

Design and Development of a 3D X-ray Microscope

by

Jordan B. Brayonov

B.S. Electrical Engineering and B.S. Mechanical Engineering
MIT, 2004

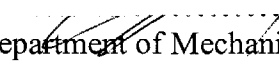
Submitted to the Department of Mechanical Engineering in Partial Fulfillment of the
Requirements for the Degree of

Master of Science in Mechanical Engineering
at the
Massachusetts Institute of Technology

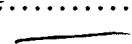
February, 2006

©2006 Massachusetts Institute of Technology
All Rights Reserved

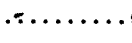
Signature of Author

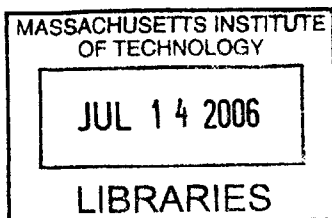

Department of Mechanical Engineering
January 20, 2006

Certified by


Ian W. Hunter
Hatsopoulos Professor of Mechanical Engineering and
Professor of Biological Engineering
Thesis Supervisor

Accepted by


Lallit Anand
Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Students



ARKER



Room 14-0551
77 Massachusetts Avenue
Cambridge, MA 02139
Ph: 617.253.2800
Email: docs@mit.edu
<http://libraries.mit.edu/docs>

DISCLAIMER NOTICE

The accompanying media item for this thesis is available in the MIT Libraries or Institute Archives.

Thank you.

Design and Development on a 3D X-ray Microscope

by

Jordan B. Brayonov

Submitted to the Department of Mechanical Engineering
on January 20th, 2006 in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Mechanical Engineering

ABSTRACT

The rapid development of needle-free injection systems demands better and faster imaging systems, capable of imaging the transient and steady state response of an injection into real tissue. X-ray radiography, x-ray microscopy, and computerized tomography (CT) were identified as the most appropriate imaging techniques used for non-invasive imaging of opaque objects and an instrument was constructed, utilizing all of those techniques.

The instrument was capable of producing x-ray radiographies with native resolution of less than 30 μm , frame rate of 32 frames per second and a field of view of 30 mm by 40 mm. Using magnifying cone-beam x-ray source the ultimate resolution of the system exceeded 5 μm , while maintaining image sharpness and clarity. Digital image reconstruction techniques, based on the inverse Radon transformation and the Fourier slice theorem were used to regenerate conventional CT images as well as contrast volumetric 3D images of the internal structures of opaque objects.

This paper presents the design and development of the 3D x-ray microscope together with experimental results obtained during the calibration and initial testing of the instrument.

Thesis Supervisor: Ian W. Hunter

Title: Professor of Mechanical Engineering and Professor of Biological Engineering

Table of Contents

ABSTRACT	2
TABLE OF CONTENTS	3
LIST OF FIGURES	6
LIST OF TABLES	9
ACKNOWLEDGEMENTS	10
1. INTRODUCTION AND MOTIVATION	11
2. HISTORICAL BACKGROUND	12
2.1. X-RAY RADIOGRAPHY	12
2.2. CONVENTIONAL X-RAY TOMOGRAPHY	15
2.3. COMPUTED X-RAY TOMOGRAPHY	17
2.4. EVOLUTION OF CT SCANNERS	19
2.4.1. FIRST GENERATION CT SCANNER	19
2.4.2. SECOND GENERATION CT SCANNER	20
2.4.3. THIRD GENERATION CT SCANNER	21
2.4.4. FOURTH GENERATION CT SCANNER	22
2.4.5. FIFTH GENERATION CT SCANNER	23
3. THEORETICAL FUNDAMENTALS	25
3.1. MATHEMATICS FUNDAMENTALS	25
3.1.1. LINEAR ALGEBRA	25
3.1.2. FOURIER ANALYSIS	29
3.2. FUNDAMENTALS OF X-RAY PHYSICS	31
3.2.1. GENERATION OF X-RAYS	31
3.2.2. X-RAYS – MATTER INTERACTIONS	33
3.2.3. RADIATION DOSE	36
4. IMAGE PROCESSING	39
4.1. IMAGE RECONSTRUCTION	39
4.1.1. LINE INTEGRAL MEASUREMENTS	39
4.1.2. ALGEBRAIC RECONSTRUCTION TECHNIQUE	41
4.1.3. RADON TRANSFORM	44

4.1.4.	THE FOURIER SLICE THEOREM	45
4.1.5.	FAN BEAM RECONSTRUCTION	50
4.2.	IMAGE ENHANCEMENT	53
4.2.1.	HISTOGRAM TRANSFORMATIONS	54
4.2.2.	IMAGE AVERAGING	56
4.2.3.	MEDIAN FILTERING	58
4.2.4.	IMAGE SHARPENING AND EDGE ENHANCEMENT	59
4.3.	IMAGE MAGNIFICATION	61
4.4.	IMAGE VISUALIZATION	63
4.4.1.	TRADITIONAL CT IMAGE VISUALIZATION	63
4.4.2.	VOLUMETRIC VISUALIZATION	65
5.	X-RAY MICROSCOPE DESIGN	67
5.1.	FUNCTIONAL COMPONENTS	68
5.2.	STRUCTURE	68
5.3.	MECHANICS	71
5.4.	ELECTRONICS	73
5.5.	SOFTWARE	74
5.5.1.	X-RAY SOURCE CONTROL SOFTWARE	75
5.5.2.	ROTARY STAGE CONTROL SOFTWARE	76
5.5.3.	IMAGE ACQUISITION SOFTWARE	76
5.5.4.	IMAGE RECONSTRUCTION SOFTWARE	76
5.6.	DESIGN SUMMARY	77
6.	X-RAY MICROSCOPE IMPLEMENTATION	79
6.1.	FUNCTIONAL COMPONENTS	79
6.1.1.	DIGITAL CCD IMAGING CAMERA	79
6.1.2.	X-RAY SOURCE	80
6.2.	STRUCTURE	81
6.3.	ELECTRONICS	82
6.4.	MECHANICS	87
6.5.	SOFTWARE	88
6.5.1.	X-RAY SOURCE CONTROL SOFTWARE	88

6.5.2. ROTARY STAGE CONTROL SOFTWARE	90
6.5.3. IMAGE ACQUISITION SOFTWARE	91
6.5.4. IMAGE RECONSTRUCTION SOFTWARE	93
6.5.4.1.TWO-DIMENSIONAL RADIOGRAPHY GENERATION	94
6.5.4.2.CONVENTIONAL 3D CT IMAGE GENERATION	95
6.5.4.3.VOLUMETRIC 3D CT IMAGE GENERATION	96
6.6. COMPLETE SYSTEM OVERVIEW	96
7. RESULTS AND DISCUSSION	97
7.1. CALIBRATION	97
7.1.1. LEAD ENCLOSURE	97
7.1.2. X-RAY SOURCE	98
7.1.3. DIGITAL CCD CAMERA	99
7.1.4. MICROSCOPE MAGNIFICATION	102
7.2. TWO-DIMENSIONAL RADIOGRAPHS	104
7.3. CONVENTIONAL 3D TOMOGRAPHIC IMAGES	106
7.4. VOLUMETRIC 3D TOMOGRAPHIC IMAGES	111
8. CONCLUSIONS AND FUTURE WORK	113
BIBLIOGRAPHY	115
APPENDIX A: EQUIPMENT SPECIFICATIONS	118
APPENDIX A.1: CAMERA SPECIFICATIONS	118
APPENDIX A.2: X-RAY SOURCE SPECIFICATIONS	119
APPENDIX A.3: POWER SUPPLY SPECIFICATIONS	120
APPENDIX B: SOFTWARE	121
APPENDIX B.1: SOURCE CODE FOR ROTARY STAGE CONTROL	121
APPENDIX B.2: IMAGE ENHANCEMENT GUI	126
APPENDIX B.3: SOURCE CODE FOR CONVENTIONAL CT RECONSTRUCTION	130
APPENDIX B.4: SOURCE CODE FOR VOLUMETRIC CT RECONSTRUCTION	131

List of Figures

Figure 2.1: Sample radiographs of human hands.	13
Figure 2.2: Early 1900s radiographic system.	13
Figure 2.3: Radiographic system using fluorescent screen.	14
Figure 2.4: Shoe fitting X-ray device.	14
Figure 2.5: Principle of the first conventional tomography device.	15
Figure 2.6. Conventional tomography.	16
Figure 2.7: Hounsfield working on the first laboratory CT scanner.	18
Figure 2.8: Early clinical CT imaging.	19
Figure 2.9: First generation CT scanner.	20
Figure 2.10: Second generation CT scanner.	21
Figure 2.11. Third generation CT scanner.	22
Figure 2.12: Fourth generation CT scanner.	23
Figure 2.13: Fifth generation CT scanner.	24
Figure 3.1: Basic Fourier transform properties.	31
Figure 3.2: Electromagnetic spectrum.	33
Figure 4.1: Monochromatic x-ray attenuation.	40
Figure 4.2: A simple example of an object and its projections.	42
Figure 4.3: Illustration of algebraic reconstruction technique (ART).	43
Figure 4.4: Line integral projection $P(p, \theta)$ of the 2-D Radon transform.	45
Figure 4.5: The Fourier Slice theorem relates the Fourier transform of a projection to the Fourier transform of the object along a radial line.	47
Figure 4.6: Sampling pattern in frequency domain based on Fourier slice theorem.	50
Figure 4.7: Different fan beam geometries.	51
Figure 4.8: Fan beam projection pattern.	51
Figure 4.9: Projection patterns for different beam geometries.	52
Figure 4.10: Sample brain CT image with histogram equalization.	55
Figure 4.11: Sample brain CT image with spatial averaging.	57
Figure 4.12: 3 by 3 pixel neighborhood.	58

Figure 4.13: Sample brain CT image with median filtering.	59
Figure 4.14: First-order gradient (Sobel filter) masks.	60
Figure 4.15: Sample brain CT image with Laplacian filtering.	61
Figure 4.16: Magnification by using cone-beam x-ray source.	62
Figure 4.17: CT image of a human head.	64
Figure 4.18: Reconstructed “slice” images sequentially arranged in a "stack."	66
Figure 4.19: MPR image of a human lumbarosacral spine.	66
Figure 5.1: General system overview.	67
Figure 5.2: CAD model of aluminum support frame.	69
Figure 5.3: CAD model of lead enclosure with missing front panels	70
Figure 5.4: CAD model of main structure	71
Figure 5.5: Picture of mechanical safety interlock switches	72
Figure 5.6: Key mechanical components.	73
Figure 5.7: Main wiring diagram.	74
Figure 5.8: Major software components interactions.	75
Figure 5.9: Image reconstruction algorithm flow diagram	77
Figure 5.10: CAD model of complete system.	78
Figure 6.1: Hamamatsu C4742-56-12NR digital CCD camera setup.	80
Figure 6.2: Source Ray SB-80-500 x-ray source.	81
Figure 6.3: Aluminum support frame.	82
Figure 6.4: Lead shielded enclosure.	83
Figure 6.5: X-ray source support electronics.	84
Figure 6.6: Illumination.	85
Figure 6.7: Power supply enclosure.	86
Figure 6.8: Electrical safety components.	86
Figure 6.9: Mechanical interlock safety switch and counterweight system.	87
Figure 6.10: Rotary stage together with its stepper motor and drive.	88
Figure 6.11: X-ray source control software GUI.	89
Figure 6.12: Communication scheme used to control the rotary stage.	90
Figure 6.13: Rotary Stage Control software GUI.	91
Figure 6.14: Wasabi main screen.	92

Figure 6.15: HyperCam interface.	93
Figure 6.16: Flowchart of the entire image reconstruction process.	94
Figure 6.17: Image enhancing application.	95
Figure 6.18: Picture of complete system.	96
Figure 7.1: Results from the cooling system evaluation test.	98
Figure 7.2: Image histogram	99
Figure 7.3: Noise distribution analysis for V=40 kV	100
Figure 7.4: Noise distribution analysis for V=35 kV	100
Figure 7.5: Contrast test phantom.	101
Figure 7.6: Resolution test phantom.	102
Figure 7.7: Magnification test.	103
Figure 7.8: Images of DAC725JP microchip.	104
Figure 7.9: Images of 256 MB I-Stick USB Drive.	105
Figure 7.10: Endoscopic probe operating inside an opaque enclosure.	105
Figure 7.11: Radiography of simple cylindrical phantom.	106
Figure 7.12: Initial reconstruction attempt of conventional CT image.	107
Figure 7.13: Improved reconstruction of conventional CT image of needle phantom.	107
Figure 7.14: Solid stainless-steel phantom (lure-lock).	108
Figure 7.15: Reconstructed conventional CT image of lure-lock phantom.	109
Figure 7.16: Snail shell phantom.	109
Figure 7.17: 12 reconstructed layers of the snail shell phantoms.	110
Figure 7.18: Radiograph of the snail shell phantom.	111
Figure 7.19: Volumetric 3D images of the snail shell phantom.	111

List of Tables

Table 3.1: Summary of basic matrix properties	27
Table 3.2: Quality factors of various types of ionizing radiation.	37
Table 3.3: Radiation sensitivity of various organs.	38
Table 3.4: Effective Dose Equivalent H_E for Clinical X-ray and CT Exams	38
Table 5.1: System requirements.	68

Acknowledgements

I would like to express my gratitude to Professor Ian Hunter for the opportunity he gave me to work on this project. I would also like to thank him for his support and encouragements in conducting this research and interpreting the results.

My gratitude extends to Dr. Andrew Taberner for helping me devise relevant experiments and assisting me in resolving issues with the laboratory equipment and my experiments.

I would like to pay my sincere gratitude to my colleague Atanas “Nasko” Pavlov and Andrea Bruno for their moral support and their insight in the subjects of x-ray physics and image processing.

I render my special thanks to my girlfriend Eleonora “Ellie” Vidolova for generously giving me her time in editing my thesis and helping me with the visually appealing arrangement of this document.

Last, but not least, I would like to thank my friends Boyan Yordanov, Nayden Kamboushev, Iliya Tsekov, and Nikolay Andreev for proof-reading my thesis and providing me with invaluable feedback.

1. Introduction and Motivation

X-ray imaging has been a popular research topic for over a century in various scientific fields. Radiography, tomography, and microscopy systems using x-rays have been studied and carefully documented for decades; yet, with the development of new technologies and image processing algorithms, rapid improvement of existing instrumentation is not unusual.

Imaging system development usually follows along the development of new experimental techniques, which need to be observed and documented. Over the last few years we had been developing a novel Needle-Free Injection (NFI) system for both veterinary and human use and we had achieved most of our goals. However, due to limitations in the conventional imaging system we had been using to document our progress, it had been extremely difficult to document and present our success.

The purpose of our injection system was to inject fluids into live tissue. Since live tissue is opaque to visible light, injections could neither be observed, nor documented in real time and so we decided to use an x-ray based 3D microscopy system, which could image inside the tissue and record the effects of the injection.

Even though 3D x-ray microscopy systems were commercially available and could be customized to meet most of our specifications, their delivery time was between six and twelve months, which meant huge delays in our progress. Furthermore, all commercial systems had fixed architecture and were not easily reconfigurable, which was essential to any future experiments for which the system could be used.

The best option we had was to develop an instrument that would meet all our requirements and provide us with the imaging resources we desperately needed. We also decided to build the system flexible and modular, so that it could easily be upgraded or modified in the future as new experimental techniques were developed.

2. Historical Background

X-rays were first observed in the late 1800s by scientists around the world. In 1887 Nikola Tesla began investigating the unknown radiation produced by the high-voltage vacuum tubes he designed. Heinrich Hertz experimented and demonstrated in 1892 that cathode rays could penetrate very thin aluminum foil and later developed a cathode ray tube that would emit x-rays capable of penetrating various metals. However it was Wilhelm Röntgen in November 1895 who first documented x-rays as a new type of radiation.

On December 28th 1895 Röntgen wrote a preliminary report “*On A New Kind of Rays,*” which was the first formal recognition and categorization of x-rays. In this report he referred to the rays as “X”, indicating that they were unknown to the scientific society at that time. Wilhelm Röntgen was awarded an honorary degree of Doctor of Medicine in 1896 and received the first Nobel Prize in Physics in 1901 for his discovery. He refused to take any patents related to his discovery on moral grounds and did not even want the rays to be named after him. However, many of his colleagues started calling them “Röntgen rays,” a term still in use in many countries.

2.1. X-ray Radiography

In parallel with the discovery of x-rays, Dr. Röntgen was working on possible applications for them. He discovered that if an object of variable density is positioned between a cathode ray tube (an x-ray source) and a photographic film and irradiated with x-rays generated by the tube, a contrast image, called a radiograph, is produced. Figure 2.1 (a) shows a radiograph of Bertha Röntgen’s hand taken on November 8th 1895. The radiograph clearly shows the contrast between tissue, bone, and a metal ring and lays the foundation of 100 years of non-invasive medical imaging. Figure 2.1 (b) shows a radiograph taken by a modern x-ray imaging system, with much better soft tissue contrast. Nevertheless, Dr. Röntgen’s radiography is remarkable in retrospect.

Since Röntgen’s discovery that x-rays can be used identify bony structures, scientists all over the world began developing medical imaging devices based on x-rays. Figure 2.2 shows an early 1900s radiography system, in which the patients had to hold the film

while being irradiated. In the earliest days a chest x-ray would require up-to 10 minutes of exposure and a radiation dosage 50 times greater than that of a modern x-ray.

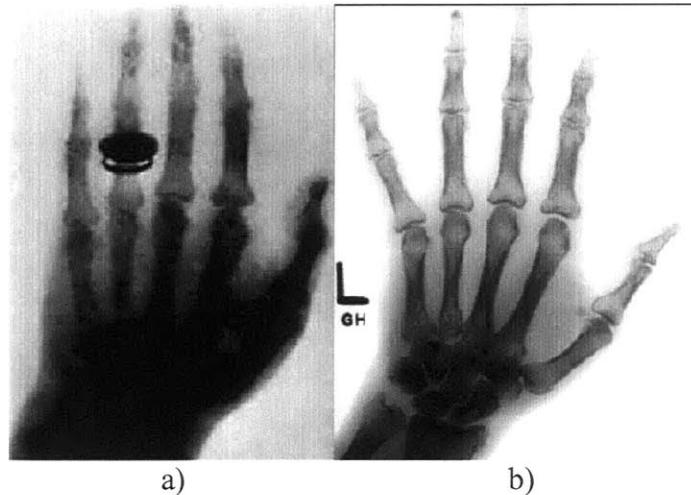


Figure 2.1: Sample radiographs of human hands.
a) Image of Bertha Röntgen's hand,
b) Image taken by a modern X-ray system.
[Images taken from Wikipedia, 2006]

In the early days of non-invasive medical imaging no one realized the danger posed by prolonged exposure to x-rays. The matter became worse with the development of fluorescent screens, shown in Figure 2.3, and special glasses, allowing doctors to see x-ray images in real time. This caused doctors to stare directly into the x-ray beam, creating unnecessary exposure to radiation.

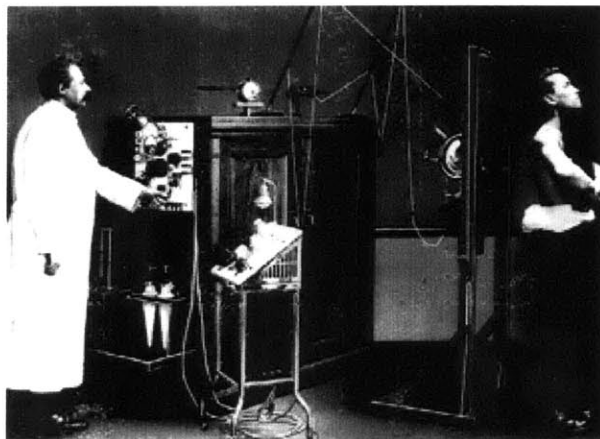


Figure 2.2: Early 1900s radiographic system.
[Image taken from Imaginis, 2006]



Figure 2.3: Radiographic system using fluorescent screen.
[Image taken from Imaginis, 2006]

In the commercial sector people started using x-rays for inspection, quality control, even shoe fitting. In the late 1940's a shoe fitting x-ray unit, similar to the one shown in Figure 2.4, was present in most shoe stores in the US. It was estimated that by the late 1950's over 10,000 of these devices were in use. It was in the 1950's when the radiation hazards associated with x-rays were first recognized and the first regulations were imposed over the use of x-rays.



Figure 2.4: Shoe fitting X-ray device.
[Image taken from MQMD, 2006]

Around the beginning of the 1920s people started to realize that they could not overcome the poor low-contrast resolution of conventional radiography. They also realized that they could not distinguish between structural layers of the subjects being studied, which led to the development of the conventional tomography.

2.2. Conventional X-ray Tomography

Conventional tomography, also known as body section radiography, was first suggested by Mayer in 1914. As early as 1921 Bocage, one of the pioneers of conventional tomography, described an apparatus which could blur structures above and below a plane of interest. The major components of his invention were an x-ray tube, a film, and a mechanical coupling which allowed synchronous movement of the tube and the film. The principle of this device is illustrated in Figure 2.5.

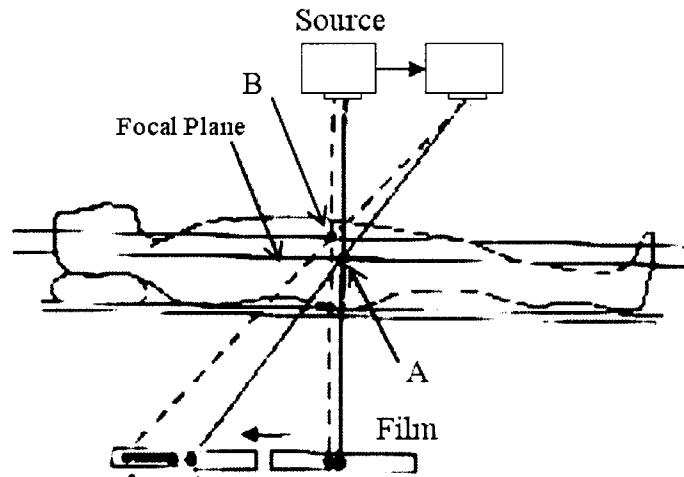


Figure 2.5: Principle of the first conventional tomography device.

[Image taken from EPM, 2006]

The idea behind the device was simple: let us consider two distinct points inside a patient, A and B, and let point A lie on the focal plane and point B lie off the focal plane. As the film and the x-ray source move, the projection of point A onto the film stays the same (which is true for all point on the focal plane), while the projection of point B changes to a different location on the film. This is due to the fact that point B is off the focal plane and the ratio of the distance from point B to the x-ray source and the distance from point B to the film deviates from that of the same distance ratio for point A. When

the x-ray tube and the film move continuously along a straight line, the shadow of point B forms a line segment instead of a point. This holds for all point above and below the focal plane. Therefore in the final image the highest intensity component would be produced by the point on the focal plane.

The first device based on that approach was successfully built in 1924. Devices using the same technology (with very minor changes) were used until the late 1960s. Figure 2.6 (a) shows a picture of such device, which instead of longitudinal travel uses a swiveling motion to ensure only one plane stays in focus. Figure 2.6 (b) shows an image produced by conventional tomography.

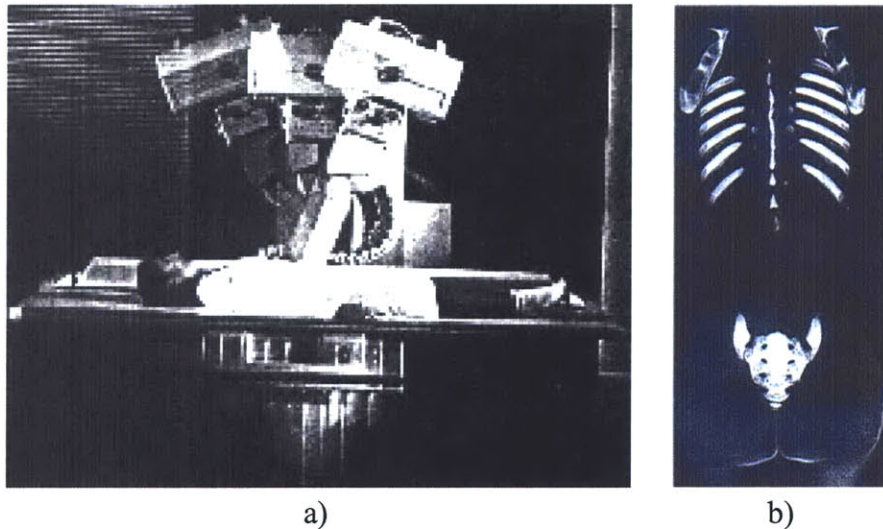


Figure 2.6. Conventional tomography.
a) Conventional tomography system,
b) Image, taken by conventional tomography.
[Images taken from EPM, 2006]

Conventional tomography was successful in producing images of the plane of interest; however these images still suffered from the fundamental limitation of radiography, namely the poor soft tissue contrast. Also, the blurred underlying structures were superimposed on the tomographic image and significantly deteriorated its quality. Combined with the larger x-ray dose to the patients and the new regulations being developed in the 1960s scientists started looking in a new direction. They began the development of Computed Tomography.

2.3. Computed X-ray Tomography

Computed tomography (CT) conceptually differs from conventional tomography in one aspect: the images are produced from series of projection, rather than from a single complex scan. It is quite remarkable that attempts at reconstruction of medical images from projections were made as early as 1940. In a patent granted in 1940 Gabriel Frank described the basic idea of today's computed tomography. The patent included drawings of equipment for generating sinograms (measured projection data as a function of angle) and an optical backprojection technique used to reconstruct the image. Needless to say, his approach could not benefit from the use of a modern computer technology so he had to rely of phosphorus screen to project the image, thus his approach suffered from blurring and other image artifacts. Nevertheless, the patent clearly indicated the fundamentals of a modern computed tomography device.

For over twenty years no advance was made in this technology until in 1963 David E. Kuhl and Roy Q. Edwards introduced transverse tomography with the use of radioisotopes, which was further developed and evolved into today's emission computed tomography. Their approach was to acquire a sequence of scans at uniform steps and regular angular intervals with two opposing radiation detectors. At each angle the film was exposed to a narrow line of light with a location and orientation corresponding to the detectors linear position. The process was repeated at 15-degree increments, as the film was rotated accordingly to sum up the backprojected views. The images produced from this device were reasonable, but what was missing from their approach was an exact reconstruction technique.

The mathematical formulation of an exact reconstruction technique for reconstructing an object from projections dates back to the 1917, when an Austrian mathematician J. Radon demonstrated mathematically that an object could be reconstructed from an infinite set of its projections taken at 0 to 180 degree angles (discussed in detail in Section 4.1.3). The concept was applied in practice for the first time in 1956 by R. N. Bracewell to reconstruct a map of solar microwave emissions from a series of radiation measurements across the solar surface. Between 1956 and 1958 several Russian papers accurately formulated the tomographic reconstruction as an inverse Radon transformation.

Allan M. Cormack started working in 1955 on what is considered the first computer assisted tomography (CAT) scanner ever built. He started his work in the Groote Schuur Hospital in Capetown where he worked as a hospital physicist part time. In 1956 he took a sabbatical and went to Harvard University where he derived the theory for image reconstruction. In 1957 Cormack joined the physics department at Tufts University and started building the first CT scanner. By 1963 he had a functioning prototype and he managed to reconstruct a circularly asymmetrical phantom made from aluminum and plastic. For this first prototype Cormack used a collimated gamma-ray beam as a source and the reconstruction required the solution of 28,000 simultaneous equations, bringing the time required for a single scan to 11 hours. Little to no attention was paid to the results from Cormack's studies at the time, due to the time and difficulty of performing the necessary calculations.



Figure 2.7: Hounsfield working on the first laboratory CT scanner.
[Image taken from Webb, 2003]

The development of the first clinical CAT scanner began in 1967 at the Central Research Laboratory of EMI, Ltd. in England. While studying pattern recognition techniques, Geoffrey N. Hounsfield discovered, independently of Cormack, that x-ray measurements taken through a body from different directions would allow the reconstruction of its internal structure. Preliminary calculations indicated that this approach would provide at least 100-time improvement of image contrast over conventional radiographs. Hounsfield completed his first laboratory CAT scanner in late 1967, as shown in Figure 2.7.

The first clinically available CT device, shown in Figure 2.8 (a), was installed in Atkinson-Morley Hospital in September 1971. Images from this device could be produced in 4.5 minutes and brain tumors less than 10 mm in diameter could be imaged, as shown in Figure 2.8 (b). Cormack and Haunsfield shared the Nobel Prize in Medicine in 1979 for their pioneer work in computed tomography.

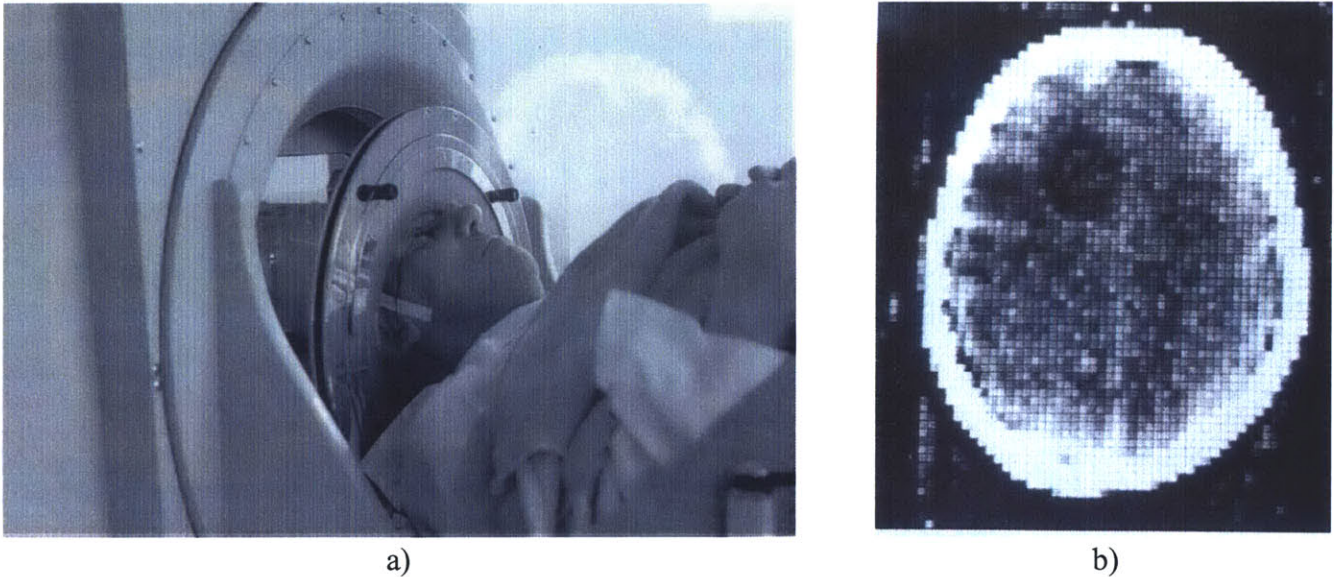


Figure 2.8: Early clinical CT imaging.
a) First clinical CT scanner,
b) Image, taken by the first clinical CT scanner.
[Images taken from Hsieh, 2003]

2.4. Evolution of CT Scanners

CT technology has been around for a little over 30 years. Yet, over those 30 years 5 generations of CT scanners have been developed, each making a breakthrough in patient treatment.

2.4.1. First Generation CT Scanner

The type of scanner built by EMI in 1971 is called “first generation” CT scanner. In a first generation scanner only one beam of data is measured at a time. In the original EMI head scanner the x-ray source was collimated to a narrow beam 3 mm wide and 13 mm long. The x-ray source and detector were linearly translated to acquire each individual

measurement. 160 measurements were acquired across the scan field, after which both the source and the detector were rotated by 1 degree to the next angular position. The schematic of the device is shown on Figure 2.9.

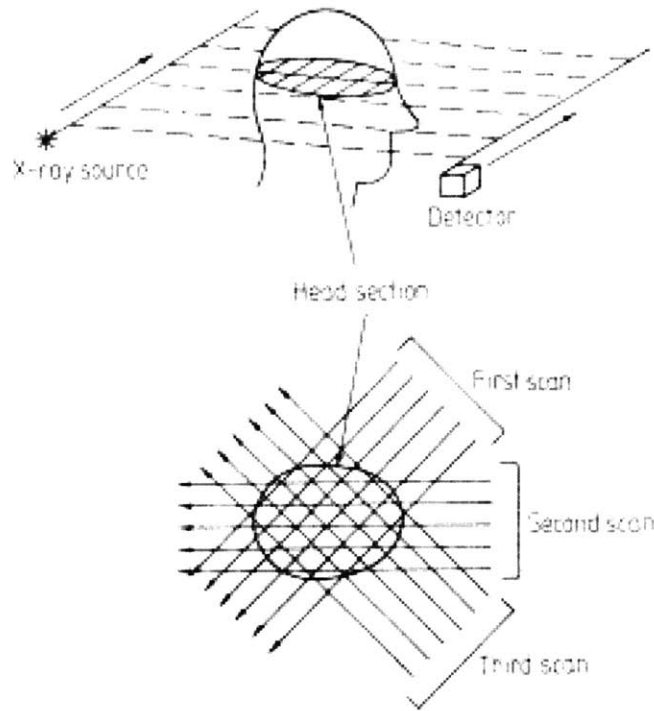


Figure 2.9: First generation CT scanner.
[Image taken from Ramapo College, 2006]

Although results from clinical evaluations of the first generation scanners were promising, there remained a serious image quality issue with patient movement during the 4.5 minute scan. The data acquisition time had to be reduced. This led to the development of the second generation scanners.

2.4.2. Second Generation CT Scanner

The basic operation of the second generation scanner is illustrated in Figure 2.10. Although this is still a translation-rotation scanner, the number of rotation steps is reduced by the use of multiple detectors. The design shown in the figure used 5 detectors and the angle between the beams corresponding to each detector is 1 degree. Therefore, for each translation the projections for 5 different angles are acquired. This allows the x-ray source and detector array to rotate 5 degrees at a time, meaning the overall scanning

time can be reduced by a factor of 5. By the end of 1975 EMI introduced a 30-detector scanner, capable of generating a complete scan in less than 20 seconds. This was a major breakthrough, since the scanning time was within the breath-holding range of most patients.

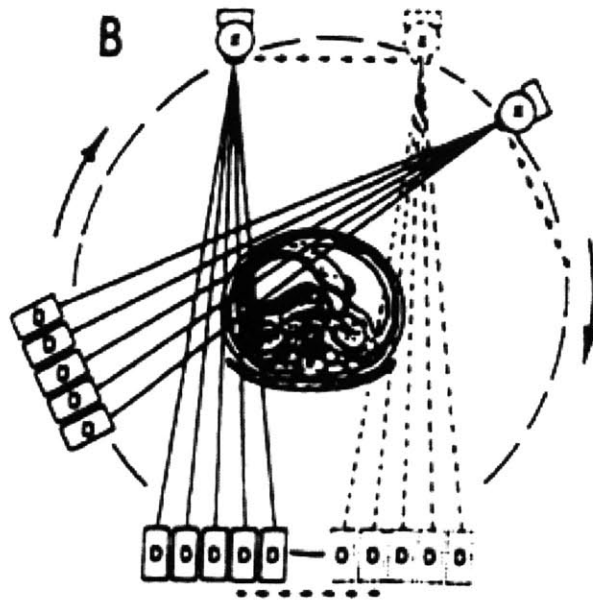


Figure 2.10: Second generation CT scanner.
[Image taken from Ramapo College, 2006]

2.4.3. Third Generation CT Scanner

One of the most popular scanner types is the third generation, illustrated in Figure 2.11. In this configuration a large number of detectors are located on a line or an arc sufficiently large so that the entire object is within the detector field. This completely eliminated the need for translation of the x-ray source of the detector array.

As the translation was eliminated, the data acquisition time was significantly reduced and was brought down to roughly 2 seconds per scan. In early models of the third generation scanners the x-ray source and the detectors were connected by cables, therefore the gantry had to rotate clockwise-counterclockwise to acquire adjacent scans. The acceleration/deceleration of the gantry, with a mass of several hundred kilograms, was the limiting factor in the overall system speed. Later models used slip rings for power and data transmission, which allowed the gantry to rotate at a constant speed and

helped reduce the total scan time to under 0.5 seconds per scan. Yet, the resolution of the instruments was limited by the spacing between detectors and by the detector cell size. This issue was resolved with the development of the fourth generation scanners.

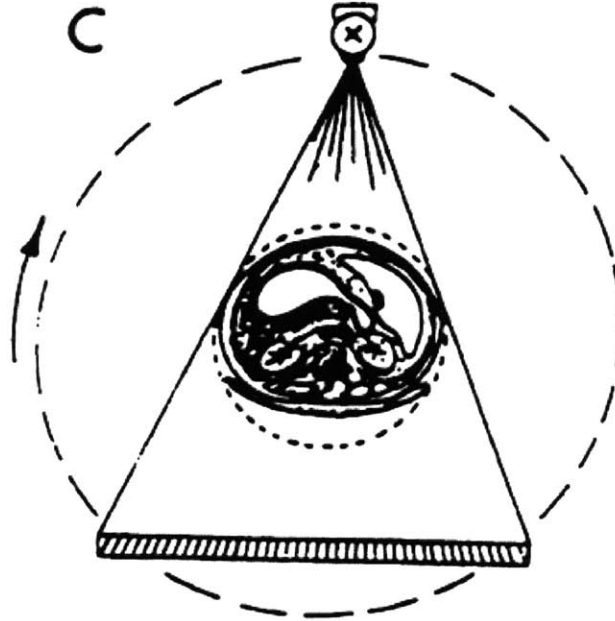


Figure 2.11. Third generation CT scanner.
[Image taken from Ramapo College, 2006]

2.4.4. Fourth Generation CT Scanner

In fourth generation scanners a stationary ring of detectors was used to capture the images as the x-ray source rotated about the patient, as shown in Figure 2.12. Unlike the third generation scanner, a projection was formed from the signals measured by a single detector as the x-ray beam swept across the object. One of the main advantages of this approach was the fact that the spacing between adjacent samples in a projection was determined by the sampling rate of the sensor, not by the spacing between sensors. The higher sampling rate helped reduce aliasing and drastically improve image quality. However, the large number of detectors required by a fourth generation scanner made it economically inefficient and undesirable.

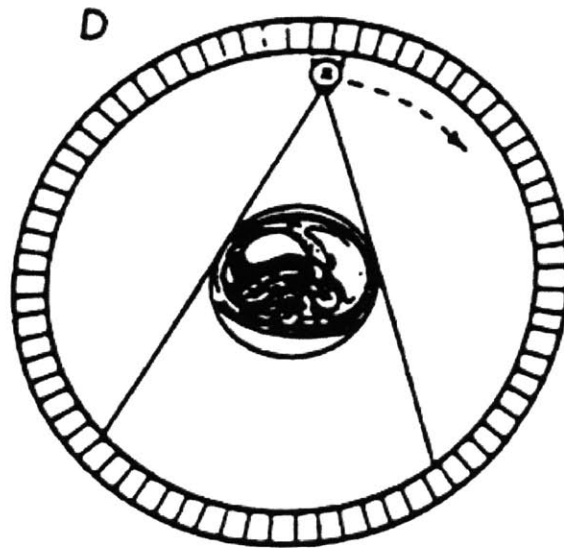


Figure 2.12: Fourth generation CT scanner.
[Image taken from Ramapo College, 2006]

2.4.5. Fifth Generation CT Scanner

The fifth generation scanner, also known as the electron-beam scanner or EBCT was developed between 1980 and 1984 mainly for cardiac applications. Its purpose was to capture still images of a beating human heart. To achieve this, a complete set of projections had to be collected in less than 100 milliseconds, something impossible for a third or fourth generation CT scanner because of the enormous forces required to rotate the gantry. In the EBCT the rotations of the x-ray source was replaced by a sweeping motion of an electron beam which was focused on the target ring, shown in Figure 2.13, which in turn emitted x-rays. The entire assembly was sealed in high vacuum, similar to a cathode ray tube. The x-rays generated at the target ring passed through the subject on the table and were collimated to a set of detectors located in the top portion of the detector ring. With the use of multiple target tracks and multiple detector arrays and with the absence of any movable mechanical parts, the scanning time was reduced to 50 milliseconds for up-to 80 mm section along the axis of the patient. Due to the high level of white noise present in these measurements, the averaging of multiple scans was often required to produce a high-quality image.

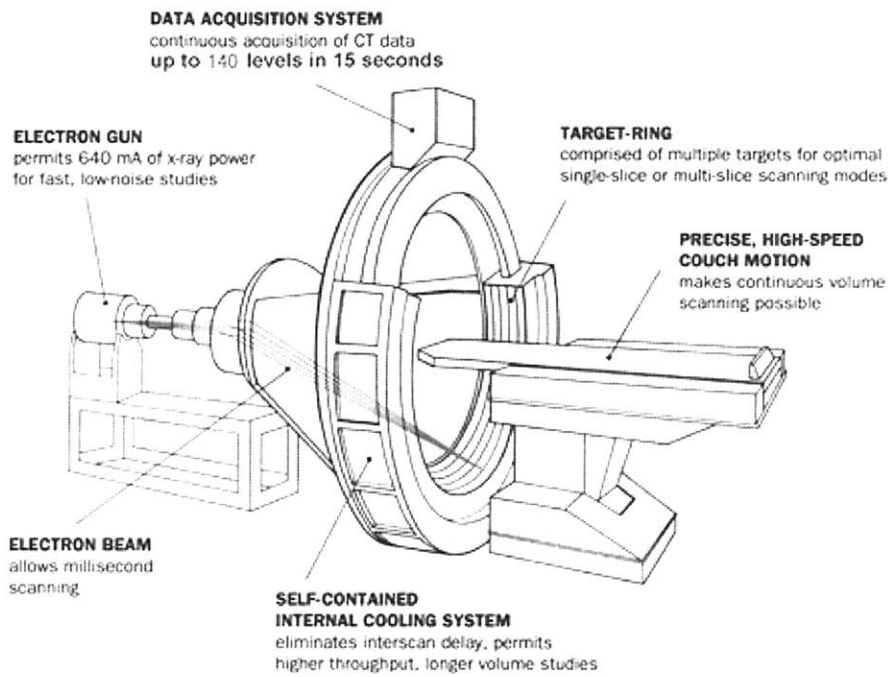


Figure 2.13: Fifth generation CT scanner.
[Image taken from Ramapo College, 2006]

3. Theoretical Fundamentals

Two sciences have helped develop the modern CT scanning technology: physics and mathematics. The objective of this chapter is to provide a review of the mathematical tools and physical phenomena used throughout the development of this project.

3.1. Mathematics Fundamentals

3.1.1. Linear Algebra

All of the data from a CT scan come in matrix form. Therefore matrix manipulations are the basis of all data analysis, image processing, and image representation. The successful understanding of the operation of a CT scanner required knowledge of basic matrix manipulations and their implementation.

An $m \times n$ matrix, A , is a rectangular array of numbers containing m rows and n columns:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}. \quad (3.1)$$

The transpose of that matrix, denoted by A^T , is an $n \times m$ matrix obtained by switching the rows and columns of A :

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}. \quad (3.2)$$

One special matrix is the square matrix for which $m = n$. Square matrixes are the most common type in imaging applications. For example, most modern CT scanners generate images with 512×512 pixels and store these images in square matrixes.

Diagonal matrixes are a sub-class of square matrixes. A diagonal matrix, D , is a square matrix whose all off-diagonal entries are zero:

$$D = \begin{bmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}. \quad (3.3)$$

Furthermore, if $d_{ii} = 1$ for $i \in [1..n]$ of a diagonal matrix, then this matrix is called the identity matrix and is denoted by I_n .

Another special type of an $m \times n$ matrix B is a matrix with $m = 1$:

$$B = [b_1 \quad b_2 \quad \cdots \quad b_n], \quad (3.4)$$

also called a row vector. The transpose of a row vector is a column vector, which is a matrix with $n = 1$. Most measurements taken from a detector array in a CT scanner form of a vector.

The sum of two $m \times n$ matrixes, $C = A + B$, is also an $m \times n$ matrix for which:

$$c_{ij} = a_{ij} + b_{ij}. \quad (3.5)$$

If A is an $m \times n$ matrix and B is an $n \times o$ matrix, the product $C = AB$ is an $m \times o$ matrix for which:

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}. \quad (3.6)$$

If A is a row vector and B is a column vector and both A and B have n elements, the product $s = AB$ is a 1×1 matrix or a scalar.

$$s = \sum_{k=1}^n a_k b_k. \quad (3.7)$$

The product of a scalar s and a matrix A is the matrix sA :

$$C = sA = \begin{bmatrix} sa_{11} & sa_{12} & \cdots & sa_{1n} \\ sa_{21} & sa_{22} & \cdots & sa_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ sa_{m1} & sa_{m2} & \cdots & sa_{mn} \end{bmatrix}. \quad (3.8)$$

These are most of the basic matrix manipulations used in the implementation of a CT scanner. Table 3.1 summarizes the properties derived from these basic manipulations.

Table 3.1: Summary of basic matrix properties

Associative Law	$(AB)C = A(BC)$
Distributive Law	$A(B + C) = AB + AC$
Additive Transpose	$(A + B)^T = A^T + B^T$
Multiplicative Transpose	$(AB)^T = A^T B^T$

So far we have only discussed traditional matrix manipulations. However, for the convenience of computer algorithms implementation we need to define a set of element-by-element matrix operators. We will only look at elemental addition, multiplication, division, and function application, denoted by: \oplus , \otimes , \backslash , and \aleph .

Elemental addition is the sum of a matrix A and a scalar s and is defined as:

$$C = s + A = \begin{bmatrix} s + a_{11} & s + a_{12} & \cdots & s + a_{1n} \\ s + a_{21} & s + a_{22} & \cdots & s + a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s + a_{m1} & s + a_{m2} & \cdots & s + a_{mn} \end{bmatrix}. \quad (3.9)$$

Elemental multiplication is the product of two $m \times n$ matrixes and is defined as

$$C = A \otimes B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1n}b_{1n} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2n}b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & a_{m2}b_{m2} & \cdots & a_{mn}b_{mn} \end{bmatrix}. \quad (3.10)$$

Similarly, the elemental division of two $m \times n$ matrixes is defined as:

$$C = A \setminus B = \begin{bmatrix} a_{11}/b_{11} & a_{12}/b_{12} & \cdots & a_{1n}/b_{1n} \\ a_{21}/b_{21} & a_{22}/b_{22} & \cdots & a_{2n}/b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}/b_{m1} & a_{m2}/b_{m2} & \cdots & a_{mn}/b_{mn} \end{bmatrix}. \quad (3.11)$$

We can also define elemental division of a scalar by an $m \times n$ matrix as:

$$C = s \setminus A = \begin{bmatrix} s/a_{11} & s/a_{12} & \cdots & s/a_{1n} \\ s/a_{21} & s/a_{22} & \cdots & s/a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s/a_{m1} & s/a_{m2} & \cdots & s/a_{mn} \end{bmatrix}. \quad (3.12)$$

Finally, we can define linear application of a linear function to an $m \times n$ matrix as:

$$C = \aleph(A) = \begin{bmatrix} \aleph(a_{11}) & \aleph(a_{12}) & \cdots & \aleph(a_{1n}) \\ \aleph(a_{21}) & \aleph(a_{22}) & \cdots & \aleph(a_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \aleph(a_{m1}) & \aleph(a_{m2}) & \cdots & \aleph(a_{mn}) \end{bmatrix}, \quad (3.13)$$

where \aleph can be any function which can operate over an element of the matrix.

Mathematicians argue that the definition of these operators is redundant, since all results from them can be obtained from a series of conventional linear algebraic operators [Strang, 1998]. However, in most modern signal and image processing software packages all these operators are pre-defined and readily available to the user, and their use greatly simplifies the implementation of most algorithms.

For further in-depth information on linear algebra and its use in CT image processing, one could refer to [Natterer, 2001].

3.1.2. Fourier Analysis

Fourier analysis is the basis of all signal and image processing. In one form or another all data acquisition, processing, and representation problems rely on the Fourier transform and its properties.

The one-dimensional Fourier transform of a continuous function $f(x)$ is defined as:

$$F(j\omega) = \int_{x=-\infty}^{\infty} f(x)e^{-j\omega x} d\omega, \quad (3.14)$$

where $j = \sqrt{-1}$. In CT scanner applications, since we are dealing with discrete, or digital, signals $f[n]$ we are more interested in the discrete Fourier transform, DFT, which is defined as:

$$F(e^{j\omega}) = \sum_{n=-\infty}^{\infty} f[n]e^{-j\omega n}. \quad (3.15)$$

The inverse DFT transform is defined as:

$$f[n] = F^{-1}(F(e^{j\omega})) = \frac{1}{2\pi} \int_{2\pi} F(e^{j\omega}) e^{j\omega n} d\omega. \quad (3.16)$$

The existence of this inverse is one of the most fundamental properties of all Fourier transforms.

The second fundamental property of the transform is its linearity. If $f[n] = af_1[n] + bf_2[n]$, then $F(e^{j\omega}) = aF_1(e^{j\omega}) + bF_2(e^{j\omega})$, where $F_1(e^{j\omega})$ and $F_2(e^{j\omega})$ are the Fourier transforms of $f_1[n]$ and $f_2[n]$ respectively. This property means that a linear combination of two or more functions yields a linear combination of their Fourier transforms.

Scaling is another property, best explained with continuous signals. The Fourier transform of $f(ax)$ is $\frac{1}{a}F\left(\frac{j\omega}{a}\right)$, which shows that scaling the function in time leads to

compression in frequency and increase of the magnitude of the frequency components. This property holds also in the DFT case; however there are mathematic complications, discussed in detail in [Oppenheim, 1998], which are irrelevant to our project and will be left out. Figure 3.1 shows examples of Fourier transform properties.

One of the most useful properties of the Fourier transform, which also hold for the DFT, is its application to convolution. The convolution of two functions, $f_1[n]$ and $f_2[n]$ is defined by:

$$g[n] = f_1[n] * f_2[n] = \sum_{k=-\infty}^{\infty} f_1[k] f_2[n-k]. \quad (3.17)$$

If we now take the DFT of Equation 3.17 we obtain:

$$G(e^{j\omega}) = F_1(e^{j\omega}) F_2(e^{j\omega}), \quad (3.18)$$

which shows that using DFT we can simplify a convolution to a simple product. This property, combined with fast numerical methods, like FFT [Bose, 2004], for calculating the Fourier transforms is the basis of all modern signal and image processing.

In CT scanner applications we are also interested in two-dimensional Fourier transforms. The two-dimensional DFT of a function $f[x, y]$ is defined as:

$$F(u, v) = \sum_{y=-\infty}^{\infty} \sum_{x=-\infty}^{\infty} f[x, y] e^{-j(ux+vy)}, \quad (3.19)$$

and its inverse is defined as:

$$f[x, y] = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} F(u, v) e^{j(ux+vy)} du dv. \quad (3.20)$$

All properties of the one-dimensional Fourier transforms can be extended to the two dimensional case [Oppenheim, 1998].

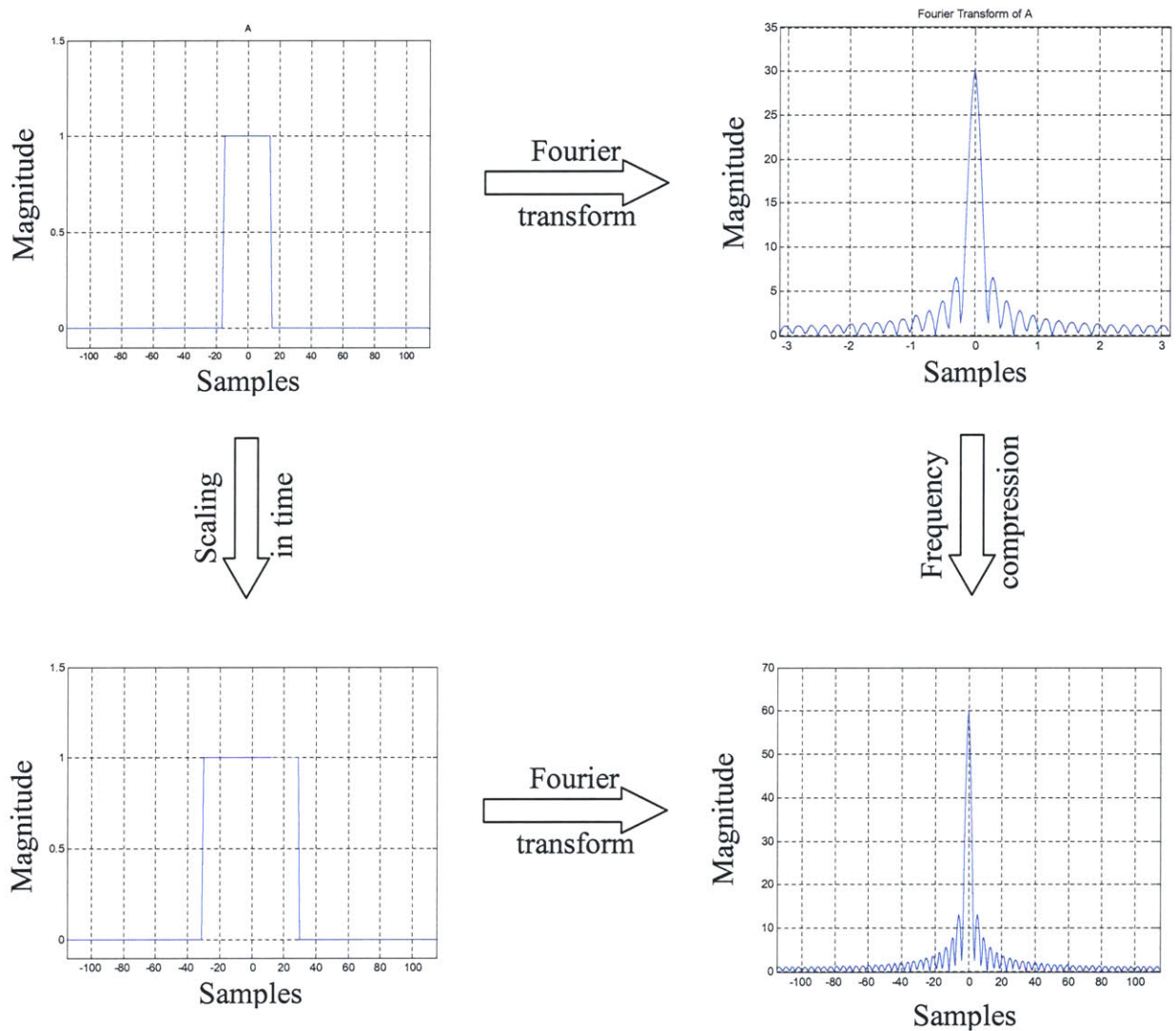


Figure 3.1: Basic Fourier transform properties.

3.2. Fundamentals of X-ray Physics

3.2.1. Generation of X-rays

X-rays are a type of electromagnetic waves, similar to microwaves, infrared, visible, and ultraviolet light, and radio waves. The wavelength of the x ray ranges from a few pico-meters ($1 \text{ pm} = 10^{-12} \text{ m}$) to a few nanometers ($1 \text{ nm} = 10^{-9} \text{ m}$). The energy of each x-ray photon, E , is proportional to its frequency, ν , and is given by:

$$E = h\nu = \frac{hc}{\lambda}, \quad (3.21)$$

where h is the Planck's constant ($h = 6.63 \times 10^{-34}$ J s), c is the speed of light, and λ is the wavelength of the x-rays. It is clear from Equation 3.21 that x-ray photons with longer wavelengths have lower energies than the photons of shorter wavelengths.

For convenience, the x-ray energy is usually expressed in units of electron-volts ($1 \text{ eV} = 1.602 \times 10^{-19}$ J). This is the amount of kinetic energy with which an electron is accelerated across an electrical potential of 1 V. Because x-ray photons are generally produced by striking a target material with high-speed electrons (the kinetic energy is transformed into electromagnetic radiation), the maximum possible x-ray photon energy equals the entire kinetic energy of the electron. For convenience in calculations, Equation 3.21 can be converted into units of eV per nm:

$$E = \frac{1.24 \times 10^3 \text{ eV} \cdot \text{nm}}{\lambda}. \quad (3.22)$$

X-rays with wavelengths in the range of 10 nm (124 eV) to 0.1 nm (12.4 keV) are often called “soft” x-rays, because they lack the ability to penetrate thicker layers of materials. These x-rays have little value in radiology. The wavelength of diagnostic x-rays varies roughly from 0.1 nm to 0.01 nm, corresponding to an energy range of 12.4 keV to 124 keV. Although x-rays with much shorter wavelengths are highly penetrating, they provide little low-contrast information and therefore, are of little interest to diagnostic imaging.

It is worth pointing out that x-rays are at the high-energy (short wavelength) end of the electromagnetic spectrum. Figure 3.2 shows the entire electromagnetic spectrum with different types of electromagnetic waves labeled.

X-ray photons are produced when a substance is bombarded with high-speed electrons. When a high-speed electron interacts with the target material, different types of collisions take place. The majority of the encounters involve small energy transfers from the high-speed electron to electrons that are knocked out of the atoms, leading to ionization of the target atoms. This type of interaction does not produce x-rays and gives

rise to delta rays and eventually heat. For a typical x-ray tube, over 99% of the input energy is converted into heat.

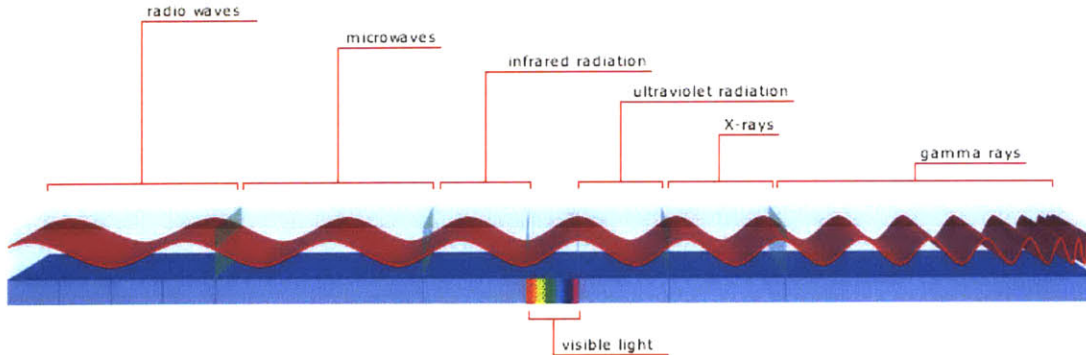


Figure 3.2: Electromagnetic spectrum.
 [Image taken from Wikipedia, 2006]

More interesting interactions occur when the electron approaches the nucleus of the atom and suffers a radiation loss. The high-speed electron travels partially around the nucleus due to the attraction between the positive nucleus and the negative electron. The sudden deceleration of the electron gives rise to the so-called bremsstrahlung radiation. The energy of the resulting radiation depends on the amount of incident kinetic energy that is given off during the interaction. If the energetic electron barely grazes the atomic coulomb field, the resulting x-rays have relatively low energy. As the amount of interaction increases, the resulting x-ray energy increases.

It is interesting to note that x-rays can also be produced by other charged particles, such as protons or alpha particles. The total intensity of the bremsstrahlung radiation, I , resulting from charged particle of mass m and charge z incident on target nuclei with charge Z , is proportional to:

$$I \propto \frac{Z^2 z^4 e^6}{m^2}. \quad (3.23)$$

Given that the mass of an electron is 2000 times less than that of a massive particle, an electron is more than 4×10^6 times more efficient in generating bremsstrahlung radiation than a massive particle such as a proton. This is why high-speed electrons are the practical choice for the production of x-rays. Equation 3.23 also

indicates that the bremsstrahlung radiation production efficiency increases rapidly as the atomic number of the target increases.

The second type of radiation generating interaction occurs when a high-speed electron collides with one of the inner-shell electrons of the target atom and liberates it. When the hole is filled by an electron from an outer shell, characteristic radiation is emitted. In the classic Bohr model of the atom, electrons occupy orbits with specific quantized energy levels. The energy of the characteristic x-ray is the difference between the binding energies of two shells. For example, the binding energies of the K, L, M, and N shells of tungsten are 70 keV, 11 keV, 3 keV, and 0.5 keV, respectively. When a K-shell electron is liberated and the hole is filled with an L-shell electron, a 59 keV x-ray photon is generated. Similarly, when an M-shell electron moves to the K-shell, the x-ray photon produced has 67 keV of energy. Note that each element in the periodic table has its own unique shell binding energies, and thus the energies of characteristic x-rays are unique to each atom.

Before concluding this section, we would like to point out the distinction between kV (electric potential) and keV (energy). The unit kVp is often used to describe an x-ray source. For instance, a 120 kVp x-ray source indicates that the applied electric potential across the x-ray tube is 120 kV. Under this condition, the electrons that strike the target have 120 keV of kinetic energy. The highest energy photons that can be produced by such a process have energy of 120 keV.

3.2.2. X-rays – Matter Interactions

The typical energy range of an x-ray photon generated in a medical CT scanner is between 20 keV and 140 keV. In this energy range there are three fundamental ways in which x-rays interact with matter: the photoelectric effect, the Compton effect, and coherent scattering.

We start with a description of the photoelectric effect. When the x-ray photon energy is greater than the binding energy of an electron, the incident x-ray photon gives up its entire energy and liberates an electron a shell of an atom. The free electron is called a photoelectron. The hole created at the deep shell is filled by an outer-shell electron. Because the outer shell electron is at higher energy state than the inner shell, a

characteristic radiation results. Thus, a photoelectron effect produces a positive ion, a photoelectron, and a photon of characteristic radiation. This effect was first explained by Albert Einstein in 1905, for which he received the Nobel Prize in Physics in 1921.

For tissue-like materials, the binding energy of the K-shell electrons is very small (roughly 500 eV). Hence, the photoelectron acquires essentially the entire energy of the x-ray photon. In addition, at such low energy, the characteristic x-rays produced in the interaction do not travel far, usually less than the dimensions of a human cell. Even for materials such as calcium (a bone constituent), the K-shell binding energy is only 4 keV. Because the mean free path of a 1 keV x-ray photon in muscle tissue is about $2.7 \mu\text{m}$, we can safely assume that all of the characteristic x-rays produced in patients by the photoelectric effect are reabsorbed.

The second type of interaction is the Compton effect, named after Arthur Holly Compton, who received the Nobel Prize in 1927 for its discovery. This is the most important interaction mechanism in tissuelike materials. In this interaction, the energy of the incident x-ray photon is considerably higher than the binding energy of the electron. An incident x-ray photon strikes an electron and frees the electron from the atom. The incident x-ray photon is deflected or scattered with partial loss of its initial energy. A Compton interaction produces a positive ion, a “recoil” electron, and a scattered photon. The scattered photon may be deflected at any angle from 0 deg to 180 deg. Low energy x-ray photons are preferentially backscattered (deflection angle larger than 90 deg), whereas high-energy photons have a higher probability of forward scattering (deflection angle smaller than 90 deg). Because of the wide deflection angle, the scattered photon provides little information about the location of interaction and the photon path.

Most of the energy is retained by the photon after a Compton interaction. The deflected photon may undergo additional collisions before exiting the patient. Because only a small portion of the photon energy is absorbed, the energy (or radiation dose) absorbed by the patient is considerably less than the photoelectric effect. The probability of a Compton interaction depends on the electron density of the material, not the atomic number Z . The lack of dependence on the atomic number provides little contrast information between different tissues (the electron density difference between different tissues is relatively small). Consequently, nearly all of the medical CT devices try to

minimize the impact of the Compton effect by either post-patient collimation or algorithmic correction.

The third type of interaction is the coherent scattering (also known as Rayleigh scattering). In this interaction, no energy is converted into kinetic energy, and ionization does not occur. The process is identical to what occurs in the transmitter of a radio station. The electromagnetic wave with an oscillating electric field sets the electrons in the atom into momentary vibration. These oscillating electrons emit radiation of the same wavelength. Because the scattering is a cooperative phenomenon, it is called coherent scatter.

Coherent scattering occurs mainly in the forward direction to produce a slightly broadened x-ray beam. Because no energy is transferred to kinetic energy, this process historically has limited interest to CT. Recent research, however, has shown that coherent scatter can be used for bone characterization [Westmore, 1997] and [Batchelar, 2000].

The net effect of these interactions is that some photons are absorbed, some are scattered, and some go through. In simple terms, x-ray photons are attenuated as they pass through matter. The attenuation can be expressed by an exponential relationship for a monochromatic incident x-ray beam and a material of uniform density and atomic number as:

$$I = I_0 e^{-\mu L}, \quad (3.24)$$

where I and I_0 are the incident and transmitted x-ray intensities, L is the material thickness, and μ is the linear attenuation coefficient of that material. This is often referred to as the Lambert-Beers law.

3.2.3. Radiation Dose

Ionizing radiation can cause tissue damage in a number of ways. The largest risk is that of cancer arising from the genetic mutations caused by chromosomal aberration [Webb, 2003]. The effects of radiation can be both deterministic and stochastic. Deterministic effects, caused by high doses of radiation, are usually associated with cell death. These effects are characterized by a dose threshold, below which no cell death is

observed. In contrast, stochastic effects occur at low radiation doses and the amount of radiation is proportional to the probability of a cell mutation.

The absorbed dose D is equal to the radiation energy E absorbed per unit mass. D is usually measured in units of grays (Gy), where 1 Gy equals 1 J/kg. Many non-scientific publications measure the absorbed dose in rads, where 1 Gy is equal to 100 rads. In a medical environment, the patient dose is often specified in terms of the entrance skin dose, with typical values of 0.1 mGy for a chest radiograph and 1.5 mGy for an abdominal radiograph. Such measurements, however, give little indication of the risk to the patient. The most useful measure of radiation dose is the effective dose equivalent, H_E , which is the sum of the dose delivered to each organ weighted by the radiation sensitivity w of that organ with respect to cancer mutations:

$$H_E = \sum_i w_i H_i, \tag{3.25}$$

where i is the number of organs irradiated and H_i is the dose equivalent for each organ. The value of H is given by the absorbed dose D multiplied by the quality factor Q_F of the radiation. Table 3.2 lists quality factors for different types of radiation.

Table 3.2: Quality factors of various types of ionizing radiation.

Radiation Type	Quality Factor
x-rays	1
γ -rays	1
neutrons	10
α -particles	20

Both H and H_E are measured in sieverts (Sv). Older literature measures the dose equivalent and the effective dose equivalent in units of rems, where 1 Sv equals 100 rems. Typical values of the radiation sensitivity of different organs, used for the calculation of H_E , are given in Table 3.3.

In CT applications the radiation dose to the patient is calculated in a slightly different manner because the x-ray beam profile across each slice is not uniform and adjacent

slices receive some dose from one another. The United States Food and Drug Administration (FDA) defines the Computed Tomography Dose Index (CTDI) for a 14-slice exam to be:

$$CTDI = \frac{1}{T} \int_{-T}^{+T} D_z dz, \tag{3.26}$$

where D_z is the absorbed dose at position z and T is the thickness of each slice. Still, in terms of addressing patient risk the value of H_E is a better measure. Table 3.4 lists typical values of H_E associated with standard clinical exams. The limit in annual radiation dose under federal law in the United States is 0.05Sv (5000 mrem) and corresponds to over 1000 planar chest x-rays, 15 head CTs, or 5 full-body CTs

Table 3.3: Radiation sensitivity of various organs.

Organ	Radiation Sensitivity
Gonads	0.2
Lung	0.12
Breast	0.1
Stomach	0.12
Skin	0.01
Thyroid	0.05

Table 3.4: Effective Dose Equivalent H_E for Clinical X-ray and CT Exams

Clinical Exam	H_E (mSv)
Breast X-ray	0.05
Chest X-ray	0.03
Skull X-ray	0.15
Abdominal X-ray	1
Barium fluoroscopy	5
Head CT	3
Body CT	10

4. Image Processing

4.1. Image Reconstruction

In this section we will deal with the mathematical basis of computed tomography with non-diffracting sources. We will show how one can go about recovering an image of the cross section of an object from projection data.

This chapter will start with the definition of line integrals and how they are combined to form projections of an object. By finding the Fourier transform of a projection taken along parallel lines, we will then derive the Fourier Slice Theorem. We will also discuss reconstruction algorithms based on parallel beam projection data and fan beam projection data.

4.1.1. Line Integral Measurements

In Chapter 2 we learned that computed tomography relies on x-ray flux measurements from different angles to form an image. At each angle, the measurement is essentially identical to that of the conventional x-ray radiography. We record the x-ray flux impinging on the x-ray detector after attenuation by an object. If we assume that the input x-ray photons are monoenergetic, the x-ray intensities measured on the entrance and the exit slides of a uniform material, shown in Figure 4.1 (a), follow the Lambert-Beers law [Herman, 1980], as discussed in Section 3.2.2.

$$I = I_0 e^{-\mu \Delta x}, \quad (4.1)$$

In this equation, I_0 is the entrance x-ray intensity, I is the exit x-ray intensity, Δx is the thickness, and μ is the linear attenuation coefficient of the material. In general, μ changes with the x-ray energy and varies with the type of material.

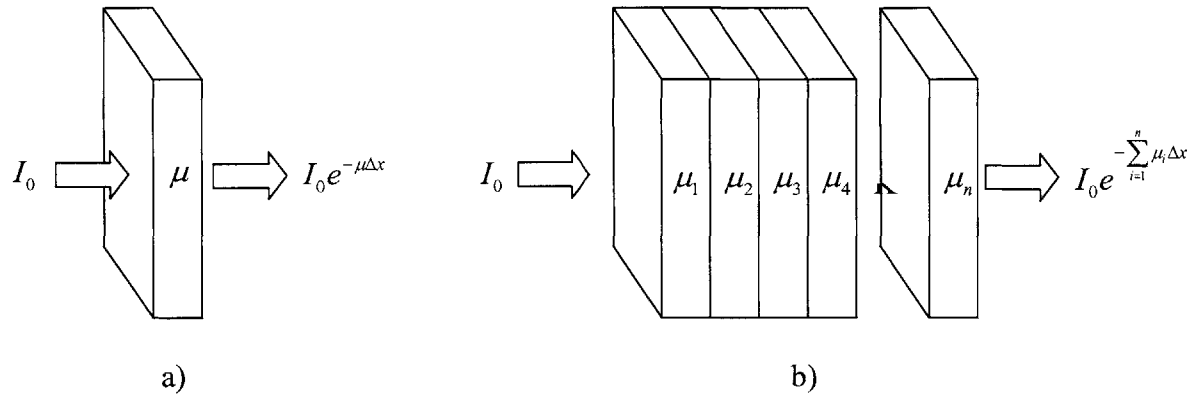


Figure 4.1: Monochromatic x-ray attenuation.
 a) Attenuation by an uniform object
 b) Attenuation by a non-uniform object

From Equation 4.1, it is clear that objects with higher μ values are more attenuating to x-ray photons than objects with lower μ values. For example, the μ value of bones is higher than that of soft tissues, indicating the fact that it is more difficult for x-ray photons to penetrate bones than soft tissues. On the other hand, the μ value for air is nearly zero, indicating the fact that the input and output x-ray flux is virtually unchanged by passage through air.

Let us now consider the case of a non-uniform object. The overall attenuation characteristics can be calculated by dividing the object into small elements, as shown in Figure 4.1b. When the size of the elements is sufficiently small, each element can be considered as a uniform object. Equation 4.1 can be used to describe the entrance and exit x-ray intensities for each element. Observing the fact that the exit x-ray flux from an element is the entrance x-ray flux to its neighbor, Equation 4.1 can be applied repeatedly in cascade fashion. Mathematically, the exiting intensity can be expressed as:

$$I = I_0 e^{-\mu_1 \Delta x} e^{-\mu_2 \Delta x} e^{-\mu_3 \Delta x} \dots e^{-\mu_n \Delta x} = I_0 e^{-\sum_{i=1}^n \mu_i \Delta x} \quad (4.2)$$

If we divide both sides of Equation 3.2 by I_0 and take the negative logarithm of the quantity, we obtain:

$$p = -\ln\left(\frac{I}{I_0}\right) = \sum_{i=1}^n \mu_i \Delta x. \quad (4.3)$$

As Δx approaches zero, the summation changes into an integral over the entire object and Equation 4.3 becomes:

$$p = -\ln\left(\frac{I}{I_0}\right) = \int_L \mu(x) dx. \quad (4.4)$$

In CT scanner applications, p is the projection measurement. Equation 4.4 states that the ratio of the input x-ray intensity over the output x-ray intensity, after logarithm operation, represents the line integral of the attenuation coefficient along the x-ray path. The reconstruction problem for CT can now be stated as the following: given the measured line integrals of an object, how do we estimate or calculate its attenuation coefficients?

4.1.2. Algebraic Reconstruction Technique

To understand some of the methods employed in the CT reconstruction, let us begin with an extremely simplified case in which the object is formed with four small blocks. The attenuation coefficients are homogeneous within each block and are labeled μ_1 , μ_2 , μ_3 , and μ_4 , as shown in Figure 4.2. Let us further consider the scenario where line integrals are measured along the horizontal, vertical, and diagonal directions. Five measurements in total are selected in this example. It can be shown that the diagonal and three other measurements form a set of independent equations. For example

$$\begin{cases} p_1 = \mu_1 + \mu_2 \\ p_2 = \mu_3 + \mu_4 \\ p_3 = \mu_1 + \mu_3 \\ p_4 = \mu_1 + \mu_4 \end{cases}, \quad (4.5)$$

Four independent equations are established for the four unknowns. From elementary algebra we know that there is a unique solution to the problem, as long as the number of equations equals the number of unknowns. If we generalize this to the case where the object is divided into N by N small elements, we could easily reach the conclusion that as

long as enough independent measurements (N^2) are taken, we can always uniquely solve the attenuation coefficient distribution of the object.

Many techniques are readily available for solving linear sets of equations. One method that comes to mind is the direct matrix inversion. Indeed, this was the method used on the very first CT scanner back in 1967 for which over 28,000 simultaneous equations were solved [Herman, 1979]. It is not difficult to see that when the object is divided into finer and finer elements, corresponding to higher spatial resolutions, the task of solving a set of simultaneous equations becomes quite a challenge, even with modern computers. In addition, to ensure that enough independent equations are formed, we often need to take more than N^2 measurements, because some of the measurements may not be independent.

A good example is the one given in Figure 4.2. Let us assume that four measurements, $p_1, p_2, p_3,$ and $p_5,$ are taken in the horizontal and vertical directions. It can be shown that these measurements are not linearly independent ($p_5 = p_1 + p_2 - p_3$). A diagonal value has to be added to ensure their orthogonality. When the number of equations exceeds the number of unknowns, a straightforward solution may not always be available. This is even more problematic when we consider the inevitable possibility that errors exist in some of the measurements. Therefore, different reconstruction techniques have to be explored. Despite its limited usefulness, the linear algebraic approach proves the existence of a mathematical solution to the CT problem.

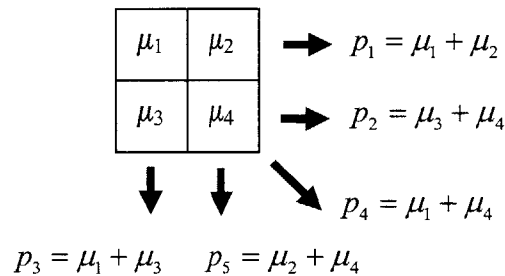


Figure 4.2: A simple example of an object and its projections.

One of the possible remedies for the problem is the so-called iterative reconstruction approach. For ease of illustration, we again start with an oversimplified example. Let us consider the four-block object problem discussed previously. This time, we assign

specific attenuation values to each block, as shown in Figure 4.3 (a). The corresponding projection measurements are depicted in the same figure.

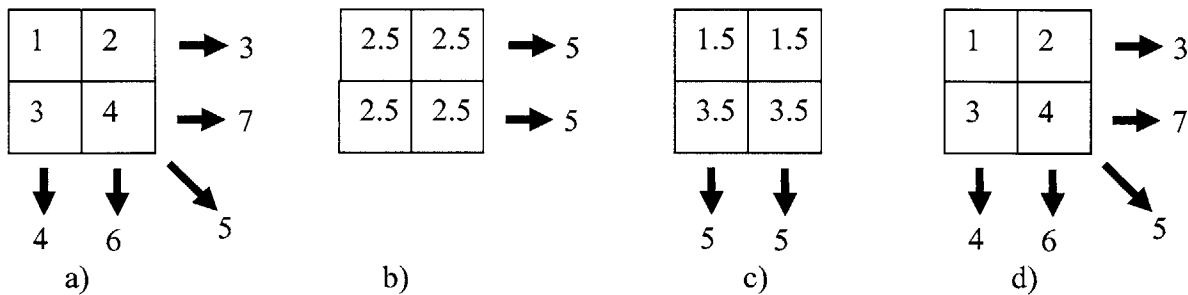


Figure 4.3: Illustration of algebraic reconstruction technique (ART).
a) Original object and measured projections,
b) Initial estimate of the object and estimated horizontal projections,
c) Updated object estimations and estimated vertical projections,
d) Final estimations and projections.

We will start with an initial guess of the object’s attenuation distribution. Because we have no a priori knowledge of the object itself, we assume it is homogeneous. We start, for example, with an initial estimate using the average of the projection samples. The sum of the projection samples ($3 + 7 = 10$ or $4 + 6 = 10$) evenly distributed over the four blocks results in an average value of 2.5. Next, we calculate the line integrals of our estimated distribution along the paths used for the original projection measurement. For example, we calculate the projection samples along the horizontal direction and obtain the calculated projection values of 5, as shown in Figure 4.3 (b).

Comparing the calculated projections against the measured values of 3 and 7, we observe that the top row is overestimated by 2 and the bottom row is underestimated by 2. Because we have no a priori knowledge of the object, we assume again that the difference between the measured and calculated projection needs to be split evenly among all pixels along each ray path. Therefore, we decrease the value of each block in the top row by 1 and increase the bottom row by 1, as shown in Figure 4.3 (c).

The calculated projections in the horizontal direction are now consistent with the measured projections. We repeat the same process for projections in the vertical direction and reach the conclusion that each element in the first column must be decreased by 0.5 and each element in the second column must be increased by 0.5 as shown in Figure 4.3 (d).

The calculated projections in all directions are now consistent with the measured projections (including the diagonal measurement), and the reconstruction process stops. The object is correctly reconstructed. This reconstruction process is called the algebraic reconstruction technique (ART) [Kak, 1986].

4.1.3. Radon Transform

The theory of image reconstruction from projection dates back to the early 1900s. In a paper published in 1917 Radon provides a mathematical proof that an object can be reconstructed from its infinite continuous projections [Radon, 1917 & Helgason 1999]. His method, known as Radon transform, was used in the first clinical CT scanner constructed by Hounsfield [Hsieh, 2002].

The Radon transform defines projections of an object mapping the spatial domain of the object to its projection space. For example, the two-dimensional Radon transform of a 2-D object defines 1-D line integrals in the projection space, which is also called Radon domain.

Let us define a two-dimensional object function $f(x, y)$ and its Radon transform by $R\{f(x, y)\}$. The Radon transform is defined by the projection function $P(p, \theta)$ in a polar coordinate system as

$$P(p, \theta) = R\{f(x, y)\} = \int_L f(x, y) dl, \quad (4.6)$$

where the line integral is defined along the path L such that,

$$x \cos \theta + y \sin \theta = p. \quad (4.7)$$

Figure 4.4 Shows a line integral that is computed along the parallel arrow lines, sampled along the p axis and defined by the angle θ . A set of line integrals and projections can be obtained for each angle θ .

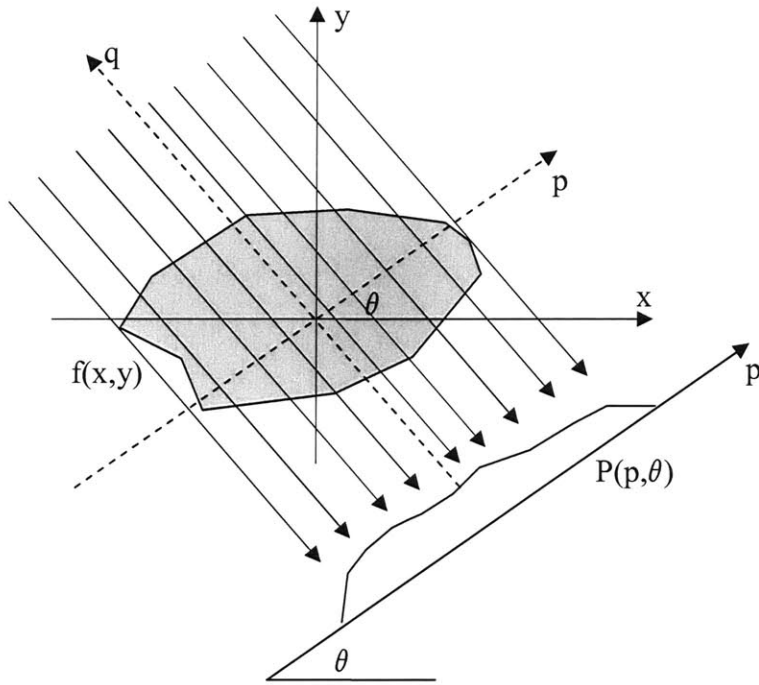


Figure 4.4: Line integral projection $P(p, \theta)$ of the 2-D Radon transform.

The polar coordinate system (p, θ) can be converted into rectangular in the Radon domain using a rotated coordinate system (p, q) defined as

$$\begin{aligned} p &= x \cos \theta + y \sin \theta \\ q &= -x \sin \theta + y \cos \theta \end{aligned} \quad (4.8)$$

The above implies that

$$R\{f(x, y)\} = \int_{-\infty}^{\infty} f(p \cos \theta - q \sin \theta, p \sin \theta + q \cos \theta) dq. \quad (4.9)$$

Higher-dimension Radon transform can be defined in a similar way [Helgason, 1999]. For example, the projection space for a 3-D Radon transform would be defined by 2-D planes instead of lines.

4.1.4. The Fourier Slice Theorem

After the illustration of an intuitive and a theoretical reconstruction technique, we will present the theory behind the tomographic reconstruction process used in most

commercial CT scanners today [Kak, 1986]. This reconstruction process is based on the Fourier Slice Theorem.

The Fourier Slice Theorem is derived by taking the one-dimensional Fourier transform of a parallel projection and noting that it is equal to a slice of the two-dimensional Fourier transform of the original object. It follows that given the projection data we should be able to estimate the object by simply performing a two-dimensional inverse Fourier transform.

The definition of a two-dimensional Fourier transform of an object, described by the function $f(x, y)$, is

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy. \quad (4.10)$$

Likewise, the Fourier transform of a projection at an angle θ , described by the function $P_{\theta}(x)$, is

$$S_{\theta}(u) = \int_{-\infty}^{\infty} P_{\theta}(x) e^{-j2\pi ux} dt. \quad (4.11)$$

The simplest example of the Fourier Slice Theorem is given for a projection at $\theta = 0$. First, consider the Fourier transform of the object along the line in the frequency domain given by $v = 0$. The Fourier transform integral now simplifies to

$$F(u, 0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi ux} dx dy, \quad (4.12)$$

but because the phase is no longer dependent on y we can split the integral into two parts

$$F(u, 0) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(x, y) dy \right] e^{-j2\pi ux} dx. \quad (4.13)$$

From the definition of a parallel projection, we recognize the term in brackets as the equation for a projection along lines of constant x

$$P_{\theta=0}(x) = \int_{-\infty}^{\infty} f(x, y) dy. \quad (4.14)$$

Substituting Equation 4.14 into 4.13 we discover that

$$F(u, 0) = \int_{-\infty}^{\infty} P_{\theta=0}(x) e^{-j2\pi ux} dx. \quad (4.15)$$

The right-hand side of this equation represents the one-dimensional Fourier transform of the projection $P_{\theta=0}$; thus we have the following relationship between the vertical projection and the 2-D transform of the object function:

$$F(u, 0) = S_{\theta=0}(u). \quad (4.16)$$

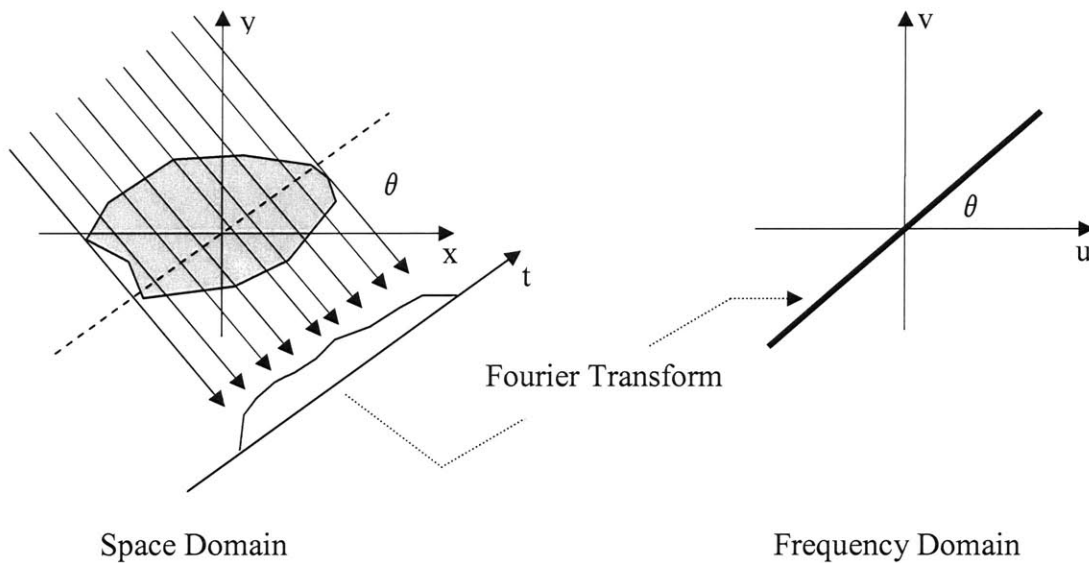


Figure 4.5: The Fourier Slice theorem relates the Fourier transform of a projection to the Fourier transform of the object along a radial line.

This is the simplest form of the Fourier Slice Theorem. Clearly this result is independent of the orientation between the object and the coordinate system. If, for example, as shown in Figure 4.5 the (x, y) coordinate system is rotated by an angle θ , the Fourier transform of the projection defined in Equation 4.15 is equal to the two-dimensional Fourier transform of the object along a line rotated by θ . This leads to the Fourier Slice Theorem, stated as:

“The Fourier transform of a parallel projection of an image $f(x, y)$ taken at angle θ gives a slice of the two-dimensional transform $F(u, v)$, subtending an angle θ with the u -axis [Kak, 1986].”

The derivation of the Fourier Slice Theorem can be placed on a more solid foundation by considering the (t, s) coordinate system to be a rotated version of the original (x, y) system as expressed by

$$\begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (4.17)$$

In the (t, s) coordinate system a projection along lines of constant t is written as

$$P_\theta(t) = \int_{-\infty}^{\infty} f(t, s) ds, \quad (4.18)$$

and its Fourier transform is given by

$$S_\theta(\omega) = \int_{-\infty}^{\infty} P_\theta(t) e^{-j2\pi\omega t} dt. \quad (4.19)$$

Substituting the definition of a projection into the above equation we find

$$S_\theta(\omega) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(t, s) ds \right] e^{-j2\pi\omega t} dt. \quad (4.20)$$

This result can be transformed into the (x, y) coordinate system by using the relationships in Equation 4.17

$$S_\theta(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi\omega(x\cos\theta + y\sin\theta)} dx dy. \quad (4.21)$$

The right-hand side of this equation now represents the two-dimensional Fourier transform at a spatial frequency of $(u = \omega \cos \theta, v = \omega \sin \theta)$

$$S_{\theta}(\omega) = F(\omega, \theta) = F(\omega \cos \theta, \omega \sin \theta). \quad (4.22)$$

Equation 4.22 is the formal mathematical proof of the Fourier Slice Theorem and it indicates that by taking the projections on an object function at angles $\theta_1, \theta_2, \dots, \theta_k$ and Fourier transforming each of these projections, we can determine the values of $F(u, v)$ on radial lines as illustrated in Figure 4.6. If an infinite number of projections are taken, then $F(u, v)$ would be known at all points in the (u, v) plane. Knowing $F(u, v)$, the object function $f(x, y)$ can be recovered using the inverse Fourier transform:

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv. \quad (4.23)$$

In practice, only a finite number of projections of an object can be taken. In that case it is clear that the function $F(u, v)$ is only known along a finite number of radial lines, as shown in Figure 4.6. In order to be able to use Equation 4.23 one must then interpolate from these radial points to the points on a square grid. Theoretically, one can exactly determine the N^2 coefficients required, provided as many values of the function $F(u, v)$ are known on some radial lines. This calculation involves solving a large set of simultaneous equations, often leading to unstable solutions. It is more common to determine the values on the square grid by interpolation from the radial points. Since the density of the radial points becomes sparser as one gets farther away from the center, the interpolation error also becomes larger. This implies there is greater error in the calculation of the high frequency components in an image than in the low frequency ones, which results in image degradation [Oppenheim, 1999].

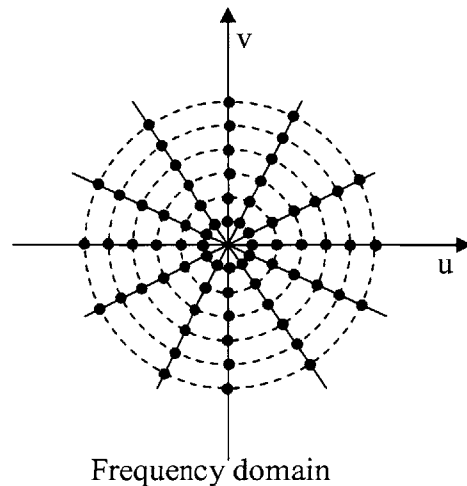


Figure 4.6: Sampling pattern in frequency domain based on Fourier slice theorem.

4.1.5. Fan Beam Reconstruction

So far, our discussion has been limited to the case of parallel-beam geometry. In nearly all of the commercially available CT scanners, the x-ray source is approximately a single point and a projection is collected by sampling nearly 1000 detector channels in a short time window [Hsieh, 2003]. Consequently, the Fourier Slice Theorem cannot be applied directly.

To solve the reconstruction issue, we need to take a look at the fan beam geometry. In practice, two of the most popular detector design choices are equiangular fan beam and equal-spaced fan beam. In an equiangular fan beam, the detector modules are placed on an arc that is concentric to the x-ray source, as shown in Figure 4.7(a). In an equal-spaced fan beam, the detector is formed on a flat surface instead of a curved surface. The sampling is uniformly spaced, as shown in Figure 4.7(b).

We discussed several algorithms for parallel-beam geometry reconstruction. If we can re-format the fan beam projections into a set of parallel-beam projections, the previously discussed reconstruction algorithms can be applied directly. The easiest way to do this is with a process, called re-binning.

The idea behind re-binning is fairly simple. Let us start by examining the sampling pattern of a set of fan beam projections. Figure 4.8 depicts a two-dimensional graph with its horizontal axis representing the distance of a ray to the iso-center and the vertical axis the angle between the ray and the x axis.

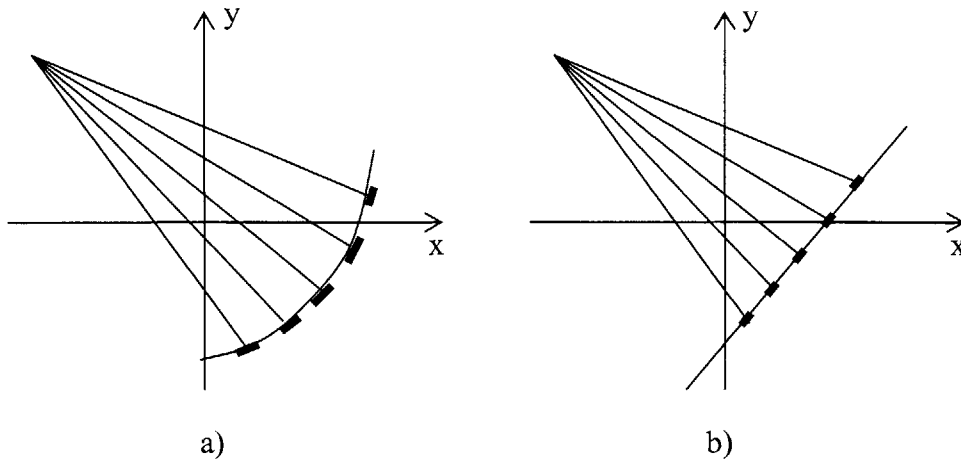


Figure 4.7: Different fan beam geometries.
 a) Equiangular fan beam,
 b) Equal-spaced fan beam.

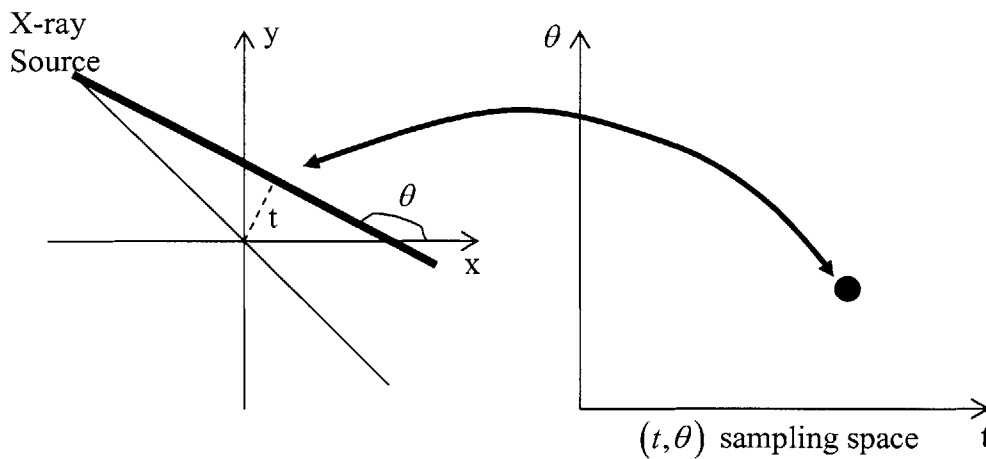


Figure 4.8: Fan beam projection pattern.

A projection ray is mapped to a single point in the (t, θ) space. For parallel projections, all samples fall onto a uniformly spaced rectangular grid, as shown in Figure 4.9 (a). Each parallel projection is represented by a single row of dots. If we map a set of fan beam samples to the same space, they in general, do not fall onto the rectangular grid, as shown in Figure 4.9 (b). Each fan beam projection is therefore mapped to a slanted row of dots indicated. This can be understood easily from the fact that the angles formed by the fan beam rays with the x axis increase (or decrease) from one end of the detector to

the other in a single projection. Similarly, the distance from the fan beam rays to the iso-center does not increase (or decrease) in a uniform step. For the equiangular fan beam sampling, the relationship between the angle, θ , and the sampling index, i , is linear,

$$\theta = (i - i_{iso}) \Delta\gamma, \quad (4.24)$$

where i_{iso} is the index for the iso-channel and γ is the angle formed by two adjacent detector channels and the x-ray source. The distance, d_i , of fan beam ray i to the iso-center is

$$d_i = r \times \sin \theta = r \times \sin (i - i_{iso}) \Delta\gamma, \quad (4.25)$$

where r is the source-to-iso-center distance. Figure 4.9(b), combined with Equation 4.23, and Equation 4.25 indicate that the fan beam samples can be mapped to the parallel-beam sampling grid by interpolation. Most modern image processing software packages, like MatLab, offer built-in re-binning functions.

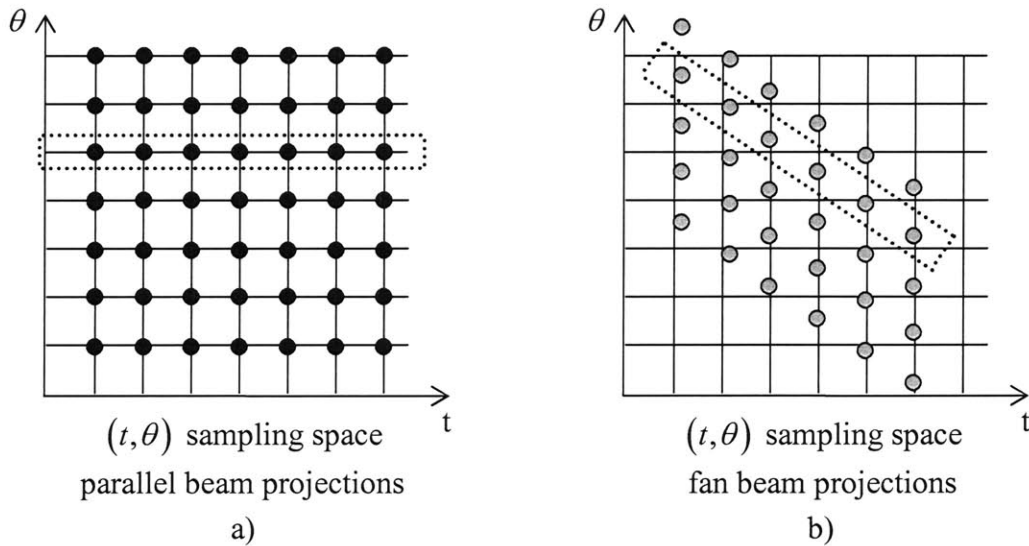


Figure 4.9: Projection patterns for different beam geometries.
a) Parallel beam projection patter
b) Fan beam projection pattern

4.2. Image Enhancement

Medical images acquired in most radiological applications are visually examined by physicians. The purpose of image enhancement is to process an acquired image for better contrast and visibility of features of interest for visual examination as well as subsequent computer-aided analysis and diagnosis.

There is no unique general theory or method for processing all kinds of medical images for feature enhancement. For example, a chest X-ray radiographic image shows the anatomical structure of the chest based on the total attenuation coefficients. If the radiograph is being examined for a possible fracture in the ribs, the image enhancement method is required to improve the visibility of hard bony structure. But if an X-ray mammogram is obtained for examination of potential breast cancer, an image-processing method is required to enhance visibility of microcalcifications, speculated masses and soft-tissue structures such as parenchyma [Dhawan, 2003]. A single image-enhancement method does not serve both of these applications.

Image-enhancement methods may also include image-restoration methods, which are generally based on minimum mean-squared error operations, such as Wiener filtering and other constrained deconvolution methods incorporating some a priori knowledge of image quality degradation [Bose, 2004].

Image-enhancement techniques can be separated in two categories: spatial domain and frequency domain techniques. Spatial domain techniques manipulate image pixel values in the spatial domain based on the distribution statistics of the entire image or local sub-regions. Typical spatial domain techniques include histogram transformation, spatial filtering, region-growing, morphological image processing, and model-based image estimation. Frequency domain techniques manipulate information in the frequency domain based on the frequency characteristics of the image and typically involve frequency filtering, homomorphic filtering and wavelet transformations.

In the rest of this section we will look at some of the most popular image enhancement techniques in both spatial and frequency domain.

4.2.1. Histogram Transformations

A histogram of an image provides information about the intensity distribution of pixels in the image. The simplest form of a histogram is the plot of occurrence of specific gray-level values of the pixels in the image. For example, there are 256 gray levels ranging from 0 to 255 in an 8-bit gray-scale image. The occurrence of gray levels can be provided in terms of the number of times a specific gray-level has occurred in the image. In mathematical terms, a histogram $h(g_i)$ is expressed as

$$h(g_i) = n_i, \forall i \in [1..2^N], \quad (4.26)$$

where g_i is the i^{th} gray level, N is the bit order of the image, and n_i is the number of occurrences of grey level g_i in the image.

A histogram may also be expressed as the probability of occurrence of a specific gray level in the image. Mathematically it will be written as:

$$p(g_i) = \frac{n_i}{n}, \quad (4.27)$$

where n is the total number of pixels in the image. The term “histogram” refers to a plot of either $h(g_i)$ or $p(g_i)$ versus r_i . Figure 4.10 (a) shows a CT image of human brain and its histogram.

A popular general-purpose image enhancement technique is histogram equalization [Gonzalez, 2002]. In this method, a monotonically increasing transformation function, $T(g)$ is used to map the original gray values $g(i)$ to new gray values, s_i , such that

$$s_i = T(g_i) = \sum_{j=0}^i p_g(g_j) = \sum_{j=0}^i \frac{n_j}{n}, \quad (4.28)$$

where $p_g(g_i)$ is the probability based histogram of the input image that is transformed into the output image with histogram $p_s(s_i)$.

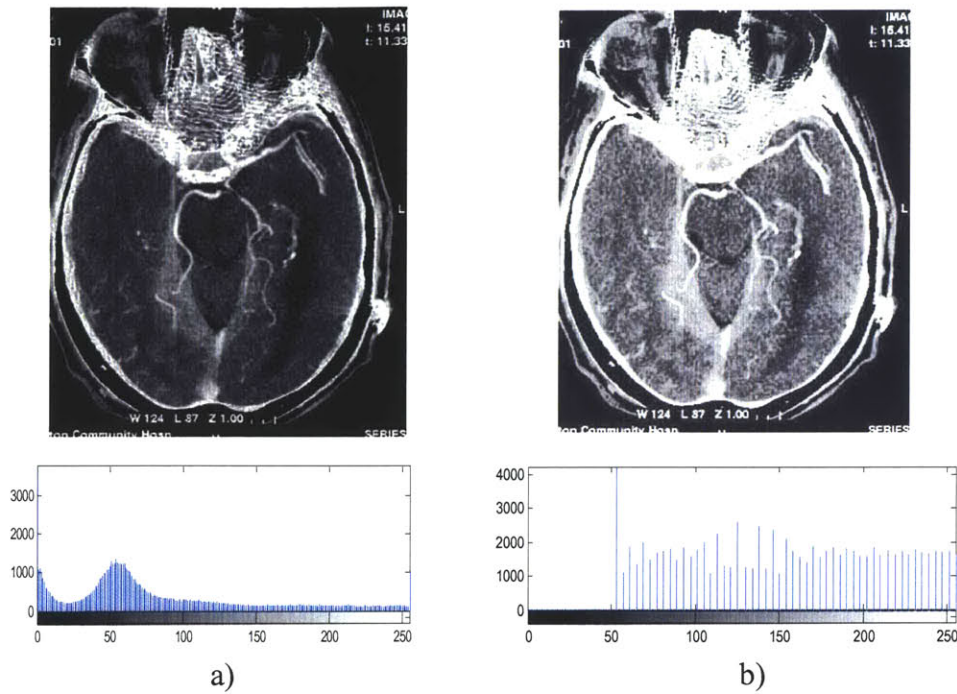


Figure 4.10: Sample brain CT image with histogram equalization.
a) Original image and its histogram,
b) Enhanced image after histogram equalization and its histogram.
 [Image taken from Boca Radiology Group, 2006]

The transformation function $T(g)$ stretches the histogram of the input image such that the gray values occur in the output image with equal probability. It should be noted that the uniform distribution of the histogram of the output image is limited by discrete computation of the gray-level transformation. Figure 4.10 (b) shows the enhanced version of the brain image shown in Figure 4.10 (a) after histogram equalization. The equalized histogram of the enhanced images is shown as well.

The histogram equalization may not provide good results in many applications. Sometimes it can cause saturation in some regions of the image resulting in loss of details and high frequency information that may be necessary for interpretation. Sometimes, local histogram equalization is applied separately on predefined local neighborhood regions, such as 16×16 pixels, to provide better results [Webb, 2003].

If a desired distribution of gray values is known a priori, a histogram modification method is used to apply a transformation that changes the gray values to match the desired distribution. The target distribution can be obtained from a good contrast image that is obtained under similar imaging conditions. Alternatively, an original image from a

scanner can be interactively modified through regional scaling of gray values to achieve the desired contrast. This image can now provide a target distribution to the rest of images, obtained under similar imaging conditions, for automatic enhancement using the histogram modification method.

The conventional scaling method of changing gray values from the range $[a..b]$ to $[c..d]$ can be expressed as a linear transformation:

$$z_{new} = \frac{d-c}{b-a}(z_{old} - a) + c, \quad (4.29)$$

where z_{new} and z_{old} are the new and original gray values of a pixel in the image.

Let us assume that $p_z(z_i)$ is the target histogram, and $p_g(g_i)$ and $p_s(s_i)$ are the histograms of the input and output image. A transformation is needed such that the output image would have the desired histogram $p_z(z_i)$. The first step in this process is to equalize $p_g(g_i)$ using Equation 4.28:

$$t_i = T(g_i) = \sum_{j=0}^i p_g(g_j). \quad (4.30)$$

A new transformation U is defined that will equalize the target histogram such that

$$u_i = U(z_i) = \sum_{j=0}^i p_z(z_j). \quad (4.31)$$

We can now use the inverse of U , U^{-1} to transform the input image histogram into the desired histogram:

$$s_i = U^{-1}(u_i) = U^{-1}(T(g_i)). \quad (4.32)$$

4.2.2. Image Averaging

Signal averaging is a well-known method for enhancing signal-to-noise ratio. In CT imaging, data from the detectors are often averaged over time or space for signal enhancement [Suetens, 2002]. However, such signal enhancement is achieved at the cost

of some loss of temporal or spatial resolution. Sequence images, if properly registered and acquired in non-dynamic applications, can be averaged for noise reduction, leading to smoothing effects. Selective weighted averaging can also be performed over specified groups of pixels in the image. Figure 4.11 (a) shows a head CT image with artificially added white noise, and Figure 4.11 (b) shows the same image after spatial image averaging.

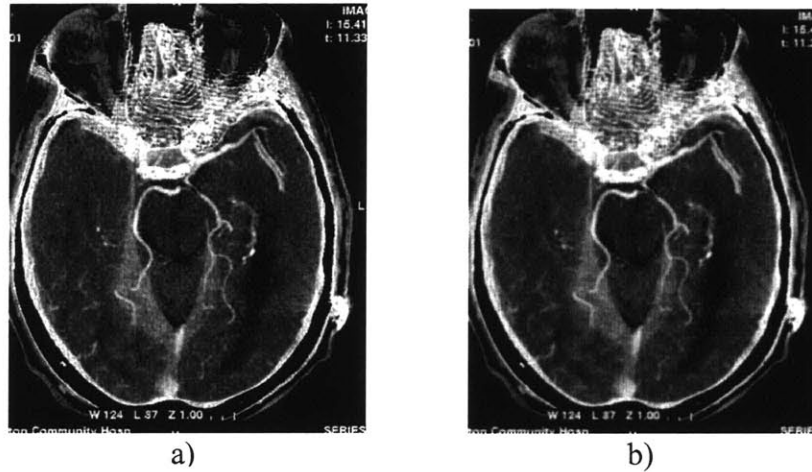


Figure 4.11: Sample brain CT image with spatial averaging.
a) Original image,
b) Enhanced image after spatial averaging.
 [Image taken from Boca Radiology Group, 2006]

Let us assume that an ideal image $f(x, y)$ is degraded by the presence of additive noise $e(x, y)$. The acquired image $f_a(x, y)$ can be expressed as

$$f_a(x, y) = f(x, y) + e(x, y). \quad (4.33)$$

In a general imaging application, the noise is assumed to be uncorrelated, random, identically distributed over the entire spectrum, and with a zero mean. If a sequence of M images is acquired for the same object under the same imaging conditions, the average image $\bar{f}_a(x, y)$ can be expressed as

$$\bar{f}_a(x, y) = \frac{1}{M} \sum_{i=1}^M f_{a,i}(x, y). \quad (4.34)$$

According to the Law of Large Numbers [Schinazi, 2001], as M increases the expected value of $\bar{f}_a(x, y)$ approaches the original image $f(x, y)$. Moreover, we can estimate the probability of an arbitrary small error b to be

$$P\left(\left|\bar{f}(x, y) - f(x, y)\right| \geq b\right) \leq \frac{\sigma^2}{b^2 M}, \quad (4.35)$$

where σ^2 is the variance of the noise.

4.2.3. Median Filtering

Median filter is a common order-statistics filter [Dhawan, 2003] that replaces the original gray value of a pixel by the median of the gray values of pixels in the specified neighborhood. For example, for a 3×3 pixels based neighborhood, shown in Figure 4.12, the gray value of the central pixel $f(0, 0)$ is replaced by the median of gray values of all 9 pixels in the neighborhood. Median filtering of $f(x, y)$ produces a smoothed image $f_m(x, y)$ and is defined as

$$f_m(x, y) = \underset{(i,j) \in N}{\text{median}} \{f(i, j)\}, \quad (4.36)$$

where N is the per-specified neighborhood of each pixel.

$f(-1,-1)$	$f(0,-1)$	$f(1,-1)$
$f(-1,0)$	$f(0,0)$	$f(1,0)$
$f(-1,1)$	$f(0,1)$	$f(1,1)$

Figure 4.12: 3 by 3 pixel neighborhood.

Figure 4.13 shows a CT brain image and the image, obtained by median filtering over the neighborhoods of 3×3 pixels. In certain applications, adaptive neighborhoods can be used for median filtering to preserve the edges better.

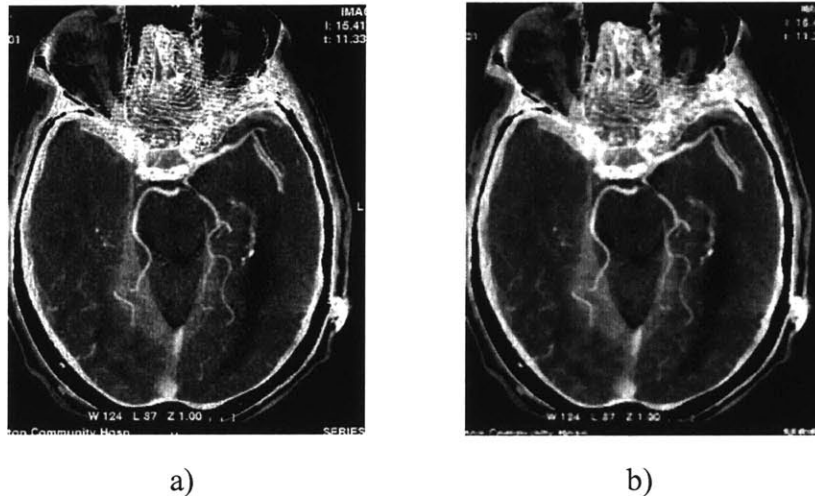


Figure 4.13: Sample brain CT image with median filtering.
a) Original image,
b) Enhanced image after median filtering.
[Image taken from Boca Radiology Group, 2006]

4.2.4. Image Sharpening and Edge Enhancement

Edges in an image are defined by the change in gray values of pixels in a neighborhood. The change of gray values of adjacent pixels in the image can be expressed by a derivative in continuous domain or a difference in discrete domain.

A first-order derivative operator, known as Sobel, computes the gradient information in a specific direction. The derivative operator can be encoded into a filter mask. Figure 4.14 shows two Sobel masks that are used, in computing the first-order gradient in x- and y-directions.

These filter masks are used in convolution to compute respective gradient images. For spatial image enhancement based on the first-order gradient information, the resultant gradient image can simply be added to the original image and rescaled using the full dynamic range of gray values.

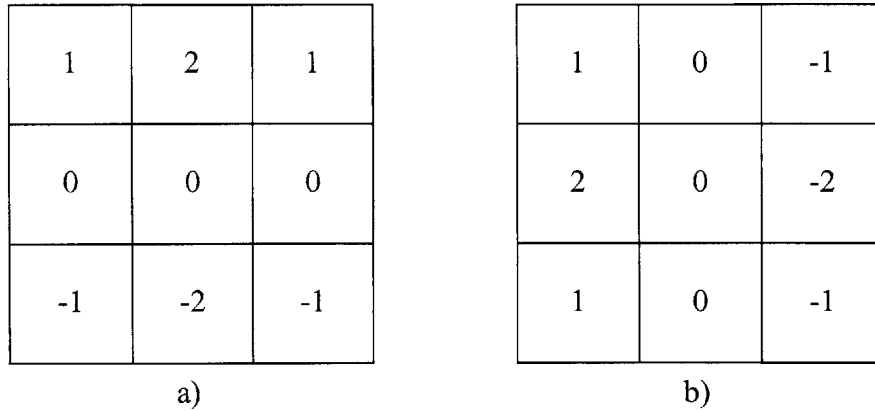


Figure 4.14: First-order gradient (Sobel filter) masks.
a) First-order gradient in x-direction,
b) First-order gradient in y-direction.

For more sophisticated edge enhancement, a second-order derivative operator, known as Laplacian is often used [Reinhard, 2004]. The Laplacian, ∇^2 , of an image $f(x, y)$ is defined as

$$\nabla^2 f(x, y) = \frac{\delta^2 f(x, y)}{\delta x^2} + \frac{\delta^2 f(x, y)}{\delta y^2}. \quad (4.36)$$

In the discrete domain the Laplacian of an image is defined as

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y). \quad (4.37)$$

A Laplacian mask with diagonal neighbors is shown in Figure 4.15 with the resultant CT brain image obtained by spatial filtering with this mask. Applying the Laplacian to the image is one of the simplest and powerful edge-based image enhancement techniques.

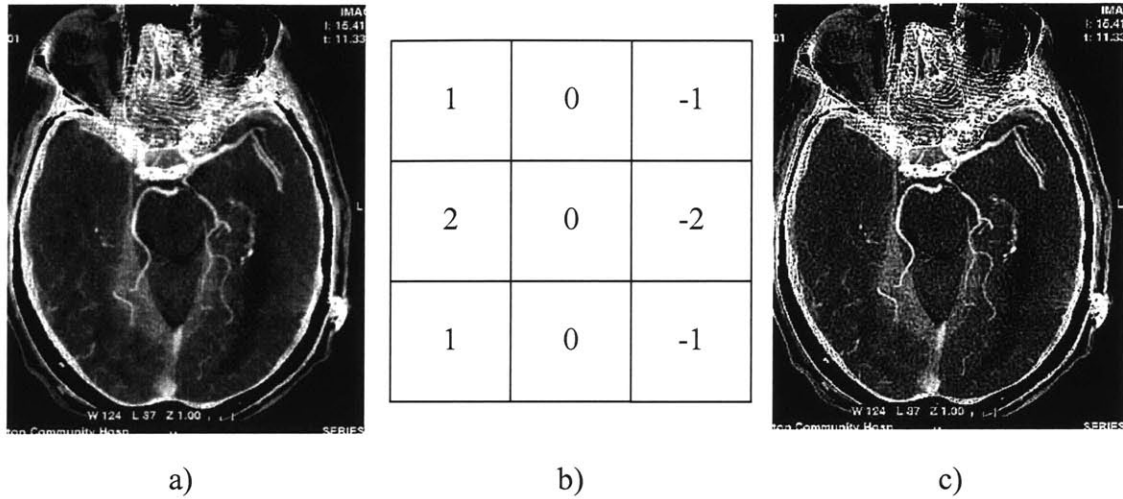


Figure 4.15: Sample brain CT image with Laplacian filtering.
a) Original image,
b) Laplacian filter mask
c) Enhanced image after Laplacian filtering.
 [Image taken from Boca Radiology Group, 2006]

4.3. Image Magnification

Unlike visible light microscope, magnification in the x-ray microscope is not achieved using a complex system of lenses. Since x-rays neither reflect, nor refract easily, conventional microscopic techniques are useless in an x-ray microscope.

Magnification in an x-ray microscope is achieved by using a cone-beam x-ray source and by altering the distance between the object, the source, and the imaging device, as shown in Figure 4.16. If an object is placed at the imaging device, its radiograph will correspond to its original size, since the x-ray beams would not have diverged after they penetrated the object, as shown in Figure 4.16 (a). If, however, we move the object halfway between the x-ray source and the imaging device, the radiograph will be larger, since the x-rays would have diverged after they exited the object, as shown in Figure 4.16 (b).

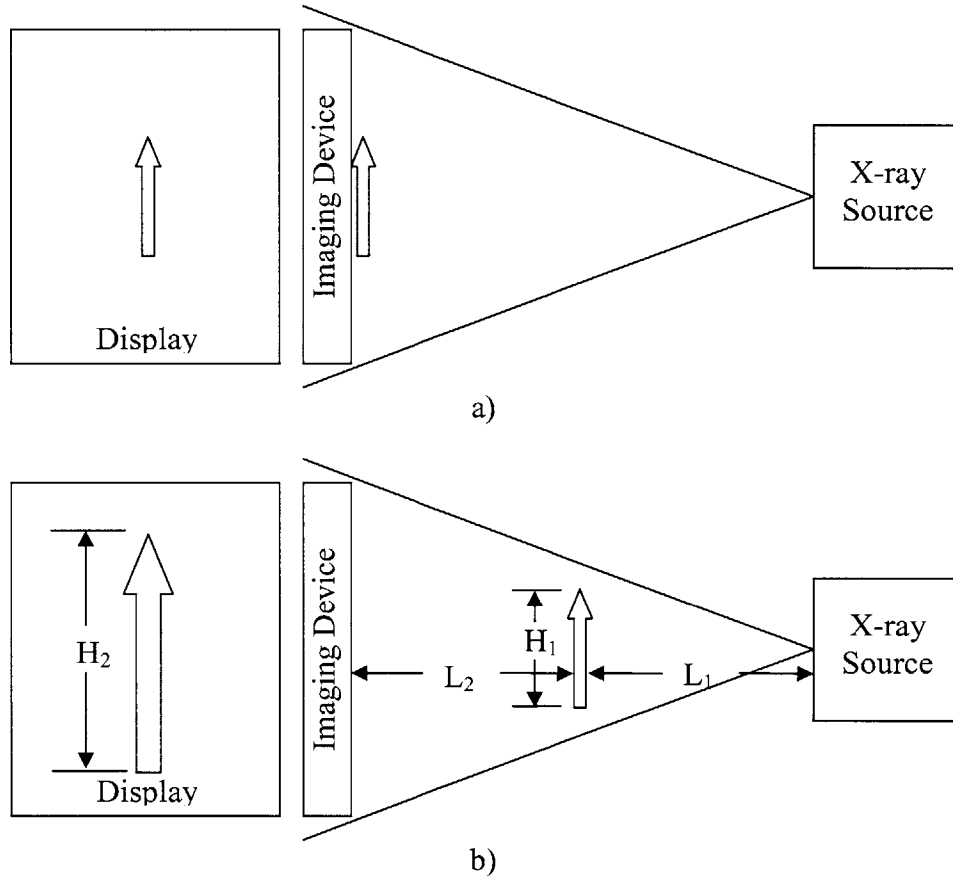


Figure 4.16: Magnification by using cone-beam x-ray source.
 a) Object placed at the imaging device,
 b) Object placed between the imaging device and the x-ray source.

Calculating the magnification ratio of any particular x-ray microscope configuration requires two parameters: the distance between the object and the x-ray source, L_1 , and the distance between the imaging device and the object, L_2 . Using basic geometric relationships in similar triangles, one can see that the height of the radiographic image, H_2 , relates to the height of the original object, H_1 , as:

$$\frac{H_2}{H_1} = \frac{L_2 + L_1}{L_1}. \quad (4.38)$$

The metric $\frac{H_2}{H_1}$ is often referred to as the “magnification” ratio, which depends only on the geometric configuration of the microscope. Theoretically, this ratio can range from

0 to infinity; however, in all practical application it is limited by the size of the object, the equipment, and by the diffraction/dispersion of the x-ray beam.

4.4. Image Visualization

The most important output from a CT scanner is the image itself. Traditionally, CT images have been represented as slices laying in planes, perpendicular to the imaging array, but recently with the advance in computer technology more and more CT scanner produce 3D volumetric representations of the experimental subjects. This section will discuss both visualization techniques.

4.4.1. Traditional CT Image Visualization

As discussed in Section 4.1, the reconstructed images in a CT scanner represent the linear attenuation coefficient (LAC) map of the scanned object. The actual intensity scale used in most CT applications is based on the Housfield unit (HU). The definition of HU is given as:

$$I = \frac{\mu - \mu_{water}}{\mu_{water}} \times 1000. \quad (4.39)$$

Water has attenuation coefficient of 0 HU, while air and vacuum are -1000 HU. Denser materials, like bone and metals, have attenuation coefficients in hundreds and thousands HU. Because of the large dynamic range of the LAC, it is impossible to adequately visualize it without modification on a standard 8-bit gray scale monitor or film.

If a CT image is displayed without transformation, the original dynamic range of well over 2000 HU must be compressed by an order of magnitude, as shown in Figure 4.17 (a). The minimum and maximum LAC of the image are -1000 HU and 1700 HU, respectively.

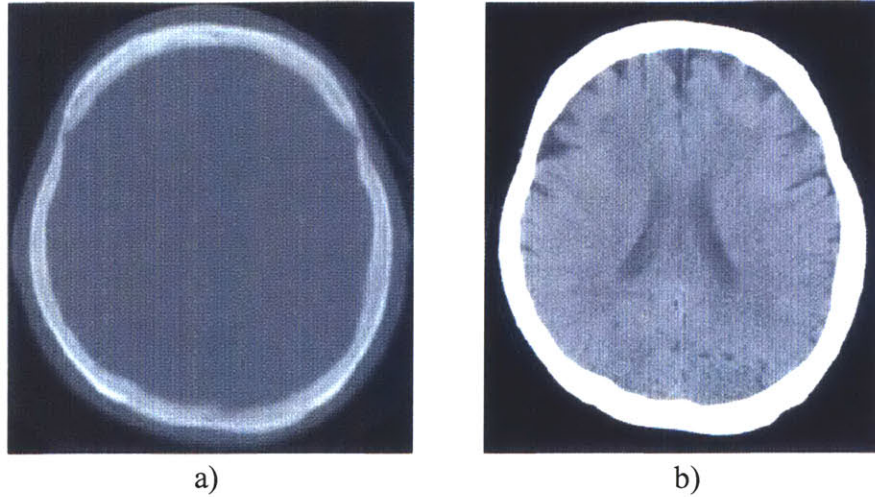


Figure 4.17: CT image of a human head.
 a) Image displayed with full intensity range (2700 HU),
 b) Image displayed with window width of 100 HU and window level of 20 HU.
 [Image taken from Hsieh, 2003]

When this dynamic range is linearly mapped to the dynamic range of the display device (0 to 255), the original gray scale is so severely compressed that little intensity variation can be visualized inside the human skull. This is clearly unacceptable and in order to overcome this difficulty, a CT image is typically displayed with a modified gray scale:

$$I_w(x, y) = \begin{cases} 0 & I(x, y) \leq L_w - \frac{W_w}{2} \\ \frac{I(x, y) - \left(L_w - \frac{W_w}{2}\right)}{W_w} & L_w - \frac{W_w}{2} < I(x, y) \leq L_w + \frac{W_w}{2} \\ I_{\max} & I(x, y) > L_w + \frac{W_w}{2} \end{cases}, \quad (4.40)$$

where L_w and W_w are the display window level and width and I_{\max} is the maximum intensity of the display or film. This transformation maps an intensity scale of interest in the interval $\left[L_w - \frac{W_w}{2}, L_w + \frac{W_w}{2}\right]$ to the full intensity scale of the visualization device.

4.4.2. Volumetric Visualization

Although the capability for generating 3D images by computer has been around nearly since the beginning of CT, utilization of this capability has become popular only recently. This gradual change is mainly due to three factors.

The first is related to the quality and efficiency of the 3D image generation. Because of the slow acquisition speed of the early CT scanners, thicker slices were typically acquired to cover the entire volume of an organ. The large mismatch between the in-plane and cross-plane resolution produced undesirable 3D image quality and artifacts. In addition, the amount of time needed to generate 3D images from a 2D dataset was quite long, because of limitations on computer hardware and lack of advanced and more efficient algorithms.

The second factor is the productivity of radiologists. Historically, CT scans have been organ-centric. Each CT exam covered only one organ, such as liver, lung, chest, or head, with either thick slices or sparsely placed thin slices. The number of images produced by a single exam was well below one hundred. With the recent advancement in CT multislice technology, a large portion of the human body can easily be covered with thin slices. For example, some of the CT angiographic applications typically cover a 120 cm range from the celiac artery to the foot with thin slices [Dhawan, 2003]. The number of images produced by a single exam can be from several hundred to over a thousand and to view these images sequentially would be an impossible task.

The third factor that influences the image presentation format is related to clinical applications. More and more CT images become useful tools for surgical planning, therapy treatments, and other applications outside the radiology department [Suetens, 2002]. It became more convenient and desirable to present these images in a format that can be understood by people without radiological training. The answer to all of these questions came from advanced computer graphical techniques [Reinhard, 2004].

One of the most popular volumetric visualization techniques is the multiplanar reformation (MPR). To generate an MPR image, the reconstructed "slice" images are arranged sequentially in a "stack," as shown in Fig. 4.18.

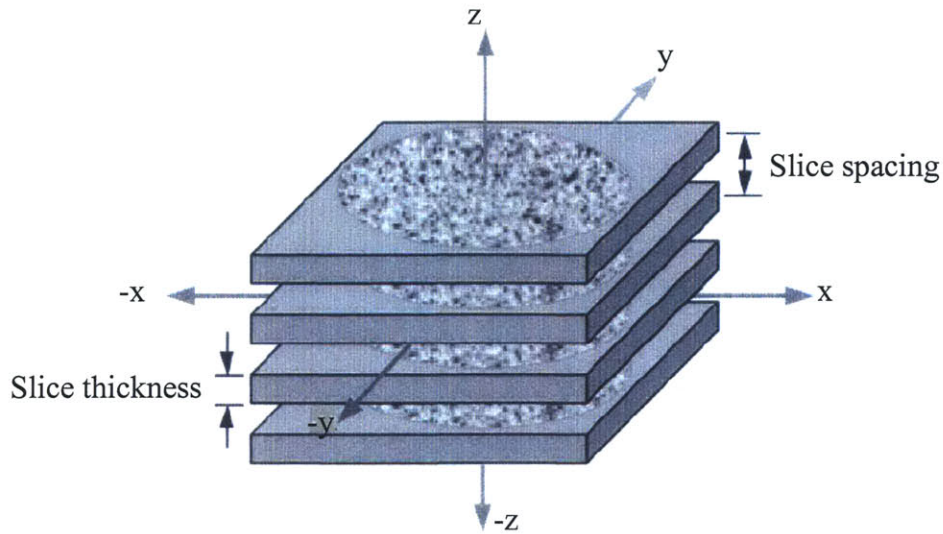


Figure 4.18: Reconstructed “slice” images sequentially arranged in a “stack.”
 [Image taken from Webb, 2003]

Each CT image has a certain slice thickness, and the distance between consecutive images represents the slice spacing. Reformatted images are produced by interpolating the image volume in different orientations. If there is no gap between images, the reformation process is somewhat simple. But even under this condition, an image pixel in the reformatted image often requires the interpolation of neighboring pixels, since it is unlikely that the reformatted pixel is located exactly at the original grid of the axial images. The interpolation can be carried out in one, two, or three dimensions. Figure 4.19 shows an example of an MPR image of a human lumbarosacral spine.



Figure 4.19: MPR image of a human lumbarosacral spine.
 [Image taken from Boca Radiology Group, 2006]

5. X-ray Microscope Design

The ultimate goal of our project was to create a high-resolution, high-speed 3D x-ray microscope which could be used in a populated laboratory environment. One of the main challenges in the design of the microscope was the potential harmful effect of the x-ray radiation on the laboratory staff and the eminent danger associated with overexposure. Precautionary measures were taken from the beginning of the project and all risks were carefully assessed throughout the instrument's design and implementation, leading to the construction of an environmentally safe instrument.

Due to the nature of the project it was clear that all functional components of the x-ray microscope had to be shielded from the environment in order to minimize the chance of an overexposure. The x-ray source, the imaging system, and the test sample had to be placed inside of a lead-lined enclosure, together with any additional instrumentation required to run the experiments. Furthermore, a data communication link had to be provided to and from the external power source and data processing CPU. Figure 5.1 shows a simplified schematic of the main microscope components with the required data and power links.

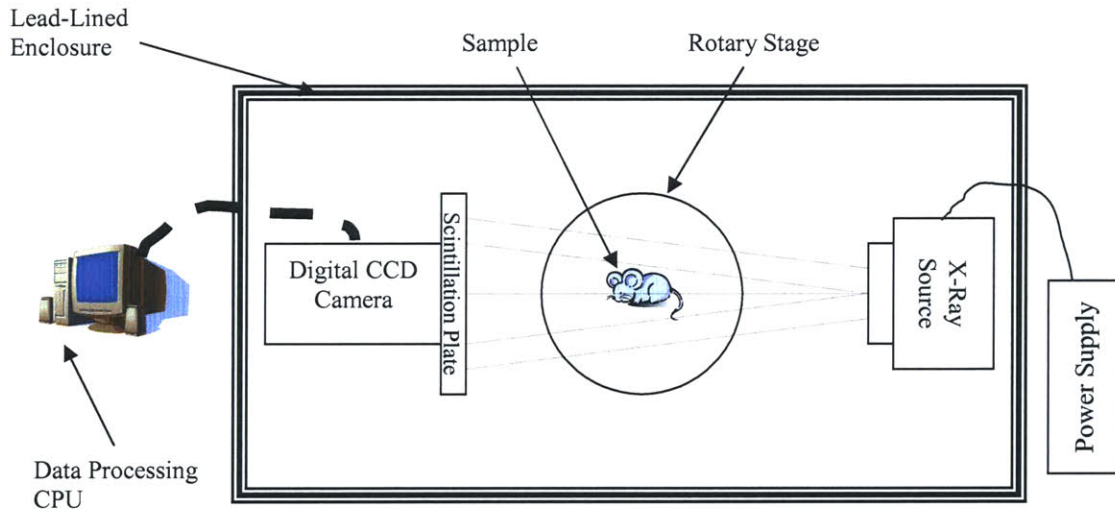


Figure 5.1: General system overview.

From design point of view the instrument was separated into five major components: functional, structural, mechanical, electrical, and software. Each component will be described and discussed separately in the later sections.

5.1. Functional Components

Based on the desired imaging capabilities of our microscope listed in Table 5.1, the first step of its design involved finding an appropriate imaging system. After series of product inquiries and performance analysis we decided to base our microscope off of a Hamamatsu C4742-56-12NR digital CCD camera. The camera was purchased together with its own power supply, image acquisition hardware, and control software; therefore the only missing functional components were an x-ray source and a data processing CPU. Details on the camera characteristics are provided in Section 6.1.1.

Table 5.1: System requirements.

Spatial Resolution	100 μm or less
Frame Rate	10 fps or more
Field of View	20 \times 20mm or more
Power Line	110 V
Sensitivity	10 bits or more

The x-ray source we chose for our instrument was a Source Ray SB-80-500. The source was purchased with an optional digital interface (RS-232) so that it could be controlled from a PC. The x-ray source was supplied to us without a power supply, so we had to design and build one and integrate it into the system. Details on the x-ray source are provided in Section 6.1.2.

The final functional component of the microscope was the data processing CPU. We decided to use a Dell XPS 600 PC, a high-performance computer system at the time of the instrument design. Details on the PC are provided in Section 6.1.3.

5.2. Structure

Safety and functionality were the two key factors considered in the design of the supporting structure of the x-ray microscope. According to Source Ray, Inc., the manufacturer of the x-ray source, the source had to be shielded by at least 2 mm thick lead sheet. For added extra safety we decided to triple that amount and to use two 3.12 mm thick lead sheets as shielding material. We also decided it would be a good idea to

have a visual port into the enclosure, which will remain transparent during experiments, allowing us to observe the operation of the instrument. The only suitable material was lead oxide glass with over 70% lead content. The minimum glass thickness required for our application was 3.5 mm, but we decided we would use 8mm for added safety.

The geometric constraints on the enclosure were imposed by the size of the instruments, which we wanted to fit inside it. We decided that the minimum height clearance inside the enclosure must be no less than 0.8 m and its footprint must be at least 1 m by 1 m. For convenience we decided to use a 1.8 m by 1.2 m optical table onto which to construct the enclosure. At this point we estimated the mass of the lead sheets required to cover the entire enclosure to be approximately 1500 kg. In order to minimize the overall mass of the instrument, extruded aluminum was chosen as structural material for the manufacturing of the enclosure's supports. The preliminary CAD model of the supporting structure, built entirely of 6061-T6 aluminum is shown in Figure 5.2.

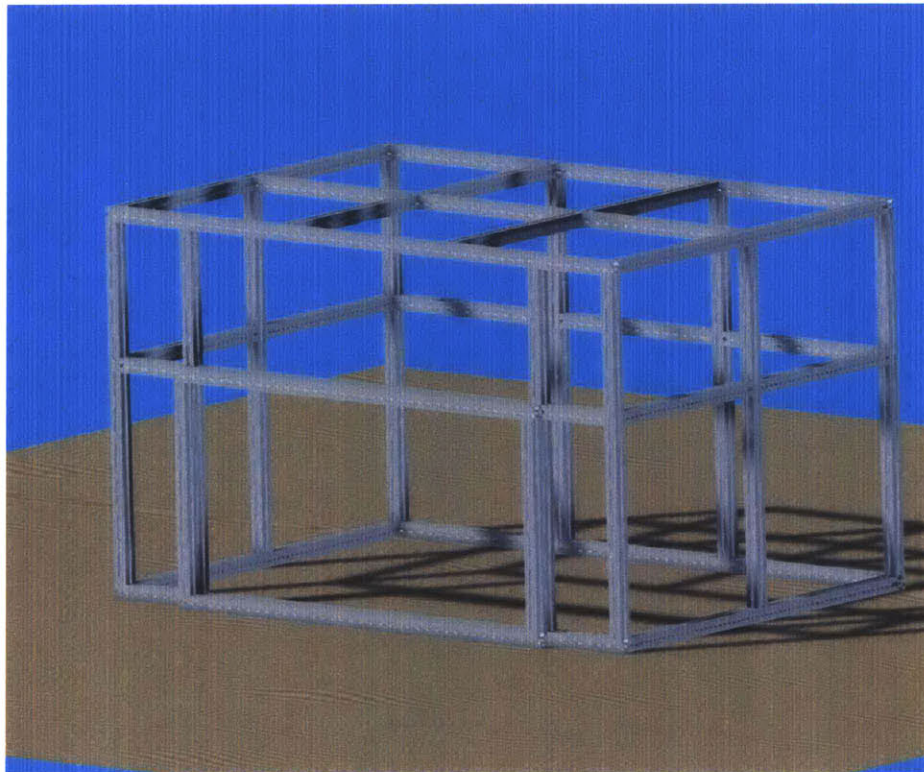


Figure 5.2: CAD model of aluminum support frame.

Basic beam-bending calculations showed that the thickness of the supporting beams had to exceed 25 mm if no deflection larger than 2 mm can occur due to the mass of the construction. Based on these analyses we decided to use 40 mm extrusions, guaranteeing deflections of less than 1mm. We also decided to space the aluminum support beam 450 mm apart on the top of the structure to prevent the lead sheet from sagging, creeping, and forming microscopic tears. The completed CAD model of the structural frame of the microscope is shown in Figure 5.3.

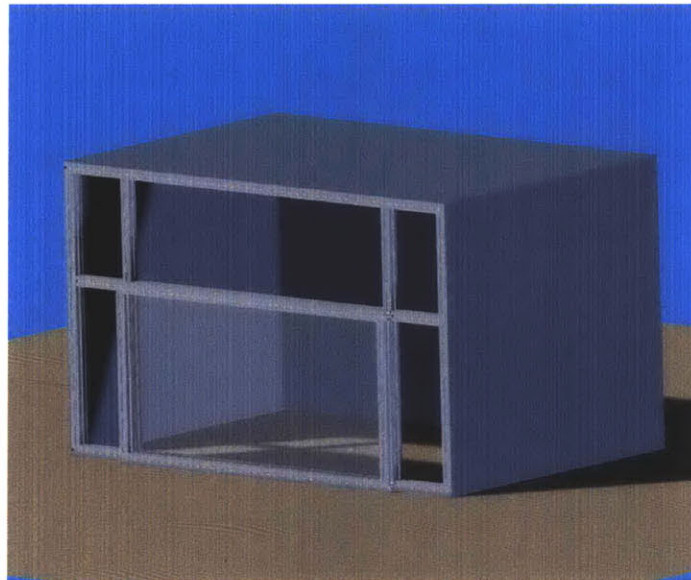


Figure 5.3: CAD model of lead enclosure with missing front panels

The total mass of the former structure was estimated to exceed 1800 kg. The additional mass of the equipment inside the enclosure meant that the underlying support structure had to be able to support at least 2.5 tons. To achieve this we decided to use a stainless steel optical table with steel support frame and without any pneumatic dampening. The CAD model of the table and the enclosure is shown in Figure 5.4.

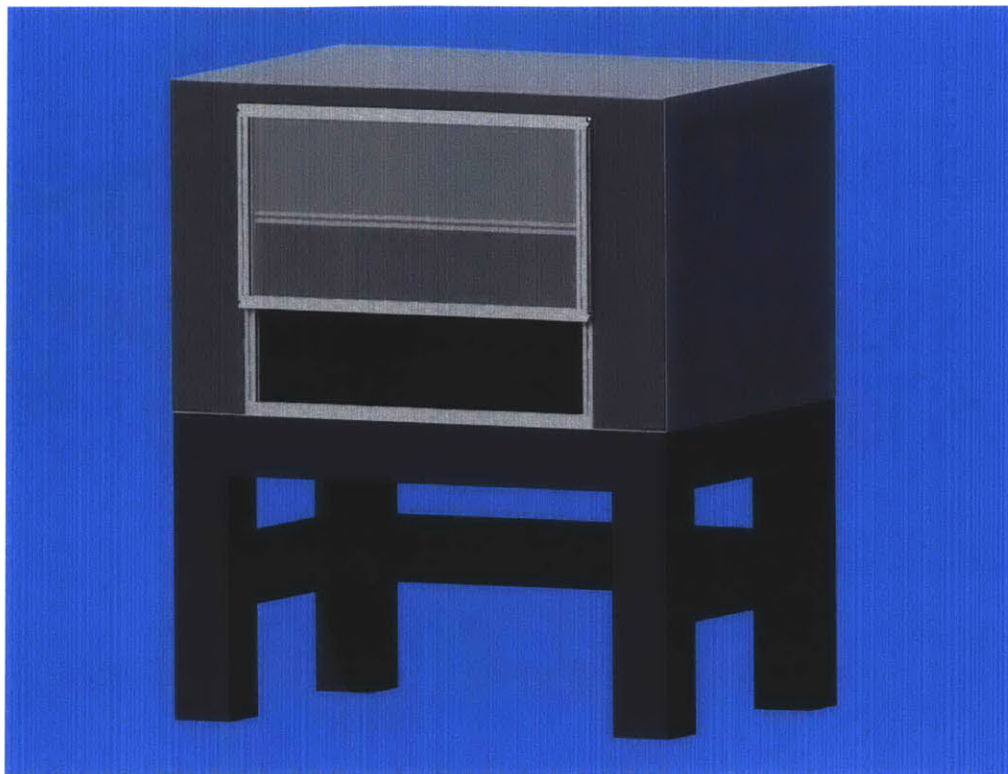
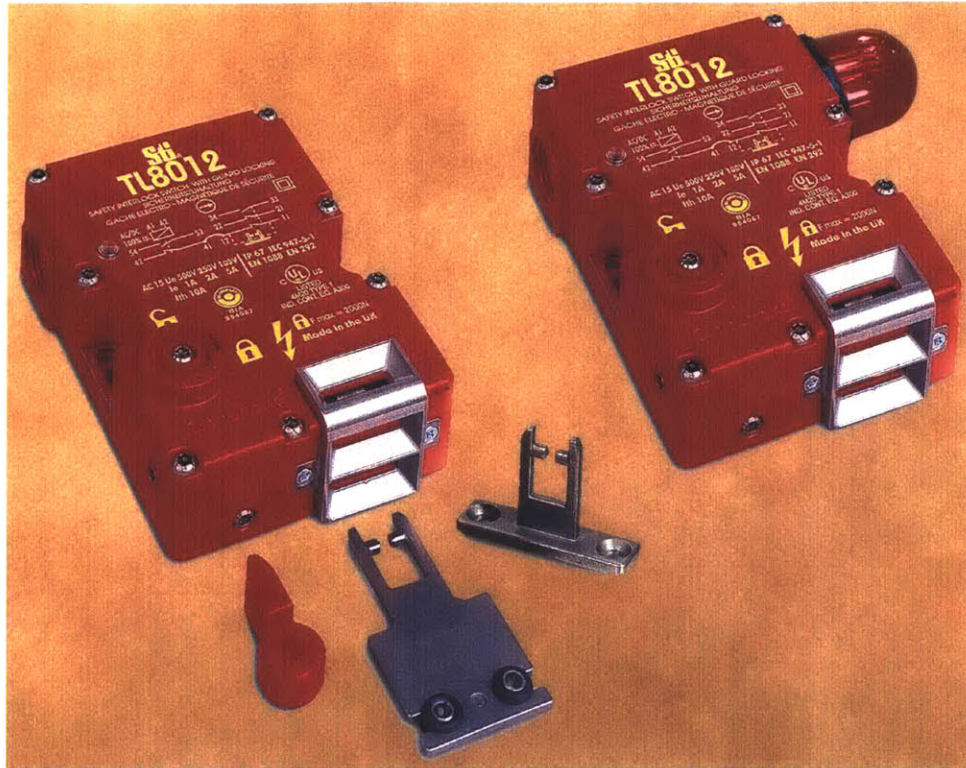


Figure 5.4: CAD model of main structure

5.3. Mechanics

Following the design guidelines for x-ray instruments provided by the Occupational Safety and Health Administration (OSHA) and the MIT Environment Health and Safety (EHS) Office we had to integrate mechanical interlock safety switches with all moveable parts of the lead shielded enclosure. The only function of those switches was to cutoff power to the x-ray source in case any portion of the enclosure was accidentally opened during an experiment. Strict functional requirements imposed by MIT EHS required the use of switches with failure rate no more than 1 time in 100,000 operations. The switches also had to mechanically engage to provide electrical connection from the x-ray source to its power supply. Typical mechanical interlock safety switch is shown in Figure 5.5.

In our microscope enclosure there was only one moveable part: the lead glass panel, which was allowed to slide up and down the front of the enclosure. The panel was restricted by mechanical stops on both ends of its travel and required an interlock safety switch at its closed position as illustrated in Figure 5.6 (a).



**Figure 5.5: Picture of mechanical safety interlock switches
[Image taken from SafetySwitch, 2006]**

The mass of the glass panel together with its aluminum support frame was estimated to be approximately 40 kg which posed a danger to the operators if released on their hands and also made it extremely difficult to open. Therefore, we designed a counterweight system to balance the mass of the lead glass panel and assist in its operation. Using steel cable and low-friction pulleys we decided to suspend a large lead-filled box on the back of the enclosure and connect it to the glass panel in front. The design of the counterweight system is shown on Figure 5.6 (b).

The last mechanical component of our x-ray microscope was a precision rotary motion stage. The function of this stage was to rotate a sample between the x-ray source and the imaging system, position the sample at a particular angle, and hold it there until a radiographic image was acquired. Continuous motion of the stage from 0° to 360° was necessary and angular positioning accuracy of at least 1' was essential for the operation of the microscope.

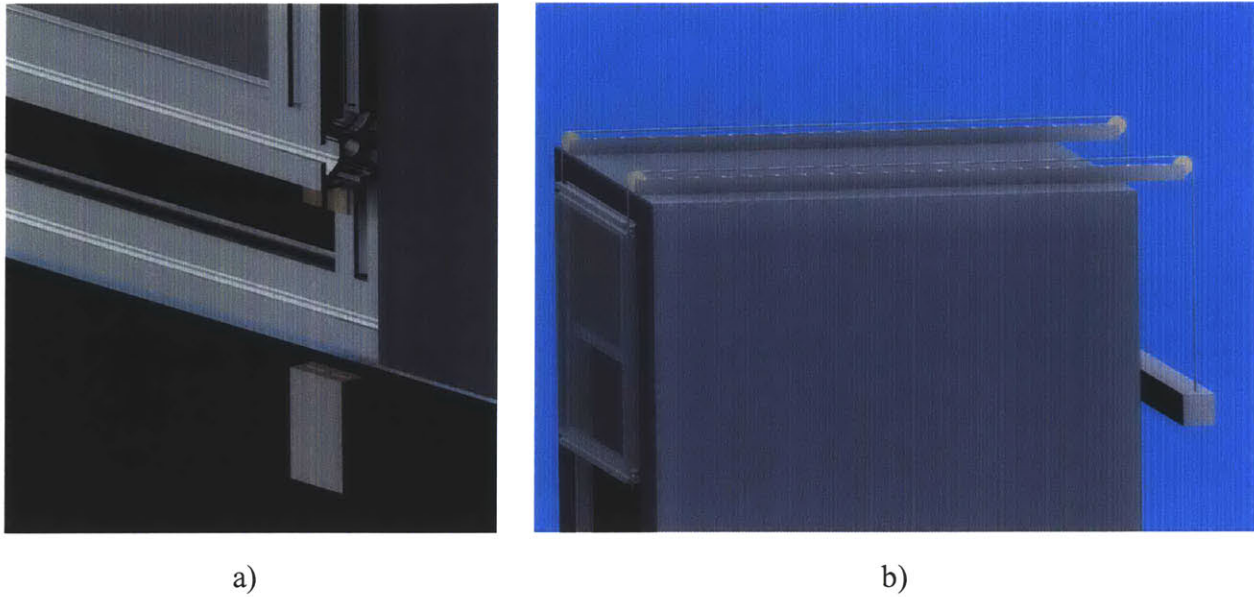


Figure 5.6: Key mechanical components
a) Interlock safety switch,
b) counterweight system.

5.4. Electronics

In the scope of this chapter we will only discuss electronic components for which additional design work was required prior to their integration into the x-ray microscope. Components, like the data acquisition PC, which came assembled and pre-wired and only had to be plugged into an outlet will not be discussed in this section.

Both OSHA and MIT EHS have strict rules about the electrical wiring of all devices, which emit harmful radiation when energized. In addition to the mechanically interlocking safety switches, discussed in detail in Section 5.2, an emergency switch was required which would cut down the power to the whole instrument if needed. Furthermore, any device using x-rays is required to have a highly visible red light, blinking whenever x-rays are being generated. Following the administrative guidelines and the functional requirements of our instrument we developed the wiring diagram shown in Figure 5.7.

Four levels of security were built into the microscope in order to prevent any x-ray overexposure to the operator. Mechanical switch type “mushroom” allowed for total power cutoff in case of emergency. Safety interlock was designed to prevent the x-ray

source from powering up in case the enclosure was left open. With the enclosure closed, the operator was required to press a “Power ON” button to power up the x-ray source and to initiate the generation of x-rays a data command was required from the operator via the RS-232 link.

In addition to the built-in security measures, two levels of warning were also designed into the microscope. A solid green light was designed to light up as soon as the “Power ON” button was pressed and a high-visibility red light, wired directly into the x-ray source, would start blinking as soon as x-rays were being generated.

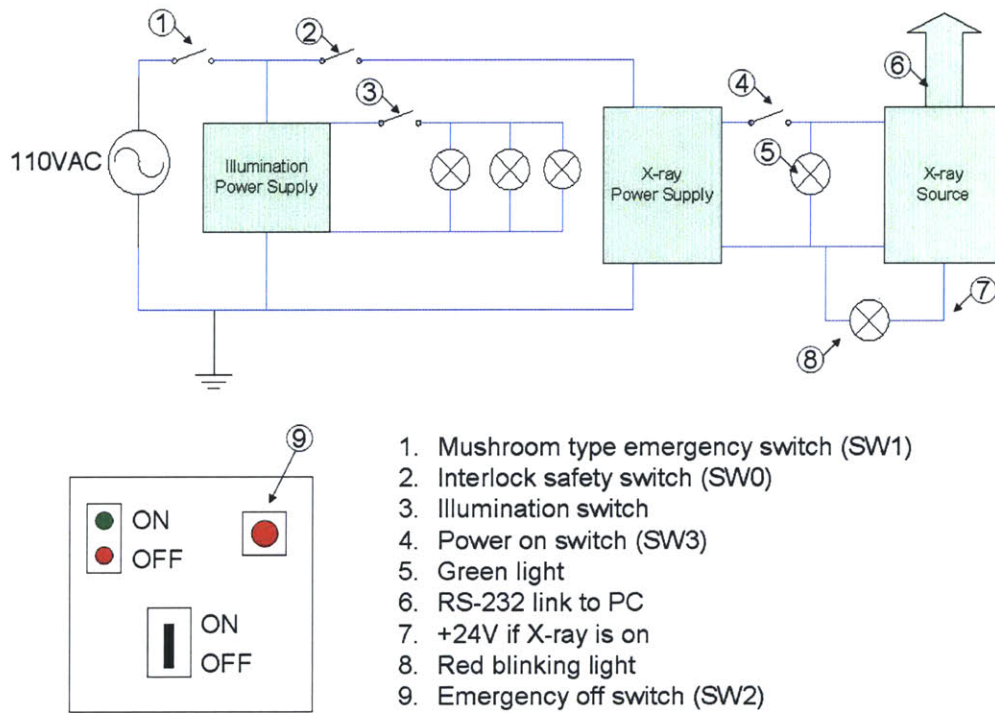


Figure 5.7: Main wiring diagram.

The last component of the electronics system was the internal illumination. As shown in Figure 5.7 the illumination was connected to the main power inlet of the microscope, but was on a separate power loop and did not interfere with the rest of the instrument.

5.5. Software

An important aspect of our x-ray microscope was its complete computerization. We wanted to ensure digital control over all functional components of the microscope from a

PC, which required the use of several custom software packages and their successful integration. Each software package had to function independently of the others; yet some of the application had to exchange data and run synchronously to ensure the proper operation of the microscope. Figure 5.8 shows a block diagram of the major software components of the x-ray microscope and the data communication links between them.

For simplicity, each software component is described in its own sub-section.

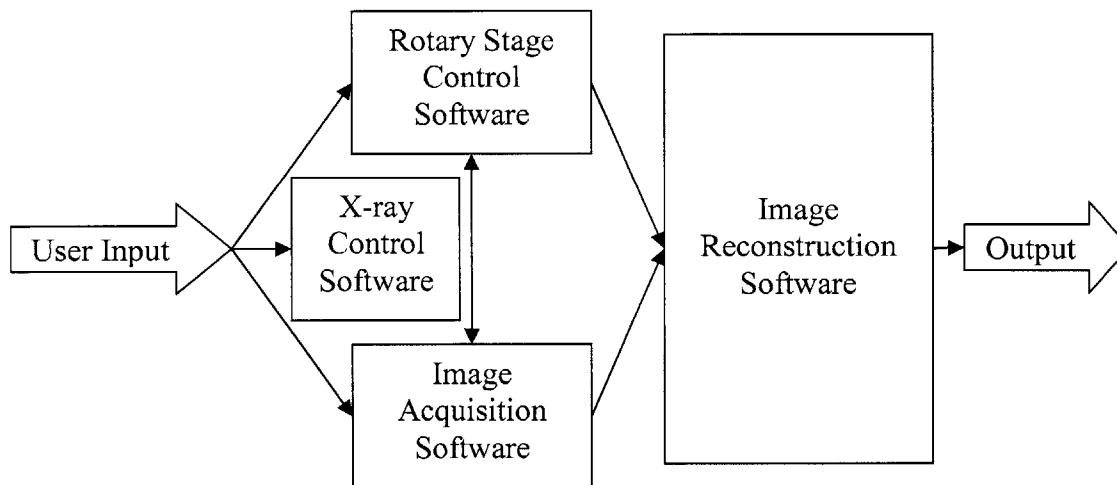


Figure 5.8: Major software component interactions.

5.5.1. X-ray Source Control Software

The control for the x-ray source was the simplest piece of software in the microscope. It was supposed to check for the presence of power to the source and wait for initiation command from the operator. It also had to provide the operator with adjustment options for the x-ray energy and power and give feedback on the current levels of x-rays being generated by the source.

The control over the energy of the x-rays allows the operator to choose the penetration depth (discussed in Section 3.2) and to set the transparency of different materials, while the change in power increased the intensity of the x-ray flux. This software can be used to adjust the contrast and brightness of the acquired radiographic images.

5.5.2. Rotary Stage Control Software

The rotary stage control software had to ensure the correct angular positioning of the sample and to provide feedback on relative and absolute position of at any time. Features of this software essential to the microscope operation were speed control and precise timing and coordination with the image acquisition software.

The rotary stage control software ensured that the sample would be stationary while radiography was being acquired and that the next angular position was reached prior to the acquisition of the next image. Successful operation of this software guaranteed minimal image blurring and reconstruction artifacts.

5.5.3. Image Acquisition Software

The image acquisition software handled the interactions between the data processing CPU, the digital CCD camera, and the digital frame grabber, shown in Figure 5.1. Important features of this software were variable capture speed, contrast adjustment, and sensitivity adjustment. Just as with the rotary stage control software, precise timing and coordination were vital to the acquisition of blur-free images.

Images acquired by the image acquisition software had to be conditioned and saved in a format, suitable for further image processing with minimal information loss and noise accumulation.

5.5.4. Image Reconstruction Software

The image reconstruction software was the single most complex feature of our microscope, even though it never interacted with any hardware. The data collected by the image acquisition and combined with the position data provided by the rotary stage control were fed into the image reconstruction software and after a series of image processing steps an image was obtained. A flowchart of the algorithm used by the image reconstruction software is shown on Figure 5.9. The general method of image reconstruction is discussed in details in Chapter 4.

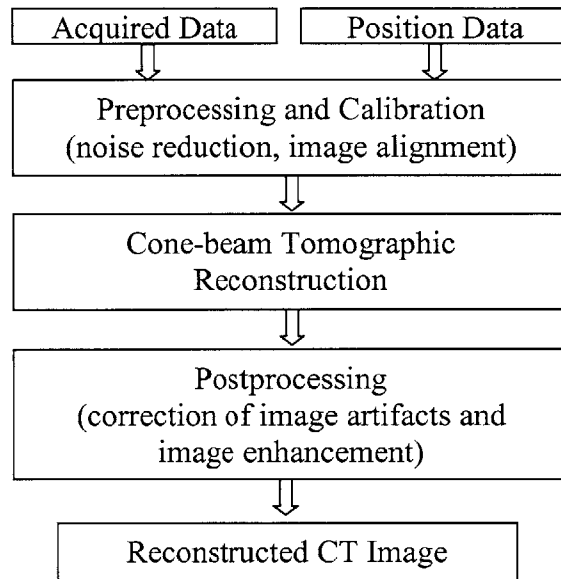


Figure 5.9: Image reconstruction algorithm flow diagram

The algorithm described by the flowchart in Figure 5.9 is greatly simplified and was only used as a guideline in the design phase of the instrument. The actual image reconstruction algorithm is described in great detail in Section 6.5.4.

5.6. Design Summary

Starting with the schematic shown in Figure 5.1 and following the appropriate safety guidelines a complete model of the x-ray microscope was designed. Even though the model lacked some fine details, overall it resembled a complete instrument and was used in the engineering of the physical device. Figure 5.10 shows the finalized CAD model of the x-ray microscope as it was envisioned prior to its construction. Chapter 6 will discuss in depth the steps taken towards the physical realization of this model.

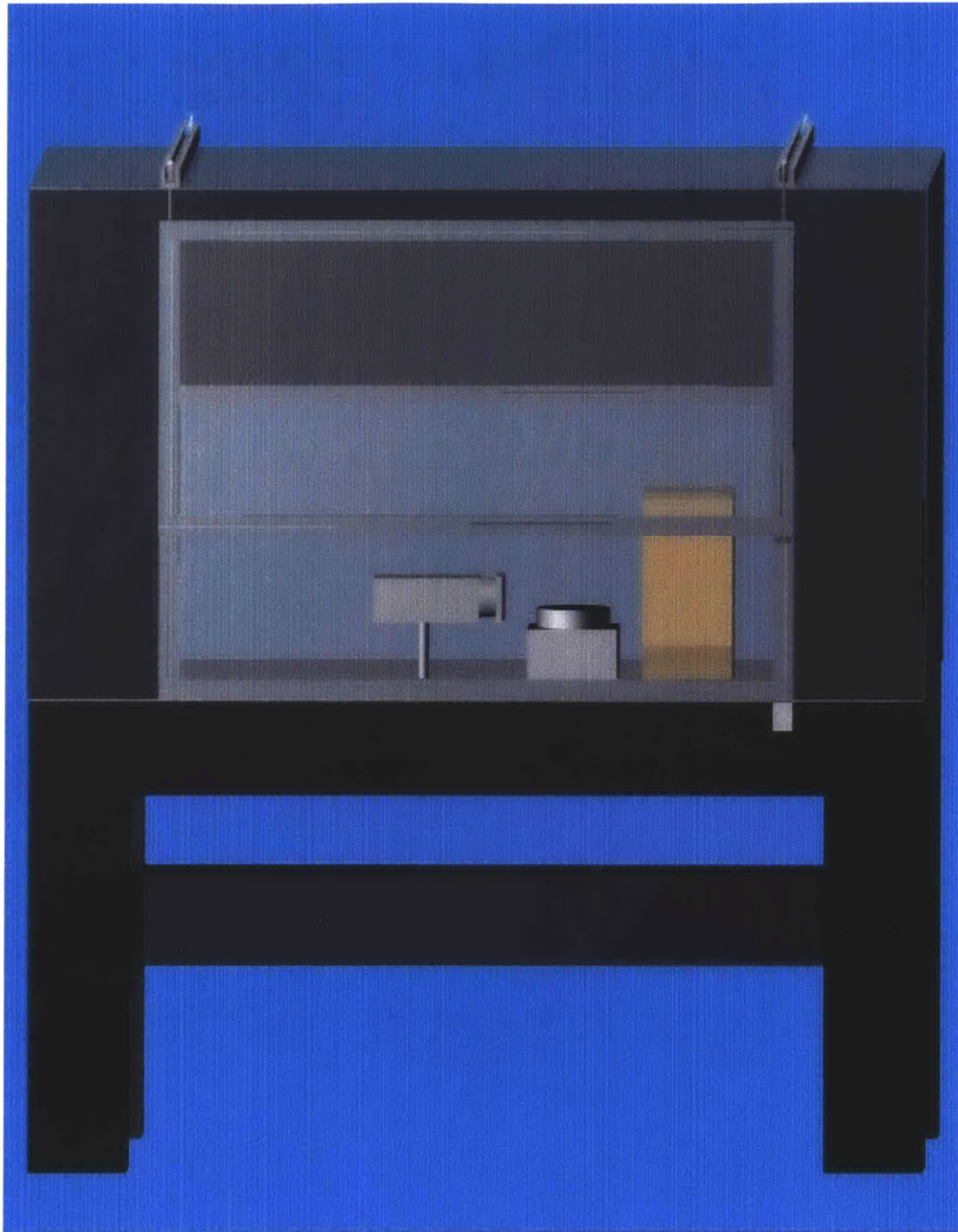


Figure 5.10: CAD model of complete system.

6. X-ray Microscope Implementation

Following the guidelines set in the design phase and described in Chapter 5, the instrument was successfully built. This chapter will describe the engineering process as well as the separate components which are part of the microscope.

6.1. Functional Components

The functional components of the instrument consist of the x-ray source and the imaging system. They form the backbone of the microscope and are essential for its performance, speed, and resolution. Although they were purchased as off-the-shelf products, their successful customization and integration were the most critical steps in the implementation of the system.

6.1.1. Digital CCD Imaging Camera

Based on performance, price, and availability, the Hamamatsu C4742-56-12NR was chosen as the most appropriate digital CCD camera for our project. Its ultimate resolution of 1280×1024 pixels, pixel size of 30-40 μm , and variable frame rate of up-to 32 frames per second (fps) outperformed all other digital CCD cameras under \$100,000. Combined with a price tag in the \$25,000-\$30,000 range and immediate availability, this camera was the best one on the market for us. Furthermore, it was supplied together with a digital controller, PCI digital frame grabber, and autonomous power supply and temperature control unit. Figure 6.1 shows a picture of the complete camera setup while Appendix A.1 contains the complete technical specification.

The installation of the camera consisted of mounting the PCI digital frame grabber into a PC, mounting the camera head onto the optical table, and attaching the control box via one RS-232 and one RS-422 port. Following the hardware installation, the software drivers were installed with no difficulties. Sample images were acquired with the supplied image acquisition software shortly after the system was powered and their quality was satisfactory even before any image enhancement was performed. Further details on the images from the camera can be found in Chapter 7.



Figure 6.1: Hamamatsu C4742-56-12NR digital CCD camera setup.

6.1.2. X-ray Source

Guided by the CCD camera requirements and by the limited space constraints, the Source Ray SB-80-500 x-ray source was chosen as the best solution for our project. The SB-80-500 was the smallest source which offered variable energy output levels in the 0-80 keV, variable power output levels in the 0-40 W range, and was supplied with a digital controller interface. The installation of the x-ray source required a connection to a 24 VDC, 3 A power line, supply of 4 m³/min room temperature air for cooling, and an RS-232 connection to a PC. Figure 6.2 shows the x-ray source after it was installed and connected into the microscope; Appendix A.2 contains its complete technical specification.

Upon installation of the drivers for the x-ray source, we discovered an incompatibility between the camera drivers and the x-ray source drivers. The camera software was constantly crashing whenever the x-ray source drivers were present on the same PC. This was a serious problem, which we could not solve, so we dedicated an additional computer to run the x-ray source. Apart from this inconvenience, the operation x-ray source has been trouble-free.

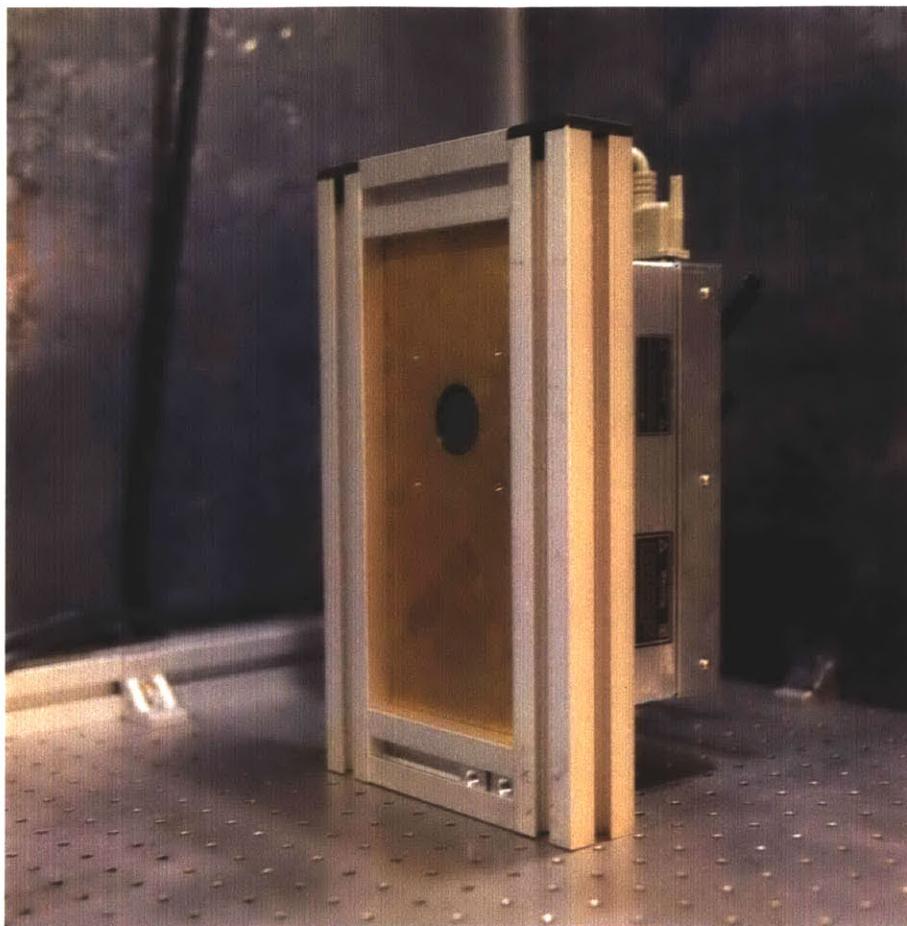


Figure 6.2: Source Ray SB-80-500 x-ray source.

Additional features of the x-ray source were the provided analog interrupt for immediate shutoff in case of emergency, which was connected to the mechanical safety interlock switch, and the 24 VDC output line, which was powered whenever x-rays were being generated inside the source. This output was used to drive the blinking red warning light, discussed in Sections 5.4. Further details on the wiring of the source can be found in Section 6.4.

6.2. Structure

Starting from the ground up with the construction of the x-ray microscope, the first major structural component was its foundation. As discussed in Section 5.2, we decided to use a 1.8 m by 1.2 m optical table and to build the whole instrument on top of it. Conveniently, a TMC optical table was available for the project and we decided to use it. After thorough research on the weight carrying capabilities of the table, it became clear

that all pneumatic components had to be removed and only the solid steel frame was left to carry all the loads.

After the foundation was secured, the assembly of the lead-lined enclosure began. MK System 40 mm profiles were used in the construction of the support frame. All profiles were secured with 6 mm stainless steel fasteners, treated with thread locking solution, to ensure structural stability and safety. The completed structure is shown in Figure 6.3.

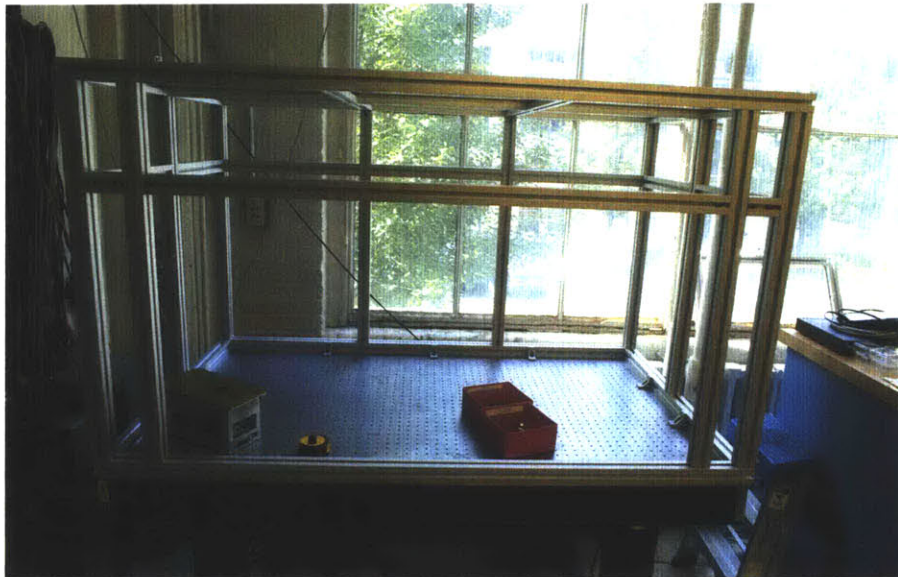


Figure 6.3: Aluminum support frame.

Structural integrity tests were performed on the aluminum frame and the results showed minimal deflection under substantial loads, ensuring the longevity of the structure. The final step in the construction of the enclosure was the attachment of the lead panels on all sides. This task was physically challenging, as the weight of the panels ranged from 40 kg to over 150 kg and lifting them required 3 to 4 people. It took several weeks of heavy lifting and lead trimming to get all the panels into the correct shape and onto the instrument. Each lead panel was fastened with multiple 6 mm bolts to the aluminum frame and extra care was taken to ensure that no two fasteners lie on the same line with the x-ray source. This ensured maximum protection from the harmful radiation. Figure 6.4 shows the completed lead enclosure, mounted on top of the optical table.

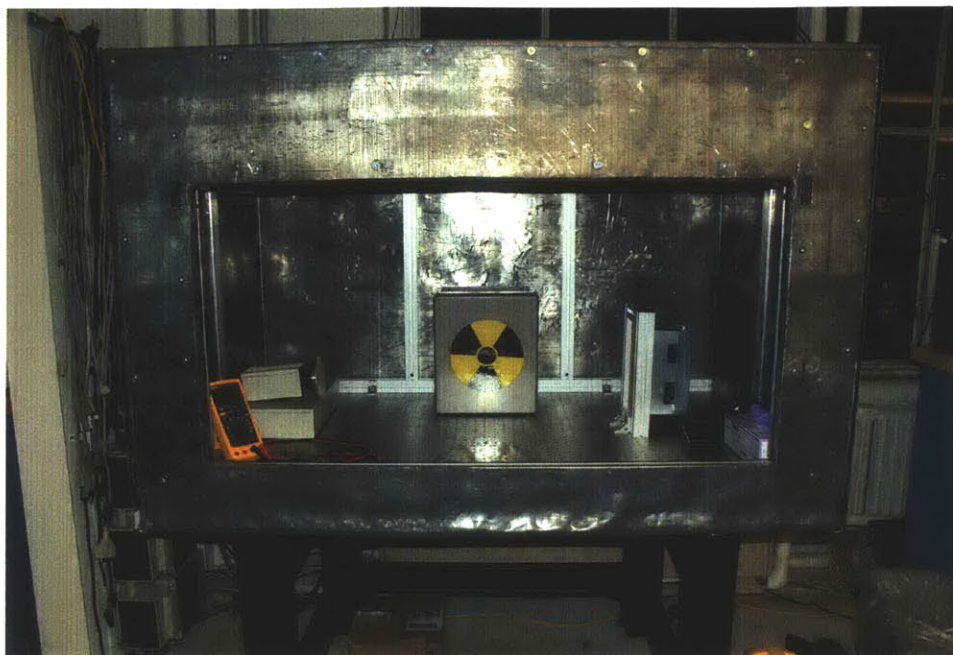


Figure 6.4: Lead shielded enclosure.

Two sliders were installed to support the front lead glass panel, but the panel was not installed at this stage, since it would block the access to the interior of the instrument and interfere with the construction. The glass was installed last, together with the mechanical safety system and the counterweight and will be discussed in Section 6.4.

6.3. Electronics

After the main structure was completed, the focus was transferred towards the electronics of the microscope. The first step was to locate a power supply for the x-ray source and integrate it, together with the cooling system, into the instrument. We decided to use a Phaseo ABL-7RE switching power supply because of its small size, built-in fault protection, and low output noise level. In addition, we decided to use two NMB 4715KL-05W-B50 24V DC Brushless Fans, directly attached to the x-ray source, providing over 5 m³/min of forced air to cool it. Figure 6.5 (a) shows the power supply and Figure 6.5 (b) shows the cooling fans installed onto the x-ray source. Specifications of the power supply and wiring diagrams are located in Appendix A.3.

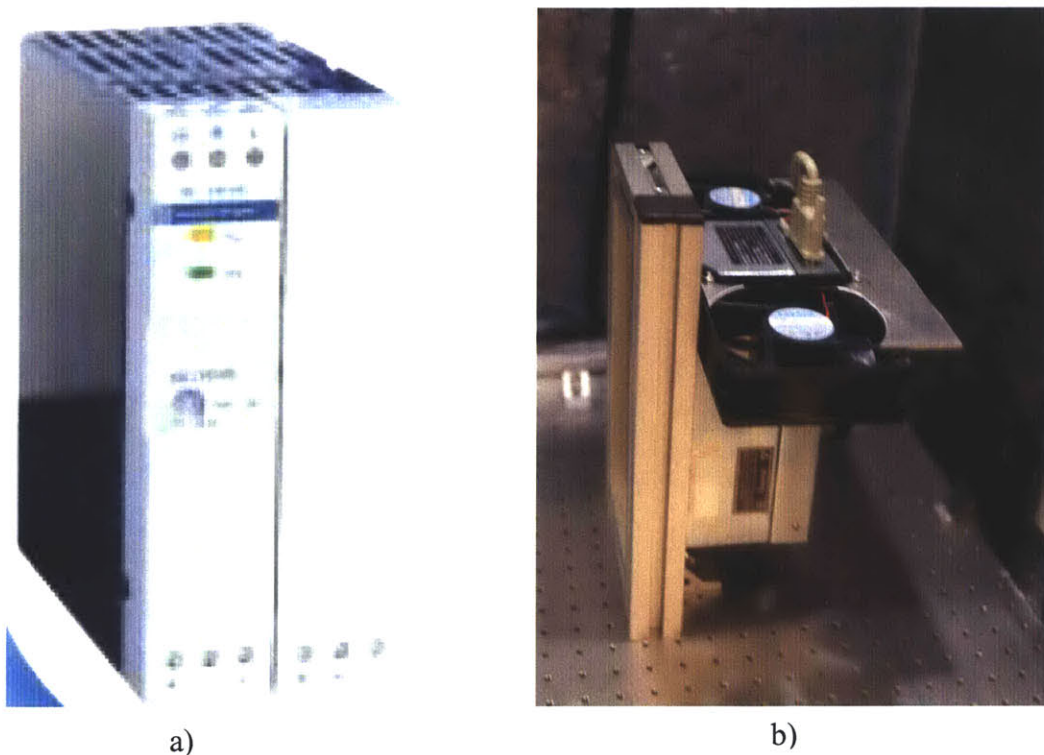


Figure 6.5: X-ray source support electronics.
a) Power supply,
b) Cooling fans installed onto the x-ray source.

After the x-ray source was wired, construction began on the interior illumination. Two aspects of the illumination were critical to the performance of the microscope: the heat dissipation from the light sources and the RF noise generated by them. The use of conventional incandescent light was impossible, since the interior of the microscope would constantly heat up and all temperature-sensitive experimental parameters would be affected. Luminescent light were also inappropriate as they emit a mixture of radio frequencies, which could interfere with our measurements. The best choice for interior illumination was LED light.

We decided to use two Luxeon Flood DS-24 LED arrays, each providing 450 lumens of 5500 K white light while dissipating less than 5 W of heat. The two LED arrays were connected in series to a fixed current source and attached with thermal paste to the aluminum frame inside the lead enclosure. Figure 6.6 (a) shows a picture of the LED arrays and Figure 6.6 (b) shows their wiring diagram.

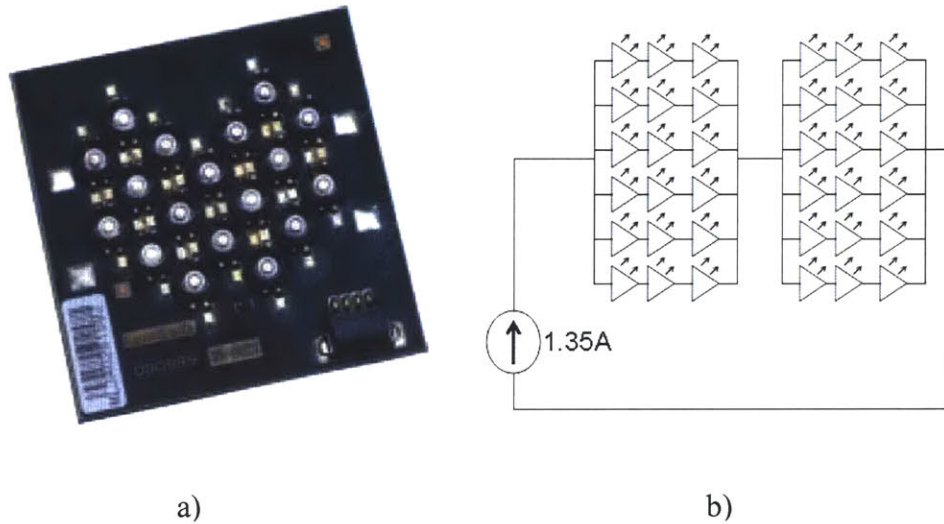
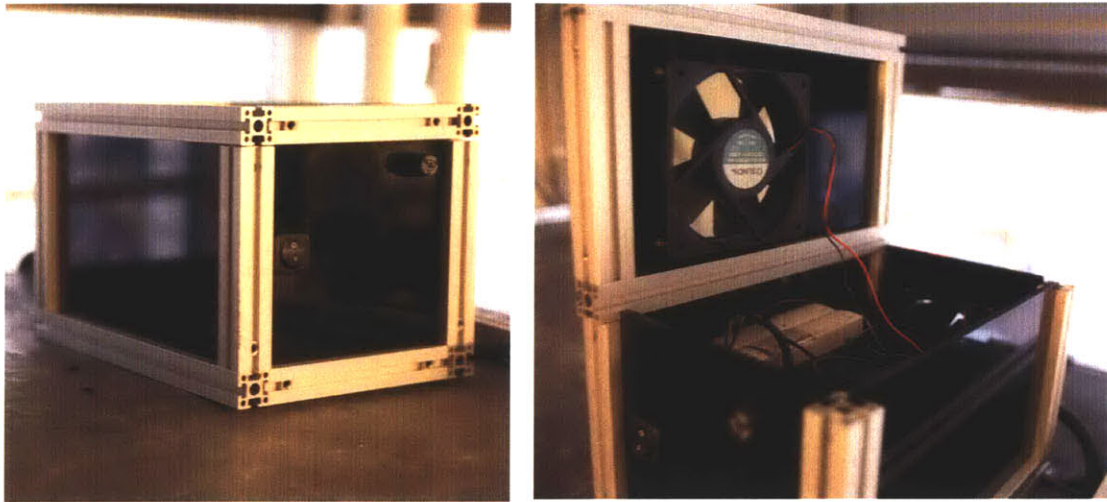


Figure 6.6: Illumination.
a) LED arrays,
b) LED arrays wiring diagram.

In order to ensure continuous operation of the microscope we had to minimize the chance of power supply overheating. To achieve this, an air-cooled power supply enclosure was constructed to contain both the x-ray source and the illumination power supplies. Figure 6.7 shows a picture of the power supply enclosure.

The final and most important step of the electrical wiring of the microscope was the wiring of the safety components. Following the design diagram described in Section 5.4 and shown in Figure 5.7 all safety features were wired and tested. A flexible conduit was routed through the back of the lead-lined enclosure and all the wiring was put through it.

The control panel and the emergency switch were installed on the right side of the enclosure, providing easy access in case of an emergency. The warning lights were installed on top of the microscope, ensuring high visibility at all times and all exposed cables were shielded by flexible plastic conduit. Once completed, the safety system was tested under all feasible working scenarios. Figure 6.8 shows pictures of the electrical safety components after they were installed on the microscope.



a)

b)

Figure 6.7: Power supply enclosure.
a) Side view,
b) Top view (open).



Figure 6.8: Electrical safety components.

6.4. Mechanics

The safety components of the mechanical system were installed together with the sliding front glass panel. The mechanical safety interlock switch, Schmersal AZ16zvrk-M20, was mounted at the lower right corner of the panel and connected to the electrical system according to the diagram shown in Figure 5.7. The counterweight was suspended from the rear of the microscope on two 4mm steel cables. The counterweight was precisely calibrated to balance the weight of the glass panel and to assist in its opening. The mechanical interlock was used to provide the additional force needed to hold the panel closed. Figure 6.9 shows the installed interlock switch and counterweight system.

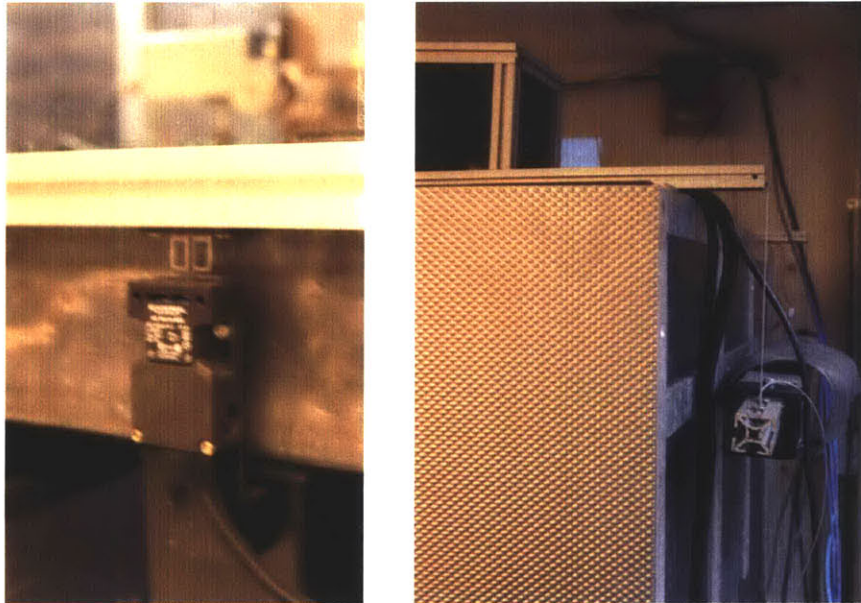


Figure 6.9: Mechanical interlock safety switch (left) and counterweight system (right).

The last hardware component of the microscope was the precision rotary motion stage. We decided to use an Aerotech ART 304 rotary stage, driven by a Compumotor Zeta57-51-MO stepper motor. The stage was capable of moving at speeds of up-to 5 rev/min and offered an ultimate resolution of approximately 10 arcsec. Driven by the Compumotor Zeta6104 indexer drive from a PC via an RS-232 connection, the response time of the stage was under 1 ms, allowing for the precise monitoring of the position of the sample at all times. Figure 6.10 shows the rotary stage together with its stepper motor and drive.

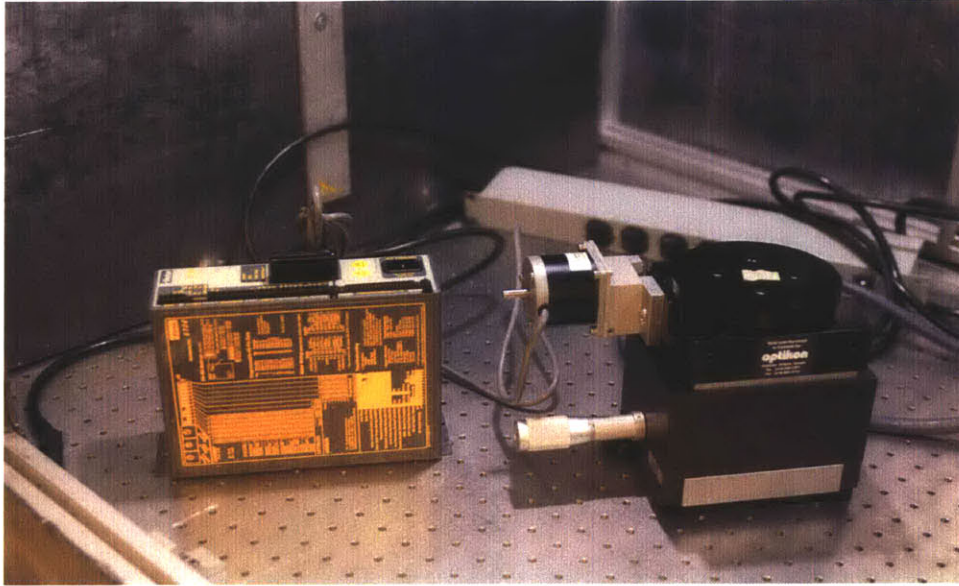


Figure 6.10: Rotary stage together with its stepper motor and drive.

6.5. Software

This section describes all software components of the x-ray microscope. The information here is intended to help potential future operators of the instrument learn how to operate it.

Algorithms and other essential information are provided in the body of the section, while the source code can be located in Appendix B. Executable files together with all the drivers, DLLs, and software developer's kits are located on the DVD, supplied with this thesis.

6.5.1. X-ray Source Control Software

The control software for the x-ray source was supplied by the manufacturer of the source in two forms: an Application Program Interface (API), to be used in LabView or Visual Studio.NET and a stand-alone program, compiled for use in a Windows environment. The API offered greater flexibility and allowed the development of a custom graphics user interface (GUI). However, for our microscope application, the stand-alone program supplied ample control capabilities.

As discussed earlier, in Section 6.1.2, the x-ray source drivers could not be installed on the same PC with the rest of the software. Luckily, the system requirements, necessary for the operation of the x-ray source control software were minimal and we managed to find an old IBM laptop from which to run this software.

Upon execution, the stand-alone application automatically verified the presence of a digital control interface for an x-ray source on serial communication ports 1 and 2 (COM1 and COM2) and established a two-way communication line via an RS-232 protocol. Afterwards, the GUI shown in Figure 6.11 was displayed.

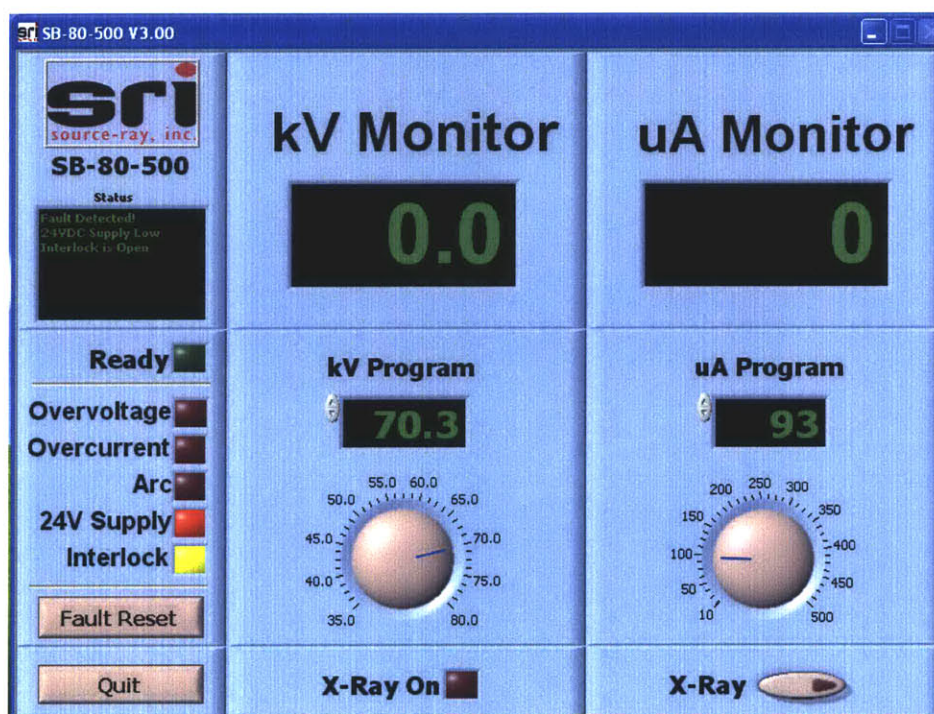


Figure 6.11: X-ray source control software GUI.

The left side of the GUI showed the status of the x-ray source as well as the status of the built-in safety features. The “OverTemp” indicator, connected to a thermo couple inside the source, would light up if the temperature of the vacuum tube exceeded the operating range. The “24V Supply” indicator would turn on if the 24 VDC power line was interrupted. The “Interlock” light indicated if the mechanical safety interlock switch was properly engaged. The “Fault” light indicated an error in the safety protocols and required an operator to press the “Fault Reset” button to turn off. The green “Ready” light

indicated all safety checks were successful and the x-ray source was ready for operation. The “X-Ray On” light indicated the current operating state of the source.

The middle portion of the GUI dealt with the energy settings of the x-ray source. The “kV Monitor” displayed the value of the average radiation energy, produced by the source. Underneath, the “kV Program” panel was used to set the energy value to a desired level. The oval button, labeled “X-Ray” was used to turn the source on and off.

The right side of the GUI set the intensity of radiation, in a similar fashion to the energy control panel. The “Quit” button, located at the lower right of the GUI, was used to close the software application, while terminating the operation of the x-ray source.

6.5.2. Rotary Stage Control Software

The rotary stage was supplied without any software, drivers, or APIs. The only information supplied with the stage was the set of command instructions, used to control and monitor the stepper motor. These instructions were sent via an RS-232 protocol to the Compumotor Zeta6104 Indexer drive and the result from their execution was supplied back to the PC. The diagram on Figure 6.12 shows a typical communication scheme, used to control the rotary stage.

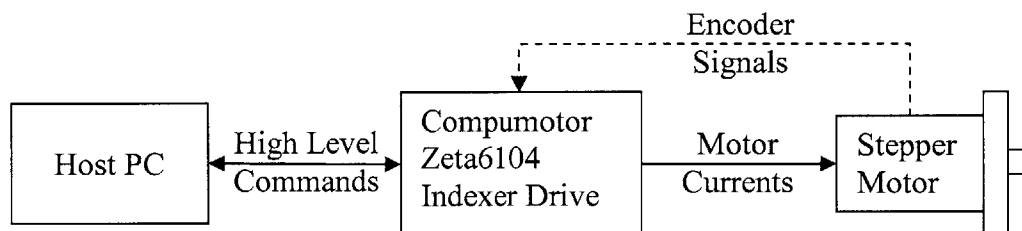


Figure 6.12: Communication scheme used to control the rotary stage.

In order to provide a user-friendly tool for controlling the rotary stage, we created a GUI with all features, necessary for the operation of the x-ray microscope. This piece of software was written in Visual Basic.NET and was compiled as a stand-alone application. Figure 6.13 shows the layout of the GUI, while the source code for it is located in Appendix B.1.

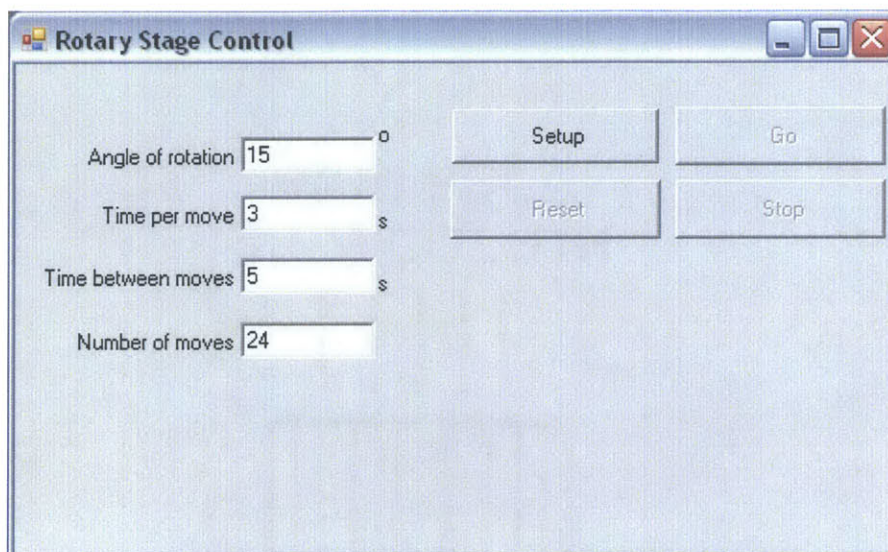


Figure 6.13: Rotary Stage Control software GUI.

Four inputs were required to control the rotary stage. The “Angle of rotation” specified the angle rotated each move, the “Time per move” was the time during which each move had to be completed, “Time between moves” was the delay after one move has completed and before the next move starts, and “Number of moves” specified how many times the motion had to be repeated.

Pressing the “Setup” button generated a program and uploaded it to the indexer drive and enabled the rest of the buttons. “Start” initiated movement, while “Stop” interrupted it. “Reset” cleared the program from the indexer drive and enabled the “Setup” button, while disabling everything else.

6.5.3. Image Acquisition Software

The image acquisition software was supplied by Hamamatsu together with their digital CCD camera. The application, called Wasabi, was easy to use and provided all the features we needed. It offered real-time video streaming at over 30fps, video capture on a frame-by-frame basis with fixed time between frames, automatic and manual exposure control, and some basic image enhancement tools. Wasabi also supported several file formats for storing or exporting data. Figure 6.14 shows the main screen of the application.

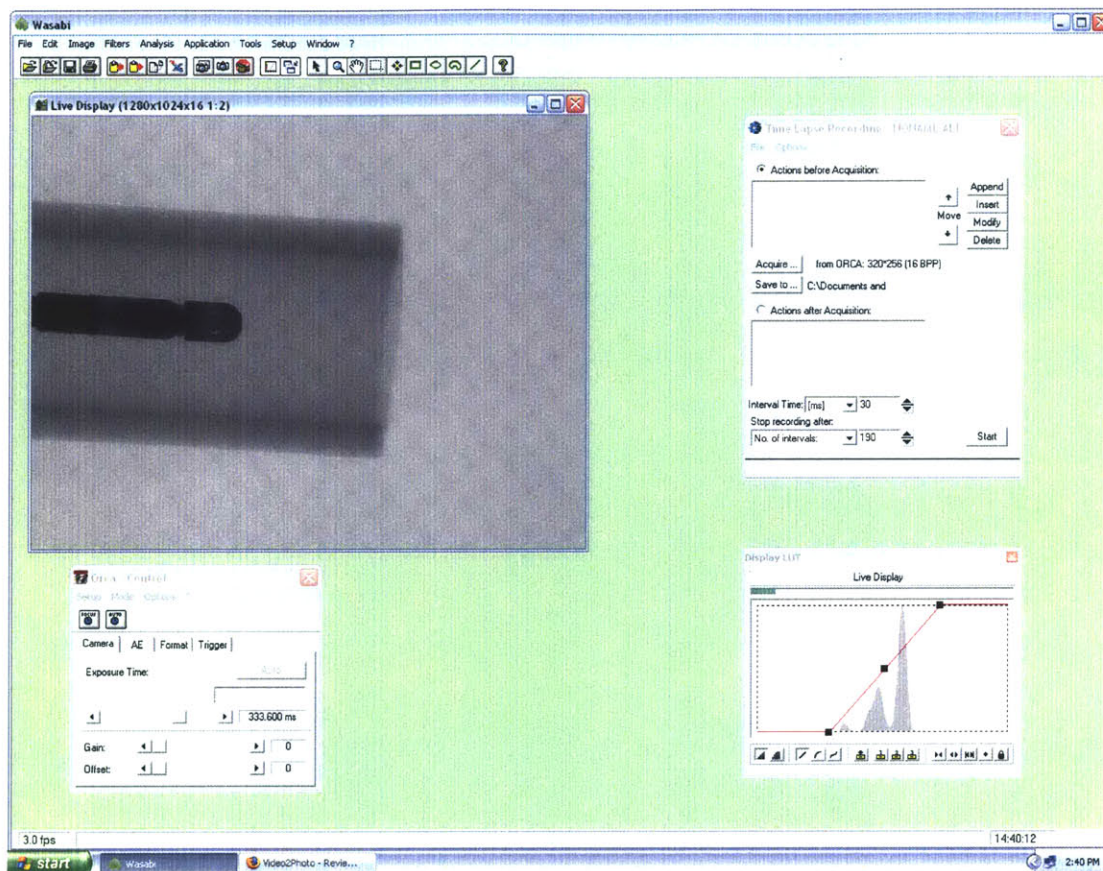


Figure 6.14: Wasabi main screen.

Upon execution, Wasabi established a connection with the camera control box and started the “Orca Control” module. This module was used to control the exposure time, gain, and offset of the camera as well as the size and resolution of the image being captured.

The “Live Display” module was used to provide real-time images from the digital CCD and also show a preview of what was being captured. The “Display LUT” module was used to correct brightness and contrast in real-time by means of simple histogram manipulations.

The last major part of Wasabi was the “Time Lapse Recording” module, which was used to record sequences of images and save them on the hard drive. Similarly to the Rotary Stage Control software, the settings for the time lapse recording included the exposure time, delay between consecutive captures, and number of images to be recorded.

The only problem we had with Wasabi was the fact that its video capture module could not record at the full bandwidth of the on-screen preview module. While we could see movies on the screen at 32 fps, we could not save them on the hard drive. This is why a screen capturing application, HyperCam v.2.0, was used to record the movies displayed by Wasabi. The interface of HyperCam is shown in Figure 6.15.

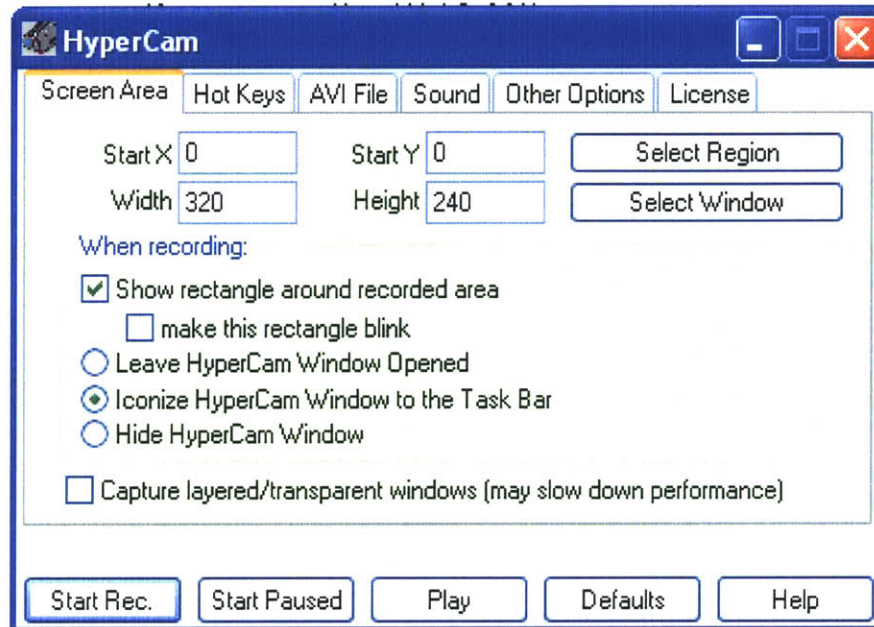


Figure 6.15: HyperCam interface.

6.5.4. Image Reconstruction Software

The image reconstruction software was implemented using the techniques described in Chapter 4. The image generation, enhancement, and visualization were written for MatLab R14 and were not compiled as stand-alone applications. Figure 6.16 shows a flowchart of the entire image reconstruction process.

The image reconstruction process can be separated into three categories: 2D radiographies, conventional 3D CT images, and volumetric 3D CT images. Each category used different reconstruction and enhancement techniques, which will be discussed separately.

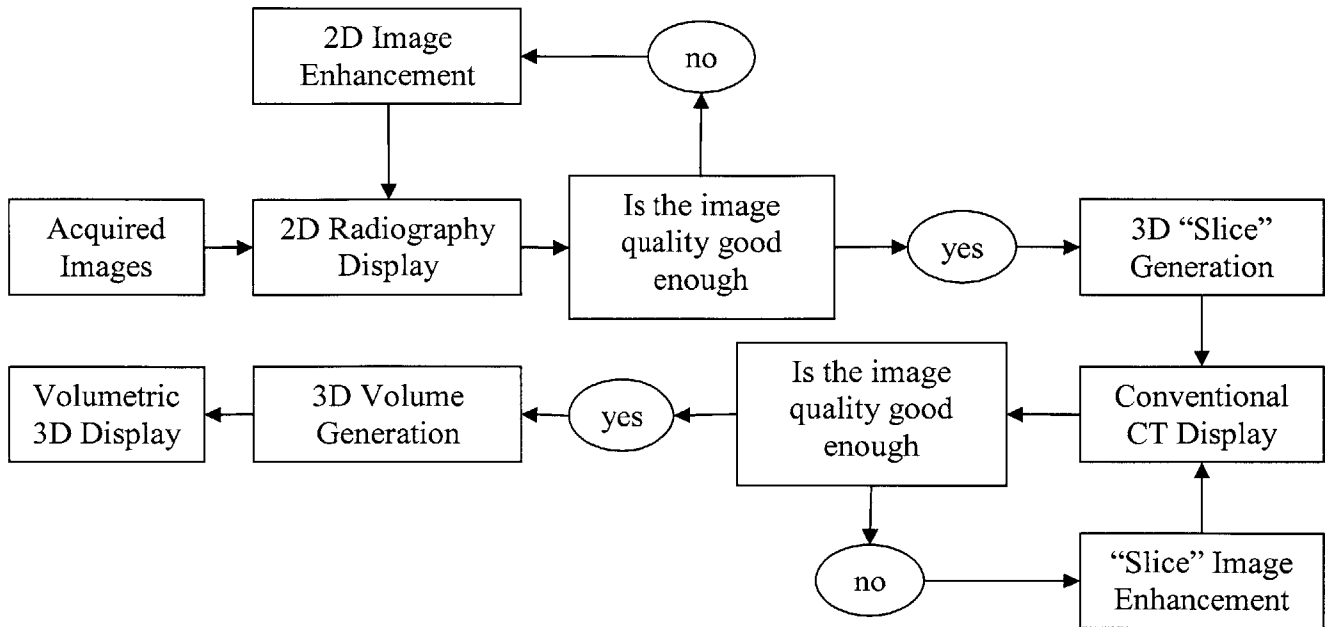


Figure 6.16: Flowchart of the entire image reconstruction process.

6.5.4.1. Two-dimensional Radiography Generation

The images acquired from the digital CCD camera were in fact 2D radiographies, hence they could be displayed as such without any further image processing. Matlab's *imread()* function was used to load the raw images saved by Wasabi and *imshow()* was used to display them.

Depending on the type of experiment, the exposure time, and the environment the quality of some images was reduced and additional image enhancement was required. This led us to the development of the image enhancing application, shown in Figure 6.17. The application loaded an image from the hard drive and applied to it, in real time, digital image enhancement techniques to improve its quality. For an in-depth discussion on the function of each technique, see Section 4.2. The application was developed as a MatLab GUI and its source code is provided in Appendix B.2.

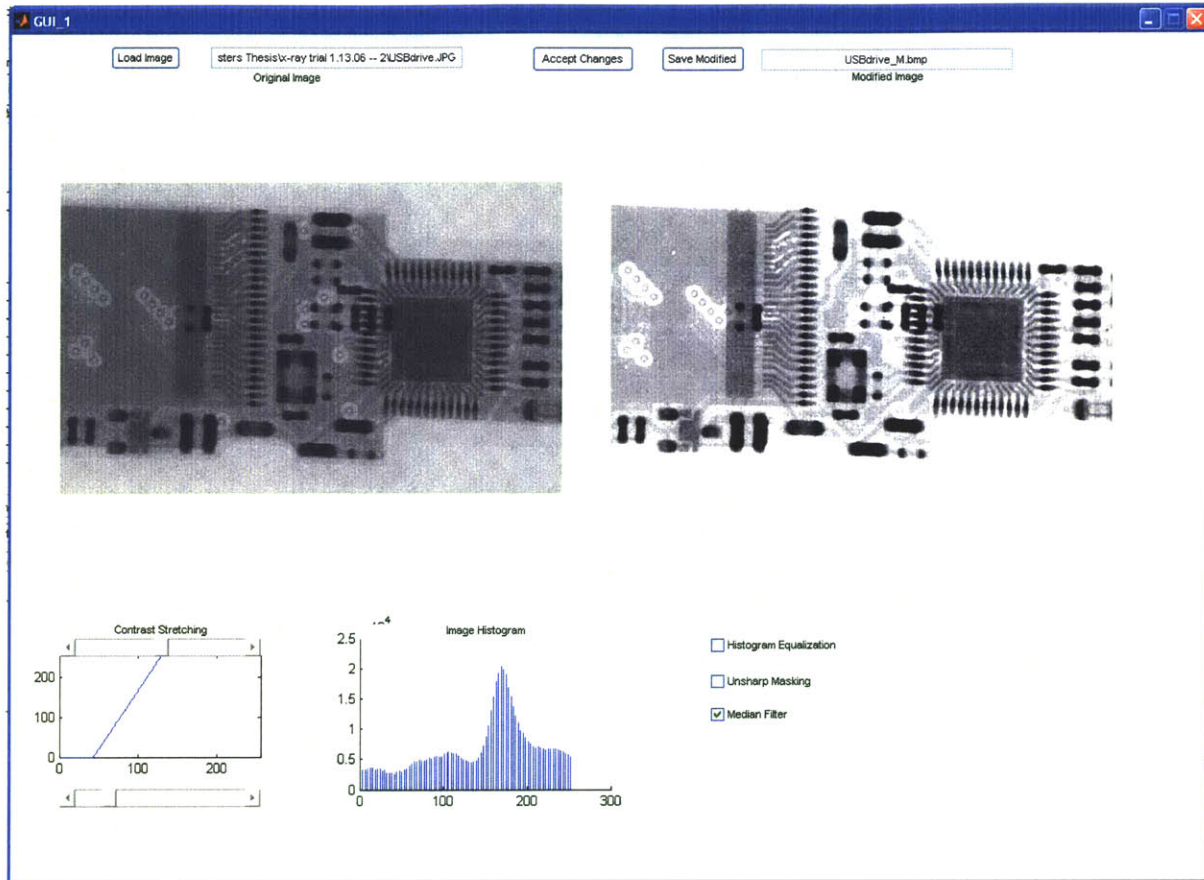


Figure 6.17: Image enhancing application.

6.5.4.2. Conventional 3D CT Image Generation

After a scan of an object was completed and all radiographic images were processed and saved, the 3D reconstruction began. Using the MatLab routine *Main.m* provided in Appendix B.3 all projection images were loaded and via inverse Radon transformation conventional CT images were generated. The parameters required for the successful execution of *Main.m* included the resolution of the radiographic images and the number of different projections acquired during the scan.

Once the 3D slices were generated, another image enhancing routine, *Iadjust.m*, was executed in order to remove reconstruction artifacts and to enhance the image contrast. The source code for *Iadjust.m* is provided in Appendix B.3.

6.5.4.3. Volumetric 3D CT Image Generation

Volumetric 3D image generation was the last step of the image reconstruction process. After all of the cross-sectional images were generated they had to be combined into a 3-dimensional volume and plotted using the *isosurface()* command in MatLab. Image artifacts and discontinuities in the volumetric objects were repaired using the *patch()* script and the results were plotted in a 3D sample space. Example program, used to generate 3D objects from slices can be found in Appendix B.4.

In addition, the third-party software Mimics 9.0, was used to reconstruct 3D objects from the slices generated from our microscope. Unfortunately at the time of the writing of this document our free license for Mimics 9.0 had expired and we were unable to generate any 3D volumetric images with it at this time.

6.6. Complete System Overview

Upon completion of the software, the construction of the microscope was finished. The instrument was ready for initial testing and for experimental work. Figure 6.18 shows a picture of the complete device.



Figure 6.18: Picture of complete system.

7. Results and Discussion

This chapter contains images, acquired by our x-ray microscope during various experiments and tests. We have also included images and image analyses, used in the calibration of the system.

7.1. Microscope Calibration

After the instrument was constructed it was important to characterize its performance prior to running any experiments. We began by testing and calibrating the main components of the system, making sure everything was working within manufacturer's specifications and the system was safe to operate.

7.1.1. Lead Enclosure

Since radiation safety was primary concern, the first tests performed on the x-ray microscope were to measure any leakage of radiation from the enclosure. According to the MIT Environmental Health and Safety office, x-ray generating equipment could be used in a populated environment as long as the radiation leaking from the equipment, measured 1 m away from it, resulted in less than 500 $\mu\text{rem/hr}$.

With the x-ray source operating at its maximum settings, initial tests were performed around the entire x-ray enclosure using a Babyline 81 EM18316 portable x-ray dosimeter and a peak of 180 $\mu\text{rem/hr}$ was measured near the top and bottom of the lead glass panel. The radiation, measured 1 m away from the enclosure did not exceed 17 $\mu\text{rem/hr}$, while the background radiation in the room, with the x-ray source disabled, was measured to be between 15 and 17 $\mu\text{rem/hr}$. For comparison, we measured the radiation outside of the laboratory under the direct sun light to be approximately 53 $\mu\text{rem/hr}$.

These results were confirmed by the Radiation Protection Officer of the MIT EHS in a study he conducted in order to issue an operation permit for our x-ray microscope.

7.1.2. X-ray Source

Since the x-ray source was purchased pre-calibrated and it came with its own control software, we decided any further calibration was unnecessary. However, the cooling system and the power supply for the x-ray source were custom made, hence their performance had to be assessed.

According to Source Ray Inc., the x-ray source required less than 3 A at 24 VDC when operating at maximum output. We connected a current meter in series with the source and measured the current into it to be between 2.73 and 2.76 A with the output set to maximum. These measurements were repeated 6 times in 30 minutes, with the x-ray source constantly on and no change was noticed.

In parallel with the current measurements, the effectiveness of the cooling system was also evaluated. The maximum external operating temperature, specified by the manufacturer, was 35 °C and we had to ensure this temperature was not exceeded in operation. Two k-type thermocouple were attached to the outside of the x-ray source and the temperature was recorded every five minutes for a total of 30 min. The results from this test are shown in Figure 7.1. It looked like the steady-state temperature on the outside of x-ray source was between 26.3 and 26.5 °C with room temperature of 22.1 °C. Since the room temperature inside the laboratory does not change by more than 5 °C annually, we concluded the cooling was sufficient.

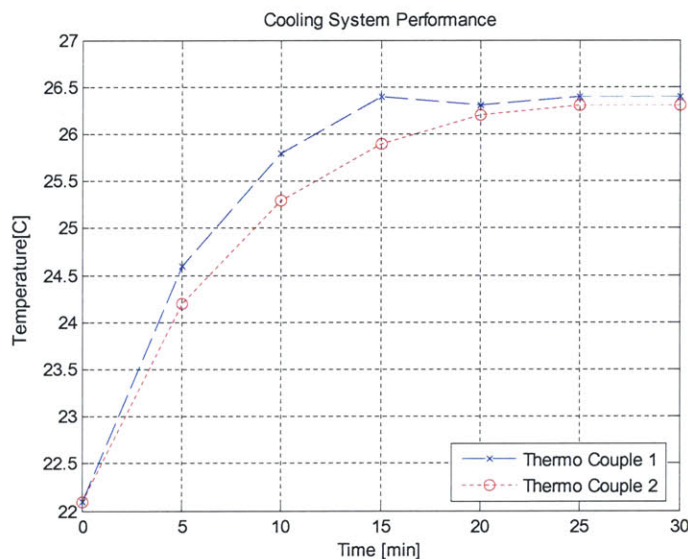


Figure 7.1: Results from the cooling system evaluation test.

7.1.3. Digital CCD Camera

The first test in the calibration of the digital camera was noise-sensitivity analysis of blank images. We began by examining the intensity spectrum of an image. A blank image was recorded with the x-ray source set at $V=40$ kV and $I=400$ μ A. After 556 μ s of exposure time, the pixel intensity distribution looked like the plot shown in Figure 7.2.

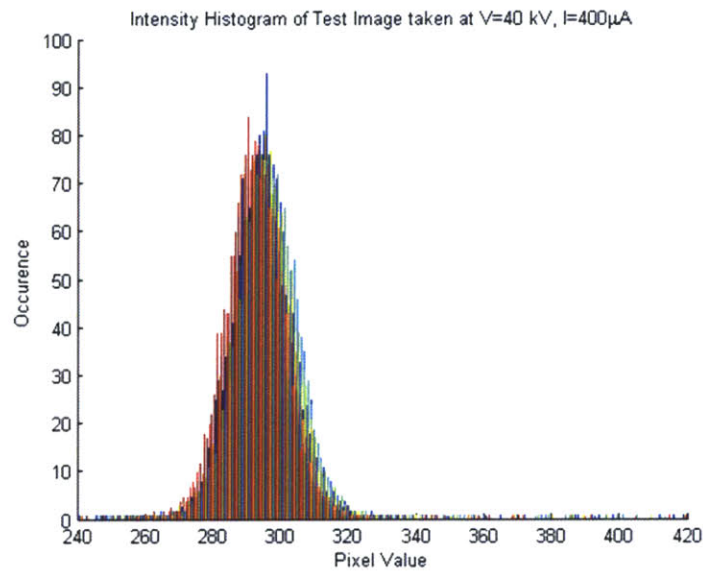


Figure 7.2: Image histogram

The distribution of the pixel intensity did not appear to be Gaussian, but was rather skewed to the right, therefore it could not be described by a mean and mean-square-error (MSE). We decided to compute the MSE for all samples greater than the mean and another MSE for those smaller than the mean. After repeating the experiment 7 times at the same voltage of 40 kV and the same exposure time of 556 μ s, a plot of the camera sensitivity and noise distribution was generated and is shown in Figure 7.3

The same experiment was repeated 7 more times at 35 kV and a similar plot was generated, shown in Figure 7.4. From the results it seemed like the noise distribution was uncorrelated with the voltage or the current settings, which meant the signal-to-noise ratio (SNR) increased as the overall pixel intensity increased.

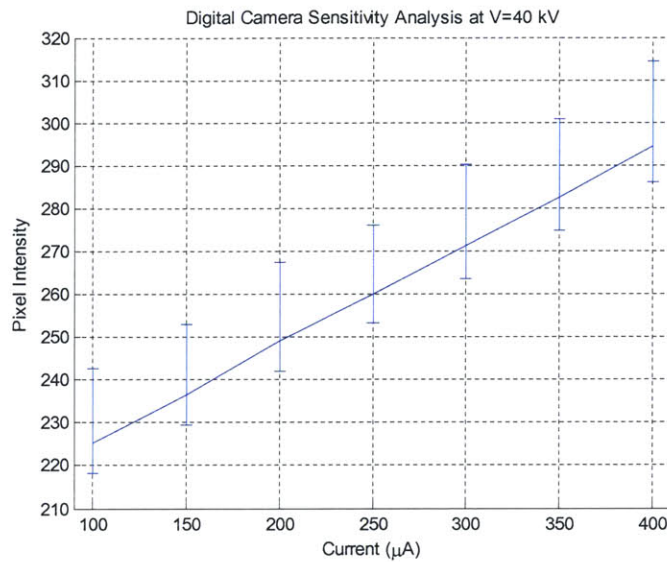


Figure 7.3: Noise distribution analysis for V=40 kV

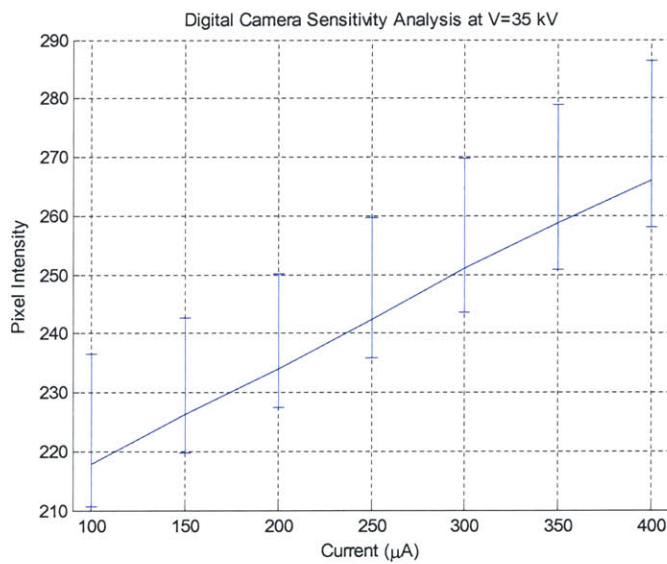


Figure 7.4: Noise distribution analysis for V=35 kV

The next step in the calibration of the digital camera was to determine its contrast resolving capabilities. In order to do this, a phantom was created, containing several different materials, each with distinct density. The phantom consisted of two multi-strand copper wires, coated individually in PVC and twisted together. The bunch was then wrapped around in paper and coated once again with PVC. Furthermore, a 26 gauge stainless steel hypodermic needle was inserted in-line with the wires, together with a

piece of 0.5 mm silver solder. A picture of this phantom is shown in Figure 7.5 (a), while the x-ray radiograph of it is shown in Figure 7.5 (b).

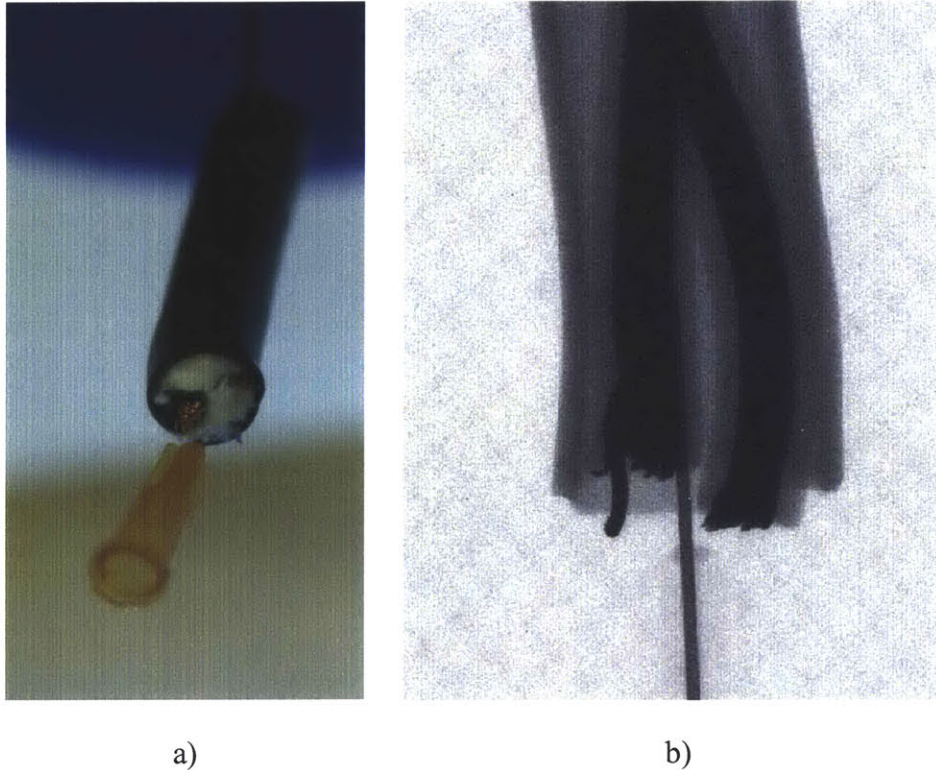


Figure 7.5: Contrast test phantom.
a) Picture of the phantom,
b) X-ray radiograph of the phantom.

Clear contrast difference between the metal, plastic, and free space is visible on the radiograph, however the contrast between the different metal objects is almost unnoticeable. This discovery was consistent with the general theory of x-ray radiography and no further attempts of contrast improvement were made.

The last step in the calibration of the digital camera was to determine its resolution. To do this, a phantom was created using several miniature objects. Three micro-drills, 50 μm , 100 μm , and 200 μm in diameter were embedded into the phantom together with a piece of 100 μm tungsten EDM wire and a piece of 250 μm copper EDM wire. A picture of the phantom is shown on Figure 7.6 (a). The radiograph, obtained by the camera, is shown in Figure 7.6 (b).

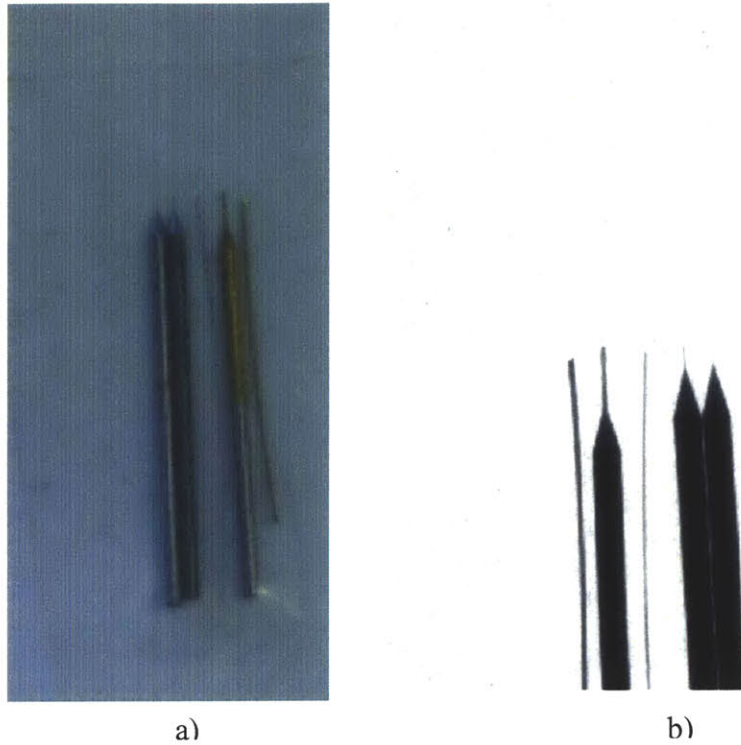


Figure 7.6: Resolution test phantom.
a) Picture of the phantom,
b) X-ray radiograph of the phantom.

The tip of the 50 μm drill bit was visible, however it was slightly blurred, which meant that the resolution of the camera was no-better than 25 μm and no worse than 50 μm . In fact, the width of the 50 μm drill bit was approximately 2 pixels, therefore we concluded the resolution was close to 25 μm .

7.1.4. Microscope Magnification

The last part of the microscope that had to be calibrated was its magnification. Unlike a conventional light microscope, the x-ray microscope's magnification comes from positioning the object closer to the x-ray source, as discussed in Section 4.3. Two parameters governed the magnification ratio of the microscope: the distance between the object and the source, x_1 , and the distance between the source and the camera, x_2 , and the magnification ratio, α , can be calculated as:

$$\alpha = \frac{x_2}{x_1}. \quad (7.1)$$

In our microscope configuration the distance between the source and any object could not be measured, since the vacuum tube was enclosed in a non-transparent enclosure. However, the distance between the camera and the object, d , could be measured easily and provided the size of the object, L_1 , and the magnified size of its projection, L_2 , we managed to calculate x_1 and x_2 as:

$$x_1 = \frac{L_1 d}{L_2 - L_1} \quad (7.2)$$

$$x_2 = d + x_1$$

Ultimately, we were interested in knowing x_2 , since it was the only fixed parameter and we could use it to determine the magnification ratio of any experimental configuration. We used once again the high-resolution phantom composed of the micro drills and we positioned it at a distance of $d = 230$ mm away from the camera. We acquired a magnified image, shown in Figure 7.7 (b) and compared it with the original image, shown in Figure 7.7 (a).

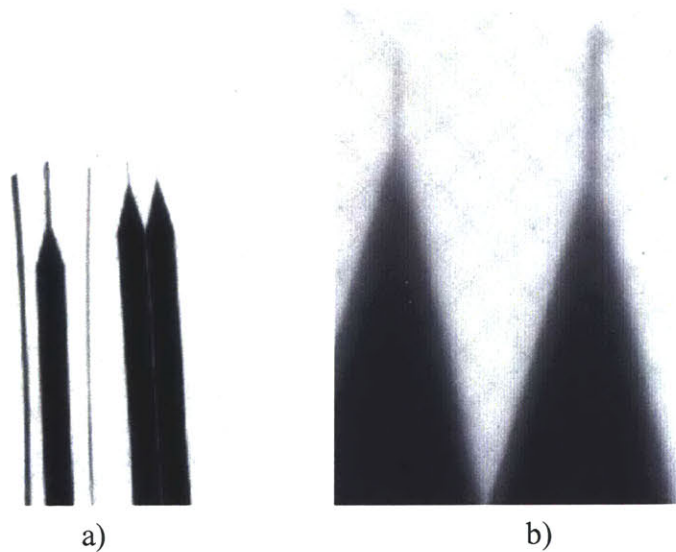


Figure 7.7: Magnification test.
a) Original radiograph, object placed at camera,
b) Magnified radiograph, object placed 230 mm away from camera.

From the radiograph we estimated the diameter of the magnified $50 \mu\text{m}$ drill bit to be approximately 11 pixels, corresponding to $275 \mu\text{m}$, resulting in 5.5x magnification. Using

these numbers we calculated the distance between the CCD and the focal spot of the x-ray source to be $x_2 = 281$ mm. With the current microscope configuration this was the only parameter required for the calculation of the magnification ratio of any future experiment.

7.2. Two-Dimensional Radiographs

Once the x-ray microscope was successfully calibrated we began imaging different objects with it. One of the first objects imaged with the completed and calibrated system was a dual 16-bit digital-to-analog converter microchip, DAC725JP, shown in Figure 7.8 (a). The radiograph of the microchip, shown in Figure 7.8 (b), clearly shows all internal features of the microchip, including the micro-wire connections.

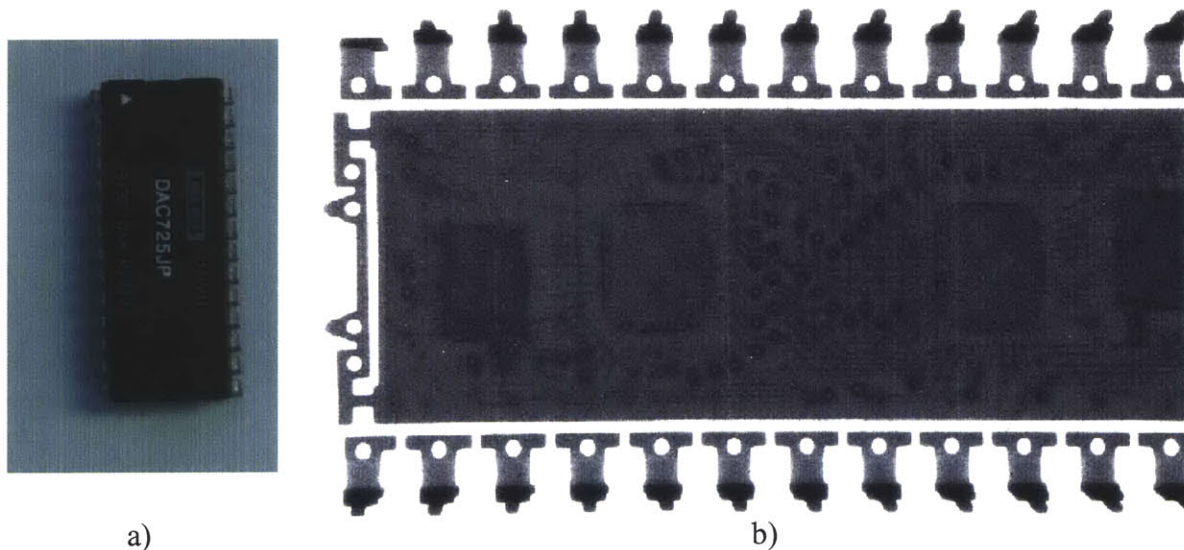


Figure 7.8: Images of DAC725JP microchip.
a) Picture of the microchip,
b) Radiograph of the microchip.

While working with electronic components we decided to also image a 256 MB Intelligent Stick miniature USB drive, shown in Figure 7.9 (a). The radiograph of the drive, shown in Figure 7.9 (b), clearly shows the layout of the device in high detail.

Additional images of various objects, studied with our x-ray microscope, can be found on the DVD supplied with this thesis.

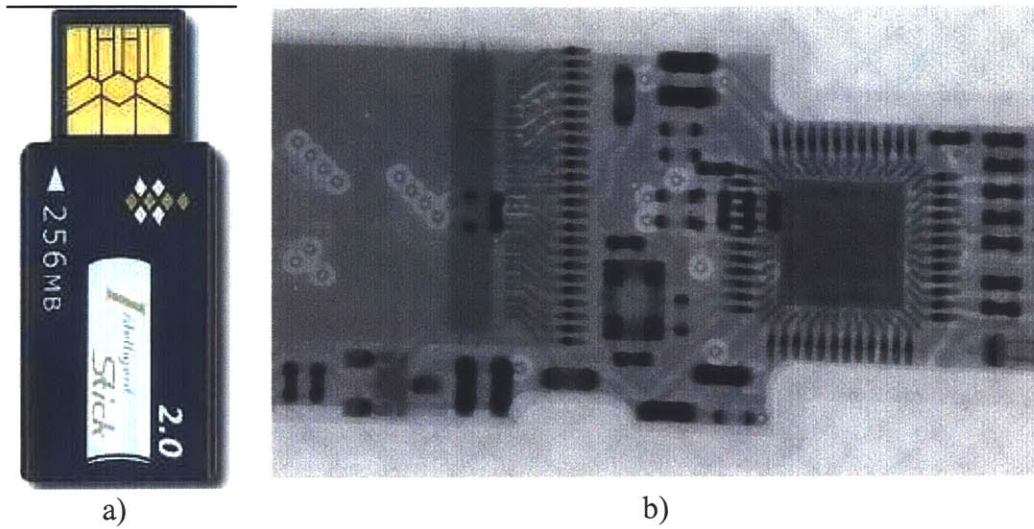


Figure 7.9: Images of 256 MB I-Stick USB Drive.
 a) Picture of the USB Drive,
 b) Radiograph of the USB Drive.

In addition to still images we captured several radiographic movies of various objects, operating inside non-transparent enclosures. All such movies are located on the DVD at the back of the thesis, while Figure 7.10 shows several key frames of an endoscopic probe, operating inside a non-transparent PTFE tube. The movie was recorded at 32 fps and shows the probe being manipulated inside the tube. The key frames show different stages in the opening of the probe.

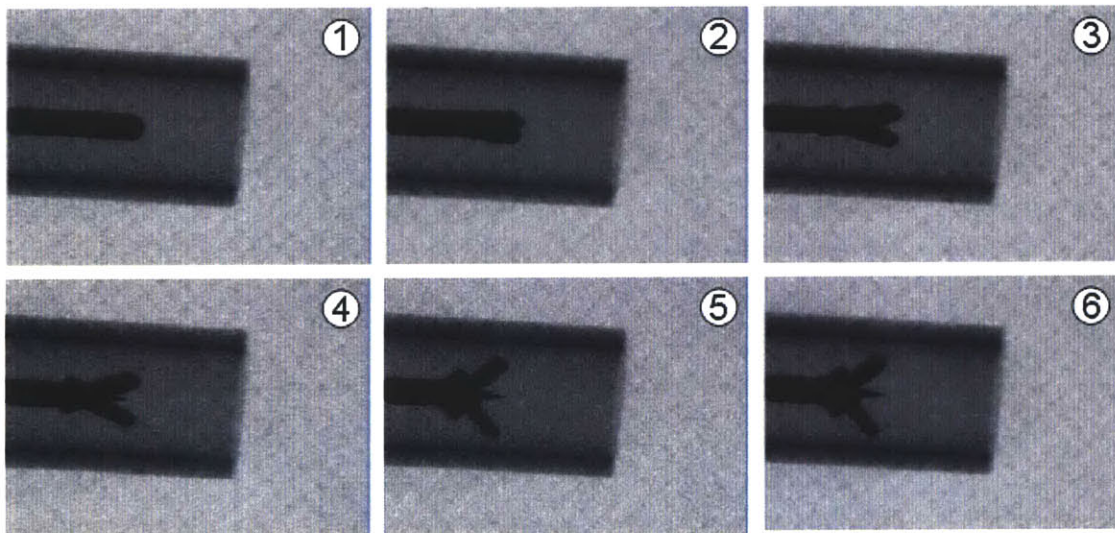


Figure 7.10: Endoscopic probe operating inside an opaque enclosure.

7.3. Conventional 3D Tomographic Images.

After all radiographic aspects of the microscope were studied, the focus was transferred towards its tomographic capabilities. Several phantoms were designed and used to study the performance and capabilities of the image reconstruction algorithms.

The first phantom we used was composed of a piece of foam with three hypodermic needles placed inside it, as shown in the radiograph in Figure 7.11. This was the simplest phantom with circularly symmetric objects and uniform density materials. Also, the foam was transparent to the x-rays, so the only objects we were reconstructing were the hypodermic needles.

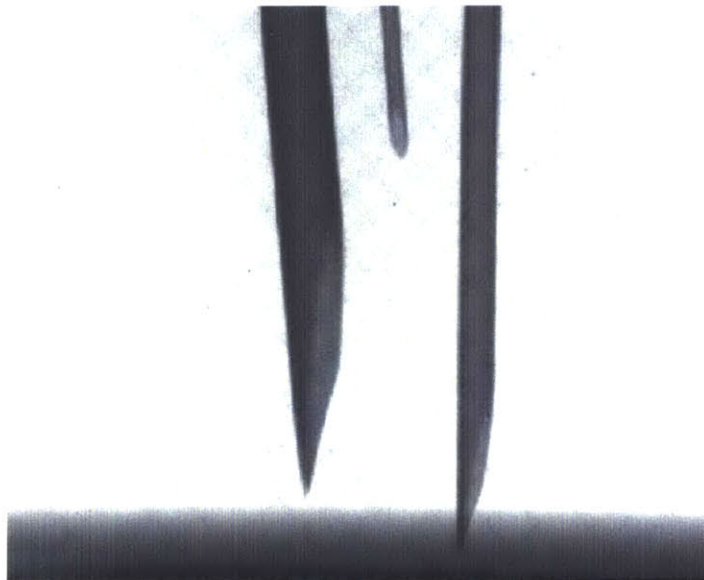


Figure 7.11: Radiography of simple cylindrical phantom.

Since the phantom was so simple we decided to take only 72 projection of it, spaced 5 degrees apart. The scanning time was approximately 5 minutes at a resolution of 640×512 pixels. The reconstruction of the perpendicular planes of the phantom required additional 3 to 4 minutes and a sample of the resulting images is shown in Figure 7.12. Clearly visible of this image are the streaks from the non-overlapping views, due to the low number of projections and also the axial offset, leading to drag marks around a fake center of rotation. We discovered that additional software alignment was required prior to image reconstruction in order to remove the axial misalignment and to regenerate the phantom successfully.

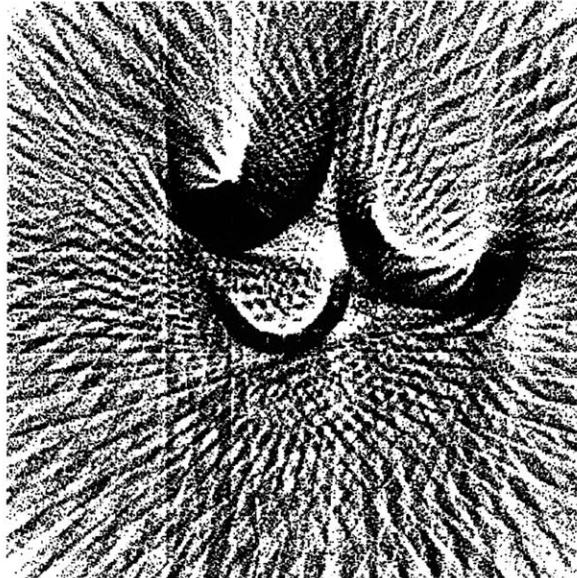


Figure 7.12: Initial reconstruction attempt of conventional CT image.

With the addition of software axial alignment and digital image post-processing we managed to reconstruct the perpendicular planes of the phantom. However, the results we obtained indicated that only the needle close to the axis of rotation was successfully reconstructed and the rest were simply blurred out, as shown in Figure 7.13. This was expected and was attributed to the fact that the density of reconstructed pixels decreases with increased distance from the center of rotation, as discussed in Section 4.1.4.

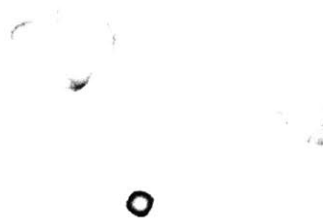


Figure 7.13: Improved reconstruction of conventional CT image of needle phantom.

The simplest solution to the blurring problem was to acquire more projections of the phantom. This would ensure the pixel density in the periphery of the reconstructed image

would be high enough, so that interpolation among those pixels could reconstruct, rather than blur, any details.

After discovering new information about the image reconstruction process, we decided to use a different phantom. It was important to study the penetrating capabilities of the microscope through solid metal, therefore the new phantom was mostly made from solid stainless steel, 2 to 3 mm thick. We used a stainless-steel lure lock fitting with a hypodermic needle attached to it, as shown in Figure 7.14 (a).

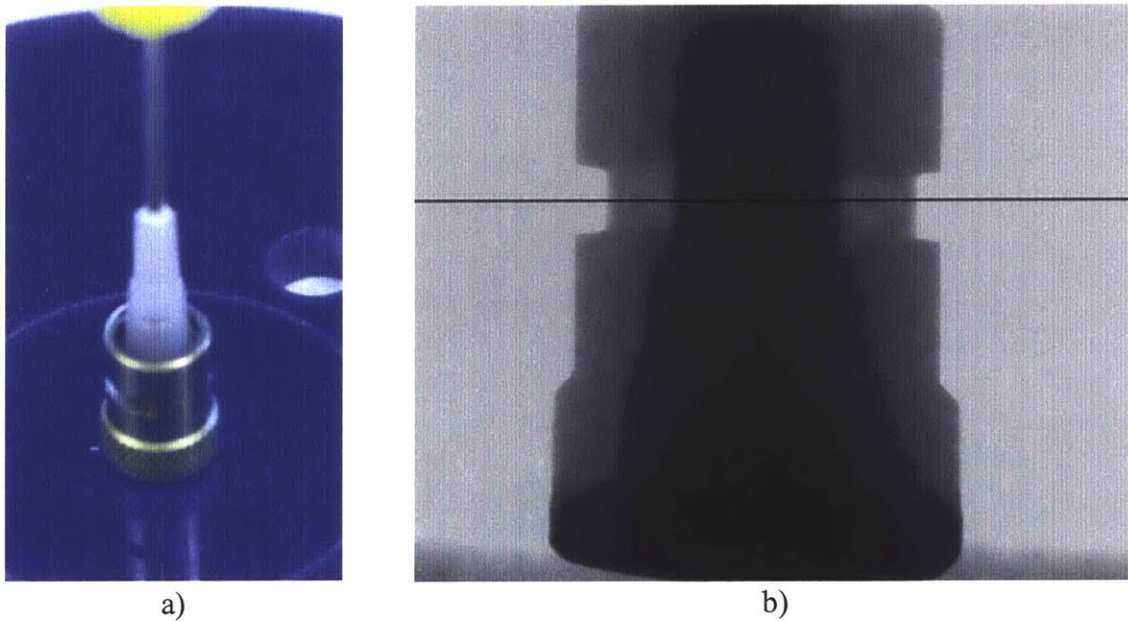


Figure 7.14: Solid stainless-steel phantom (lure-lock).
a) Picture of the phantom,
b) Radiograph of the phantom.

Using the new phantom a more detailed scan was performed, acquiring 180 images spaced 2 degrees apart. Figure 7.14 (b) shows a sample radiograph with solid black line indicating the reconstruction plane used for this paper. We used the modified reconstruction algorithm as described in Section 6.5.4.2 and obtained the cross-sectional image shown in Figure 7.15. From the reconstructed image it was clear that the reconstruction algorithm maintained the objects external aspect ratios, but could not reconstruct the uniform wall thickness. In the picture, thickening of the walls is clearly visible in places where the x-rays had to penetrate through more material. This was an unfortunate artifact, but we could not figure out an easy way to get rid of it.



Figure 7.15: Reconstructed conventional CT image of lure-lock phantom.

Once all the reconstruction details were worked out, we decided to image and study a more complex object. We acquired a miniature snail shell, shown in Figure 7.16 (a), and used it as a phantom. The shell was made of bone-like material and had non-uniform structure in all three dimensions.

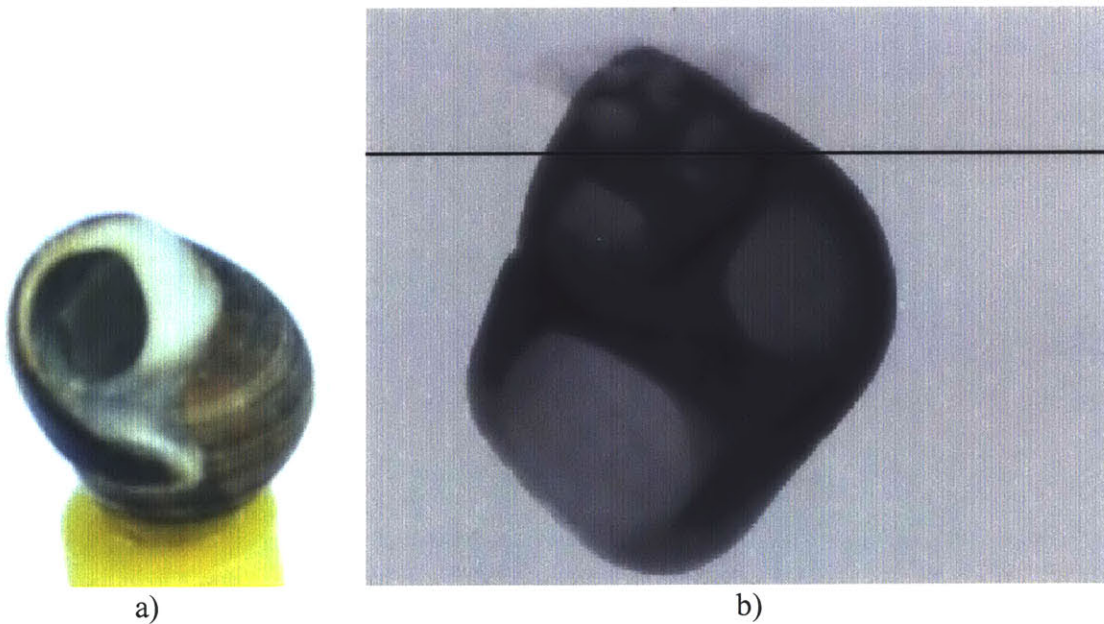


Figure 7.16: Snail shell phantom.
a) Picture of the phantom,
b) Radiograph of the phantom.

Complete scan was performed on the shell at a resolution of 1280×1024 and 180 projections were acquired, spaced 2 degrees apart. The entire scan required almost 20 minutes and over 1 GB of hard drive space. Upon completion of the scan, we discovered that the image data were too large to fit into the memory of the data processing computer, therefore we could not reconstruct the object from them.

A new scan was initiated with the resolution dropped to 320×256 , while maintaining the number of projections. The new data were much more manageable and the reconstruction could proceed as planned. Figure 7.17 shows 12 distinct layers of the shell, starting from layer number 70 (indicated by a black line in Figure 7.16 (b)) and proceeding to layer number 125, recording every fifth layer.



Figure 7.17: 12 reconstructed layers of the snail shell phantoms.

The results in Figure 7.17 show clearly increase in size of the crosssection, as expected from the object, as well as consistent “unwinding” of the internal helical structure. The solid external wall of the shell remains consistently denser than the more porous internal structure. No major reconstruction artifacts are visible on the reconstructed images, except the clipping on the last few images due to the size of the reconstruction field.

7.4. Volumetric 3D Tomographic images.

Using the conventional CT images generated from the snail shell and described in Section 7.3 we decided to reconstruct a volumetric image of the shell using the multiplanar reformation, discussed in Section 4.4.2. Starting at layer 70, indicated by the upper black line in Figure 7.18, and going through to layer 100, indicated by the lower black line in Figure 7.18, we generated 30 crosssectional images of the shell.

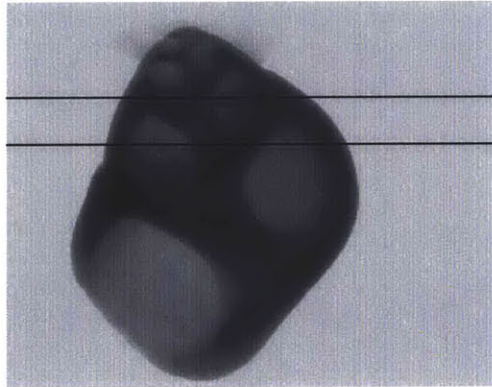


Figure 7.18: Radiograph of the snail shell phantom.

Once all the crosssectional images were generated, the volumetric reconstruction algorithm described in Section 6.5.4.3 was used to generate the images shown in Figure 7.19.

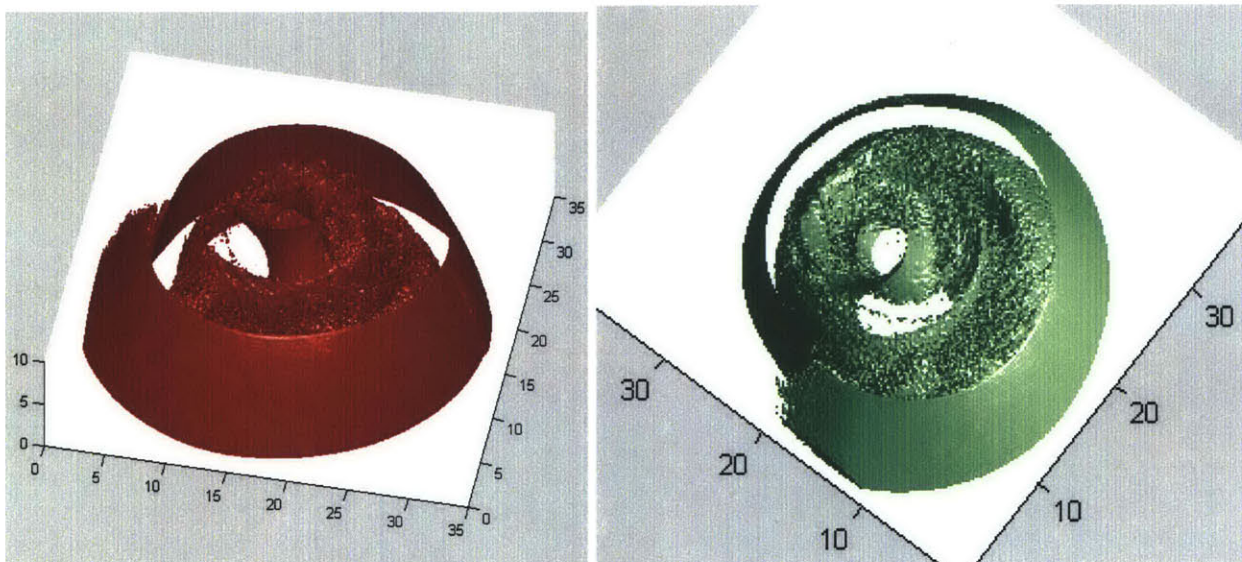


Figure 7.19: Volumetric 3D images of the snail shell phantom.

The reconstructed volumetric image showed sufficient level of detail and captured the general outline of the solid structure of the shell. The “unwinding” of the bone-like material on the inside and the outside of the shell is clearly visible. The denser outside surface is indicated by the solid color, while the more porous internal structures are filled with microscopic pores.

8. Conclusions and Future Work

The main goal set by this project was to design, develop, and test an environmentally friendly digital 3D x-ray microscope. We believe this goal was successfully achieved and the results obtained from the instrument indicate a wide range of applications in various experiments and studies.

During the design phase of the instrument a theory was laid out that predicted the behavior of the microscope and studied any potentially useful image enhancement techniques. Starting with the mathematical and physical fundamentals of the problem we worked our way up to the most advanced algorithms for regenerating and visualizing three-dimensional tomographic images.

The development of the microscope involved a lot of fabrication, manufacturing, and customization of equipment components and required the successful integration of mechanical components, electronics, and software control modules into a complete system. Safety guidelines and proper engineering etiquette guided us through the development and initial testing of the x-ray microscope.

Initial radiography results from the instrument were promising and exiting, allowing us to run series of experiments and study the internal structures of various objects. The high-speed capability of the system proved to be useful in recording the transient response of otherwise unobservable events.

Initially, the tomographic images were full of undesirable reconstruction artifacts and provided poor image quality. Nevertheless, improvements in the reconstruction techniques based on the image enhancement techniques studied during the design phase helped drastically improve the quality and reduce the noise in the reconstructed images. Ultimately we were able to precisely reconstruct the internal structure of opaque objects, regardless of the objects material composition or thickness.

The generation of the volumetric 3D images was the final stage of our project and it provided an exciting addition to an already functioning instrument and provided a path for future improvement and exciting new applications.

Since the x-ray microscope was developed as a prototype, many aspects of it were never optimized. One major aspect of the system that can easily be improved in the future would be an unification of all the software under a common application which can

synchronize the operation of the main functional components, provide better feedback on the status of the instrument and allow for simpler use and operation. Most of the drivers, application program interfaces, and software developer's kits that would be required for such a transition are included in the DVD supplied with this thesis. It is the author's opinion that if the entire system control, image acquisition, processing, and display are done in Visual Studio .NET the speed of the instrument can be improved by an order of magnitude.

Additional problems, faced during the processing of the huge data sets acquired with high-resolution scans were related to the poor memory management in MatLab and could potentially decrease the usefulness of the instrument in its current state for some high-quality 3D imaging applications. Nevertheless, most these issues could be resolved with better hardware, which will likely be available in the near future.

Bibliography

- [Batchelar, 2000] D. L. Batchelar, W. Dabrowski, and I.A. Cunningham. “Tomographic imaging of bone composition using coherently scattered x-rays.” *SPIE*, 397, pp. 353-361, Bellingham, WA, 2000.
- [Boca Radiology Group, 2006], <http://www.bocaradiology.com/cases/neuro/>. 2006.
- [Bose, 2004] T. Bose. “Digital signal and image processing.” *Willey & Sons, New York*, 2004.
- [Cosslett, 1960] V. E. Cosslett and W. C. Nixon “X-ray microscopy.” *Cambridge University Press*, 1960.
- [Dhawan, 2003] A. Dhawan “Medical Image Analysis.” *IEEE Press*, 2003.
- [EPM, 2006], <http://www.virtual.epm.br/material/tis/curr-med/med3/2003/ddi/matdid/cap2.htm>. 2006.
- [Feldkamp, 1984] L.A. Feldkamp, L.C. David, and J.W. Kress “Practical cone-beam algorithm.” *J. Opt. Soc. Am. A., Vol. 1, No. 6*, pp. 612-619, 1984.
- [Garratt-Reed, 2003] A. J. Garratt-Reed and D. C. Bell “Energy-dispersive x-ray analysis in the electron microscope.” *Oxford University Press*, 2003.
- [Gonzalez, 2002] R. C. Gonzalez “Digital image processing.” *Prentice Hall, NJ*, 2002.
- [Helgason, 1999] S. Helgason “The Radon transform.” 2nd Edition, *Birkhäuser*, 1999.
- [Herman, 1979] G. T. Herman and M. D. Altschuler “Image reconstruction from projections: implementation and applications.” *Willey & Sons, New York*, 1979.
- [Herman, 1980] G. T. Herman “Image reconstruction from projections: the fundamentals of computerized tomography.” *San Francisco Academic Press*, 1980.
- [Hilbert, 1952] D. Hilbert and S. Cohn-Vossen “Geometry and imagination.” *Chelsea Publishing Company*, 1952.

- [Hsieh, 2002] J. Hsieh “Computed Tomography: Principles, Design, and Recent Advances.” *SPIE, Washington*, 2002.
- [Imaginis, 2006], <http://imaginis.com/faq/history.asp>. 2006.
- [Jähne, 2002] B. Jähne “Digital image processing.” *5th revision, Prentice Hall, NJ*, 2002.
- [Kak, 2001] A. Kak, M. Slaney “Principles of Computerized Tomographic Imaging.” *SIAM, Philadelphia*, 2001.
- [McAndrew, 2004] A. McAndrew “Introduction to Digital Image Processing with MATLAB.” *Thomson, Course Technology, Boston*, 2004.
- [Natterer, 2001] F. Natterer “The mathematics of computerized tomography.” *John Wiley & Sons Ltd. and B.G. Teubner, Stuttgart*, 2001.
- [MQMD, 2006], <http://www.mtn.org/quack/devices/shoexray.htm>. 2006.
- [Oppenheim, 1996] A. V. Oppenheim, A. Willsky, S. H. Nawab, “Signals and Systems.” *2nd edition, Prentice Hall, NJ*, 1996.
- [Oppenheim, 1999] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck “Discrete-time signal processing.” *Prentice Hall, NJ*, 1999.
- [Radon, 1917] J. Radon “Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten.” *Ber. Verh. Sächs. Acad. Wiss. Leipzig. Math. Nat. Kl., Vol. 69*, pp. 262-277, 1917.
- [Ramapo College, 2006], <http://physics.ramapo.edu/~dbuna/ct.htm>. 2006.
- [Reinhard, 2004] K. Reinhard and A. Rosenfield “Digital geometry: geometric methods for digital picture analysis.” *Prentice Hall, NJ*, 2004.
- [SafetySwitch, 2006] <http://www.safetyswitch.net/>. 2006.
- [Scherz, 2000] P. Scherz “Practical electronics for inventors.” *McGraw-Hill*, 2000.

- [Schinazi, 2001] R. B. Schinazi "Probability with statistical applications." *Birkhäuser, Boston*, 2001.
- [Smith, 1985] B. D. Smith "Image reconstruction from cone-beam projections: necessary and sufficient conditions and reconstruction methods." *IEEE Trans. Med. Imag. Vol. 4*, pp. 14-25, 1985.
- [Stadtmiller, 2003] D. Stadtmiller "Applied electronic design.", *Prentice Hall, NJ*, 2003
- [Strang, 1998] G. Strang "Introduction to Linear Algebra." 2nd edition, *Wellesley-Cambridge Press*, 1998.
- [Suetens, 2002] P. Suetens "Fundamentals of medical imaging." *Cambridge University Press*, 2002.
- [Tuy, 1983] H. K. Tuy "An inversion formula for cone-beam reconstruction." *SIAM J. Appl. Math. Vol. 43*, pp. 546-552, 1983.
- [Webb, 2003] A. Webb "Introduction to Biomedical Imaging." *IEEE Press, NJ*, 2003.
- [Westmore, 1997] M. S. Westmore, A. Frenster, and I.A. Cunningham. "Tomographic imaging of the angular-dependent coherent-scatter cross section." *Med. Phys. 24(1)*, pp. 3-10, 1997.
- [Wikipedia, 2006], <http://en.wikipedia.org/wiki/X-ray>. 2006.

Appendix A: Equipment Specifications

Appendix A.1: Camera Specifications

Part number: C4742-56-12NR

Incorporating a large area scintillator coated fiber taper, the C4742-56 is an excellent choice for obtaining high resolution, high magnification X-ray images. Hamamatsu's outstanding ORCA series cooled digital camera is the integral readout component of the C4742-56 allowing it to detect even the faintest signal.

Key Specifications

Name of product	ORCA X-ray Digital Camera
Pixel clock rate Scan Mode 1	14,75
A/D converter Scan Mode 2	12
Cooling Method	passive
Output type	RS-422
Contrast Enhancement	yes
Sensitivity Range - min	10 KV
Sensitivity Range - max	100 KV
Pixels	1344x1024
Detection Method	GdO2S:Tb
Resolution	22/43/86 μ m
Speed	9/30 fps

Features

- 3.7:1 fiber taper ratio
- 1280x1024 resolution
- P43 (GdO2S:Tb) scintillator
- 9 fps full resolution frame rate, 30 fps at 1280 x 432 sub-array resolution
- On-chip integration, binning and sub-array features
- Low 8e- readout noise and passive peltier cooling allows exposures to 10 seconds
- 12-bit digital output, greater than 10-bit dynamic range
- RS-422 digital output/RS-232 control
Electronic shutter down to 130 μ sec
- On-chip integration and binning
- Digitally controlled analog contrast enhancement
- ORCA compatible software control

Applications

- Food inspection
- Automotive applications
- Metal and other foreign matter detection
- Package item counting
- X-ray CT
- Electronics Inspection

Appendix A.2: X-ray Source Specifications

Manufacturer	<u>Source-Ray Inc.</u>
Product Name	X Ray Source
Product Model Number	SB-80-500
Type	Medical
Energy Range Min	35 kV
Energy Range Max	80 kV
Focal Spot Size	50 micron
Duty Cycle	CW, pulsed
Power Supply	24 V
Min Operating Temperature	10 Celsius
Max Operating Temperature	35 Celsius
Operating Conditions	DC supply voltage, peak value of energy range
Other Attributes	Storage temperature is 0-45 degree
Features	Low inherent filtration, high power output
Application	Used in NDT or industrial X-ray analysis, medical X-ray (BMD, radiographic and fluoroscopic imaging) and EDS

Appendix A.3: Power Supply Specifications



ABL7RE and ABL7RP Single Phase Switch Mode Power Supplies

The **ABL7RE** and **ABL7RP** single phase regulated switching power supplies are designed to provide the dc voltage necessary for most control system circuits. These power supplies are totally electronic and regulated. The use of electronics makes it possible to significantly improve the performance of these power supplies. They provide:

- Compact size
- Integrated overload, short circuit, over voltage and under voltage protection
- A wide range of permitted input voltages (100–240 Vac), without any adjustment
- A high degree of output voltage stability
- Good performance
- Considerably reduced weight in comparison to competition 1.15–4.85 lb. (0.52–2.19 kg)

These power supplies, which are accurate to within $\pm 3\%$ regardless of the load or the type of supply voltage, within the range of 85–264 Vac. The inclusion of overload and short circuit protection makes downstream protection unnecessary in most applications. They have an output under voltage control which will cause the power supply to trip if the output voltage of a 24 Vdc supply drops below 19 Vdc, to ensure that the voltage is always usable by the devices being supplied. All products have an output voltage adjustment potentiometer (24 to 28.8 V on a 24 Vdc supply) which allows for compensation of any line voltage drop in installations with long runs.

The **ABL7RE** products are excellent for typical industrial applications.

The **ABL7RP** products are for general purpose applications. These supplies have an input filter (Power Factor Correction or PFC) which allows them to be used in commercial environments. They have 2 operating modes:

“**AUTO**” mode which automatically restarts as soon as the fault is cleared.

“**MANU**” mode which requires the power supply to be reset before restarting is possible. Resetting is achieved by switching off the supply voltage and reapplying it.

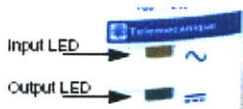
Harmonics (Power Factor) – The current drawn by a power supply is non-sinusoidal. This leads to the existence of harmonic currents which pollute the main supply. European standard EN 61000-3-2 limits the harmonic currents produced by power supplies. This standard covers all devices of more than 75 W, drawing up to 16 A per phase, and connected directly to the utilities. Devices connected downstream of a private, low voltage, general transformer are excluded. However, switching power supplies produce harmonic current. Therefore, a filter circuit (PFC) must be added to the circuit when using an **ABL7RE** power supply to comply with standard EN 61000-3-2. The **ABL7RP** power supplies also conform to the EN 61000-3-2 standard and can be connected directly to the public power supply system.

Input LED—When illuminated, this LED indicates the supply voltage is present.

Output LED—This LED indicates if a fault has occurred.

Steady: normal voltage out

Flashing: overload or short circuit



Appendix B: Software

Appendix B.1: Source Code for Rotary Stage Control

```
-----Form1.vb-----
Public Class Form1
    Inherits System.Windows.Forms.Form

    Private WithEvents moRS232 As Rs232
    Private t As New System.Timers.Timer()
    Dim n As Integer

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents Button2 As System.Windows.Forms.Button
    Friend WithEvents Button3 As System.Windows.Forms.Button
    Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents TextBox2 As System.Windows.Forms.TextBox
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents Label4 As System.Windows.Forms.Label
    Friend WithEvents Label5 As System.Windows.Forms.Label
    Friend WithEvents Label6 As System.Windows.Forms.Label
    Friend WithEvents TextBox3 As System.Windows.Forms.TextBox
    Friend WithEvents Label7 As System.Windows.Forms.Label
    Friend WithEvents TextBox4 As System.Windows.Forms.TextBox
    Friend WithEvents Timer1 As System.Windows.Forms.Timer
    Friend WithEvents Button4 As System.Windows.Forms.Button
    <System.Diagnostics.DebuggerStepThrough(> Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container
        Me.Button1 = New System.Windows.Forms.Button
        Me.Button2 = New System.Windows.Forms.Button
        Me.Button3 = New System.Windows.Forms.Button
        Me.TextBox1 = New System.Windows.Forms.TextBox
        Me.Label1 = New System.Windows.Forms.Label
        Me.Label2 = New System.Windows.Forms.Label
        Me.TextBox2 = New System.Windows.Forms.TextBox
        Me.Label3 = New System.Windows.Forms.Label
        Me.Label4 = New System.Windows.Forms.Label
        Me.Button4 = New System.Windows.Forms.Button
```

```

Me.Label5 = New System.Windows.Forms.Label
Me.Label6 = New System.Windows.Forms.Label
Me.TextBox3 = New System.Windows.Forms.TextBox
Me.Label7 = New System.Windows.Forms.Label
Me.TextBox4 = New System.Windows.Forms.TextBox
Me.Timer1 = New System.Windows.Forms.Timer (Me.components)
Me.SuspendLayout()
'
'Button1
'
Me.Button1.Location = New System.Drawing.Point(232, 24)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(112, 32)
Me.Button1.TabIndex = 0
Me.Button1.Text = "Setup"
'
'Button2
'
Me.Button2.Enabled = False
Me.Button2.Location = New System.Drawing.Point(352, 24)
Me.Button2.Name = "Button2"
Me.Button2.Size = New System.Drawing.Size(112, 32)
Me.Button2.TabIndex = 1
Me.Button2.Text = "Go"
'
'Button3
'
Me.Button3.Enabled = False
Me.Button3.Location = New System.Drawing.Point(352, 64)
Me.Button3.Name = "Button3"
Me.Button3.Size = New System.Drawing.Size(112, 32)
Me.Button3.TabIndex = 2
Me.Button3.Text = "Stop"
'
'TextBox1
'
Me.TextBox1.Location = New System.Drawing.Point(120, 40)
Me.TextBox1.Name = "TextBox1"
Me.TextBox1.Size = New System.Drawing.Size(72, 20)
Me.TextBox1.TabIndex = 3
Me.TextBox1.Text = "15"
'
'Label1
'
Me.Label1.Location = New System.Drawing.Point(192, 32)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(16, 24)
Me.Label1.TabIndex = 4
Me.Label1.Text = "o"
'
'Label2
'
Me.Label2.Location = New System.Drawing.Point(16, 39)
Me.Label2.Name = "Label2"
Me.Label2.Size = New System.Drawing.Size(100, 23)
Me.Label2.TabIndex = 5
Me.Label2.Text = "Angle of rotation"
Me.Label2.TextAlign = System.Drawing.ContentAlignment.MiddleRight
'
'TextBox2
'
Me.TextBox2.Location = New System.Drawing.Point(120, 72)
Me.TextBox2.Name = "TextBox2"
Me.TextBox2.Size = New System.Drawing.Size(72, 20)
Me.TextBox2.TabIndex = 6
Me.TextBox2.Text = "3"
'
'Label3
'
Me.Label3.Location = New System.Drawing.Point(192, 79)
Me.Label3.Name = "Label3"

```

```

Me.Label3.Size = New System.Drawing.Size(16, 24)
Me.Label3.TabIndex = 7
Me.Label3.Text = "s"
'
'Label4
'
Me.Label4.Location = New System.Drawing.Point(16, 71)
Me.Label4.Name = "Label4"
Me.Label4.Size = New System.Drawing.Size(100, 23)
Me.Label4.TabIndex = 8
Me.Label4.Text = "Time per move"
Me.Label4.TextAlign = System.Drawing.ContentAlignment.MiddleRight
'
'Button4
'
Me.Button4.Enabled = False
Me.Button4.Location = New System.Drawing.Point(232, 64)
Me.Button4.Name = "Button4"
Me.Button4.Size = New System.Drawing.Size(112, 32)
Me.Button4.TabIndex = 13
Me.Button4.Text = "Reset"
'
'Label5
'
Me.Label5.Location = New System.Drawing.Point(5, 105)
Me.Label5.Name = "Label5"
Me.Label5.Size = New System.Drawing.Size(111, 23)
Me.Label5.TabIndex = 16
Me.Label5.Text = "Time between moves"
Me.Label5.TextAlign = System.Drawing.ContentAlignment.MiddleRight
'
'Label6
'
Me.Label6.Location = New System.Drawing.Point(192, 113)
Me.Label6.Name = "Label6"
Me.Label6.Size = New System.Drawing.Size(16, 24)
Me.Label6.TabIndex = 15
Me.Label6.Text = "s"
'
'TextBox3
'
Me.TextBox3.Location = New System.Drawing.Point(120, 106)
Me.TextBox3.Name = "TextBox3"
Me.TextBox3.Size = New System.Drawing.Size(72, 20)
Me.TextBox3.TabIndex = 14
Me.TextBox3.Text = "5"
'
'Label7
'
Me.Label7.Location = New System.Drawing.Point(16, 140)
Me.Label7.Name = "Label7"
Me.Label7.Size = New System.Drawing.Size(100, 23)
Me.Label7.TabIndex = 19
Me.Label7.Text = "Number of moves"
Me.Label7.TextAlign = System.Drawing.ContentAlignment.MiddleRight
'
'TextBox4
'
Me.TextBox4.Location = New System.Drawing.Point(120, 141)
Me.TextBox4.Name = "TextBox4"
Me.TextBox4.Size = New System.Drawing.Size(72, 20)
Me.TextBox4.TabIndex = 17
Me.TextBox4.Text = "24"
'
'Form1
'
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.ClientSize = New System.Drawing.Size(469, 269)
Me.Controls.Add(Me.Label7)
Me.Controls.Add(Me.TextBox4)
Me.Controls.Add(Me.Label5)

```

```

Me.Controls.Add(Me.Label6)
Me.Controls.Add(Me.TextBox3)
Me.Controls.Add(Me.Button4)
Me.Controls.Add(Me.Label4)
Me.Controls.Add(Me.Label3)
Me.Controls.Add(Me.TextBox2)
Me.Controls.Add(Me.Label2)
Me.Controls.Add(Me.Label1)
Me.Controls.Add(Me.TextBox1)
Me.Controls.Add(Me.Button3)
Me.Controls.Add(Me.Button2)
Me.Controls.Add(Me.Button1)
Me.Name = "Form1"
Me.Text = "Rotary Stage Control"
Me.ResumeLayout(False)
Me.PerformLayout()

End Sub

#End Region
Private Sub Generate_and_Send_Code()
    Dim sTx As String
    Dim velocity As String
    Dim distance As String
    Dim stepsrev As Integer      'stepsrev is equal to the resolution in steps / rev
of the motor
    Dim vel As Double           'TextBox1 is the velocity
    Dim dis As Double           'TextBox2 is the time
    Dim time As Double          'hours as a double
    Dim GearRatio As Double
    '-----
    'Setup program parameters
    stepsrev = 50000
    GearRatio = 22.5
    dis = Convert.ToDouble(TextBox1.Text) * GearRatio / 360 * stepsrev 'in steps
    time = Convert.ToDouble(TextBox2.Text) 'Assigning a string to a double
    'Number of Steps per revolution
    vel = dis / time / stepsrev * 2
    velocity = "V" & vel
    distance = "D" & dis
    '// Clear Tx/Rx Buffers
    moRS232.PurgeBuffer(Rs232.PurgeBuffers.TxClear Or Rs232.PurgeBuffers.RXClear)
    'Write Program
    sTx = "DEL PROG1"           'Line1
    sTx += ControlChars.Cr
    moRS232.Write(sTx)

    sTx = "DEF PROG1"          'Line2
    sTx += ControlChars.Cr
    moRS232.Write(sTx)

    sTx = "DRIVE1"             'Line3
    sTx += ControlChars.Cr
    moRS232.Write(sTx)

    sTx = "VAR1=1PE"           'Lined1
    sTx += ControlChars.Cr
    moRS232.Write(sTx)

    sTx = velocity              'Line4
    sTx += ControlChars.Cr
    moRS232.Write(sTx)

    sTx = distance              'Line5
    sTx += ControlChars.Cr
    moRS232.Write(sTx)

    sTx = "GO"                  'Line6
    sTx += ControlChars.Cr
    moRS232.Write(sTx)

```

```

sTx = "VAR2=1PE"           'Line2
sTx += ControlChars.Cr
moRS232.Write(sTx)

sTx = "END"               'Line7
sTx += ControlChars.Cr
moRS232.Write(sTx)
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
moRS232 = New Rs232
Try
    '// Setup parameters
    With moRS232
        .Port = 2
        .BaudRate = 9600
        .DataBit = 8
        .StopBit = Rs232.DataStopBit.StopBit_1
        .Parity = Rs232.DataParity.Parity_None
        .Timeout = 1500
    End With
    '// Initializes port
    moRS232.Open()
    '// Set state of RTS / DTS
    moRS232.Dtr = True
    moRS232.Rts = True
Catch Ex As Exception
    MessageBox.Show(Ex.Message, "Connection Error", MessageBoxButtons.OK)
Finally
    Button1.Enabled = Not moRS232.IsOpen
    Button2.Enabled = moRS232.IsOpen
    Button3.Enabled = moRS232.IsOpen
    Button4.Enabled = moRS232.IsOpen
End Try
Generate_and_Send_Code()
End Sub

Private Sub Form1_Closed(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Closed
If moRS232.IsOpen Then
    t.Enabled = False
    n = 0
    moRS232.Close()
End If
Button1.Enabled = Not moRS232.IsOpen
Button2.Enabled = moRS232.IsOpen
Button3.Enabled = moRS232.IsOpen
Button4.Enabled = moRS232.IsOpen
End Sub

Private Sub TimerFired(ByVal sender As Object, ByVal e As
System.Timers.ElapsedEventArgs)
Dim sTx As String
If n > 0 Then
    n = n - 1
Else
    t.Enabled = False
    Exit Sub
End If
sTx = "PROG1"           'Start PROG1
sTx += ControlChars.Cr
moRS232.Write(sTx)
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
Dim delay As Integer
n = Convert.ToInt16(TextBox4.Text)
delay = Convert.ToInt16(TextBox3.Text) + Convert.ToInt16(TextBox2.Text)
t.Interval = delay * 1000
AddHandler t.Elapsed, AddressOf TimerFired

```

```

        t.Enabled = True
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        Dim sTx As String
        t.Enabled = False
        n = 0
        sTx = "!K"           'ABORT Process
        sTx += ControlChars.Cr
        moRS232.Write(sTx)
        sTx = "VAR2=1PE"     'ABORT Process
        sTx += ControlChars.Cr
        moRS232.Write(sTx)
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
        If moRS232.IsOpen Then
            t.Enabled = False
            n = 0
            moRS232.Close()
        End If
        Button1.Enabled = Not moRS232.IsOpen
        Button2.Enabled = moRS232.IsOpen
        Button3.Enabled = moRS232.IsOpen
        Button4.Enabled = moRS232.IsOpen
    End Sub
End Class
-----Form1.vb-----

```


Appendix B.2: Image Enhancement GUI

```
-----GUI_1.m-----
function varargout = GUI_1(varargin)
% GUI_1 M-file for GUI_1.fig
% GUI_1, by itself, creates a new GUI_1 or raises the existing
% singleton*.
%
% H = GUI_1 returns the handle to a new GUI_1 or the handle to
% the existing singleton*.
%
% GUI_1('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in GUI_1.M with the given input arguments.
%
% GUI_1('Property','Value',...) creates a new GUI_1 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before GUI_1_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to GUI_1_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_1

% Last Modified by GUIDE v2.5 12-Sep-2005 08:41:16

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @GUI_1_OpeningFcn, ...
                  'gui_OutputFcn',  @GUI_1_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI_1 is made visible.
function GUI_1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_1 (see VARARGIN)

% Choose default command line output for GUI_1
handles.output = hObject;
handles.I = [];
handles.res = [];
handles.cd = cd;
set(handles.slider_UL, 'Value', 1);
set(handles.slider_LL, 'Value', 0);
handles.csUL = 1;
handles.csLL = 0;
a = round(handles.csLL*255);
b = round((1-handles.csUL)*255);
handles.csMAP = [zeros(1,a),...
                [0:1/(255-a-b):1],...
                ones(1,b)]*255;
```

```

axes(handles.axes_cs);
plot([0:255],handles.csMAP);
axis([0,255,0,255]);

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_1_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton_load.
function pushbutton_load_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton_load (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

cd(handles.cd);
[filename, pathname] = uigetfile( { '*.jpg', 'Jpeg file (*.jpg)'; ...
    '*.bmp', 'Bitmap file (*.bmp)'; ...
    '*.tif', 'Tiff file (*.tif)'}; , ...
    'Pick an image');
handles.I = imread([pathname, filename]);
handles.I = rgb2gray(handles.I);
handles.res = handles.I;
axes(handles.axes_original);
imshow(handles.I);
axes(handles.axes_modified);
imshow(handles.res);
axes(handles.axes_hist);
[C,X]=imhist(handles.res,256);
stem(X(2:255),C(2:255), 'Marker','none');
set(handles.edit_fileName, 'String', [pathname, filename]);
set(handles.edit_save, 'String', [filename(1:size(filename,2)-4), '_M.bmp']);
handles.cd = pathname;

guidata(hObject, handles);

function edit_fileName_Callback(hObject, eventdata, handles)
% hObject handle to edit_fileName (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_fileName as text
% str2double(get(hObject,'String')) returns contents of edit_fileName as a double

% --- Executes during object creation, after setting all properties.
function edit_fileName_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit_fileName (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider2_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to slider2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider_UL_Callback(hObject, eventdata, handles)
% hObject    handle to slider_UL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of slider
handles.csUL = get(handles.slider_UL,'Value');
if handles.csUL<=handles.csLL
    handles.csLL = handles.csUL-0.01;
    set(handles.slider_UL,'Value',handles.csUL);
    set(handles.slider_LL,'Value',handles.csLL);
end
a = round(handles.csLL*255);
b = round((1-handles.csUL)*255);
handles.csMAP = [zeros(1,a),...
    [0:1/(255-a-b):1],...
    ones(1,b)]*255;
axes(handles.axes_cs);
plot([0:255],handles.csMAP);
axis([0,255,0,255]);
guidata(hObject, handles);

redraw(hObject,eventdata,handles);

% --- Executes during object creation, after setting all properties.
function slider_UL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_UL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider_LL_Callback(hObject, eventdata, handles)
% hObject    handle to slider_LL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.csLL = get(handles.slider_LL,'Value');
if handles.csLL>=handles.csUL
    handles.csUL = handles.csLL+0.01;
    set(handles.slider_LL,'Value',handles.csLL);
    set(handles.slider_UL,'Value',handles.csUL);
end
a = round(handles.csLL*255);
b = round((1-handles.csUL)*255);
handles.csMAP = [zeros(1,a),...
    [0:1/(255-a-b):1],...
    ones(1,b)]*255;

```

```

axes(handles.axes_cs);
plot([0:255],handles.csMAP);
axis([0,255,0,255]);
guidata(hObject, handles);

redraw(hObject,eventdata,handles);

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes on button press in checkbox_HE.
function checkbox_HE_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_HE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
redraw(hObject,eventdata,handles);

% Hint: get(hObject,'Value') returns toggle state of checkbox_HE

% --- Executes on button press in pushbutton_AC.
function pushbutton_AC_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_AC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.I = handles.res;
axes(handles.axes_original);
imshow(handles.I);

guidata(hObject, handles);

% --- Executes on button press in checkbox_UM.
function checkbox_UM_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_UM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
redraw(hObject,eventdata,handles);
% Hint: get(hObject,'Value') returns toggle state of checkbox_UM

% --- Executes on button press in checkbox_MF.
function checkbox_MF_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_MF (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
redraw(hObject,eventdata,handles);
% Hint: get(hObject,'Value') returns toggle state of checkbox_MF

function redraw(hObject, eventdata, handles)
% hObject    handle to checkbox_MF (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.res = handles.I;
handles.res = imadjust(handles.I, [handles.csLL handles.csUL], [0 1]);
if get(handles.checkbox_HE,'Value')
    handles.res = histeq(handles.res);
end
if get(handles.checkbox_UM,'Value')
    G = fspecial('unsharp');
    handles.res = imfilter(handles.res,G);
end
if get(handles.checkbox_MF,'Value')
    handles.res = medfilt2(handles.res);
end
axes(handles.axes_modified);
imshow(handles.res);
[C,X]=imhist(handles.res,256);
axes(handles.axes_hist);
stem(X(2:255),C(2:255), 'Marker', 'none');

guidata(hObject, handles);

% --- Executes on button press in pushbutton3.

```

```

function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
A = uint8(handles.res);
imwrite(A,get(handles.edit_save,'String'),'bmp');

function edit_save_Callback(hObject, eventdata, handles)
% hObject    handle to edit_save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_save as text
%        str2double(get(hObject,'String')) returns contents of edit_save as a double

% --- Executes during object creation, after setting all properties.
function edit_save_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
-----GUI_1.m-----

```

Appendix B.3: Source Code for Conventional CT Reconstruction

```

-----Main.m-----
M = 320;%1280;
M1 = 271;
N = 256;%1024;
lineR = [70:100]; %(<=N)
pixelsize = 40/M; % in mm
x = 10;
D = (82+x)/pixelsize;
NA = 180;
%---
% A = zeros(NA,N,M);
% for i = 0:NA-1
%     nums = num2str(i);
%     if size(nums,2)==1
%         nums = strcat('00',nums);
%     end;
%     if size(nums,2)==2
%         nums = strcat('0',nums);
%     end;
%     tmp = strcat('Shell',nums ,'.DAT');
%     IMG = load(tmp);
%     A(i+1, :, :) = IMG;
%     i
% end;
%---
for line = lineR
    a = zeros(NA,M1);
    if M1<=M
        for i = 1:NA
            for j=1:M1
                a(i,j) = A(i,line,j);
            end;
        end;
    else
        for i = 1:NA
            for j=1:M
                a(i,j) = A(i,line,j);
            end;
        end;
    end;
    I = ifanbeam(a',D,'FanSensorGeometry' , 'line','Filter','Hamming');
    %I = iradon(a',[0:5:355],'pchip','Hann');
    IM3D(line, :, :) = Iadjust(I,M1,line);
    %line
end;
-----Main.m-----
-----Iadjust.m-----

function [res]=Iadjust(I,M1,line)
%Ifb = ifanbeam(a',641,'FanSensorGeometry' , 'line','FanRotationIncrement',2);
Image = I;
LT = -6;
UT = -6;
SF = 255/(UT-LT);
for i=1:size(Image,1)
    for j=1:size(Image,2)
        if Image(i,j)<LT
            Image(i,j) = 0;
        else
            if Image(i,j)>UT
                Image(i,j) = 255;
            else
                Image(i,j) = (Image(i,j)-LT)*SF;
            end
        end
    end
end
end
end

```



```

% ImageF = Ifb;
% LTF = -70;
% UTF = 120;
% SFF = 255/(UTF-LTF);
% for i=1:size(ImageF,1)
%     for j=1:size(ImageF,2)
%         if ImageF(i,j)<LT
%             ImageF(i,j) = 0;
%         else
%             if ImageF(i,j)>UT
%                 ImageF(i,j) = 255;
%             else
%                 ImageF(i,j) = (ImageF(i,j)-LTF)*SFF;
%             end
%         end
%     end
% end
%N = load(tmp);
%N(line,:) = zeros(1,M);
%subplot(1,2,1);
Image = uint8(Image);
imshow(Image);
TMPI = 255*ones(400,400);
dx = (size(TMPI,1)-size(Image,1))/2;
dy = dx;
for i=1:size(Image,1)
    for j=1:size(Image,1)
        TMPI(dx+i,dy+j)=I(i,j);
    end;
end;
Image = TMPI;
%imshow(Image);
%xlabel([line,M1,size(Image)]);
res = Image;
%I2 = im_filter(Image);
%subplot(1,2,2);
%imshow(I2);
%subplot(1,2,2);
%imshow(uint8(N/16));
-----Iadjust.m-----

```

Appendix B.4: Source Code for Volumetric CT Reconstruction

```

-----Untitled1.m-----
x = zeros(400,400,30);
y = zeros(400,400,30);
z = zeros(400,400,30);
%v = zeros(382,382,3);
xx = [0.1:0.1:40];
yy = xx;
[XX,YY]=meshgrid(xx,yy);
for i = 1:30
    x(:,:,i) = XX;
    y(:,:,i) = YY;
    z(:,:,31-i) = i*ones(400,400)/3;
    v(:,:,i) = IM3D(i+69, :, :);
end;
%v = v/255;
p = patch(isosurface(x,y,z,v);
isonormals(x,y,z,v, p)
set(p, 'FaceColor', 'red', 'EdgeColor', 'none');
daspect([1 1 1])
view(3)
camlight; lighting phong
-----Untitled1.m-----

```