REMOTE QUALITY ASSESSMENT AND RESEARCH SUPPORT SYSTEM FOR ADVANCED MRI STUDIES

by

Ertuğrul Akbaş

Doctor of Medicine, Kadir Has University Faculty of Medicine, 2005

Submitted to the Institute of Biomedical Engineering
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Biomedical Science

Boğaziçi University
September 2009

REMOTE QUALITY ASSESSMENT AND RESEARCH SUPPORT SYSTEM FOR ADVANCED MRI STUDIES

APPROVED BY:

Cengizhan Öztürk (Thesis Advisor)	....
Albert Güveniş	....
Alp Dinçer	....

DATE OF APPROVAL: 09.09.2009

ACKNOWLEDGMENTS

I am greatly thankful to Assoc. Prof. Dr. Cengizhan Öztürk, for his supervising, encouragement, support and endless motivation. His endless support, goal-oriented vision and sincerity is priceless. He is also the one who gave us a speech when I was in first year in medical faculty and I'm still remembering the title of his speech 'Heart as a pump', I decided to apply Biomedical Institute that day. So he is also responsible for my career in biomedical area: thank you.

I'd like to thank Onur Özyurt and Engin Demirel for their invaluable support for this work. I'm sure that we're going to work together in many projects.

I would like to thank my dear friends, collegues and managers for their understanding and support during hard times in my thesis preperation.

I have many reasons to thank my family but I should mention about their love that I always feel which is priceless, many thanks goes to them since they gave me the chance to feel the honour of being a part of our family.

Last but surely not least, I'd like to thank Nurten as she is enlighting my life.

ABSTRACT

REMOTE QUALITY ASSESSMENT AND RESEARCH SUPPORT SYSTEM FOR ADVANCED MRI STUDIES

User interventions and quality assessments are usually necessary for diagnostic and experimental MRI studies. The main motivation of this study is to develop a software package which enables users to remotely control distant console, communicate with physician and/or technician via on-line messaging system/file transfer system and control MRI scanner software when necessary. This platform consists of main console and sub-modules. Main console and sub-modules were programmed with Borland Delphi 7 which is object oriented programming language based on object Pascal. While current sub-modules were programmed with Delphi, it's possible to include future applications which are developed with different languages/compilers. Application is based on client-server architecture and connection protocol is TCP/IP. Remote control system and its sub-modules let users/researchers connect to distant computer systems (e.g. one at laboratory, one at hospital etc) and allow them to operate remotely which could improve productivity and optimize time and error management. The main idea is development of a remote control system for quality assessment and remote support system for MRI scanners. **Keywords:** Quality assessment, Remote support, MRI, Remote control, Delphi.

ÖZET

İLERİ DÜZEY MRI ÇALIŞMALARI İÇİN KALİTE DEĞERLENDİRME VE ARAŞTIRMA DESTEK SİSTEMİ

Deneysel ve diyagnostik amaçlı MRI çalışmalarında genellikle kullanıcı müdahalesi ve kalite değerlendirmesi gerekmektedir. Bu çalışmanın amacı, uzak erişim sağlayarak gerektiğinde MRI araştırma yazılımına müdaheleye, doktor ve/veya teknisyen ile çevirimiçi haberleşme ve dosya transferine ve uzak bilgisayar kontrolüne imkan sağlayacak bir yazılım paketi geliştirmektir. Platform, ana bir konsol ve alt modüllerden meydana gelmektedir. Ana modül ve alt modüller Pascal tabanlı nesneye yönelik yazılım geliştirme platformu olan Borland Delphi 7 ile geliştirilmiştir. Her ne kadar alt modüller Delphi ile geliştirilmiş olsa da gelecekte eklenebilecek uygulamalar farklı dillerde yazılabilecektir. Uygulama istemci/sunucu mimarisi üzerinde TCP/IP bağlantı protokolü üzerinde çalşmaktadır. Uzak erişim sistem ve alt modülleri, kullanıcıların/araştırıcıların uzak bilgisayara (örn. laboratuvarda, hastanede) erişerek uzaktan kontrol gerçekleştirebilmesine olanak sağlayarak üretkenliği artırabilecek, zaman ve hata yönetimi işlemlerini optimize edebilecektir. Çalışmanın ana amacı, MRI tarayıcıları için kalite yönetimi ve uzaktan destek sistemi geliştirmektir.

Anahtar Sözcükler: Kalite değerlendirme, Uzak erişim, MRI, Uzaktan kontrol, Delphi.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

s	Second
T	Tesla

LIST OF ABBREVIATIONS

ADSL	Asymmetric Digital Subscriber Line
3G	3rd Generation
ARP	Address Resolution Protocol
ARPA	Advanced Research Projects Agency
AVI	Audio Video Interleave
CRC	Cyclic Redundancy Check
CT	Computed Tomography
DFR	Direct Fourier Reconstruction
DFT	Discrete Fourier Transform
DNS	Domain Name System
FCE	Fast Content Expression
FID	Free Induction Decay
FLASH	Fast Low-Angle Shot Imaging
fMRI	functional Magnetic Resonance Imaging
FTP	File Transfer Protocol
Hlen	Header Length
HTTP	Hyper Text Transfer Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IR	Inversion Recovery
ISO	International Standards Organization
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
MS	Microsoft
NCP	Network Control Protocol

NMR	Nuclear Magnetic Resonance
OSI	Open Systems Interconnection
RDP	Remote Desktop Protocol
RF	Radio Frequency
RGB	Red Green Blue color system
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SNR	Signal-to-Noise Ratio
STIR	Short Time Inversion Recovery
T_1	Longitudinal relaxation time constant
T_2	Transverse relaxation time constant
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/ Internet Protocol
TI	Delay time
TI_1, TI_2	QUIPSS parameters
ToS	Type of Service
TTL	The Time to Live
UDP	User Datagram Protocol
Vers	Version
VNC	Virtual Network Computing
VoIP	Voice over IP
VPN	Virtual Private Network

1. INTRODUCTION

1.1 Motivation

The main motivation of this thesis is to develop a software package which enables users to remotely control distant console, communicate with physician and/or technician via on-line messaging system/file transfer system and control MRI scanner software when necessary.

1.2 The Scope of the Study

This thesis study involves developing a quality assessment, research and technical support system. Main modules and sub-modules are listed below:

1. **Development of Server Platform:** Server program is developed to be run on a computer at distant clinical site. Server listens employed TCP/IP ports and provide necessary data to client and intervenes MS Windows handlers to let distant users remotely control the computer on which server program runs.

2. **Development of Client Platform:** Client program can be executed on any computer with MS Windows XP or Vista with appropriate user privileges. Client requests necessary data from server and sends user data when remote control is in demand.

In the next section, the theoretical and practical aspects of this system is explained in detail.

2. MAGNETIC RESONANCE IMAGING (MRI)

Magnetic resonance imaging is an imaging technique that utilizes radiofrequency electromagnetic radiation and large magnetic fields which are produced by superconducting magnets which contains superconducting wires that has minimal electrical resistance. As a tomographic imaging modality, it's possible with to obtain both structural and functional data in three dimensions with high spatial resolution. Image data are constructed digitally from frequency space. Due to complexity of the technique, scanning last longer comparing to CT (e.g. several minutes when considering those in CT: few seconds). Moreover, due to expensive components (like superconducting magnet), it's more costly than CT scanner. Following section reviews the basics of NMR and MRI. Further information and details can be found in the related reference books [1, 2, 9, 10, 11, 12, 13].

The NMR spectrum produces only a single peak if the same tissue were in two different positions (Figure 2.1). An axial image can be reconstructed with multiple projections that are obtained around a patient.

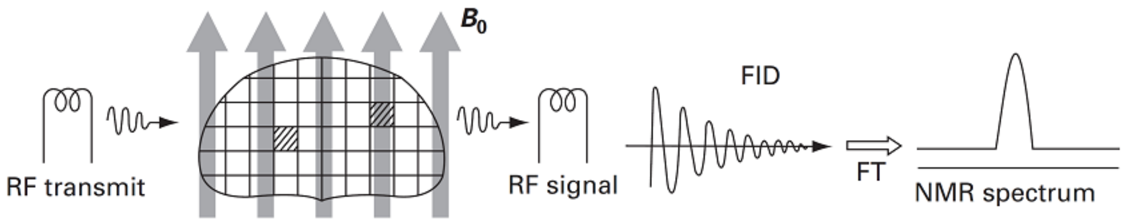

Figure 2.1 The NMR spectrum of a patient with protons at two different positions. Reproduced from [1].

With using a frequency selecting RF pulse simultaneously with a magnetic field gradient, it's available to select a slice. Sagittal, coronal or axial orientation of image can be obtained by choosing field gradient, x, y or z. Two gradients can be used to obtain an oblique slice.

For example, if it's required to acquire an axial image, slice selection along the z direction is necessary. A gradient in z direction is turned on, the resonant frequency is smaller towards the patient's feet, unchanged at the isocenter and greater along the head. By simultaneously using a special RF pulse, only a portion of spins around the isocenter is stimulated into the transverse plane. Thickness or position of the slice can be manipulated by using different RF bandwidths and gradient strengths. Slice can be moved by changing the center frequency of the frequency selecting RF pulse.

After selecting a slice, x and y directions are needed to be encoded in signal. One direction is encoded with changes in frequency during scanning and another gradient is turned on. For the other dimension, signal is encoded in terms of phase; number of gradients are needed to create phase changes so there would be enough information that will be used to encode final image. There are three separate gradient coils which are required to encode three spatial dimensions and multiple projections can be obtained by rotating the magnetic field around the subject. Each echo signals can be processed with Fourier transform and these one dimensional projections can be processed together to produce two dimensional Fourier transform of subject. Finally, inverse Fourier transformed to produce an axial image. This technique, direct Fourier reconstruction (DFR) is widely used in MRI reconstruction.

The basic components of a MRI scanner are shown in Figure 2.2. The primary magnet can be an electromagnet using superconducting coils that requires cooling to very low temperatures or it can be a permanent magnet constructed of alloys which makes it possible to have more open structure. Primary magnet is responsible for polarization of the protons in the subject. In order to produce linear variations on the magnetic field which makes proton resonant frequencies within the patent are spatially coded. RF coil is responsible for producing oscillating magnetic field which is required for creating phase coherence.

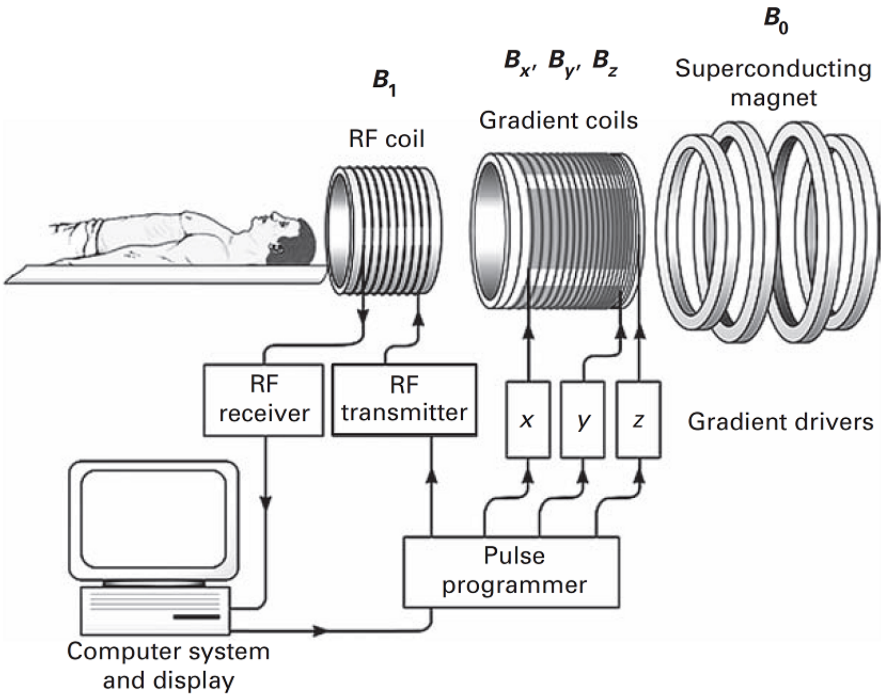

Figure 2.2 Basic components of a MRI scanner. Reproduced from [1].

2.1 Basics of MRI

MRI imaging is founded on nuclear magnetic resonance (NMR). Due to it's abundance in body, hydrogen is of high importance in MRI, all current clinical MR scanners use that proton signal. During unexcited period (e.g. in nature), magnetic moment of a nuclei is random but they line up according to direction of the magnetic field if an external magnetic force is applied and starts to precess around it (Figure 2.3). The frequency of this precession is known as Larmor frequency and is dependent to external electromagnetic field, B_0. There are two scenarios for the magnetic moments, in lower energy states they can either line up in the same direction with the magnetic field or the can line up opposite to the field in higher energy state.

While alignment direction protons are continually changes, there will be a minor number of protons that aligned parallel to the field in order to minimize overall energy. As external field increases, difference in energy level also increases which results in excess number aligned parallel to the field.

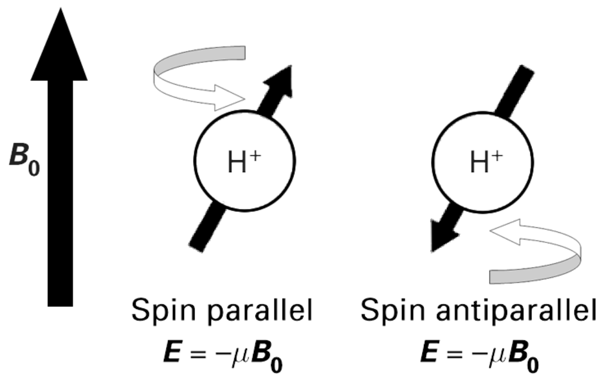

Figure 2.3 The precession of the nuclear magnetic moments of a hydrogen nucleus about an external magnetic field giving rise to two distinct states, parallel and anti-parallel. Reproduced from [1].

As energy difference between two states (high energy and low energy states) is basis of MR imaging, it's essential to apply a pulse of electromagnetic radiation that is exactly equal to the energy difference between two states which can excite some of the parallel spins and change their status from low energy state to high energy state hence from parallel to anti-parallel direction. The frequency of this electromagnetic radiation is equal to the Larmor (resonant) frequency and if and only if that frequency can stimulate these switches. There are two commonly used pulses based on their lengths and effects, termed as 90° pulse and 180° pulse. 90° pulse produces maximum magnetization in the x-y plane and 180° pulse causes magnetization to be tilted in -z direction so no transverse magnetization occurs (Figure 2.4) (M_0= net magnetization, B0= external electromagnetic field, B_1= The magnetic field of the RF pulse).

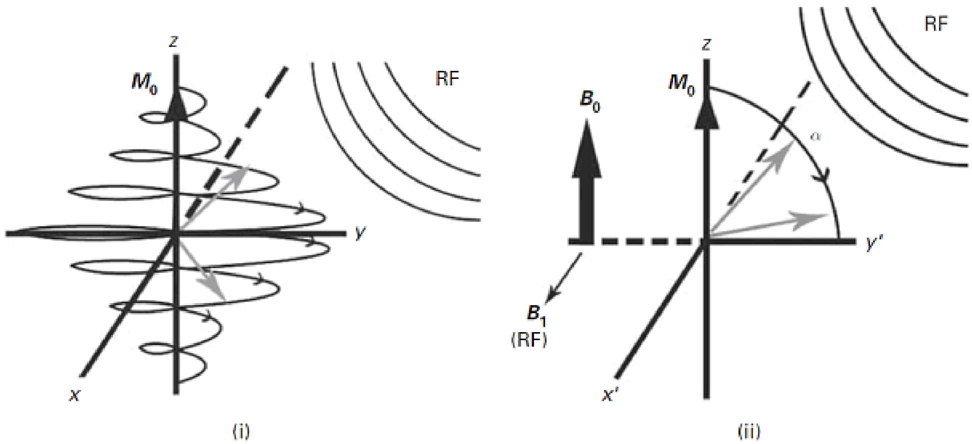

Figure 2.4 The effect of a radiofrequency pulse, at the Larmor frequency, on the net magnetization as seen from (i) the laboratory frame and (ii) the rotating frame of reference. Reproduced from [1].

By switching off RF 90° pulse, M_0 induces a current which is known as free induction decay (FID) signal.

T_1 or spin lattice relaxation causes the net magnetization vector to grow back to M_0 in the z direction (T_1 recovery). The process can be explained by a characteristic time of T_1 (Figure 2.5).

T_1 mostly depends on the tissue characteristic; for fat tissue, T_1 is about 100 ms and it's about 2000 ms for water. T_2 or spin-spin relaxation results from spins interacting with each other; individual spins rotate at different frequencies and get out of phase with each other so transverse magnetization decays to zero. The process can be described where T_2 is the characteristic time (Figure 2.5).

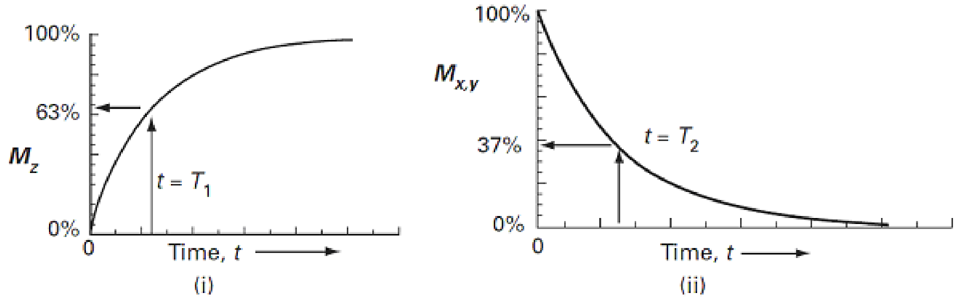

Figure 2.5 (i) Spin-lattice relaxation resulting in M_z increasing to M_0 with a time constant of T_1. (ii) Spin-spin relaxation resulting in $M_{x,y}$ dropping to zero with a time constant of T_2. Reproduced from [1].

In practice, due to local magnetic variations and non homogeneities within tissue also de-phasing, so that T_2 needs to, be replaced by T_2^* where $T_2^{'}$ characterizes local variations and can be up to 100 times shorter than T_2. This spin echo technique is used to record the true T_2 decay (Figure 2.6). To sum up, nuclear spins align with or against the magnetic field in the external magnetic field, B_0. Small number of excess spins aligned parallel with the magnetic filed since parallel and anti parallel spins terminate each other. Applying a RF signal at the Larmor frequency causes individual spins resonate, absorbing energy and precess. Protons flip towards the x-y plane produces transverse magnetization depending on the magnetic field of the applied pulse and its length. This transverse magnetization in the x-y plane, this induced signal then becomes the MR signal.

Protons de-phase when RF pulse is turned off. Longitudinal magnetization increases (T_1 recovery) while transverse magnetization decreases (T_2 decay). FID signal is composed of superimposition of all individual FID since all spins are not in identical chemical and magnetic environments. Processing FID signal with Fourier transform gives the information of NMR spectrum (Figure 2.6).

Each decaying sinusoid creates a Lorentzian line shape at the frequency of the sinusoid (its Larmor frequency). The positions of the peaks provide proton density map and this structure is related to the configurations of protons in the patient hence the relaxation times.

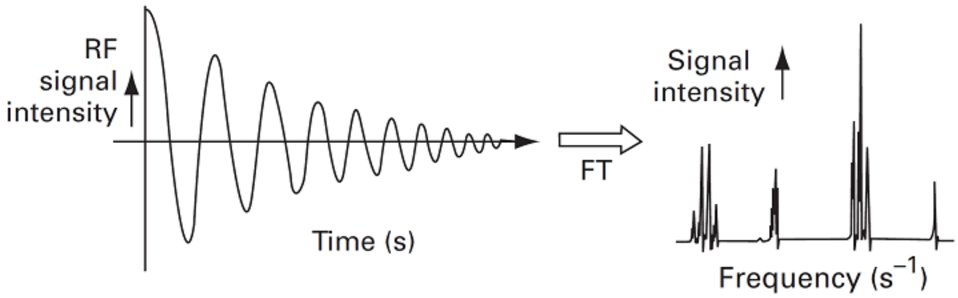

Figure 2.6 FID signal and its Fourier transform. Reproduced from [1].

2.1.1 Pulse Sequences

2.1.1.1 Spin Echo. Due to its flexibility and simplicity, spin echo pulse sequence is widely used in clinical imaging studies. Spin echo pulse sequence allows acquiring images with either T_1 or T_2 . This sequence starts with 90° RF pulse which moves the net magnetization into transverse plane, then 180° RF is applied as re-phasing pulse. After turning off 90° RF, phase dispersal causes T_2^* decay but if 180° RF is applied before dying of transverse component, spins start to de-phase again (Figure 2.7). This new echo, spin echo, occurs as the spins re-phase (Figure 2.8, Figure 2.9).

As phase encoding is in the y axis and its phase encoding gradient Gy is illustrated with multiple lines showing that amplitude changes each time the sequence is repeated. Frequency encoding gradient Gx is applied at the time of signal detection.

Lastly, the slice-selection gradient Gz is applied at the time of both RF pulses in order to excite those spins within the area of interest. Multiple acquisitions are necessary to improve the signal-to-noise ratio (SNR).

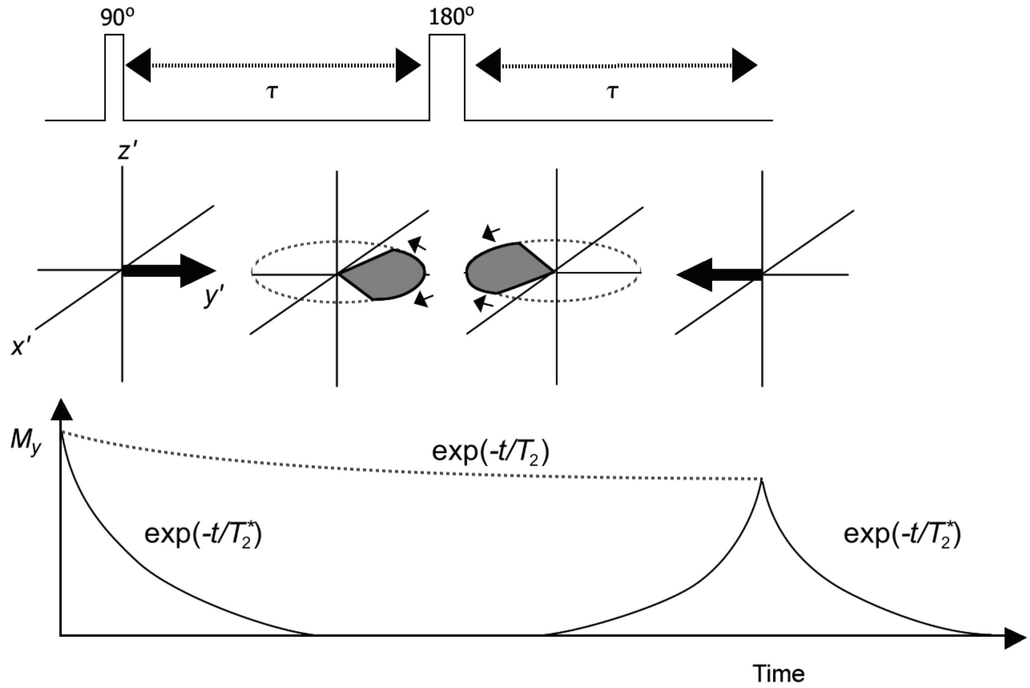

Figure 2.7 The 90° pulse tips the magnetization into the x-y plane; after it finishes, de-phasing begins; the 180° pulse flips the individual spins by 180° in the x-y plane, so that they start moving into phase again. Reproduced from [1].

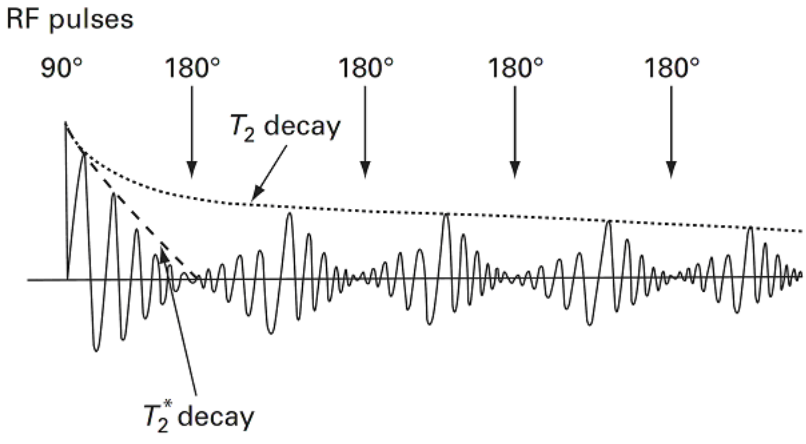

Figure 2.8 A spin echo pulse sequence, comprising a 90° pulse followed by carefully timed 180° pulses produces a series of spin echoes of decreasing amplitudes. Although each individual echo decays as T_2^*, the envelope decays as T_2. Reproduced from [1].

2.1.1.2 Gradient Echo. Gradient echo sequence is developed in order to overcome long imaging times needs of spin echo imaging (Figure 2.10). In this sequence,

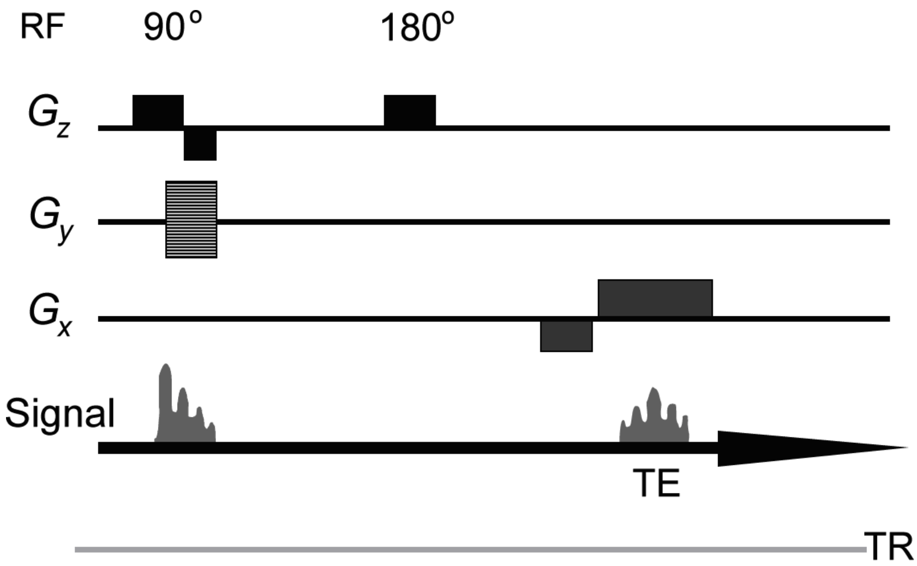

Figure 2.9 Pulse sequence for spin echo imaging. Reproduced from [1].

no 180° pulse is used which makes it faster but images are in influenced by T_2^* and its susceptibility to artifacts is higher.

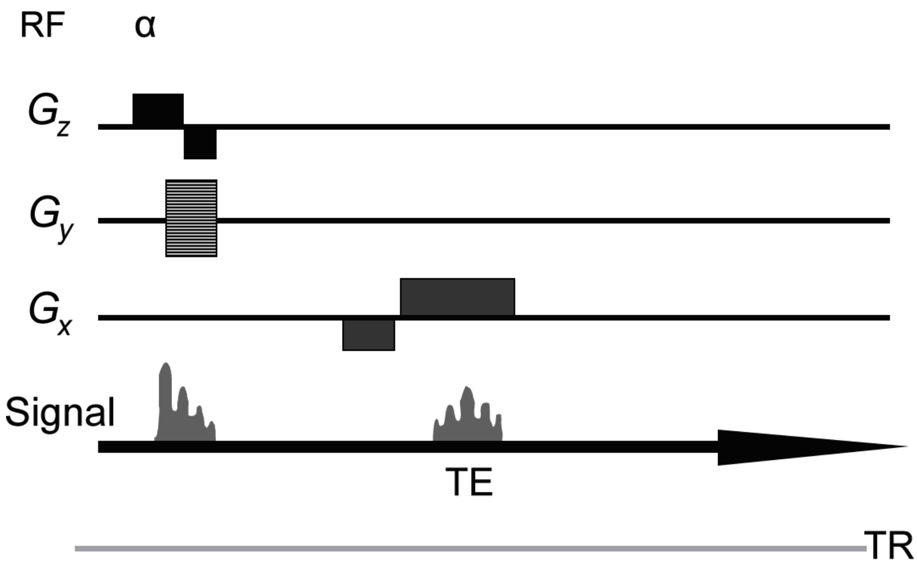

Figure 2.10 Pulse sequence for gradient echo imaging. Reproduced from [1].

While there are many varieties of pulse sequences like inversion recovery (IR) imaging to enhance T_1-weighted images, short time inversion recovery (STIR) sequence for nulling of signal from tissues having a specific T_1 relaxation time, fast low-angle shot imaging (FLASH) which uses a lower flip angle, many of them are evolved from spin echo or gradient echo imaging.

2.1.2 K Space

K-space is the raw data space that temporarily stores MR signals during scanning period as it is filled with one line per one repetiton. By setting frequency and phase encoding matrix, matrix size of the final image and the k-space are also set. While both k-space and real space have the same matrix size, the pixels do not correspond directly each other since reconstruction process uses Fourier transform. Data in the middle portion of k-space contains signal-to-noise contrast information and data around the outside contains image resololution data. Figure 2.11 shows this attribute of k-space. MR signals stored in k-space is transformed to MR image by applying two Fourier transforms (Figure 2.12)

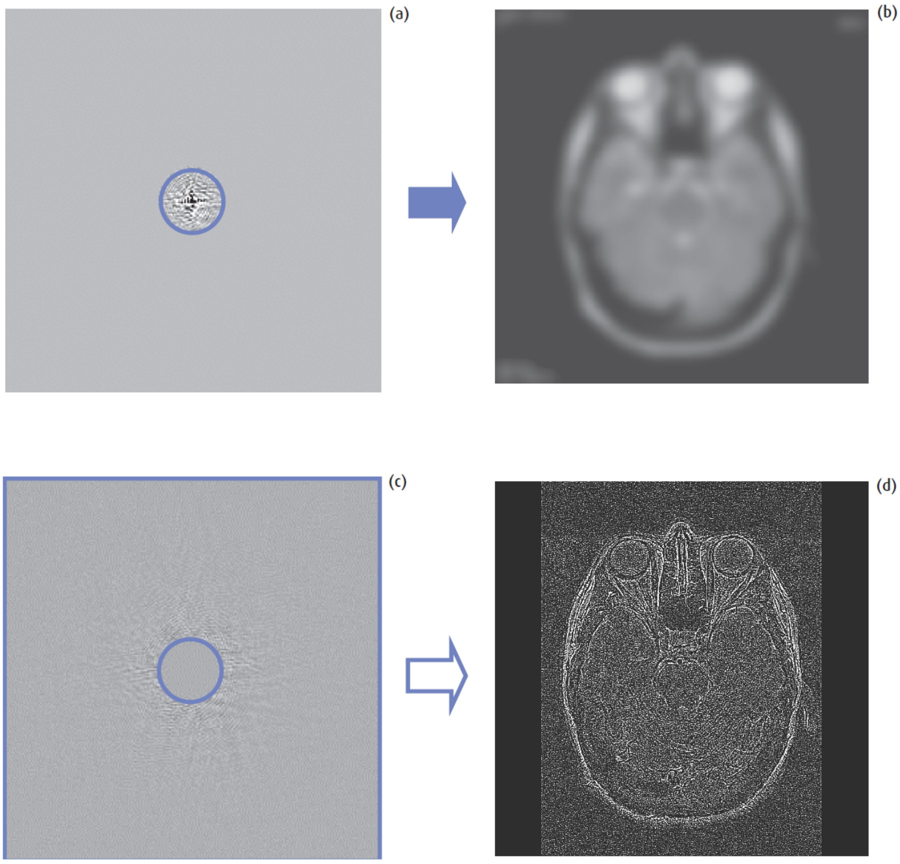

Figure 2.11 Signal and resolution information in k-space. (a) By reconstructing only the data from the middle of k-space we get all the signal and contrast information (b), but it is very blurred. (c) If we erase the middle of k-space and just reconstruct the outside data we can see where the tissue boundaries are (d), but the signal-to-noise ratio is very low and we have no contrast information. Clearly we need both parts of k-space to get a useful MR image. Reproduced from [2].

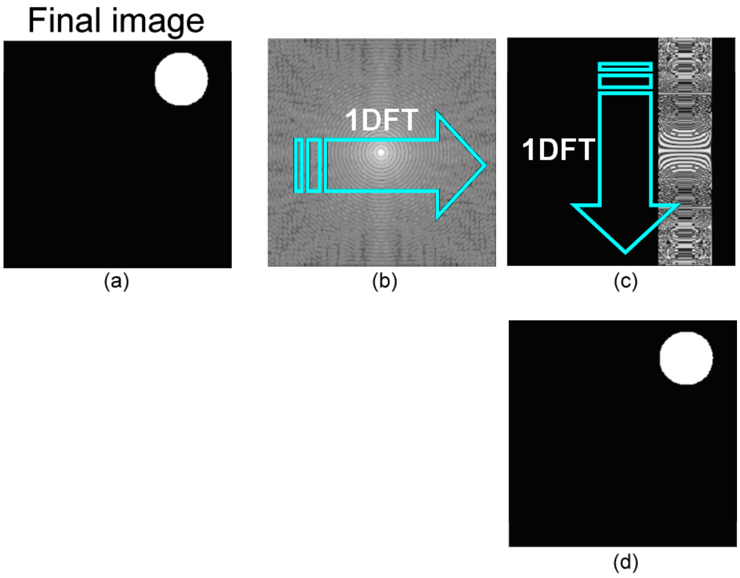

Figure 2.12 Original image (a). Applying two consequent discrete Fourier transform of K space equivalent of origial image produces initial image (b,c,d). Reproduced from [3]

2.2 Advanced MRI Applications

2.2.1 ASL

Arterial Spin Labeling (ASL) is a non-invasive method that allows assessment of perfusion. In ASL, arterial blood is labeled magnetically and after a period of transit time, traced water molecules flows into the different areas where it exchanges with local water. Inverted spins that exchanged local water alter local magnetization hence MR signal. Image taken during this period is called tag image and it's subtracted from control image to assess blood flow (Figure 2.13) [14, 2].

2.2.2 fMRI

Blood-oxygen-level dependent (BOLD) is a method which measures regional differences in oxygenated blood and majority of fMRI studies uses BOLD (Figure 2.16) [15, 2].

Neural activity is closely linked to blood flow and oxygenation of the brain.

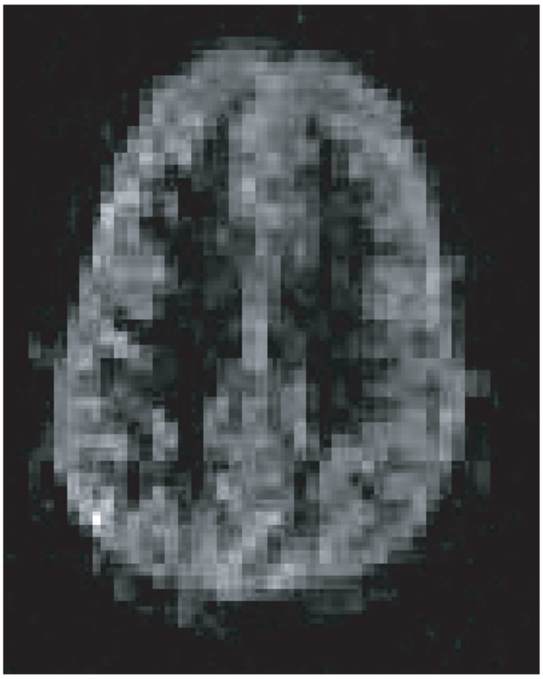

Figure 2.13 Arterial spin labelling perfusion map in a normal subject, showing higher perfusion as bright signal intensity. Reproduced from [2]

Active cells utilize oxygen compared to passive state and the blood flow to regions with active cells increases. Deoxygenated blood has a shorter T_2^*(it means lower MR signal) than oxygenated blood (Figure 2.15) [16, 2].

fMRI is used in many areas in medicine like cognitive psychology, neuroanatomy, electropysiology and surgery. It's use in surgery includes pre-surgical planning, post-surgical assessment and beyond this, it gives intra operative guidance, gives possibility of maximization of lesion resection [17, 18, 2].

2.2.2.1 Paradigm. A paradigm is an activity done and it's designed to increase neuronal activity in a specific area of the brain. Each activity is fallowed by a resting period and paradigm is repated to increase BOLD signal smilar for using avarages (Figure 2.14) [4].

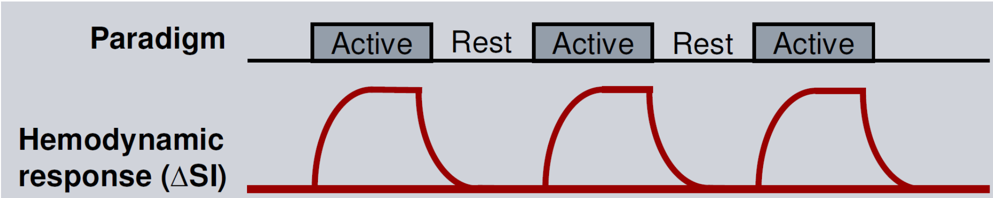

Figure 2.14 The Paradigm. Reproduced from [4]

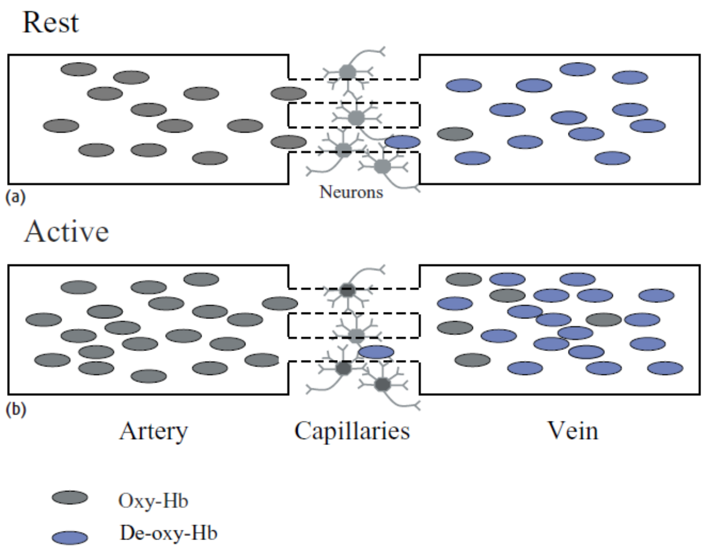

Figure 2.15 The origin of the BOLD effect. In activation (below) the over–provision of fully oxygenated blood leads to a reduction in de–oxy–Hb and an increase in local T_2^* in the draining veins compared with the rest condition (above). Reproduced from [2]

2.2.3 DTI

Diffusion refers to the random (Brownian) movement of molecules. Diffusion Tensor Imaging (DTI) is the evaluation of tensor straight from the diffusion-weighted data. A tensor is used to express diffusion in anisotropic systems. DTI allows the visualization of the position, direction and anisotropy of the white matter tracts. To assess the tensor, diffusion gradient in diffusion MRI is applied in at least six directions which describe the shape of the diffusion in three dimensions and direction of the fiber is calculated by tensor's main eigenvector (Figure 2.17) [19, 2].

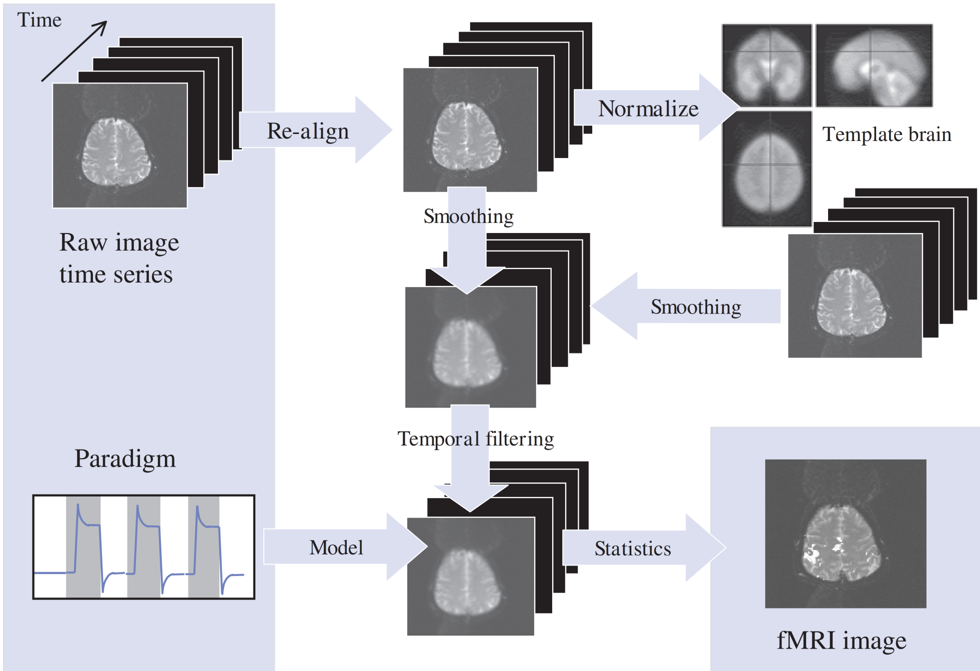

Figure 2.16 fMRI processing from the input of a time series of EPI data and a paradigm. The final image is a superposition of a statistical map on a raw image. Reproduced from [2]

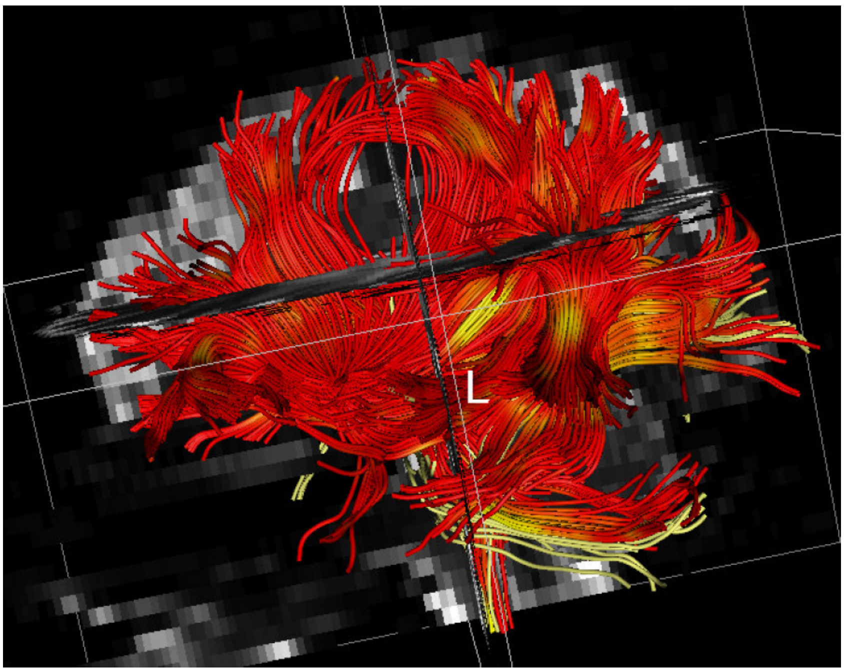

Figure 2.17 DTI example: Glioma, Courtesy of Dr. Dai, Tiantan Hospital, Beijing, China

3. PRINCIPLES OF TCP/IP NETWORKING

3.1 Historical Background of TCP/IP

It's ironic that many technologies that used today were firstly introduced for military purposes. This is also true for Internet which has evolved from ARPAnet, a network that created by Advanced Research Projects Agency (ARPA) and launched in 1969. ARPA is formed by U.S. Department of Defense with support by RAND, one of the America's leading think tank. One goal of the agency was creating a fault-tolerant network that enable U.S. military to stay connected in case of nuclear war.

As those days, there wasn't any common protocol for computers to connect each other, Network Control Protocol (NCP) was used as a protocol.

There were some requirements that ARPAnet should fulfill [5]:

- No one point more critical than any other
- Redundant routes to any destination
- On-the-fly rerouting of data
- Ability to connect different types of computers over different types of networks
- Not controlled by a single corporation

ARPAnet had four hosts in 1969 and was made up of computers at the University of California at Los Angeles, the University of California at Santa Barbara, the University of Utah, and Stanford Research Institute.

3.1.1 The Birth of TCP/IP

As ARPAnet grew, NCP became insufficient; it had many limitations for a network that started to be grow out of control. Cerf and Kahn introduced a paper: 'A Protocol for Packet Network Interconnection' which describes Transmission Control Protocol (TCP) in 1974. A new suite of protocols is developed based on TCP which is called as Transmission Control Protocol/Internet Protocol (TCP/IP) in 1978. NCP was replaced by TCP/IP in 1982 and ARPAnet switched over to TCP/IP on January, 1983 [5, 20].

3.1.2 Overview

Having its roots from military to academic research intends, TCP/IP is now a protocol that affects lives of many million people. Beyond the Internet, many organizations use TCP/IP protocol as a basis for intranets.

3.1.2.1 Applications. Today, there are many applications that uses TCP/IP system and many of them like electronic mail, file transfer and remote terminal operations were developed relatively earlier period [5].

- **Electronic Mail**: Text based electronic mail system was one of the first applications for TCP/IP protocol.
- **File Transfers**: During earlier era, there was a need for transfer of large quantities of data between research laboratories and universities, while "large quantities of data" might mean different on those days, File Transfer Protocol (FTP) was developed to fulfill this need.
- **Remote Terminal Access**: As there is a still need for remote access nowadays hence this thesis study was done to answer such a need. Remote access was very essential those days, e.g. for a researcher without sufficient computer

system/software it's possible to connect a robust system at a research site, done complex computing at that computer and get results which otherwise might take many times longer if it's possible. This capability was resulted in the development of Telnet application.

- **Web Surfing**: For majority of Internet users, web surfing is equal to Internet. While web pages were very static during their first introduction, web means social life for many people nowadays.

- **Some Other Applications**: Comparing to other applications, there are three relatively new and popular TCP/IP applications; audio and video players, Voice over IP (VoIP) networks, and virtual private networks.

- **Audio and Video Players**: During first introduction of internet connection to public, connection speeds were very limited for streaming audio or video data. As computer capacities and connection speeds have been improved, distribution of music and video files became very popular.

- **Voice Over IP**: Voice over IP (VoIP) is a extremely delay sensitive technology for transmission of digitized voice over TCP/IP. VoIP technology may require extra equipment and/or software for encoding/decoding and prioritizing fragmented voice data transfer.

- **Virtual Private Networking**: Virtual Private Network (VPN) systems utilize Internet for connecting distant computers on a virtual network platform. Leased lines or other means of direct connecting methods are relatively expensive and needs specialized hardware systems. However, setting up a Internet based VPN system needs very little, if any, investment. Downside of the using VPN over Internet is concerns regarding to security and privacy but the risk can be minimized with using security measures like firewalls, router access lists, authentication and encryption systems.

3.1.3 The ISO Reference Model

TCP/IP protocol suite consists of different layers for specific functions. While TCP/IP protocol suite predated the International Standards Organization (ISO) Open Systems Interconnection (OSI) Reference Model, it's worth to review ISO's Reference Model [21].

ISO launched an outline for standardization of communication system which is called as Open System Interconnection Reference Model. This model consists of seven layers which have unique functions. Each layer covers a lower year except the last layer providing isolation from higher layer functions. As OSI Reference Model consists of layered structure, it allows communicating through networks based on different vendor products and provides an open architecture for users and vendors.

Table 3.1
The ISO Open System Reference Model [5, 7, 8].

Layer no	Function
Layer7	Appication
Layer6	Presentation
Layer5	Session
Layer4	Transport
Layer3	Network
Layer2	Data Link
Layer1	Physical

- **Layer 1: The Physical Layer**: Layer 1 of the ISO Reference Model is physical layer that enables physical and electrical connection between different devices.
- **Layer 2: The Data Link Layer**: Next layer of the ISO Reference Model is data link layer and it controls accessing the medium in layer 1.
- **Layer 3: The Network Layer**: Third layer of the ISO Reference Model is network layer and this layer arranges connection between clients of source and destination.

- **Layer 4: The Transport Layer**: Transport layer manages the transfer of information once a route has been established by network layer protocol.
- **Layer 5: The Session Layer**: Session layer is responsible for creating and terminating data streams between clients at network.
- **Layer 6: The Presentation Layer**: Presentation layer prepares transmitted data into the appropriate format for receiver.
- **Layer 7: The Application Layer**: Seventh layer of the model allows application to gain access to all of the services provided by the OSI Reference Model.

3.1.4 The TCP/IP Protocol Suite

The Transmission Control Protocol/Internet Protocol (TCP/IP) suite contains two separate protocols: TCP and IP and are collectively used as TCP/IP. Figure 3.1 comparison of the structure of the TCP/IP protocol and OSI Reference Model. It is worth to note that physical and data link layers are not part of the TCP/IP protocol suite [22].

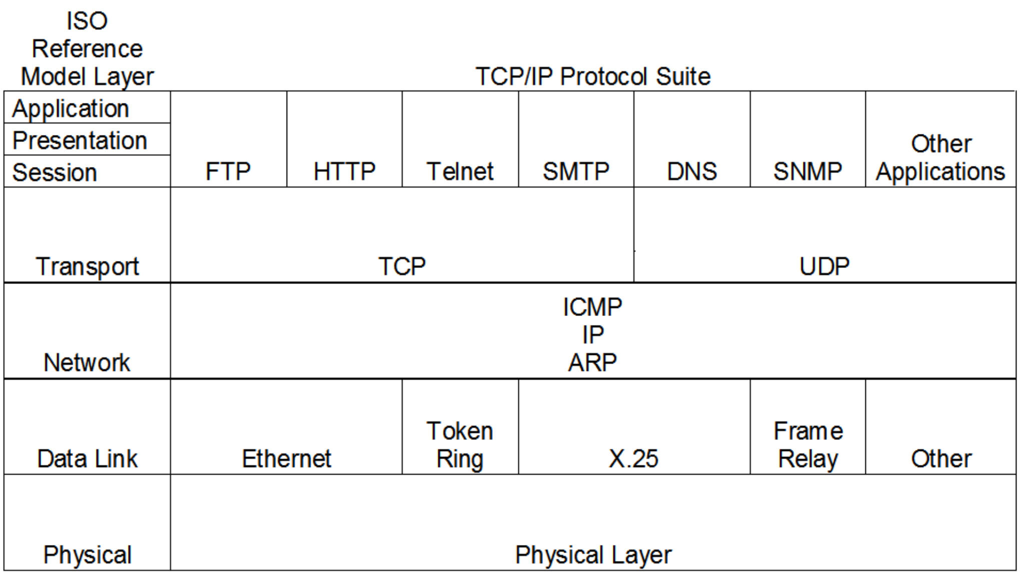

Figure 3.1 Comparing the TCP/IP Protocol Suite to the ISO Reference Model [5].

3.1.4.1 The Network Layer. Network layer in TCP/IP protocol consists of Internet Protocol (IP), Address Resolution Protocol (ARP) and the Internet Control Message Protocol (ICMP). Data that transferred between network layers is referred as datagram [23].

- **IP**: The Internet Protocol (IP) is essential since it allows addressing of datagrams between networks. IPv4 is the current IP version which is 32 bits.
- **ARP**: Address Resolution Protocol is responsible for address conversion; e.g. IPv4 uses 32 bit addressing while IPv6 uses 128 bit addressing.
- **ICMP**:The Internet Control Message Protocol (ICMP) is responsible for controlling messaging, checking response and requests.

3.1.4.2 The Transport Layer. TCP/IP protocol suite has two transport layer protocols: Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) [5, 23, 24, 25, 26, 27].

- **TCP**: TCP is error-free 'guaranteed' protocol. TCP protocol ensures the establishment of connection between source and destination sites before data transfer. Data is transmitted as segments and each data segment is checked for error and if any error is detected, receiver requests retransmission of the data which makes TCP as an error free protocol. Due to nature of the protocol, lagging of data transmission can occur and UDP protocol is developed to solve these issues.
- **UDP**: The User Datagram Protocol (UDP) is non-error checking and best effort transport protocol. In case of transmission of small data, using connection oriented protocol can cause delay in the process. UDP sends datagram without establishing a connection with receiver which makes it 'best effort' protocol. Checking receiver is done by higher protocol layer and in case of transmission problem; this layer decides resending the data or cancelling the transmission.

Applications such as FTP, e-mail, remote terminal transmission (Telnet) uses TCP as a transfer protocol due to needs of the data integrity. Applications that transmit relatively short packages like Domain Name Service (DNS), Simple Network Management Protocol (SNMP) etc uses UDP. Some applications like VoIP utilize both TCP and UDP protocols. Connection is established with TCP protocol and data transmission is carried by UDP protocol (Figure 3.2).

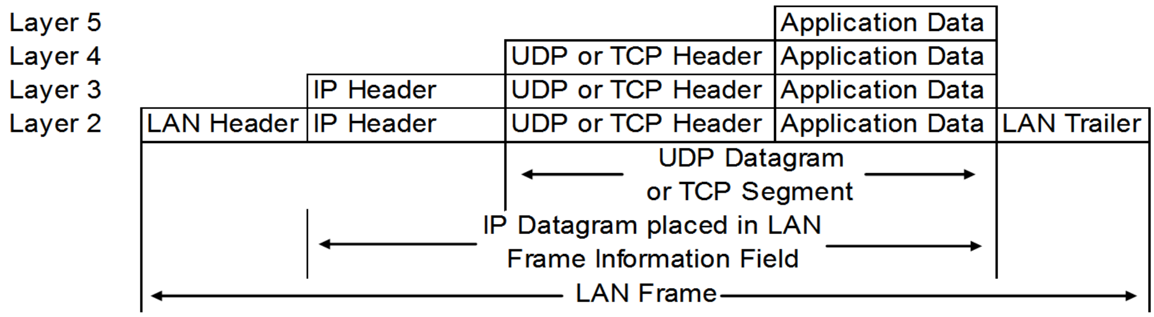

Figure 3.2 Data Flow within a TCP/IP Network for Delivery to a Station on a LAN. Reproduced from [5]

3.1.5 The Internet Protocol

Network layer function of the TCP/IP protocol is performed by the Internet Protocol (IP). The term 'internet' with lower case 'i' addresses connection of more than one TCP/IP based network [5].

- **Datagrams and Segments**: IP receives data with either UDP (UDP datagram) or TCP (TCP segment) header. IP header with UDP datagram or TCP segment forms IP datagram which contains destination IP for routing issues.

- **The IP Header**: Version 4 is the current version of the Internet Protocol, referred as IPv4 of which header consists of at least 20 bytes of data. IPv6 is the next generation version of Internet Protocol (Figure 3.3).

0	4	8	16		31
Vers	Hlen	Servive Type	Total Length		
Identification			Flags	Fragment Offset	
Time to Live Protocol			Header	Checksum	
Source IP Address					
Destination IP Adress					
Options + Padding					

Figure 3.3 The IPv4 Header. Reproduced from [5].

- **Vers Field**: The Vers field identifies the version of the IP protocol used. Vers field supports 16 version numbers with four bits.

- **Hlen Field**: The IP header may vary and Hlen field indicates the length of the field. The Hlen field is four bits in length And represents the number of 32 bit words in the header.

- **Service Type Field**: The Service Type field or Type of Service (ToS) field is an eight-bit field and consists of two sub-fields: Type of Service (ToS) and Precedence (Figure 3.4). The Type of Service sub-field indicates how a datagram should be handled. Precedence sub-field represents the transmission priority of datagram.

7	6	5	4	3	2	1	0
R	Type of Service				Precedence		

Figure 3.4 The Service Type Field. Reproduced from [5]

- **Total Length Field**: The Total Length field represents the total length of an IP datagram in bytes including options, followed by a header and the data following the header. The Total Length field is 16 bits in length which means that an IP datagam may have a maximum defined length of 65,535 bytes.

- **Identification and Fragment Offset Fields**: An IP datagram can range up to 65,535 bytes in length but some networks can support smaller transport frames that carries a datagram, fragmentation of datagram is necessary in such circumstances. Three fields of the IP header is used if datagram is fragmented: Identification, Flags, and Fragment Offset fields. The Identification is used to indicate which datagram belong to which one and this field is 16 bytes in length. Fragment Offset field is eight bytes in length and points out the position where each fragment belongs in the whole message.

- **Flag Field**: Flag field is four bytes in length and contains fragmentation information.

- **Time to Live Field**: The Time to Live (TTL) field is eight bits in length and is used to specify the maximum amount of time that a datagram can exist, the value placed into the TTL field is a router hop count.

- **Protocol Field**: The Protocol field is eight bits in length, allows up to 256 protocols to be defined under IPv4. Two most commonly used protocols are TCP and UDP. Exhibit 4.4 lists the current assignments of Internet protocol numbers.

- **Header Checksum Field**: The Header Checksum field contains a 16-bit cyclic redundancy check (CRC) character and provides a method for guaranteeing the integrity of the IP header.

- **Source and Destination Address Fields**: Both the Source and Destination Address fields are 32 bits in length under IPv4. As their name implies, The Source Address and the Destination Address represent the originator of the datagram and recipient respectively.

4. REMOTE CONTROL SYSTEMS

Remote technology has becoming more and more popular through last decades and many technologies emerged aiming to fulfill this need [28, 29, 30]. Internet Data Corporation expected that software sells in this field would be third fastest growing portion in the software market [31].

Remote controlling is a general term that literally means connecting a computer or a software from a distant site and remotely managing and controlling it [32, 33, 34, 35]. Generally, a remote control software is developed on Client/Server architecture on TCP/IP basis. Connection between two distant computer or system can be provided by different types of networking systems, e.g. dial up connection through a modem, leased line connections, local area networking etc. As described before, virtual private networking (VPN) through internet connection is one of the most commonly used method for connecting two distant computer 4.1.

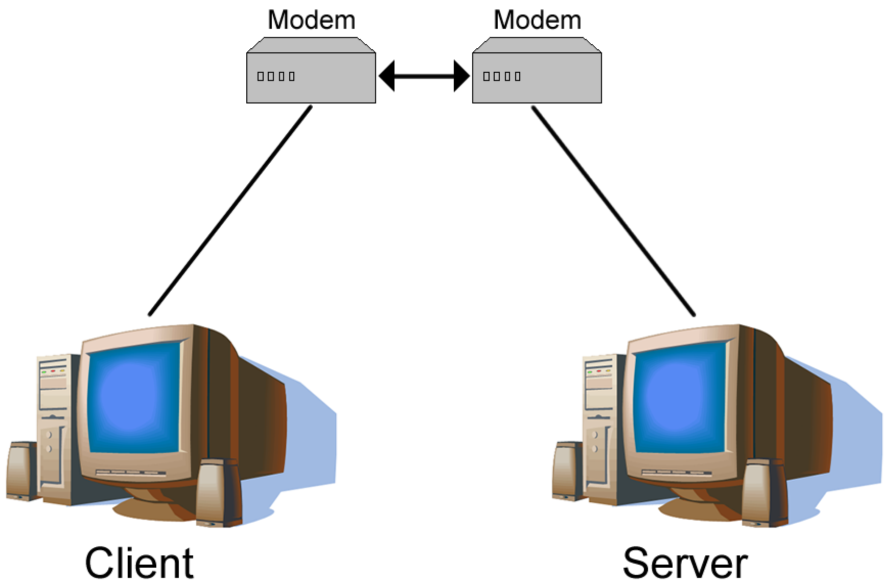

Figure 4.1 One-to-one network topology remote control.

While there may be differences between different development approaches, client first connects to server computer with a password set by server side previously. Port forwarding could be necessary if VPN or similar technology is not used. If connection between two systems are established, data transfer begins. Main component of transferred data is screen capture image of server computer. Client computer shows this image and user can manipulate server computer with client's input devices (e.g. keyboard, mouse). Control data of client computer is sent to server side and executed.

4.1 Screen Transfer

As described before, many systems have been developed in the past [35, 36, 37, 38, 39, 40] . There are two main techniques that utilizes different approaches to process display data. One technique is capturing low level commands (i.e. kernel functions of MS Windows for drawing a windows, rendering text strings etc.). X-system [40] and Microsoft Remote Desktop Protocol (RDP) [38] utilizes this method. Main advantage of this method is efficiency in representing the display of graphical interface unless there is a intense graphical data (e.g., video playback, gaming). One drawback of this method is it is highly dependent to interpretation of low level commands on client operating system. Moreover, it's hard to implement, if possible, this kind of systems to computers with different operating systems due to different rendering mechanisms of different operating systems. Second technique utilizes different compression algorithms to compress the screen data and sends it to client computer like VNC and THINC system. Several methods was proposed for efficient screen compression. One study utilizes lossless screen compression adopting predictive coding techniques [41] but this system is not convenient in low bandwidth environments. Another technique that has been proposed compression is effective for discrete tone images and useful in text rich images but it's not that successful in continuous-tone images (e.g. video playback) [42]. DjVu uses a different approach by separating images into image layer and mask layer and uses different compression algorithms but this technique is computationally complicated and not convenient to use in real time applications (Figure 4.2) [6].

Tight encoding data uses data analyzer for applying different kind of filters for different image data types. This method is useful for images that contain many same-color pixels but it is not as effective as for complex images. Another pixel based screen update algorithm uses optimal linear interpolation in which server just transmits a small sample of pixels and client updates new screen by piecewise linear interpolation but this method also requires updating every region as static image [43]. Fast content expression (FCE) constructs a table of same pixel values in a given region and converts each data into an index but different regions may be updated at different times and this can cause temporal redundancy [44].

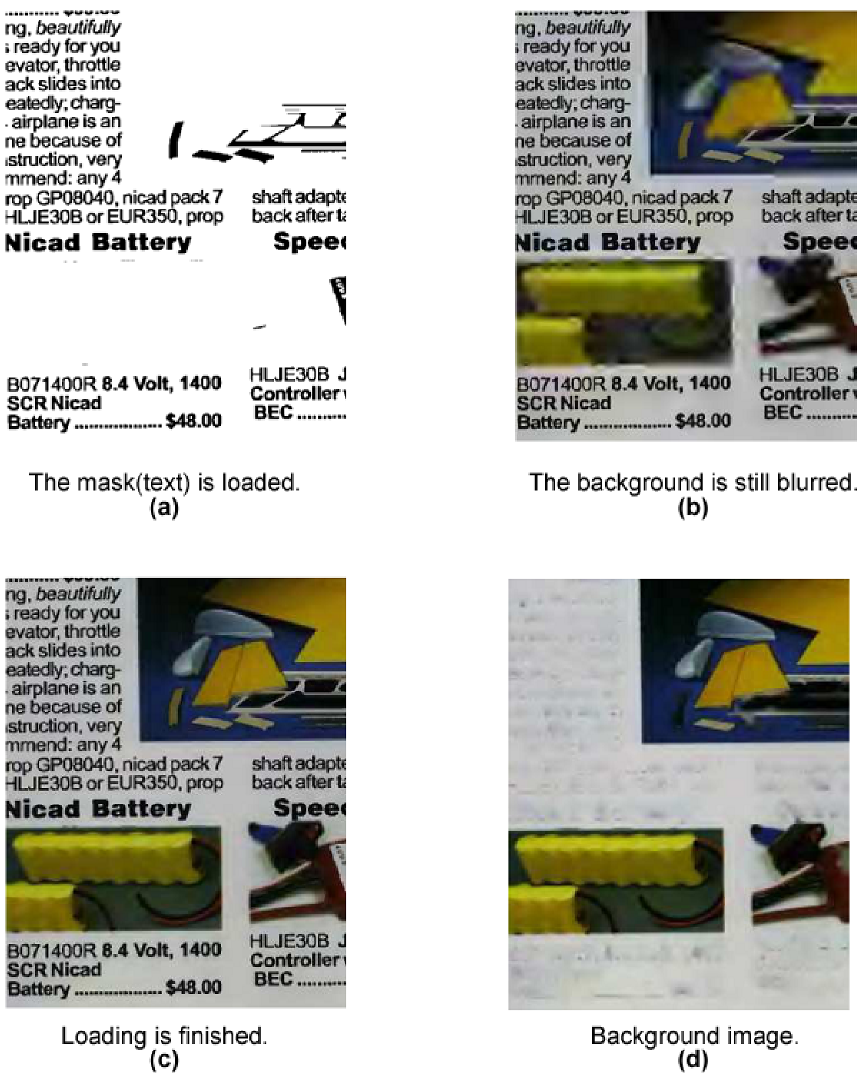

Figure 4.2 Downloading through a 56K modem: progressive decompression of text first (a), followed by the background at increasing quality (b, c, d). Reproduced from [6].

5. PROJECT

Aim of the thesis study is developing a remote control and technical research support software for advanced neuroimaging applications. With this system, users will be able to get visual feedback from MR console and perform operations on research workstation at clinical site remotely. This techical research support system will let researchers connect to several clinical application sites and allow them to observe and guide them remotely. This could improve productivity and optimise time and error management in multicenter coordinated neuroimaging research protocols.

5.1 The Scope of the Study

This thesis study involves developing a quality assessment, research and technical support system. Main modules and sub-modules are listed below:

1. **Development of Server Platform** Server program is developed to be run on a computer at distant clinical site. Server listens employed TCP/IP ports and provide necessary data to client and intervenes MS Windows handlers to let distant users remotely control the computer on which server program runs.

2. **Development of Client Platform** Client program can be executed on any computer with MS Windows XP or Vista with appropriate user privileges. Client requests necessary data from server and sends user data when remote control is in demand. Details of system frame is discussed in Appenix A.

In the next section, the theoretical and practical aspects will be explained in detail.

5.2 Server Platform

As mentioned, server program is developed to be used in distant computer at clinic/research site.

Before detailing the server platform, it's necessary to overview the whole system which is summarized in Figure 5.1 and Figure 5.2 .

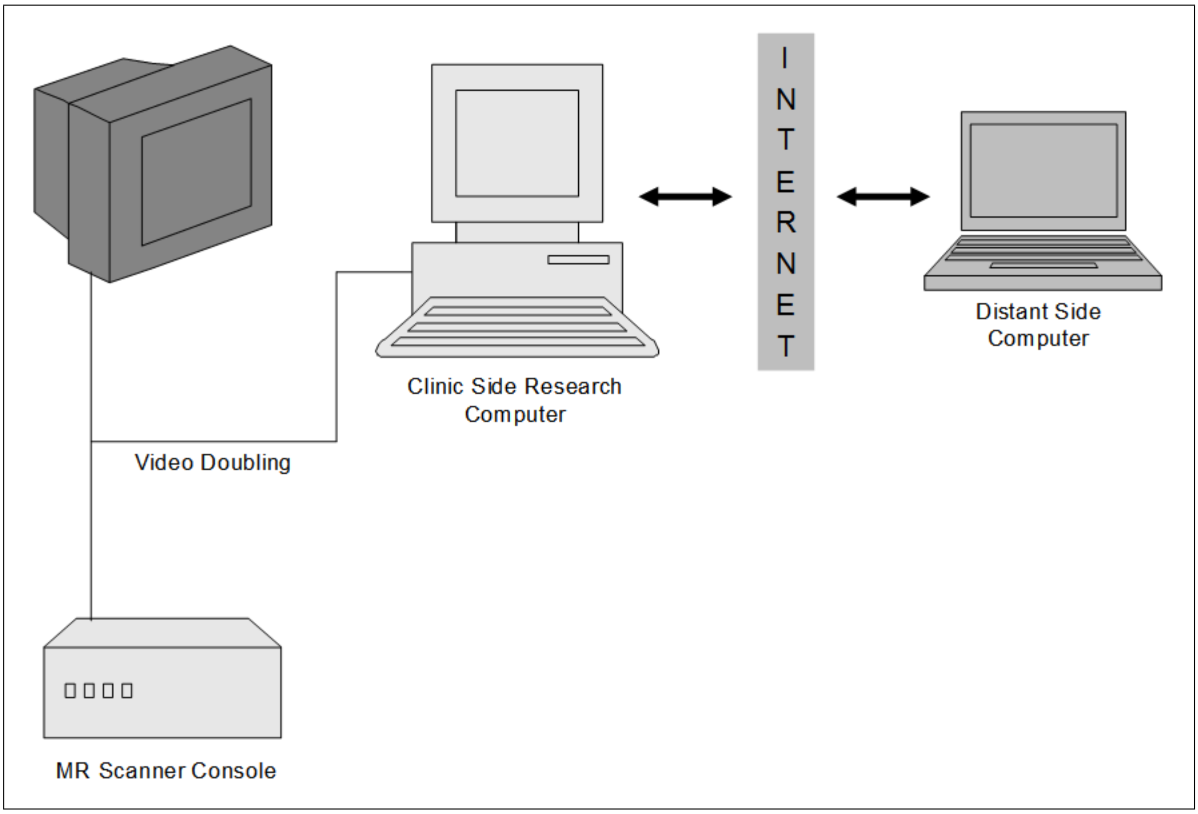

Figure 5.1 Diagram of the system

Physically, server side consists of two computers; MR scanner console and clinic side research computer, namely "server computer" on which server program runs.

Initially, two separate server programs were planned to be developed for MR scanner console and clinic side research computer but it was realized that it's not convenient to install a program which requires high privileges and open TCP/IP ports to a highly secure and strict MR scanner system. Proposed solution to this problem

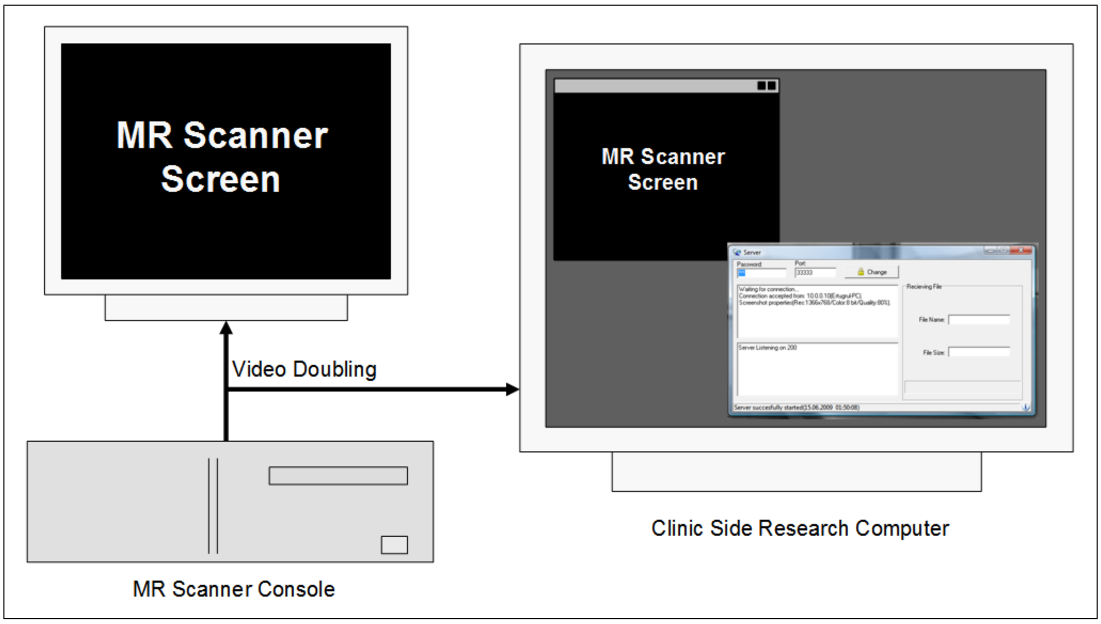

Figure 5.2 Server program captures self screenshots and MR scanner screen shots.

is doubling video output of MR scanner console computer and capturing this data on clinic side research computer with video capture card. Server program reads this data and sends it to client program as a stream.

Depending on the problem, two approaches are proposed as problem solving strategies (Figure 5.3). Initially, connection speed is a main factor for determining the strategy, if high speed internet is available then remote assistanve with full screen remote control is the best aproach. When limited connection speed is an issue, user can decide to grab raw data from remote computer, process locally and send it back or get user activity as an AVI file and inspect it.

5.2.1 Server Program

Server program is basically receives orders from client program and executes this orders. Four different TCP/IP ports are used by the program, on of which can be changed at both server and client side.

Program automatically loads settings from server.ini file when executed and

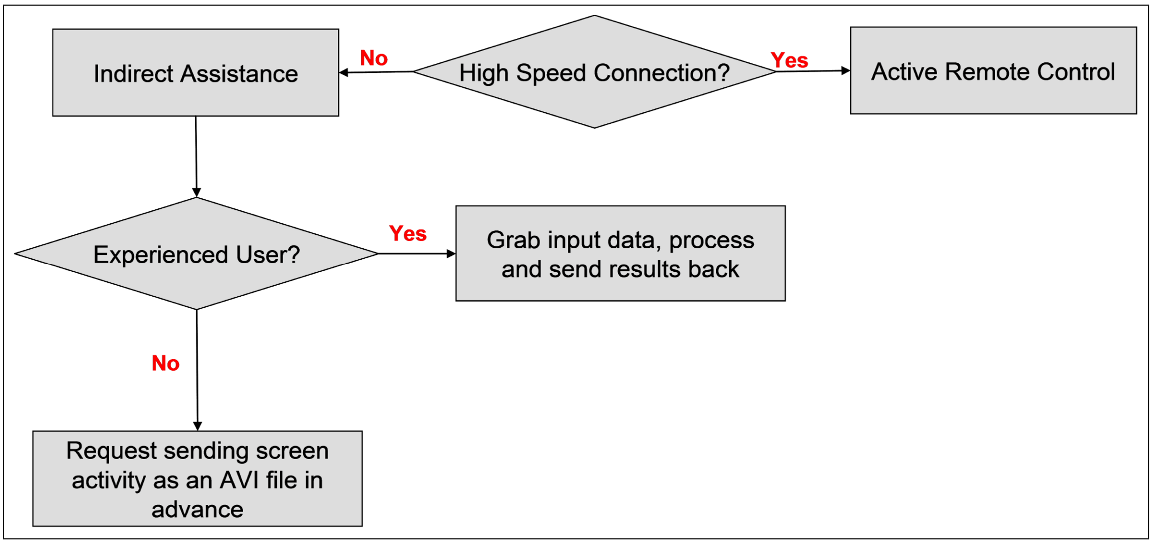

Figure 5.3 Diagram of problem solving strategies.

changes made by user on settings are also automatically saved. After execution, a series of event is also executed and server program starts to listening for the client program to properly log in to server and start communication (Figure 5.4, Figure 5.5).

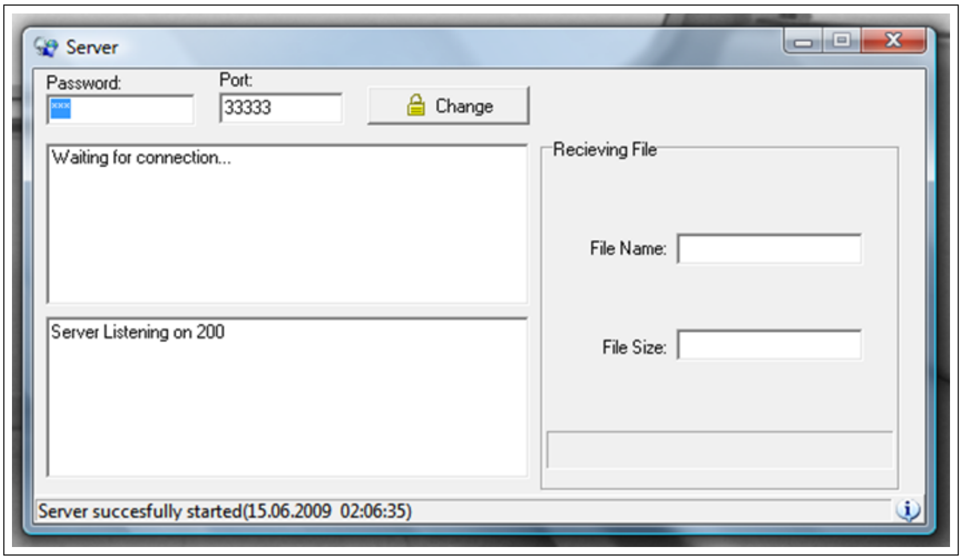

Figure 5.4 Server program screen.

After successfully establishment of communication between server and client programs, server program opens two TCP/IP ports, one for file receiving and one for remote controlling. Each time a data stream is received by the server program, server program processes the message to execute appropriate action. There can be several types of messages that client could send to server computer. Any message sent by client has a five bytes header and the remaining part of the stream contains data depends to the command coded by first five bytes.

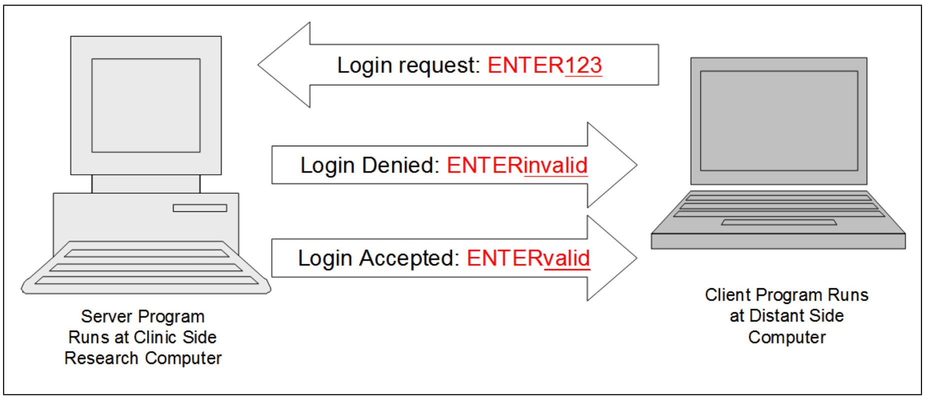

Figure 5.5 Authorization process of client.

5.2.2 Client Program

Client program runs remotely and responsible for requesting connection with server. Same number of ports (four different TCP/IP ports) is used by the program as server program uses. Client program currently has four modules (Figure 5.6):

- Remote control of clinic side research computer
- MR scanner screen viewer
- File transfer module
- Visual support system (chat system)

Another module that will allow users to save continuous screen images as a AVI file is also embedded into the server program. This module allows physicians/technicians to save screen activities when online support is not available and send it later for inspection.

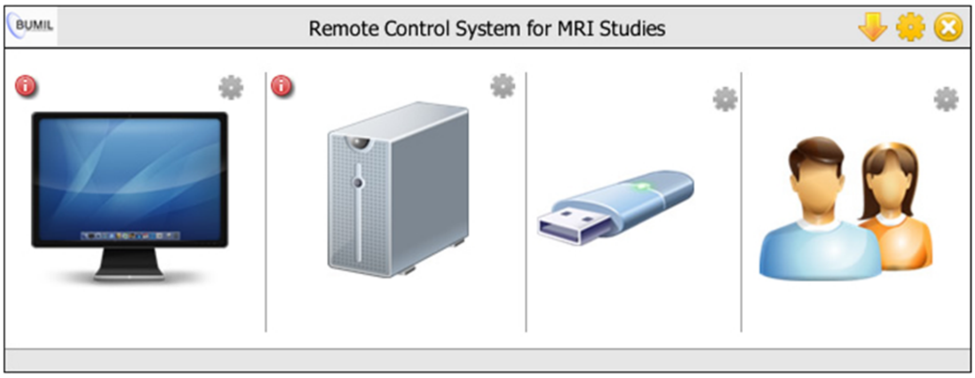

Figure 5.6 Main screen of the client program.

5.2.2.1 Remote control of clinic side research computer. First module is responsible for connecting and sending commands to the server program. This module has two panels, left panel contains a mini screen to show images sent by server and right panel contains controllers for connection and remote control (Figure 5.7).

Right panel contains controllers for IP, Port, Password edit boxes, Video Quality track bar, Image Quality radio buttons, Remote Control check box, Connect button and Fullscreen Remote Control buttons. When user clicks Connect button, client program opens a TCP/IP port which is specified with Pass edit box and sends connection request to the server program which is running on a computer with IP number specified with IP edit box.

5.2.3 MR Scanner Screen

Second module is responsible for acquiring screen data of MR Scanner Console captured by clinic side research computer via video capture card. As these images are captured passively from MR Scanner, remote controlling of MR Scanner is not available (Figure 5.8).

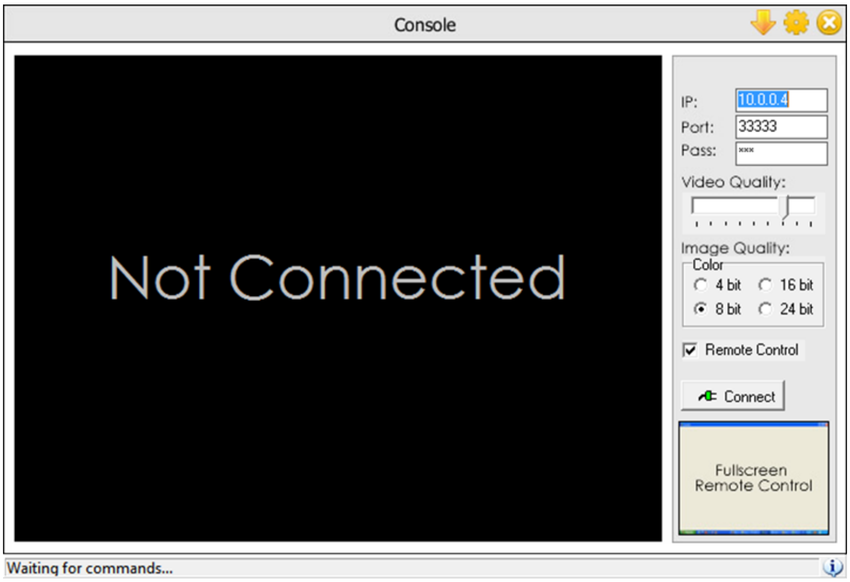

Figure 5.7 Remote control of clinic side research computer screen.

Figure 5.8 MR Scanner Screen.

5.2.4 File Transfer Module

Third module is responsible for file transfer to server side with FILE! command on TCP/IP port 5999 (Figure 5.9). When user pushes Send File button, a dialog box opens and asks user to select which file user wants to send (Figure 5.10).

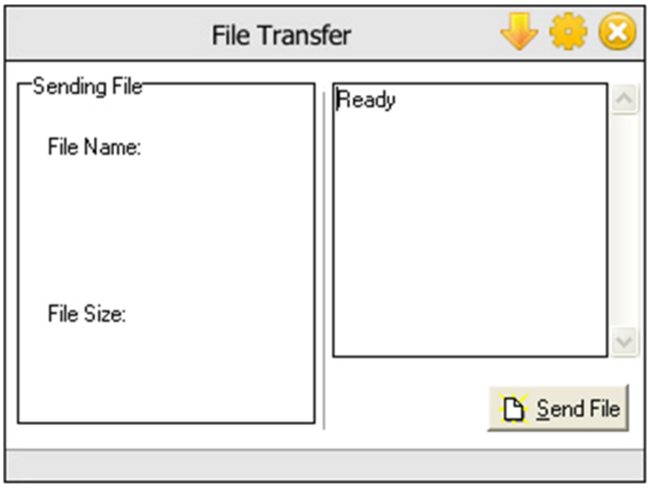

Figure 5.9 File transfer screen.

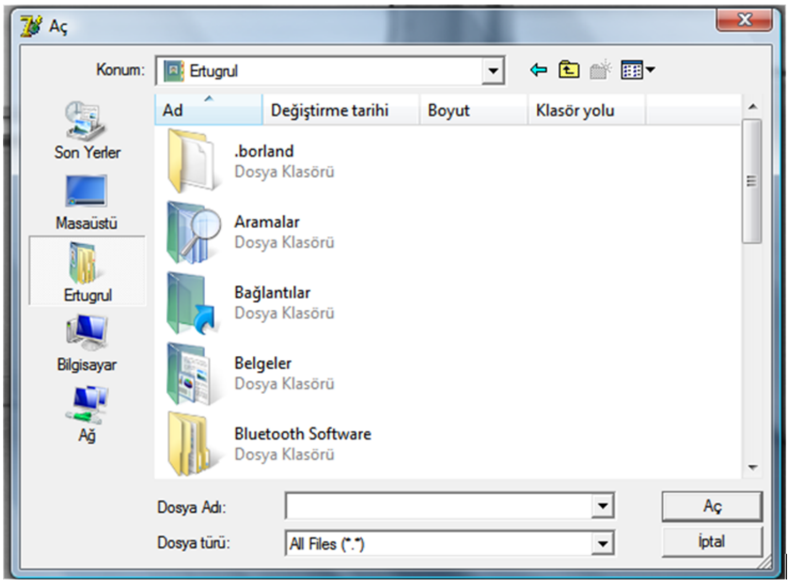

Figure 5.10 Open Dialog Box for selecting file to be sent.

5.2.5 Visual Support (Chat) Module

Visual support module lets users of client program and server program to communicate each other without need to another program or platform. Messages are sent between client and server programs with a data stream starting with MESG! command.

6. CASE STUDIES

Three case studies were conducted to assess the remote control system and evaluate it's use.

6.1 fMRI Case

6.1.1 MRI Specifications

Siemens 3T Trio (Erlangen, Germany) Sytem, Single-shot EPI sequence, sequence parameters: 36 slice, 3x3x3 mm pixel size, TR: 2660 ms, Echo time: 30 ms, 1 paradigm consists of 1 baseline and 1 active blocks. Each block consist on 10 TRs.

6.1.2 Background

As discussed before fMRI is used to assess neural activity via the haemodynamic response. It has wide use area and in the case study, it's used to evaluate language lateralization (e.g. localization of Brocka and Wernicke centers and their lateralization characteristics).

6.1.3 Case

This study was conducted with a healthy volunteer. Physician had tried to run a series of stimuli to assess Brocka and Wernicke centers with using STIDE with 20 stimuli. STIDE is a user friendly system developed in Boğaziçi University Biomedical Engineering Institute for stimulus designing in functional studies (e.g fmri) which is developed in Phyton language and GTK gui tools . STIDE works in coordination with in–house hardware in order to present stimulus simultaneously with MR machine.

Physician decided to show 10 initial stimuli (fixation period) then 10 stimuli composed of either meaningful or unmeaning words (active period) (Figure 6.1, 6.2, 6.3). In order to set up random stimulus pattern in active period, physician should add two stimuli (Group 1 and Group 2) with 50% probability per stimuli in active period to be shown during that cycle. Group 1 consists of meaningful words (e.g. object names) while group 2 consists of unmeaning words. So, physician should have added 10 entry for fixation period and 20 entry for active period but physician added 10 fixation period stimuli and 10 active period stimuli which equals to total 15 stimulus (Figure 6.4). As a result, fMRI study was not successfully completed and physician used the system that proposed in this thesis. After connection and communication via visual support system of the system, console screen and research side computer screen was inspected remotely. After seeing the premature end of the study at 16th stimuli (Figure 6.5), and inappropriate stimuli settings, correct stimulus sequence was entered remotely and study was repeated correctly (Figure 6.6).

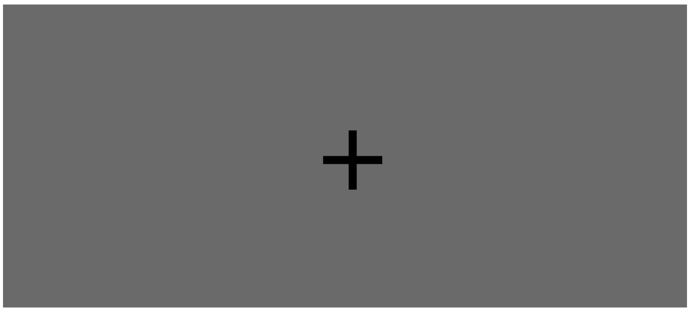

Figure 6.1 Fixation stimulus in fixation phase (Baseline stimulus).

Figure 6.2 Meaningful stimulus in active phase.

Figure 6.3 Unmeaning stimulus in active phase.

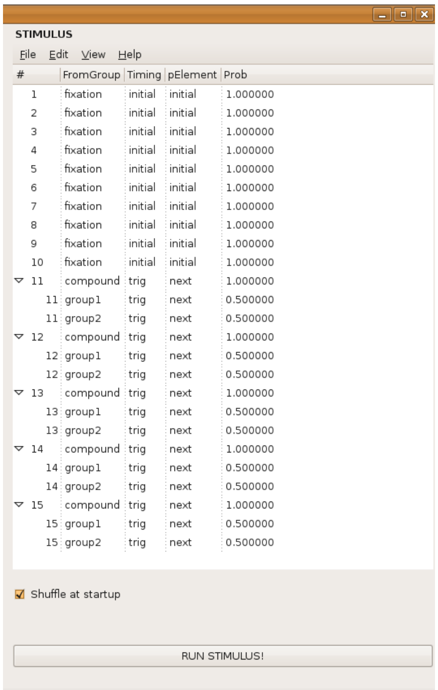

Figure 6.4 Incorrect stimuli design which contains 15 stimulus entry.

6.2 DTI Case

6.2.1 MRI Specifications

30 gradient directions, 2x2x2 mm pixel size, 2 B volumes (0, 1000 mm^3), Echo time: 109 ms.

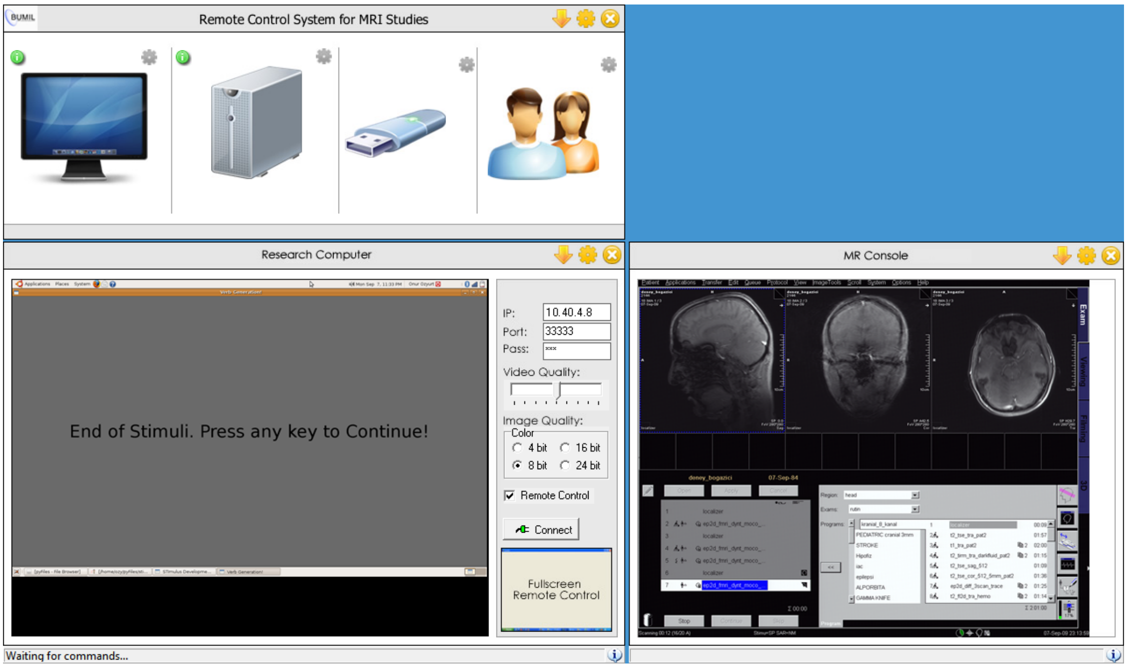

Figure 6.5 Premature end of study. Left screen (research side computer which runs stimulus program) shows end-screen and right screen (from MR console) shows MR image simultaneously.

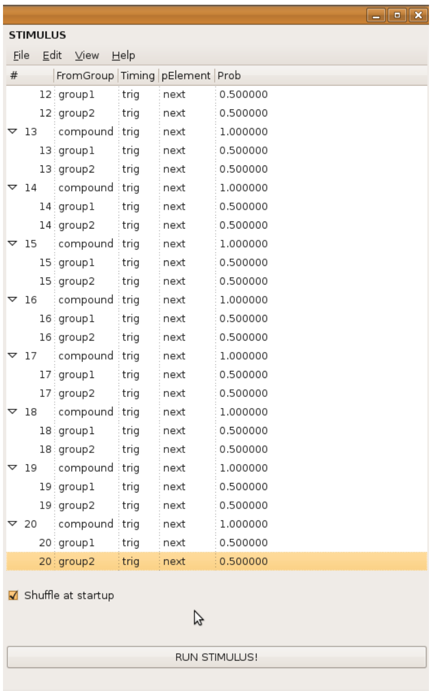

Figure 6.6 Correct stimuli design set via remote controlling.

6.2.2 Background

As discussed, DTI allows the visualization of the position, direction and anisotropy of the white matter tracts [19, 2]. In this experiment, using remote assistance for DTI analysis is evaluated.

6.2.3 Case

In this case, physician decided to make an DTI analysis of a healthy volunteer. Due to the big size of raw data (nearly 70 MB), remote accessing to research computer and processing data on that computer was preferred.

Raw data was composed of diffusion weighted images in various gradient directions. Processed data was composed of diffusion tensor parametric maps. Same MRI scanner was used for this case as above.

As server program is running on Windows XP operating system and data analysis tool (FSL) is running on Linux system, we proposed a soluton as using VMWARE to run virtual Linux on Windows system and made analysis on virtual Linux system. Figure 6.7 and 6.8 shows visualisation of processed data.

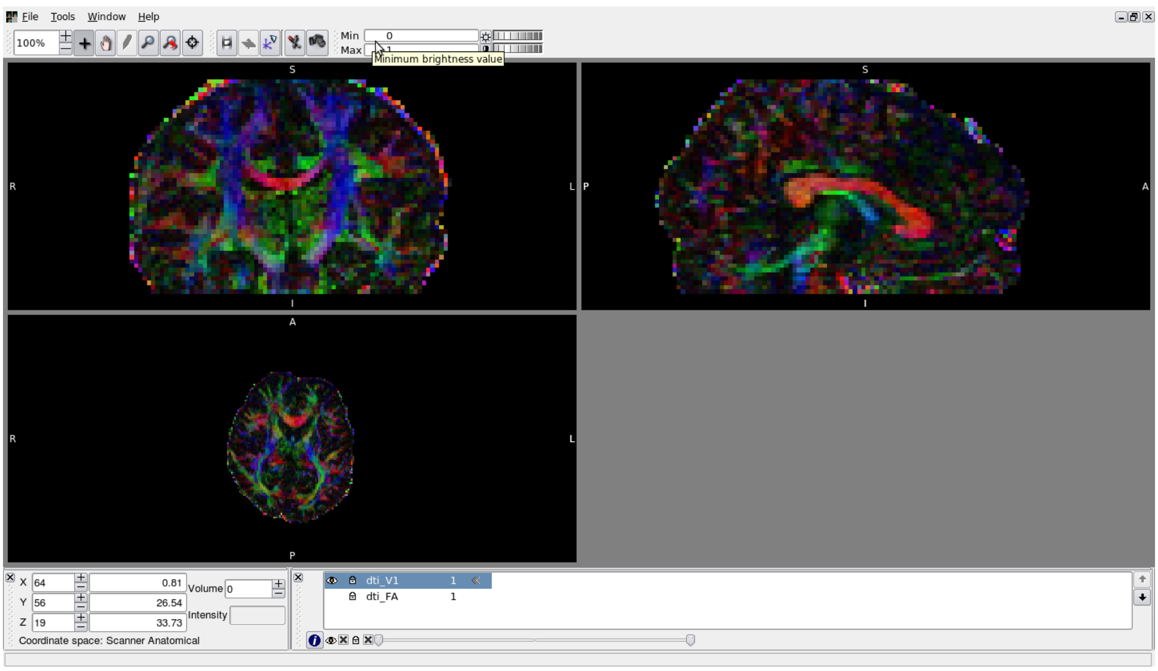

Figure 6.7 Visualisation of processed data represented as RGB.

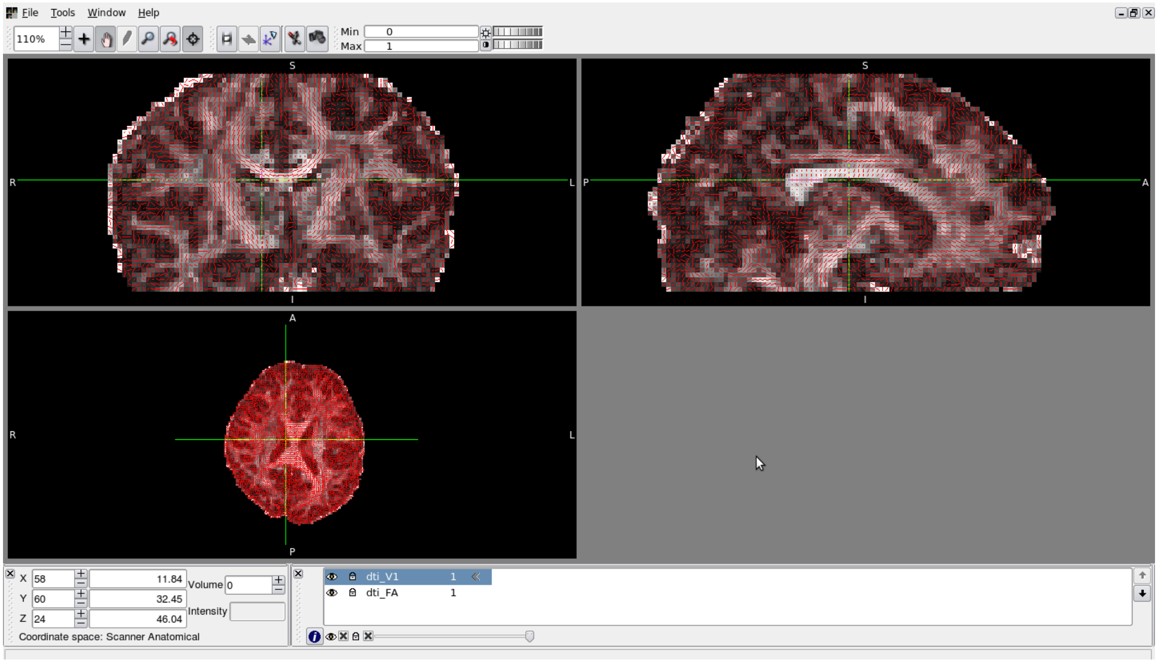

Figure 6.8 Visualisation of processed data, tensor vectors are represented as red lines.

6.3 ASL Case

6.3.1 MRI Specifications

30 repetitions, 4x4 mm plane resolution, 10 mm slice thickness, EPI readout sequence, QUIPSS sequence, TI_1: 600 ms, TI_2: 1600 ms, TR: 2600 ms, ET: 22 ms, Echo time: 109 ms.

6.3.2 Background

As discussed before, Arterial Spin Labeling (ASL) is a non-invasive method that allows assessment of perfusion[14, 2]. In this case, physician decided to make an ASL analysis, due to relatively small fize size of the raw data (3 MB), proposed solution is getting local raw data to remote computer for processing and sending processed data back.

6.3.3 Case

In this case, physician decided to make an ASL analysis of a healthy volunteer. Same MRI scanner was used for this case as above.

During our test, it took 3 minutes to send 3 MB raw data from local computer to remote computer (which is connected to Internet with 4 MBit shared wireless connection) which is acceptable. After getting raw data, a series of analysis were made with Mathlab and processed data was sent back to local computer.

Raw data was composed of magnetization difference maps (Figure 6.9) in single or multiple timepoints and processed data was composed of quantified perfusion values in units of ml/100 g.min (Figure 6.10). Processing parts were composed of brain extraction, motion correction, perfusion parameter estimation using a model which is usually General Kinetic Model [45].

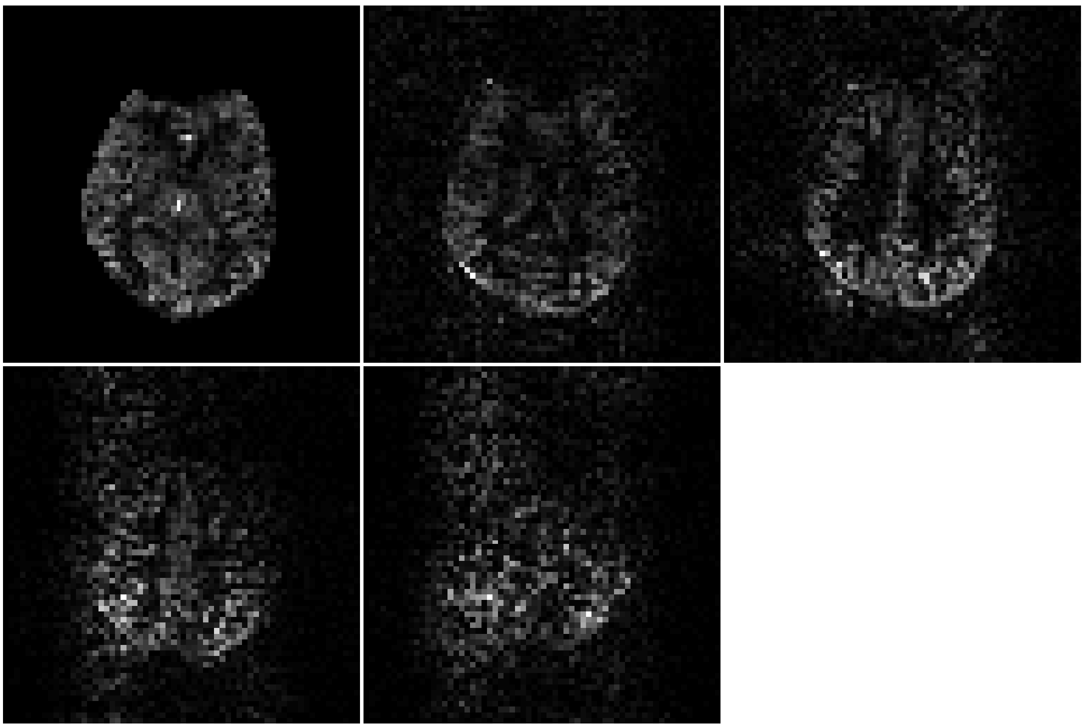

Figure 6.9 Magnetization difference maps.

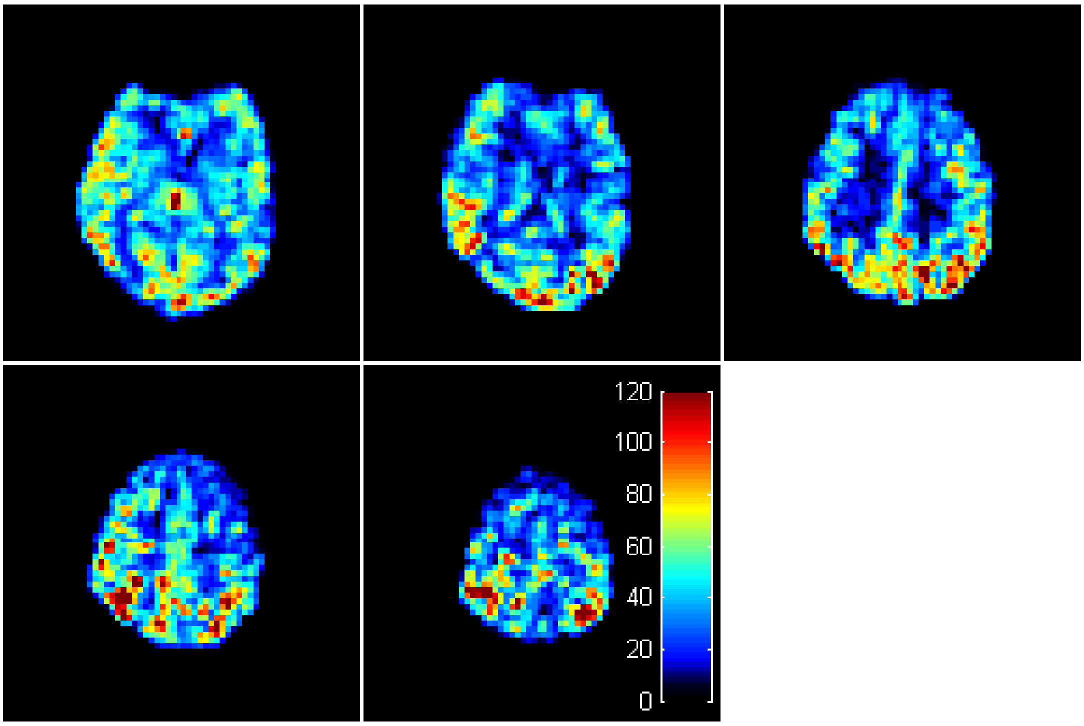

Figure 6.10 Processed data.

6.4 Discussion and Future Work

This thesis project is the first study as a remote control and quality assesment system for advanced MRI studies developed in Turkey. There are some limitations regarding to this system which can be improved.

During tests and case studies, some potential improvements and future work is observed.

Screen data is sent by server program as a single JPEG image which is very bandwidth friendly (Figure 6.11). On the other hand, dividing screen with e.g. 16x16 blocks and checking for differences between blocks each time before sending screen data and sending only changed parts could decrease the size of the data to be sent to the client computer.

During our case study, we realized that real time grab of a limited area might

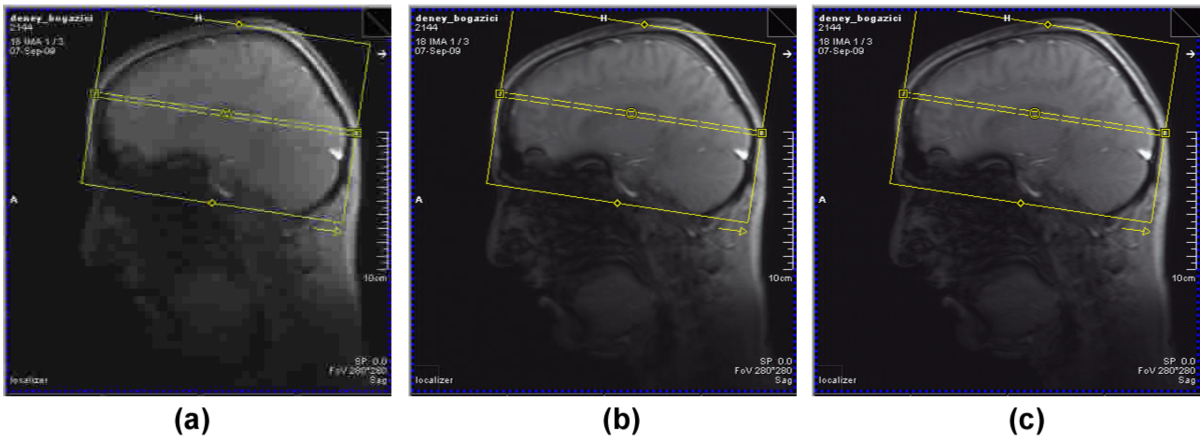

Figure 6.11 3 different version of a 416x419 image. (a) JPEG quality level = 25%, file size= 7 KB, (b) JPEG quality level = 75%, file size= 27 KB, (c) 24 bit uncompressed bitmpap, file size= 511 KB

be enough rather than full screen captures, so using area of interest for screen transfer could also improve tracking.

Another improvement that could increase the productivity is using two monitors at the remote side; one for console and one for research side computer. Using two monitors will allow users to get two full screen images from server side with appropriate software modifications.

During our studies, we generally acquired one full screen images per one to two seconds depending on the upload speed of the server side and download speed of client side. ADSL is generally used for Internet access but this could result in the decreased upload speed of server computer since upload and download speeds of ADSL varies. In order to improve data transfer, other connection approaches could be more convenient to use (e.g. cable net connection, 3G connection etc.).

APPENDIX A. PROGRAM STRUCTURE AND CODES

A.1 SERVER

A.1.1 ENTER

ENTER command is used for authentication of client. If client sends a data stream beginning with 'ENTER', server understands that a client is trying to log in. Server program reads the remaining part of the stream and compares it with the password, if it's not identical then server sends invalidation message to client and connection is closed in advance, otherwise server sends successful validation message to the client (Figure 5.5, Figure A.1).

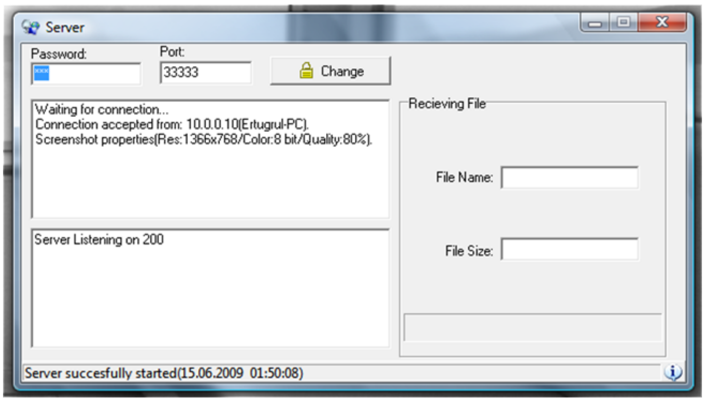

Figure A.1 Authorized connection.

A.1.2 BITM

BITM! command is sent by client as a screenshot request. When server program receives BITM! command, it takes a screenshot image as a bitmap, processes this raw data with requested compression ration and color depth and converts it to JPEG image and sends it to client with a message starting with 'BITM' and remaining part of the message is JPEG data which was converted to memory stream before transfer. Server program sends screen capture images of MR scanner computer and clinic side research computer alternately (Figure 5.2). Streams start with 'BITM!' contains screen data of clinic side research computer.

A.1.3 BITX!

BITX! is identical to BITM! with a one difference, server sends image data of MR Scanner Console with BITX! command.

A.1.4 RESOL

RESOL command is sent by client for defining the properties of screenshot that is sent by server program on request. Remaining part of a message starting with 'RESOL' contains color depth and image quality as a percentage. After receiving and processing RESOL command, server sends a message to client that contains screen resolution, color depth and image quality data starting with DEMOD, which will be reviewed at client specification part.

A.1.5 CONON

CONON command is sent by client to prepare server program for remote controlling. As server gets CONON command, it starts to listen TCP/IP port 111 for remote control data. Message syntax consists of event codes and $$$ sign between codes (Figure A.2).

Message codes for keyboard events starts with KEY code, next code state whether key is pressed (KDW) or released (KUP) and the last code is the integer value of key code.

Message codes for mouse events start with MOU code, next code state whether

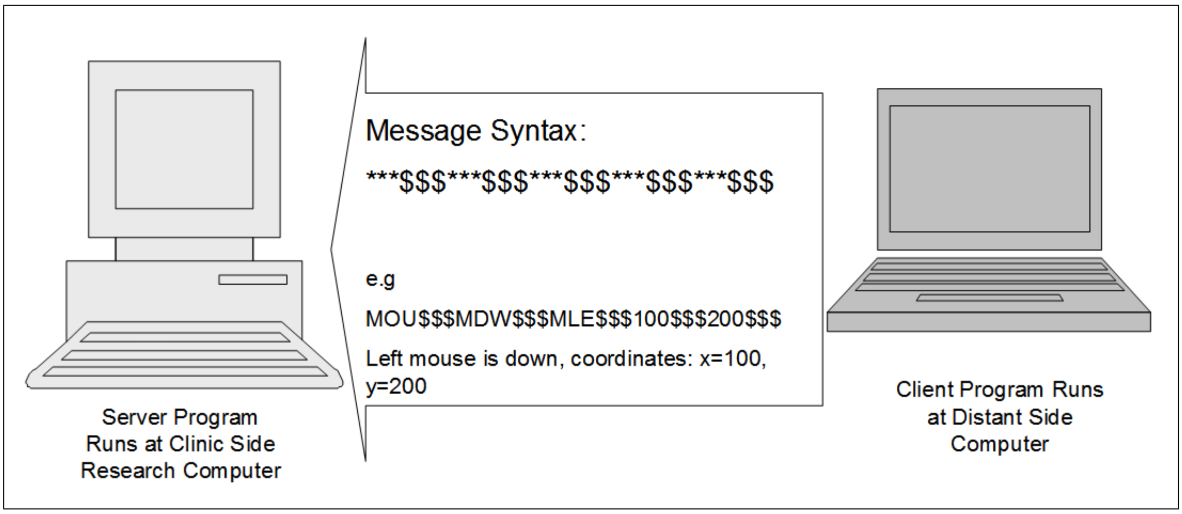

Figure A.2 Message syntax with an example.

mouse button is pressed (MDW) or released (MUP) or mouse is moved (MMO), if not moved, next one specifies which button is involved (MLE, MRI, MMI for left, right or middle buttons respectively). Next two codes are represents X and Y values of mouse pointer.

Depending on the command type, server runs Windows events to simulate client side events. Keyboard and mouse event messages are simulated with keybd_event and mouse_event procedures respectively which are native windows procedures that are called as:

- **keybd_event(bVk: Byte; bScan: Byte; dwFlags, dwExtraInfo: DWORD); stdcall;**
- **mouse_event(dwFlags, dx, dy, dwData, dwExtraInfo: DWORD); stdcall;**

If user presses left mouse button, server will run a Delphi code as:

- **Mouse_Event(mouseeventf_absolute or mouseeventf_leftdown, X_value,**

Y_value, 0, 0);

After processing messages, server program automatically releases shift, control and alt buttons in order not to create conflict with Delphi code below, respectively:

- **Keybd_event(16,0,KEYEVENTF_KEYUP,0);**
- **Keybd_event(17,0,KEYEVENTF_KEYUP,0);**
- **Keybd_event(18,0,KEYEVENTF_KEYUP,0);**

A.1.6 CONOF

CONOF command is sent by client to deactivate remote controlling.

A.1.7 FILE

FILE! command is sent by client to prepare server for file transferring. The remaining part of data stream that begins with FILE! command contains file name. Server program starts to listen TCP/IP port 5999 after FILE! command as file is sent by the client as a stream to port 5999. After downloading the file, server program sends 'DONE!' command to the client to inform that file was successfully transferred. After transmission, server asks if user wants to open received file with native Windows shell api:

- **shellExecute(frmMain.Handle, 'open', pchar(Filepath),nil, nil, sw_normal);**

A.1.8 MESG

MESG! command is used for bidirectionally messaging between client and server programs. Data streams starting with MESG! code contains raw text data.

A.1.9 SIZE

SIZE! command is sent by client to inform the server about size of the file which is prepared to be downloaded soon.

Server program has a visual communication window which is activated on connection with client. This module uses another TCP/IP port and unlike other services, it directly sends messages as data stream without five bytes of header code and get messages directly.

A.2 CLIENT

Many of the GUI elements are developed without using standard Windows controllers. A timer component catches three main mouse events:

- **onmousedown** Timer gets x and y values of mouse and stores that values. If mouse button is pusged while mouse cursor is on the upper part (header) of the program, then system acts as user try to move the program window. If user clicks mouse button while mouse cursor is over the module buttons (actually images), then corresponding module is activated.
- **onmousemove** If user initiated positional change of program window, new position is set by calculating differences between current x, y values and previous x, y values respectively. If mouse cursor is over the module buttons (actually images), then corresponding image is animated.

- **onmouseup** This event is called when user releases mouse button.

A.2.1 Client/Server Commands

- **ENTER** ENTER command is used for authentication of client. Assuming that user tries to connect with a password '123', client program sends message 'ENTER123'. First five bytes of stream is command and the rest of the stream is data. If server authorizes the client request and server and client programs are not running on the same computer then connection establishment occurs.

  After connection, server requires/sends some information with five byte codes from client depending on the current activities.

- **BITM** Client program prepares mini screen on the left side of the module when it receives a stream starting with BITM! code. If client user is not in full screen mode, incoming image file is resized to fit mini screen. In order not to produce image artifacts, a smooth resize algorithm is used which averages original pixel values presented in resized image (Figure A.3).

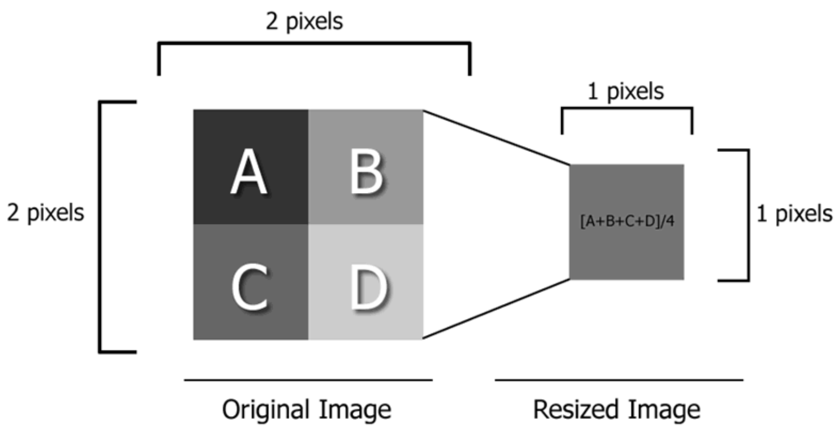

Figure A.3 2x2 bitmap image is converted to 1x1 bitmap image with averaging four pixel values. As palette of this figure is grayscale, it's relatively easy to average color values of pixels since Windows store color values in $RRGGBB format, in grayscale images, red, green and blue values are identical so averaging only one component of RGB palette is enough. Averaging pixels of 24 bit images are a bit complicated.

- **BITX** Client program prepares mini screen of second module when it receives a stream starting with BITX! code and shows image on that mini screen.
- **RESOL** RESOL command defines the properties of screenshot that is sent by server program on request. Remaining part of a message starting with 'RESOL' contains color depth and image quality as a percentage. After client sends RESOL command, server replies with a DEMOD message that contains screen resolution, color depth and image quality data. Video quality value is calculated by multiplying Video Quality track bar values with 10. Image quality is defined by Image Quality radio button and can be 4, 8, 16, 24 bits.
- **DEMOD** Stream starting with DEMOD command contains resolution, color and quality data of server computer.
- **CONON** CONON command is sent by client program to server when Remote Control check box is checked and user starts full screen remote control. After CONON message, client starts to send remote connection data with TCP/IP port 111. Message syntax consists of event codes and $$$ sign between codes. During full screen mode, user can interrupt and close full screen window with shift + f12 buttons.

  Connection codes:

  - KEY: Keyboard event is occured
    - KDW: Key down (key is pressed)
    - KUP: Key up (key is released)
  - MOU: Mouse event is occured
    - MDW: Mouse down (mouse button is pressed)
    - MUP: Mouse up (mouse button is released).MDW and MUP is followed by:
      - MLE : Left mouse button
      - MRI : Right mouse button
      - MMI : Middle mouse button
      - MMO : Followed by X and Y values of mouse

- **CONOF** CONOF command is sent to server to deactivate remote controlling.
- **FILE** FILE! command is sent to prepare server for file transferring and remaining part contains file name. After sending FILE! command, client sends file as a stream on TCP/IP port 5999.
- **SIZE** Messages starting with SIZE! command contains name of the file to be sent with FILE! command.
- **MESG** MESG! command acts same on client side as it does on server side. MESG! command is used for bidirectionally messaging between client and server programs. Data streams starting with MESG! code contains raw text data.

REFERENCES

1. Dougherty, G., *Digital Image Processing for Medical Applications*, Cambridge University Press, 2009.

2. McRobbie, D. W., E. A. Moore, M. J. Graves, and M. R. Prince, *MRI From Picture to Proton*, Cambridge University Press, 2nd ed., 2007.

3. *Revise MRI Tutorials.* http:/www.revisemri.com/tutorials/.

4. Schmeets, S., “fmri in clinical practice,” *Siemens Medical Solutions*, 2002.

5. Blank, A. G., *TCP/IP JumpStart–Internet Protocol Basics*, SYBEX Inc., 2nd ed., 2002.

6. Haffner, P., L. Bottou, P. G. Howard, P. Simard, Y. Bengio, and Y. LeCun, “High quality document image compression with djvu,” *Journal of Electronic Imaging*, pp. 410–425, July 1998.

7. Dostálek, L., and A. Kabelová, *Understanding TCP/IP: A clear and comprehensive guide to TCP/IP protocols*, Packt Publishing, 2006.

8. Mansfield, N., *Practical TCP/IP: Designing, using, and troubleshooting TCP/IP networks on Linux and Windows*, Addison–Wesley, 2003.

9. Bankman, I., *Handbook of Medical Image Processing and Analysis*, Academic Press, 2nd ed., 2000.

10. Kuperman, V., *Magnetic Resonance Imaging Physical Principles and Applications (Electromagnetism)*, Academic Press, 2000.

11. Reiser, M. F., W. Semmler, and H. Hricak, *Magnetic Resonance Tomography*, Springer, 2007.

12. Brown, M. A., and R. C. Semelka, *Mri: Basic Principles and Applications*, Wiley-Liss, 1999.

13. Lambert, J. B., and E. P. Mazzola, *Nuclear Magnetic Resonance Spectroscopy: An Introduction to Principles, Applications, and Experimental Methods*, Pearson Prentice Hall, 2003.

14. JA, J. D., J. Leigh, D. Williams, and A. Koretsky, “Perfusion imaging,” *Magnetic Resonance in Medicine*, no. 23, pp. 37–45, 1992.

15. Ogawa, S., T. Lee, A. Kay, and D. Tank, “Brain magnetic resonance imaging with contrast dependent on blood oxygenation,” *Proceedings of the National Academy of Sciences*, Vol. 87, pp. 9868–9872, 1990.

16. Dirnagl, U., K. Niwa, U. Lindauer, and A. Villringer, “Coupling of cerebral blood flow to neuronal activation: Role of adenosine and nitric oxide,” *American Journal of Physiology*, Vol. 267, pp. 296–301, 1994.

17. Edelman, R., B. Siewert, D. Darby, V. Thangaraj, A. Nobre, M. Mesulam, and S. Warach, “Qualitative mapping of cerebral blood flow and functional localization with echo–planar mr imaging and signal targeting with alternating radio frequency,” *Radiology*, Vol. 192, pp. 513–520, Aug 1994.

18. Tuunanen, P., and R. Kauppinen, "Effects of oxygen saturation on bold and arterial spin labeling perfusion fmri signals studied in a motor activation task," *Neuroimage*, Vol. 20, pp. 102–109, 2006.

19. Basser, B., J. Mattiello, and D. L. Bihan, "Estimation of the effective self–diffusion tensor from the nmr spin echo," *Journal of Magnetic Resonance*, Vol. 103, pp. 247–254, 1994.

20. Casad, J., *Teach Yourself TCP/IP in 24 Hours*, Sams Publishing, 3rd ed., 2003.

21. Naugle, M., *Illustrated TCP/IP: A Graphic Guide to the Protocol Suite*, John Wiley and Sons, 1999.

22. Sportack, M. A., *TCP/IP First–Step*, Cisco Press, 2004.

23. Wright, G. R., and W. R. Stevens, *TCP/IP Illustrated, Volume 2: The Implementation*, Addison Wesley, 1995.

24. Parziale, L., D. T. Britt, C. Davis, J. Forrester, W. Liu, C. Matthews, and N. Rosselot, *TCP/IP Tutorial and Technical Overview*, IBM International Technical Support Organization, 2006.

25. Loshin, P., *TCP/IP Clearly Explained*, Morgan Kaufmann Publishers, 4th ed., 2003.

26. Osterloh, H., *TCP/IP Primer Plus*, Sams Publishing, 2002.

27. Held, G., *TCP/IP Professional Reference Guide*, Auerbach, 2001.

28. Xueyu, G., W. Shunyao, W. Jinlong, and L. P, "An effective remote control system based on tcp/ip," *International Conference on Advanced Computer Theory and Engineering*, pp. 425–429, 2008.

29. Huifeng, S., L. Yan, W. Feng, and L. Shipeng, "A high–performance remote computing platform," *International Conference on Advanced Computer Theory and Engineering*, pp. 1–6, 2009.

30. Wenbin, J., J. Hai, G. Minyi, S. Zhiyuan, and Z. Zhangnan, "A novel remote screen synchronization mechanism for ubiquitous environments," *Proceedings of 1st International Symposium on Pervasive Computing and Applications*, pp. 725–730, August 2006.

31. Zhang, Y., *Remote control programming technology*, Beijing: lectronics Industry Press, 2002.

32. Wang, J., and S. Zhang, "Internet remote control system," *Computer Applications*, Vol. 20, no. 12, pp. 62–65, 2000.

33. Yang, S., J. Fu, Z. Shi, and D. Liao, "Remote control of the industrial robot based on internet," *Aplication Research of Computers*, Vol. 19, no. 6, pp. 151–153, 2005.

34. Yi, C., and R. He, "Application of client/server architecture in the multimedia remote classroom measuring and monitroing system," *Beijing Industry and Commerce University Journal*, Vol. 23, no. 3, pp. 35–37, 2005.

35. Bennett, K., P. Layzell, D. Budgen, earl Brereton, L. Macaulay, and M. Munroe, "A service–based software: The future for flexible software," *Proceedings of Asia–Pasific Software Engineering Conference (APSEC)*, pp. 214–221, December 1999.

36. Richardson, T., Q. Stafford-Fraser, K. R. Wood, and A. Hopper, "Virtual network computing," *IEEE Internet Computing*, Vol. 2, pp. 33–38, January 1998.

37. Kaplinsky, K. V., "Vnc tight encoder – data compression for vnc," *Proceedings of the 7th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists on Modern Techniques and Technologies*, pp. 155–157, February 2001.

38. *Microsoft Corporation, Remote Desktop Protocol.* http:/msdn.microsoft.com/en–us/library/aa383015.aspx.

39. Baratto, R. A., J. Nieh, and L. Kim, "Thinc: A remote display architecture for thin-client computing," tech. rep., Columbia University, CUCS-027-04, July 2004.

40. Scheifler, R. W., and J. Gettys, "The x window system," *ACM Transactions on Graphics*, Vol. 5, pp. 79–109, April 1986.

41. Yang, C., Y. Niu, Y. Xia, and C. Xu, "A fast and efficient codec for multimedia applications in wireless thin–client computing," *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1–12, June 2007.

42. Christiansen, B. O., K. E. Schauser, and M. Münke, "A novel codec for thin client computing," *Proceedings Of the Data Compression Conference*, pp. 13–22, March 2000.

43. Li, F., and J. Nieh, "Optimal linear interpolation for server–based computing," *Proceedings of IEEE International Conference on Communications*, pp. 2542–2546, April 2002.

44. Li, Q., and F. Li, "Fce: a fast content expression for server–based computing," *Proceedings of IEEE International Conference on Communications*, pp. 1426–1430, June 2004.

45. Buxton, R., L. Frank, E. Wong, B. Siewert, S. Warach, and R. R. Edelman, "A general kinetic model for quantitative perfusion imaging with arterial spin labeling," *Magn Reson Med*, Vol. 40, pp. 383–396, Sep 1998.