DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AN OBJECT ORIENTED TRANSFER FUNCTION EDITOR FOR INTERACTIVE MEDICAL VOLUME VISUALIZATION

by
Mustafa Alper SELVER

August, 2005
İZMİR

AN OBJECT ORIENTED TRANSFER FUNCTION EDITOR FOR INTERACTIVE MEDICAL VOLUME VISUALIZATION

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

by
Mustafa Alper SELVER

August, 2005
İZMİR

M.Sc. THESIS EXAMINATION RESULT FORM

We have read this thesis entitled **"AN OBJECT ORIENTED TRANSFER FUNCTION EDITOR FOR MEDICAL VOLUME VISUALIZATION"** completed by **Mustafa Alper SELVER** under supervision of **Assist. Prof. Dr. Mehmet KUNTALP** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Mehmet KUNTALP

Supervisor

Prof. Dr. Cüneyt GÜZELİŞ

(Jury Member)

Prof. Dr. Oğuz DİCLE

(Jury Member)

Prof. Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to thank my supervisor Assist. Prof. Dr. Mehmet KUNTALP for his valuable guidance and support not only during my master study but also in every stage of my experience in assistantship.

I also would like to thank Prof. Dr. Walter HILLEN, Prof. Dr. Cüneyt GÜZELİŞ and Prof. Dr. Oğuz Dicle for giving me the opportunity to work in FH-Juelich Medical Informatics Laboratory and for their valuable guidance during preparation of this thesis.

I wish to express my appreciation to M.Sc. Dipl. Ing. Felix FISCHER for his great support on completing this thesis and for his friendship during my stay in Germany.

I would like to thank my parents, sister, parents-in-law and sister-in-law for their trust and patience.

Finally, I would like to thank my dear wife, for her trust. Her kindness and supportiveness mean a lot to me. Without her continuous encouragement; I could not start and finish this work. Thank you for believing in me.

M. Alper SELVER

AN OBJECT ORIENTED TRANSFER FUNCTION EDITOR FOR INTERACTIVE MEDICAL VOLUME VISUALIZATION

ABSTRACT

Visualization of 3D medical image data, obtained from CT/MR can provide valuable assistance for data analysis and decision making tasks. The effective and interactive visualization of 3D datasets requires appropriate adjustment of several parameters such as color effects, varying levels of transparency, etc. As being the tool that assigns these parameters to the original values of dataset being visualized, transfer functions have very important effects on the quality of volume rendered images. Unfortunately, finding proper transfer functions is a very difficult task because of the variety of possible transfer functions. This motivates the study of transfer function design, the development of new interfaces for transfer function specification, and a consideration of how interaction techniques in visualization systems can simplify transfer function creation. In this study, an object oriented Transfer Function Editor (TFE) is implemented by using Java programming language. For testing, TFE is added as a module to exploreDICOM software and is evaluated by medical experts.

Keywords: Transfer function, 3D imaging, user interface, DICOM, volume rendering

ETKİLEŞİMLİ TIBBİ ÜÇ BOYUTLU GÖRÜNTÜLEME İÇİN NESNE TABANLI BİR TRANSFER FONKSİYON EDİTÖRÜ

ÖZ

BT, MR gibi görüntüleme cihazlarından elde edilen kesit görüntülerin, üç boyutlu (3B) görüntü haline dönüştürülmesi, tıp uzmanlarının görüntü analizine ve tanı koymasına önemli yardımlarda bulunmaktadır. 3B data kümelerinin amaca uygun ve interaktif olarak görüntülenmesi renk, donukluk başta olmak üzere pek çok parametrenin uygun şekilde ayarlanmasını gerektirmektedir. 3B hacim görüntülemede, bu parametrelerin özgün veri kümesine atanmasını sağlayan transfer fonksiyonu (TF), elde edilen 3B görüntünün amaca uygun olmasında ve kalitesinde önemli ölçüde etkilidir.

Ancak, uygun TF saptanması, olası transfer fonksiyon uzayının genişliği sebebiyle zaman alıcı bir süreçtir. Bu yüzden TF geliştirilmesi, TF belirtimi için yeni arayüzler geliştirilmesi ve hangi interaktif yöntemlerin uygun TF'lerin belirlenmesini kolaylaştıracağı yönündeki çalışmalar sürmektedir. Bu çalışmada, Java dili kullanılarak, nesne tabanlı, 3B görüntüleme programları ile çalışabilen, kullanımı kolay ve elde edilen faydalı TF'lerin arşivlenmesini sağlayacak dosyalama sistemine sahip bir Transfer Fonksiyon Editörü (TFE) geliştirilmiştir. Geliştirilen yazılım, exploreDICOM tıbbi görüntüleme yazılımına modül olarak eklenerek, 3B görüntülemeyi rutin olarak kullanan doktorlar tarafından değerlendirilmiştir.

Anahtar Kelimeler: Transfer fonksiyonu, kullanıcı arayüzü, DICOM, 3-Boyutlu Görüntüleme

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

1.1. Digital Radiology, DICOM Standard and Viewers

Advances in technology over the past two decades have resulted in significant changes in the field of radiology. Due to innovative approaches and developments on computer technology, image capturing, high speed networks and ultrahigh resolution monitors during the past 15-20 years, transition to filmless radiology is becoming a standard. All imaging modalities are becoming primarily digital and thus, new workflow in radiology departments is becoming completely digital (Siegel & Kolodner (Eds.), 1999). This change requires new hardware and software for communication, storage, and display of images.

Filmless radiology has the potential to provide two major benefits. The first is the improved accessibility, integration, and efficiency made possible by the incorporation of images into the patient's electronic medical record (Figure 1). The second is the creation of new techniques to take advantage of the recent developments in image acquisition, display and processing.

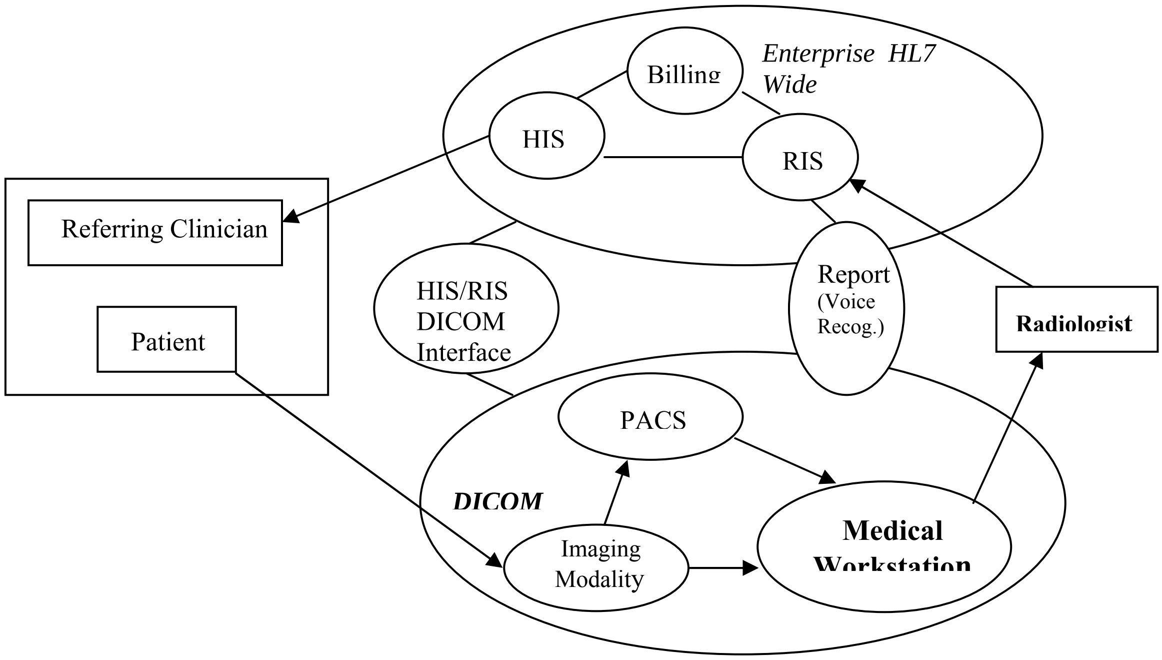


Figure 1.1 Radiology workflow

1.1.1. PACS

PACS (Picture Archiving and Communication Systems) are complete medical systems with all hardware and software included, which are installed to run digital medicine workflow. They usually include archiving servers (one or many, for online and deep storage) and diagnostic workstations, integrated with digital image acquisition devices. Current PACS are DICOM (Digital Imaging and Communications in Medicine) based, which ensures their conformance to the standards. Each PACS vendor releases a "DICOM conformance statement," explaining to what extent their PACS support DICOM (Siegel & Kolodner, 1999).

The related elements of digital radiology with this study are DICOM standard and all kinds of viewers (diagnostic/clinical workstations and web viewers), which are briefly described below.

1.1.2. An Overview of DICOM Standard

DICOM is an industry standard for the transmission of digital medical images and related information. It is by far the most universal standard in digital medicine. It provides all the necessary means for diagnostically correct representation of medical data. Moreover, DICOM is not just an image or file format - it is primarily a networking protocol, built on top of TCP/IP and designed to store and transmit medical data.

As an image file format, DICOM supports up to 65536 (16 bits) shades of gray for the monochrome image display, still dominating many modalities. With this respect, converting DICOM images into JPEGs or bitmaps (with 256 shades of gray) makes them practically unusable for diagnostic reading. On the other hand, DICOM does include various image-encoding formats, from basic bitmapped data to wavelets.

DICOM does not only store the image pixels, but also image acquisition parameters. It records a multitude of other image-related parameters such as patient 3D position, physical sizes of the objects in the image, slice thickness, image exposure parameters, etc. This data enormously enriches the informational content of DICOM images, and facilitates processing and interpreting the image data in various ways - for instance, creating 3D images from the sequences of 2D CT / MRI slices.

Complete encoding of medical data: DICOM files and messages use about 4000 standardized fields ("DICOM dictionary") to convey various medical data - from patient name to image color depth to current patient diagnosis. This data is often essential for correct diagnosis.

1.1.3. DICOM Information Object Definitions

DICOM Information Object Definitions, IODs, are description of entities to which the standard refers. These entities include objects such as Patient, Study, etc. These definitions specify the attributes for each object. For example, Patient has attributes including ‘Name’, ’Patient ID’ etc. An instance of an IOD refers to an actual object of the corresponding class. For example, an instance of ‘patient’ is an individual attending the hospital. Attributes of an IOD is divided into three classes, which specify whether or not their presence is required in every instance of an IOD. Some attributes are compulsory, some are desired, and some are optional.

Unique Identifiers (UIDs) are used to identify instances and definitions, and are long numeric strings. Each IOD and attribute has its own UID. In the present study, Series Instance UID, which is a compulsory attribute, is used as the identifier.

1.1.4. Diagnostic/Clinical Workstations & Web/DICOM Viewers

Workstations are high performance computers that are capable of communicating through a PACS and of viewing DICOM or non-DICOM images. All components of a PACS are needed for the system to operate, but the workstations have the unique function of serving as the human interface for the display role. As such, it has to support a user interface that is readily learned and used, yet it must be flexible enough to meet the varied needs of different healthcare professionals. In the filmless environment, workstations must be able to display faithfully the information they are sent or, at least indicate to user that there may be changes in the original data.

The speed of the workstations must be fast enough to avoid user frustration, but not so fast as to drive costs for the hardware to impractical levels. Above all, the workstations must offer users clear advantages over films, in speed, functions performed, or ability to do tasks not possible with films (Siegel & Kolodner, 1999).

With medical workstations, it is possible to examine multiple studies simultaneously, measure, rotate, zoom, and annotate in all-in-one applications. Also 3D/MIP, MPR, and Measurement Tools are increasing the clinical functionality of the workstations. Moreover, workstations incorporates many features required for day to day work such as window/level presets, user defined profiles, customizable toolbars, presentation states, cine function and exporting images.

Another related important point is legal issues, according to which, making a diagnosis is only possible with diagnostic workstations which have license and which must satisfy necessary requirements (e.g. interpolation)

Clinical workstations are the computers on which, post processing operations are done by technicians before the image is sent to diagnostic workstations used by medical experts.

Web/DICOM viewers are the tools which are used in radiology, other clinics, or outside the hospital to access medical images. They can be standalone applications or browser based tools (i.e. Java applet).

1.2. Java Medical Image Viewer – ExploreDICOM

Diagnostic workstations require high-end hardware equipment, which increases the costs and flexibility of usage at different locations. They are mostly located in the report rooms and are in limited numbers. Since medical experts are constantly visiting conferences, exhibitions etc., it is important to have viewers that can be used in personal PCs, laptops etc. The low cost of personal computers and wide variety of communication choices to access the internet or an intranet mean that the goal of wide availability of images and other medical information can be met. It is likely that the majority of future workstations used outside the radiology department, or other specialties that do not provide primary diagnosis from workstations, will be PC-based and access necessary information through a browser.

It is also observed that a practical imaging tool, which can work not only with PACS but also with RIS, would be very useful and increase efficiency of radiology workflow when the image is required to be checked outside the report room or at any other place in the hospital.

ExploreDICOM is a clinical & web (non-diagnostic) viewer, which is specially developed due to reasons mentioned above. It includes most of the features that medical experts and technicians need in their daily workflow but also does not require high-end hardware.

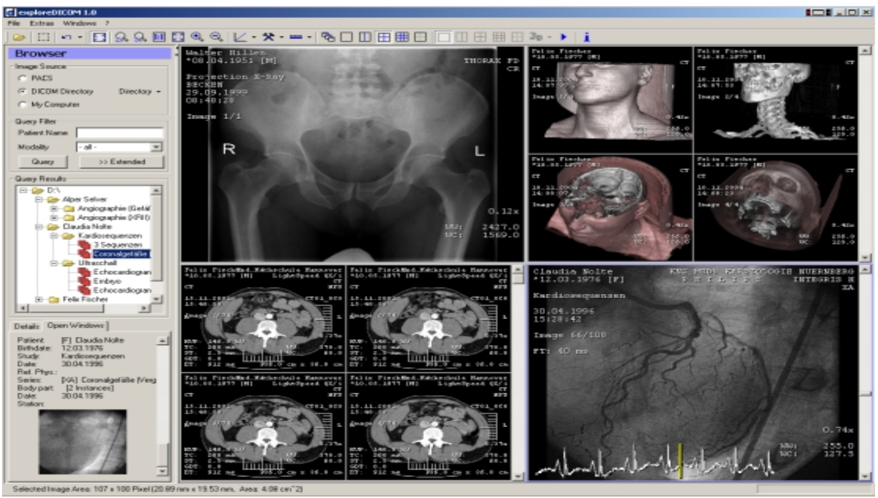


Figure 1.2 ExploreDICOM Graphical User Interface

The graphical user interface (GUI), which is illustrated in Figure 1.2, is designed in such a way that the user can figure out how to use it without referring to a manual. The style of GUI elements depends on the operating system (OS) of the computer so that the user would be familiar with most of the functions (i.e. save, print, zoom…) since these are provided by the OS.

As illustrated in figures 1.2, 1.3, 1.4, 1.5 and 1.6, it is possible to examine multiple studies simultaneously, by measuring, rotating, zooming and applying image processing operations (LUT adjustment, filtering…) which are important and helpful tools for medical experts.

Figure 1.3 Zoom option

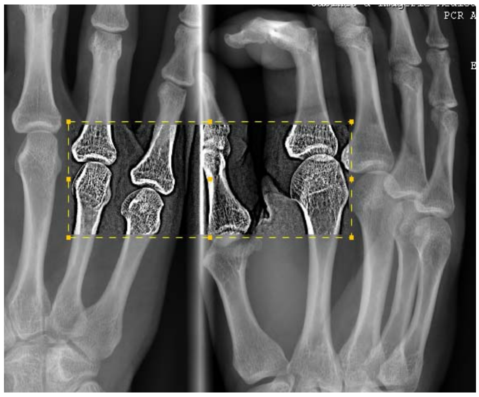


Figure 1.4 Filtering option

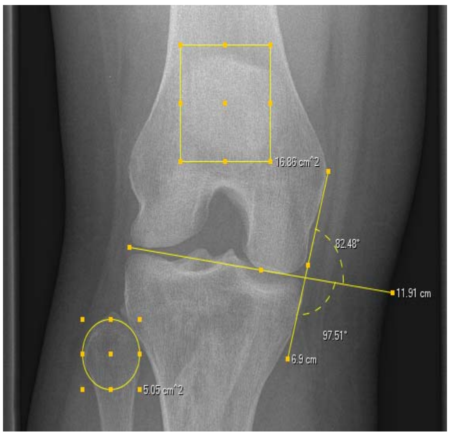


Figure 1.5 Measurement option

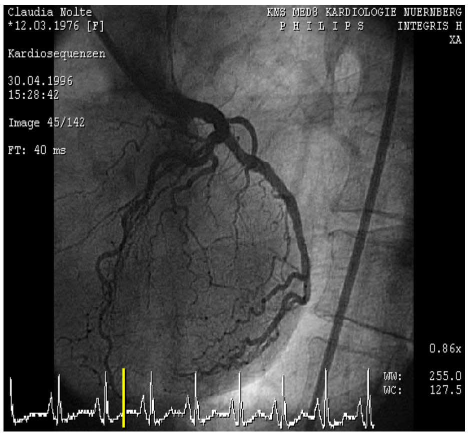


Figure 1.6 Cine option

By using the browser (shown on the left side of the user interface in Figure 1.2), the user can connect the program to PACS to get DICOM images or can use a CD or computer's hard disk to open any DICOM or non-DICOM image. Browser works on the basis of standard query-retrieve procedure. When the program is connected to PACS or any directory that includes DICOMDIR file, the contents can be seen in a tree format: patient name→ study →series. Standard searching/filtering option, on the other hand, is available in terms of the modalities, patient name, etc…

ExploreDICOM has also the functionality of obtaining MPR and 3D images from CT and MR studies, as illustrated in Figure 1.7. What are available so far in this part is oblique sectioning and image processing tools for MPR; Surface rendering and volume rendering including isosurface detection, maximum intensity projection (MIP), compositing, several interpolation methods and rendering options for 3D visualization.

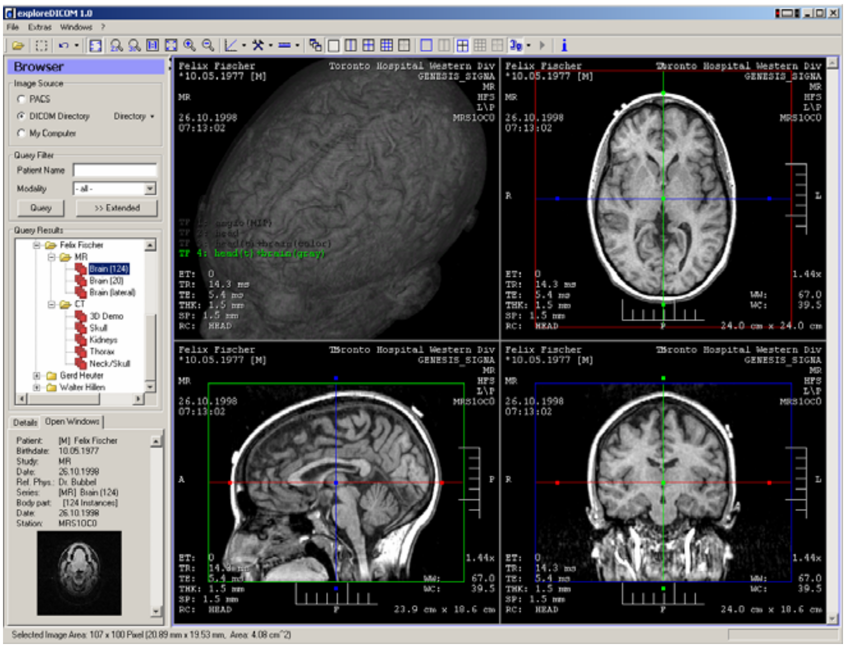


Figure 1.7 MPR and 3D visualization

Software is pure Java-based, which gives the opportunity of efficient memory usage and extensibility of the software. It is developed by using object-oriented programming which makes it possible to add new modules (such as a Transfer Function Editor in this study) without a significant change in the main program.

1.3. 3D Visualization in Medicine

Visualization of medical image data, obtained from CT and MR can provide valuable assistance for data analysis and decision making tasks. The goal is to produce clear and informative pictures of the important structures in a dataset. Volume visualization is currently in use in surgery, radiological treatment planning, and anatomical education. It is also used as a tool to help diagnosis. Several studies are going on to improve performances of current systems (Peter & Duff, 1984)

Besides these benefits, medical volume visualization is still not cost-effective in terms of applications, despite the fact that hardware and software technologies emerge very rapidly. Several factors are responsible for this limited application success:

- Routine use of 3D visualization in medicine is still limited to research and teaching hospitals. (Legal issues and restrictions are also playing important role on this limited usage of 3D visualization)
- There is no standard or consensus on software programming interfaces resulting in vendor specific solutions, which limit medical experts. This is because of the unpractical experience of getting used visualization software in their limited time during daily work.
- The volume techniques are difficult to use by medical experts when compared to experienced engineers or scientists.
- Volume visualization that can be used in clinical environment still requires expensive workstations with large amounts of memory and special graphics hardware extensions.

That is why several research activities are addressing these limitations. In this study, a user-friendly, platform independent, and low-cost interface is realized for the same purpose.

Visualization of medical data can roughly be categorized into two parts: surface and volume rendering. Surface rendering is based on creating intermediate geometric structures, such as polygons comprising an isosurface. However, with volume rendering, visualizations can be created simply by a direct mapping from volume data points to composite image elements. For medical applications, direct volume rendering is a key technology because of the opportunity of interaction with volume data in terms of color, opacity, etc.

1.4. The Task of Finding Transfer Functions

The effective and interactive visualization of 3D data sets requires appropriate adjustment of several parameters such as color effects, varying level of transparencies, emittance, scattering, etc. As being the tool that assigns these parameters to the original values of the data set being visualized, transfer functions have very important affects on the quality of direct volume rendered images. Unfortunately, finding good transfer functions is a very difficult task because of the variety of possible transfer functions. Since this flexibility is generally can not be kept in strict bounds, finding an appropriate transfer function for a meaningful and intelligible volume rendering is also one of the hardest parameters to set appropriately. This motivates the study of transfer function design, the development of new interfaces for transfer function specification, and a consideration how interaction techniques in visualization systems can simplify transfer function creation. Although several different transfer functions are currently being used, none of them achieves the status of being called as a standard transfer function. These current approaches can be divided into four groups (Pfister et al., 2000).

1.4.1. Trial and Error Approach

This approach addresses the need for human intervention in generating a final product. It states that data exploration is an essential element of creating the transfer function if the images were to fulfill the observer’s expectations and to be considered efficient. It is based on the idea that methods which served to generate images without human interaction would produce nice but ineffective images since they do not consider specific need/aim of the user (Pfister et al., May/June 2001).

1.4.2. The Data-Centric, Without Data Model Approach

This method is based on a technique for isovalue determination. A contour spectrum consists of computed metrics over a scalar field. By using these metrics, a set of functions can be defined which provide a useful tool to enhance a data set's interactive query. In a 3D isosurface display, for example, where one contour may be hidden inside another, if the isocontour display is associated with a contour tree, it is realized that the isosurface has two components. By providing a compact visual representation of metrics evaluated over the range of possible isovalues, the user can decide, based on rendering goals, which isolevel to use (Bajaj et al., 1997).

1.4.3. The Data-Centric, With Data Model Approach

This approach makes the assumption that the features of interest in the data are the boundary regions between areas of relatively homogenous material. This is done by defining the values which are consistently associated with these boundaries as opaque. This model, therefore, defines a semi-automatic method with little user interaction as a preprocessing stage where edge detection concepts from computer vision technologies lead to the creation of a 3D histogram of data values versus their first and second derivatives. From these values, a distance map is generated which is restrained by the boundary information of the given data set. This method is described as problematic with data sets in which noise and course boundaries occur. Advantage of this method is its ability to create transfer functions concerning not just data value but also a 2D space of data value and gradient magnitude which is efficient at disambiguating complex material boundaries (Kindlmann et al., 1998)

1.4.4. The Image Model Approach

This approach is based on evaluating transfer functions on the basis of images they produce and not in terms of the dataset properties. It is an approach with no user interaction except that the user can select from presented possibilities of rendered images, which satisfy specific requirements and thus optimize the transfer function.

As potentially hundreds of different renderings have to be made, this technique relies on fast rendering hardware being available to reach its full potential as an interactive method (Konig & Gröller, April 2001).

1.4.5. Conclusion on Existing Approaches

As mentioned above, several approaches exist for solving specific problem of transfer function design and development of new interfaces for transfer function specification. However transfer function specification for medical volume visualization is a very subjective task. Except in the cases that boundaries are important, medical experts always want to interact with volume data easily and quickly. The lack of mathematical background of medical experts is also makes a question mark on the need of providing statistical or metric information of scalar data. Finally it should be stressed that, except advanced research clinics or institutes; trial and error approach based transfer function specification is mostly used during clinical daily work.

CHAPTER TWO
THEORY

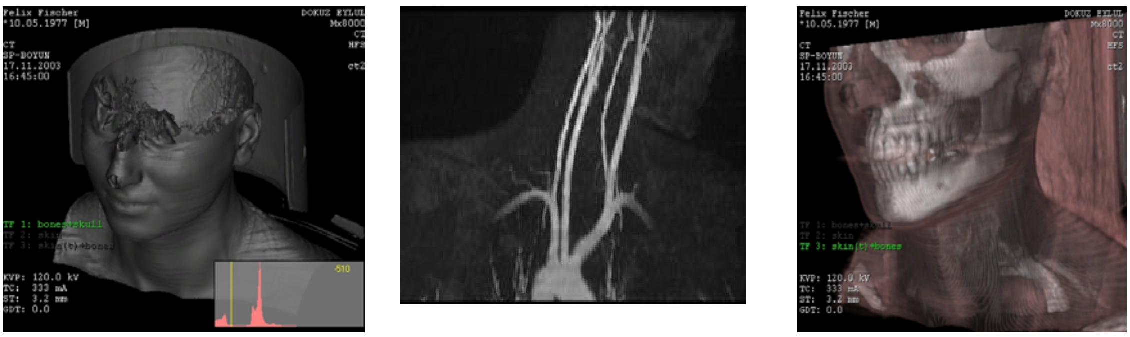
2.1. Visualization Pipeline for Volume Rendering

Visualization techniques can be divided into two groups: Surface and Volume Rendering. Surface rendering deals with geometric feature extraction at a predefined isovalue and provides poor user interaction. The advantage of surface rendering is that once the surfaces are rendered, no further calculation is done unless a new isovalue is defined. However, as computers are becoming more powerful, other (i.e. costlier) methods can be implemented (Porter & Duff, July 1984).

The term “volume rendering” contains many different techniques to map 3D data onto 2D images. The major volume rendering techniques are “raycasting” and “texture mapping”. Texture mapping is limited with the graphics hardware while raycasting performance is determined by the technique being used.

The three major raycasting techniques are:

- Isosurface detection: surfaces of equal values are displayed as isovalues
- Maximum Intensity Projection (MIP): the most intense pixel on the way of the ray is selected and viewed on the screen
- Composite Volume Rendering: Composite volume rendering consists of several steps as can be seen in Figure 2.1. Transfer function specification is an element of this method so further on; rendering pipeline will be described based on composite volume rendering.



(a) **(b)** **(c)**

Figure 2.1 (a) Isosurface detection (b) MIP (c) Composite Volume Rendering

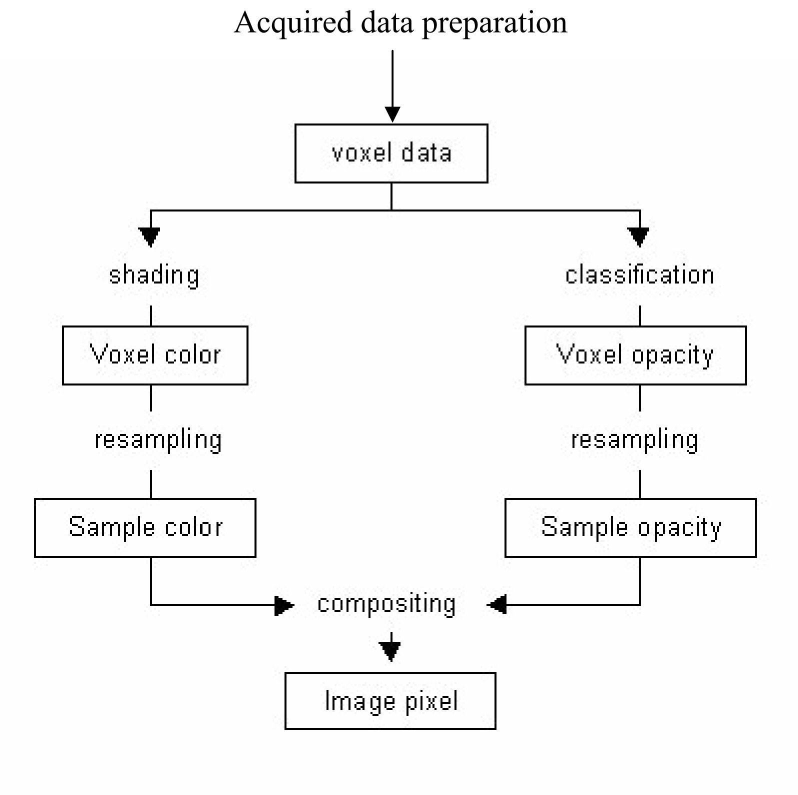


Figure 2.2 The composite volume rendering pipeline

The first step of the rendering pipeline is preprocessing. In this step interpolation, segmentation, and filtering stages remove Gaussian noise and other interferences or extract interesting features. The next two steps process the resulting data from the previous step. Optical properties like color and opacity are assigned by specific functions to the interpolated values. These properties are calculated along viewing rays as shown in Figure 2.3.

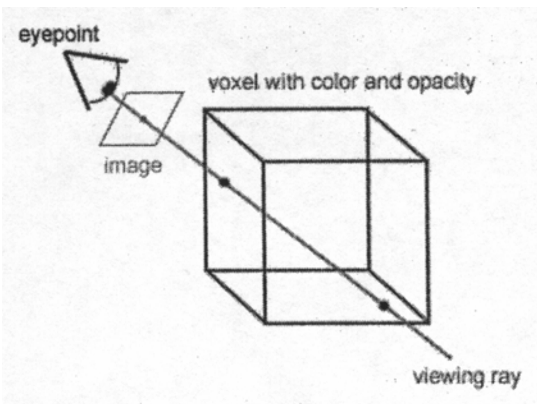


Figure 2.3 Raytracing: A viewing ray is sent from the eye-point to every pixel.

2.2. Transfer Function Specification & Classification

First of all, transfer functions are functions in the strict mathematical sense. Therefore, it is important to identify the domain and range of the transfer function in question. This is also a natural way to categorize the different types of transfer functions.

In the simplest type of transfer function, the domain is the scalar data value (assuming the volume dataset itself is scalar) and the range is opacity. Moreover, any optical property that can be represented and composited by computer graphics can be in the range of a transfer function. These elements can be represented in varying degrees of sophistication; for instance, opacity can vary according to color, instead of being a constant scalar value.

Transfer functions can also be generalized by increasing the dimension of the function's domain. These are generally termed *multi-dimensional* transfer functions. This approach is based on the idea that in scalar volume datasets, a second dimension of gradient magnitude characterizes how quickly values are changing in a given neighborhood, which tells the distinction between homogeneous regions (Kniss et al., October 2001).

Another type of *multi-dimensional* transfer functions arises from the visualization of volume data, which is not a single scalar field. For instance, MR imaging can measure many different physical quantities in living tissue (proton density, T1 and T2 relaxation times), and any combination of these can serve as axes in the domain of a multi-dimensional transfer function (Marks et al., August 1997).

The flexibility of the transfer functions has the advantage of making direct volume rendering to be extremely expressive. The disadvantage is that they are difficult to be found. The reasons for this deficiency are:

1. Transfer functions have high number of degrees of freedom even when using simple linear ramps (every control point adds two degrees of freedom).

2. The user interfaces are generally not constrained or guided by the problem domain from which the dataset came because the histogram (which is the most common background figure) does not contain spatial information.
3. Transfer Functions are inherently non-spatial in the sense that their assignment of color and opacity does not include spatial position as a variable in their domain. This can be frustrating if the user is interested in isolating one feature of the volume, which is spatially localized, but not distinguishable, in terms of data value, from other regions in the volume.

However the disadvantages can be overcome in medical volume visualization. Since different anatomical regions are known to lie on different ranges (i.e. Hounsfield values for CT), the user (medical expert) can have an idea of how a manipulation in transfer function may change the volume that is being visualized spatially or which parts of the transfer function should be changed in order to be able to see a region that is spatially localized.

2.3. Opacity Functions of Scalar Data Value

Since the direct volume rendered image is generally composed of repeated applications of the over operator, the extent to which a data value is visible in the final image is determined by how much opacity it contributes. Only the important features should receive high opacity, otherwise they can be blocked by uninteresting opaque regions. Because of the fundamental role that opacity plays in creating intelligible volume visualization, this particular type of transfer function can be given more specific name opacity function. On the other hand, the process of assigning opacity to a volume data point is often called classification.

Opacity functions can be generalized to different types of transfer function by augmenting the function's range. The range often includes color because color is a simple and natural way to visually distinguish between structures.

In this study, we use one of the most common forms of transfer functions. This is based on moving control points that are defining a set of linear ramps. Transfer function's domain is selected as gray value (or Hounsfield value) and its range is selected as opacity. The range includes color in a manner that the user can specify a color to the control points. The color values between two control points are distributed by a linear interpolation. Moreover, the user can also add a control point that does not have a color value. In this study, this type of nodes is called opacity nodes while the other ones are called as color nodes. Color nodes are shown by their color and opacity nodes are represented by their interpolated color values (appearance of the color is also dependent on opacity value of the node which is described in detail in chapter 4).

2.4. Hounsfield Values

Every acquired CT slice is subdivided into a matrix of up to 1024 × 1024 volume elements **(voxels).** Each voxel has been traversed during the scan by numerous X-ray photons and the intensity of the transmitted radiation is measured by specific detectors. From these intensity readings, the density or **attenuation value** of the tissue at each point in the slice can be calculated. Specific attenuation values are assigned to each individual voxel. The viewed image is then reconstructed as a corresponding matrix of picture elements **(pixels).**

Each pixel is assigned a numerical value (CT number), which is the average of all the attenuation values contained within the corresponding voxel. This number is compared to the attenuation value of water and displayed on a scale of arbitrary units named **Hounsfield units (HU)** (Hounsfield, December 1979).

This scale assigns water as an attenuation value (HU) of zero. The range of CT numbers is 2000 HU although some modern scanners have a greater range of HU up to 4000. Each number represents a shade of grey with +1000 (white) and -1000 (black) at either end of the spectrum, as shown in Figure 2.4.

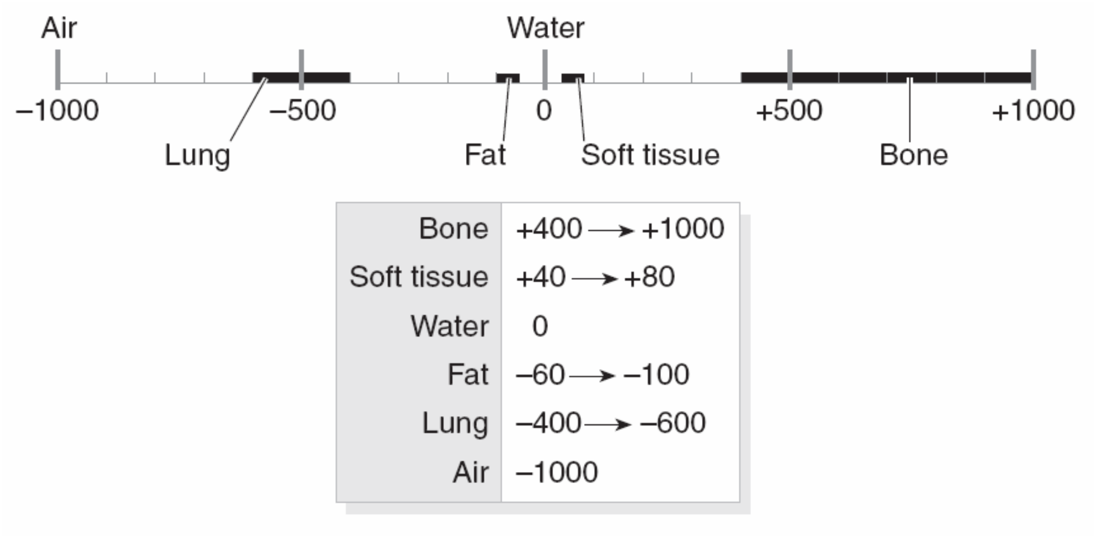
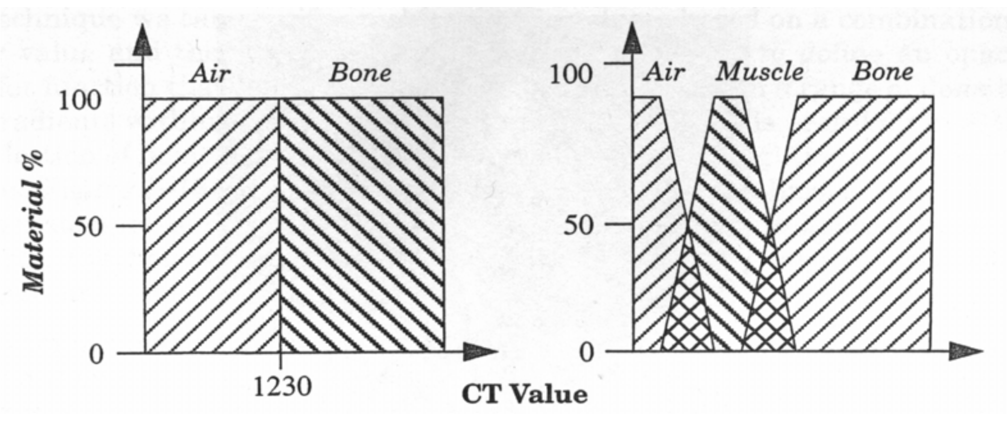


Figure 2.4 Hounsfield values

Whilst the range of CT numbers recognized by the computer is 2000 to 4000, the human eye cannot accurately distinguish even 2000 different shades of grey. Therefore to allow the observer to interpret the image, only a limited number of HU are displayed (by using Window Level [WL], Window Width [WW]).

Hounsfield values are also useful to classify data in 3D imaging. When HUs are defined to be the domain (x axis) of Transfer Functions, CT densities are classified into material percentages (if the range of Transfer Functions is selected as Material %). This classification can be a simple binary classification as given in Figure 2.5 (a) or gradual transitions from one material (i.e. air) to another (i.e. bone), as shown in Figure 2.5 (b).

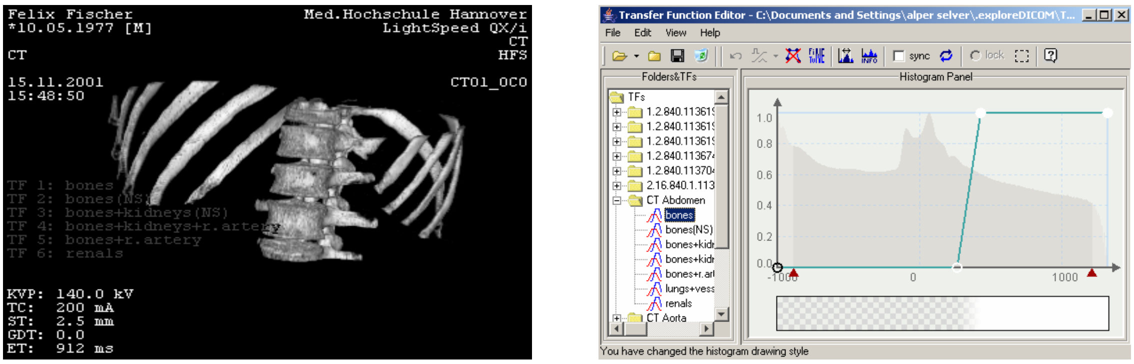


(a) **(b)**

Figure 2.5 HU vs. Material Percentage (a) binary classification (b)gradual transition

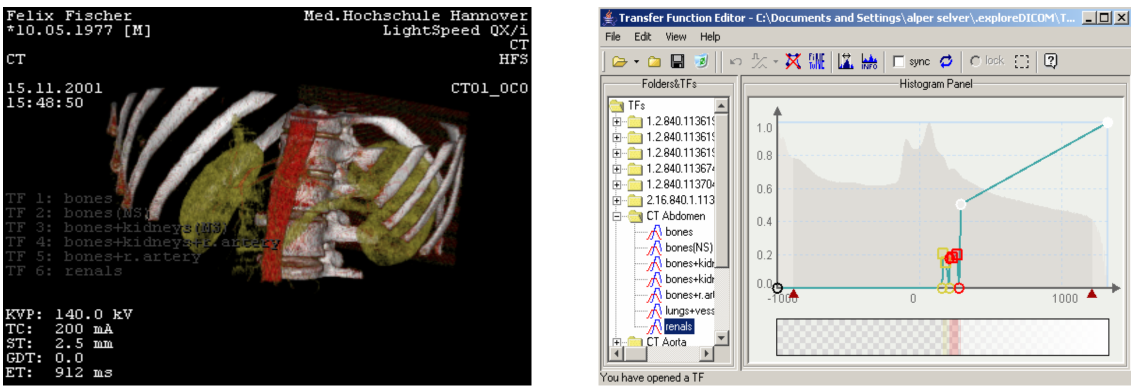
Similar to material percentage Transfer Functions, the range can also be defined as opacity, color or both. In Transfer Function Editor, the joint points of Transfer Functions are defined as nodes, by which the user can define the shape of the Transfer Function. These nodes have color property (if they are color nodes), and by changing their position in y direction, the user can adjust the opacity of corresponding HUs which affect the visibility of the corresponding body part. As color changes from one node to another, an interpolation is done to assign color for the HUs between two nodes.

Considering figures 2.6 and 2.7, it can be seen that, for Hounsfield values greater than 270, only the bones are visible. For Hounsfield values greater than 159, the bones, kidneys, and aorta are all visible.



(a) (b)

Figure 2.6 (a) 3D image (b) Image in (a) as obtained for HU>270



(a) (b)

Figure 2.7 (a) 3D image (b) Image in (a) as obtained for HU>159

Several Transfer Function files have been prepared as default by using Hounsfield values. The main categories of this default Transfer Function files are Abdomen, Aorta, Skull, Neck and Renals. Transfer Function Editor also provides the user with the functionality of adding new default Transfer Function files or changing the existing ones.

Even when the same program is used at the same computer, if the users log in as different users, the individual manipulations on Transfer Function files will be stored automatically in different personal folders of different users. Since every user uses his/her own directory, confusion that can occur due to changes on the default or any Transfer Function files by different users can be prevented.

CHAPTER THREE
PROGRAMMING ENVIRONMENT

3.1. Java Programming Language

Java programming language is based on the need of a system that must enable the development of *secure, high performance,* and *highly robust* applications on *multiple platforms* in *heterogeneous*, *distributed networks*, due to the massive growth of the Internet and the World-Wide Web.

The system that is developed to meet these needs should be

- *simple*, so it can be easily programmed by most developers;
- *familiar*, so that developers can easily learn the programming language;
- *object oriented*, to take advantage of modern software development methodologies and to fit into distributed client-server applications;
- *multithreaded*, for high performance in applications that need to perform multiple concurrent activities, such as multimedia;
- *interpreted*, for maximum portability and dynamic capabilities.

3.1.1. The Java Platform

A *platform* is the hardware or software environment in which a program runs. Some of the most popular platforms are Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

The Java VM is the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as GUI widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*.

3.1.2. Platform Independence

With most programming languages, either compilation or interpretation of a program is done so that it can run on the computer it is compiled/interpreted. In Java programming language, a program is both compiled and interpreted. With the compiler, the program is first translated into an intermediate language called *Java bytecodes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java bytecode instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. Java bytecodes can be thought as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM.

Java bytecodes help make "write once, run anywhere" possible. Programs can be compiled into bytecodes on any platform that has a Java compiler. The bytecodes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on a Windows 2000 PC, a Solaris workstation, or an iMac.

Native code is code that after it is compiled, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time bytecode compilers can bring performance close to that of native code without threatening portability.

3.1.3. Object Orientedness

The Java programming language is designed to be *object oriented*. The needs of distributed, client-server based systems coincide with the encapsulated, message-passing paradigms of object-based software. To function within increasingly complex, network-based environments, programming systems must adopt object-oriented concepts. Java technology provides a clean and efficient object-based development platform.

Programmers using the Java programming language can access existing libraries of tested objects that provide functionality ranging from basic data types through I/O and network interfaces to GUI toolkits. These libraries can be extended to provide new behavior.

3.1.4. Robustness and Security

The Java programming language consists of compile-time checking, followed by a second level of run-time checking. Language features guide programmers towards reliable programming habits.

The memory management model is based on the rule that objects are created with the new operator. There are no explicit programmer-defined pointer data types, no pointer arithmetic, and automatic garbage collection. This simple memory management model eliminates entire classes of programming errors that bedevil C and C++ programmers.

Java technology is designed to operate in distributed environments, which means that *security* is of paramount importance. With security features designed into the language and run-time system, Java based applications can't be invaded from outside. In the network environment, applications written in the Java programming language are secure from intrusion by unauthorized code attempting to get behind the scenes and create viruses or invade file systems.

3.1.5. High Performance

Performance is an important consideration for a programming language. The Java platform achieves good performance by adopting a scheme by which the interpreter can run at full speed without needing to check the run-time environment.

The *automatic garbage collector* runs as a low-priority background thread, ensuring a high probability that memory is available when required, leading to better performance. Applications requiring large amounts of compute power can be designed such that compute-intensive sections can be rewritten in native machine code as required and interfaced with the Java platform.

3.1.6. Architecture Neutral and Portable

Java is designed to support applications that will be deployed into heterogeneous network environments. In such environments, applications must be capable of executing on a variety of hardware architectures. Within this variety of hardware platforms, applications must execute atop a variety of operating systems and interoperate with multiple programming language interfaces. To accommodate the diversity of operating environments, the Java Compiler generates *bytecodes*--an *architecture neutral* intermediate format designed to transport code efficiently to multiple hardware and software platforms. The interpreted nature of Java technology solves both the binary distribution problem and the version problem; the same Java programming language byte codes will run on any platform.

Architecture neutrality is just one part of a truly *portable* system. Java technology takes portability a stage further by being strict in its definition of the basic language. Java technology puts a stake in the ground and specifies the sizes of its basic data types and the behavior of its arithmetic operators. Your programs are the same on every platform--there are no data type incompatibilities across hardware and software architectures.

The architecture-neutral and portable language platform of Java is known as the *Java virtual machine*. It's the specification of an abstract machine for which Java programming language compilers can generate code. The Java virtual machine is based primarily on the POSIX interface specification--an industry-standard definition of a portable system interface. Implementing the Java virtual machine on new architectures is a relatively straightforward task as long as the target platform meets basic requirements such as support for multithreading.

3.2. Basics of Object Oriented Programming

3.2.1 What Are Objects?

Object technology is a collection of analysis, design, and programming methodologies that focuses design on *modeling* the characteristics and behavior of objects in the real world. The objects are software programming *models*. In everyday life, we're surrounded by objects: cars, coffee machines, trees, etc.

Software applications contain objects: buttons on user interfaces, spreadsheets and spreadsheet cells, property lists, menus, etc. These objects have *state* and *behavior*. A programmer can represent all these things with software constructs called objects, which can also be defined by their *state* and *behavior*.

For example, a *car* can be modeled by an object. A car has *state* (how fast it's going, in which direction, its fuel consumption, and so on) and *behavior* (starts, stops, turns, slides, and runs into trees). The *coffee machine* can also be modeled as an object. It has *state* (water temperature, amount of coffee in the hopper) and it has *behavior* (emits steam, makes noise, and brews a perfect cup of *java*).

An object's *behavior* is defined by its *methods*. Methods manipulate the instance variables to create new state; an object's methods can also create new objects. Figure 3.1 represents a graphical representation of an object.

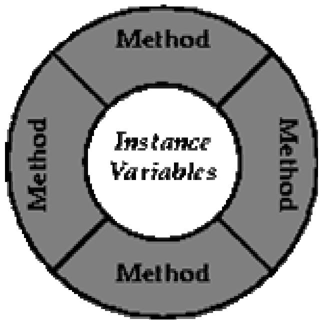


Figure 3.1 Graphical representation of an object

An object's *instance variables* (data) are packaged, or encapsulated, within the object. The instance variables are surrounded by the object's *methods*. With certain well-defined exceptions, the object's methods are the only means by which other objects can access or alter its instance variables. In Java, classes can declare their instance variables to be public, so that the instance variables are globally accessible to other objects.

3.2.2 Object Technology in Java

Four characteristic of an "object oriented" programming language are:

- *Encapsulation*--implements information hiding and modularity (abstraction)
- *Polymorphism*--the same message sent to different objects results in behavior that's dependent on the nature of the object receiving the message
- *Inheritance*-- define new classes and behavior based on existing classes to obtain code re-use and code organization
- *Dynamic binding*--objects could come from anywhere, possibly across the network. The need is being able to send messages to objects without having to know their specific type at the time the code is written. Dynamic binding provides maximum flexibility while a program is executing

As an object oriented programming language, Java meets all these requirements

3.2.3. Node Object

Node object is the basis object, as being the key element that forms a TF. As a TF is created via a number of nodes, an instance of a TF Object is created via a number of instances of node objects. The attributes of a node object are:

- X-coordinate (Hounsfield or gray-level value)
- Y-coordinate (Opacity value between 0-1)
- Color (if the node is a color node)
- InterpolatedColor (if the node is an opacity node)
- Size (for graphical representation of a node on the screen)

Since instances of node object will be elements of instances of TF object, Node object implements Serializable which allows saving operation.

The attributes and methods of the Node object are shown at the UML diagram in Figure 3.2.

Node
x : double y : double nodeColor : java.awt.Color size : integer interpolatedColor : double
Node (double, double) Node (double, double, Color) getColorInterpolated ( ) setColorInterpolated ( ) getColor ( ) setColor ( ) getX ( ) setX ( ) getY ( ) setY ( )

Figure 3.2 Node Object

3.2.4. TF Object

TF object mainly consist of several instances of the Node object. Moreover, it has attributes that specifiy each instance of the TF object such as:

- Folder Name (includes the folder name of the TF file)
- TF Name (defines the name of the TF file)
- tfElements (a vector that consists of instances of the Node object)
- offsetToHounsfieldValues (defines an offset for difference between minimum value in data array and minimum Hounsfield value. This is used when TF is transferred to Visualisation Toolkit (VTK))
- seriesInstanceUID (defines a unique number, which is equal to the seriesInstance UID of the series, for an instance of the TF object)

Since instances of TF object will be saved as TF files, TF object implements serializable which allows saving operation. The attributes and methods of the TF object are shown at the UML diagram in Figure 3.3.

TF
folderName : String tfName : String tfElements : vector offsetToHounsfieldValues : integer tfColor : Color seriesInstanceUID : String
TF (TF) TF (String, String) TF (String, String, Vector) setTFName (String) getTFName ( ) setFolderName (String) getFolderName ( ) getTransferFunction (Hashtable, Hashtable) getSeriesInstanceUID ( ) setSeriesInstanceUID (String) setOffsetToHounsfieldValues (int) getOffsetToHounsfieldValues ( )

Figure 3.3 TF Object

3.3. Programming Structure of Transfer Function Editor (TFE)

TFE is programmed by object oriented technique, provides several advantages. TFE consists of 8 packages (Appendix B). These packages are designed to work together in an object oriented manner. This means the changes made in one package does not affect any other packages. The contents of these can be listed as follows:

1. **tfe:** Consists of three classes : Starter, Main and Constants. Main class is written only for testing TFE independent of any other program. If a main visualization program starts the TFE, it starts from Starter class. Constants class is designed to contain all constant variables including tooltips, warnings, status bar messages, icons, etc…

   It is important to note that, ***'static'*** keyword is only used in 'Constants' class, which is in fact an interface, but nowhere else. ***'static'*** keyword provides flexibility but decreases object orientedness of a program. Writing the code without ***'static'*** keyword gives the opportunity to open as much TFE as possible. It is guaranteed that each of them works independent of the other ones. Therefore, the user can make as much visualization as the hardware allows and can interact with each of them independently.
2. **tfe.gui:** Contains GUI elements but not their functionalities.
3. **tfe.gui.functions:** Contains methods that is provided by GUI elements
4. **tfe.io:** Builds the file tree by searching the operating system's related directories and prepare it for being displayed in the browser.
5. **tfe.tf:** Contains TF and Node objects which are described in 3.2.4 and 3.2.5.
6. **tfe.tf.functions:** Contains the methods provided by GUI elements which create a new TF/Node instance or change an instance via these methods
7. **tfe.utilities:** Is responsible of creating icons which will be used on the toolbar and setting "look and feel style". "Look and feel style" changes depending on the operating system. This functionality improves the user friendliness of the TFE
8. **tfe.viewing:** Consists of all the classes which are responsible for TF Panel drawing. This includes TF, histogram, labels, grids etc…

   A more detailed programming structure of TFE can be found in Appendix B.

3.4. Communication with ExploreDICOM

TFE is communicating with ExploreDICOM via some methods. These methods can be grouped in two ways.

The first group consists of the methods that carry information and data to TFE. These methods and their contents are described below:

- ***setDataArray (int[ ]) / getDataArray ( ):*** This method receives/sends the histogram data in an array of integer values.
- ***setDataRangeMax (int) / getDataRangeMax ( ):*** This method receives/sends the maximum gray level value for the data array for MR values and maximum Hounsfield value for CT values. Since Hounsfield values which are greater than 3000 do not contain any information on anatomy, the values bigger than 3000 are skipped by the TFE and dataRangeMax is set to 3000.
- ***setDataRangeMin (int) / getDataRangeMin ( ):*** This method receives/sends the minimum gray level value for the data array for MR values and minimum Hounsfield value for CT values. Since Hounsfield values which are smaller than -1000 do not contain any information on anatomy, the values smaller than -1000 are skipped by the TFE and dataRangeMax is set to -1000.
- ***getHistogramDataRange ( ):*** This method sends the data range of histogram values, which is calculated as dataRangeMax-dataRangeMin+1. This method is used as a controller for the changes made to dataRangeMin and dataRangeMax.
- ***setSeriesInstanceUID (String) / getSeriesInstanceUID ( ):*** This method receives/sends the seriesInstanceUID string of the series which is being visualized. As explained in Appendix A and Part 1.1.3, this value is used to specify Transfer Functions that are designed for specific DICOM series.

Before explaining the second group of methods, which provides the sending of Transfer Function specification results, it is better to explain the internal structure of communication. First of all, in Starter class, an instance of observer object is created and then attached to the toolbar. As the user presses 'apply TF' button, these observers are notified by 'notifyObservers ( )' method which is a natural method of Observer class.

After notifying, the second group of methods are activated by ***getTF()*** method. With this method, the active TF, which is being displayed on TF panel, is send to the main program, i.e. exploreDICOM. TF object has a method called ***getTransferFunction()***, which returns two hashtables. The first hashtable consists of node coordinates and color values while the second one consists of node coordinates and opacity values. The main program uses this information and combines it in an appropriate format for sending it to Visualization Toolkit (VTK). Figure 3.4 shows the code of ***getTransferFunction()*** method.

```
public void getTransferFunction(Hashtable htColor, Hashtable htOpacity) {

 //   System.out.println("offset:"+getOffsetToHounsfieldValues());
 //   System.out.println("hashTables:"+htOpacity);

    if (htColor == null || htOpacity == null) return;
    // Color
    for(int i=0;i<this.tfElements.size();i++){
        Node dummyNode=(Node)(this.tfElements.get(i));
        if(dummyNode.getColor()!=null)
            htColor.put(new Integer((int)dummyNode.getX()), new java.awt.Color(dummyNode.getColor().getRed(), du
    //      System.out.println("htColor["+i+"] : "+dummyNode.getColor());
    }
    // Opacity
    for(int i=0;i<this.tfElements.size();i++){
        Node dummyNode=(Node)(this.tfElements.get(i));
        if(dummyNode.getColor()!=null)
            htOpacity.put(new Integer((int)dummyNode.getX()), new java.awt.Color(dummyNode.getColor().getRed(),
        else
            htOpacity.put(new Integer((int)dummyNode.getX()), new java.awt.Color(0, 0, 0, (int)(dummyNode.getY()
    // System.out.println("htOpacity["+i+"] : "+dummyNode.getY());
    }
}
```

Figure 3.4 Corresponding JAVA code for data communication with exploreDICOM

Two last methods are ***getPresetsForSeries()*** and ***checkPresets()*** methods. The ***getPresetsForSeries()*** finds the root directory of the TF files and calls the ***checkPresets()*** method. This method finds all the TF files that have the same ***seriesInstanceUID*** with the image series that is being visualized. This method returns a vector of TF files, like ***getPresetsForSeries()*** which allows the main program to display default TF files and specified TF files without opening TFE.

CHAPTER FOUR
TRANSFER FUNCTION EDITOR

4.1. Importance of User Interface for Finding Transfer Functions

As previously described in Chapters 1 and 2, volume visualization has the potential to significantly reduce the amount of time to segment medical data and, therefore, several methods have been developed and tested. Various optimization and acceleration techniques have also been developed for volume rendering. To take advantage of these improvements in interactive visualization, an effective user interface is needed.

Since the only interactive part of volume visualization is TF specification, it is important to design an easy to use tool for handling this parameter specification. Although several attempts have been made, ease of use is still an issue. Fast, robust techniques to create color and opacity transfer functions are needed to move volume rendering from the lab to the hospital. Even though the technique (automatic, semiautomatic, or manual) is particularly important to obtain efficient results; they must be presented in a simple and user friendly way.

4.1.1. An Introduction to User Interfaces for Current Methods

For efficient transfer function specification, user must:

1. know which function is useful
2. be able to easily experiment with parameters necessary for the visualization.

However:

1. searching for a good set of parameters can be time-consuming and frustrating,
2. rendering a scene after specifying new parameters can take several seconds,
3. user interfaces to specify parameters are not always carefully designed or tested.

Since automatic and semi-automatic methods cannot take the advantage of user intuition, these methods use limited user control. Thus user input is easier because of constraints on parameters, but exploring the entire parameter space is no longer possible.

Another alternative is to have the computer generate many different transfer functions and sample images and allow the user select useful ones. This effectively changes the user's search from an abstract mathematical one to a visual one. However, this strategy requires substantial user testing that has not yet been done. Also removing user from the data exploration process may be counter productive.

Investigations, for new user interfaces that allow a more traditional data-centric transfer function approach, without loosing the intuition typically present in image-centric transfer function design approaches, are currently going on; however no new design is fixed yet. As mentioned above, image and data based methods both need user testing because providing users with more information about the data, such as histograms and similar metrics, may help them to choose good transfer functions, but there exist no user studies that have been done neither generally nor for medical environments to verify this (Jankun et al., September 2001). It is also practically difficult to find and arrange medical experts, who are using volume visualization during their daily clinical work, and make them test these new approaches which are based on complicated mathematical properties and do not really present user friendly GUIs.

4.1.2. Our Approach

This study has been done to implement a tool that will satisfy the requirements mentioned above and be used for transfer function specification in medical environments as well as for general use. Therefore, a user (medical expert) and task based design is realized (Tory & Torsten, January/February 2004). As being so:

- User knowledge, experience, and thought processes,
- Physical environment of the user,
- Working style of the user,
- Tasks to be performed by the system,

- Problems that the user would like the system to solve,

are considered as constraints. As a task analysis, detailed functional specifications and user interface limitations have been determined. As required by the task analysis, interviews with potential users have been made before and during the design process.

Since trial and error approach is used in this study, an effective user interface, which will improve the weak points of trial & error approach (mentioned in Part 1.4), is required. To provide this, several functionalities and options have been developed based on both new ideas and published experiences.

4.2. Graphical User Interface

The Transfer Function Editor (TFE) user interface consists of six main regions as illustrated in Figure 4.1. This includes the Title Bar, the Menu Bar, the Toolbar, the Browser, the Transfer Function Panel, and the Status Bar. A detailed description of these regions is provided at Appendix A- i.e. Transfer Function Editor Manual. In this section, the needs and reasons of implementing the functions of TFE are described. In appendix A, the way of using these functions are explained.

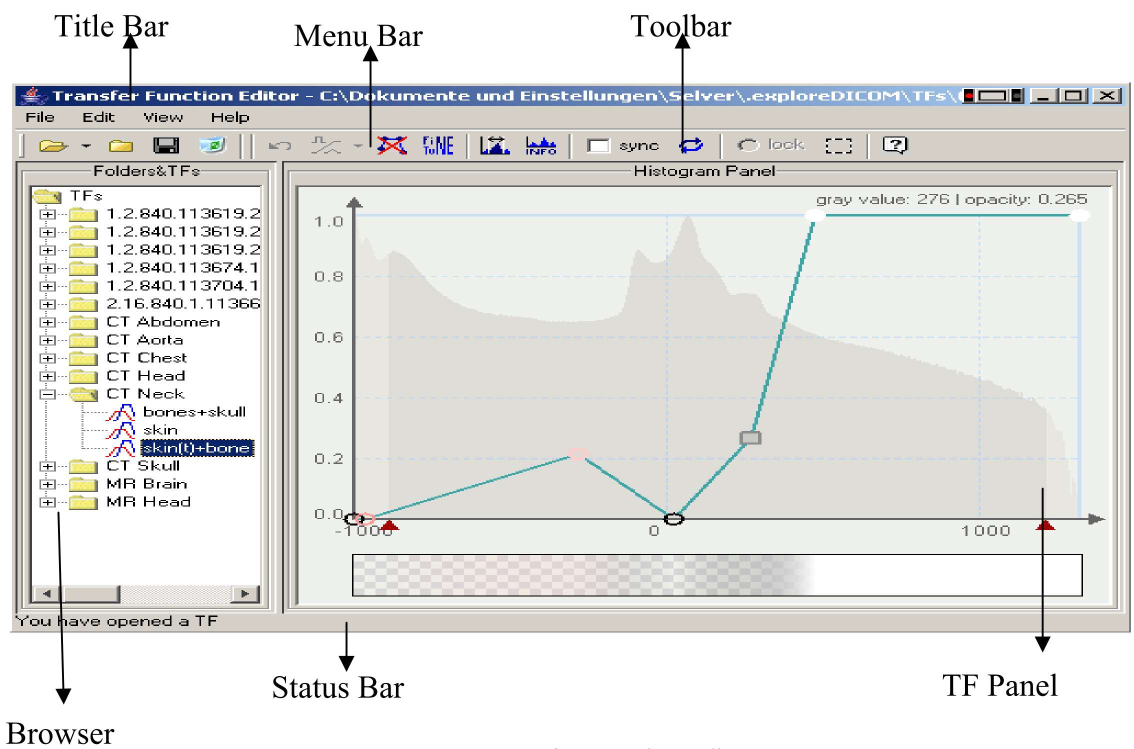


Figure 4.1 Transfer Function Editor

Flexibility of the TFE in terms of resizing and moving is found out to be important due to the reason of having different display opportunities of the medical experts. Some experts have multiple monitors or very big screen size opportunity (i.e. users of diagnostic workstations) while some only have an ordinary display (i.e. users of diagnostic workstations). Therefore, TFE is designed so that the user can maximize, minimize, or adjust the TFE to any size that is proper for interaction. The user can also change the size of the transfer function panel, relative to browser and active presets panel without changing the size of the TFE. This helps the user to focus on the area that is being monitored at that moment. During resizing, histogram and TF sizes change automatically to fit on the area and the toolbar can also be removed from TFE or can be taken outside of TFE for ease of use. The main ideas behind implementing corresponding GUI elements are summarized below:

The Title bar provides the user to see the name and directory path of the active transfer function file which is displayed on the transfer function panel. The Menu bar provides pull down menus along the top of the editor under the following headings: File, Edit, View and Help. The File menu provides functionalities for opening, closing, loading, saving, and deleting Transfer Function files and folders. The Edit menu provides options for applying active Transfer Function to volume data and applying the undo function. The View menu provides functionality for displaying labels, grids, texture and logarithmic histogram. It is also possible to make the editor to sit always on top of the screen. The Help menu provides information on using the Browser and Transfer Function panel. The toolbar is on interface area, directly below the menu bar that displays pictorial representation of different available functionalities for Browser (the first four buttons) and Transfer Function Panel. The Browser is the panel on the interface area, on the left side, below the toolbar. The Browser panel provides functionalities for creating, opening, closing, deleting, and saving Transfer Function files and folders. The Transfer Function panel lies on the right side of the Browser directly under the toolbar. Transfer Function panel provides functionalities for manipulating transfer functions. Histogram of the volume data is also displayed on this panel. The Status bar lies along the bottom of the editor and gives information about the last changes made on the transfer function panel.

4.2.1. The Title Bar

The title bar allows the user to see the active transfer function, which is displayed on the Transfer Function panel. As illustrated in Figure 4.2, the phrase displayed on title bar shows the exact location of the active transfer function file and the type of the file in brackets.

Figure 4.2 The Title Bar

There are two types of transfer function files:

The first type is "Global" functions which are valid for all studies of the same kind (e.g. CT Kidney, CT Aorta etc. saved in "Global" mode). When one of these kinds of transfer functions is active, then "Global Function" phrase will be displayed in brackets.

The second type is "Only for the series" functions which are specially made for only one series and positioned in the folder, which is named with the series instance UID of that series (saved in "Only for this series" mode). When one of these kinds of transfer functions is active, then "Series` Function" will be displayed in brackets,

4.2.2. The Menu Bar:

As illustrated in Figure 4.3, the menu bar provides four pull down menus (File, Edit, View, and Help) which are described in details at Appendix A.

Figure 4.3 The Menu Bar

The File menu contains options that allow the user to create, open, close, delete and save transfer function files or folders and exit the editor. As illustrated in Figure 4.-Appendix A, the options are enabled or disabled based on the selected folder or transfer function file on the browser. This menu provides an easy to use interaction between user and file system. Therefore, user can interact with files and folders as he/she used to do with well known files (i.e. MS Word/Excel document). Since TF specification is a time consuming and subjective procedure, previously well defined TFs help the user to find an appropriate 3D view faster and better. This is important for the continuity and fast remembering of previous studies done by user or any other expert which provides more efficient work.

The Edit menu contains the options that allow you to apply the transfer functions and undo function. It mostly consists of the functionalities of the toolbar and designed to provide these functionalities when user removes the toolbar.

The View menu contains the options that allow the user to change the display of the transfer function panel in terms of histogram, labels, grids, and textures. Labeling menu option allows the user to make labels that show Hounsfield values in x-direction and opacity in y-direction to be either visible or invisible. Since these labels are generally needed to create TFs, this option is selected as default. Grids and textures are designed to provide better visual effects on the screen by using grids that make the TF panel screen look more like a scaled area and textures that allow the user to see the opacity of the nodes better. This helps the user understand in advance how that color will be seen on 3D view.

Logarithmic Histogram: This menu option allows the user to see the histogram in logarithmic scale (in opacity y- direction). This option is useful in two cases:

1) When the histogram has (a) very strong peak(s), which makes the rest of the histogram data not clearly visible. Logarithmic view of histogram allows the user to see the histogram in detail which makes the intense distribution of Hounsfield values more visible. This option is not selected as default to prevent confusion.

2) It is observed that when rendering data sets with relatively large regions of roughly uniform density, the resulting images are most sensitive to detailed changes in the transfer function when the opacity is set nearly to zero. Editing the transfer function within the lowest five to ten percent of the range often results in the most

visible differences in the rendered images, allowing us to differentiate finer structures in the data, while larger values corresponded to renderings that appeared nearly opaque, thus obscuring large portions of the present finer structures. However, for smaller, thinner regions of some density in the data, it is sometimes necessary to use higher opacity values to make the region to appear in the rendering at all.

These observations about volume renderings are due to *compositing,* an iterative process that closely approximates the occlusion along a viewing ray in a volume. It turns out that when even a small opacity value is assigned to a region in the data of any significant size, the final rendered intensity will quickly increase to near-opaque, and its color will be based primarily on to the sampled voxels closest to the viewer. Each sample is obscured in part by every sample in front of it, resulting in a contribution to the intensity that is exponentially smaller as distance through the volume increases.

It is proposed that a better approach is obtained by observing that it is possible to scale the vertical axis of the graph, though the horizontal axis scale is fixed by the rendering method (alpha compositing). If the transfer function opacities are scaled logarithmically, the result is a graph in which the regions of intensity are better distributed without "pushing" any of them off of the graph. Consequently, it would be easier, both conceptually and physically, for a user to precisely control the intensity of a region in the volume. This should result in more efficient transfer function editing, which is significant since transfer function editing is one of the most time-consuming aspects of creating clear and informative renderings of medical, mathematical or generally scientific data sets.

The Help Menu provides two options which are **How to use TF Browser** menu option that pops up a dialog box which gives information and tips on the usage of the browser and **How to use TF Editing Area** menu option which gives information and tips on the usage of the transfer function panel, transfer function manipulation, and editing.

4.2.3. The Toolbar

The Toolbar allows the user to apply or activate different functionalities for the Browser (the first four buttons) and Transfer Function Panel. The Toolbar is illustrated in Figure 4.4. The icons are designed to be simple and as meaningful as possible. In addition, some common icons that are not legally protected are used as the first four and last buttons. Main effort is given to present easy-to-understand icons on the toolbar.

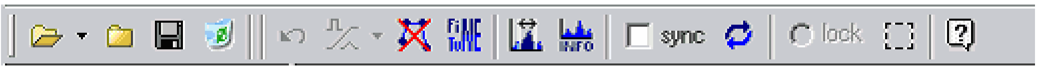


Figure 4.4 The Toolbar

The first buttons of the toolbar are related to Browser (File System) and is explained in Part 4.3.-File System. Following ten buttons are placed for helping TF creation and manipulation. The last button is for Help Menus. The functions and importance of these buttons are described in detail at below and the way of using them is explained in Appendix A.

Undo: Since TF specification based on trial & error approach requires several trials, it is important for the user to turn back if he/she does not find the manipulation steps good enough or generally thinks he/she is on a wrong way (i.e. removing a node to previous HU value is easier with undo button than manually removing back). Undo toolbar option allows the user to undo the last actions made on the transfer function panel (inserting a node, removing a node, etc.). The user can go as much steps back as possible.

This function does not undo any operation except the ones that is done on the transfer function panel because it can be confusing and hard to follow if the user makes changes in the file system (i.e. delete, save, etc.) between two operations done at the transfer function panel.

Insert Default TF: Some standard forms of TFs (i.e. Step, Ramp, Triangle, etc…) are generally constructing the basic form of a TF. To support this basic TF shapes, an option allows the user to insert one of the six default transfer functions on the toolbar. To prevent confusion, it is only enabled when all nodes are removed from the transfer function panel.

Reset**:** Resetting option is placed for easy clearing of TF panel. If the user decides to change the TF completely or wants to start a new TF, this option provides quick removing.

Fine Tune: Since many of the tissues lie in a very narrow range of HU values (i.e. soft tissues as white matter, grey matter, and CSF), it could be hard to put the nodes of TF to the exact desired position. It may also be easier to change the coordinates of all nodes if a TF is previously designed with exact coordinates. This toolbar option provides a dialog box, which shows the Hounsfield values and opacity values of the nodes of the TF. The user can change the Hounsfield value or opacity of a node by directly filling the corresponding fields with numerical values. The Hounsfield value areas for the first and the last nodes are not editable because they have to be fixed to the minimum and maximum Hounsfield values of the volume data, respectively.

Histogram Range Scaling**:** As described above, moving a node to an exact position can be a difficult task when the user is dealing with a range that consists of several nodes. Fine Tune panel provides a solution to put all the nodes to exact locations; however, it is not very practical to try several positions one after another. Therefore, this toolbar option allows the user to adjust the range of the histogram which is displayed on the transfer function panel. By using the pointers shown at Figure 4.6, the range can be changed to ease the manipulation of TF.

Histogram Info**:** This toolbar option allows the user to see the minimum and maximum Hounsfield values of volume data, histogram data range and the maximum point of the histogram. This information is especially important to see the resolution of CT.

Apply TF and ***sync*****:** This toolbar option allows the user to apply the active transfer function, which is displayed on the transfer function panel, to the volume data. If the user checks the checkbox on the left side of this button (labeled as "sync"), the transfer function will be applied automatically whenever the user changes a parameter in the transfer function (dragging a node, changing color of a node, etc.). The "sync" option should be used when powerful computers are used for visualization. Otherwise it will be time consuming to apply TF at each parameter change.

ROI**:** It is also determined that the user may want to change a part of TF that consists of several nodes, without changing their relative positions. On the other hand, only this specific part may be wanted to be scaled (i.e. enlarge, narrow). For instance, a step shaped part of a TF, as illustrated in Figure 4.1, may be wanted to be shifted without changing its shape or enlarged/narrowed without changing the relative distances of nodes to each other. ROI option provides this by giving the opportunity of selecting the nodes with a region of interest and moving/scaling them after locking.

Help: This menu option opens a dialog box which gives information and tips on the usage of the browser, the transfer function panel, TF manipulation, and editing.

4.2.4. The Status Bar:

The status bar lies along the bottom of the editor and gives information about the last changes made on the transfer function panel. This helps the user to follow his/her steps when manipulating and using "undo" function.

Figure 4.5 The Status Bar

4.3. The Transfer Function Panel and Transfer Function Manipulation

The transfer function panel is designed for allowing the user to easily view and manipulate transfer functions. By using this panel, the user can manipulate the transfer functions in terms of adding/removing new nodes and changing opacities/colors. Figure 4.6 shows the transfer function panel and its elements.

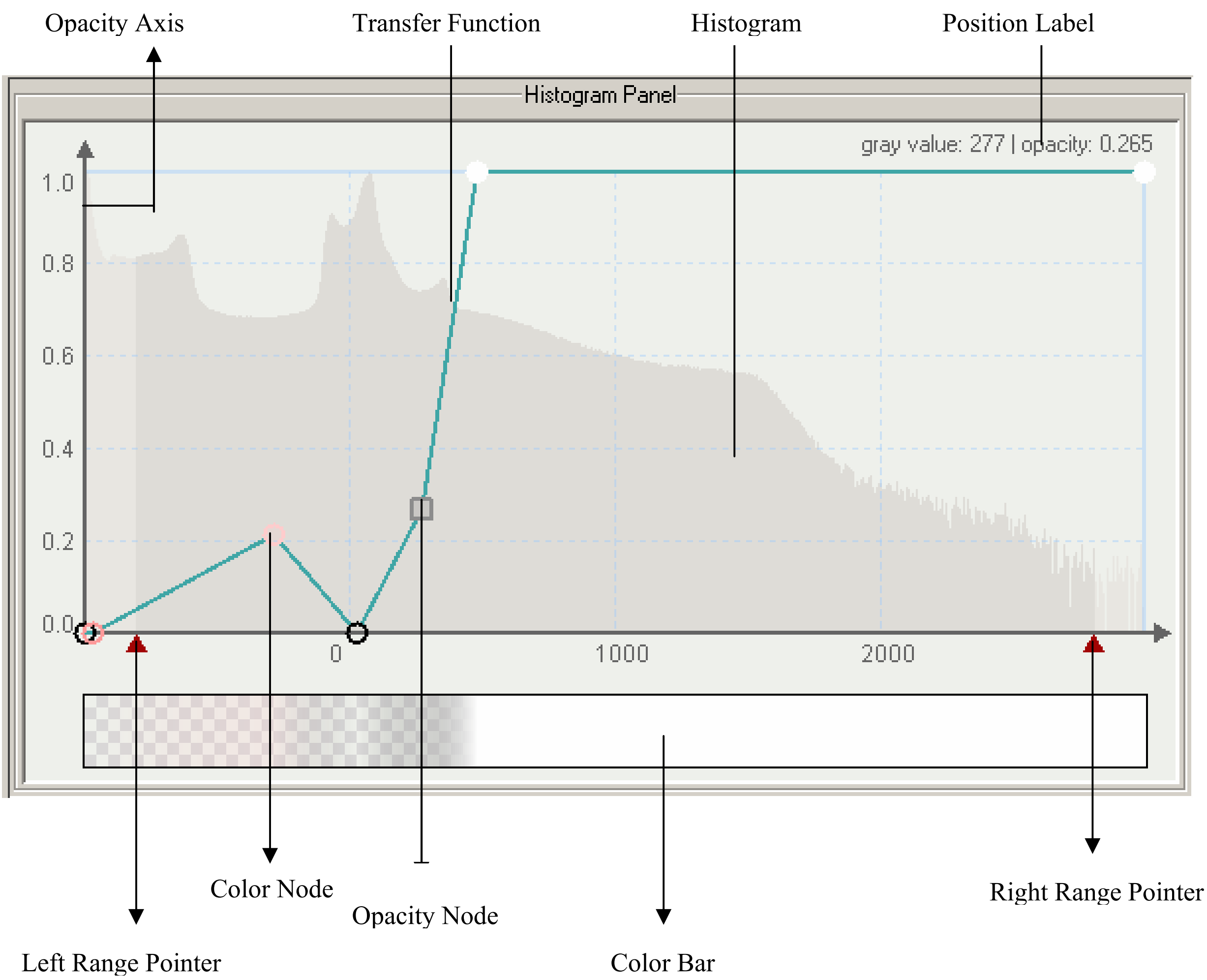


Figure 4.6 Transfer Function Panel

Manipulating the transfer function in terms of adding or removing nodes is a relatively simpler task as a GUI. In TFE, this is done by using a pop-up menu (Figure 4.7) that appears with the right click over the area bounded by the domain (HU values)-range (opacity values) surface of transfer function. By using the pop-up menu, user can add or remove color and opacity nodes to the TF. It is also possible to change a(n) color/opacity node to a(n) opacity/color node, simply by right clicking over the node and selecting the related option from pop-up menu. It is also possible to change the color of a node (if it is a color node) by using the same approach. The last functionality of the pop up menu is the ability to remove all the nodes of TF. These seven options are found out to be enough to manipulate transfer function, therefore the simplest way to realize these options is implemented via the pop-up menu

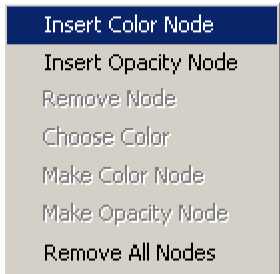


Figure 4.7 The transfer function panel pop-up menu

Although the same pop-up menu appears on everywhere over the transfer function panel, the options are enabled or disabled due to the position of the mouse on the panel (i.e. the mouse is over a node or over an empty area).

Right mouse button can be used to drag a node of the TF. Simply by pressing right mouse button over a node –user is notified via a symbol change from default cursor to hand cursor- and dragging it.

More important issue on designing the transfer function panel is to represent color and opacity change via an easy to understand/use interface. Several different user interfaces have been designed to solve this problem with an efficient way. One of them is to represent color variation and opacity variation on different graphs. However this leads to confusion due to the control of two functions instead of one. Another disadvantage is that the user cannot see directly if a color will be visible or not (because of the corresponding opacity value). Also using two different graphs divides the editing area which limits the user.

Concerning these issues, color and opacity change is realized at the same graph in TFE. This is done by giving a color attribute to the (color) node and representing it with its color on the screen. When user changes the opacity value of a node, its color does not change in the program; however, the user sees the color relative to the node's opacity. This is visually represented by changing the transparency filling color on the screen as the user drags the node. The frame and frame color of the nodes do not change because, otherwise, it would be impossible to see a fully transparent node on the screen.

The approach described above gives the user the ability to easily change opacity and color values with visual feedback however it is not sufficient because user is unaware of the overall color variation. Therefore, a colorbar is designed and put at the bottom of the transfer function panel. This colorbar shows which color corresponds to which HU value and its visibility. If a color is opaque, it will be seen clearly on the colorbar. As the color of a HU value becomes transparent, the background texture of the colorbar becomes more visible, warning the user about the visibility of corresponding color.

At the background of the transfer function panel, histogram plot is given as background to inform the user about the intense of HU values in the volume. Histogram does not help the user to see which part of the volume image will be affected by that color; instead, it gives the quite important information of how many pixels will be affected due to a color/opacity change. At the right top part of the transfer function panel, two information labels occur when a node is being dragged by the user. They show the HU value and opacity of the dragged node to help the user in placing the node to exact position. Two pointers are also used to help the user in defining the range to be displayed. Pointers are doing this in two ways. The first way is when the user adjusts the right and left pointer. Only the area between these two pointers is focused and the rest of the histogram is ignored. The second way is when the user presses ***Histogram Range Scaling*** button which is described in page 36, only the corresponding part of the histogram and the transfer function is displayed to help the user to focus on that area where several tissues may be placed. Pointers can be dragged in a similar way to dragging of nodes.

4.4. Browser and File System

The Browser is designed to allow the user to easily control (i.e. create, save, delete, rename, etc.) TF files. Control over the file system is a necessary function that should be provided.

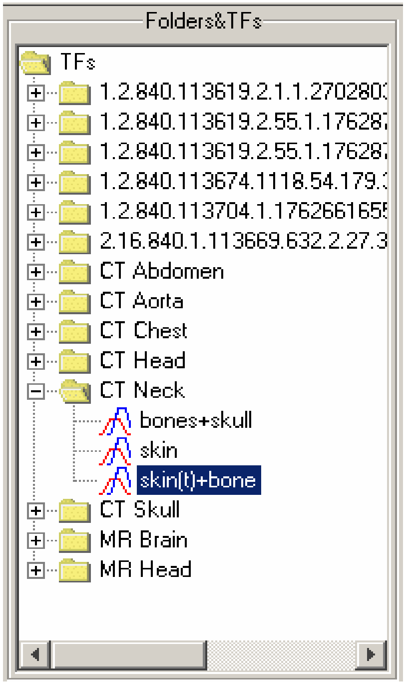


Figure 4.8 The Browser

This necessity comes from two reasons.

The first reason is the need for a file format by which the user can make the necessary changes in an easy and flexible way. To provide this, TF files are designed to be as a common file format (i.e. MS Word file, MS Excel file, etc.). After creating a TF file, the user can save, delete, rename, open or close it using the browser. TF files can also be renamed, deleted, copied, or replaced directly by using the operating system (OS) operations. If OS operations are used, the browser can be updated by using the browser pop-up menu-refresh button. After refreshing, the changes that have been made in OS will be visible. If TFE is opened after these changes, the browser is updated automatically.

When TFE is first installed, a directory with the name 'TF' is created under logged user's working area. Under this folder, default TF files that are previously designed based on HU values are placed. The user can use these default TF files or can create new ones. Under the main TF folder, the user can also create new directories.

The second reason is the need for a history tool. Since TF specification is a time consuming procedure, TF files which were previously found useful, help the users find an appropriate TF function. The file control system provided by TFE is easy to use and flexible that helps the user in the systematical storage of TF files. With this opportunity, the browser can also be used as a history tool. Since every logged user automatically uses the directory that belongs to his/her account, he/she can store TF files as they want. Even when all the users use the same account, they can create their own directories and store the TF files.

All these functionalities can also be reached by using the File drop-down menu and the Browser pop-up menu. The user is guided by the pop-up menu as in the TF panel pop-up menu. The browser pop up menu appears when the user clicks the right mouse button over the browser. For instance, if a TF file is selected on the browser, 'create Folder' option will be disabled since a folder cannot be created under a TF file. However 'close TF' option will be enabled since it is a valid functionality of a TF file. This example is illustrated in Figure 4.9.

Figure 4.9 The Browser pop-up Menu

4.5. DICOM Gray Scale Softcopy Presentation State and Our Approach

The DICOM standard defines in detail how medical images can be transmitted and stored. However, there have been no precise rules on how to interpret the parameters contained in a DICOM image which deal with image presentation. Since DICOM image is a sequence of bits which are displayed on the monitor after several image processing steps, lack of standardization and archiving processes (i.e. window level/width) can cause problems; for example, the same image could frequently look different when displayed on different workstations or in general, a consistent, reproducible image display between different display devices is not guaranteed.

Therefore, three DICOM extensions have been added to the standard:

- Grayscale Standard Display Function (GSDF)
- Presentation Look Up Table (PLUT)
- Gray Scale Softcopy Presentation State (GSPS)

4.5.1. Grayscale Standard Display Function (GSDF)

Consistency between images (softcopy and hardcopy or displayed on different monitors) can only be guaranteed if monitor and printers are calibrated with a standard calibration which includes:

1. Measurement of characteristic curve, i.e. real luminance of the display device at different gray levels
2. Correction (by means of software or hardware) by adaptation to a standardized reference curve.

The DICOM standard defines such a reference curve GSDF, which describes the human visual perception with a mathematical model, defines p-values that standardize value range for gray scales and applicable to different types of display devices.

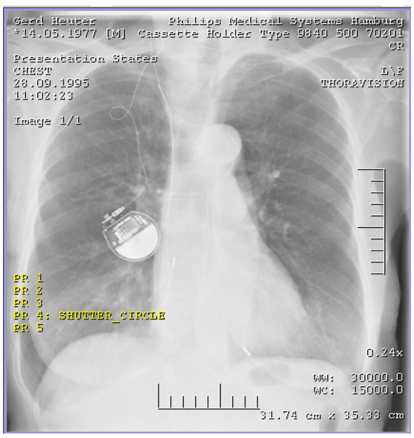
4.5.2. Presentation Look-Up Table (PLUT)

PLUT extends the DICOM print service with parameters that allow the GSDF model to be used on hardcopy devices. PLUT is an optional extension of the DICOM Print Management Service Class.

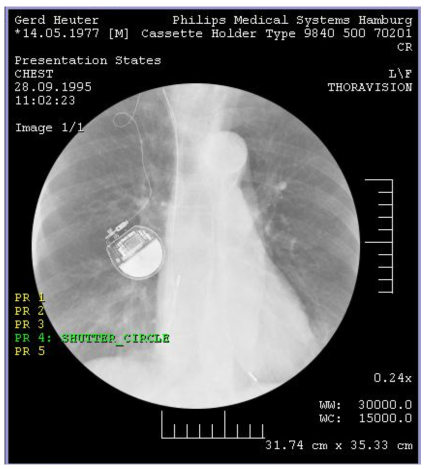
4.5.3. Gray Scale Softcopy Presentation State (GSPS)

GSPS collects all relevant information about the visual representation of a DICOM image into a separate DICOM object. To describe how an image should be presented on a softcopy display, GSPS objects precisely define all image processing steps and supports textual and graphical annotations, shutters, rotation, flipping, polarity, zoom, etc. GSPSs are separate series in DICOM study in which source images are located only via references (i.e. not copied). Source images always remain unchanged and multiple presentation states can be applied to the same image.

ExploreDICOM supports DICOM GSPS in a user friendly way. As shown in Figure 4.10 (a) below, if a GSPS object exists for the series displayed, GSPSs are displayed at the left-bottom corner of the image. When the user presses the “PR” button, corresponding GSPS is applied to the image as shown in Figure 4.10 (b).



(a)

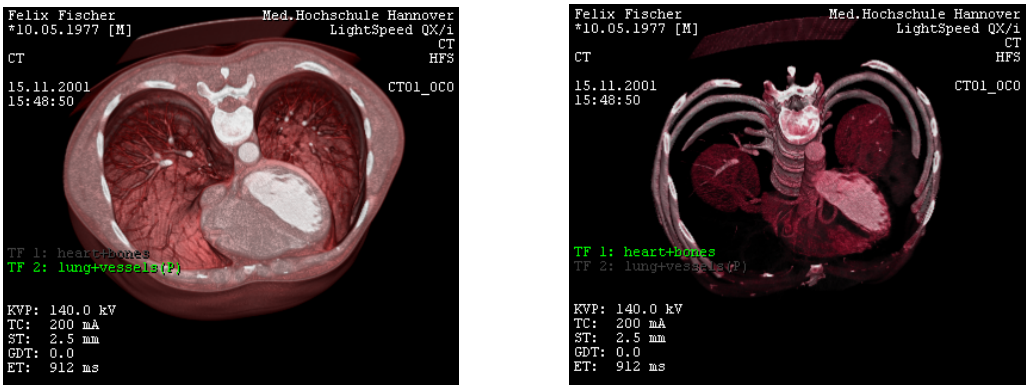


(b)

Figure 4.10 (a) GSPS object display (b) Display when GSPS is applied

4.5.4. Our Approach for 3D presentation

As GSPSs are giving the possibility to store and distribute the presentation of an image between softcopy devices in an efficient way, the same method can be applied to the 3D data via TFs. Previously found and saved TF files (by using “only for this series” mode saving as described in Part 4.3. File System-) can be seen in yellow color at the bottom left of the screen as GSPSs and can be applied to the 3D data with a mouse click, as shown in Figures 4.11 (a) and (b).



(a) **(b)**

Figure 4.11 Difference between two 3D presentation states

With this system, the user does not spend any time to see the 3D data which is previously classified by the same or another user. Since classification is a subjective method that depends on the expert’s ideas and experience, it is quite important to see the 3D image with a view that is found to be important previously by another expert.

This approach has also several advantages such as:

- Since all modalities send their images to PACS (archive) after acquisition, adjustments performed on 3D data during diagnosis can be stored as a TF file instead of saving a big size 3D image to archive that will in fact only be duplication.
- Since TFs are very small, they are well suited for 3D in Teleradiology applications such that the images could be transferred in advance and only TFs are exchanged online.

This approach can also be seen as a starting point of 3D Presentation States in such a manner that as 3D medical imaging becomes a standard, it will obviously be necessary to have an object (such as GSPS) which has information on rotation, manipulation, annotation, opacity, color, etc. Similar to GSPS, this 3D presentation state object could be stored to prevent unnecessary storage of medical data. For instance, instead of storing a full 3D image, this small object can be stored and when 3D image is displayed, this object can be used to apply the previously saved information to 3D image.

CHAPTER FIVE
CLINICAL EVALUATION

5.1. Clinical Evaaluation

Dokuz Eylül University Radiology Department has a complete digital radiology system including digital modalities, RIS, PACS, diagnostic/clinical workstations, and web viewers. 3D visualization programs are routinely in use for diagnosis and treatment planning in the department. Especially, medical experts who are working on brain and abdominal studies are more frequently using this technology.

Twelve medical experts are asked to test the TFE and to fill an evaluation form. It is determined that the experts joining this study are using 3D visualization from 1 year up to 7 years with a frequency of 5 times up to 12 times a month. Twelve questions are asked under four headings and grading was from 1 to 5 where 1 is the best: The results for each heading are as follows:

1. GUI and ease of use: Experts find the TFE GUI elements easy to use (average evaluation value (AVV): 2) and understand (AVV:2)

2. TF Panel and specification properties: Menu options are found to be enough except the need for an information panel contains HU values with corresponding tissues.(AVV:2). The visual feedback and manipulation properties of TF are found to be sufficient (AVV:2)

3. The Browser and file system: The Browser usability, TF file format and its properties as a history tool are found out to be very helpful and necessary (AVV:1)

4. 3D Presentation states approach: The approach is found to be very helpful and useful (AVV:1)

It is found that this form is sufficient to evaluate the software. Overall evaluation of these forms shows that the initial results are very promising and optimistic for the TFE. TFE is found to be very useful for CT and angiographic datasets.

However it would be more efficient to use TFE after segmentation process that eliminates the unnecessary information from the data, especially in MR datasets where tissues are overlapping. For instance, after segmenting liver from the other abdominal organs, it is possible to assign different color and opacities to liver tissue, vessels and tumors without overlapping. This is an important example since radiologists use 3D imaging for determining vessel structure in liver tissue and to determine the surgical procedure with surgeons before liver transplantation operations.

CHAPTER SIX
DISCUSSION

6.1. Discussion

In this study, a Transfer Function Editor (TFE) has been developed for medical volume visualization applications.

Limitations of the transfer function specification, especially spatial information drawback, have been overcome by taking efficient advantage of Hounsfield values. So, developed technique works much better on CT and angiographic studies. Due to high tissue overlapping in MR studies, tissue classification is mostly very time consuming and even not possible in some cases.

A user friendly and easy to use GUI has been implemented for providing fast adaptation of medical experts to the software. Based on recent research activities on transfer function specification and human factors in visualization, several properties and functionalities have been developed as follows:

- TF files that can be used to store useful transfer functions for global or study specific future usage.
- The Browser that has the capability of handling TF files and all file system in a similar manner provided by operating system.
- The Toolbar and the Transfer Function Panel that provide a broad range of visual and control flexibility for manipulation and design of transfer functions.

In literature, there is no study concerning storage of useful TFs. The Browser and TF file system provides this via TF file format.

All of the functionalities and options have been implemented to overcome the frustrated procedure of trial and error method and to make it more efficient for medical experts. Other techniques that are discussed in this thesis provide some useful information to the user for better renderings. However, they are limiting the parameter space and reducing interactivity. .Our design does not provide a new method for TF specification but offers quick adjustment of parameters for a faster procedure, which is also the industry standard.

What is new about the developed software package is its flexibility and adjustability to different platforms and heterogeneous environments. Since JAVA programming language has been used in the design, the editor is object oriented andplatform independent. This makes the editor adoptable to any visualization software that provides histogram data of the image series to be visualized.

A new approach that can be named as 'simple 3D presentation states' has been implemented which provides:

- A history tool that is based on the usual file operations in the OS
- A small size file that can be used in teleradiology for transferring only TF files instead of complete images.

Moreover, user testing has also been made for usability of the developed software. Clinical evaluation of the new TFE has been made by medical experts who are currently using medical visualization in their daily treatment planning and diagnosis. Their feedback was very optimistic for the present system.

REFERENCES

Bajaj, C. L., Pascucci, V., & Schikore, D. R. (1997). The Contour Spectrum. In *Proceedings Visualization '97,* pages 167-173.

Fujishiro, I., Azuma, T., & Takeshima, Y. (1999). Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topology Analysis. In *Proceedings IEEE Visualization,* pages 467—470,563, October.

He, T., Hong, L., Kaufman, A., & Pfister, H. (1996). Generation of Transfer Functions with Stochastic Search Techniques. In *Proceedings Visualization '96,* pages 227-234.

Hounsfield, G. N.(1979), Computed Medical Imaging, *Nobel Lecture*, 8 December.

Jankun, T. J., Ma, K., & Ma, K. L. (2001). Visualization Exploration and Encapsulation via a Spreadsheet-Like Interface. *IEEE Transactions on Visualization and Computer Graphics,* 7(3):275-287, July-September 2001.

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

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

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

Konig, A., & Gröller, E (2001). Mastering Transfer Function Specification by Using VolumePro Technology. In Tosiyasu L. Kunii, editor, *Spring Conference on Computer Graphics 2001,* volume 17, pages 279-286, April 2001.

Lacroute P. & Levoy, M. (1994). Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transform. In *ACM Computer Graphics (SIGGRAPH '94 Proceedings),* pages 451-458, July 1994.

Levoy, M. (1988). Display of Surfaces from Volume Data. *IEEE Computer Graphics & Applications,* 8(5):29-37.

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 '97Proceedings),* pages 389-400, August 1997.

Pekar, V., Wiemker, R. & Hempel, D. (2001). Fast Detection of Meaningful Isosurfaces for Volume Data Visualization. In *Proceedings IEEE Visualization 2001,* pages 223-230.

Pfister, H., Lorensen, B., Bajaj, C., Kindlmann, G., Schroeder, W., Sobeierajski L. A., Martin K., Machiraju R., & Lee J. (2001). The Transfer Function Bake-Off. *IEEE Computer Graphics and Applications,* 21(3): 16-22, May/June 2001.

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

Porter, T., & Duff, T. (1984). Compositing Digital Images. In *ACM Computer Graphics (SIGGRAPH '84 Proceedings),* pages 253-259, July 1984.

Rezk-Salama, C., Hastreiter, P., Scherer, J., & Greiner, G. (2000). Automatic Adjustment of Transfer Functions for 3D Volume Visualization. In *Proceedings Vision, Modeling, and Visualization 2000,* pages 357-364, November 200.

Shiaofen, F., Tom, B., & Mihran, T. (1998). Image-Based Transfer Function Design for Data Exploration in Volume Visualization. In *Proceedings IEEE Visualization 1998,* pages 319-326,546. IEEE, October 1998.

Siegel E. L., & Kolodner R. M. (Eds.) (1999). Filmless Radiology, New York, Springer-Verlag, 1999.

Tory, M., & Torsten, M. (2002). Human Factors in Visualization Research, *IEEE Transactions on Visualization and Computer Graphics,* Vol. 10, No.1, January/February 2004

APPENDICES

APPENDIX A. TRANSFER FUNCTION EDITOR MANUAL

1. INTRODUCTION

The Transfer Function Editor user interface consists of six main regions as illustrated in Figure 1. This includes the Title Bar, Menu Bar, the Toolbar, the Browser, the Transfer Function Panel and the Status Bar. A detailed description of each of these regions is provided at Part 2-User Interface.

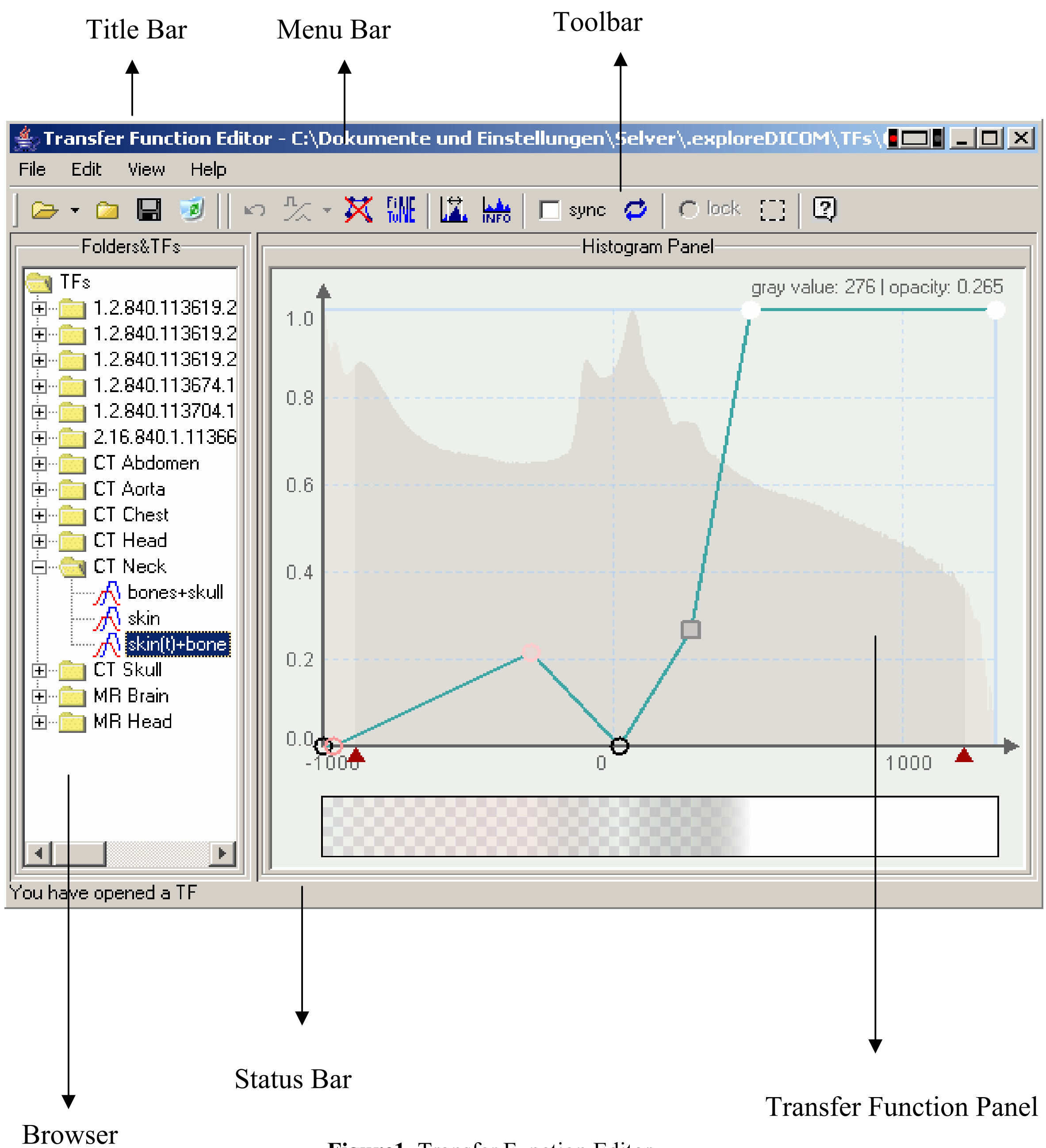


Figure1. Transfer Function Editor

Title Bar : The title bar provides you to see the name and directory path of the active transfer function file which is displayed on the transfer function panel.

Menu Bar : The menu bar provides pull down menus along the top of the editor, under the following headings: File, Edit, View and Help.

The File menu provides functionality for opening, closing, loading, saving, and deleting Transfer Function files and folders. The Edit menu provides options for applying active Transfer Function to volume data, applying "Apply-Save TF- Exit Editor" combination and applying undo. The View menu provides functionality for displaying labels, grids, texture and logarithmic histogram. It is also possible to make the editor always on top on the monitor. The Help menu provides information on using browser and Transfer Function panel.

Tool Bar : The toolbar is on interface area, directly below the menu bar that displays pictorial representation of different available functionalities for Browser (the first four buttons) and Transfer Function Panel.

Browser : The browser is the panel on interface area, on the left side, below the toolbar. Browser panel provides the functionality for creating, opening, closing, deleting, saving Transfer Function files and folders.

Transfer Function Panel : The Transfer Function panel lies on the right side of the Browser and directly under the toolbar. Transfer Function panel provides the functionality for manipulating transfer functions. Histogram of the volume data is also displayed on this panel.

Status Bar: The status bar lies along the bottom of the editor and gives information about the last change made on the transfer function panel.

NOTE : You can change the size of the transfer function panel, relative to browser without changing the size of the transfer function editor. For resizing, you can use the vertical splitter between browser and transfer function panel. When you are over the splitter the mouse cursor will change from default cursor to "double sided arrow" cursor. When you press left mouse button over splitter, you are able to drag the splitter to right or left. As you drag the width of the browser and the transfer function panel will change according to the change.

The browser has scroll panes that appear when the name of a folder or transfer function file does not fit into browser's width and/or height (otherwise scroll pane(s) are invisible). When the size of the transfer function panel is changed, the histogram, grids, labels and the transfer function is automatically resized. This may cause changes in grid width, labels etc. This is done automatically for best visual view. Also you can change the place of the toolbar which is explained in part 2.3.

2. USER INTERFACE

2.1. Title Bar : The title bar allows you to see the active transfer function, which is displayed on the Transfer Function panel. As illustrated in Figure 2. The phrase displayed on title bar shows exact location of active transfer function file and the type* of the file in brackets.

Figure 2. The Title Bar

*There are two types of transfer function files:

The first type is "Global" functions which are valid for all studies of same kind (i.e. CT Kidney, CT Aorta etc.) (saved in "Global" mode). When one of these kinds of transfer function is active, then "Global Function" phrase will be displayed in brackets,

The second type is "Only for the series" functions which are specially made for only one series and positioned in the folder, which is named with the series instance UID of that series. (Saved with "Only for this series" mode) When one of these kinds of transfer function is active, then “Series` Function” will be displayed in brackets,

2.2. Menu Bar: As illustrated in Figure 3, the menu bar provides four pull down menus (File, Edit, View, Help) which are described in details below.

Figure3. The Menu Bar

2.2.1. The File Menu: The File menu contains options that allow you to create, open, close, delete and save transfer function files or folders and exit the editor. As illustrated in Figure 2 below, the options are enabled or disabled due to the selected folder or transfer function file on the browser.

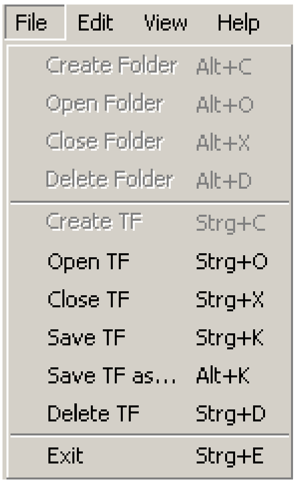


Figure4. The File Menu

Create Folder : This menu option will pop up an input dialog box. When you enter the name for the folder you want to create and press “OK”, a new folder will be created and appear on the browser.

Open Folder: This menu option will pop up the File Chooser Dialog Box. You can select the folder you want to open and press “OK” button. When you open a folder, it will appear on the browser.

Close Folder: This menu option will close the folder which is selected on the Browser. The closed folder and the transfer function files under this folder will disappear from Browser.

Delete Folder : This menu option will delete the folder which is selected on the browser. All the transfer function files under this folder will also be deleted and disappear from the browser panel.

Create TF : This menu option will pop up an input dialog box. When you enter the name for the transfer function file you want to create and press "OK", a new transfer function file will be created. When you create a transfer function file, it will appear on the browser and the active transfer function which is displayed on transfer function panel will be changed to the created file.

Open TF : This menu option will pop up the File Chooser Dialog Box. You can select the transfer function file you want to open and press "OK" button. When you open a transfer function file, it will appear on the browser and the active transfer function which is displayed on transfer function panel will be changed to the opened file.

Close TF : This menu option will close the transfer function file which is selected on the Browser. The closed file will disappear from Browser.(If the closed transfer function file is the active file which is displayed on the transfer function panel, the it will also disappear from the transfer function panel.)

Save TF : This menu option will save the active transfer function file which is displayed on the transfer function panel. (If active transfer function file is not saved before, this button will work as "Save TF As" option which is described below)

Save TF As : This menu option will pop up a selection box which allows you to save the transfer function as "Global" or "Only for the series that you are working on".

If you choose the "Global" option a Save As dialog box will be displayed and the active transfer function on the screen will be saved **under the folder you select** with the name you entered.

If you choose the "Only for this series" option, a Save As dialog will be displayed and the active transfer function on the screen will be saved **under the folder, which has the name with the Series Instance UID of the series,** with the name you entered. If there is no folder with that name, a folder will automatically be created. With this saving mode, you will be able to see the saved transfer function near the 3D image (on 3D image panel) whenever you open that series, which allows you to apply that transfer function to volume data without opening the Transfer Function Editor.

In both modes of saving, the active transfer function which is displayed on the transfer function panel will be changed to the new transfer function file and the new file will appear on the browser.

Delete TF : This menu option will delete the transfer function file which is selected on the browser and it will disappear from the browser panel. .(If the deleted transfer function file is the active file which is displayed on the transfer function panel, the it will also disappear from the transfer function panel.)

Exit : This menu option will pop up a selection box. If you select "Yes", the transfer function editor will be closed. (3D image will be displayed with the last transfer function you have applied to the volume)

2.2.2. The Edit Menu: The Edit menu contains the options that allow you to apply the transfer functions and undo function.

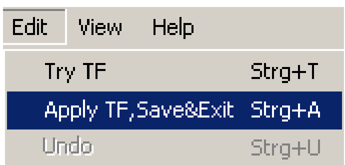


Figure 5. The Edit Menu

Try TF : This menu option allows you to apply the active transfer function, which is displayed on the transfer function panel, to the volume data.

Apply TF, Save & Exit : This menu option allows you to apply the active transfer function, which is displayed on the transfer function panel, to the volume data. Then automatically the active transfer function will be saved and the transfer function editor will be closed. If the active transfer function is not saved before(so that it does not have a file name; a Save As dialog box will be open for saving the transfer function. You can get more information about "Save As" function from previous part of this document(page 4).)

Undo : This menu option allows you to undo the last action made on the transfer function panel (inserting a node, removing a node etc.) This function does not undo any operation except the ones on the transfer function panel. When you press undo button the change will be shown on the status bar. The undo button will be disabled(as illustrated in Figure 5), when undo function is not possible.

2.2.3. The View Menu : The View menu contains the options that allows you to change display of the transfer function panel in terms of histogram, labels, grids and textures; to refresh browser and to keep the editor always on top on the screen.

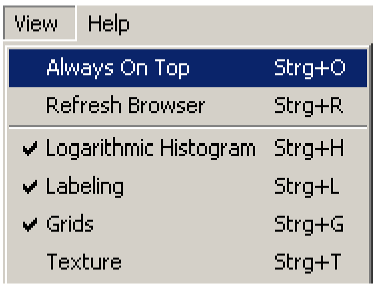


Figure 6. The View Menu

Always On Top : This menu option will keep the transfer function editor always on top on the screen. This feature is especially useful when you want to see the 3D image at the same time you are changing the parameters on the transfer function panel. This option is not selected in default.

Refresh Browser : This menu option allows you to refresh the browser. Refreshing browser is especially useful in two cases:

If you change the file system in TFs directory manually (this means copying, deleting, creating a transfer function file or folder without using browser or menu functions but using operating system's (i.e. Windows commands) , refreshing browser allow you to see the changes in the browser without closing and re-opening the transfer function editor.

If you close some transfer function files and/or folders and you want to make them visible again, this can also be done by using this option.

Logarithmic Histogram : This menu option allows you to see the histogram in logarithmic scale(in opacity -y- direction). This option is useful when histogram has a very strong peak(s) which causes the rest of the histogram data not clearly visible. This option is not selected in default.

Labelling : This menu option allows you to make labels, which show hounsfield values in x direction and opacity in y direction, to be visible or invisible. This option is selected in default.

Grids : This menu option allows you to make grids, which are automatically drawn with dashed blue lines for better visual effect, to be visible or invisible. This option is selected in default.

Texture : This menu option allows you to make texture, which appears at the back of the histogram to point the opacity of a node, to be visible or invisible. This option is not selected in default.

2.2.4. The Help Menu : The help menu provides two options which are described below. The help menu is illustrated in Figure 6.

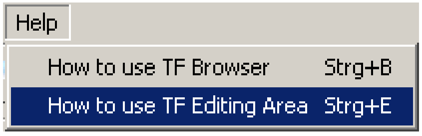


Figure 7. The Help Menu

How to use TF Browser : This menu option pops up a dialog box which gives information and tips on the usage of the browser.

How to use TF Editing Area : This menu option opens a dialog box which gives information and tips on the usage of the transfer function panel, transfer function manipulation and editing..

2.3. The Toolbar : The toolbar allows you to apply or activate different functionalities for Browser (the first four buttons) and Transfer Function Panel. The toolbar is illustrated in Figure 7. By clicking the line, which lies vertically at the most left side of the toolbar, you can change the position of the toolbar. After clicking this line, you can position the toolbar to any place on the screen or reposition the toolbar vertically to the right or left side of the transfer function editor, by dragging.

Figure 8. The Toolbar

 **Open Folder / TF** : This toolbar option pulls down a menu, on which you can select to open a folder or transfer function file. When you select the Folder/File Chooser Dialog Box will pop up. You can select the folder/transfer function file you want to be opened and press "OK" button. When you open a folder/transfer function file, it will appear on the browser.

NOTE : Open Folder menu option is enabled only when the main folder (TFs) is selected on the browser. Open TF menu option is enabled only when a folder is selected on the browser and the transfer function file you will open must be in this selected folder (otherwise a pop up dialog box will warn you)

 **Close Folder / TF** : This toolbar option allows you to close the folder/transfer function file which is selected on the Browser. The closed folder/file

will disappear from Browser.(If the closed transfer function file is the active file (or under the closed folder) which is displayed on the transfer function panel, the it will also disappear from the transfer function panel.)

Save TF : This toolbar option allows you to save the active transfer function file which is displayed on the transfer function panel. (If active transfer function file is not saved before, this button will work as "Save TF As" option which is described in page 4)

Delete TF : This toolbar option allows you to delete the folder (and all the files included)/transfer function file which is selected on the browser and it will disappear from the browser panel. .(If the deleted transfer function file is the active file, which is displayed on the transfer function panel, or the active transfer function file is under the deleted folder, it will also disappear from the transfer function panel.)

Undo : This toolbar option allows you to undo the last action made on the transfer function panel (inserting a node, removing a node etc.) This function does not undo any operation except the ones on the transfer function panel. When you press undo button the change will be shown on the status bar. The undo button will be disabled(as illustrated in Figure 5), when undo function is not possible.

Insert Default TF : This toolbar option allows you to allow you to insert one of the six default transfer functions. When you press this button, a pull down menu, which includes pictorial representations and names of six default transfer functions, will appear. It is only enabled when all nodes are removed from the transfer function panel.

Reset : This toolbar option opens a confirmation box. If you press yes, all the nodes will be removed from the transfer function and only "Insert Default TF" button will be enabled. You can either insert a default transfer function by using this button or switch to another transfer function file.

Fine Tune : This toolbar option opens a dialog box, which shows the hounsfield values and opacity values of the nodes of the transfer function. You can change the hounsfield value or opacity of a node by directly filling the corresponding fields. The hounsfield value areas for the first node and the last node are not editable because they have to be fixed to the minimum and maximum hounsfield values of the volume data, respectively. If you enter invalid value(s) for hounsfield values or opacities, then a dialog box will pop up for warning and all values will be set back to the previous valid values.

Histogram Range Scaling : This toolbar option allows you to adjust the range of the histogram which is displayed on the transfer function panel. This option is especially useful when you need to control and fine tune large number of nodes in a small range of hounsfield values. When you press this button, the histogram range, which is displayed on the transfer function panel will be changed to the range defined by the range pointers(shown in Figure 11). After adjusting the place of the pointers(described in part E), pressing this button will allow you to see only the histogram range and transfer function nodes between two pointers.

Histogram Info : This toolbar option allows you to see the minimum and maximum hounsfield values of volume data, histogram data range and the maximum point of the histogram.

Apply TF : This toolbar option allows you to apply the active transfer function, which is displayed on the transfer function panel, to the volume data. If you check the checkbox on the left side of this button (labelled as “sync”), the transfer function will be applied automatically whenever you change a parameter in the transfer function (dragging a node, changing colour of a node etc.)

ROI : This toolbar option allows you to move more than one node at the same time. This property is especially useful when you want to move a group of nodes from one hounsfield range to another, without changing the distance between them. When you enable ROI button, you can draw a ROI by dragging the left mouse button over the transfer function panel. The nodes which are inside the

ROI will be selected when you press the “lock” radio button on the left side of the ROI button. After locking the ROI, you can resize or move the ROI. The position of the nodes inside the ROI will change proportional to the change (in terms of size or position) on the ROI.

Help **:** This menu option opens a dialog box which gives information and tips on the usage of the browser, the transfer function panel, transfer function manipulation and editing.

2.4. The Browser: As illustrated in Figure 9, the browser allows you to create, open, close, delete, save Transfer Function files and folders via browser pop-up menu (Figure 10) and menu bar. Menu bar options are described at pages 3-6.

The browser pop up menu appears when you click right mouse button on the browser. The options are enabled or disabled due to the selected point on the browser. The statuses of the options on the pop up menu is shown in Table 1 for all possible cases.

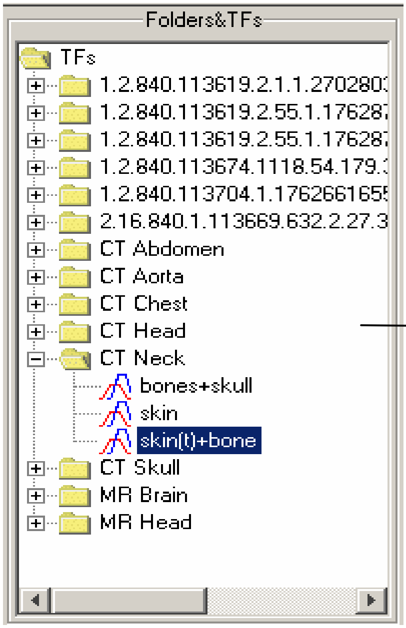


Figure 9. The Browser

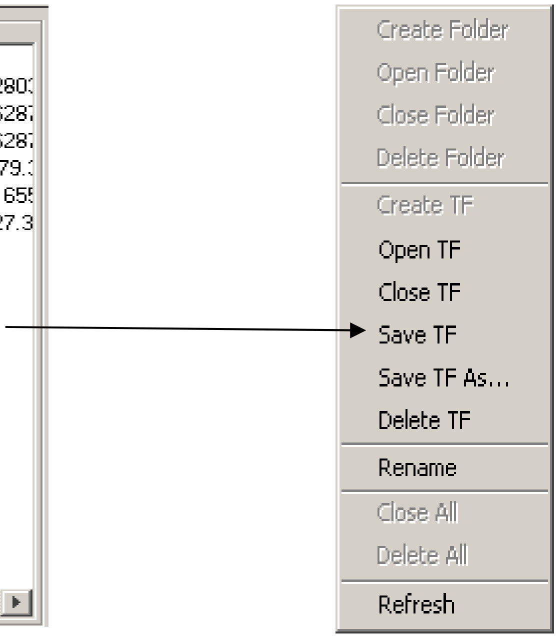


Figure 10. The pop-up Menu for the Browser

Selected Node / Menu Option	**Main Folder**	**Folder**	**TF File**
Create Folder	√	X	X
Open Folder	√	X	X
Close Folder	X	√	X
Delete Folder	X	√	X
Create TF	√	√	X
Open TF	√	√	√
Close TF	X	X	√
Save TF	X	X	√
Save TF As…	X	X	√
Delete TF	X	X	√
Rename	X	√	√
Close All	X	X	X
Delete All	X	X	X
Refresh	√	√	√

Table 1. Statuses of browser pop up menu

All the browser pop up menu options are functioning as they are described in menu bar (pages 3-7). You can refer to those explanations except the option "Rename" described below :

Rename : This option allows you to change the name of a folder/transfer function file.

2.5. The Transfer Function Panel : The transfer function panel allows you to view and manipulate transfer functions. By using this panel, you can manipulate your transfer functions in terms of changing opacity and colour. The elements of the panel are illustrated in Figure 10. and the ways of manipulating a histogram is described in quickstart part (Part-3). In this part the transfer function panel pop up menu options will be described.

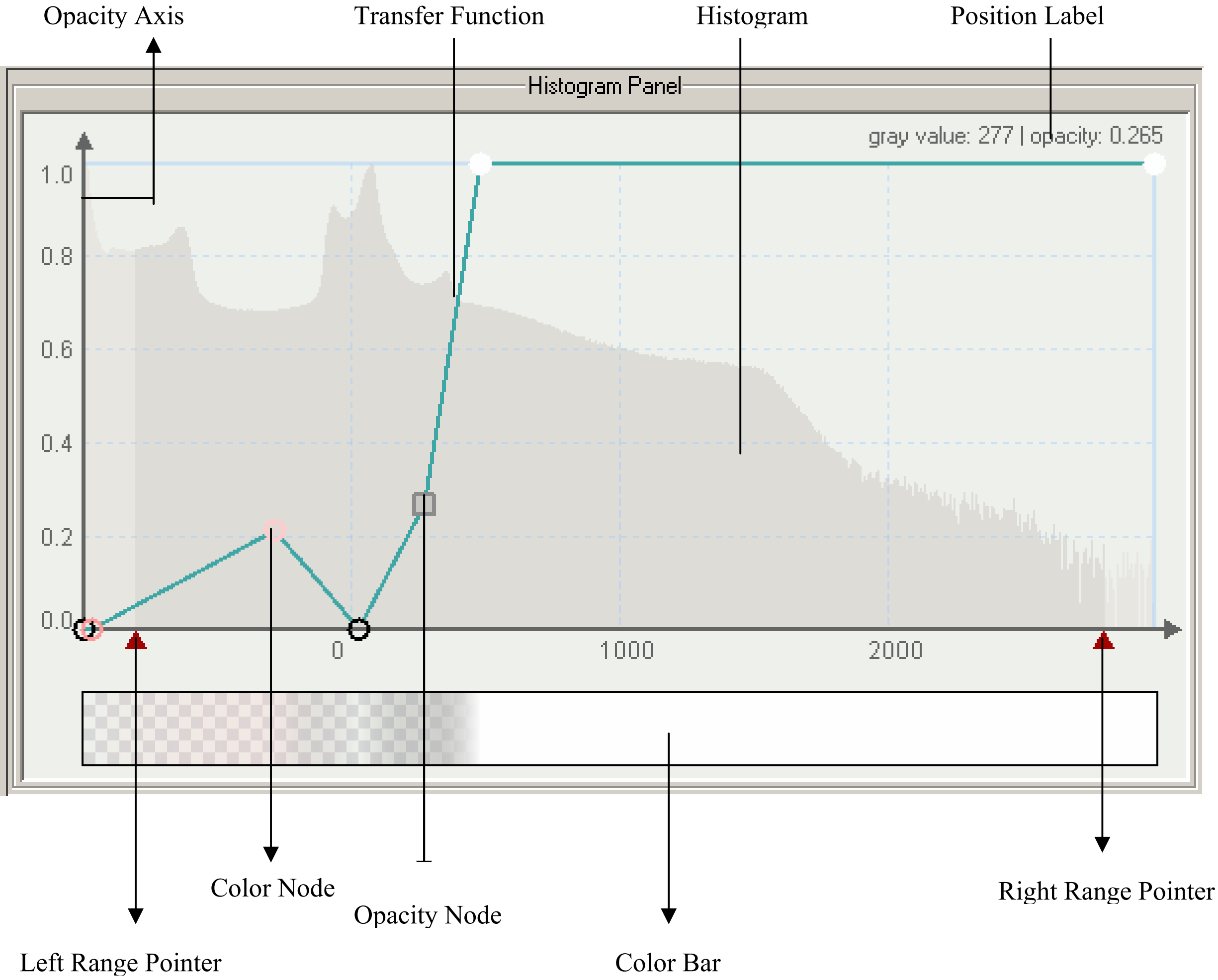


Figure 11. Transfer Function Panel

The transfer function panel pop up (Figure 11.) menu appears when you click right mouse button on the transfer function panel. The options are enabled or disabled due to the position of the mouse on the panel. The statuses of the options on the pop up menu is shown in Table 2 for all possible cases.

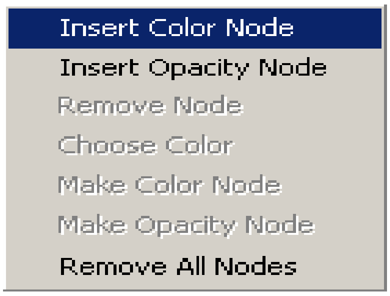


Figure 12. The transfer function panel pop up menu

Mouse Position / Menu Option	Over Colour Node	Over Opacity Node	Elsewhere
Remove Node	√	√	X
Choose Color	√	X	X
Make Color Node	X	√	X
Make Opacity Node	√	X	X
Remove All Nodes	√	√	√

Table 2. Statuses of transfer function panel pop up menu

Insert Color Node: This option allows you to insert a color node to the position where the mouse cursor enters the panel after pressing this button. If the mouse enters outside of the panel or an invalid place for node inserting, then no nodes will be inserted.

Insert Opacity Node: This option allows you to insert an opacity node to the position where the mouse cursor enters the panel after pressing this button. If the mouse enters outside of the panel or an invalid place for node inserting, then no nodes will be inserted.

Remove Node: This option allows you to remove the node, which you positioned the cursor over, before pressing right mouse button to open pop up menu. You cannot remove the first node and the last node of the transfer function, unless you use “remove all nodes” option, to keep the transfer function valid.

Choose Color: This option allows you to change the color of a color node. You can change the color of a node by directly selecting one of the colors from the palette, by using HSB palette or entering HSB values or by entering RGB values.

Make Color Node: This option allows you to change an opacity node to a color node.

Make Opacity Node: This option allows you to change a color node to an opacity node.

Remove All Nodes: This option allows you to remove all of the nodes of the transfer function. Before moving all the nodes a confirmation dialog box will appear. When you press "YES", all nodes will be removed and transfer function panel pop up menu options and toolbar buttons (except Insert Default TF button) will be disabled, unless you have opened a new TF or insert a default transfer function.

2.6. The Status Bar: The status bar lies along the bottom of the editor and gives information about the last change made on the transfer function panel. The status bar does not give information about any changes on browser or other user interface elements except transfer function panel. But if you undo any operation which was previously made on transfer function panel, the change will appear on the status bar.

Figure 13. The Status Bar

3. QUICK START

- When you open the transfer function editor, an unsaved (unnamed) default transfer function will be available on the panel. You can simply continue with manipulating this function or you can double click to the file you want to open(or use open TF button) to open any transfer function file.
- When you have the transfer function, that you want to start with, on the screen, you can manipulate the transfer function by inserting nodes, changing node colors etc. by using transfer function panel pop up menu.
- You can watch the color spectrum of your transfer function by using the color bar shown in Figure 11. This color bar represents color and opacity changes from one node to another in your transfer function. As transparency increases the background of the color bar is more visible and vice versa. This affect is also available for the nodes of your transfer function. As you increase the opacity of a node it is filled with its color. As you decrease the opacity of a color it becomes more transparent. You can always understand the color of your node by looking at its border color, because the border color of the nodes does not change with the opacity of the node. Although color property is only available for color nodes, transparency nodes are painted with a gradient color, which is calculated due to their position between color nodes.
- Whenever you want to see the affect of your manipulations, you can press "Apply TF" button on the toolbar. (Or if you want to see the affects of your changes synchronously, you can check the "sync" checkbox)
- If you are working on a very narrow range, then you can focus to the range by adjusting range pointers (shown in Figure 11) and pressing "Histogram Range Scaling" button on toolbar. If you want to move more than one node at the same time, you can use the "ROI + lock" buttons on toolbar. (Described in detail in part 2.3)
- If you are satisfied with the result, you can save your transfer function by using "Save TF" button in toolbar or "Save TF" and "Save TF As…" options in menu bar and browser pop up menu.

APPENDIX B. PROGRAMMING STRUCTURE