evaluated to store 3-D rendering information. For performance reasons, the binary (uncompressed) raw format was chosen. For formats like Virtual Reality Modelling Language (VRML) data has to be encoded in ASCII, which takes a lot of time and also space on the hard drive. It is not efficient for large datasets, especially if they are not stored within databases, but transported only.

6.2.3 Interaction Mechanisms for Supporting Segmentation

Except fully automatic ones, segmentation techniques require adjustment of parameters and/or other kind of interactions (i.e. insertion of seed points) by the clinician. This dictates that the plug-in mechanism must provide for a GUI so that the user can select parameters, adjust values or seed points. Some of these interaction mechanisms are hard to be implemented in a plug-in interface. For example, it would be inefficient to determine seed points on a plug-in interface instead of the doing it directly over the images.

Therefore, some basic interaction mechanisms are implemented to support plug-in interface of the host software which are: i) Possibility of selecting a VOI using MPR images (i.e. axial, sagittal, coronal), ii) Possibility of inserting seed points on MPR images (i.e. axial, sagittal, coronal), iii) Interactive cropping of the volume by adjusting the size of a bounding cube. The viewer provides necessary buttons and other mechanisms automatically if the above mentioned interaction mechanisms are requested by the plug-in.

6.3 Plug-in Architecture and Workflow

Each segmentation object, which is responsible for handling a segmentation procedure (i.e. pre-processing, post-processing and rendering parameters), can be controlled by an interface element called “Object Manager” (OM) (Fig. 6.3). The OM is designed to provide flexibility in application of different segmentation methods and rendering their results together. Thus, any number of different segmentation methods can be applied to a data set and/or the same segmentation method can be applied several times with different parameters and all of the segmentation results can be rendered together using OM. As shown in Fig. 6.3, each segmented object (i.e. segmentation result) have its own name, color, visibility (i.e. Boolean: Visible or not) and opacity. User can change any of these parameters easily using the OM and can determine which segmented objects should be visible and rendered with which color and opacity. Moreover, each object has a default TF, which is a useful tool that assigns optical parameters, (i.e. color, transparency) to the voxels in interactive visualization (Selver et al., 2007).

Since TF specification provides an interactive classification step, where on the fly combinations of these selected parameters (i.e. opacity, color) can be determined during the rendering pipeline, it is always important to provide a powerful TF specification tool on the GUI. In the host software, TFs can either be used as individual objects or at the final classification step of an object. A TF of an object covers the whole range of histogram of that object after the segmentation process. For each object, defining any number of TF is also possible for further classification of segmented data based on volume histogram (Selver et al., 2007) and/or intensity-gradient graphs (Kindlmann, & Durkin, 1998). More advanced strategies are also being host to create a more powerful tool (Selver, & Güzeliş, 2009).

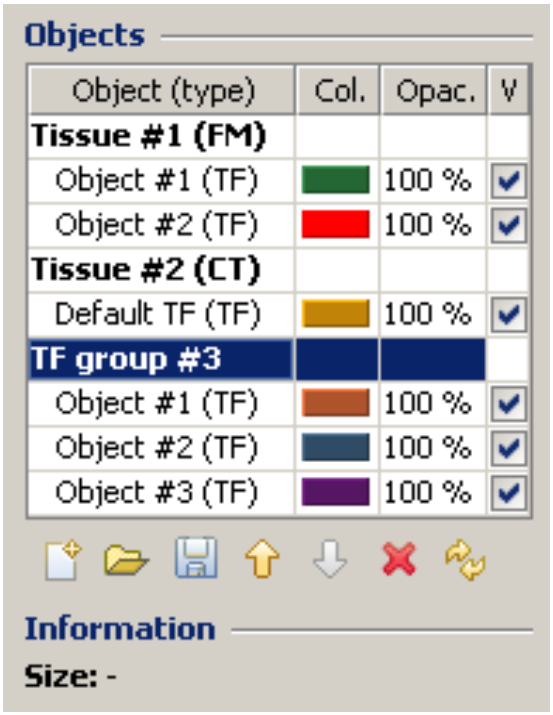


Figure 6.3 Object Manager (OM). Each segmentation procedure (i.e. provided by a plug-in), which applied to the data set, is presented as an object (i.e. Tissue #1, #2 and #3). The abbreviations in the parenthesis refer to plug-in name (i.e. FM: Fast Marching, CoT: Connected Threshold, TF: Transfer Function). Each object can also have a number of TFs as sub-objects. Each sub-object has its own color and opacity values. V (at the right most of the panel) refers to the visibility of an object (i.e. boolean). 'Size' label is used to display the volume of a segmented object using DICOM info and number of voxels. Layout of the user interface of the host software in 3-D mode.

6.3.1 Visualization and Plug-in Workflow

One of the most interesting aspects of the mechanism used for integrating Java, ITK or MATLAB based plug-in capabilities into the visualization software is the fact that communication between the visualization program and the plug-in tool do not have to be exposed to the internal structure of the visualization software. Instead, the plug-in

interface has been defined in the host visualization software that allows very generic methods to interface with the internal data representation. The overall workflow of the segmentation process, which is explained in detail in the following paragraphs, is illustrated in Fig. 6.4.

When the 3-D mode of the viewer is initiated, the visualization software detects if a plug-in is available or not by searching the corresponding directory automatically. Then, it adjusts the GUI and rendering techniques to accommodate the new data and shows a list of available plug-ins in the plug-in selection panel (Fig. 6.5 (a)). Among all, the user selects the segmentation method (plug-in) to use together with the rendering type of the resulting data (i.e. VR, SR)

Another interesting aspect of the software-design is that the plug-ins can be attached at run-time. Although the plug-in itself has its own specified and optimized algorithm for the segmentation process, the user can still use a high number of ITK method during the pre-processing and post-processing stages of the segmentation process. Together with the pre- and post-processing parameters, segmentation parameters can be defined by using the “algorithm interface” or in other words “plug-in GUI” (i.e. shown at bottom right part of Fig. 6.1 and Fig. 6.2 and in Fig. 6.10). The “algorithm interface” for the determination of the segmentation parameters is loaded dynamically from the plug-in. It might inherit some predefined buttons form the plug-in library (i.e. button that allows inserting seed points over MPR images) as well as any other kind of GUI elements (i.e. sliders, text fields etc.) to adjust and determine the parameters for the corresponding segmentation algorithm.

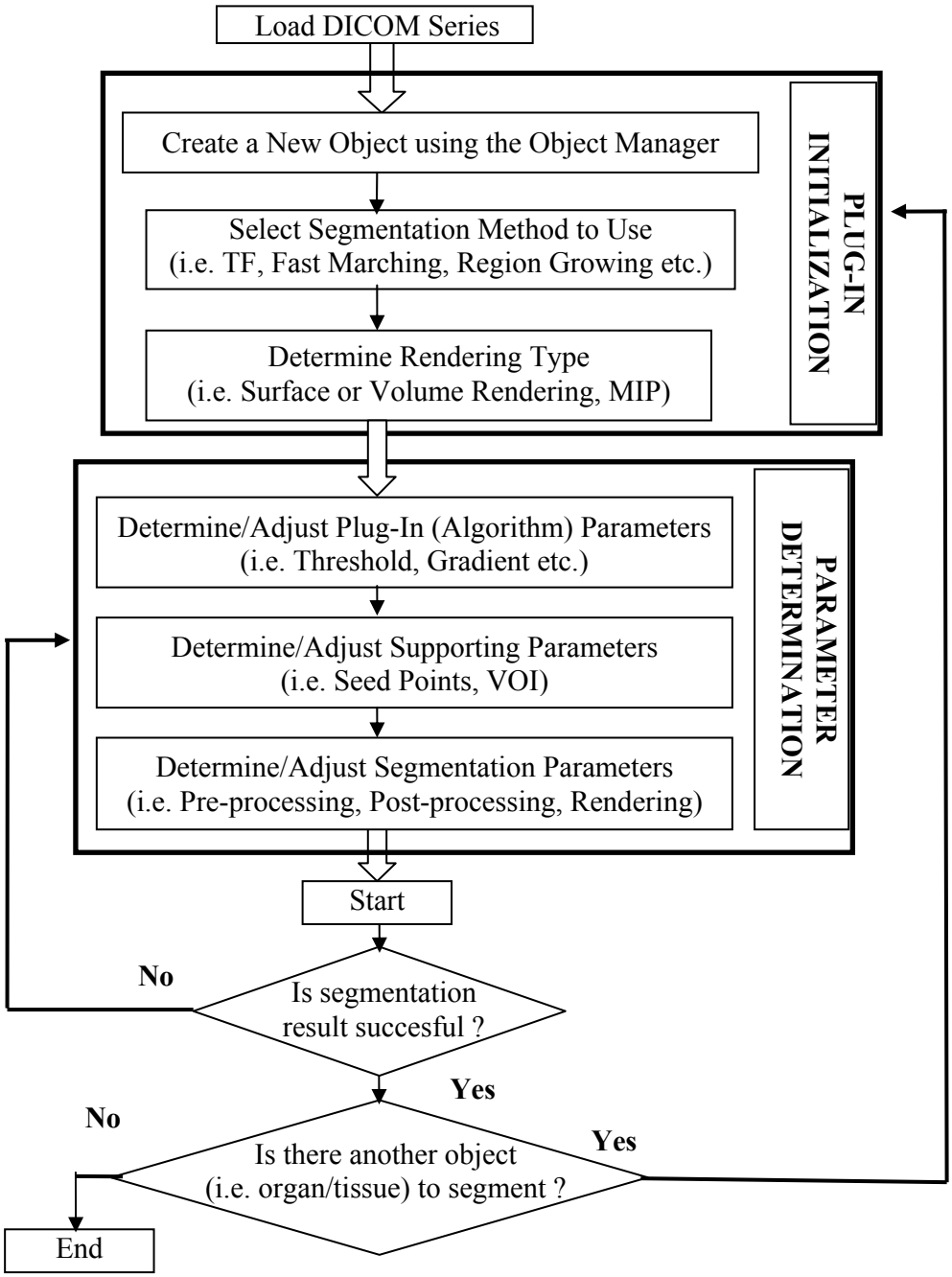


Figure 6.4 The overall workflow of the segmentation process.

Then, the user can press the "Start" button to apply all stages (i.e. Pre-processing, algorithm and post-processing) to the data set. After the segmentation result is rendered at the 3-D Panel, the user can adjust the parameters of all steps and re-start the procedure until obtaining a satisfying result. At any moment of this process, the user can also adjust the global properties (Fig. 6.5 (d).) and the rendering parameters (Fig. 6.5

(e)) to change the appearance of the rendering result. All of these elements are explained in the following sub-sections.

6.3.2 Plug-in Interface

Most segmentation techniques require that the clinician specify a number of parameters or seed points for the segmentation which necessitates that the plug-in mechanism to provide a GUI to adjust these parameters. These parameters differ from one plug-in to other so a plug-in specific interface with pre- and post-processing operations. Moreover, segmentation results of different tissues might be better visualized with different rendering parameters such as the position of the light source, type of shading, type of projection etc.

Hence there are global parameters that can be adjusted for all plug-ins and plug-in specific parameters that are needed for the segmentation method in use. Considering that the space to handle all these parameters should be small enough to prevent inefficient usage of screen space, an efficient and still intuitive interface should be designed. In this study, such an interface is implemented with a tabbed panel system.

The main two (upper) tabs (shown inside the dashed red circle on Fig. 6.1) can be used to determine “Segmentation specific” or “Global” properties. In Global properties, user can determine intensity, ambient color, diffuse color, specular color and position of the light source together with the projection type (i.e. parallel or perspective with projection angle). On the other hand, “Segmentation” properties consists of four other tabbed panels (Fig. 6.5 (b)-(d)) that are arranged based on the assumed workflow of a segmentation process (Fig. 6.4).

These tabbed panels include Pre-processing, Algorithm, Post-processing and the Rendering tabs. The idea behind this arrangement is that a segmentation procedure starts with the determination of pre-processing operations, continues with the execution of the main algorithm, then followed by post-processing operations and finished by the determination of the rendering parameters. The mentioned Pre/Post-processing (i.e. ITK functions) and the Rendering tabs are integrated in the base application. In the tabbed panel, the plug-in specific GUI reserves a tab (i.e. Algorithm). The tab before that is the

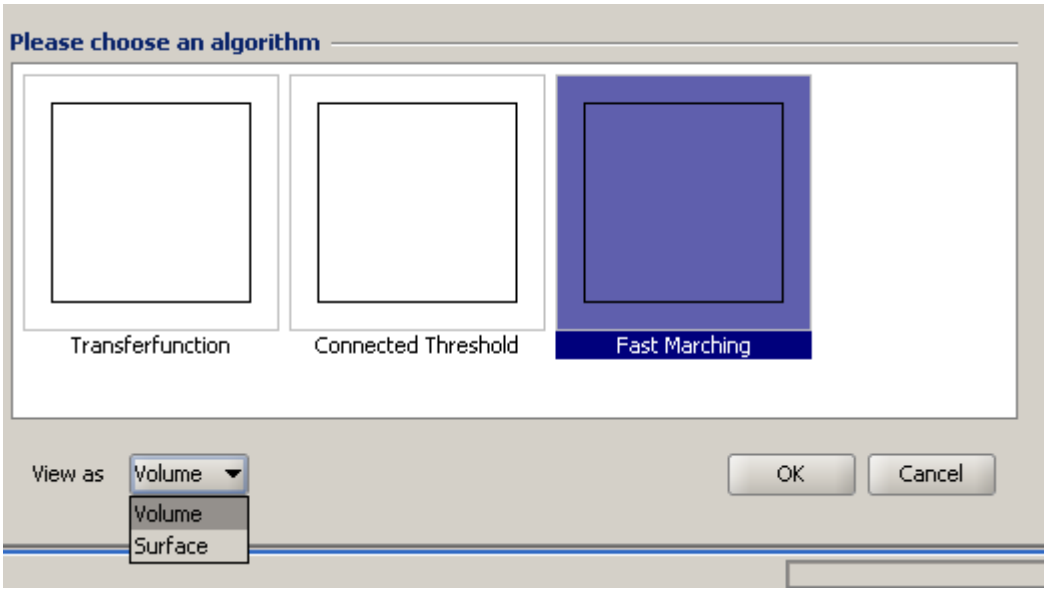
pre-processing and the tabs after that are the post-processing and rendering parameters. So the user can determine and use pre/post processing operations provided by the host on the runtime (using their GUIs). For any other pre/post-processing operations, the programmer should include them inside the plug-in during the development phase.

Pre-processing tab (Fig. 6.5 (b)) includes smoothing filters (i.e. Gaussian and anisotropic diffusion), threshold determination and sampling rate selection which can be used to reduce data to be processed for computationally expensive plug-ins. The default selection is automatic in which the visualization software estimates how much memory will be required to run the plug-in. This option prevents crushes for segmentation techniques that require a large amount of memory for intermediate results which can easily exceed the computer's memory for a large medical data set.

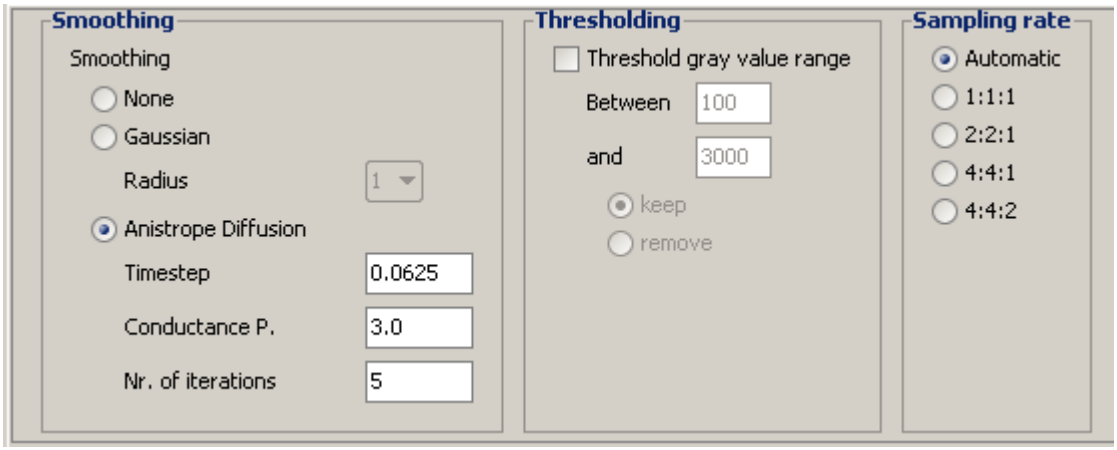
Algorithm tab includes GUI elements that are needed to determine the parameters required for an algorithm. Developing plug-in GUI in Algorithm tab is an easy task, which is done on the Java-side. One panel has to be provided, which can be created either with a Visual Editor (e.g. Net Beans) or manually. Any kind of GUI elements can be used to increases the intuitiveness of the GUI (Fig. 6.10). The Algorithm tab, especially the data exchange between GUI and algorithms is explained in detail in part C.

Post-processing operations include smoothing filters and anti-aliasing binary image filter from ITK which can be used to enhance the rendering quality of a segmented object if surface rendering is chosen.

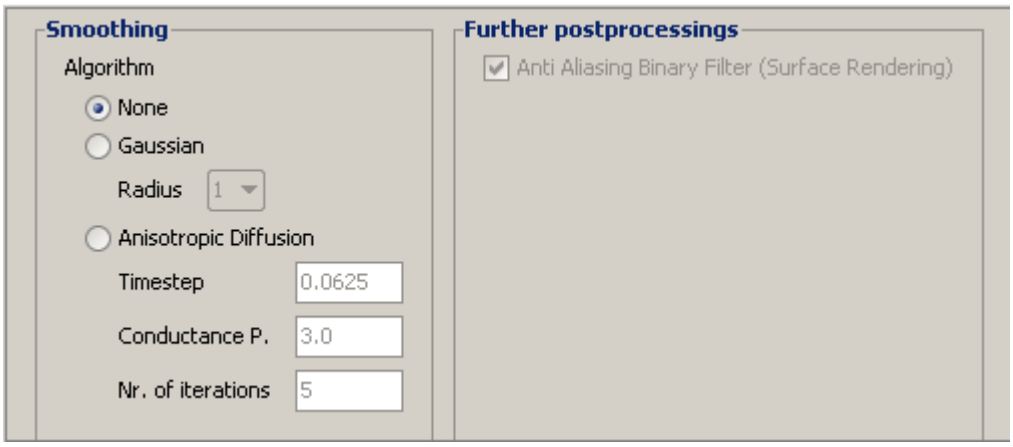
Finally, the Rendering tab (Fig. 6.5 (d)) includes volume rendering parameters such as algorithm selection (i.e. ray-casting or texture mapping), interpolation methods (i.e. nearest neighbor or tri-linear) and sampling rate. It is also possible to adjust shading type, reflection (i.e. diffuse and specular) and specular power using this tab.


Please choose an algorithm
Transferfunction
Connected Threshold
Fast Marching
View as
Volume
Volume
Surface
OK
Cancel

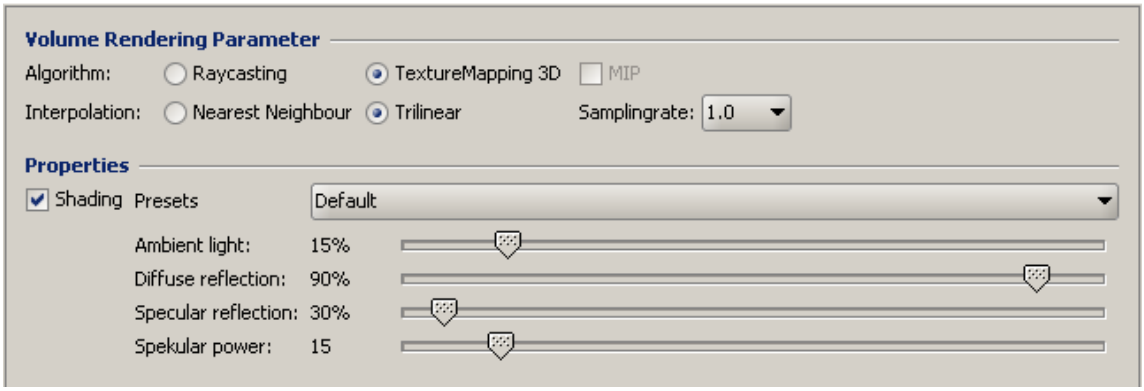
(a)


Smoothing
Smoothing
None
Gaussian
Radius
1
Anistrope Diffusion
Timestep
0.0625
Conductance P.
3.0
Nr. of iterations
5
Thresholding
Threshold gray value range
Between
100
and
3000
keep
remove
Sampling rate
Automatic
1:1:1
2:2:1
4:4:1
4:4:2

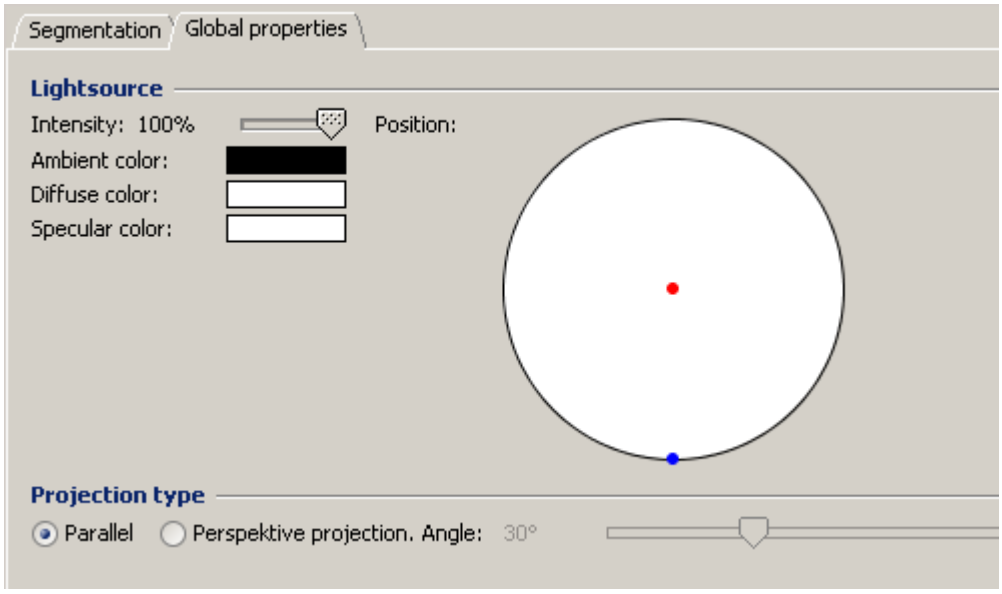
(b)


Smoothing
Algorithm
None
Gaussian
Radius
1
Anisotropic Diffusion
Timestep
0.0625
Conductance P.
3.0
Nr. of iterations
5
Further postprocessings
Anti Aliasing Binary Filter (Surface Rendering)

(c)



(d)



(e)

Figure 6.5 User Interface of the developed tabbed panels, (a) Selection of the plug-in and rendering method to be used (The square shapes are designed for possible icons of the plug-ins. An icon representing the corresponding plug-in can be inserted inside that area) (b) Pre-processing options (c) Global properties tab and (d) Rendering tab that includes sliders and checkboxes to determine some of the rendering parameters (e) Global properties of the object to be visualized.

6.3.3 Plug-in Life Cycle and Writing a Plug-In

The life-cycle of a plug-in consists of five key functions: Registration, Initialization, GUI Update, Data Processing and Destruction (Fig. 6.6). In this section, the steps of writing a plug-in is described by using examples from an existing plug-in developed for liver segmentation (Selver et al., 2008).

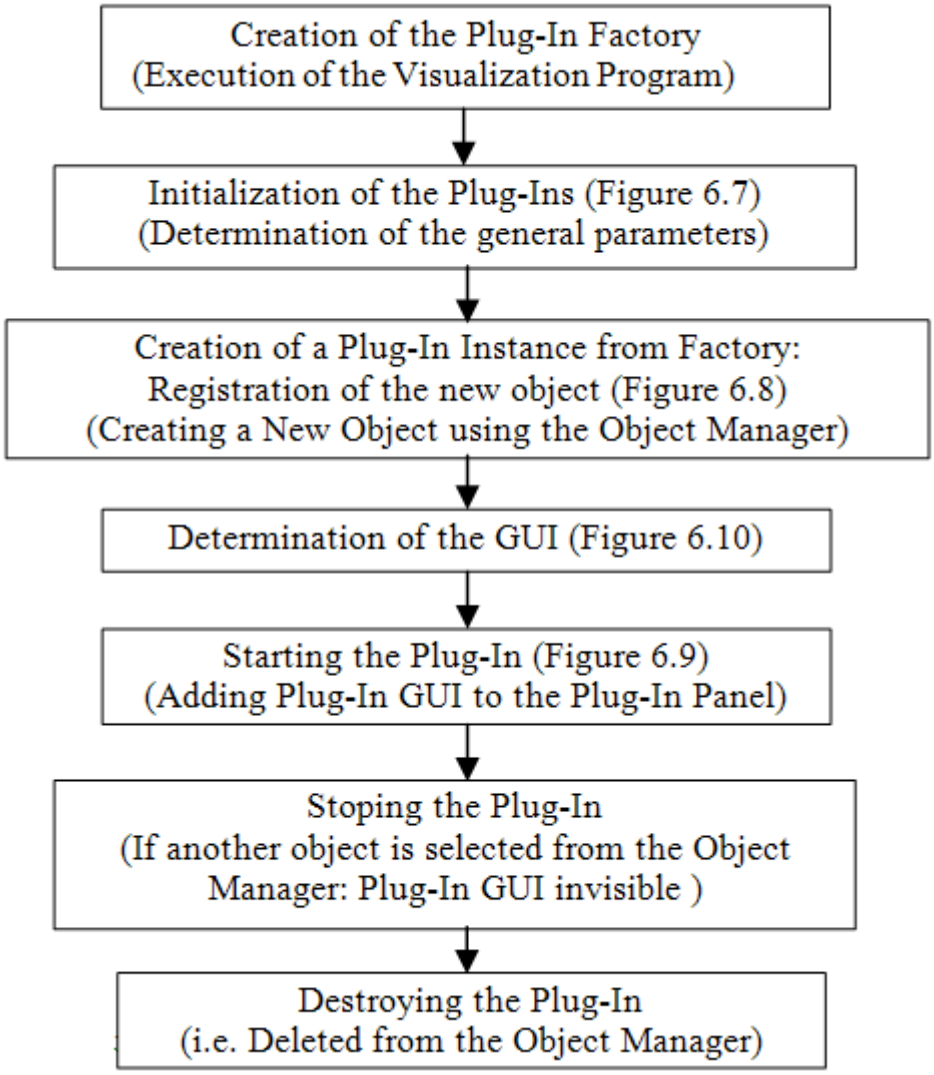


Figure 6.6 Plug-in life cycle.

When the visualization software is executed, it checks the “plug-ins” folder for the available plug-ins. Then it creates a plug-in factory by assigning a number to each available plug-in. Transfer of the data between the host and the plug-ins is always handled on the Java side. There are two ways of communication: 1.) Memory based, 2.) Disk based. The advantage of 1 is the high speed, but the disadvantage is that the whole data set has to be stored twice in memory. Approach 2 is slightly slower than 1 but the data is stored only once in memory. This is important in Windows, because there the 2 GB RAM limit exists. Mac OS X and Linux do not have such memory limitations. The plug-in factory stores the information about the data that will be needed by the plug-in, such as, seed point locations (i.e. for Fast Marching), if the plug-in uses the whole

volume or only a defined VOI etc. The plug-in can receive a full description of the input data it needs, including aspects such as dimensions, spacing, data type, window level, window width units, volume histogram as well as specifying some those properties for the output it will produce. Fig. 6.7 shows that the liver segmentation plug-in, namely "KMeans Algorithm", which has an embedded GUI, does not need seed points, uses whole volume data etc., and extends PluginFactory.

	Plug-in Property	Plug-in Value
1	*KEY_NAME*	KMeans Algorithm
2	*KEY_TYPE*	*VALUE_TYPE_SEG_JAVA*
3	*KEY_GUI_MODE*	*VALUE_GUI_MODE_EMBEDDED*
4	*KEY_GUI_ISVISIBLE INPLUGINMENU*	Boolean.*FALSE*
5	*KEY_SEGMENTATION_MEMORY_CONSUMPTION_FACTOR*	5
6	*KEY_SEGMENTATION_SEED_POINTS_NEEDED*	Boolean.*FALSE*
7	*KEY_SEGMENTATION_USE_WHOLE_DATASET_ALWAYS*	Boolean.*TRUE*
8	*KEY_SEGMENTATION_PREFERRED_RESULT_TYPE*	*VALUE_SEGMENTATION_PREFERRED_RESULT_TYPE_SURFACE_NOT_FIXED*

Figure 6.7 A plug-in (i.e. K-MeansPlugin) needs to extend PluginFactory and determine the plug in properties for initialization. The figure shows the parameters for the initialization 1) Name and 2) type of the plug-in 3-4) The plug-in can be shown as a panel in the 3-D view (like KMeans). Alternatively, a plug-in can open an own window (if there are a lot of parameters for example and a lot of space is needed for the GUI). The the plugin window can be activated in the "Plugin Menu bar". 5) For the automatic scaling of the data the host has to know how much memory one plug-in needs. This can not be detected automatically, since the host can not know the internal data structures of the plug-in (e. g. level sets are much more complex than region growing). So this value represents the factor of memory. A value of 8 means that the plug-in needs 8 bytes for 1 input byte (so it needs 8 times the memory of one data set). 6) Seed point usage 7) VOI or whole data set usage 8) Here, the result type can be set to VR or SR.

Plug-in class files can be placed in the 'plug-in' directory of host to be recognized. The format is always a Java class file, which implements a special Interface (so it can be recognized by host as a plug-in). When the user wants to start a segmentation process by creating a new object using the OM, a new instance of the corresponding plug-in is generated by the Plug-In Factory. This approach provides the possibility of using the same plug-in (with different objects) for the same data set or studying in parallel with

different datasets. Then, the visualization software registers the created instance and assigns an ID to it. An example is given in Fig. 6.8.

```
public class KMeansPluginInstance extends Plugin {
KMeansAlgorithm algKMeans= new KMeansAlgorithm();
KMeansGUI guiKMeans = new KMeansGUI();
public void init() {
  ExploreDicom.registerObject(algKMeans, getID());
   this.pluginPanel = new JPanel();
   this.pluginPanel.setLayout(new BorderLayout());
}
```

Figure 6.8 Next, the GUI of the plug-in function is called to handle the elements of the plug-in such as creating and updating the GUI and defining the properties of the output data.

There are some additional features that the plug-in architecture provides which significantly enhance what can be done through a plug-in. The architecture supports plug-ins to send the resulting data in binary form, in indexed form or with gray values that can be original or defined by the plug-in. Returning of resulting (multiple) objects is done with indexed data together with a lookup-table, which assigns each index value a color and opacity. This way the ITK algorithms has to be executed only once, even if several objects are created (e. g. liver segmentation: vasculature and parenchyma). Some plug-ins may produce surfaces (or meshes) instead of an output volume. A plug-in may require more than one volume of input data. This is common for many level set segmentation techniques that require a feature image as well as an initial segmentation in order to operate.

These processes are not dynamic but object based which allows the usage of same the technique and interface more than once. Construction of the GUI elements is also performed through a generic Application Programming Interface (API) to facilitate changes in the data. For instance, the range of a slider in the segmentation GUI is based on the actual scalar range of the current volume. If a new volume is loaded or an operation such as thresholding is applied, then the range of the scale will change to match the new volume.

In the segmentation process, the values of the GUI elements are typically used to control the processing. The GUI elements can be text fields, sliders or any other

interaction mechanisms. If the user interaction mechanisms (i.e. inserting and removing seed points) are needed to be used, then these buttons appear at the top left of the plug-in GUI.

The plug-in interface provides methods for exchanging data and parameters between the GUI and the algorithm implemented in ITK. For this, a C++ class was developed which makes use of Java Native Interface (JNI) (Liang, 1999). The developed ITK class extends an abstract basic class (belonging to the plug-in interface), which contains methods and data structures for the data exchange. Internally, a hash-table is used as data structure. From the Java (GUI) side, data can be transmitted by the method putParameter(String name, Object value), e. g. putParameter("UpperThreshold", 120.0). The ITK algorithm (written in C++) has access to this parameter by calling the method getParameter("UpperThreshold") in the same class. The plug-in interface also creates the necessary instance of itkImage, which is passed to the ITK algorithm. As a result, the algorithm outputs an instance of itkImage as well (if no segmentation is performed, the output corresponds to the input). The visualization (including the creation of VTK data structures) is completely done by the plug-in interface.

For the plug-ins implemented in Matlab, there are two ways for the data exchange: The first way is to build a standalone application and the second way is to convert a Matlab function into a library file (i.e. Java Archive - JAR) and call that function like native Java libraries. In the first way, the pre-processed data is written to a storage medium (i.e. hard disk) and the plug-in is executed with the necessary parameters. After the execution of the plug-in, the segmented data is again written to disk where the host visualization program reads it. The read-write operations are done in less than a millisecond and an instance of "Runtime" class are used to make Java (i.e. host visualization program) waits for the plug-in to execute. The advantage of this method is that the plug-in does not share the memory with host visualization program but it requires the installation of Matlab Compiler Runtime (MCR). The second way uses the Matlab Builder, by using which M-code functions from the Matlab can be wrapped into one or more Java classes. When deployed, each MATLAB function is encapsulated as a method of a Java class and can be invoked from within a Java application. However, to

do this, Matlab Builder and Java Runtime Environment must be the same versions (releases).

```
public void processCommandCall(Command command, String name, Object
argument) {
        if (argument == Command.CALL_SEGMENTATION_START) {

           // 1. Setting parameters (GUI -> algorithm).
           this.algKMeans.setKidneyStartValue(this.guiKMeans.getKidne
           yStartValue());

           this.algKMeans.setReferenceImageValue(this.guiKMeans.getRe
           ferenceImageValue());

           // 2. Initialize algorithm
           this.algKMeans.init();

           // 3. Start algorithm
           this.algKMeans.start((short[][])command.getData(Command.
           PROPERTY_VOI_IMAGE_DATA),

           command.getDataIntArray(Command.PROPERTY_VOI_IMAGE_EXTENT
           ));
       }
}
```

Figure 6.9 Setting the required parameters for the plug-in and starting it by sending the necessary data asked by the plug-in. When the user starts the segmentation by pressing 'Start' button in the user interface, the host program informs the plug-in (for this, the ID of the plug-in is important, see Fig. 8) that data is ready and waiting, together with segmentation parameters. The plug-in then collects data and parameter from the host program and then starts the segmentation process.

For the example plug-in, namely "K-Means Algorithm", the parameters to be defined prior to segmentation is the slice numbers where the kidney and liver have distant borders and the border of the liver does not intersect with the heart. Two text fields are used to get these parameters from the users and set as given in Fig. 6.9. After all segmentation parameters are set, the plug-in is initialized and the start( ) function is invoked with the start button when the user requests that the plug-in to be executed. This function processes the input volume to produce the resulting surface and/or output volume. The resulting volume appears on the 3-D panel and the plug-in is placed to the OM. If the user wants to remove an object (and corresponding segmentation result), "Delete" button can be used to destroy the plug-in. The plug-in then can do some cleaning work, like freeing memory (Fig. 6.10).

```
public void destroy() {
        ExploreDicom.unregisterObject(algKMeans);
        algKMeans = null;
        guiKMeans = null;
        if (this.pluginPanel != null) {
          this.pluginPanel.removeAll();
          this.pluginPanel = null;
      }
  }
```

Figure 6.10 Destruction of the plug-in.

6.4 Application to Medical Datasets

The capabilities of ITK, MatLab and Java based plug-ins for solving real medical image problems are demonstrated in this section with three challenging applications. The first application shows the advantages of plug-in communications where each step of the transplantation donor evaluation (i.e. liver segmentation, vasculature extraction, lobe determination) can be done in a cascaded way. The second application shows varying cases of tumor segmentation which necessitates the use of different methods, thus different plug-ins, for the same clinical problem. Finally, the last application presents how the aorta-graft visualization problem can be solved by applying different segmentation methods individually and visualizing their results together.

Here, it is worth to point that VR and SR objects can be displayed simultaneously, if desired. In other words, some segmented objects can be rendered with SR while others with VR at the same time. MIP is not possible on per-object-base, since it is camera-based which means that it affects all objects. But it is important that the graphics board supports multiple objects with different styles (i.e. VR, SR).

The segmentation accuracy of CT/MR imaging–based volumetric techniques depends on the type and quality of the imaging data, as well as on the segmentation algorithm used. The DICOM data for these experiments were provided from the PACS of Dokuz Eylül University, School of Medicine, Department of Radiology. Four plug-ins were prepared for the tests one of which is based on TF specification, two of which are ITK based general purpose methods and the last one is a Java/MatLab based algorithm for liver segmentation (i.e. named KMeans Algorithm). As previously stated

in Section 3.A, the selection of the algorithm to be used is done at the beginning of the process (Fig. 6.5 (a)).

6.4.1 Developed Plug-Ins

The first plug-in is designed for VOI supported TF specification based on volume histogram and/or gradient intensity map. The two ITK based plug-ins are named as Fast-Marching (FM) and Connected Threshold (CoT) (i.e. Region Growing) referring to the algorithms behind. ITK toolkit expands the possibilities with extensive segmentation, and image filtering techniques, but does not provide GUI or methods for visualizing data. Thus, established processes are used to integrate the power of ITK with the robustness of VTK for visualization. Developed and implemented in C++, ITK guarantees cross-platform support by relying on C-Make for the compilation and configuration process. By using this advantage, ITK wrapper for Java is used for integration.

FM plug-in uses the fast marching level set method which is described in (Adalsteinsson, Sethian, 1995) and (Sethian, 1996). It was selected for performing semi-automatic segmentation because it usually has a reasonable computational time. This method is implemented in ITK in the FastMarchingLevelSet- ImageFilter. The fast marching method makes the simplifying assumption that the contour evolves in a monotonic pattern. In terms of level sets, this means that only speed functions can be used, which assure, that the curve passes any pixel only once during the evolution process. Among several parameters, the adjustment of maximum gradient and the time threshold is chosen to be the two of the most intuitive and important parameters. Therefore, the GUI is designed with a slider bar controlling the maximum gradient and a field for “Time Threshold” selection which is a parameter only for limiting the execution time of the algorithm in case of a false gradient determination (Fig. 6.11 (a)). Since the method also need seed points, the buttons for inserting and removing seeds are automatically located at the top of the plug-in (Fig. 6.11 (a)).

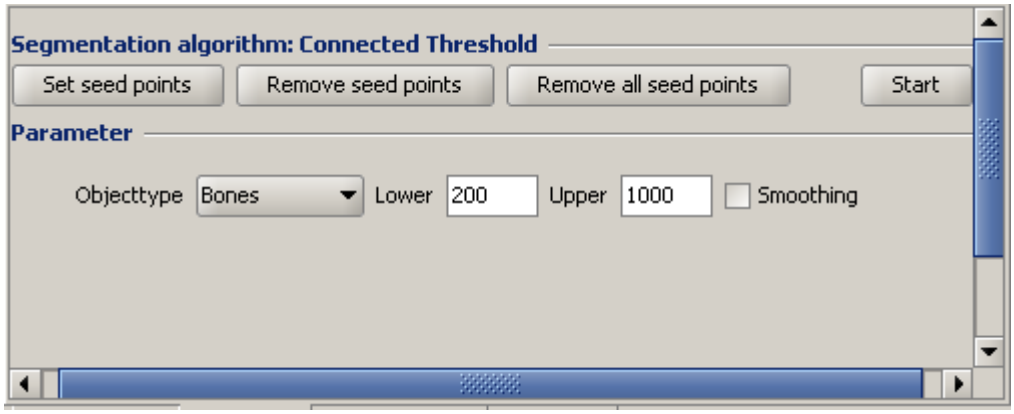
The second ITK based plug-in is the CoT, which is basically an implementation of well-known region-growing technique, starts from one or more seed points. The neighbors of the seed points are visited and the ones, which fulfill the criteria, are added

to the same region as its corresponding seed point. Pixels which satisfy the predicate of more than one region are added to one of them arbitrarily (Adams, 1994). The interface of the “Connected Threshold” plug-in consists of seed point buttons, upper and lower threshold fields which constitutes the selection criteria of the algorithm (Fig. 6.11 (b)).

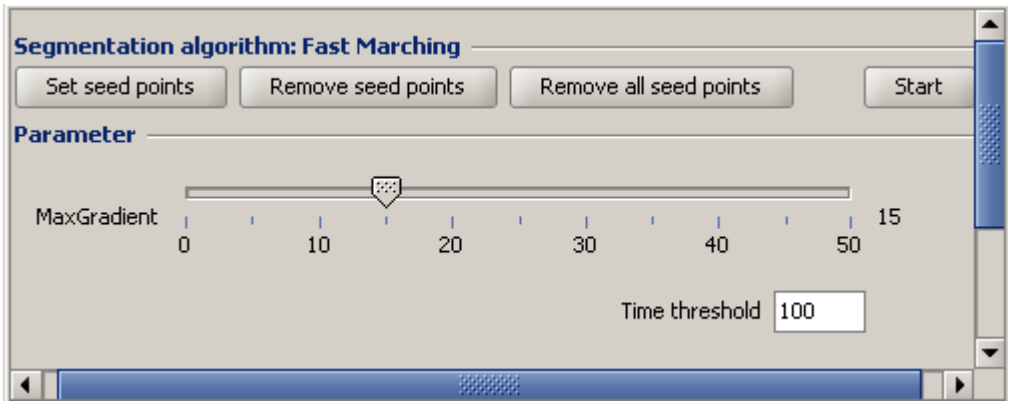
The fourth plug-in, namely K-Means plug-in, is based on the algorithm presented in (Selver et al., 2008) and developed specifically for segmenting liver from CTA images of liver transplantation donor candidates. The algorithm follows a pipeline that includes, pre-processing (i.e. removal of the right kidney, removal of the ribs and bones), classification and post-processing (i.e. morphological opening, skeletonization and connected component analysis). All steps are implemented in Java except removal of the bones and ribs which is implemented in MatLab due to computational efficiency and the difference in the time required. This kind of interactions left to the developer’s responsibility. As mentioned before, a plug-in can be Java, C++ or Matlab based. If it is C++ based the interface consists of an entry point of a DLL. From there, the developer can called ITK, Matlab or other functions. However, this is not directly supported by the host but this way the concept is more flexible and lets the developer decide, which toolbox to use.

The fifth plug-in, namely Hessian plug-in, is based on the algorithm presented in (Dogan, Dicle, & Guzelis, 2009) and developed for extracting vasculature of liver from segmented liver images or from series of CTA images. This is also a purely ITK based plug-in which uses Hessian based vessel filter for extracting liver vasculature and labeling the main vessels (hepatic and vein).

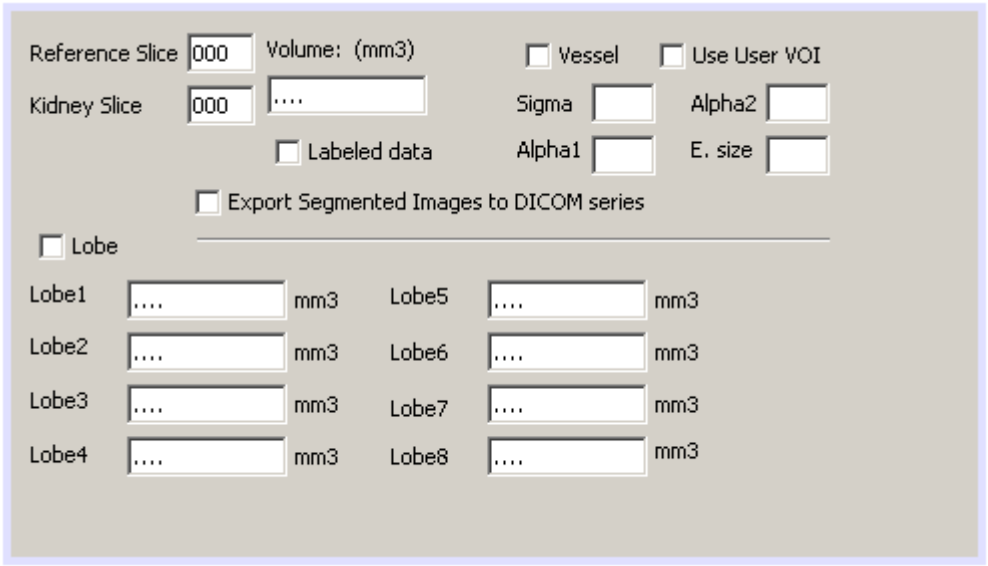
All of the applications presented in this section are done using Pentium 4 PCs with 4GB memory, 3GHz processor and a 256 MB graphics accelerator. No differences other than processing time and number of slices to process are observed in configurations between the base requirements (Section 6.2) and test configuration mentioned above. However, the operating system has an important effect on the size of the data to be processed since there is 2 GB RAM limit in Windows (up to 300 slices (512x512) in 16 bit) while Mac OS X and Linux do not have such memory limitations.



(a)



(b)



(c)

Figure 6.11 GUIs of (a) Fast-Marching plug-in (b) Connected Threshold plug-in (c) Liver segmentation plug-in.

6.4.2 Liver Segmentation Plug-in for Donor Evaluation

Living donated liver transplantation (Wachs et al., 1998) is a procedure where a healthy voluntary donor gives a part of his/her liver to another person. Measurement of the liver volume, analysis of the liver vasculature and determination of the liver sub-lobes are important stages to decide whether a candidate for transplantation is suitable or not. Thus, this application requires segmentation of three challenging problems where each of these problems should be initiated with the result of the previous one.

CTA is currently the most widely used technique for the rendering of liver parenchyma, vessels and tumors in living liver transplantation donors. Before 3-D rendering of the vasculature and the measurement of liver volume, accurate segmentation of the liver from surrounding tissues and organs is necessary.

Since the number of image slices used for 3-D rendering is very large, manual segmentation of the liver on each slice is time consuming and tedious. Also the results highly depend on the skill of the operator. Therefore an automatic segmentation procedure to segment the liver in all slices is needed.

Besides its several advantages over manual segmentation, automatic segmentation of the liver is very challenging and requires complicated algorithms (Selver et al., 2008). These challenges arise from the overlapping Hounsfield range of adjacent organs, from varying gray-level value range of organs due to the injection of contrast medium or different modality settings and varying anatomical structure/shape of the liver in different image slices and from patient to patient.

To cover all necessary steps of liver analysis, an exemplary pipeline is given in Fig. 6.13 (a). Starting from original CTA series; one can use a plug-in for liver segmentation (i.e. KMeans or Fast-Marching plug-in) that gives liver volume (segmented liver) as its output (Fig. 6.13 (b) and Fig. 6.13 (c)). Then, a plug-in for liver vasculature extraction (i.e. Hessian plug-in) can be initiated with previous output and can be used to obtain liver vasculature (Fig. 6.13 (d)). The results of two plug-ins can be rendered together (Fig. 6.13 (e)) or separately. If the data set is appropriate, the vasculature can be analyzed using only the result of segmentation plug-in and interacting with it via

transfer functions (Fig. 6.13 (f)). The same procedure (i.e. from segmented liver to liver vasculature) can be repeated for hessian plug-in (i.e. from segmented liver and liver vasculature to lobe determination), thus it can call a plug-in for lobe determination and initiate it with its output.

Each of these three plug-ins (i.e. segmentation, vasculature extraction and lobe determination) can also be used solely and independent of each other while using them together provides a more complete analysis. For instance, one can use the hessian plug-in for analysis of original CTA series and can obtain the vasculature of the liver. But it might probably cause mis-segmented objects (i.e. bronchus at lung) and increase processing time since all volume is included.

Considering the effectiveness of such an approach in liver analysis, the following conclusions can be made. In liver segmentation, FM plug-in should be used instead of CoT plug-in since it is determined to be the one that can successfully segment the complex borders of the liver. Even when FM is used, the volumetric measurements of the livers are found less than the correct volumes, although the visual appearances of the segmented livers are found out to be satisfying by the physicians.

A detailed analysis showed that, in most of the cases, the volume of the segmentation result appears to be smaller than the reference value. The internal analysis of the segmented liver shows that FM method does not include contrast enhanced liver vessels and some parts of the parenchyma, where contrast media leaks from the vessels, in segmentation result. As we see here, the challenges in liver segmentation require the usage of complicated algorithms designed especially for liver segmentation.

Since the proposed plug-in interface allows the integration of such specific and complicated methods, the K-Means plug-in (i.e. the fourth plug-in), which is based on the algorithm presented in (Selver et al., 2008), is integrated and used for segmenting the liver. The results obtained for this automatic method, which were around 90% success rate, are introduced in (Selver et al., 2008). All steps of the algorithm in (Selver et al., 2008) are implemented in Java except removal of the bones and ribs which is implemented in MatLab. The disadvantage of such a usage is the time required for file

exchange between MatLab and Java and the advantage is that MatLab does not share the same memory with Java, thus the application is not limited to 2 GB.

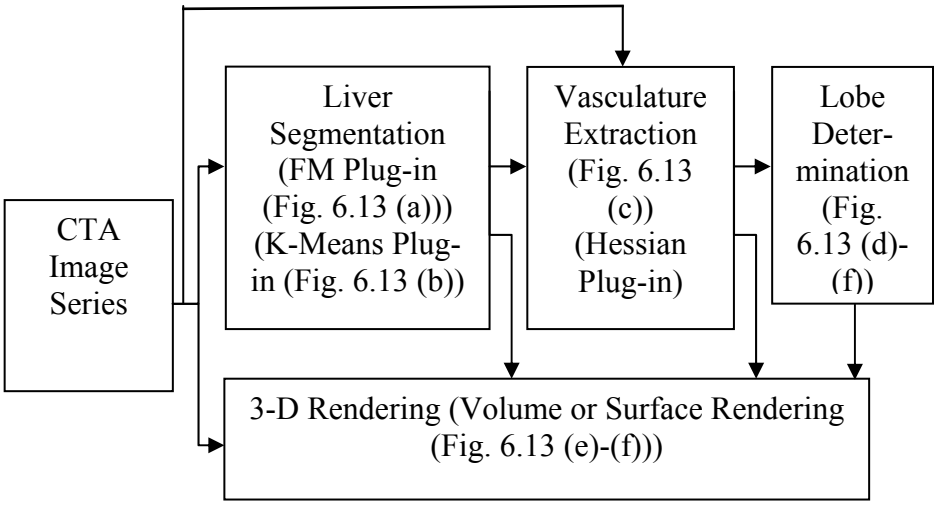
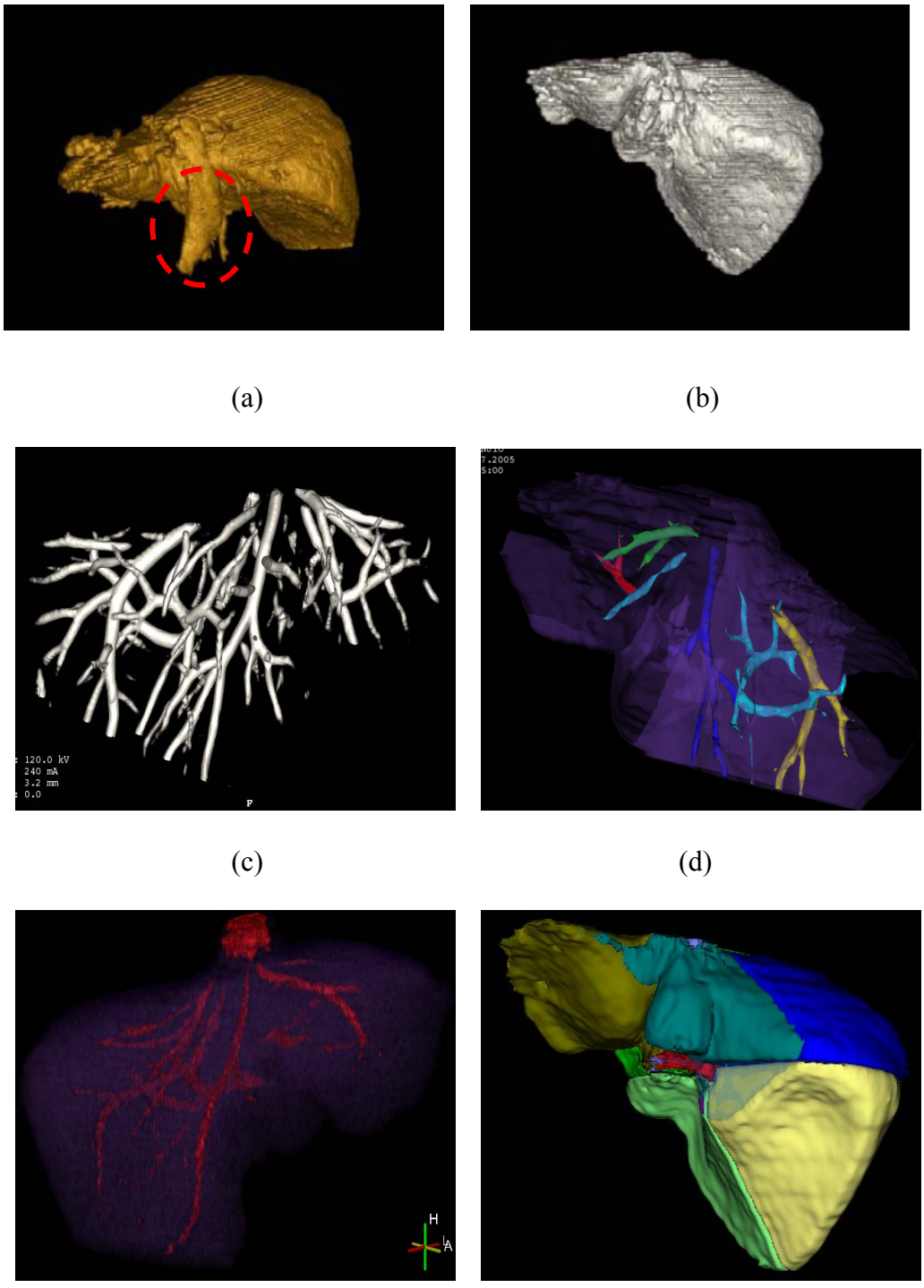


Figure 6.12 Liver analysis pipeline.

Other than success rates, the usage of automatic and semiautomatic plug-ins for liver segmentation can also be compared at the user level. When semi-automatic, general purpose plug-ins are used, the segmentation of the liver highly depends on the selection of the VOI and on the insertion of the seed points, since its position and shape change drastically along the slices of a series (Fig. 6.13 (b) and 6.13 (c)). Therefore, the user should search through the series for locating seed points and to make sure that the VOI covers the liver at all slices. The preparation of seed points and the VOI takes around 3 minutes and the FM plug-in runs around 4 minutes for an average of 90 images. On the other hand, K-Means plug-in works in a fully automatic manner and requires approximately 5 minutes.

As mentioned previously, the segmentation and the analysis of the liver can further be done using a second plug-in for vessel extraction by rendering only the vessel system (Fig. 6.13 (d)), the vessel system together with the liver parenchyma (i.e. indexed data to represent liver and different vessel systems) (Fig. 6.13 (e)), and the vessel system with lobes (Fig. 6.13.h). It is also possible to use TFs (i.e. pixel classification) for on-the-fly analysis of a segmented object at any step of the process (please compare Fig. 6.13.g and 13.h where lobes are rendered with only 2% opacity to allow vasculature visibility).



(a) (b)

(c) (d)

(e) (f)

Figure 6.13 (a) Fast-Marching plug-in result, inferior vena cava might be included (see the red circle) in the final rendering and/or inner vasculature vanishes if the maximum gradient value is not high enough, (c) The output of K-Means plug-in is more accurate since it is designed especially for liver segmentation, (d) The result of the ITK based vasculature extraction (i.e. hessian) plug-in that proceeds after K-Means plug-in, (e) Visualization of the

liver and its vasculature using indexed data, (f) Visualization of the liver and its vasculature using transfer function specification,(g) visualization of liver lobes and labeled vessels (i.e. hepatic, vein) together by rendering lobes as opaque,

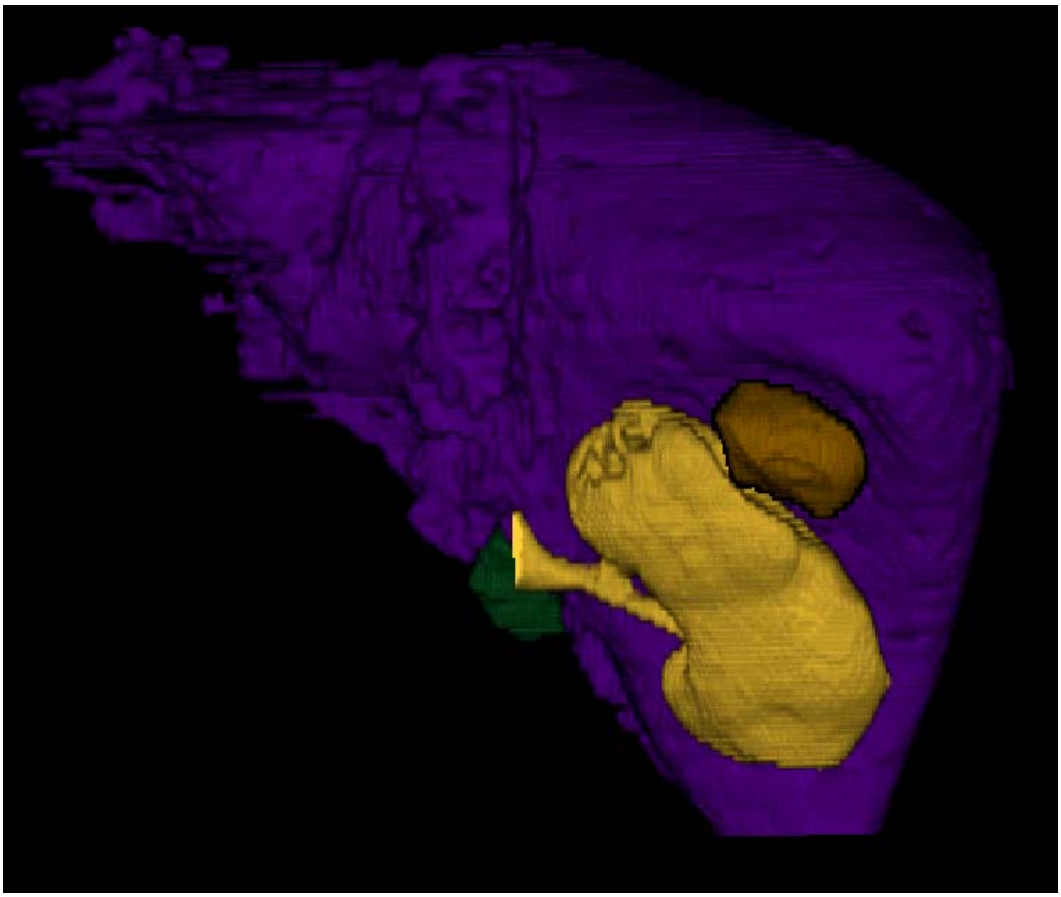


Figure 6.14 Visualization of abdominal organs and tissues together using different plug-ins together: Liver (K-Means plug-in, purple), Kidney (yellow) and gall bladder (green) (FM plug-in), Lesion (CoT plug-in, brown).

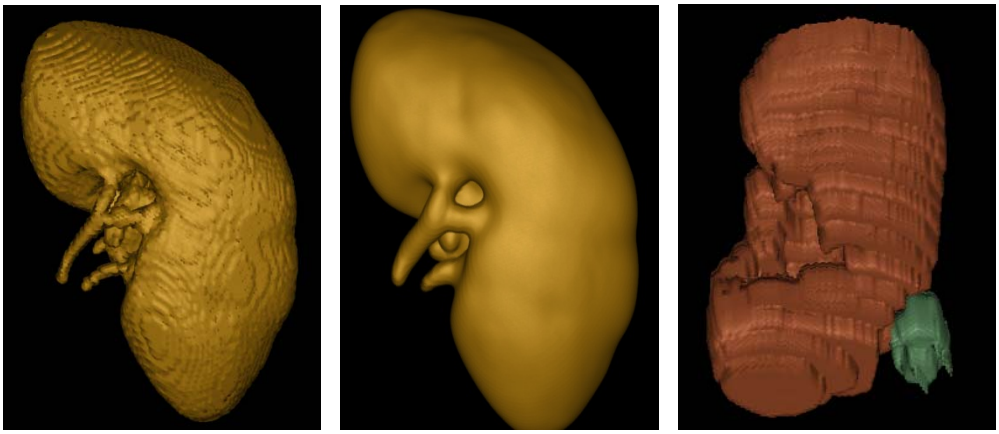
6.4.3 Segmentation of Kidney Tumors and Lesions

Determination of the volumes of abdominal solid organs and focal lesions/tumors has great importance in clinical analysis. Monitoring the response to therapies, the progression of diseases and preoperative examination of the patients are the most common clinical applications of tumor evaluation. CT and MR imaging all have been used for solid organ volumetry, which has been performed by using different measurement techniques and yielded variable results. Volumetric information about abdominal organs is not routinely generated for clinical use mainly because accurate,

reliable, and operationally practical segmentation algorithms are not readily available. Thus, the purpose of our study was to compare kidney lesion/tumor volumes calculated by using semi-automated segmentation techniques with the volumes calculated by using the standard manual contour-tracing technique.

Researchers in this field need a graphical interface with volume visualization features to quickly check the results of volume data processing. The developed framework has such an interface that allows it to be used as a visualization platform for evaluation of advanced image processing algorithms. In this work, the volume measurement algorithm involves voxel counting.

The study group has used all of the segmentation methods provided by the developed software. The plug-ins show satisfactory performances in the segmentation of kidneys (Fig. 6.15 (a)). Sometimes, the combinations of these methods are used also in segmentation of heterogeneous tissues. `Tumor and lesions` are generally segmented with FM plug-in except the cases in which the CoT plug-in is successful (usually the homogeneous lesions/tumors). Several pre and post processing operations (i.e. filters) has been used to improve the segmentation results (Fig. 6.15 (b)) and these processes are applied both to CT and MR series (Fig. 6.15 (c)).



(a) (b) (c)

Figure 6.15 User Interface of the developed software. The GUI layout is given in Fig. 6.2. The software is capable of displaying single images, image series, or other sequences. The 3-D rendering based visualization of DICOM series is integrated and supported by VOI selection, seed point insertion and segmentation plug-ins.

The results show that the homogeneous tumor and lesions can be segmented using one of the techniques offered by the program. The level of interaction and required time strongly depend on the application. However, a strong limitation arises in organ boundaries. When only a particular part of the organ's boundary overlaps with another organ/tissue, the gradient parameter itself is not enough. `Time threshold` also has a very limited role in the refinement of the problem. Then, increasing the gradient increases the overlapping problem and decreasing the gradient results with the loss of correctly segmented parts of the organ or tumor.

6.4.4 Abdominal Aortic Aneurysm and Graft Segmentation

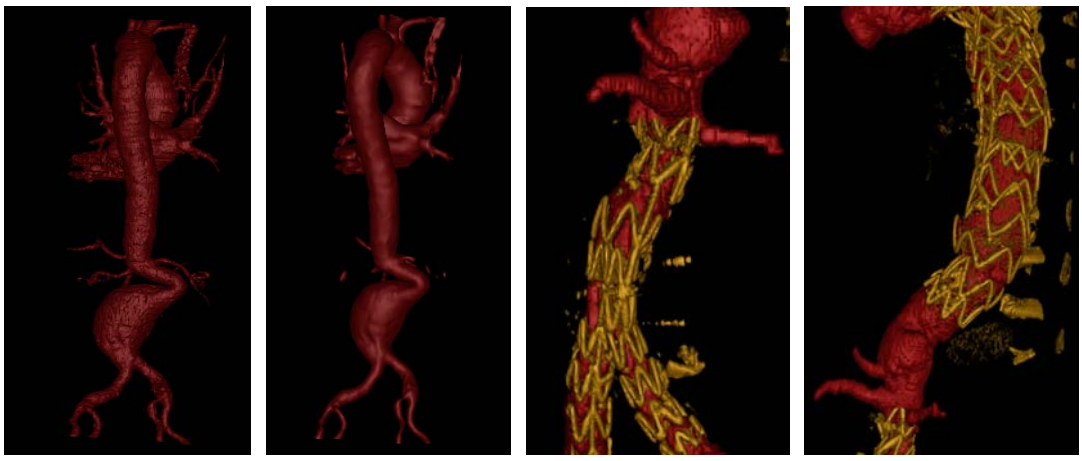
As mentioned before, AAA is a chronic degenerative disease with life threatening implications (Steinmetz, Buckley, & Thompson, 2003). One of the pathologic processes contributing to the changes observed in AAA is the depletion of vascular smooth muscle cells. This change results in progressive aortic dilatation accompanied by alterations in vessel geometry and redistribution of wall stresses.

A common treatment for AAA consists of the insertion of a vascular graft that creates a barrier between the blood flow and the weakened vascular wall of the aneurysms. Imaging the patient before and after the insertion of the graft is fundamental for controlling the efficacy of the procedure. The images considered here correspond to CT scans acquired after the graft has been implanted in which the network-like structure of the graft should be visible. The aim of this application is to illustrate how a non-trivial problem in medical image analysis can be solved by application of different segmentation methods together with application of the same method with different VOIs and parameters.

For the segmentation of aorta and aneurysm ITK based 3-D segmentation methods are used. Both, FM and CoT plug-ins produce acceptable results (Fig. 6.15 (a), 6.15 (b)) however, the result of CoT plug-in is found more realistic by the physicians due to the inclusion of small vascular trees. TF plug-in can also produce useful results in segmentation of aorta, but their use is limited because at some parts of the volume, the VOI cannot be adjusted enough to separate vertebra and aorta which have overlapping Hounsfield range.

Due to its network like structure and overlapping Hounsfield range, the graft is very hard to segment by using segmentation methods. Therefore, graft material is visualized by several TF plug-ins instead of using a single segmentation method. By using different TF objects, the developed software allows the selection of different VOIs for each TF (Fig. 6.15 (c), 6.15 (d)). Thus, the graft material is visualized by four different TFs with different VOI selections each of which is used to render a part of the graft without rendering the bones and the ribs (Fig. 6.15 (c), 6.15 (d)). All of the objects (i.e. plug-ins), which are used to segment aorta, aneurysm and graft, are listed in the OM.

The time required for the insertion of the seed points FM and CoT is around 2 minutes since the insertion should be done in several slices to prevent mis-segmentation of thin vessels. At the slices with thin vessels, the seed points are inserted with the help of zoom functionality. The required time for both algorithms to segment the aorta is also around 2 minutes. The only parameters to define in CoT are the upper-lower thresholds and gradient in FM which are clear due to the known intensity range of contrast media. The TF objects, which are used to render the graft, have almost the same functions (i.e. trapezoids) and ranges. But the VOIs, which they are applied, are different to represent different parts of the graft. The adjustment of the four TF objects takes around 15 minutes and the rendering is completed approximately in 20 minutes.



(a) (b) (c) (d)

Figure 6.15 Segmentation of aorta using (a) Connected Threshold, (b) Fast-Marching plug-ins. Segmentation of the graft using a VOI and transfer function for visualizing the graft (c) at the bottom half (d) at the upper half of the aorta.

This application demonstrates that the 3-D rendering of organs/tissues of interest are not only a matter of segmentation (i.e. CoT or FM for representing aorta), which prepares the interested part of the volume data for visualization, but also depend on visualization techniques and their adjustable/interactive parameters (i.e. multiple TF specification for graft visualization). These conclusions agree with the studies (Tiede, 1990, Schiemann, 1992) which are based on implementation of combined interactive segmentation and visualization tools and claim that visualization and more especially volume rendering can still provide information about the scene even if segmentation reaches its limits. Thus, aortic aneurysm with graft insertion provides a good demonstration of mixing the segmentation and the visualization capabilities.

CHAPTER SEVEN
DISCUSSIONS AND CONCLUSIONS

In this thesis, three novel studies on abdominal image processing and visualization are presented. The first study is the development of a segmentation algorithm for pre-evaluation of liver transplantation donor candidates based on their CTA images. The second is on the enhancement of the visualization of abdominal organs by introducing a new domain and a technique for multi-stage approximation to this domain which is then used for transfer function specification. Finally, the third one is on the implementation of the liver segmentation plug-in that is developed for the first topic (i.e. liver segmentation). The following sections of this chapter cover discussions and conclusions on these three studies, details of which are introduced in previous chapters.

7.1 Discussions and Conclusions on Liver Segmentation

In Chapter 3, a robust and efficient method, which can automatically segment the livers of transplantation donor candidates in any CTA series, is proposed. The success rate is calculated as %94.91 over a data set of diverse CTA series of 20 patients according to the evaluation of the expert radiologist experienced on pre-evaluation of transplantation donors for more than 100 cases.

The robustness of the method follows from its capability of dealing with the contrast variations and atypical liver shapes and this capability is provided by the patient oriented structure of the algorithm. For qualifying ‘patient oriented’, the algorithm learns the data set characteristics in parallel to segmentation process, and adapts its parameters to these characteristics. This strategy involves a segmentation method which does not utilize a common parameter set found from all patient datasets. Instead, the method is capable of adapting the parameter set to each patient. So the wide ranges of the parameter values are covered and the developed system is sensitive to all variations in a data set by adopting its parameters due to the data set characteristics.

The ability of dealing with the contrast variations and atypical liver shapes is provided first by recognizing the existence of these problems and then by solving the

segmentation problem using inter-slice information provided by the distance transform. This ability is gained:

i) by introducing the distance transform as a feature for each slice and then using this information in the succeeding slice to reveal three dimensional properties of the liver which can not be obtained by the set of slices processed individually,

ii) by devoting different (MLP) classifiers for different slices each of which is fed by three features such as mean, standard deviation and distance transform as opposed to the automatic organ segmentation methods available in the literature which use a single classifier for the whole set of slices, some uses statistical features extracted from three dimensional data (Koss et al., 1999) and some uses five features for each slice (Husain, & Shigeru, 2000),

iii) by reducing the number of features and by initializing each (MLP) classifier's weights with the weights of the previous one, so getting a good efficiency in terms of time and memory requirements,

iv) firstly by segmenting the initial slice in an unsupervised way, secondly by using the segmented image as the target in the (supervised) training of the classifier devoted to the initial slice, and finally by segmenting each slice in a supervised way with its associated classifier whose weights are obtained via training the classifier of the preceding slice which uses the segmented image of that slice as the target. (Such an approach makes the design of the overall classification system fully unsupervised that depends on the given CTA series only without requiring any given training set of CTA series. This is a very interesting feature of the overall classification system preventing the generalization errors originated from the dependence of the classifiers' performance on the used training set of CTA series.)

Results show that several problems in liver segmentation are addressed including gray level value similarity of adjacent organs, partial volume effects, atypical liver shapes and different modality settings. The method's ability of adaptation to data set characteristics increases the tolerance capability of the system and makes it feasible for clinical usage.

Although some of them produce very effective results in CT series, the deformable model based and gray level value based techniques generally produce segmentation results with holes inside the liver volume in CTA series even when the outer border is found correctly, because of the fact that the internal structure of the liver is acquired heterogeneous due to contrast media injection in CTA series. This results with incorrect measurement of the liver volume which is also handled properly with the proposed method.

The proposed algorithm has also been applied to the datasets provided in (Van Ginneken, Heimann, & Styner, 2007). Although these datasets are acquired with CT, the series obtained for patient datasets with no tumors are segmented with high performance (Fig. 7.1). This result also shows the benefit of patient oriented approach which is affected minimally from modality settings and does not need a training set prior to the application. In the datasets with tumors, it is observed that the segmented area generally includes the liver without including the tumor area (Fig. 7.2). This is an expected result since the proposed algorithm is designed to segment healthy liver parenchyma for the evaluation of transplantation donors who should not have any tumors in their liver.

Together with the datasets provided in (Van Ginneken, Heimann, & Styner, 2007) and in (Rosset, Spadola, & Ratib, 2004), the proposed method has been tested with the CT and CTA series acquired from four different modalities. The successful results obtained by all these modalities also show that the proposed method does not have dependence on the modality.

The disadvantage of the proposed method is its dependency to the correct segmentation of the ‘initial image’. If the automatically selected initial image does not satisfy the necessary requirements, the user should select an appropriate slice or might need to manually segment the initial image.

Since the algorithm is developed for the pre-evaluation of the transplantation donor candidates, the series that are acquired with rotated patient position (i.e. datasets 1 and 2 in (Van Ginneken, Heimann, & Styner, 2007)) can not be segmented.

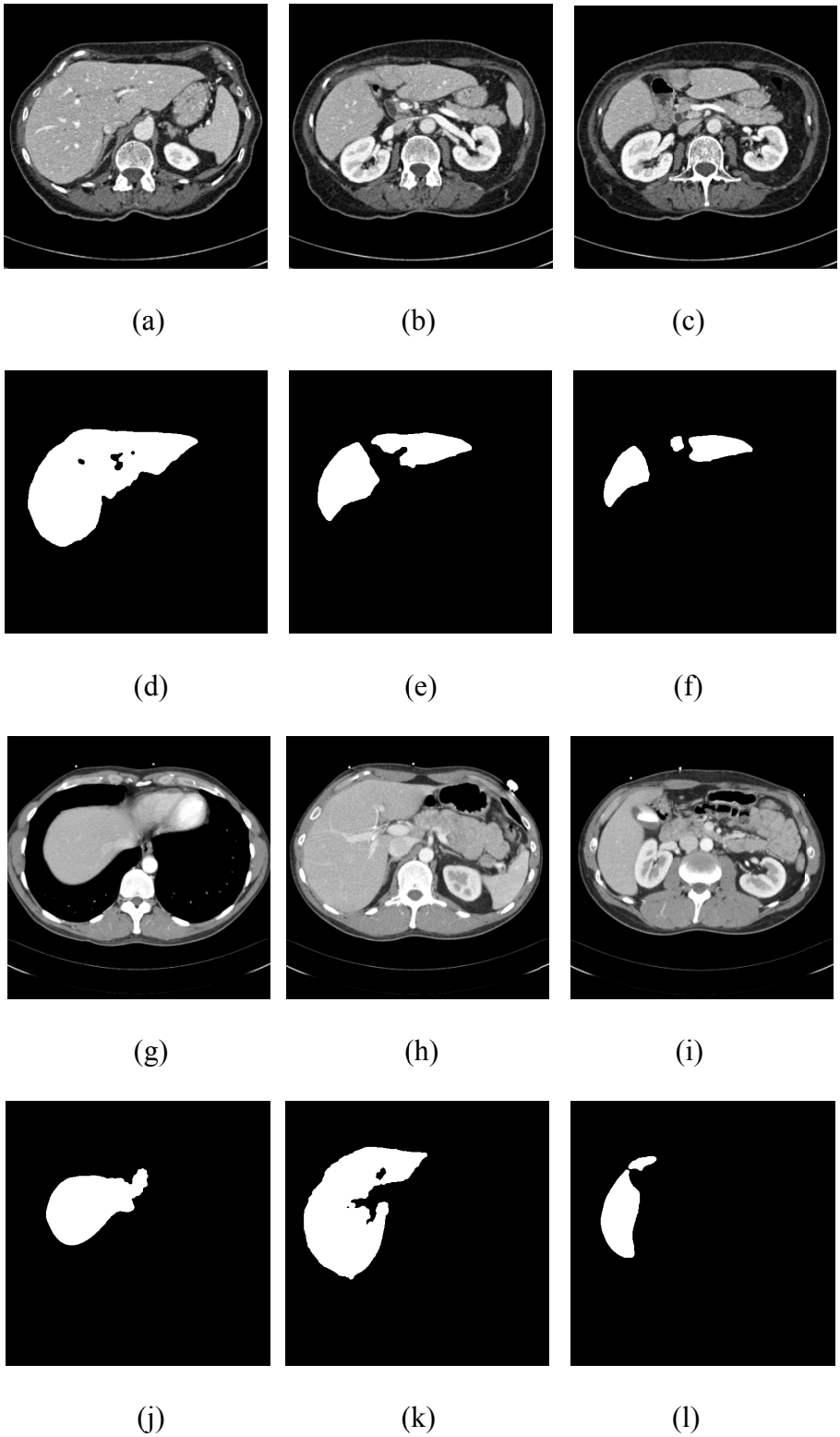
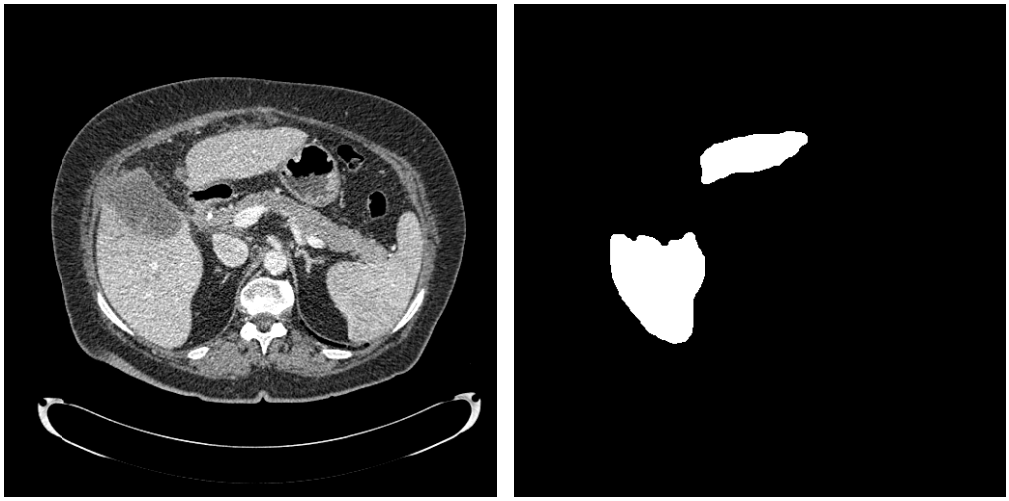
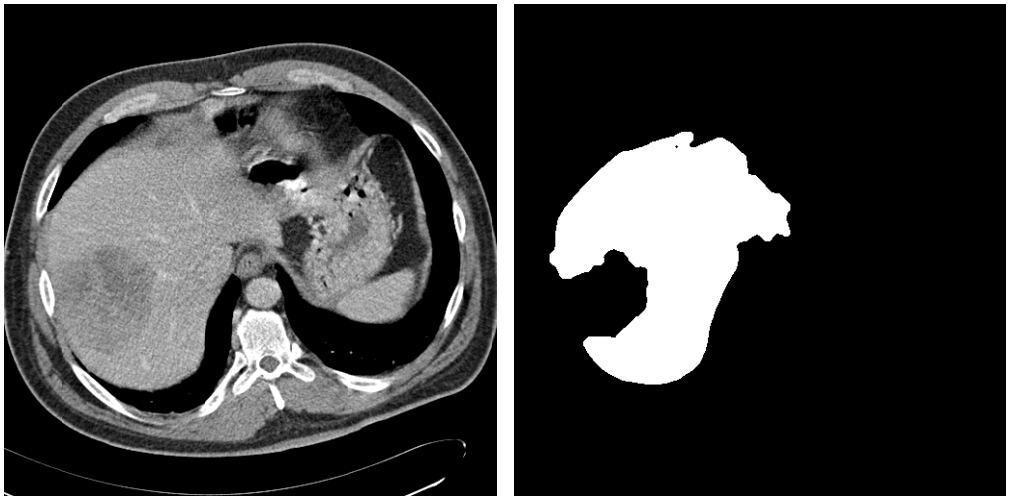


Figure 7.1 (a-c) Sample images from data set 7, (d-f) Segmentation results for the images from data set 7, (g-i) Sample images from data set 8, (j-l) Segmentation results for the images from data set 8.



(c) (b)



(c) (d)

Figure 7.2 Segmentation result samples for the datasets 3 and 4. In the datasets with tumors, it is observed that the segmented area generally includes the liver without including the tumor area. This is an expected result since the proposed algorithm is designed for the evaluation of transplantation donor who should not have any tumors in their liver.

Application of the proposed method to angiographic MR data sets of the transplantation donors have also been tested with minor modifications. These modifications include

i) Adjustments in sizes of morphological filters since the size of MR images (i.e. 256x256) are half of the CT images (i.e. 512x512).

ii) Adjustments histogram analysis and threshold values.

After these minor modifications, the developed method has been applied to MRA series without any other change. Even with this minor set of modifications, the segmentation results are found promising (Fig. 7.3). Thus, the same method can be modified and used for the segmentation of liver from MRA series if the following challenges are handled carefully.

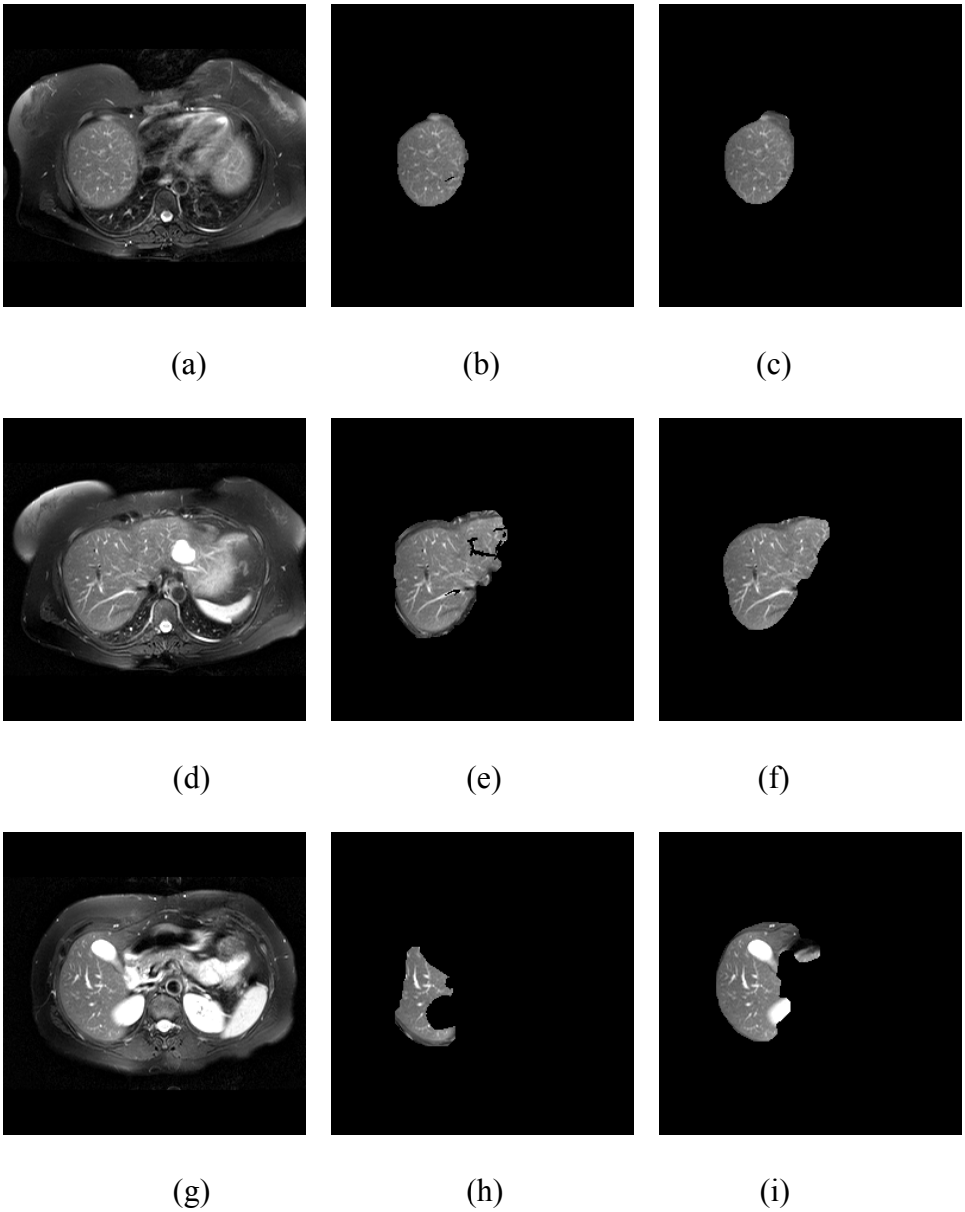


Figure 7.3 (a) Initial slice #21, Segmented initial slice using (b) K-Means (c) MLP, Sample images from data set 7, (d) Slice #18, Segmentation slice using (e) K-Means (f) MLP, (g) Slice #25, Segmentation slice using (h) K-Means (i) MLP.

First of all, it is observed that the performance of the algorithm is much more dependent on the selection of initial slice when compared to its importance in CTA series. Also, the increase in slice thickness (i.e. 6-10 mm) sometimes results with drastic changes in appearances (i.e. size and shape) of liver between adjacent slices. Such changes can not be represented completely by three features (i.e. mean, standard deviation, distance transform), therefore more features and/or series with less slice thickness might be used to increase segmentation accuracy. Also the pre and post processing steps should be revisited since the organ boundaries and parenchyma of the liver is not as separable as they are in CTA series. Thus, anisotropic filter etc. can be used to improve the smoothness of the liver while preserving its boundary.

Areas that may further be examined include speeding up the process by improving the programming structure and supporting the system with hardware accelerators or DSP boards, more robust detection of the 'initial image' and the 'initial kidney image'.

7.2 Discussions and Conclusions on Transfer Function Initialization

Volume rendering would be used more often in clinical practice if the complexity of interaction (i.e. setting a TF for volume rendering) becomes less. To reduce the complexity of TF design, a semi-automatic method for TF initialization and a new, effective and interactive domain for TF optimization is introduced in this thesis. The proposed method is based on a volume histogram stack, i.e. VHS, instead of conventional volume histogram and handles TF specification as a (vector-valued) function approximation problem where the domain is the 2-D input space of Hounsfield value and slice number and the range variables are opacity and color. The method automates and simplifies the optimization of a TF.

The newly introduced VHS data model allows the detection of tissues both in calibrated (i.e. CT) and uncalibrated (i.e. MR) medical datasets. As a consequence of the fact that each slice histogram is represented separately, VHS preserves inter-slice spatial domain knowledge, so it exploits more priori information. It also demonstrates changes in the gray values through the series of slices, thus including information on local histogram distributions of the tissues. In other words, VHS can represent the intensity

values of the tissues as well as their spatial information and local distributions which are not available in conventional volume histograms.

Moreover, VHS can be constructed not only for the major slicing axis, but also for an arbitrary axis. As illustrated in Figure 5.5 and corresponding application, the VHS can be generated based on the organ to be visualized and a suitable axis to differentiate it from the other organs. This provides an independency from slicing axis and ability to distinguish structures which are separated both x-, y- and z-dimensions.

Here, it is worth to point that this method can further be generalized in such a way that the VHS may be calculated on arbitrary aligned axis (i.e. a slice plane with an arbitrary normal vector, not aligned with x, y or z as given in (7.1)) which can be obtained with the help of not only MPR (as presented in Fig. 5.4) but also of Curved MPR or Oblique sectioning techniques. More general, 2-D VHS can be constructed by using 2-D surfaces as expanding the current approach (i.e. orthogonal directions of slice planes (Fig. 4.1 (a)) defined by $x=\delta$, $y=\delta$ or $z=\delta$) with more general surface equations $\varphi(x, y, z) = 0$ (Fig. 7.4 (a)). In particular, nested diamond shape surfaces (Fig. 7.4 (b) for a 1D version) are obtained by

$$w_1.x + w_2.y + w_3.z = \delta \tag{7.1}$$

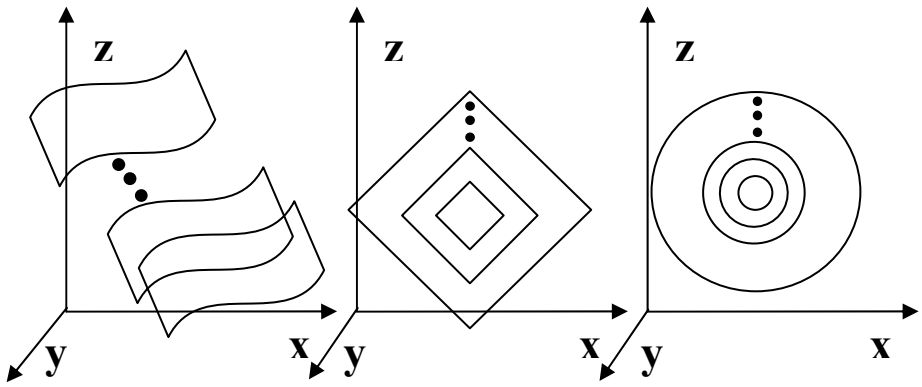
for different values and nested spherical shape surfaces around a specified center point (Figure 7.4 (c) for 1D version) are obtained by

$$\left(\frac{x-c_x}{d_x}\right)^2 + \left(\frac{y-c_y}{d_y}\right)^2 + \left(\frac{x-c_z}{d_z}\right)^2 = R^2 \tag{7.2}$$

for different R values.

The geometry to generate the VHS data can be selected depending on the nature of the organ to be visualized. From mathematical point of view, arbitrary surfaces should be simple, closed and bounded. Selecting x, y and z as continuous variables would require interpolation to determine the pixel coordinate values among existing ones. It

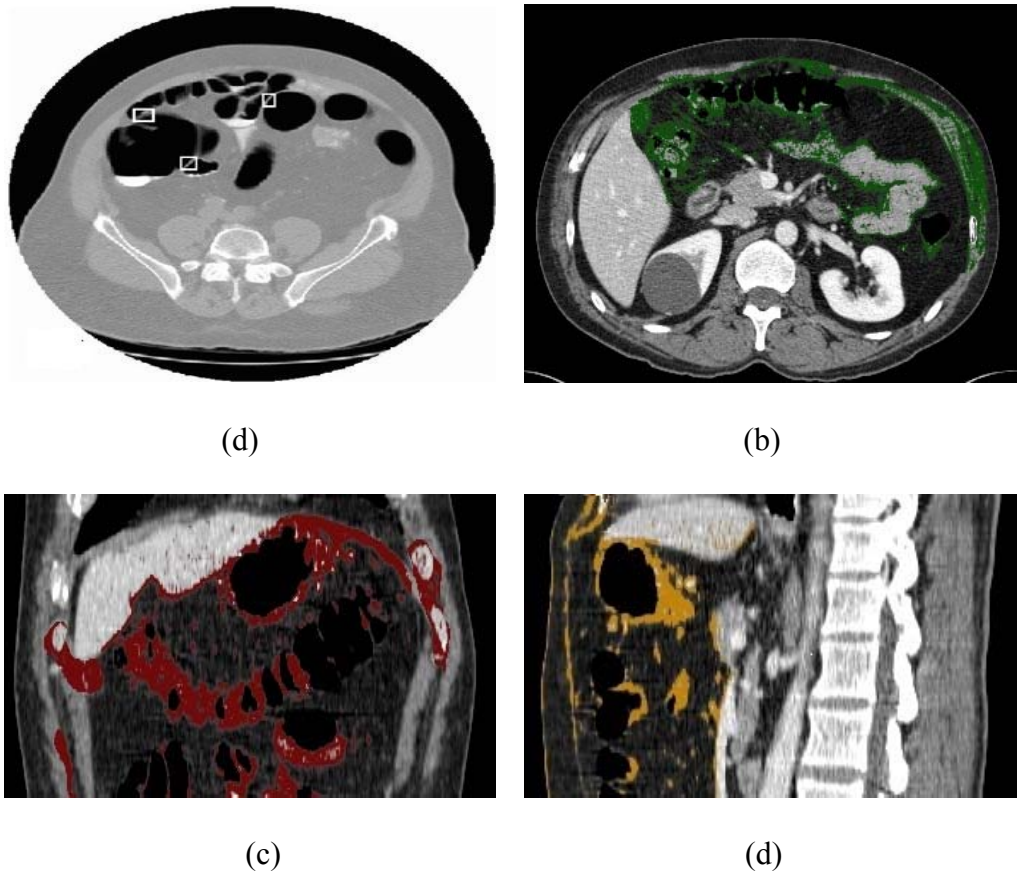
should be noted that selecting these parameters as integers and surfaces such as in (7.2) does not need interpolation.



(a) (b) (c)

Figure 7.4 Examples of (a) arbitrary surfaces (b) diamond shaped, (c) spherical surfaces that can also be used in future studies to generate VHS data (Diamond and spherical surfaces are illustrated in 1-D for simplicity).

Although extending the implementation to work along an arbitrary axis or with an arbitrary surface would pro-vide many advantages, it would still be not enough to address the studies where the object of interest does not follow a single axis such as intestines. Simulations show that the main problem of the proposed method in transfer function generation for intestines, which is filled with air, is the presence of adjacent organs in all directions of abdominal region that are also filled with air (i.e. lungs, stomach and small intestine) and/or have similar HU value range, thus faking the lobes belonging to the intestines. Consequently, selecting one axis can only differentiate some of the overlapping tissues while others remain (Fig. 7.5). In this case, using a single axis is not enough and our approach can further be extended in such a way that different approximations (i.e. a number of VHS data generated using different directions) are combined with a method to produce a better differentiation of organs. Depending on the application, this method can be on collecting the results obtained for each VHS or rendering only the areas that are found in all VHS.



(d) (b)

(c) (d)

Figure 7.5 CT image examples for visualization of intestines (a) image from (Bakker et al. 1999) shows a typical 2-D CT slice. The rectangular frames show thin walls with air on both sides. Moreover, colon can be buried in the residue fluid and some parts might be collapsed in sigmoid colon, (b) Axial image (original slicing axis), the outer walls of intestines with stomach both of which are filled with air. Selecting more units results in overlapping with spleen, (c) Reconstructed coronal images where the overlapping with has been solved but intestines and stomach can not be differentiated, (d) Reconstructed sagittal images where the same affect still exists.

The proposed improvements on arbitrary selection of surfaces and combination of the results obtained by different approximations would push the task of picking the axis even more to the user. Two possible methods to overcome this drawback can be implemented to guide the user in picking an appropriate axis. The first method would be the construction of a look-up table providing the information to the user about which axis is the best for a given organ and a study type (i.e. CT, MR, Angio etc.). The construction of a reliable look-up table requires an extensive study that should be carried out in close co-operation with expert physicians. The second method would be based on automatic selection of an appropriate axis among several arbitrary axes not only

including x-, y-, z- axes, which are considered in the current paper, but also including the rotated ones of them with a determined angle (i.e. 45, 22.5 degrees) and even including any direction defined by a plane as its normal vector. The selection here might consider a predefined criterion such as choosing the one with the highest number of lobes in a specified HU value range.

An approximation to VHS data done by SEG-HRBFN detects the tissues to be visualized and provides an informative spectral representation for them in terms of centers and widths of the Gaussians associated to the lobes of these tissues. The hierarchical learning (design) strategy that is carried out with SEG-HRBFN allows the recognition of suppressed lobes corresponding to sup suppressed tissues and the representation of overlapping regions where the intensity ranges of the tissues overlap. Unlike the previous HRBFN models, SEG-HRBFN determines the number of necessary Gaussian units, their locations and widths, in an automatic way. This approach significantly reduces the number of Gaussian units produced at the end of approximation, so allowing user interaction at the optimization level. Hence, the physicians work on a spectral domain so they deal with the width and position of the Gaussian distributions (basis functions) instead of dealing with a complex function itself. Approximation to VHS with a minimum set of basis functions (i.e. Gaussian units) also provides the construction of the network in a reasonable time.

The effectiveness of the VHS data is demonstrated in the paper via different abdominal studies where rendering results show that it generates a proper multidimensional space for clear classification of the tissues that overlap in conventional volume histograms. It is shown in the paper that SEG-HRBFN constructs an effective initial TF representing the feature space and also that SEG-HRBFN is very successful also in the reconstruction of 3-D range data, such as the data obtained through the auto-scan system from a baby doll face (Ferrari, Maggioni, & Borghese, 2004).

All of the applications presented in this paper are done in a Pentium 4 PC with 4GB memory, 3GHz processor and a 256 MB graphics accelerator. The smallest data set is the fourth application (i.e. 92 images) and the biggest data set is the first application (i.e. 326 images). The time required for the initial generation of the VHS data is between

0.91 and 3.26 seconds for all of these four applications. No preprocessing is done before the generation of the VHS data, however a smoothing filter can be used or the peak detection kernel can be enlarged to prevent the false detection of the spikes instead of lobe maxima. Also, the range of the VHS data can be reduced to a range which covers the all density distribution of the organ(s) to be visualized.

As demonstrated in the comparison with HRBFN (See Table 5.1), SEG-HRBFN requires more time but needs less number of units to reach the same approximation. However, the required number of units is much less for TF specification application. The time required for the generation of sufficient number of units depends on the number of lobes as well as the number of determined overlapping regions and suppressed lobes, which in-crease the required number of layers, rather than the size of the data set. The application of SEG-HRBFN results with 45, 36, 51 and 49 units for the four applications presented in this paper, respectively. The required time for the generation of these units is 28, 19, 35 and 42 seconds respectively. It is worth to point that these times are required for once at the initial calculation made by SEG-HRBFN.

The algorithmic complexity difference between HRBFN and SEG-HRBFN mainly comes from two points. The first one is the determination of the centers and widths of the Gaussians. In HRBFN, the center locations and widths are pre-determined and need no other computation. However, SEG-HRBFN requires a peak detection algorithm based on gradient calculation to determine the centers. The second difference comes from the number of units used for approximation. The HRBFN inserts fixed Gaussians at each layer producing a high number of units. On the other hand, SEG-HRBFN, which usually has around four times less units, uses a gradient descent based weight determination algorithm to determine the heights of the Gaussian basis.

Later on, the useful units among all are selected and adjusted by the user (i.e. the physician). This selection and adjustments (i.e. position/width of the units, grouping, opacity and color selection) take around 10 to 15 minutes before completing the adjustment of the TF. Considering the time required to segment challenging organs, such as liver, manually or with semi-automatic/automatic segmentation methods (Selver

et al., 2008), the time reported in this study can be found acceptable from the clinical point of view.

The overall goal of this part of the thesis is to improve the rendering quality for visualizing the tissues of overlapping intensities and also to shorten the physician controlled optimization stage in TF design. The proposed method is capable of representing the Gaussian like lobes in a histogram very efficiently and it also provides a good framework for the physicians to optimize TF in an interactive way. This ability makes the method promising especially for abdominal studies because the tissues of interest in abdominal images are large enough in a number of slices, so producing Gaussian like lobes in volume histogram stack.

The selection of the Gaussian units produced by SEG-HRBFN to construct groups corresponding to the tissues is the only stage which is not held in a fully automatic way. The grouping in the proposed method is left to the physician since the optimality of initial TF is subjective which differs from one physician to another and depends on the physician's interest in a practical application. It is possible to perform grouping stage in an automatic way if objective criteria for optimal initial TF are determined by a future work surely in collaboration with a group of radiologists, surgeons and other physicians.

7.3 Discussions and Conclusions on Programming and Implementation

According to reasons mentioned in Chapter 6, the goal of this part of the study is to develop a plug-ins for the techniques presented in Chapter 3 and 4. Thus, chapter 6 introduced the proposed interface via the plug-in that is developed for the liver segmentation algorithm proposed in Chapter 3. This plug-in program is implemented in an object based segmentation frame that aid in the development, deployment, and refinement of segmentation algorithms to benefit heath care delivery, education, and research.

The method for achieving this aid is to provide access to the 3-D rendering (i.e. volume rendering, surface rendering and MIP) capabilities that can be used to visualize the results of new segmentation algorithms. By doing so, it benefits practitioners by allowing them to make use of their advanced algorithms developed by different tools

(i.e. MatLab, Java, ITK) with a low learning curve and it can assist algorithm developers by proving a simple application. Thus, the developers are enabled to easily and routinely make use of their algorithms with little to no learning curve from within a DICOM application. As opposed to direct use of the ITK and Java, researchers do not need to deal or spend time to gain programming experience on loading data, displaying images or showing the results in a proper way which requires a high experience on VTK and Java due to the due to various cases of DICOM format and different medical applications. Compared to other visualization software, which can be used to integrate segmentation plug-ins, the developed software has the following advantages. First of all, it is specialized for medical DICOM images. That means the software supports decoding 12-bit DICOM images of all flavors, rotates the volume data to the x-, y- and z-axis if necessary, provides support for setting markers in three multi-planar planes, which can be defined arbitrarily by the user. The Graphical User Interface (GUI) is designed like in typical medical workstations with which the user (i.e. radiologist) is already familiar. Loading and displaying the 2d-data is done automatically, thus the user starts directly in the 3d-layout. The data set can be explored immediately by defining a TF (Selver et al., 2007) (or selecting among the predefined ones). Currently, the number of plug-ins is limited however; all the elements required for creating new plug-ins are available. Creation of plug-ins for new medical image processing applications is possible without having to expose the source code of the main visualization software.

The segmentation follows the typical visualization pipeline, consisting of the modules ‘Pre-processing’, ‘Segmentation’, ‘Post-processing’ and ‘Rendering’. The module ‘Segmentation’ can be provided by the user with a plug-in. Thus, the direct link between the object manager used for visualization and the segmentation interface used to obtain a specific object. Therefore, the advantage of the proposed method is to drive the segmentation from a visualization viewpoint.

The application of the developed plug-ins to different medical datasets show that the object based segmentation approach is suitable for measurements such as the organ volumes as well as for observing anatomical and/or physiological examinations using suitable algorithms for each organ/tissue of interest. In this manner, different plug-ins can be used one after another (or the same plug-in can be used for many times) to

segment and then render several organs and tissues together (Figure 6.14). This is a very important advantage of the proposed approach since in most of the cases, what is interesting from clinical point of view is not only an organ or tissue itself but its properties together with adjacent organs or related vessel systems. Therefore, the usage of different segmentation methods in one application and then a proper combination and representation of the results together provides an informative rendering.

By taking the advantages of the object based segmentation approach, a plug-in is developed for the segmentation of liver from CTA images. As presented in Chapter 6, the developed plug-in can be used for automatic calculation of the liver volume. If automatic result is not good enough, user can use the GUI of the plug-in to adjust some parameters (i.e. defining initial kidney image, reference image etc.) which would possibly increase the performance. The developed plug-in can also be used as an element of a more general system that consists of other plug-ins which are followed by the developed plug-in and initialized by the resulting segmented data of the developed plug-in. which can extract the vasculature and determine the individual volumes of the lobes of the liver.

The programming and integration to a host medical visualization software of the implemented plug-in is done according to the steps presented in Chapter 6. After using several programming techniques that are applied for improving the performance of the plug-in, the resulting program completes the segmentation process in 4.5 to 6 minutes and requires 600 MB memory for series that 80 to 100 images. For the series that have 100 to 130 images, the software requires 1 GB of memory and completes the process in 6.5 to 8.5 minutes.

Manual manipulation of the segmentation results, removal of selected object or preserving selected object and removal of the others are necessary functions that should be provided in the future. This is due to the fact that even in a very efficient segmentation result, a minor part might be wanted to be modified by the physician.

REFERENCES

A Free Scientific Software Package, Scilab, (n.d.) Retrieved September, 5, 2008 from http://www.scilab.org.

Adalsteinsson, D., Sethian, J.A. (1995) A fast level set method for propagating interfaces. *Journal of Computational Physics.* 118, 269–277.

Adams, L. B. R. (1994). Seeded Region Growing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 641-647.

Advanced Visual Systems, AVS, (n.d.) Retrieved September, 6, 2008 from http://www.avs.com, 2006.

Akiba, H., Fout, N., & Ma, K.L. (2006). Simultaneous Classification of Time-Varying Volume Data Based on the Time Histogram. *In Proceedings of EuroVis*, 171-178.

American College of Radiology, National Electrical Manufacturers Association. (2005) Digital imaging and communications in medicine (DICOM): Version 3.0. *in ACR/NEMA Committee, Working Group, ACR/NEMA Standards Pub*.

Amira Advanced 3D Visualization and Volume Modeling, (n.d.). Retrieved September, 19, 2008 from http://www.amiravis.com/.

Analyze, (n.d.) Retrieved September, 14, 2008 from http://www.mayo.edu/.

Anthea; M., Hopkins, J., McLaughlin, C.W., Johnson, S., Warner, Q. M., LaHart, D., & Wright J. D., (1993). *Human Biology and Health*. Englewood Cliffs, New Jersey, USA: Prentice Hall.

Bae, K.T., Giger, M.L., Chen, C.T., & Kahn, Jr.C.E. (1993). Automatic segmentation of liver structure in CT images, *Medical Physics*, 20. 71-78.

Bajaj, C.L., Pascucci, V., & Schikore, D.R. (1997). The Contour Spectrum, *In Proceedings of IEEE Visualization*, 167-173.

Bakker J., Olree M., Kaatee R., de Lange, E. E., Moons, K. G. M., Beutler, J. J., & Beek, F. J. A. (1999). Renal Volume Measurements: Accuracy and Repeatability of US Compared with That of MR Imaging, *Genitourinary Imaging*, 211, 623-628.

Bezdek, J.C., Hall, L.O., & Clarke, L.P. (1993). Review of MR image segmentation techniques using pattern recognition. *Medical Physics*, 20, 1033–1048.

Bidaut L. (2000). Data and Image Processing for Abdominal Imaging, A*bdominal Imaging*, 25, 341-360.

Bitter, I., Van Uitert, R., Wolf, I., Ibanez, L., & Kuhnigk, J. M. (2007) Comparison of Four Freely Available Frameworks for Image Processing and Visualization that Use ITK, *IEEE Transactions on Visualization and Computer Graphics*,13(3), 483-493.

Brodlie K. W., Gallop, J. R., Grant, A. J., Haswell, J., Hewitt, W. T., Larkin, S., Lilley, C. C., Morphet, H., Townend, A., Wood, J., & Wright H. (February 1991). *Review of Visualization Systems*, Retrieved September, 21, 2008 from http://www.agocg.ac.uk/reports/visual/vissyst/rvsrepor.htm.

Caban, J. J., Joshi, A., & Nagy, P. (2007) Rapid Development of Medical Imaging Tools with Open-Source Libraries, *Journal of Digital Imaging*, 20, 83-93.

Cerveri, P., Forlani, C., Borghese, N., & Ferrigno, G. (2002). Distortion correction for X-ray image intensifiers: A comparison between local unwarping polynomials and adaptive neural networks, *Medical Physics*, 29, 1759-1771.

Chen, S., Billings, S.A., & Luo, W. (1989). Orthogonal least squares methods and their application to nonlinear system identification, *International Journal of Control*, 50, 1873–1896.

Chen, S., Cowan, C.F., & Grant, P.M. (1991). Orthogonal least squares learning algorithms for radial basis function networks, *IEEE Transactions Neural Networks*, 2, 302–309.

Chou, J.S., Chen, S.Y., Sudakoff, G.S., Hoffmann, K.R., Chen, C.T. & Dachman, A.H. (1995). Image Fusion for Visualization of Hepatic Vasculature and Tumors., *Proceedings of SPIE*, 2434, 157-163.

Clark., J.W. (1991). Neural network modelling. *Physics in Medicine and Biology*, 36, 1259–1317.

Coleman, G.B., & Andrews, H.C. (1979). Image segmentation by clustering. *Proceedings of IEEE*, 5, 773–785.

Condon, B.R. Patterson, J., Wyper, D., Jenkins, A., & Hadley, D. M. (1989). Image non-uniformity in magnetic resonance imaging: its magnitude and methods for its correction. *The British Journal of Radiology*, 60, 83–87.

Doğan, H., Dicle, O., & Güzeliş, C. (2009). Extraction of the liver vessels from the abdomen CTA images by Hessian-based vessel filter, *Proceedings of Biomedical Engineering Meeting, 14th National, BIYOMUT*, 1-4, Turkey.

Drebin, R.A., Carpenter L., & Hanrahan, P. (1988). Volume rendering, *In Proceedings of ACM Siggraph*, 65–74.

Duda R.O. & Hart, P.E. (1973). *Pattern Classification and Scene Analysis*, New York: Wiley.

Duda, R., Hart, P. E., & Stork, D. G. (2000), *Pattern Classification,* Wiley-Interscience.

Dunn, J.C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-sparated clusters. *Journal of Cybernetics*, 3:32–57.

Fang, Z., Möller, T., Hamarneh, G., & Celler, A. (2007). Visualization and Exploration of Time-Varying Medical Image Data Sets, *In Proceedings of Graphics Interface 2007*, 281 - 288.

Ferrari, S., Frosio, I., Piuri, V., & Borghese, N.A. (2005). Automatic Multiscale Meshing Through HRBF Networks, *IEEE Transactions on Instrumentation and Measurement*, 54(4), 1463–1470.

Ferrari, S., Maggioni, M., & Borghese, N.A. (2004). Multiscale Approximation with Hierarchical Radial Basis Function Networks, *IEEE Transactions on Neural Networks*, 15(14), 178–188.

Fischer F., Selver M.A., & Hillen W. (2005). A Platform Independent Visualization Software for DICOM series, *in Proceedings: IEEE EMBS Symposium: Innovative Medical Imaging Modalities Proceedings*, 330.

Flohr, G., Schaller, S., Stierstorferi K., Bruder, H., Ohnesorge B.M., & Schoepf U.J. (2005). Multi Detector Row CT Systems and Image Reconstruction Techniques, *Radiology*, 235, 756-773.

Flohr, T.G., Schaller, S., Stierstorfer, K., Bruder, H., Ohnesorge, B.M., & Schoepf, U.J. (2005). Multi-detector row CT systems and image-reconstruction techniques, *Radiology*; 235(3):756-73.

Fujishiro, I., Azuma, T., & Takeshima, Y. (1998). Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topology Analysis, *In Proceedings of IEEE Visualization*, 467-470.

Gao, J., Kosaka, A., & Kak, A. (2000). A Deformable Model for Automatic CT Liver Extraction, *Proceedings of SPIE*, 2434, 157-163.

Gao, L., Heath, D.G., Kuszyk, B.S., & Fishman, E.K. (1996). Automatic liver segmentation technique for three-dimensional visualization of CT data. *Radiology*, 201., 359-364.

Gonzalez, R.C. & Woods, R.E. (1992). *Digital Image Processing*, Addison-Wesley Publications.

Gonzalez, R.C., & Wintz, P. (1987). *Digital image processing*. Reading, MA: Addison-Wesley.

H.A.Blum. (1967). Transformation for Extracting New Descriptors of Shapes. In W.Wathen-Dunn, (Ed.), *Models for the Perception of Speech and Visual Form* (362-380) Cambridge: MA, MIT Press.

Ha, K.V., (1998). Hierarchical Radial Basis Function Networks, *Neural Network Proceedings*, 1893-1898.

Hahn, H.K., Link, F., & Peitgen, H.O. (2003). Concepts for Rapid Application Prototyping in Medical Image Analysis and Visualization, *Proceedings of SimVision & Visualisierung Conf.*, 283-298.

Hall, L.O., Bensaid, A.M., Clarke, L.P., Velthuizen, R.P., Silbiger, M.S., & Bezdek, J.C. (1992) A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain. *IEEE Transactions on Neural Networks*, 3, 672–682.

Hansen, C.D., & Johnson, C. (2005). *The Visualization Handbook*. Elsevier.

Hartigan, J.A. (1975). *Clustering Algorithms*, New York: Wiley.

Haykin S. (1999) *Neural Networks: A comprehensive foundation*, (2nd ed.) Upper Saddle River, NJ: Prentice-Hall.

He, T., Hong, L., Kaufman, A., & Pfister, H. (1996). Generation of Transfer Functions with Stochastic Search Techniques. *In Proceedings of IEEE Visualization*, 227-234.

Heimann T., Meinzer H.P., & Wolf I. (2007). A Statistical Deformable Model for the Segmentation of Liver CT Volumes, In Heimann T, Styner M, & Van Ginneken B. (Eds). *MICCAI Workshop Proceedings: 3D Segmentation in the Clinic - A Grand Challenge*, 161-166.

Heimann, T., Wolf, I., & Meinzer, H.P. (2006). Active Shape Models for a Fully Automated 3D Segmentation of the Liver – An Evaluation on Clinical Data, *In proceedings of MICCAI, LNCS*, 4191, 41-48.

Hoffmann, H. (2000). *Filmless Radiology*, VDE-Verlag GmbH, Berlin and Offenbach, Germany.

Husain, S. A., & Shigeru, E. (2000). Use of neural networks for feature based recognition of liver region on CT images. *Neural Networks for Signal Processing Proceedings of the IEEE Workshop*, 2, 831-840.

Ibanez, L., & Schroeder, W., (November 21, 2005). *The ITK software guide*. Retrieved August, 2, 2008 from: http://www.itk.org/.

Image Processing and Analysis in Java, ImageJ, (n.d.) Retrieved September, 6, 2008 from http://rsb.info.nih.gov/ij.

IRIS Explorer, (n.d.) Retrieved September, 9, 2008 from http://www.nag.co.uk/Welcome_IEC.asp.

Jain, A.K. & Dubes. R.C. (1988), *Algorithms for clustering data*. Prentice Hall.

Kanjilal, P.P., & Banerjee, D.N. (1995). On the application of orthogonal transformation for the design and analysis of feedforward networks, *IEEE Trans. Neural Networks*, 6, 1061-1070.

Kaufman, A.E., Sobierajski, L.M. (1995). *Continuum volume display* (171–202). CRC Press, Boca Raton, FL.

Kindlmann, G. & Durkin, J.W. (1998). Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering, *In Proceedings of IEEE Visualization*, 79-86.

Kniss, J., Kindlmann, G., & Hansen C. (2001). Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widget. *Proceedings of Visualization 2001*, 255-262

Kniss, J., Kindlmann, G., & Hansen, C. (2002). Multi-Dimensional Transfer Functions for Interactive Volume Rendering, *IEEE Transactions on Visualization and Computer Graphics*, 8(3), 270-285.

Kniss, J.M., Van Uitert, Jr. R.L., Stephens, A., Li, G.S., Tasdizen, T., & Hansen, C.D. (2005). Statistically Quantitative Volume Visualization, *In Proceedings of IEEE Visualization*, 287-294.

Kohonen, T. (1995). *Self-Organizing Maps*, Berlin, Germany: Springer-Verlag.

Konig, A., & Gröller, E. (2001) Mastering Transfer Function Specification by Using VolumePro Technology, *In Proceedings of Computer Graphics'01*, 17, 279-286.

Koss, J.E., Newman, F.D., Johnson, T.K., & Kirch, D.L., (1999). Abdominal Organ Segmentation Using Texture Transforms and a Hopfield Neural Network, *IEEE Transactions on Medical Imaging*, 18(7), 640-648.

Lee C.C., Chung P. C., & Tsai H. M. (2003). Identifying Multiple Abdominal Organs From CT Image Series Using a Multimodule Contextual Neural Network and Spatial Fuzzy Rules, *IEEE Transactions on Information Technology in Biomedicine*, 7(3), 208-217.

Levoy, M., (1988) Display of surfaces from volume data, *IEEE Computer Graphics Applications*, 8 (5), 29–37.

Li H, & Santago, P. (2005). Automatic Colon Segmentation with Dual Scan CT Colonography, *Journal of Digital Imaging*, 18(1), 42-54.

Li, S.Z. (1995). *Markov random field modeling in computer vision*. Springer.

Liang, S. (1999). The Java Native Interface. Prentice Hall, NJ.

Lim, S.J., Jeong, Y.Y., & Ho, Y.S. (2006). Automatic Liver Segmentation for Volume Measurement in CT Images, *Journal of Visual Communication and Image Representation*, 17, 860-875.

Lorensen, W.E. & Cline, H.E. (1987) Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*; 21(3), 163–169.

Lowe, D. (1989). Adaptive radial basis function nonlinearities, and the problem of generalization, *in Proc. 1st IEE Conf. Artificial Neural Networks*, 171–175, London, UK.

Lum, E.B., Shearer, J., & Ma, K.L. (2006). Interactive multi-scale exploration for volume classification, *The Visual Computer*, 22(9-11): 622-630.

Lundström C., Ljung, P., & Ynnerman, A. (2006). Local Histograms for Design of Transfer Functions in Direct Volume Rendering. *IEEE Trans. On Visualization and Computer Graphics*, 12(16), 1570-1579.

Lundström, C., Ynnerman, A., Ljung, P., Persson, A., & Knutsson, H. (2006). The alpha-histogram: Using spatial coherence to enhance histograms and transfer function design. *In Proceedings of IEEE-VGTC Symposium on Visualization*,

Lutz, R.W., Pun T., & Pellegrini, C. (1991). Colour displays and look-up tables: real time modification of digital images. *Computerized Medical Imaging and Graphics,* 15(2), 73–84

MacQueen J. (1967) Some Methods for classification and analysis of multivariate observations, *in Proceedings 5th Berkeley Symposium Math. Statist. and Prob.*, 1, 281-297.

Maintz, J.B.A., & Viergever, M.A. (1998). A survey of medical image registration. Medical Image Analysis, 2, 1–36.

Marks, J., Andalman, B., Beardsley, P.A., & Pfister, H. (1997) Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. *In ACM Computer Graphics (SIGGRAPH '97 Proceedings)*, 389-400.

Martin K., Ibanez L., Avila, L., Barre, S., & Kaspersen, J. H. (2005). Integrating segmentation methods from the Insight Toolkit into a visualization application, *Medical Image Analysis*, 9, 579–593.

Masumoto, J. Hori, M., Sato, Y., Murakami, T., Johkoh, T., Nakamura, H., & Tamura, S. (2003). Automated Liver Segmentation Using Multislice CT Images, *Systems and Computers in Japan*, 34(9), 2150-2161.

MatLab, (n.d.) Retrieved August, 25, 2008 from http://www.mathworks.com.

Mebius R.E., & Kraal, G., (2005). Structure and function of the spleen. *Nature Reviews Immunology.* 5(8), 606-616.

Medical Image Processing, Analysis and Visualization, MIPAV, (n.d.) Retrieved September, 26, 2008 from http://mipav.cit.nih.gov.

MeVisLab: A Development Environment for Medical Image Processing and Visualization, MeVis, (n.d.) Retrieved August, 30, 2008 from http://www.MeVisLab.de.

Meyer, C.R., Peyton, H.B., & Pipe, J. (1995). Retrospective correction of intensity inhomogeneities in MRI. *IEEE Transactions on Medical Imaging*, 14, 36–41.

Mikic, I., Krucinski, S., & Thomas, J.D. (1998) Segmentation and tracking in echocardiographic sequences: active contours guided by optical flow estimates. *IEEE Transactions on Medical Imaging*, 17, 274–284.

Mojsilovic, A., Popovic, M.V., Neskovic, A.N., & Popovic, A.D. (1997) Wavelet image extension for analysis and classifcation of infarcted myocardial tissue. IEEE Transactions on Biomedical Engineering, 44, 856–866.

Montagnat J., & Delingette H. (1996). Volumetric Medical Image Segmentation Using Shape Constrained Deformable Models, *Lecture Notes in Computer Science : CVRMed-MRCA*, 1205, 13-22.

Moore K.L., & Dalley A.R. (1999). *Clinically Oriented Anatomy*, (4th Ed.), Lippincott Williams & Wilkins, Toronto.

Musavi, M.T., Ahmed, W., Chan, K.H., Faris, K.B., & Hummels, D.M. (1992). On the training of radial basis function classifiers, *Neural Networks*, 5, 595–603.

Ney, D.R., Fishman, E.K., Magid, D., & Drebin, R.A. (1990). Volumetric Rendering of Computed Tomography Data Principles and Techniques, *lEEE Computer Graphics and Applications*, 10(2), 24-32.

Olabarriaga, S.D., & Smeulders, A.W. (2001). Interaction in the segmentation of medical images: A survey, *Medical Image Analysis*, 5(2), 127-142.

OsirisX official website. (n.d.). Retrieved September, 17, 2008 from http://www.osirix-viewer.com/AboutOsiriX.html

Osirix, 3D DICOM Medical Viewer for MacOS X, (n.d.) Retrieved September, 17, 2008 from http://sourceforge.net/projects/osirix.

Otsu N. (1979). A Threshold Selection Method from Gray-Level Histograms, *IEEE Transactions on Systems, Man and Cybernetics*, 9(1), 62-66.

Park J. & Sandberg, I.W. (1991). Universal Approximation Using Radial-Basis Function Networks, *Neural Computing*, 3, 257–546.

Perona, P. & Malik, J. (1987). Scale-space and edge detection using anisotropic diffusion. *Proceedings of IEEE Computer Society Workshop on Computer Vision*, 16–22.

Pfister, H., Lorensen, B., Bajaj C., Kindlmann G., Schroeder, W., & Machiraju R. (2000). The Transfer Function Bake-Off, *In Proceedings of IEEE Visualization 2000*, 523-526.

Pham, D.L., Chenyiang, X., & Jerry, L.P. (2000). A survey of current methods in medical image segmentation, *Annual Review of Biomedical Engineering*, 2, 315-337.

Porter, T., Duff, T. (1984). Compositing digital images, *in: ACM Computer Graphics (SIGGRAPH'84 Proceedings)*, 253–259.

Prechelt, L. (2000). *An Empirical Comparison of C, C++, Java, Perl, Python, Rexx, and Tcl for a Search/String-Processing Program*, Fakultät für Informatik,

Universität Karlsruhe, Germany, Retrieved August, 17, 2008 from http://digbib.ubka.uni-karlsruhe.de/volltexte/372000.

Rautek, P., Bruckner, S., & Groller, M.E. (2007). Semantic Layers for Illustrative Volume Rendering, *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 1336 - 1343.

Robb RA. (1995). *Three dimensional biomedical imaging*. New York: Wiley–VCH Publishers.

Robb, R. A. (1999). *Biomedical Imaging, Visualization and Analysis*, Wiley-Liss, USA.

Roettger, S., Bauer, M., & Stamminger, M. (2005) Spatialized Transfer Functions, *In Proceedings of EuroVis '05*, 271-278.

Rojas, I., Pomares, H., González, J., Bernier, J.L., Ros, E., Pelayo, F.J., & Prieto, A. (2000). Analysis of the functional block involved in the design of radial basis function networks, *Neural Processing Letters*, 12(1), 1–17.

Rosenblatt, F. (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Cornell Aeronautical Laboratory, Psychological Review*, 65(6), 386-408.

Rosset, A., Spadola, L., & Ratib, O. (2004). OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images, *Journal of Digital Imaging*, 17(3):205-216.

Sahoo, P.K., Soltani, S., & Wong, A.K.C (1988) A survey of thresholding techniques. *Computer Vision Graphics Image Processing*, 41, 233–260.

Salama, C.R., Keller, M., & Kohlmann, P. (2006). High-Level User Interfaces for Transfer Function Design with Semantics, *IEEE Transactions on Visualization and Computer Graphics*, 12(5), 1021-1028.

Sato, Y., Westin, C.F., Bhalerao, A., Nakajima, S., Shiraga, N., Tamura, S., & Kikinis, R. (2000). Tissue Classification Based on 3D Local Intensity Structures for Volume Rendering, *IEEE Trans. On Visualization and Computer Graphics*, 6(2), 160-180.

Schalkoff, R.J. (1992) Pattern *recognition: statistical, structural and neural approaches*. John Wiley and Sons.

Schiemann, T., Bomans, M., Tiede, U., & Hoehne, K. H. (1992). Interactive 3-D segmentation, *Proceedings of SPIE*, 1808, 376.

Schroeder W., Martin K., & Lorensen, W. (1998). *The Visualization Toolkit: An Object Oriented Approach to 3-D Graphics* (2nd ed)., Prentice-Hall, Old Tappan, N.J.

Schroeder, W.J., Zarge, J.A., & Lorensen W.E. (1992) Decimation of triangle meshes. *Computer Graphics*; 26(2), 65–70.

Selver, M.A., Fischer, F., Kuntalp, M., & Hillen, W. (2007). A software tool for interactive generation, representation, and systematical storage of transfer functions for 3D medical images. *Computer Methods and Programs Biomedicine*, 86, 270-280.

Selver, M.A., Güzeliş, C. (2009). Semi-Automatic Transfer Function Initialization for Abdominal Visualization using Self Generating Hierarchical Radial Basis Function Networks, *IEEE Transactions on Visualization and Computer Graphics*, 15(3), 395-409.

Selver, M.A., Kocaoglu, A., Demir, G., Doğan, H., Dicle, O., & Güzeliş C. (2008) Patient oriented and robust automatic liver segmentation for pre-evaluation of liver transplantation, *Computers in Biology and Medicine*, 38(7), 765-784.

Seo K.S., Ludeman L.C., Park S.J., & Park. J.A. (2004). Efficient Liver Segmentation Based on the Spine. *LNCS ADVIS*, 3261, 400-409.

Sethian, J. (1996). *Level set methods and fast marching methods.* Cambridge University Press, Cambridge.

Sherstinsky, A., & Picard, R.W. (1996). On the efficiency of the orthogonal least squares training method for radial basis function networks, *IEEE Transactions Neural Networks*, 7, 195–200.

Shiaofen, F., Tom, B., & Mihran, T. (1998) Image-Based Transfer Function Design for Data Exploration in Volume Visualization, *In Proceedings of IEEE Visualization*, 319-326.

Simmons, A., Tofts, P.S., Barker, G.J., & Arridge, S.R. (1994) Sources of intensity nonuniformity in spin echo images at 1.5T. *Magnetic Resonance in Medicine*, 32, 121–128.

Slone, R.M., Fisher, A.J., Pickhardt, P.J., Gutierrez, F.R., & Balfe, D.M. (2000) *Body CT: A Practical Approach*. New York: McGraw-Hill.

Soler, L., Delingette, H., Malandain, G., Montagnat J., Ayache, N., Koehl, C., Dourthe O., Malassagne, B., Smith, M., Mutter, D., & Marescaux, J. (2001). Fully Automatic Anatomical, Pathological, and Functional Segmentation From CT Scans for Hepatic Surgery., *Computer Aided Surgery*, 6(3), 131-142.

Spielmann, A. L.; DeLong, D. M., & Kliewer, M. A. (2005). Sonographic Evaluation of Spleen Size in Tall Healthy Athletes. *American Journal of Roentgenology (American Roentgen Ray Society)*. 184, 45–49.

Steinmetz, E. F., Buckley, C., & Thompson, R.W. (2003). Prospects for the Medical Management of Abdominal Aortic Aneurysms, *Vascular and Endovascular Surgery*, 37(3), 151-163.

Suetens, P., (2002). *Fundamentals of Medical Imaging*, Cambridge University Press, United Kingdom.

The Data Visualization & Analysis Platform, IDL, (n.d.) Retrieved September, 11, 2008 from http://www.rsinc.com/idl.

Tian, J. Xue, Y. Dai, Chen J. & Zheng J. (2008). Novel Software Platform for Medical Image Processing and Analyzing, *IEEE Transactions on Information Technology in Biomedicine*, 12(6), 800-812.

Tiede, U., Hoehne, K.H., Bomans, M., Pommert, A., Riemer, M., & Wiebecke, G. (1990) Investigation of medical 3D-rendering algorithms, *IEEE Computer Graphics and Applications*, 10(2), 41-53.

Tortora, G. J., & Anagnostakos, N. P. (1984) *Principles of Anatomy and Physiology*, Harper & Row Publishers, New York

Tory, M., & Torsten, M. (2004) Human Factors in Visualization Research, *IEEE Transactions on Visualization and Computer Graphics*, 10(1), 72-84.

Tsai, D.F. (1994): Automatic segmentation of liver structure in CT images using a neural network. *IEICE Trans. Fundamentals*, 77(11), 1892-1895.

Tzeng, F.Y., Lum, E.B., & Ma K.L. (2005) An Intelligent System Approach to Higher-Dimensional Classification of Volume Data. *IEEE Trans. On Visualization and Computer Graphics,* 11(3): 273-284.

Upson, C., Faulhaber, T., Kamins, D., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz, R., & Van Dam, A. (1989). The Application Visualization System: A Computational Environment for Scientific Visualization. *IEEE Computer Graphics and Applications*, 30-42.

Uykan, Z., Güzeliş, C., Çelebi, M.E., & Koivo, H.N. (2000). Analysis of input-output clustering for determining centers of radial basis function networks, *IEEE Transactions of Neural Networks*, 11, 851–858.

Van Ginneken, B, Heimann, T, & Styner, M. (2007). 3D Segmentation in the Clinic: A Grand Challenge. In Heimann T, Styner M, van Ginneken B (eds) (7-15). *MICCAI 2007 Workshop Proceedings: 3D Segmentation in the Clinic - A Grand Challenge*.

Vilarino, D.L., Brea, V.M., Cabello, D., & Pardo. J.M. (1998) Discrete-time CNN for image segmentation by active contours. *Pattern Recognition Letters*, 19, 721–734.

Vincent L. (1993). Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms, *IEEE Transactions on Image Processing*, 2(2), 176-201.

Vis5d, (n.d.) Retrieved September, 12, 2008 from http://vis5d.sourceforge.net.

VisAD Java Visualization, (n.d.) Retrieved September, 12, 2008 from http://sourceforge.net/projects/visad.

VisiQuest Visual Framework, (n.d.) Retrieved September, 30, 2008from http://www.accusoft.com/imaging/visiquest.

Vivanco, R. A., & Pizzi, N. J. (2004). Scientific computing with Java and C++: a case study using functional magnetic resonance neuroimages, *Software: Practice and Experience*, 35(3), 237 - 254.

VolView: A Volume Visualization System, (n.d.) Retrieved August, 4, 2008 from http://www.VolView.com.

Wachs, M. E., Bak, T. E., Karrer, F. M., Everson, G. T., Shrestha, R., Trouillot, T. E., Mandell, M. S., Steinberg, T. G., & Kam, I. (1998) Adult Living Donor Liver Transplantation Using A Right Hepatic Lobe, *Transplantation*, 66(10), 1313-1316.

Watt, A. (1993) *3D Computer Graphics*. Reading, MA: Addison-Wesley.

Zadeh., L.A. (1965) Fuzzy sets. *Information and Control*, 8, 338–353.