

California State University, San Bernardino

CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2005

Optimization of a sequential alignment verification and positioning system (SAVPS) for proton radiosurgery

Mahesh Raj Neupane

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Software Engineering Commons](#)

Recommended Citation

Neupane, Mahesh Raj, "Optimization of a sequential alignment verification and positioning system (SAVPS) for proton radiosurgery" (2005). *Theses Digitization Project*. 2784.
<https://scholarworks.lib.csusb.edu/etd-project/2784>

This Thesis is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

OPTIMIZATION OF A SEQUENTIAL ALIGNMENT
VERIFICATION AND POSITIONING SYSTEM
(SAVPS) FOR PROTON RADIOSURGERY

A Thesis
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Mahesh Raj Neupane
September 2005

OPTIMIZATION OF A SEQUENTIAL ALIGNMENT
VERIFICATION AND POSITIONING SYSTEM
(SAVPS) FOR PROTON RADIOSURGERY

A Thesis
Presented to the
Faculty of
California State University,
San Bernardino

by
Mahesh Raj Neupane
September 2005


Approved by:




Dr. Keith Evan Schubert, Committee Chair



Dr. Yasha Karant, Committee Member



Dr. Reinhard Schulte, Committee Member,
Loma Linda University Medical Center



Dr. Ernesto Gomez, Committee Member

August 8, 2005

Date

ABSTRACT

Stereotactic Radiosurgery (SRS) is a specialized form of radiation therapy that involves delivering a high dose of radiation to a very specific area in the body. SRS is an alternative to the traditional neurosurgery procedures which require invasive techniques such as drilling a hole in a patient's skull in order to gain access to the area of interest such as a tumor which is risky, especially for elderly patients. Proton-beams, due to their favorable physical characteristics, provide the ideal means to perform SRS. When SRS is used to create very small lesions in functional areas of the brain, this is called functional SRS.

Functional proton-beam SRS requires sub-millimeter alignment accuracy in order to be implemented for clinical trials. A patient tracking system, called Sequential Alignment and Position Verification System (SAVPS) is under development at Loma Linda University Medical Center (LLUMC), which will be used for functional proton SRS. An optical positioning system (OPS) is the key element of the SAVPS. It is manufactured by Vicon Peak and has been chosen to verify the correct alignment of the target point with the proton beam axis. The system provides the

position of retroreflective markers attached to the patient's head and beam delivery cone within ± 0.1 mm.

The main objective of this thesis is to optimize an existing version of SAVPS by conducting quantized error analysis. The coordinate transformation between global and the local coordinates, which is required for the alignment and verification process, is the major focus area of this thesis. Orthogonal, Least-Square based and Constrained Least Square based coordinate transformations were researched and compared in order to find the most accurate transformation algorithm. An image processing algorithm was developed and applied to estimate the error introduced by the Patient Positioning System (PPS) in order to derive the true error of the SAVPS. In addition to these procedures, efficient camera calibration patterns were developed to minimize the system error.

It was found that the Orthogonal Transformation outperforms both the standard Least-Square and the Constrained Least-Square based transformations by about one order of magnitude. The SAVPS error when using the orthogonal transformation had a mean error of 0.6 mm with a standard deviation of 0.3 mm.

ACKNOWLEDGMENTS

I would like to express my gratitude to the following individuals without their help this thesis would not have been possible.

To my advisor, Dr. Keith Schubert, for his guidance, encouragement and support throughout this thesis work. His enthusiasm in research and teaching has been a perennial source of inspiration to me. Working with him provided me an excellent learning opportunity.

To Dr. Reinhard Schulte (Loma Linda University Medical Center, LLUMC), for sharing his expertise in the Radiosurgery field, showing confidence on me and providing a platform for my research.

To my other thesis committee members, Dr. Yasha Karant and Dr. Ernest Gomez, for their support and helpful comments.

To Dr. Arturo Concepcion, chair of Computer Science Department, who has guided me through my graduate coursework and motivated me to do better in academic as well as in the life.

To my colleagues, Veysi Malkoc and Fadi Shihadeh, for their help and contribution to the successful completion of this work.

To my family, especially my charismatic brother Suresh and my ever supportive parents for their immense love, patience, and support.

Finally, I would like to thank my wife, Pratima, for giving her love, patience and friendship during this process of earning my graduate degree. Without her continuous love and encouragement, I could never have completed this degree.

This thesis is dedicated to my grandparents.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	v
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER ONE: BACKGROUND	
1.1 Introduction	1
1.2 Thesis Background	3
1.3 Nature of the Problem	7
1.4 Statement of the Problem	7
1.5 Purpose of the Thesis	8
1.6 Significance of the Thesis	9
1.7 Scope of the Thesis	9
1.8 Limitations	9
1.9 Definition of Terms	10
1.10 Organization of the Thesis	15
CHAPTER TWO: RADIOSURGERY MODALITIES	
2.1 Introduction	16
2.2 Stereotactic Radiotherapy and Radiosurgery	16
2.3 Gamma Knife Radiosurgery	20
2.4 Linear Accelerators Radiosurgery	22
2.5 Proton Beam Therapy and Stereotactic Radiosurgery	24
2.5.1 Factors Favoring Proton Beam Therapy	25

CHAPTER THREE: SYSTEM COMPONENTS

3.1 Introduction	30
3.2 Optical Positioning System	30
3.3 Geometrical Arrangement of the Cameras	31
3.4 Marker System	35
3.4.1 Camera Calibration Markers	36
3.4.2 Caddy Markers and Halo	38
3.4.3 Cone Markers	40
3.4.4 Phantom Base Markers and Reference Marker	41
3.5 Treatment Cone	43
3.6 Patient Positioning System	44
3.7 Assistive Software Systems	44
3.7.1 Serial Communication Software	44
3.7.2 Image Processing Software	45
3.7.3 Coordinate Transformation Software	46
3.8 Summary	46

CHAPTER FOUR: RESEARCH DESIGN AND PROCEDURE

4.1 Introduction	47
4.2 Theoretical Constructs and Assumptions	48
4.3 Experimental Procedures	49
4.3.1 Camera Calibration	49
4.3.2 Image Processing System	55
4.3.3 Stereotactic Coordinate Transformation for Functional Radiosurgery	65

4.3.4 Error Estimation	75
CHAPTER FIVE: RESULTS AND ACHIEVEMENTS	
5.1 Calibration Errors	80
5.1.1 Inward-Outward Circular Pattern Results	82
5.1.2 Vertical-Horizontal Pattern Results	84
5.2 Improved Image Processing Algorithm	86
5.3 Distance Error Calculations	91
5.4 Comparison of Coordinate Transformations	93
5.4.1 Introduction	93
5.4.2 Orthogonal Coordinate Transformation	94
5.4.3 Least Square-based Coordinate Transformation	97
5.4.4 Constrained Least Square Based Transformation	101
5.4.5 Summary	103
CHAPTER SIX: SUMMARY, CONCLUSION AND RECOMMENDATIONS	
6.1 Summary	104
6.2 Conclusion	106
6.3 Recommendations	107
APPENDIX A: IMAGE PROCESSING ROUTINES	109
APPENDIX B: ORTHOGONAL TRANSFORMATION	116
APPENDIX C: LEAST SQUARE BASED TRANSFORMATION	144
APPENDIX D: CONSTRAINED LEAST SQUARE BASED TRANSFORMATION	151
REFERENCES	159

LIST OF TABLES

Table 1. Calibration Results Produced by Random Pattern	81
Table 2. Inward-Outward Circular Pattern Calibration Results	83
Table 3. Calibration Parameters Using Vertical-Horizontal Calibration Pattern	85
Table 4. Distance Errors	92
Table 5. Alignment Errors Produced by the Orthogonal Transformation	96
Table 6. Alignment Errors Produced by the Least Square Based Transformation	100
Table 7. Alignment Errors Produced by Constrained Least Square Method	102

LIST OF FIGURES

Figure 1.	Proton Beam Treatment Gantry and Patient Positioning System	4
Figure 2.	Halo (Left) and Fiducial System (Right)	5
Figure 3.	Gamma Knife	21
Figure 4.	Wave Guide of a Linear Accelerator	22
Figure 5.	Geometrical Arrangement of the Cameras in the Front View	32
Figure 6.	Top View of the Experimental Camera Arrangement with Respect to Halo and Cone	34
Figure 7.	Side View of the Experimental Camera Arrangement with Respect to Halo and Cone	35
Figure 8.	Retroreflectivity Principle of Vicon System Markers	36
Figure 9.	Static L-Shaped Marker System	37
Figure 10.	Wand Used for Dynamic Calibration	38
Figure 11.	Marker Caddy and Halo Coordinate System	39
Figure 12.	Marker Cross and Treatment Cone	41
Figure 13.	Phantom Base Markers and Reference Marker	42
Figure 14.	Treatment Cone and Marker Cross	43
Figure 15.	Experimental Setup	48
Figure 16.	Static Calibration Setup	50
Figure 17.	Dynamic Markers	51
Figure 18.	Random Pattern for Dynamic Calibration	52

Figure 19. Inside and Outside Circular Pattern	54
Figure 20. Vertical-Horizontal Pattern	55
Figure 21. Digitization of a Continuous Image	57
Figure 22. Processing of Alignment and Image Capturing	62
Figure 23. Previous Image Processing Algorithm	63
Figure 24. Image Processing Algorithm Developed in this Thesis	64
Figure 25. Inequality Based Constrained Least Square Based Algorithm	75
Figure 26. Geometry of the Shortest Distance Between Target and Beam Axis	78
Figure 27. Mean Residuals and Reproducibility for the Random Pattern	82
Figure 28. Mean Residuals and Reproducibility for the Inward-Outward Circular Pattern	84
Figure 29. Mean Residuals and Reproducibility for the Horizontal-Vertical Pattern	86
Figure 30. Border and Center Estimation Produced by the Previous Image Processing Algorithm	87
Figure 31. Input Image for the Image Processing System	89
Figure 32. Processed Image Corresponding to the Image of Figure 31	90
Figure 33. Output Script Produced by the Matlab Image Processing System	91
Figure 34. Caddy and Cone Markers Distance Errors	93
Figure 35. System Alignment Errors Produced by the Orthogonal Transformation	97

Figure 36. System Alignment Errors Produced by
the Least Square Based
Transformation101

Figure 37. System Alignment Errors Produced by
Constrained Least Square103

CHAPTER ONE

BACKGROUND

1.1 Introduction

The content of this Chapter gives an overview of the thesis. The contexts of the problem with necessary background are discussed followed by the purpose, significance of the thesis, and assumptions. Next, the limitations that apply to the thesis are reviewed. Finally, definitions of terms are presented.

Stereotactic Radiosurgery (SRS) is a specialized form of radiation therapy that involves delivering a high dose of radiation to a specific anatomical area in the body. This technique was introduced by Lars Leksell in 1949 to treat brain tumors and create functional brain lesions using many small stationary treatment beams and immobilizing the patient in a stereotactic frame. Since then it has been used for more than 40 years to treat a variety of diseases in the brain.

The concept of SRS evolved from the basic principle of radiation therapy, in which diseased tissue is treated at a sufficiently high dose to achieve local control, while sparing as much healthy surrounding tissue as possible. Since its introduction, SRS has undergone much

transformation and is well supported by fast developing technologies such as increased computer capacity, modern, precise imaging techniques such as Computed Tomography and Magnetic Resonance Imaging (MRI), accurate targeting technology, and precise patient immobilization and positioning systems.

Loma Linda University Medical Center (LLUMC), located in Loma Linda, California, is the one of the prime facilities to implement SRS with protons. The LLUMC's Proton Treatment Center was the first hospital-based proton-beam facility in the world, and has performed research and development in the proton radiosurgery field since 1991. In 1995, a small field project group was established out of a group of internationally acclaimed researchers and staff from the center. The group's purpose was and still is to develop novel proton radiosurgery techniques for practical use in treating cancer and functional disorders such as Parkinson's disease with narrow beams and high doses.

A sub-project entitled "Sequential Alignment and Positioning Verification System for Functional Proton Radiosurgery (SAVPS)," was established in 2000 by LLUMC group members Dr. R. Schulte, Dr. M. Moyers, Dr. R. Levy, Dr. D. Miller in the context of a clinic project at Harvey

Mudd College, located in Claremont, California. After the initial system layout was developed, Veysi Malkoc, a graduate student from California State University, San Bernardino continued to work on this project experimentally and theoretically under the supervision of Dr. Yasha Karant, Dr. Keith Schubert, and Dr. Ernesto Gomez. The main objective of this project is to provide an accurate verification system for highly accurate and precise patient positioning both before and during the course of functional proton SRS.

1.2 Thesis Background

Traditional neurosurgery procedures require invasive techniques such as opening the patient's skull in order to gain access to the area of interest such as a cancer or tumor. Because of the complications involved, these procedures are risky, especially for elderly patients. In order to provide a smooth and more accurate technique, an alternative technique known as proton-beam radiosurgery is being developed, which involves targeting multiple narrow high-energy proton beams to destroy a small target in the patient's brain.

The high-energy proton beam needed for proton SRS is generated by the LLUMC proton synchrotron, located outside

the treatment room, and then transported through a narrow evacuated tube to a large gantry in the treatment room. The gantry, part of it is shown in Figure 1, has a full 360-degree rotation range about a horizontal axis. For proton RSR, a cone at the end of the beam delivery system collimates the proton beam to a narrow beam of 2-3 mm diameter. The gantry rotates within a plane such that the central beam axis will always be approximately directed at a point on the gantry's rotation axis, called the isocenter.

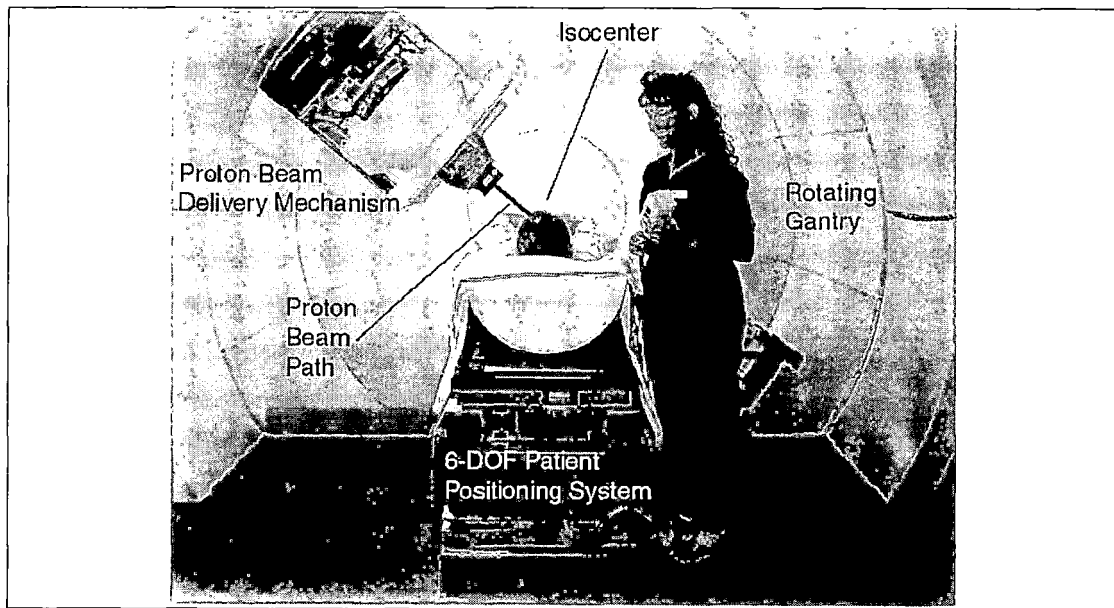


Figure 1. Proton Beam Treatment Gantry and Patient Positioning System

Prior to the treatment, a circular metal frame, called a stereotactic halo, is firmly affixed to the

patient's skull to establish a local coordinate system about the patient's head. A box-like frame called a fiducial system is attached to the halo prior to imaging the patient for target localization. Both devices are shown in Figure 2. An MRI study of the head is used to determine the precise location of the target area in the patient's brain relative to the fiducial system. Consequently, it can be assumed that the position of the target area will be known in the halo's coordinate system. The imaging fiducial system is removed from the patient, who will then be placed on a six-degree-of-freedom (6-DOF) table, called a Patient Positioning System (PPS), in the treatment room.

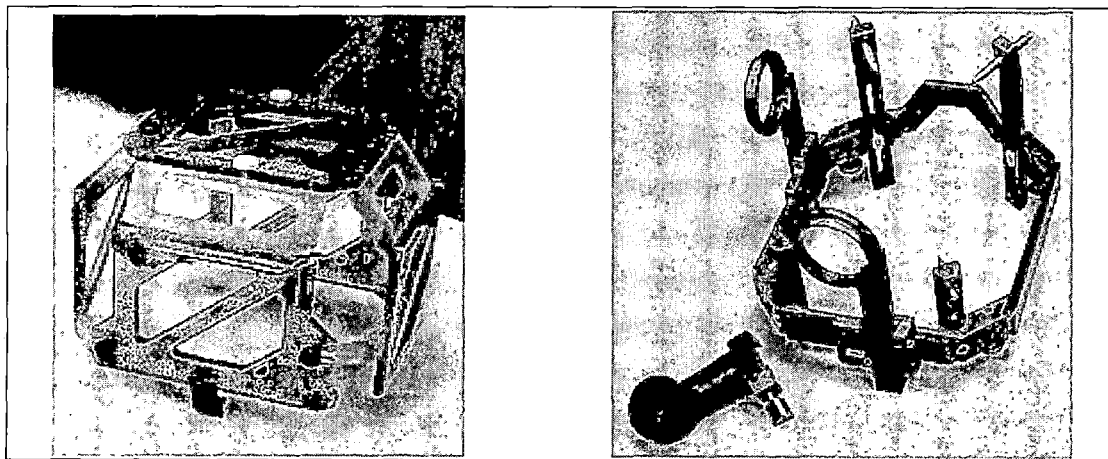


Figure 2. Halo (Left) and Fiducial System (Right)

The PPS and Gantry will be positioned and oriented such that the proton beam path intersects the target area

from multiple directions, creating a highly focused dose distribution. For each treatment angle, the proton beam radiation will be applied for approximately forty seconds. The Patient Positioning System (PPS) will be repositioned approximately five times and at each PPS position, the radiation will be applied from five to seven different gantry positions, resulting in thirty to thirty-five narrow beams per treatment. The proton beam axis must intersect the target center within a tolerance of at least ± 0.5 mm to avoid the risk of injury to critical brain structures located in proximity to the target area.

The effectiveness of the SAVPS depends foremost on its accuracy. Therefore, accuracy is a bounded core objective, which applies to both position and orientation. The required position accuracy implies that the target center should be positioned to within at least ± 0.5 mm from the proton beam axis for any beam direction; however, even greater accuracy of better than ± 0.25 mm is desirable. Orientation accuracy, which refers to the angle at which the proton beam intersects the target area, is less critical and should be within $\pm 5^\circ$ of the desired angle, which is met by the current proton beam delivery system.

1.3 Nature of the Problem

Due to its complexity, the SAVPS is susceptible to many hardware as well as software deficiencies affecting its overall accuracy. Although its performance is critical for only the beam delivery part of SRS, it holds the key to the success of the entire radiosurgical procedure. Inaccuracy of dose delivery means normal tissue injury and / or treatment failure. Hence, the effectiveness of the proton radiosurgery procedure largely depends upon how well the target is aligned to the proton beam during the treatment, which is the objective of the SAVPS system. Because even relatively small component errors may contribute to a possibly large systematic or random error, the performance of the SAVPS system is based the efficiency of individual components.

1.4 Statement of the Problem

For functional proton radiosurgery, patients are placed at specified positions and orientations so that the desired treatment area in the patient's brain is accurately aligned with the path of the proton beam. The SAVPS in its current form is only able to achieve an accuracy of no better than ± 1.5 mm, which is far away from the goal of aligning the anatomical target center to

within ± 0.5 mm with respect to the center of the proton beam.

1.5 Purpose of the Thesis

The purpose of the thesis is to improve the existing version of the Sequential Alignment and Position Verification System (SAVPS) for functional proton radiosurgery and to evaluate its performance after improvement. Improvement is to be researched by determining the most accurate camera calibration pattern, image processing algorithm, and coordinate transformation. Coordinate transformation methods to be evaluated include unitary transformation, unconstrained least square based transformation and constrained least square based transformation. Possible outcomes of the research results of this thesis include development of new alignment hardware for the system, a new calibration pattern for the cameras, more efficient software for the image processing to determine the offset between the central beam axis and the target, and decision regarding the best transformation method among the three methods listed above. Eventually, I will determine the alignment accuracy that can be achieved with the system after the improvements in calibration

pattern, image processing algorithm, and transformation software development have been implemented.

1.6 Significance of the Thesis

The significance of the thesis is to develop and analyze an efficient procedure for aligning the anatomical target with respect to the center of the proton beam with a new method that is more accurate, precise and robust than the existing.

1.7 Scope of the Thesis

Based on the results and insights of the thesis, it may serve as the reference for further research of the SAVPS system. This thesis may be a significant step towards the clinical use of the SAVPS system in the treatment of patients suffering from Parkinson's disease and Trigeminal Neuralgia (attack of shooting pain in facial muscles).

1.8 Limitations

During the development of the project, a number of limitations were noted. These limitations are presented here.

1. The positioner table is accurate in "fine" translational coordinates ± 0.4 mm, ± 0.02 mm, ± 0.08 mm z, t, s axis respectively.
2. The positioner table induced ± 0.1 mm of vertical error when moved in vertical direction.
3. All of the measurements are referenced to the Dimension Inspection Laboratory coordinate values (DIL), which are accurate to within ± 0.1 mm.
4. The treatment cone projects the laser beam to the target point (marker). The projection occurs on a flat surface with minimal distortion of the beam shape and the marker shadow.

1.9 Definition of Terms

The following terms are defined as they apply to the thesis.

6 - Degree of Freedom (DOF) - 6 types of movements

performed by The Patient Positioner System (PPS), including translations along three orthogonal axes (horizontal, vertical, longitudinal) and three rotations (pitch, roll, yaw) about these axes.

Binary Image - Binary images are images whose pixels have only two possible intensity values (0's and 1's). The

darker regions are specified as 1's and brighter regions are specified as 0's.

Bragg peak - The region at which protons (and other heavy charged particles) deposit most of their energy. This region occurs near the end of the protons' paths. By varying the beam's energy, radiation oncologists can spread and position the depth of this peak to match the contours of tumors or other targets.

Cancer - Uncontrolled, abnormal growth of cells, which will invade and destroy healthy tissues if not controlled by effective treatment.

Cobalt 60 - A naturally radioactive substance that is used in machines to treat cancer by external beams.

Conduit - The proton beam generated by channeling protons from a proton accelerator outside the treatment room connects through a narrow conduit to a large cylindrical gantry.

Cross - A localization device attached to the treatment cone. It is made of metal, shaped like a cross and has a marker system that has also the shape of a cross.

Edge pixels - Pixels that belong to the border of an object.

Fiducial system - A box-like frame, which is attached to the halo.

Gamma rays - High-energy rays that come from a radioactive source such as cobalt-60.

Gantry - A device for rotating the radiation delivery apparatus around the patient, so as to treat from different angles and mainly used in radiation therapy. The gantry has a full 360-degree rotation range about a horizontal axis.

Halo - A circular metal frame, which is firmly affixed to the patient's skull to establish a coordinate system about the patient's head.

Immobilization device - A device that prevents the patient from moving during radiation treatment. One example, used for proton treatment of body targets at Loma Linda, is a form-fitting foam liner surrounded by a rigid plastic shell, in which a patient can lie comfortably during treatment.

Isocenter - At the end of the conduit is a cone that will collimate the proton beam to ensure a straight and narrow beam. The conduit rotates with the gantry on a plane and can move radially but will always be approximately directed at a point on the gantry's rotation axis, called the isocenter.

Laser beam - A very directional, very tight, very intense and concentrated beam that is formed by stimulated emission of photons from a crystal.

Linear accelerator - A machine that creates high-energy photons to treat cancers, using electricity to form a stream of fast-moving subatomic particles. Also called a megavoltage (MeV) linear accelerator or "linac" (pronounced LYNN-ack).

Marker - Marker is a plastic sphere covered with retro-reflective tape.

Marker caddy - A frame, which has a marker system on and fixed to the halo in order to track patient's head by the cameras.

Matlab - A mathematical packaging software used for precise and efficient mathematical calculation.

Patient Positioner System (PPS) - A table which allows precise and accurate patient positioning within its specifications.

Parkinson's disease - Parkinson disease is a functional brain disorder leading to impairment of the motor function. It occurs when certain nerve cells (neurons) in a part of the brain called the substantia nigra die or become impaired

Phantom - A device that has pins carrying a target marker and holes where the pins can be placed. In this thesis, it was used to test alignment accuracy.

Photon - A quantum (energy packet) of electromagnetic radiation; the elementary particle of photon radiation therapy. X rays and gamma rays are photon radiation.

Proton - Positively charged subatomic particle, which forms the nucleus of the hydrogen atom.

Proton Radiation Therapy - It is a form of external-beam radiation treatment.

Radiation oncologists (physicians who specialize in radiation treatments) can treat the tumors, functional lesions, etc. by using various forms of high-energy radiation such as gamma rays, high-energy photons from a linear accelerator, or protons.

Radiosurgery - Radiosurgery is pinpoint precision radiation using multiple, finely-contoured beams from many different angles - all directed at the target and minimizing radiation to normal tissue while the patient's body is maintained in a stable, reproducible position.

Thresholding - The technique used to differentiate the object from the background.

Treatment cone - The actual treatment device that directs and collimates proton radiation beams.

Tumor - An abnormal mass of tissue. Tumors are either benign or malignant.

Vicon - The company that produces the cameras used in this thesis.

X rays - Ionizing radiation consisting of high-energy photons that can be used at low doses to diagnose disease or at high doses to treat cancer.

1.10 Organization of the Thesis

The thesis was divided into five chapters. Chapter One provides an introduction to the context of the problem, purpose of the thesis, significance of the thesis, limitations, and definitions of terms. Chapter Two consists of a review of relevant literature in radiosurgery. Chapter Three documents the system components used in this thesis. Chapter Four presents research design and procedure for the thesis. Chapter Five presents the results and milestones achieved in the thesis. Chapter Six illustrates the summary, recommendation and discussion on the milestone achieved in the thesis. The Appendices for the Thesis follows Chapter Six. Finally, the references for the Thesis are presented.

CHAPTER TWO

RADIOSURGERY MODALITIES

2.1 Introduction

Chapter Two describes in detail stereotactic radiotherapy and radiosurgery as well as other types of radiosurgery. At last it also summarizes the advantage of using the proton beam for the stereotactic radiosurgery.

2.2 Stereotactic Radiotherapy and Radiosurgery

In the early 60's, treating patients with brain tumors and other ailments non-invasively and accurately was a dream because of limited radiological tools and supporting technologies such as large computer capacity, modern imaging techniques, complex patient immobilization devices and targeting technology. These technologies are a result of the work and research of countless individuals over the past 3-4 decades.

Stereotactic radiation techniques provide the ultimate form of precision therapy. These techniques are based on the premise that the exact target location is known in a well-defined 3D space with an accuracy that normally ranges from 1-3 mm. This principle combined with a rigid patient immobilization can be applied to any form of radiation, and the treatment can be delivered in a

single session (radiosurgery) or in more than one session (fractionated stereotactic radiotherapy).

Fractionated stereotactic radiation treatments, which extend over a period of two days to many weeks, are administered with the assistance of removable masks or frames that provide a degree of immobilization somewhat less than that achievable with invasive stereotactic frames used for radiosurgery. This treatment modality has been mostly limited to the head and neck region as these areas can be immobilized with skeletal fixation devices that restrict the head's movement, permitting precise and accurate treatment. More recently also body stereotactic devices have been developed but their use is limited.

Stereotactic radiosurgery, a one-session radiation treatment with a highly effective dose, has such a dramatic effect on the tissue in the target zone that the resulting changes are considered "surgical." Through the use of three-dimensional computer-aided planning and a high degree of immobilization provided by invasive halos affixed to the skull, the treatment can minimize the amount of radiation to healthy brain tissue. Stereotactic radiosurgery is routinely used for inoperable brain tumors and to detect lesions in functional disorders such as Parkinson's disease and epilepsy. It may also be used as a

boost or adjunct to surgery for recurring or malignant tumors.

Stereotactic radiosurgery and radiotherapy are based on the same mechanisms as other forms of radiation treatment. Radiation therapy uses high-energy photon beams (X-rays or gamma rays), neutrons, or light and heavy charged particles (electrons, protons, or heavy ions) to damage critical biological molecules in target cells. One important characteristic of any ionizing radiation technique is the localized release of large amounts of radiation in the target area. Ionizing radiation produces substantial biological effects for the relatively small amounts of energy involved because the energy is released in "packets" large enough to break chemical bonds and initiate the chain of events that ultimately lead to biological effect. It does not remove the tumor or lesion, but it damages the DNA of the tumor cells. The cells then lose their ability to reproduce or die from apoptosis (a programmed cell death). The tumor reduction occurs at the rate of the normal growth rate of the specific tumor cell. In lesions such as Arterio-Venous Malformations (AVMs) consisting of a tangle of blood vessels in the brain), radiosurgery causes the blood vessels to thicken and close off, which is the desired therapeutic effect. The

shrinkage of a tumor or closing off of AVM vessels occurs over a period of time. For benign tumors and AVM vessels, the desired response will usually take many months to years. For malignant tumors and metastasis tumors, results may be seen much sooner as these cells are very fast-growing.

Treatment of brain tumors with stereotactic radiotherapy and radiosurgery has been an area of intense research activity over the past several decades. Through clinical research, conducted on patients, much has been learned about how to appropriately use these techniques for various types of brain tumors and functional disorders. External beam radiation therapy, both stereotactic and non-stereotactic, is a valuable component of therapy for nearly all brain tumors. The ability to assure uniform doses of radiation to the areas being targeted is one of the major strengths of modern external beam radiation therapy based on the use of high-energy radiation sources.

Radiosurgery and fractionated stereotactic radiation therapy nowadays are performed with three distinct methods. Each method operates with a different source of radiation and may be preferable in one way or another depending on factors such as costs, accuracy, and

availability. These methods are Gamma Knife radiosurgery, the LINAC radiosurgery and radiation therapy, and the proton radiosurgery and radiation therapy. The following sections will describe each method in greater detail.

2.3 Gamma Knife Radiosurgery

The Gamma Knife is recognized worldwide as the preferred radiosurgery instrument for small brain tumors, AVMs, and functional disorders such as trigeminal neuralgia, epilepsy, and Parkinson's disease. Like the other radiosurgical instruments, the Gamma Knife offers a non-invasive alternative for many patients for whom traditional brain surgery is not an option and removes the physical trauma and the majority of risks associated with open surgery. Gamma Knife radiosurgery is performed in one session with extreme precision, sparing tissues adjacent to the target. Based on pre-radiosurgical radiological examinations, such as CT-scans, MR-scans, or angiography, the unit provides for highly accurate irradiation of deep-seated targets, using a multitude of collimated Cobalt gamma radiation beams with scalpel-like precision.

The Gamma Knife contains 201 cobalt-60 sources of approximately 30 curies each, placed in a circular array in a heavily shielded unit (lead-lined helmet). The helmet

has slit-like openings (collimators) that direct gamma rays to the one or several isocenters to which the patient's target is aligned. The beam from each individual Co source is collimated through the outer collimator helmet and then through the inner collimator helmet that narrows the beam further, reducing penumbra typically inherent with Co sources. All 201 finely focused beams intersect at the isocenter, sparing normal tissue and maximizing the dose to the target volume to within 0.5 mm accuracy. The Gamma Knife principles of operation are based upon the "center of arc" principle, in which the center of the target is at the center of the circular arc of rotation. The gamma knife principle and gamma knife lead-line helmet are shown in Figure 3.

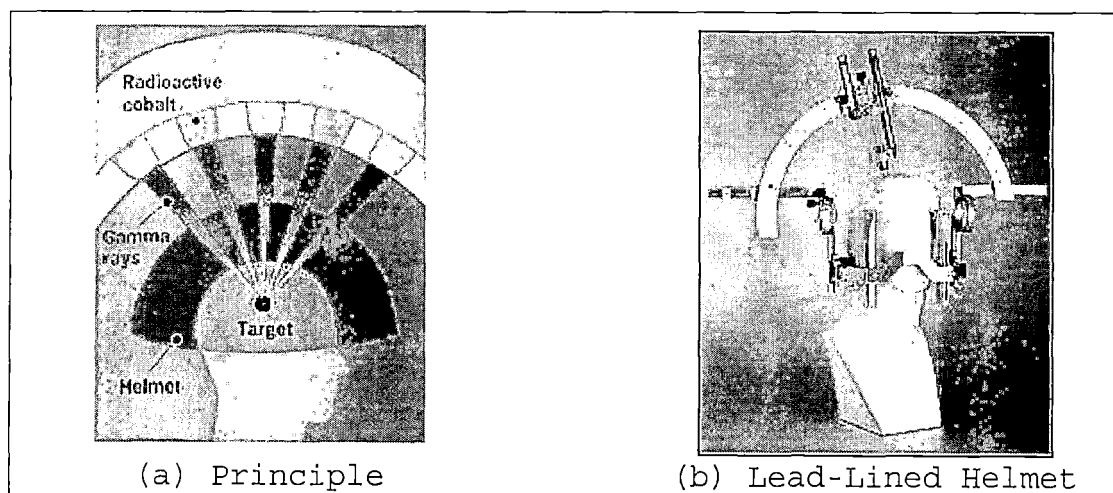


Figure 3. Gamma Knife

2.4 Linear Accelerators Radiosurgery

Linear Accelerators (LINACs) are physics devices used to accelerate atomic and sub-atomic particles to high velocities. The radiotherapy LINACS are based on microwave technology (similar to that used for radar); they accelerate electrons in a part of the accelerator called the "wave guide" (Figure 4), which then collide with a heavy metal target. As a result of the collisions, high-energy photons (so called "bremsstrahlung") are produced in the target. A portion of these photons is collected and then shaped to form a beam that matches the patient's tumor. The photon beam is delivered by a gantry, which rotates around the patient. The patient lies on an adjustable treatment couch and wall-mounted lasers are used to make sure the patient is in the proper position. Radiation can be delivered to the tumor from any direction by rotating the gantry and the treatment couch.

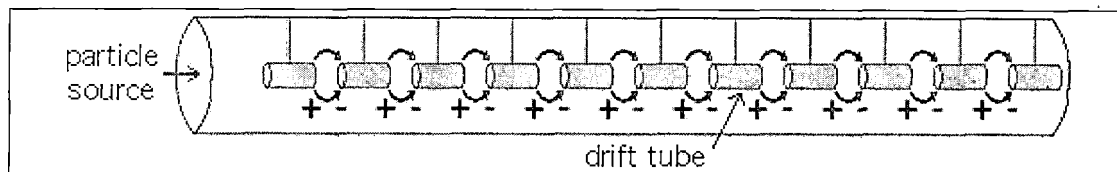


Figure 4. Wave Guide of a Linear Accelerator

Electrons or other charged particles are injected from the left and are guided and accelerated by a high-frequency electromagnetic field.

Figure 4 shows the principle of a LINAC's wave-guide, charged particles (here electrons) enter on the left and are accelerated towards the first drift tube by an electric field. Once inside the drift tube, they are shielded from the field and drift at a constant velocity. When they arrive at the next gap, the field accelerates them again until they reach the next drift tube. This continues, with the particles picking up more and more energy in each gap, until they are injected from the linear accelerator on the right. The intermediate drift tubes are necessary because an alternating field is used and without them the field would alternately accelerate and decelerate the particles. The drift tubes shield the particles from the field influence for the length of time that the field would be decelerating. Thus, the LINAC is an electromagnetic catapult that brings electrons from a standing start to relativistic velocity, i.e., a velocity near the speed of light.

LINACs are mostly used in fractionated non-stereotactic radiotherapy but dedicated or multipurpose LINACs exist that are used for radiosurgery

and fractionated stereotactic radiotherapy alone or in addition to conventional radiotherapy. Stereotactic radiosurgery treatments with LINACs are performed with multiple rotational arcs. An arc is a segment of gantry rotation during which the radiation is delivered continuously while the gantry moves. A combination of several rotational arcs produces a concentric focal dose similar to that of the Gamma Knife. However, the LINAC lacks the mechanical stability of the stationary Gamma Knife and is therefore not used for functional radiosurgery treatments which require submillimeter accuracy and precision.

2.5 Proton Beam Therapy and Stereotactic Radiosurgery

Proton beams have a distinct physical advantage over conventional photon beams (x rays). Photons can successfully be used as an instrument for radiosurgery and fractionated stereotactic radiation therapy, but due to their physical characteristics they deliver doses of irradiation to a substantial amount of normal tissues surrounding the target. Proton beams, on the other hand, stop abruptly at a prescribed depth which can be adjusted by choosing the right proton energy. The pattern of energy deposition is characterized by the Bragg peak, wherein the

dose is minimal on entry and reaches a maximum at the region where protons stop, which is usually within or just behind the target volume. Proton beams can thus be shaped to deliver homogeneous radiation doses to irregular three-dimensional volumes. By modulating the energy of the beam during the treatment, the radiation oncologist can spread out the Bragg peak to encompass larger volumes. This essentially reverses the photon pattern: while protons build their dose up near the end of their travel, photons deliver their maximum dose near the surface.

The most desirable characteristic of a conventional proton beam is the sharp dose fall-off to near zero shortly after the peak in dose. This allows clinicians to attack tumors that may be very close to organs at risk, with a high dose to the target volume and limited dose to critical structures. Protons are therefore useful for both non-stereotactic as well as stereotactic treatments and can be delivered in single and multiple fractions depending on the radiation technique. As such, protons may be considered the most useful three-dimensionally conformal radiation technique.

2.5.1 Factors Favoring Proton Beam Therapy

The following are the factors favoring proton beams for applications in radiation therapy.

2.5.1.1 Charge. The proton's electric charge (positive elementary charge) enables the radiation oncologist to stop protons in a desired location thus offering a potential therapeutic advantage due to the ability to locate the beam precisely. Neutral particles such as neutrons and photons cannot be stopped inside the patient and can therefore only be controlled laterally, i.e., in two dimensions. Electrons are also charged particles and, in fact, are employed in radiation therapy as a stopping beam. However, because electrons are very light particles they are scattered considerably and their dose distribution does not show the Bragg peak effect of heavier charged particles such as protons. One can also use heavy ions such as carbon for Bragg peak radiation therapy but these are much more expensive to accelerate as they require very large accelerators and gantries.

2.5.1.2 Mass. As already mentioned the mass of the subatomic particle used in radiation therapy influences the manner in which it is depositing energy within the patient. As the mass of the particles diminishes, that is when one goes from heavy ions such as carbon or helium to protons and to electrons scattering, due to the electric fields of the target atoms increasingly occurs. Such scattering tends to defocus the beam. Because the primary

interaction of the incoming beam of charged particles within the patient is with atomic electrons, the particles in the incoming beam ideally should have a mass much greater than the orbiting electrons, to avoid being scattered as their individual electric fields interact.

The mass of protons is 1,835 times that of an electron; hence, lateral scattering is reduced greatly as compared to an electron beam. High-energy photon beams are also scattered relatively little however their dose deposition pattern and charge makes them less favorable. The least amount of proton scattering of protons occurs at the highest energies available for treatment (for example, at 250 MeV). Such high-energy proton beams may be used without stopping them in a patient (so called shoot-through beams) to take advantage of their exquisite lateral sharpness in functional radiosurgery treatments.

The physical characteristics of protons can thus be exploited in different ways depending on the disease and target characteristics. While a high-energy shoot through beam may be used to precisely create small lesions within a diseased brain area, larger and irregular targets require a Bragg-peak treatment. A modern proton treatment facility has the ability to provide the technology that manipulates and modulates the direction and primary energy

of each proton beam. The task of the radiation oncologist is then to place the focused energy in targeted cells. This requires three-dimensional control of each beam used. The finer the control the physician has over the therapy beam, the better treatment the patient will receive regardless of the particular application.

2.5.1.3 Linear Energy Transfer. Radiation oncologists tend to think of protons and helium ions as "light" ions, in the sense that both are characterized by relatively sparse ionization, or linear energy transfer (LET) as they pass through tissue. Basic radiation effects in living tissue are determined by two main factors: the physical dose distribution and the LET. Thus, a high-LET beam may be more effective than a low-LET beam despite the same physical dose being delivered. Photons, electrons, and protons are categorized as low-LET particles whereas neutrons and heavy ions are considered to be high-LET particles in therapeutic terms. The use of heavy ions in radiation therapy has been advocated and practiced in a limited number of places, usually in high energy physics laboratories. Although theoretically they have advantages in very resistant tumors there is very little clinical experience to date. They are also very expensive in their production and may produce more severe effects in the

surrounding normal tissues. Neutrons have been used more widely in radiation therapy but the lack of three-dimensional control makes them unsuitable for radiosurgery.

CHAPTER THREE

SYSTEM COMPONENTS

3.1 Introduction

The purpose of the Sequential Alignment Verification and Positioning System (SAVPS) is to detect the position of a patient- and proton beam centered marker system in space in an online-fashion. Since the relationship of the beam-centered marker systems relative to the proton beam axis and the patient-centered marker system relative to the anatomical target are known, it is possible to calculate the offset between beam axis and target using an appropriate mathematical transformation. The major components of the SAVPS are the Optical Positioning System, Marker Systems, Treatment Cone, Halo, Patient Positioning System and Assistive Software System. These components will now be presented in detail. Chapter Three documents all the major components that constitute the SAVPS.

3.2 Optical Positioning System

The Optical Positioning System (OPS) is comprised of a set of three infrared-strobes cameras used to capture the image of retroreflective markers in 3D space attached to the patient's halo and the beam delivery cone. The

position of the patient's target can only be determined when the position of a minimum of three separate non-collinear markers is determined. The OPS is commercially available from Vicon Peak, Lake Forest, CA 92630 (former Vicon Motion Systems). The advantages of the OPS include: no direct contact with the patient or other equipment, no audio disturbances, and the use of infrared-light-emitting diodes (ILEDs) for target illumination, which eliminates visual disturbances. The system used for this thesis operates with three M-Cam series cameras, which have a resolution of 1,000,000 pixels, ideal for the desired application. The system requires pre-experiment static calibration using an L-shaped marker frame and a dynamic calibration using a wand with two spherical markers.

3.3 Geometrical Arrangement of the Cameras

The efficiency of the system mainly relies on the accuracy by which the OPS cameras capture marker images. In order to achieve optimum accuracy, cameras have to be placed in a proper geometrical position with respect to the marker systems. There are geometrical constraints imposed by the proton treatment room at Loma Linda University Medical Center, where the system will be used.

In the real application, the cameras will be placed in an equilateral configuration at the edges of a circular disk of 60 cm radius at the back of the gantry (Figure 2) located about 160 cm from the gantry's isocenter. When the proton gantry is moved to a new treatment angle, the cameras will rotate with it. In order to provide constant geometrical conditions, each camera will be aimed directly at the isocenter. The resulting camera configuration in the front view is shown in Figure 5.

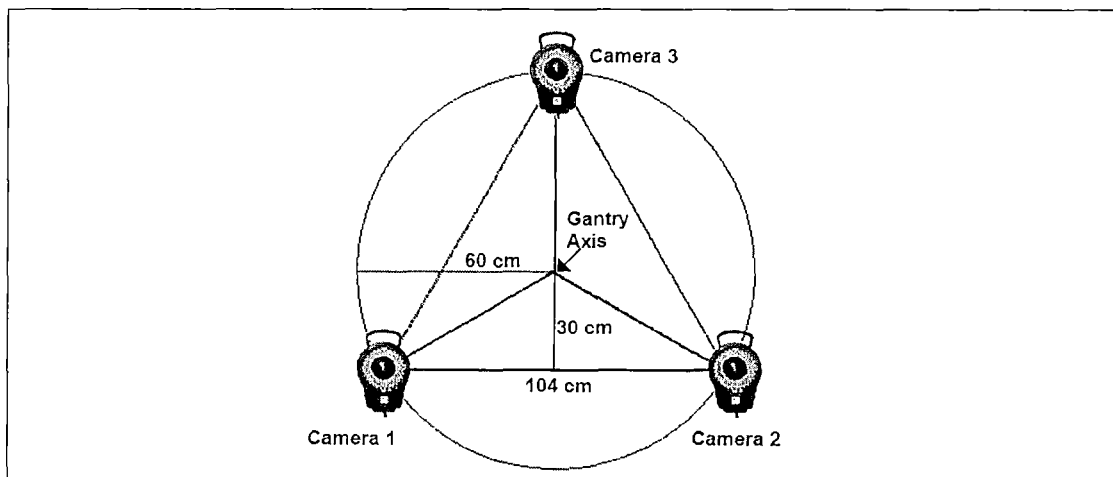


Figure 5. Geometrical Arrangement of the Cameras in the Front View

Since the back of the gantry and the proton beam delivery cone rotate as one unit, the position of the cone relative to the cameras will be fixed except for a small deviation due to the mechanical sag of the gantry. The camera placement at the back of the gantry will minimize

obstructions to the cameras' field of view of the marker systems attached to the patient's head and proton beam delivery cone.

For the purpose of the experiments conducted in this thesis, which could not be carried out in the real gantry environment due to access limitations and time constraints, the cameras were arranged on sturdy Bogen tripods matching the configuration shown in Figure 5 as closely as possible. Due to limitations of the room where the experimental setup was placed, the exact distance of the camera plane from the marker systems that will be used in the treatment room could not be reproduced but was not too different either (100 cm instead of 162 cm). The top and side views of the experimental camera location are shown in Figures 6 and 7 below.

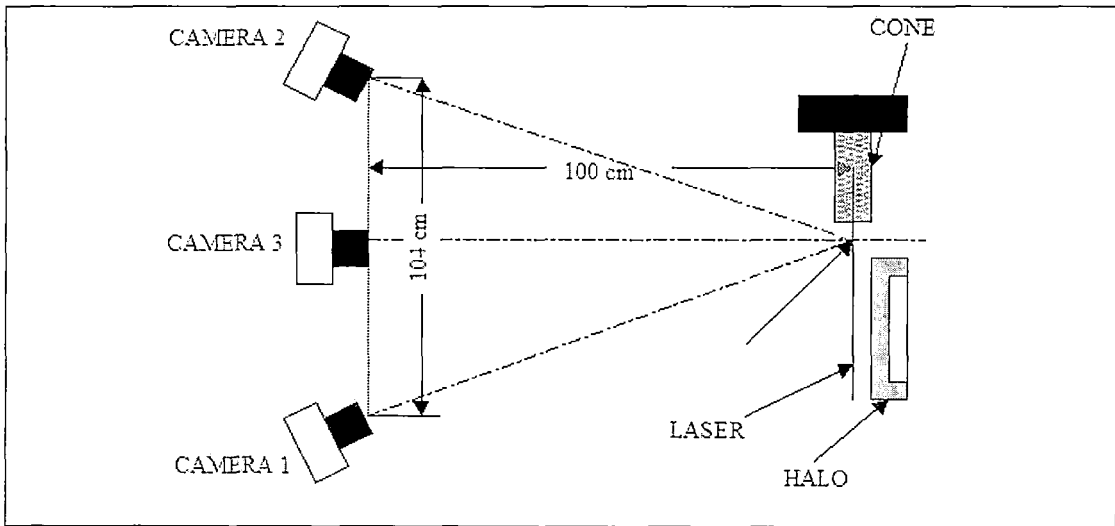


Figure 6. Top View of the Experimental Camera Arrangement with Respect to Halo and Cone

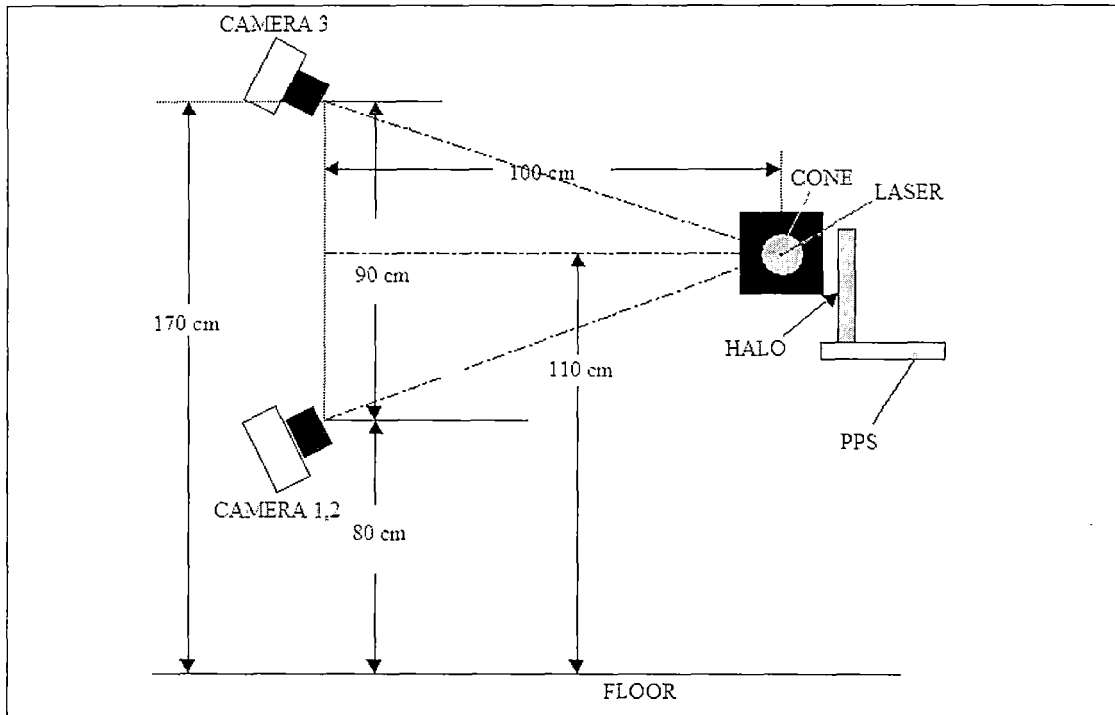


Figure 7. Side View of the Experimental Camera Arrangement with Respect to Halo and Cone

The cameras used for the experiments in this thesis were equipped with lenses of 25 mm focal length. The 25-mm lenses achieve higher resolution than the standard 50-mm lenses, but at the cost of a more limited field of vision (FOV). The 25-mm lenses provide, at minimum, a cone of 20° FOV. For the camera arrangement shown in Figures 5-7, this is sufficient to cover a volume of at least 0.3 m x 0.6 m x 0.6 m at the focal distance of 100 cm, in which the marker systems have to be placed.

3.4 Marker System

Vicon recommends using spherical markers covered with retroreflective tape for the camera system.

Retroreflective surfaces reflect a large fraction of incident light directly back at the light source. The retroreflective characteristic of the marker system is shown in Figure 8.

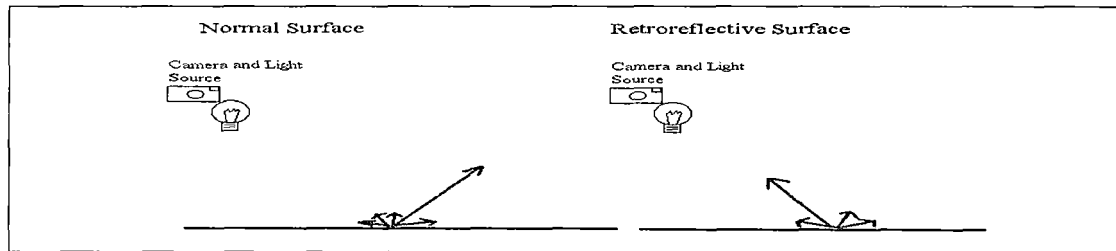


Figure 8. Retroreflectivity Principle of Vicon System Markers

The marker systems used for our experiment can be categorized according to their function into camera calibration markers, caddy markers, cone markers, and phantom base markers. These will be presented in more detail below.

3.4.1 Camera Calibration Markers

Camera Calibration Markers serve to calibrate the OPS before its use. Vicon provides two types: a static L-shaped marker system and a dynamic marker wand.

3.4.1.1 Static L-Shaped Markers. An L-shaped marker arrangement, shown in Figure 9 is used for a static calibration, which is performed before the dynamic calibration.

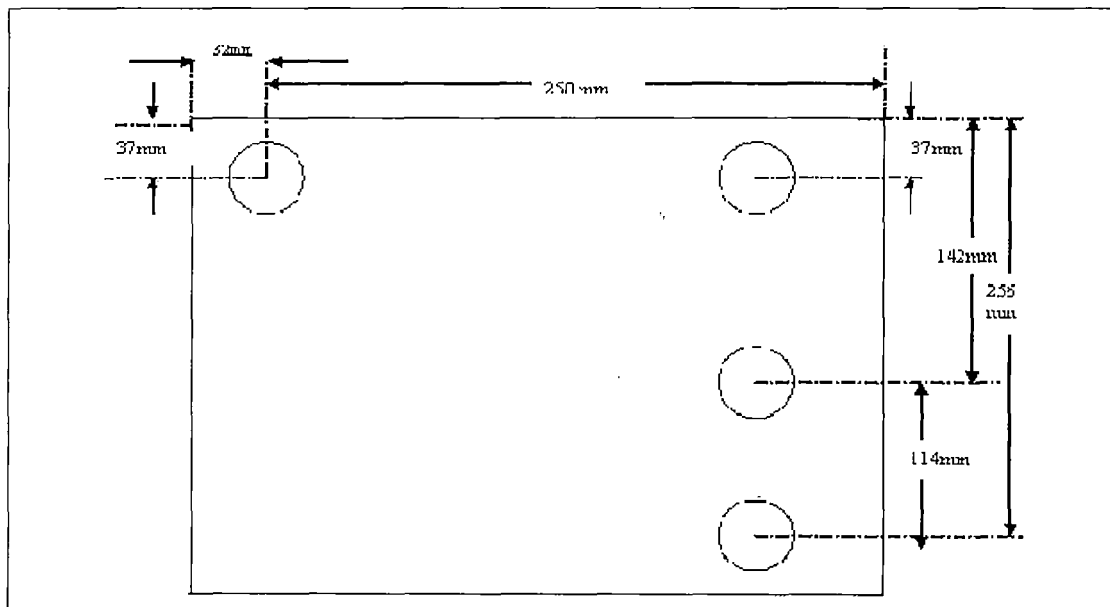


Figure 9. Static L-Shaped Marker System

There are a total of four static spherical markers of 1 cm diameter attached to a non-reflective plate in an L-shaped configuration. The plate is attached to a tripod and placed in front of the cameras at approximately the same distance as the marker systems in such a way that all the markers can be viewed clearly by each camera. Once the static calibration is done, these markers are removed from the view of all cameras without changing the location of the cameras. The static calibration is done only once and

applies to all subsequent measurements as long as the camera position remains unchanged.

3.4.1.2 Dynamic Calibration Markers. The dynamic calibration marker system consists of a wand with two spherical markers of 1 cm diameter located 10 cm apart (Figure 10). This distance is used by the Vicon system to establish a camera-based coordinate system in the volume the operator defines by dynamically moving the wand in a specific pattern (details are explained in Chapter 4).

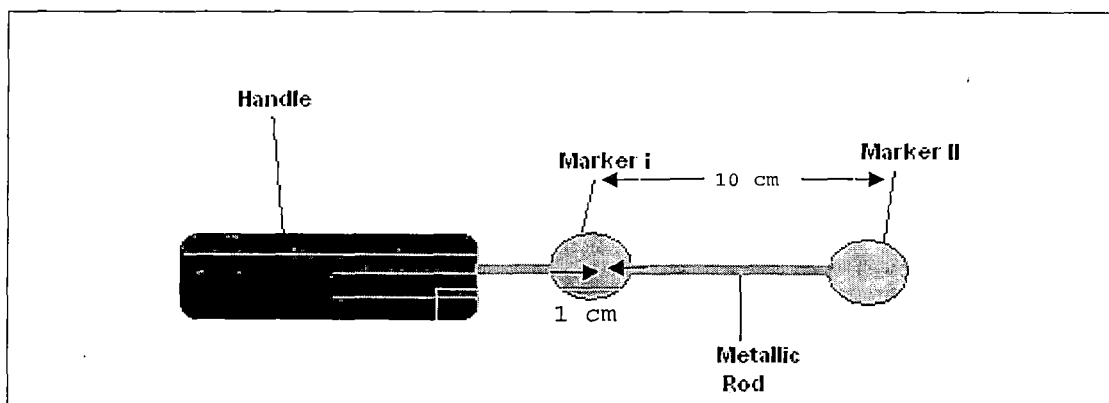


Figure 10. Wand Used for Dynamic Calibration

3.4.2 Caddy Markers and Halo

A collection of markers attached to a metallic frame called the marker caddy provides the patient-centered marker system. In the version used for the experiments of this thesis, there are 23 spherical markers of 5 mm diameter attached to the caddy (Figure 11). The caddy itself can be attached reproducibly to the halo affixed to

the patient's skull. The halo used for this purpose is the Leksell Coordinate Frame Model G from Elekta, the producer of the Leksell Gamma Knife. It is machined from a non-conducting metallic material to prevent it from causing magnetic disturbances during MRI scans. In order to uniquely locate the caddy marker system in space, at least three markers must be visible in at least two cameras through the entire range of PPS motion. However it is more desirable that all three cameras see at least three markers for the sake of increased accuracy. It is important that the markers stay fixed in reference to each other and the patient during the whole tracking procedure.

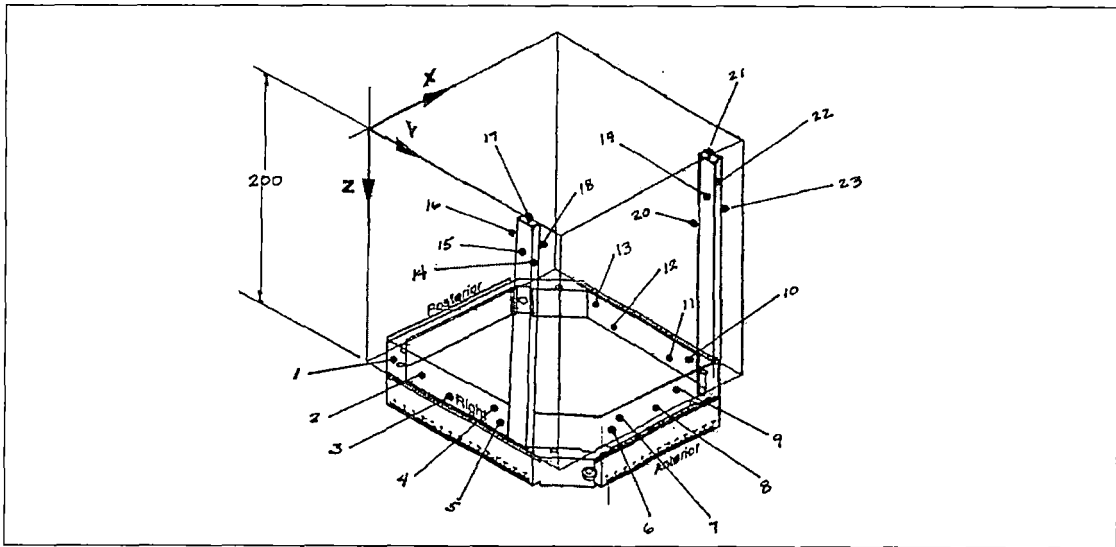


Figure 11. Marker Caddy and Halo Coordinate System

The marker caddy frame used for the experiments in this thesis consisted of the three major sides of a

cut-corner square frame with two posts, one on each side near the top of the frame as shown in Figure 11. The fourth side of the frame was eliminated so that the marker caddy can be slid on to the halo without disturbing fixtures that attach the halo to the head. The marker caddy frame reproducibly attaches to the halo by having pegs on the caddy inserted into four peg holes already present on the halo and then clamping the two pieces together, eliminating any relative motion between the caddy and the halo. The position of the markers within the halo-centered stereotactic coordinate system was determined by a certified dimensional inspection lab with ± 0.1 mm accuracy.

3.4.3 Cone Markers

The beam-centered marker system is attached to the proton beam delivery cone. It consists of a metallic cross-shaped plate with nine spherical markers of 5 mm diameter attached to it also in a cross-shaped configuration. Neighboring markers are at equal distances from each other.

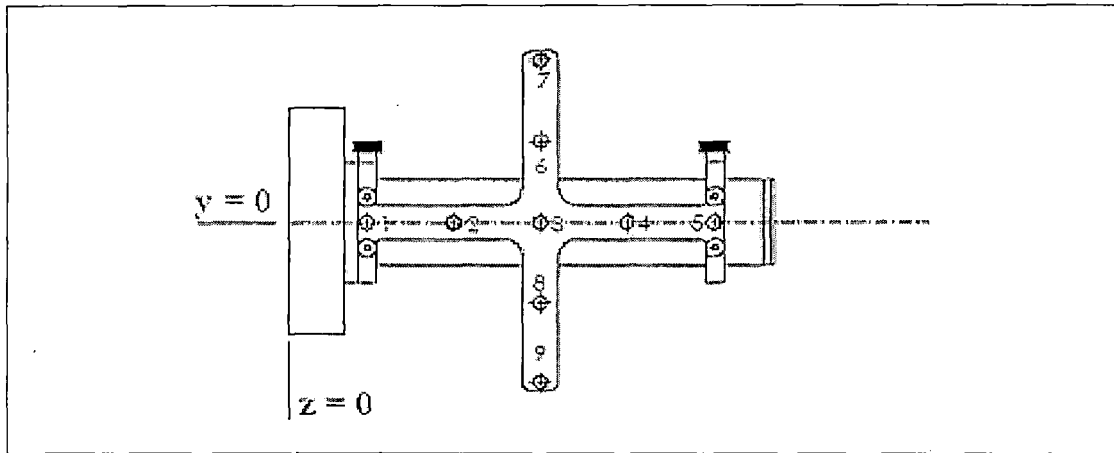


Figure 12. Marker Cross and Treatment Cone

The cross can be removed from the treatment cone but attaches reproducibly via two pegs. The position of the markers within a cone-centered coordinate system was determined by a certified dimensional inspection lab with ± 0.1 mm accuracy.

3.4.4 Phantom Base Markers and Reference Marker

Phantom base markers provide an independent reference that is used to measure the accuracy of the SAVPS. The spherical markers of 5 mm diameter are attached via pins of different length to a metallic phantom base plate, which in turn attaches the markers to the halo (Figure 13). The position of the markers within the halo-centered stereotactic coordinate system was determined by a certified dimensional inspection lab with ± 0.1 mm accuracy. Both reflective and non-reflective phantom base markers are in use. The reflective markers serve to verify

the performance of a stereotactic transformation from the halo coordinate system to the camera coordinate system. Applying the transformation to the known stereotactic position of these markers results in a prediction of their position in the camera system, this can be compared to the actual measurements. Non-reflective markers are used in 15 different locations in stereotactic space in order to measure the accuracy of the whole system. In this work, they were aligned perfectly with the laser beam providing the location of the proton beam axis, and the system error was measured by letting the system predict the actual position of the marker (which was invisible to the cameras) with respect to the beam axis.

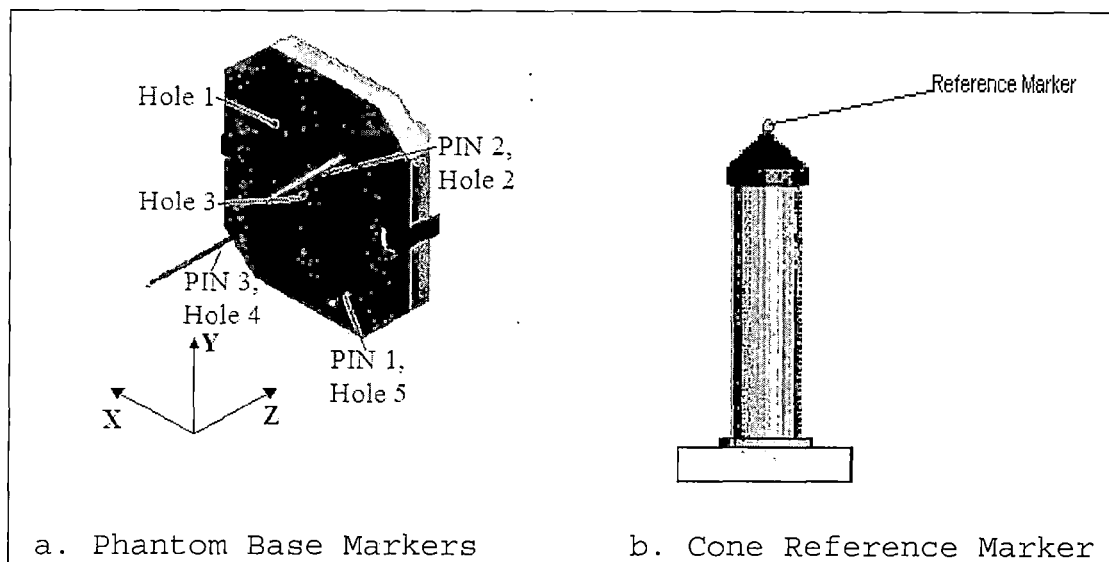


Figure 13. Phantom Base Markers and Reference Marker

A combination of five holes and three marker pins provides 15 different locations in stereotactic space.

The reference marker is a removable marker that can be placed at the top of the cone. The location of the cone reference marker is compared with DIL (local) coordinates value with the Vicon camera (global) coordinates. This comparison can be used to verify the validity of the coordinate transformation.

3.5 Treatment Cone

The treatment cone is a metallic, non-reflective cylindrical device (Figure 14) that directs and collimates proton radiation beams for the radiosurgery treatment. It also holds the marker cross as mentioned before.

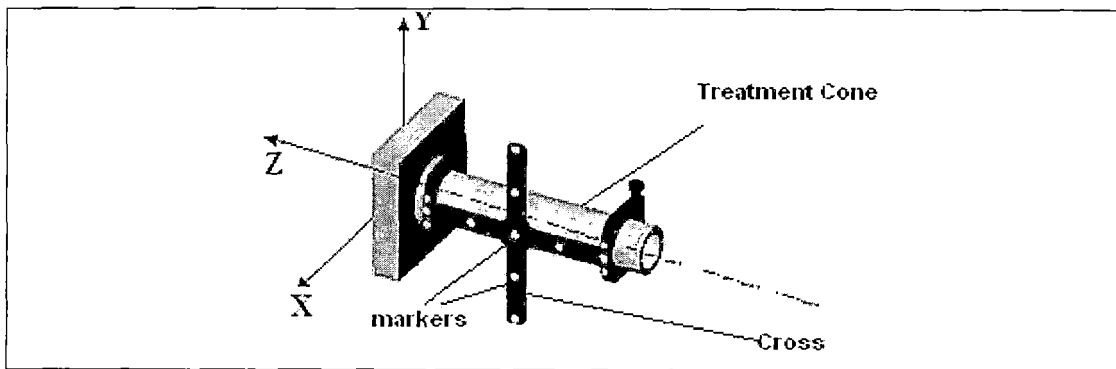


Figure 14. Treatment Cone and Marker Cross

The cone was provided with a laser insert that simulates a proton beam. The laser produces a circular

beam of 1 cm width that is aligned to one of the non-reflective phantom base makers.

3.6 Patient Positioning System

The Patient Positioning System (PPS) allows precise and accurate positioning of the patient within its specifications. The patient is positioned on a flat table that has 6 degrees of freedom (three orthogonal translations and three rotations) as well as specific software that communicates with the SAVPS software for driving the table motions.

3.7 Assistive Software Systems

In addition to the hardware components and the software that drives the PPS, there are several important software components that play a vital role in making the entire system function. The assistive software comprises Serial Communication Software, Image Processing Software, and Transformation Software.

3.7.1 Serial Communication Software

RS-232 is the external interface for the communication protocol between the positioner table and the computer. Since the operating system used is Windows NT, the serial communication between the computer and the table is more complicated.

In Windows NT, the serial communication can be done in two ways: overlapped or non-overlapped. Because of applicability and portability, we use Overlapped I/O written in Visual C++.

Overlapped I/O is not as straightforward as non-overlapped I/O, but allows more flexibility and efficiency. A port open for overlapped operations allows multiple threads to do I/O operations at the same time and perform other work while the operations are pending. Furthermore, the behavior of overlapped operations allows a single thread to issue many different requests and do work in the background while the operations are pending. The advantage of overlapped I/O is that it allows a thread to do some work between the time of the request and its completion.

3.7.2 Image Processing Software

This is software, written with the MATLAB software package, used to process digital images of the circular laser beam spot striking at a phantom base marker in order to find the distance between the centers of these objects. This distance (offset) was brought close to zero (within the accuracy of the digital image processing) and used as a measure of alignment accuracy and precision of the entire SAVPS. Details about this software and error

measurement are explained in Chapter 4 and in the Appendix.

3.7.3 Coordinate Transformation Software

This is the core component of the SAVPS, also written using the MATLAB software package. For this thesis various transformation methods were coded, namely: Orthogonal Transformation, Least-Square Based Transformation, and Constrained Least-Square Based Transformation. The main objective of the transformation software is to calculate the distance between the phantom base marker and the beam axis, which involves transforming local stereotactic coordinates into the global camera coordinates. Chapter 4 explains the different transformation algorithms in detail.

3.8 Summary

System components used for the SAVPS to determine the alignment error in proton radiosurgery are explained in Chapter Three.

CHAPTER FOUR
RESEARCH DESIGN AND PROCEDURE

4.1 Introduction

Radiosurgery is a non-invasive treatment technique applying focused radiation beams. It requires high geometric accuracy as misalignment can cause damage to the surrounding healthy tissues and loss of the therapeutic effect. One promising technique to insure submillimeter alignment accuracy of the radiation beam is to optically monitor the position of the beam axis relative to a frame firmly attached to the patient's skull using an optical alignment system. The optical alignment method requires three-dimensional coordinate transforms. The overall submillimeter accuracy could be achieved by following various procedural steps: Camera Calibration, Image Processing, Coordinate Transformation and Error Analysis. Each of the above steps are performed in the order they are listed, i.e. Camera Calibration must be the first operation to be completed followed by Image processing and analysis and finally Coordinate Transformation must be proceed by Error analysis. Figure 15 shows the experimental setup required for functional proton radiosurgery.

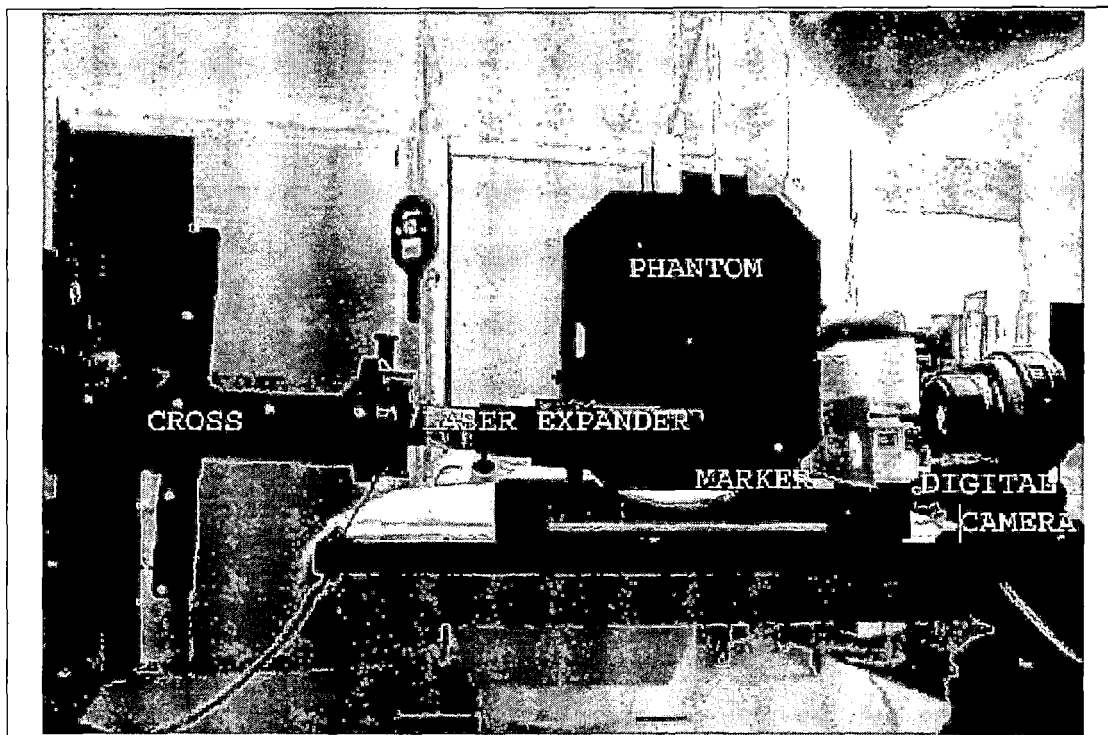


Figure 15. Experimental Setup

4.2 Theoretical Constructs and Assumptions

The main objective of this thesis is to develop a procedure for aligning the anatomical target with respect to the center of the proton beam with a new method that is more accurate and precise than the existing one. Though the accuracy of the system relies heavily on practicality and successful application of the system, the foundation of the system is constructed on some theoretical assumptions. The initial assumptions made regarding the systems are:

1. The caddy and cone fiducial marker sets captured by the Vicon cameras have a known position with

respect to the patients target and the beam delivery system (cone), respectively.

2. The proton beam is simulated by an expanded circular laser beam, which has a 10 mm diameter.
3. The center of the 5 mm spherical marker is surrogate for the anatomical target point in space with given halo coordinates.
4. Potential errors of target localization due to image distortion are not considered in this thesis.
5. The potential error introduced by the image processing algorithm to determine the actual offset between marker and beam axis is neglected.

4.3 Experimental Procedures

4.3.1 Camera Calibration

The process of initializing the cameras for the optimum visibility and to allow subsequent data capturing with high accuracy is known as "Camera Calibration". This is the first step of the experimental procedure. Before each experimental session, it was assured that the cameras were set up based on geometrical and physical layout considerations (Chapter 3.1). The next step was to check

for the visibility range of each camera within the 3D space representing the patient's head and the beam delivery system (cone). This was followed by the proper calibration procedure, which is a two-step process consisting of Static Calibration and Dynamic Calibration.

4.3.1.1 Static Calibration. By performing the Static Calibration process, the global Vicon coordinate system is defined with the help of L-shaped marker pattern described in Chapter 3. In order to perform static calibration, L-Shape Marker pattern was placed in front of the cameras as shown in the Figure 16.

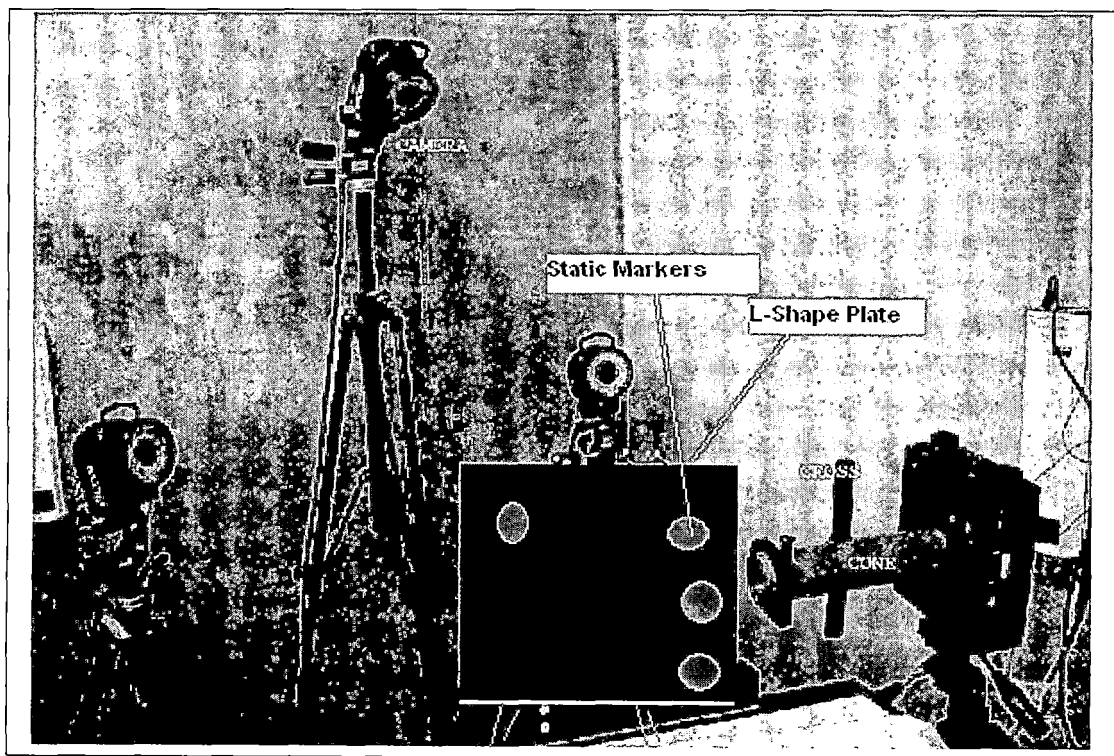


Figure 16. Static Calibration Setup

Once the static markers were placed such that all three cameras could see them, calibration data for the static calibration were captured by using the camera-computer user interface on Vicon's Workstation computer. Interface parameters such as "sensitivity" and "tolerance" were adjusted as needed. After performing static calibration, the markers-plate was removed from the sight of the camera without changing location and orientation of the camera.

4.3.1.2 Dynamic Calibration. The second step in the calibration process is Dynamic Calibration which was done immediately after Static Calibration. For the dynamic calibration, a wand with two reflective markers provided by Vicon was used (figure 17).

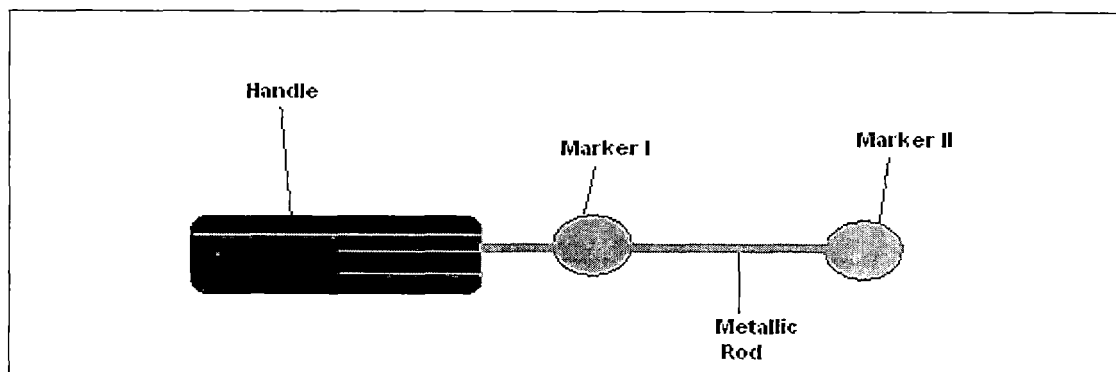


Figure 17. Dynamic Markers

Dynamic calibration was initiated from the Workstation. Before performing actual calibration, the range of view was checked for each camera by moving the

T-Shape markers through the marker detection volume. After determining the actual volume of calibration, dynamic calibration was started by pressing start button in the interface in the Workstation. As the start button is pressed, the wand is moved in random pattern as suggested by the system manufacturer, Vicon. Vicon suggested moving the wand in random pattern for less than 30 frames (figure 18) for better calibration results.

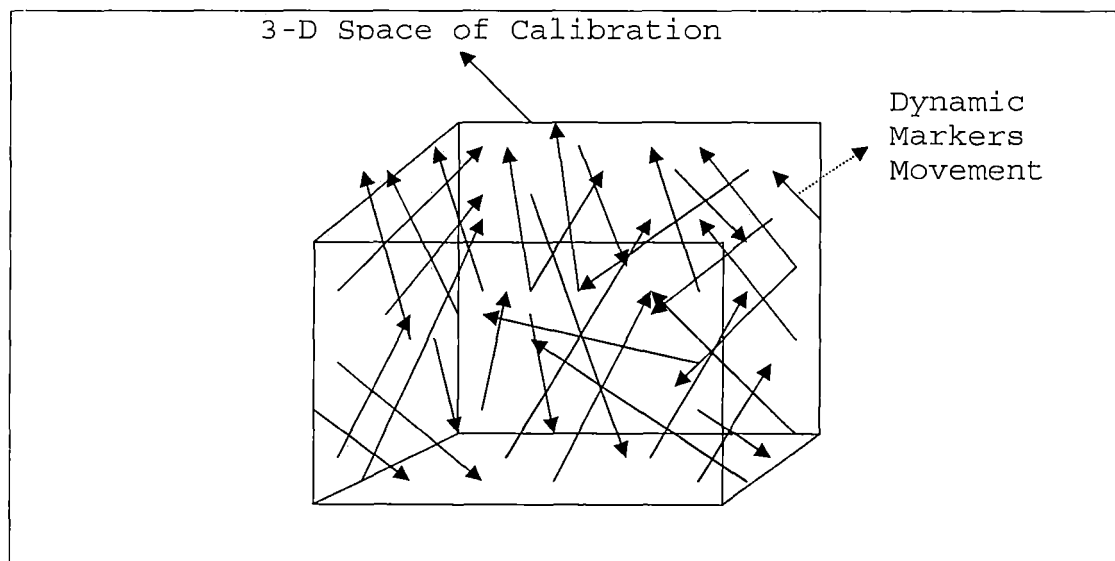


Figure 18. Random Pattern for Dynamic Calibration

The dynamic calibration results consisted of mean residual, visibility, and reproducibility were observed on the Workstation. For an acceptable calibration value, the mean residual should be less than 0.5 mm, visibility should be greater than 90%, and the reproducibility should be better than 1%.

After initial studies using the random pattern, it was found that the mean residual values were always greater than 0.5 mm and hence leading to an unacceptably large overall alignment error of the system. Hence, the need for the new calibration pattern was realized.

Two systematic dynamic calibration patterns were tested in an attempt to get consistently better calibration results, needed for an optimized data capturing process. These two dynamic calibration patterns were "Inward-Outward Circular Pattern" and "Vertical-Horizontal Pattern".

4.3.1.2.1 Inside-Outside Circular Pattern. This pattern is an "ice breaker", giving directions to developing new calibration patterns for a more efficient and successful camera calibration. In this calibration pattern, the wand is moved in spiral circular pattern toward and away from the isocenter (target) through the calibration volume. The wand was rotated clockwise while approaching the isocenter (target) and anti-clockwise while taking away it (figure 19), or vice versa.

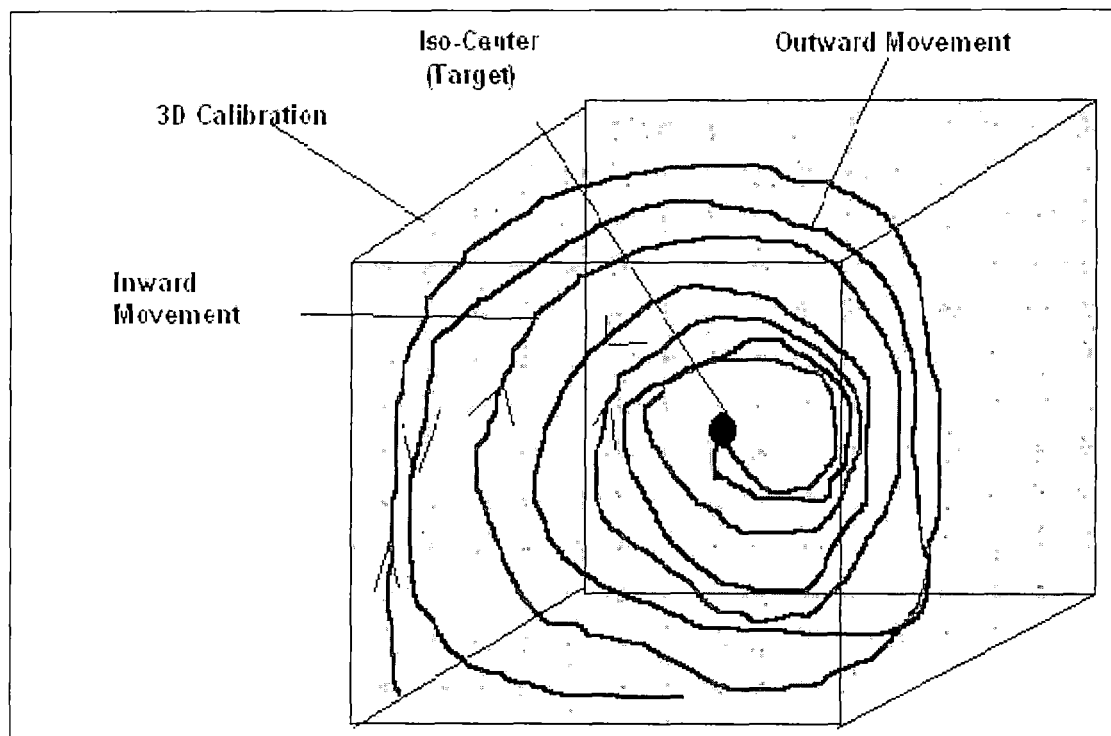


Figure 19. Inside and Outside Circular Pattern

Although, promising calibration results were observed by using this pattern, in the long run, they were still marred by outlying results, yielding average residuals around 0.8 mm, ranging from 0.3 mm to 1.5 mm.

4.3.1.2.2 Vertical-Horizontal Pattern. In an effort to overcome the inconsistent calibration results generated by Inward-Outward Pattern, a "Vertical-Horizontal Pattern" was introduced and tested. In this pattern, the wand is moved zigzag vertically towards the isocenter (target) and then moved horizontally away from the isocenter (target) as shown in the Figure 20.

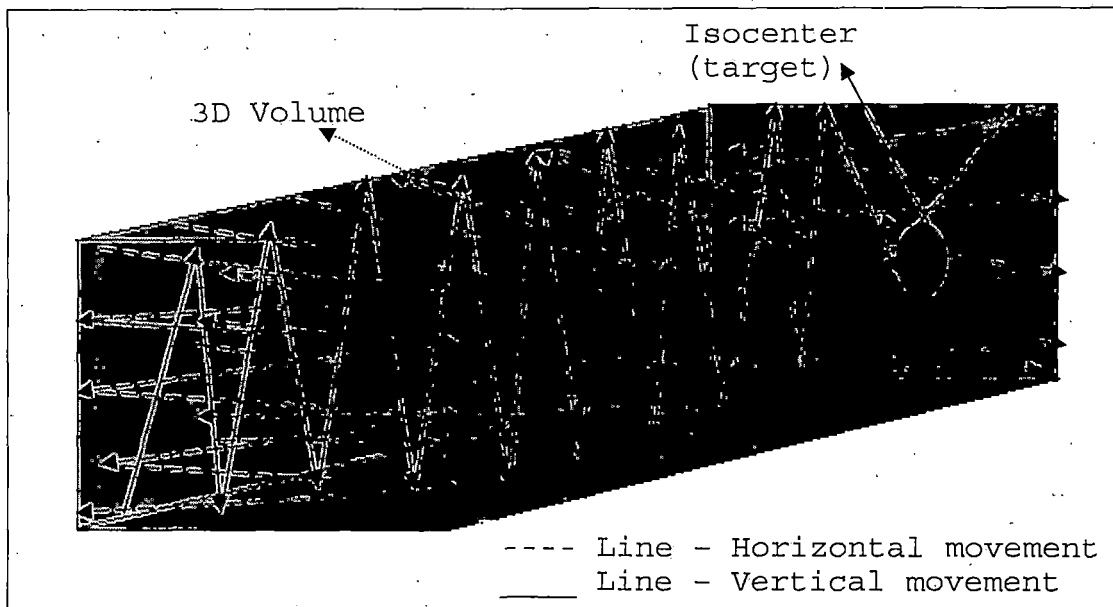


Figure 20. Vertical-Horizontal Pattern

With this pattern, calibration mean residuals were consistently less than 0.3 mm, visibility greater than 95% and reproducibility less than 1%, as desired. Therefore, this Horizontal-Vertical pattern was used for all subsequent experimental work and is also being suggested for future experimental and clinical work.

4.3.2 Image Processing System

Generally, an image is considered as an abstract of an object (living or non-living). In scientific terms, an image is considered as a continuous function of two or three variables. A *digital* image is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels. Pixels are stored in computer memory as the 2D matrices. Digital images can be

created by a variety of input devices and techniques, such as digital cameras, scanners, coordinate-measuring machines etc. In a sophisticated image processing system it is possible to apply specific image processing operations to selected regions. *Digital image processing* is the method of applying an algorithm to perform specific operation to an image or a selected region of an image.

A digital image $a[m, n]$ described in a 2D discrete space is derived from an analog image $a(x, y)$ in a 2D continuous space through a *sampling* process that is frequently referred to as "digitization". In fact, in most cases $a(x, y)$ might be considered represent a physical signal that impinges on the face of a 2D sensor, and, therefore, is a function of many variables including depth (z), color (λ), and time (t). An image could be further subdivided into smaller sections also known as segments.

Unless otherwise stated, we will consider the case of 2D, monochromatic, static images for the application described in this thesis. The principle of digitization process is shown in Figure 21 below.

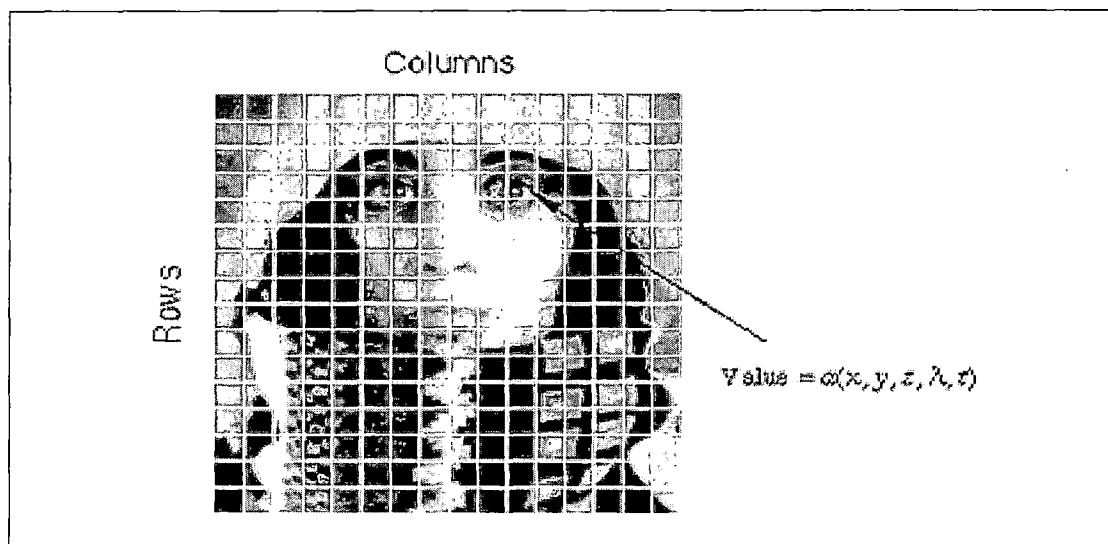


Figure 21. Digitization of a Continuous Image

The process of representing the intensity of the 2D signal at a given coordinate as an integer value with L different gray levels is usually referred to "quantization".

4.3.2.1 Image Representation Types. Images are classified into several representation types based on their compression format for portability and storage. These compressed image formats are: TIF, JPG, GIF, BMP, PNG and RAW.

TIF

TIF image format allows flexibility in terms of choosing compression or not. The compression used for .tif files is lossless. Whether the image is compressed or not, it will be of same quality as the original. Tif files are well

suited for virtually any type of image processing as they maintain image quality throughout the image processing.

JPG

JPG image format are compressed image files, but there are different levels of compression. JPG files are most commonly used for photographs. For this thesis work, this image format is used.

GIF

GIF image format uses a color palette with a fixed number of colors (256 colors to be exact). GIF files are typically used when there are no gradients and/or when there are a limited number of colors.

BMP

BMP is an uncompressed proprietary format invented by Microsoft. There is really no convincing reason to use this format.

PNG

PNG is probably the most flexible files but is not widely supported. PNG files use lossless compression and produce relatively small file sizes. These would be perfect for the internet but, for some reason, they are not fully supported by IE.

RAW

RAW is an image output option available on some digital cameras. Though lossless, RAW files are a factor of three or four smaller than TIFF files of the same image. The disadvantage is that there is a different RAW format for each manufacturer, and so one may have to use the manufacturer's own software to view the images. (Some graphics applications, however, can certain proprietary RAW formats.)

4.3.2.2 Types of Digital Images. In principle, there are two types of digital images— color and black and white. Color images are made up of colored-coded pixels while black and white images are made of pixels in different shades of gray.

Black and White Images

The pixels of a black and white image hold a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different grays.

Color Images

The pixels of a color image hold three numbers corresponding to the red, green, and blue levels of the image at a particular location. Red, green, and blue

(sometimes referred to as RGB) are the primary colors, which are different from the subtractive primary colors used for mixing paints (cyan, magenta, and yellow). Any color can be created by combining the correct amounts of red, green, and blue light. Assuming 256 levels for each primary additive color, each color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colors. Note that for images of the same size, a black and white version will use three times less memory than a color version.

Indexed Color Images

Some color images are created using a limited palette of colors, typically 256 different colors. These images are referred to as indexed color images because the data for each pixel consist of a palette index indicating which of the colors in the palette applies to that pixel. There are several problems with using indexed color to represent photographic images. First, if the image contains more different colors than are in the palette, techniques such as dithering must be applied to represent the missing colors and this degrades the image. Second, combining two indexed color images that use different palettes or even retouching part of a single indexed color image creates

problems because of the limited number of available colors.

Binary or Bi-level Images

Binary images use only a single bit to represent each pixel. The number of distinct gray levels is usually a power of 2, that is, $L = 2^B$ where B is the number of bits in the binary representation of the brightness levels.

When $B > 1$ we speak of a gray-level image; when $B = 1$ we speak of a binary image. Thus, in a binary image there are just two gray levels which can be referred to, for example, as "black" and "white" or "0" and "1". The inability of binary images to represent intermediate shades of gray limits their usefulness in dealing with photographic images.

4.3.2.3 Image Processing Algorithm. An Image Processing algorithm is a process that involves analyzing and manipulating images with a computer. Image processing generally involves three steps:

1. Import an image with an optical scanner or directly through digital photography.
2. Manipulate or analyze the image in some way. This stage can include image enhancement and data compression, or the image may be analyzed to find patterns that are not visible by the

human eye. For example, meteorologists use image processing to analyze satellite photographs.

3. Output the result. The result might be the image altered in some way or it might be a report based on analysis of the image

For the purpose of this thesis, an image processing algorithm was developed to measure the distance between the Phantombase marker and the axis of the laser beam. Using the PPS, the phantombase spherical marker was aligned to the center of the laser beam expanded to a circular beam of 10 mm diameter. The laser beam projected a shadow of the marker onto a flat screen, which was captured with a digital camera (Figure 22).

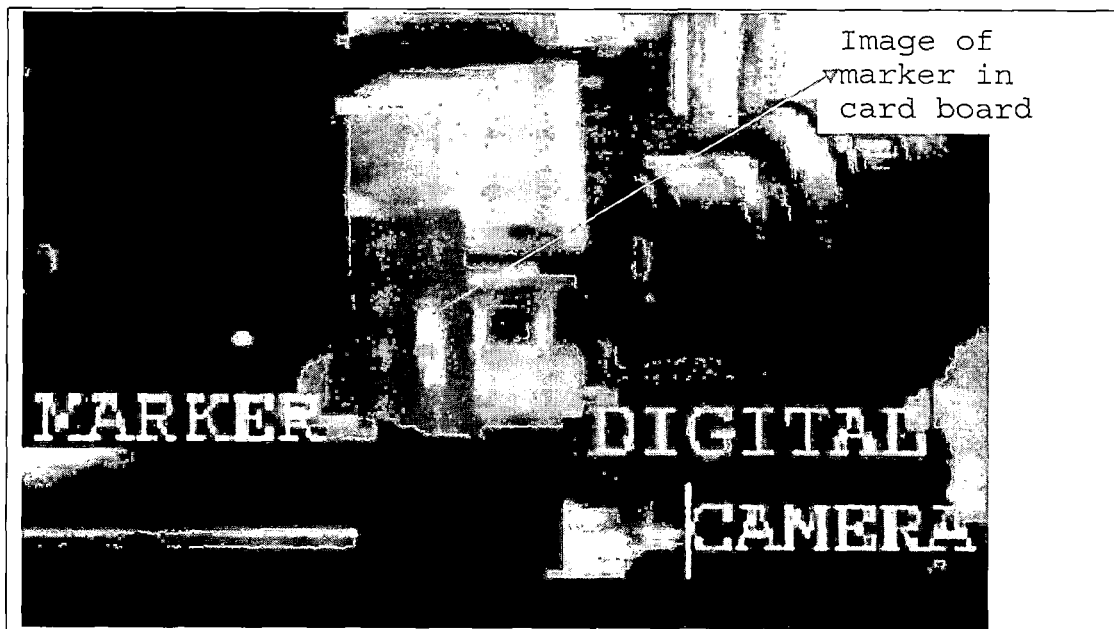


Figure 22. Processing of Alignment and Image Capturing

The captured digital image was supplied as the input to the image processing system. The image processing algorithm that existing at the time this work was started is outlined in Figure 23. The output of the program, which is the submillimeter distance between marker and laser beam axis, was supplied as input to calculate overall alignment error.

```
Read the image file  
Threshold the image  
Trace the boundary by initializing a point in the image  
Fit a circle to the boundary
```

Figure 23. Previous Image Processing Algorithm

One goal of this thesis was to improve the image processing system. In the previous algorithm, a "conventional" way of thresholding the image and tracing the boundary around the beam spot and marker in between the spot was used. Further, a manual calculation had been used to determine the offset between two circles fitted around the marker and the beam spot.

In the improved algorithm, a bisecting method and the brightness of the pixels surrounding the beam spot were used to estimate the contour segmenting the marker and the

beam spot more accurately. The outline of the new improved algorithm is shown in Figure 24.

1. Convert the RGB image to Gray scale image.
2. Determine the threshold between the foreground and background.
3. Find Right boundary
4. Find Lower and Upper boundary.
5. Find right side of the outer circle
6. Initialize maximum sector missing from outer circle and set no. of points to be fitted into the outer circle.
7. Set the tolerance values
8. While diameter is less then the tolerances then trace the boundary over the half outer circle.
9. Calculate the center of the outer circle using the parameters calculated by using least-square method
10. Use the center point to determine the radius of the circle.
11. Start at the outer circle center and determine the next true continuous bright region.
12. Initialize maximum sector missing from the inner circle and set no of points to be fitted into the inner circle.
13. Set the tolerance value
14. While the tolerance is less than desirable (preset) values then trace the boundary over the half inner circle.
15. Calculate the center of the inner circle using the parameters calculated by using least-square method
16. Use the center point to determine the radius of the circle.
17. Using both centers of inner and outer circle draw circles to fit the traced contours.
18. Calculate the distance between the centers

Figure 24. Image Processing Algorithm Developed in this

Thesis

The improvement resulting from using this algorithm and its advantages become obvious in the results presented in Chapter 5.

4.3.3 Stereotactic Coordinate Transformation for Functional Radiosurgery

4.3.3.1 Introduction. The coordinate system is a common tool for recording the location of a point or object in space. Coordinates transformations are used to calculate the coordinates of a point or an object in different coordinate systems, through translating, rescaling, rotating, or reflecting, without altering its desired geometrical properties.

Theoretically speaking, there are three processes that are involved in a coordinate transformation:

1. Translation of axes or change of origin
2. Change of scale;
3. Rotation of the axes.

Accurate stereotactic proton beam delivery for functional radiosurgery procedure requires a mathematical transformation of coordinates from two local coordinate systems, stereotactic coordinate system of the patient and the beam centered cone coordinate system, which change position in space during a treatment session, to a room

fixed global coordinate system defined by Optical Positioning System (OPS).

Local coordinates of the various markers were measured by a certified Dimensional Inspection Laboratory (Dimensional Metrology Laboratory, Riverside, CA, USA), and are therefore called DIL coordinates. In particular, the location of each caddy marker and phantom base marker was measured in the stereotactic coordinate system of Leksell G frame, and the location of each cross marker in the cone system were measured the DIL to an accuracy of better than ± 0.1 mm.

There are various mathematical methods of stereotactic transformation for the sequential alignment and positioning verification system for functional proton radiosurgery. Out of these, the following three methods were selected for the investigations in this thesis work: Orthogonal Transformation, Least-Square Based Transformation and Constrained-Least Square Based Transformation. These will now be discussed in more detail.

4.3.3.2 Orthogonal Transformation.

4.3.3.2.1 Introduction. A coordinate transformation is a useful device in ordinary 2D geometry. The same method can be established in d-dimensional space

($d \geq 2$). Consider a point with coordinates $x = (x_1, x_2, x_3, \dots, x_d)$ in one coordinate system and $y = (y_1, y_2, y_3, \dots, y_d)$ in a second system. Then the coordinate transformation between the two systems is given by

$$y_i = \sum_{j=1}^d l_{ij} x_j + a_i \text{ where } i = 1, \dots, d$$

where $a = (a_1, \dots, a_d)$ is a translation vector and $L = [l_{ij}]$ is a rotation matrix.

Consider only the rotation part of this transformation

$$y_i = \sum_{j=1}^d l_{ij} x_j$$

If $\sum_{j=1}^d l_{ij} l_{jk} = \begin{cases} 1, & i=k \\ 0, & i \neq k \end{cases}$ then the rotation is said to be

orthogonal.

4.3.3.2.2 Implementation. In general, the axes of the different coordinate systems will not be parallel with respect to each other. Therefore, the coordinate transformations mapping each point of one reference system into another one involves both translations and rotations as mentioned above. At least three linearly independent points, i.e., points that are not located on one straight line, with known coordinates in both coordinate systems are needed to uniquely define the coordinate transformation between the two systems.

For the SAVPS, the task is finding the transformation that maps the local marker set onto the global marker set. The Orthogonal transformation used includes three major transformational steps:

1. Find matrix making local triangle of three well visible markers parallel to the corresponding global triangle.
2. Find matrix aligning vertices of the transformed coplanar local triangle with the vertices of the global triangle.
3. Apply translation to collinear local triangle to make it coincident with global triangle.

These steps were applied to two well-visible marker triangles of caddy and cross and the transformation results (matrix and vector) were averaged to improve the statistics of the estimation. The mathematical characteristic of the orthogonal coordinate transformation was a motivation factor, which influenced our decision to use orthogonal coordinate transformation for the functional proton radiosurgery. In particular, orthogonal coordinate transformation preserves the length between objects and points even for larger distances.

The mathematical details of the orthogonal transformation used in this thesis are summarized in the Appendix B.

4.3.3.3 Least-Square Based Coordinate Transformation.

4.3.3.3.1 Introduction. Least square

minimization (LSM) is a time honored parameter estimation procedure that has been in use since the early nineteenth century. It is, for example, the most widely used technique in geophysical data analysis. Unlike maximum likelihood, which can be applied to any problem for which we know the general form of the joint Probability density function (PDF), in the LSM, the parameters to be estimated must arise in expressions for the means of the observations. When the parameters appear linearly in these expressions, as is the case here, the least squares estimation problem can be solved in closed form, and it is relatively straightforward to derive the statistical properties for the resulting parameter estimates.

In the least-squares problem, a function $f(x)$ that is a sum of squared residuals is minimized.

$$\min_{x \in \mathfrak{R}^n} f(x) = \frac{1}{2} \|F(x)\|_2^2 = \frac{1}{2} \sum_i F_i(x)^2$$

Problems of this type occur in a large number of practical applications, especially when fitting non-linear model

functions to noisy data, i.e., nonlinear parameter estimation. They are also prevalent in control where one want the output, $y(x,t)$ to follow some continuous model trajectory, $\phi(t)$, for vector x and scalar t . This problem can be expressed as

$$\min_{x \in \mathfrak{R}} n \int_{t_2}^{t_1} (y(x,t) - \phi(t))^2 dt$$

Where $y(x,t)$ and $\phi(t)$ are scalar functions.

When the integral is discretized using a suitable quadrature formula, above equation can be formulated as a LS problem as:

$$\min_{x \in \mathfrak{R}} f(x) = \sum_{i=1}^m (\bar{y}(x,t_i) - \bar{\phi}(t_i))^2$$

where \bar{y} and $\bar{\phi}$ include the weights of the quadrature scheme. Note that in this problem the vector $F(x)$ is:

$$F(x) = \begin{bmatrix} \bar{y}(x,t_1) - \bar{\phi}(t_1) \\ \bar{y}(x,t_2) - \bar{\phi}(t_2) \\ \dots\dots\dots \\ \bar{y}(x,t_m) - \bar{\phi}(t_m) \end{bmatrix}$$

In problems of this kind, the residual $\|F(x)\|$ is likely to be small at the optimum since it is general practice to set realistically achievable target trajectories.

If the general linear equation function is given by:

$$Y = AX + C$$

the solution is unique if rank (A) = number of column in A.

However, regardless of the rank of A there is always a unique minimal 2-norm solution to the LS problem given QR factorization method as:

$$X = A \setminus C$$

For linear models, the LS minimization is usually done analytically using calculus. For nonlinear models, on the other hand, the minimization must almost always be done using iterative numerical algorithms.

In LSM problem there is an underlying assumption that all the errors are confined to the observation vector C. Unfortunately, this assumption is frequently unrealistic; sampling errors, human errors, modeling errors and instrument errors may preclude the possibility of knowing the data matrix X exactly.

4.3.3.3.2 Implementation. The procedure of least squares (LS) minimization was used as an alternative

method to Orthogonal Transformation described above to establish a coordinate transformation between local and global reference systems. Here, the LS method minimizes the sum of the squares of the residuals after transformation of all available markers resulting in the best estimate of the value of the unknown coefficients of the transformation matrix.

Global and Local coordinates were used as the input data for the general equation required for the LS based parameter estimation. The transformation matrix was then found using QR factorization methods.

The mathematical details of the implementation of the LS Based Transformation in this thesis are in the Appendix C.

4.3.3.4 Constrained Least Square Based Transformation.

4.3.3.4.1 Introduction. Least squares minimization with a quadratic inequality constraint- the LSQI problem - is a technique that can be used whenever the solution to the ordinary Least Square problem needs to be regularized. A simple LSQI problem that arises when attempting to fit a function to noisy data is:

$$\text{Minimize } || A.x - B ||_2, \text{ subject to } || B.x ||_2 \leq \alpha$$

Where $A \in \mathfrak{R}^{m \times n}$, $b \in \mathfrak{R}^m$, $B \in \mathfrak{R}^{n \times n}$ (nonsingular), and $\alpha \geq 0$.

In practice, it is often necessary to impose constraints on a LS solution. For example, in curve fitting, inequality constraints may arise from such requirements as monotonicity, non-negativity, and convexity.

Equality constraints, on the other hand, may arise from the need to guarantee continuity (and possibly smoothness) of the curves. One popular class of such constraints is linear-equality constraints; that is, the solution x to above equation has to satisfy the following system of linear algebraic equations

$$Ax = B$$

These constraints defines a hyperellipsoid in \mathfrak{R}^n and are usually chosen to damp out excessive oscillation in the fitting function. This can be done, for example if B is a discretized second derivative operator.

More generally, in equality-constrained LS we have the problem

$$\text{Minimize } \|Ax - b\|_2 \text{ subject to } \|Bx - d\|_2 \leq \alpha$$

Where $A \in \mathfrak{R}^{m \times n}$ (with $m \geq n$), $b \in \mathfrak{R}^m$, $B \in \mathfrak{R}^{p \times n}$, $d \in \mathfrak{R}^p$, and $\alpha \geq 0$.

The generalized singular value decomposition of A and B, then transforms to

$$\text{Minimize } \|D_A y - \bar{b}\|_2 \text{ subject to } \|D_B y - \bar{d}\|_2 \leq \alpha$$

Where \bar{b} and \bar{d} is solved by using the U and V parameter from the Singular value decomposition method and $y = X^{-1}x$.

The simple form of the objective function is given by:

$$\|D_A y - \bar{b}\|_2^2 = \sum_{i=1}^n (\alpha_i y_i - \bar{b}_i)^2 + \sum_{i=n+1}^m \bar{b}_i^2$$

and the constraint equation

$$\|D_B y - \bar{d}\|_2^2 = \sum_{i=1}^r (\beta_i y_i - \bar{d}_i)^2 + \sum_{i=r+1}^m \bar{d}_i^2$$

facilitate the analysis of the LSQI problem. Here, $r = \text{rank}(B)$ and \bar{d} = diagonal value. The final solution for the general equation is determined by finding root of the equation i.e. value of α . The above literature about the Constrained Least Square method is summarized from Gene H. Golub and Charles F. Van Loan's "Matrix Computations," [11].

4.3.3.4.2 Implementation. Based on the application dealt with in this thesis, an equality based constraint was forced to the generalized LS estimation method.

The core task during implementation was setting the objective functions and constraint function i.e. orthogonality condition for the transformation, based on

available parameters (for details see Appendix D). After determining these functions and forcing the orthogonal constraint to the constraint function, available Matlab library was applied to calculate the desired parameters. The orthogonal constraint is forced as follows:

$$f = X'.X - I = 0$$

where X is the 3 X 3 matrix which describes the rotational part of the transformation, and I is the identity matrix of the same size as X.

The algorithm is given in the figure 25.

1. Construct the objective function individually to X and Y parameters.
2. Set the initial starting point for x_0 = value of X generated by general Least Square Based estimation.
3. Set all the optional parameters as empty (if needed it can be re-set to desired value)
4. Construct the Constraint Equation forcing orthogonality to the desired value.
5. Call a minimum of a constrained nonlinear multivariable function by passing all the required parameters.

Figure 25. Inequality Based Constrained Least Square Based Algorithm

The Matlab implementation of the constrained LS algorithm is explained in detail in the Appendix D.

4.3.4 Error Estimation

4.3.4.1 Introduction. Error estimation was an important part of this thesis work. It involved, on a

lower level, distance errors between the marker positions, which have to be minimized, for example, by appropriate camera calibration, and, on the highest level, the alignment error between the laser beam and the Phantombase marker.

4.3.4.2 Distance Error Estimation. Distance verification is a pre-test for the resulting accuracy of the alignment. The marker distances derived from global coordinates coming from the camera system are compared to distances among local coordinates, which were measured by the DIL. DIL values are the gold standard for the measurements in this thesis.

There are three "structures" that were used within the distance verification algorithm. First, "Point" is the class containing the coordinates of the marker in space. Second, for each marker, there is a Vector structure that has the dimension of the number of frames captured by the camera pertaining to the particular marker. Each frame has x, y, z coordinates respectively that fills in the coordinates in "Point" structure. Third, there is a second Vector structure that contains all the markers in the system. Later, for each marker in space an average set of coordinates is obtained by averaging arithmetically over the number of frames.

The geometrical illustration of distance vector d calculation is shown in Figure 26.

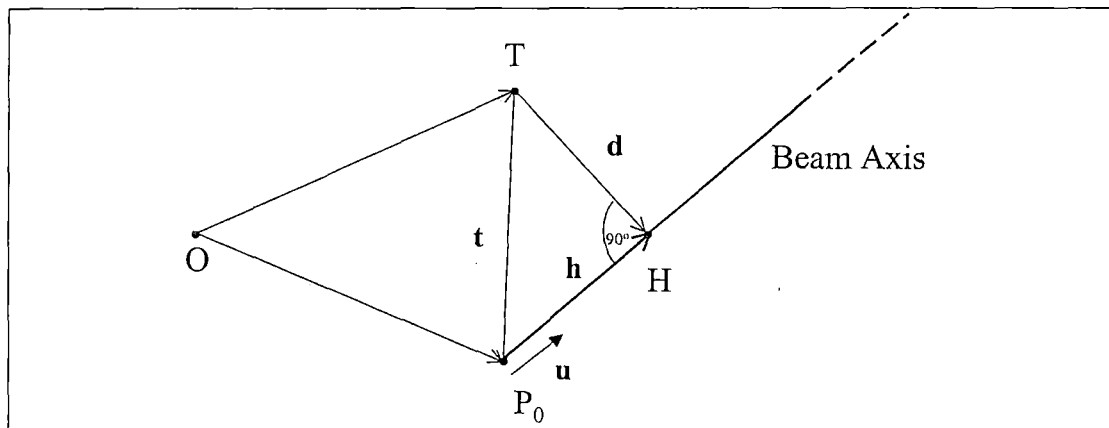


Figure 26. Geometry of the Shortest Distance Between Target and Beam Axis

The following consideration is in the local stereotactic coordinate system. Given are the origin O of the global coordinate system, the target point T , and the unit vector of the beam axis u . Furthermore, we know the vector t , which is defined as $t = P_0 - T$, where P_0 and T are the position vectors of the points P_0 and T from the origin O , respectively. Since vectors d and u are orthogonal, their inner product is zero:

$$d \cdot u = 0;$$

and, with $d = h + t$ and, we have

$$(h + t) \cdot u = 0$$

$$h \cdot u + t \cdot u = 0$$

$$h = -t \cdot u$$

Where $h = h \cdot u$ is the magnitude of the vector h . Since h is collinear with the vector u , we can write,

$h = hu = -(t \cdot u)u$, and finally

$$d = - (t \cdot u) u + t$$

which is the shortest vector between target and axis, i.e. the solution of the distance problem.

CHAPTER FIVE

RESULTS AND ACHIEVEMENTS

5.1 Calibration Errors

The researcher of the thesis preceding mine found the camera calibration to be less enticing when trying to improve the overall system. He followed the conventional way of analyzing a system as a whole, i.e., monitoring it as a single block. The downfall of that approach was that it yielded only the overall result without analyzing possible sources of error and options for improvement.

After conducting extensive research initially, I found that there is a need for analyzing the overall efficiency of the system in a stepwise fashion. The major advantage of analyzing it in this way is provides sources of error at every procedural step and leads to a better solution reducing the overall SAVPS error.

As a first step of the optimization, Camera Calibration results produced by the random calibration pattern suggested by Vicon (Figure 18) were extensively analyzed. The value of camera parameters given by the Vicon Workstation interface, such as mean residual, visibility, and reproducibility, were consistently above

the desirable values as shown in the Table 1, and thus not acceptable.

Table 1. Calibration Results Produced by Random Pattern

Experimental Runs	Mean Residuals (mm)	Camera Visibility (%)	Reproducibility
1	0.867	92	0.60
2	0.90	91	0.73
3	0.67	94	0.52
4	0.850	93	0.412
5	1.43	91	0.64
6	1.731	90	0.66
7	0.838	93	0.79
8	0.910	92	0.70
9	1.651	93	0.69
10	0.740	94	0.60

Initial analysis lead to the breakthrough conclusion that, out of the three camera calibration parameters, the mean residual values were most inconsistent with the desired value which should be < 0.5 mm, whereas the reproducibility parameter was always less than 1 as desired. This is illustrated graphically in Figure 27.

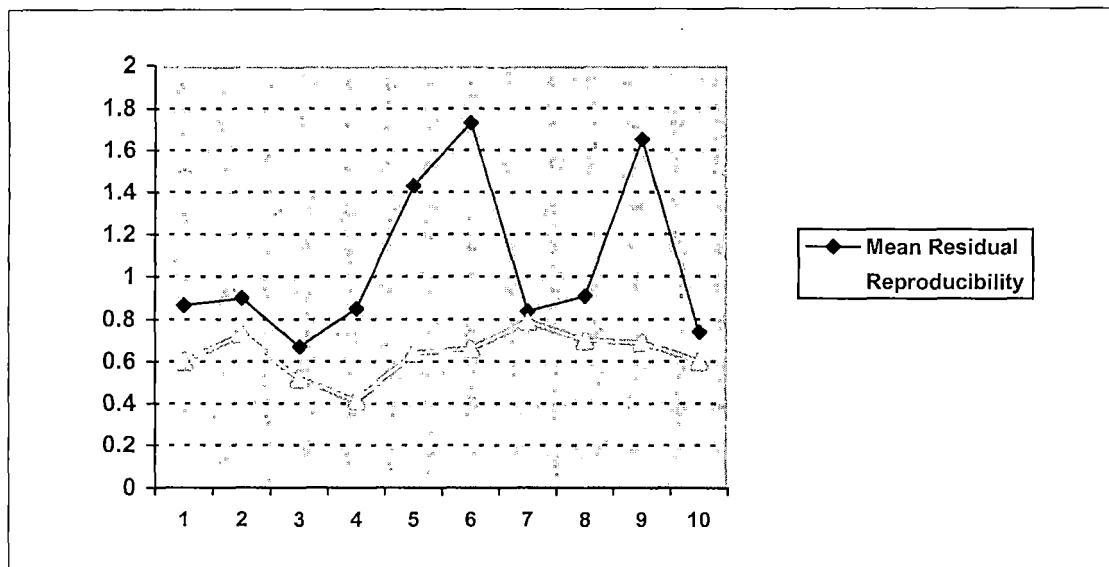


Figure 27. Mean Residuals and Reproducibility for the Random Pattern

The same held true for the visibility parameter, which always remained above the desirable limit of 90%. I concluded from this data analysis, that the mean residuals depend mostly upon the method of Wand movement during the dynamic calibration. The validity of the wand movement patterns described in Chapter 4 was further justified by observing the calibration parameters produced during the calibrations using these patterns.

5.1.1 Inward-Outward Circular Pattern Results

The mean residuals observed during the camera calibration using Inward-Outward Circular Pattern described in Chapter 4 (Figure 19) clearly showed an improvement compared to the random pattern. This pattern

yielded more stable and consistent results. Table 2 and Figure 28 show the calibration parameters produced during 10 experimental runs with this pattern.

Table 2. Inward-Outward Circular Pattern Calibration Results

Experimental Runs	Mean Residuals (mm)	Camera Visibility (%)	Reproducibility
1	0.93	91	0.72
2	0.821	96	0.50
3	0.316	95	0.51
4	0.304	94	0.62
5	1.51	92	0.80
6	0.82	96	0.76
7	0.79	91	0.83
8	0.66	95	0.57
9	0.73	96	0.60
10	0.89	92	0.66

This pattern yielded the mean residuals between nearly 0.8 and 0.31, while the other two parameters remained consistent as usual. The consistency of this pattern during 10 experimental runs can also be observed in Figure 25.

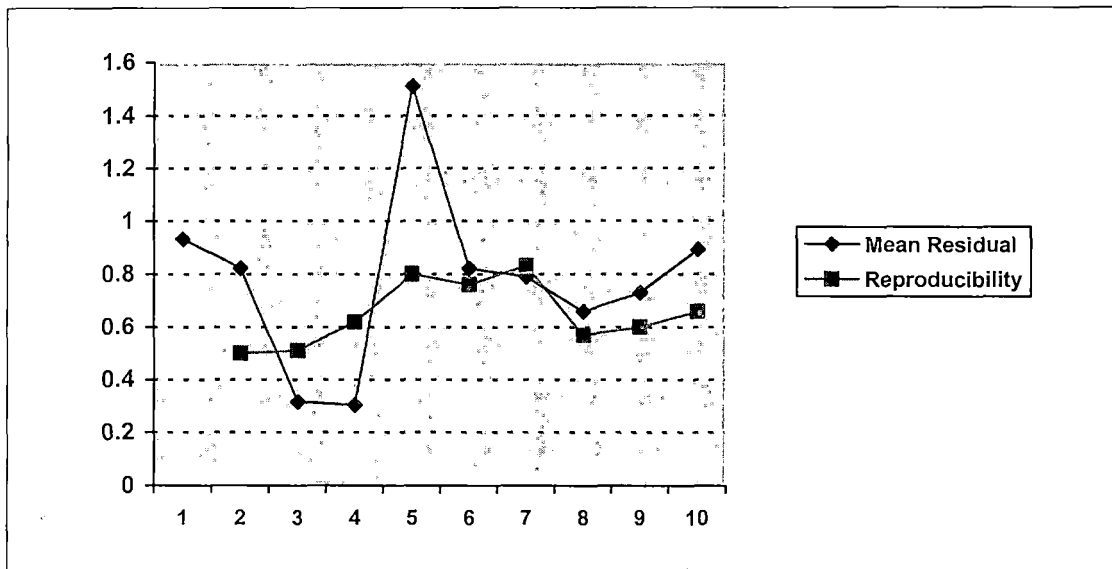


Figure 28. Mean Residuals and Reproducibility for the Inward-Outward Circular Pattern

5.1.2 Vertical-Horizontal Pattern Results

This method was another milestone in the process of improving the camera calibration results. The calibration results produced by this pattern were consistent and were always close to the ideal value. These parameters are listed in Table 3 and displayed in Figure 29.

Table 3. Calibration Parameters Using Vertical-Horizontal Calibration Pattern

Experimental Runs	Mean Residuals (mm)	Camera Visibility (%)	Reproducibility
1	0.455	90	0.80
2	0.405	92	0.53
3	0.262	96	0.56
4	0.304	94	0.68
5	0.255	92	0.50
6	0.455	91	0.81
7	0.255	97	0.53
8	0.257	96	0.59
9	0.322	94	0.64
10	0.273	93	0.68

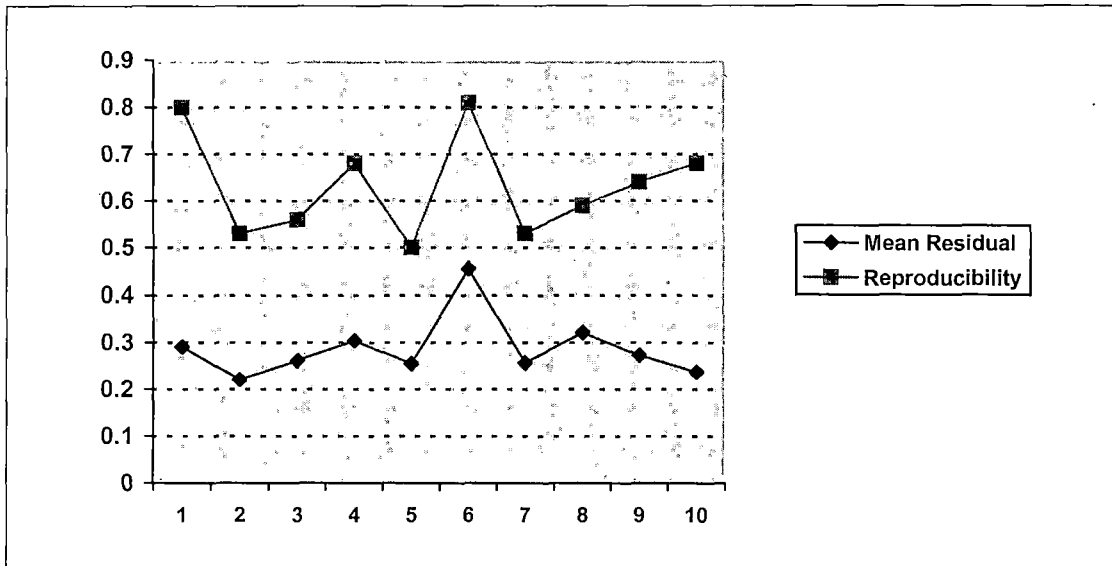


Figure 29. Mean Residuals and Reproducibility for the Horizontal-Vertical Pattern

Figure 29 also showed that fluctuations in the value of the mean residual paralleled that of the reproducibility. All three parameters during different experimental runs remained consistent by producing values nearer to the ideal values. The significant advantage of this pattern was its ability to reproduce good results.

Hence, this pattern is recommended for all future dynamic calibrations which can constantly produce better results necessary for best overall performance of the system.

5.2 Improved Image Processing Algorithm

The image processing system used in this work takes an image of the laser beam spot and marker shadow

projected a white screen as input. The output produced by the system is the processed image along with vertical and horizontal distances between marker center and beam center.

In order to illustrate the improvement of the novel approach for the image processing system, it is necessary to show an output image produced by the preceding researcher. Figure 30 shows an image produced by the inefficient old image processing algorithm as explained in Chapter 4.

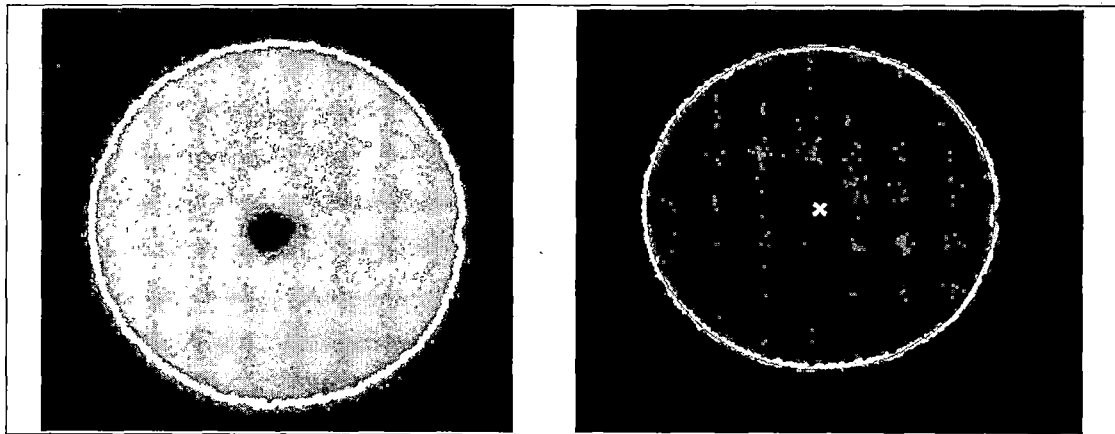


Figure 30. Border and Center Estimation Produced by the Previous Image Processing Algorithm

In particular, the previous image processing method was vulnerable to the error introduced during border and center estimation in three ways: First, the contour around the marker and laser spot was drawn without taking consideration to the distortions imposed by thin rod

holding the marker in space (note that this rod has meanwhile been reduced in diameter). Second, the number of points to be drawn around the circle was supplied manually during the processing, using a method of trial and error. Third, the offset between the centers of the beam-spot and the marker was calculated manually rather than within the algorithm. Since, the older version demanded greater user involvement during image processing; there was a greater chance of inducing human errors in the distance estimation.

In the new image processing algorithm, developed during this thesis work, the distortions produced by the marker post was observed and dealt with by restricting contour plotting to the regular part of the circular shapes, leaving the distorted parts untouched. The contours drawn form semicircular shapes and hence yield a more accurate estimate of the centers of the contours. Once centers had been calculated, best-fit circles were drawn based on the center estimates for visual crosscheck. The centers of the circular shapes were used to calculate the distance between marker and beam spot. The final result produced by this image processing algorithm is an image with best fit circles, distance offset and message to the user regarding PPS moving directions. The MATLAB

package was used to implement this algorithm taking the digital image as input and producing a processed image along with calculated distance offset as output.

An example of the input image supplied to the MATLAB image processing program is shown in Figure 31.

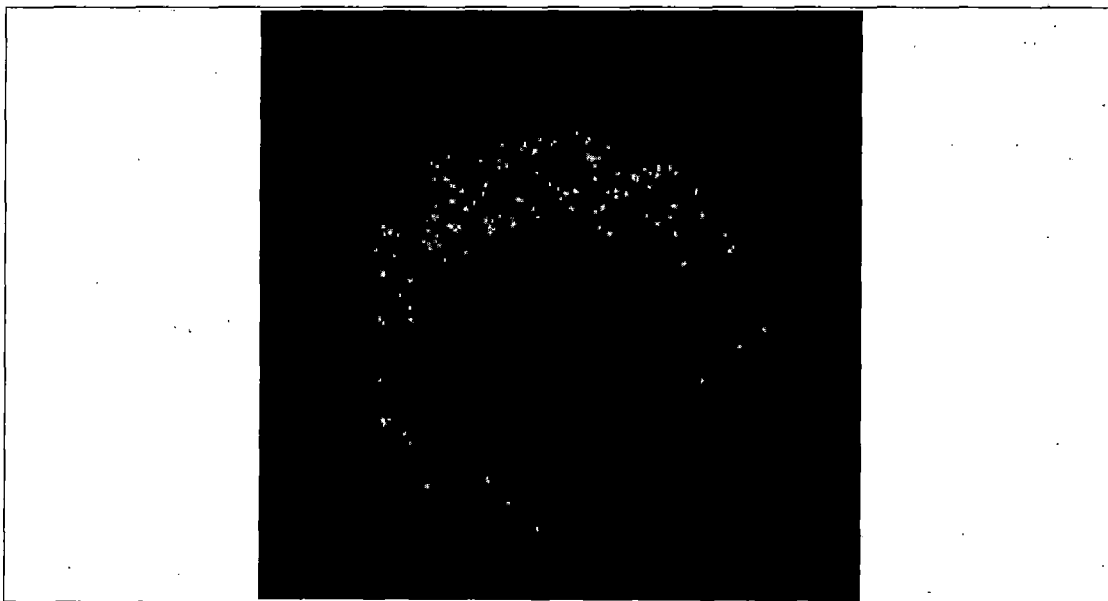


Figure 31. Input Image for the Image Processing System

The corresponding processed image with estimated contours and the centers is shown in the Figure 32.

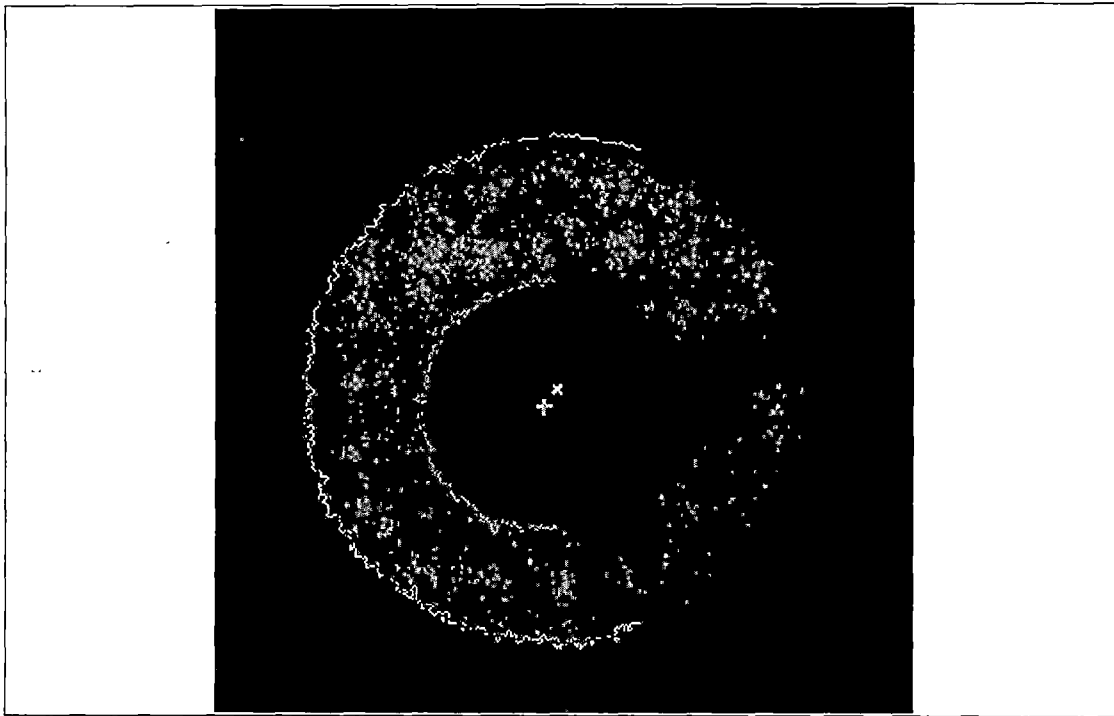


Figure 32. Processed Image Corresponding to the Image of Figure 31

This image processing algorithm not only supports a user-friendly and efficient system but also calculates the distance offset between the centers with five digits of accuracy. The output script in MATLAB is shown in Figure 33.

```
EDU>> Image has been Uploaded
Processing Outer Circle
Processing Inner Circle
distance =
    0.4045

TF_offset =
    0.2407

ZF_offset =
   -0.3251

table is near to the camera and TF_offset value need to
be subtracted from the current position
Table is lower than the beam and ZF_offset value need to
be added to the current position
EDU>>
```

Figure 33. Output Script Produced by the Matlab Image Processing System

5.3 Distance Error Calculations

The distances between pairs of cone and caddy markers were examined in order to estimate the errors introduced before the coordinate transformation. The error calculation was based on a comparison of camera-measured distance values and the corresponding DIL values, the latter considered a gold standard.

The means and standard deviations of caddy distance errors and cone distance errors observed during 15 different experimental runs are shown in Table 4.

Table 4. Distance Errors

Data Run	mean caddy distance error (mm)	SD caddy distance error (mm)	mean cone distance error (mm)	SD cone distance error (mm)
1	-0.10	0.28	0.01	0.07
2	-0.27	0.38	-0.03	0.14
3	-0.18	0.30	0.02	0.07
4	-0.26	0.25	0.03	0.08
5	-0.43	0.35	0.05	0.11
6	-0.24	0.29	0.00	0.07
7	-0.05	0.25	0.01	0.06
8	-0.42	0.33	-0.05	0.11
9	-0.50	0.34	-0.01	0.08
10	-0.20	0.23	0.00	0.08
11	-0.14	0.44	0.01	0.08
12	-0.12	0.28	-0.03	0.09
13	-0.26	0.31	-0.01	0.09
14	-0.17	0.27	0.00	0.08
15	-0.18	0.63	0.04	0.10
Average	-0.23	0.33	0.00	0.09

The average mean error induced by caddy markers is -0.23 mm as compare to zero error for the cone markers. This shows a small systematic error. The mean distance error for caddy markers ranges from -0.05 mm to -0.50 mm, whereas the mean distance error for cone remained well below 0.15 mm. Similarly, the average standard deviation of the caddy distance errors was 0.33 mm, while for cone markers it was only 0.09 mm. The fluctuation between distance errors of caddy and cone markers during 15 different experimental runs is also plotted in Figure 34.

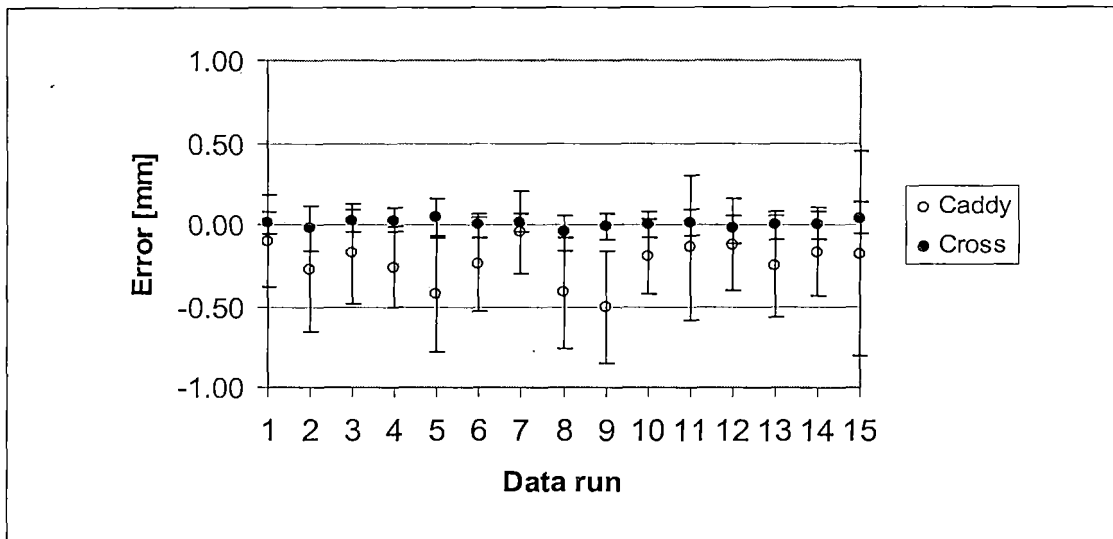


Figure 34. Caddy and Cone Markers Distance Errors

The difference in the errors between caddy and cone markers for cone and caddy may be related to the asymmetrical distribution of caddy markers as well as their spread in three dimensions rather than two dimensions for the cone markers.

5.4 Comparison of Coordinate Transformations

5.4.1 Introduction

The comparison of different coordinate transformation methods was a core area of research for this thesis. Every transformation method used here for the SAVPS system had as a common objective the determination of the alignment error. This subsection discusses the results produced and milestones achieved when investigating orthogonal

coordinate transformation and least square (LS)-based transformation.

5.4.2 Orthogonal Coordinate Transformation

The orthogonal coordinate transformation was described by the researcher of the preceding thesis as unitary transformation, since it preserves the length of the transformed objects. A major milestone achieved in this thesis is porting the previous Mathcad (Version 11) implementation of the algorithm in to the more versatile and widely used mathematical Matlab package.

While porting older version of the program code into the new and more efficient version, a few new concepts were also introduced in order to correct for any systematic error induced during the operation. A Scaling Factor (SF) was implemented, which corrects systematic distance errors introduced by a non-unity scaling of the camera system. The distance errors observed using Matlab routine is used to calculate correction factors required for better accuracy of overall transformation. The correction was done by multiplying the observed coordinate values for each marker by the scaling factor (which was usually of the order of 0.995).

In addition, it was observed that in the previous version there were several inversion calculations of the

transformation matrices, which tend to give unstable results. These inversions were replaced with the more robust QR factorization method, yielding more precise results.

At last, out of all 23 caddy markers and 9 cross markers, only the markers best visible by all cameras were used consistently for the coordinate transformations. In particular, out of the 23 caddy markers and 9 cross markers, 6 non-collinear markers were used, respectively, formed two relatively large, symmetrical triangles.

Fifteen independent experimental runs were conducted for different target marker locations of the phantom base. For each of the five locations, three experimental runs were performed. The main objective of testing the performance of the transformation at different locations was to detect any dependence of the performance on target location. Table 5 shows the observed system alignment errors produced by the orthogonal coordinate transformations.

Table 5. Alignment Errors Produced by the Orthogonal Transformation

Data Run	Hole/Pin	Y error (mm)	Z error (mm)	Total error (mm)
1	12	-0.22	0.08	0.2292
2	12	-0.44	0.78	0.8952
3	12	-0.15	0.13	0.1975
4	22	0.49	0.36	0.6075
5	22	1.24	-0.97	1.572
6	22	0.33	-0.50	0.6008
7	42	-0.03	0.20	0.1991
8	42	-1.23	0.26	1.253
9	42	-0.83	0.04	0.8302
10	52	-0.12	0.34	0.3597
11	52	0.56	0.12	0.5764
12	52	-0.18	0.12	0.2123
13	32	0.04	0.41	0.4165
14	32	0.35	0.21	0.405
15	32	0.90	0.13	0.9098
			Average	0.617613333
			Standard Deviation	0.395241433

It can be observed that the mean system error was 0.6 mm and the standard deviation 0.39 mm. The new results produced are much better than those reported in the preceding thesis, i.e., a mean system error of 2.4 mm and standard deviation of 2.2 mm.

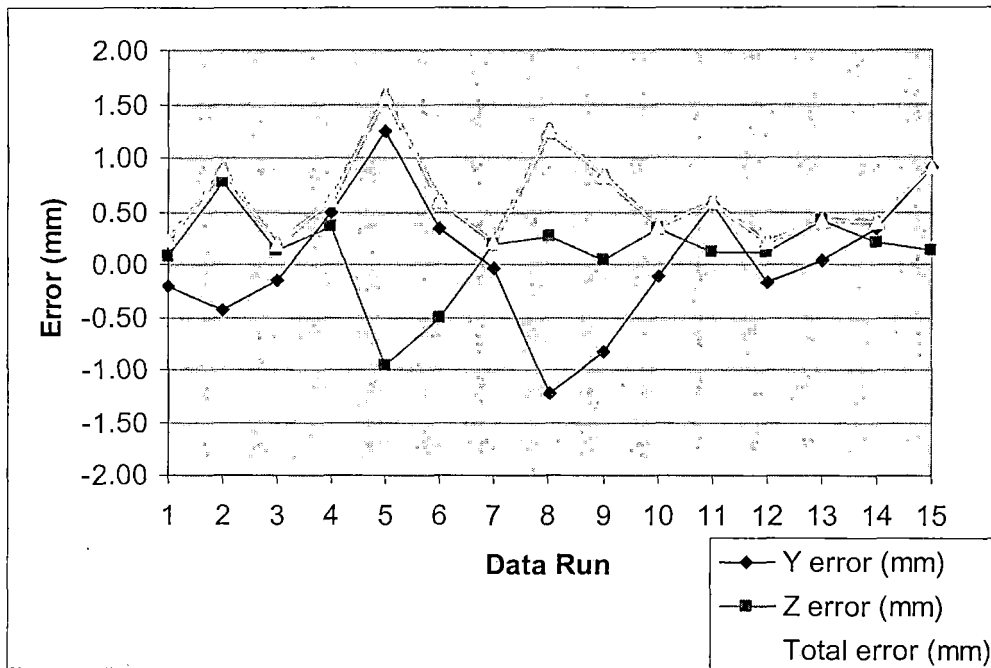


Figure 35. System Alignment Errors Produced by the Orthogonal Transformation

The results are presented graphically in Figure 35. With the exception of a few outliers, the errors are generally acceptable for the clinical purpose of the SAVPS.

5.4.3 Least Square-based Coordinate Transformation

This conceptually different method of coordinate transformation was implemented in a similar way as the orthogonal transformation using the Matlab package and porting a previous version of the transform algorithm written in Mathcad. The LS method uses a simplistic way to calculate the rotational and the translation vectors

required for the coordination transformation. Homogeneous transformation matrices (4x4) were calculated using the scaling-factor corrected marker coordinates observed from the camera and the DIL reference coordinates as the two data sets required by least square estimation. This results in two transformations X and Y:

X = Local coordinates (DIL coordinate) \ Global coordinates (Vicon)

Y = Global coordinates (Vicon) \ Local coordinates (DIL coordinate)

The major change from the previous Mathcad version of the LS-based transformation was updating the asymptotic linearized inversion with a robust QR factorization in the implementation. This methods impose restrictions such that (1) factors underlying the Y and X variables are extracted from the local and global matrices, respectively, and never from cross-product matrices involving both the Y and X variables, and (2) the number of prediction functions can never exceed the minimum of the number of Y variables and X variables.

The general disadvantage of the LS approach is that the accuracy of the solution is limited in view of the asymptotic nature of the approximate inverse operator used to compute it. In case of a larger problem, linearized

least square inversion usually involves the use of an iterative process. While this increases the cost of the method, it also allows more control of the accuracy of the approximate solution.

The same experimental runs which were used to estimate the error introduced by the orthogonal transformation were also used for the LS transformation. The results obtained during 15 different experimental runs are shown in Table 6.

Table 6. Alignment Errors Produced by the Least Square Based Transformation

Data Run	Hole/Pin	Total error (mm)
1	12	16.83
2	12	20.54
3	12	23.46
4	22	30.74
5	22	26.78
6	22	17.64
7	42	30.2
8	42	18.41
9	42	24.86
10	52	28.64
11	52	35.24
12	52	35.74
13	32	14.18
14	32	17.28
15	32	27.03
	Mean	25
	SD	8

The overall transformation error generated using least square transformation was larger than what was expected. In particular, this transformation method yielded a mean error of 25 mm and a standard deviation of 8 mm, which is unacceptable for clinical use, even though, it is an improvement compared to the even larger values reported in the preceding thesis. The fluctuation of the

errors, which ranged from 14.18 mm to 35.74 mm, is shown in Figure 36.

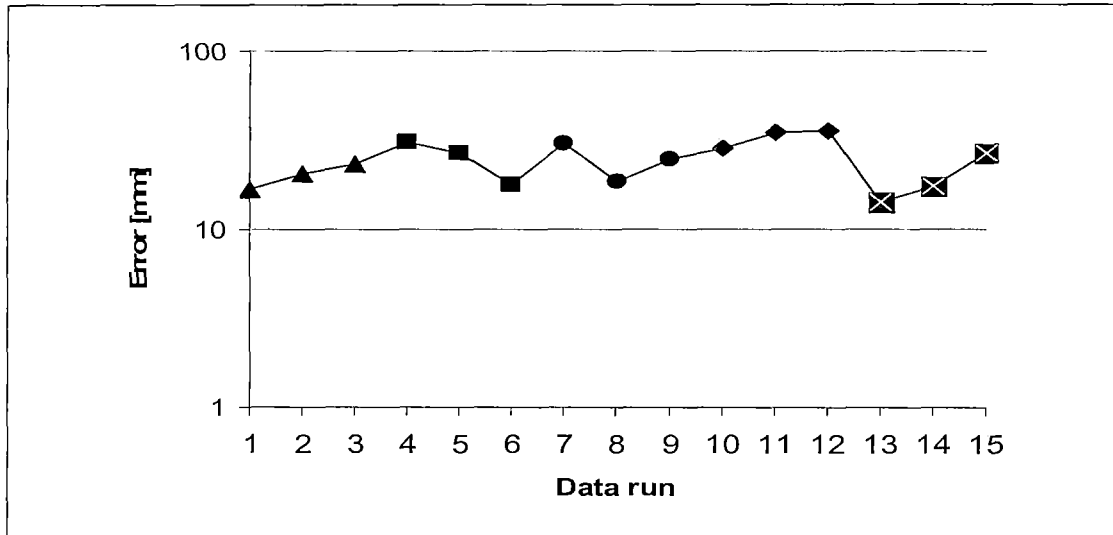


Figure 36. System Alignment Errors Produced by the Least Square Based Transformation

5.4.4 Constrained Least Square Based Transformation

This method was used for the first time for the coordinate transformation required for functional radiosurgery. The main objective of investigating this transformation method in this thesis was to explore the possibility of a more efficient transformation method which might yield less alignment error than the orthogonal transformation and least-square based transformation. Overall, the goal was to reduce the alignment error generated by the least-square based transformation by forcing several minimizations constrained. Table 7 illustrates the

alignment errors produced by constrained least square transformation (CLS) method during 15 different runs.

Table 7. Alignment Errors Produced by Constrained Least Square Method

Data Run	Hole/Pin	Total error (mm)
1	12	25.51
2	12	3.53
3	12	12.54
4	22	32.63
5	22	4.061
6	22	39.28
7	42	40.98
8	42	4.772
9	42	15.3
10	52	45.24
11	52	33.43
12	52	46.66
13	32	31.79
14	32	4.151
15	32	7.504
	Mean	23.61
	SD	17.83

It can be observed from the above table (table 7) that alignment errors generated by the CLS method are scattered, ranging from 3.53 mm to 45.24 mm. Even though, CLS shows some promising transformation by generating results around 4 mm, its overall performance seems to be

instable and fluctuating. The fluctuation in the alignment errors is shown in the figure 37.

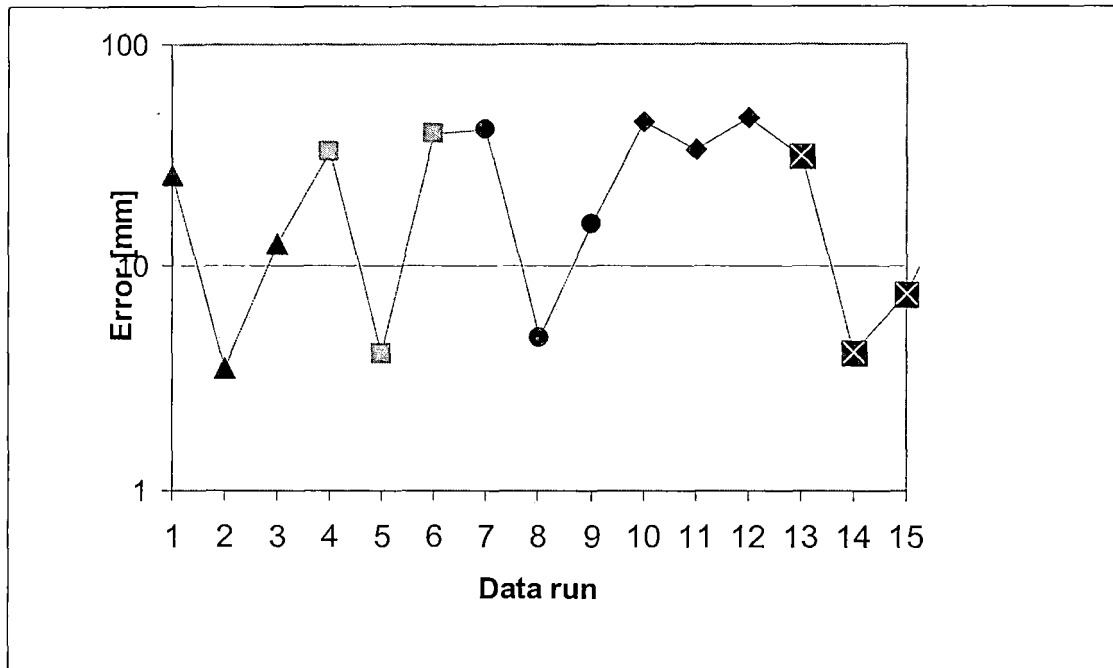


Figure 37. System Alignment Errors Produced by Constrained Least Square

Constrained Least Square method's encouraging minimal error performance is superceded by its maximum value. Because of its instable performance, it is inapplicable for the implementation in the clinical trail.

5.4.5 Summary

The above experimental runs lead to a conclusion that in spite of generating good transformation matrices, LS and Constrained LS based transformation produced higher alignment errors than the Orthogonal transformation.

CHAPTER SIX

SUMMARY, CONCLUSION AND RECOMMENDATIONS

6.1 Summary

The first major milestone in the optimization process of the SAVPS reached in this thesis work was the implementation of a new dynamic camera calibration pattern, i.e., the Vertical-Horizontal Pattern. The average mean residuals produced by this method are less than 0.3 mm which is far better than the 0.9 mm observed with previous methods. This method leads to a stable performance of the Optical Positioning System, which lays the foundation for an efficient overall operation of the system.

The second milestone reached in the optimization process was the implementation of a new image-processing algorithm that added four-digit accuracy to the calculation of alignment offset between the laser beam and the target. This accuracy was achieved without sacrificing the robustness of the image-processing algorithm. The new algorithm is able to process even images of poor quality more precisely.

These achievements in the optimization process have led to a decrease of the mean distance error between

markers inherent in the camera visualization from 0.15 mm previously to 0.00 mm now for the cone markers and from -0.35 mm previously to -0.23 mm now for the caddy markers. It is suspected that the reason for the slightly higher systematic distance error in the visualization of the caddy markers is due to the asymmetrical geometrical distribution of these markers.

Further, as a result of these improvements, the overall alignment errors produced by camera errors and coordinate transformation errors reduced significantly. In 15 independent experimental runs, the orthogonal transformation method produced a mean alignment error of 0.61 mm with a standard deviation of 0.39 mm, while the least square (LS) based transformation produced a mean error of 27 mm with a standard deviation of 8 mm. The constrained least square (CLS) transformation, using orthogonality of the transformation matrix as a constraint, showed some promise in reducing the minimum error from 14 mm for the ordinary LS based transformation to 3 mm for the CLS transformation. On the other hand, the mean error produced by the CLS transformation was 23 mm, and thus only slightly smaller than that of the LS based transformation (27 mm) due to a larger standard deviation (18 mm) and maximum error (47 mm).

6.2 Conclusion

At this point, the results demonstrate that the orthogonal transformation outperforms both ordinary unconstrained LS-based transformation and orthogonality-constrained least square transformation. The consistently better performance favors this method for the planned clinical application since it is able to reach submillimeter accuracy.

The analysis presented in this thesis and the results obtained from many experimental runs leads to the several conclusions regarding future steps toward optimization of the SAVPS system.

Due to the complexity of the system, it is susceptible to both systematic and random errors. Further optimization to achieve alignment errors consistently less than 0.5 mm will be an arduous task.

The ultimate limitation to the alignment process is imposed by the increments of the patient positioning system (PPS), which currently are of the order of ± 0.1 mm in vertical direction and about half of that in horizontal directions. This limitation might not be significant for the application at larger prospect, but as the errors produced by the alignment verification reach into the

submillimeter range, it may be necessary to include this PPS uncertainty in the net alignment error calculation.

6.3 Recommendations

Although the SAVPS has limitations that one may not be able to overcome at a reasonable cost, there are several components and areas in the system where improvements can still be made. A core area of improvement will be the redesign of the marker caddy system. It has been observed that out of the available 23 markers of the caddy, only 6 markers were useful for the application. It appears that the geometric arrangement of these markers is currently not optimal. Hence, redesigning the marker caddy system might lead to a better result with respect to camera error and transformation error. The other possible area of improvement is the coordinate transformation method. The constrained least square method showed some improvement of the minimum observed error, but only one constraint was tested within this thesis work. This transformation method can be further researched and refined by adding additional constraints to LS minimization.

Adding these and other possible improvements to the SAVPS, it seems quite possible to achieve an alignment

error below a value of 0.3 mm, approaching the resolution limitation of the cameras and that of the patient position system movements. In this case, the SAVPS system can be used for the planned clinical application in functional proton radiosurgery.

APPENDIX A
IMAGE PROCESSING ROUTINES

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This is the Function which calculations the offset
between the reference marker and the laser beam.
% This takes an JPEG image taken of image of reference
marker and the laser beam in a card board.
%   Input: Path to the image location in the computer
%           %
%   output:
%           %
%           distance = net offset between two
center           %
%           TF_offset = horizontal offset
%           %
%           ZF_offset = vertical offset
%           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

[TF_offset, ZF_offset, distance]= function image_process()
path = 'C:\Documents and
Settings\raj\Desktop\IMGA0677.JPG';
RGB = imread(path);
text(12,12,path)
%RGB = imread('C:\IMGA0394.JPG');

text(15,15,'Estimate radius of circle',...
      'FontWeight','bold','Color','y')
I= rgb2gray(RGB);
threshold = graythresh(I);
BW = im2bw(I,threshold);
dim = size(BW);
col = dim(2);
while max(BW(:,col)) < 1, %find right boundry
    col = col -1;
end

rowu = dim(1);
while max(BW(rowu,:)) < 1, %find upper boundry
    rowu = rowu -1;
end
rowl = 1;
while max(BW(rowl,:)) < 1, %find lower boundry
    rowl = rowl + 1;
end
%row = floor(( rowu + 2*rowl) / 3);
row = rowl + 10;
while BW(row,col) < 1, %find right side of shape

```

```

        col = col -1;
        %col = 1-col;
end
num_circ = 10;
xca = zeros(num_circ,1);
yca = zeros(num_circ,1);
radiusa = zeros(num_circ,1);
residual = zeros(num_circ,1);
num_points = 400;%number of points to be fitted
phi = 2*pi;
tol = pi;%maximum sector missing from outer
circle%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
imshow(BW);
hold on;
radius = 0;
tol2 = 50;
while (phi > tol) | (radius < tol2),
    connectivity = 8;
    contour = bwtraceboundary(BW, [row ,col ], 'N',
connectivity, num_points, 'counterclockwise');
    if (~isempty(contour))
        x = contour(:,2);
        y = contour(:,1);
        % solve for parameters a, b, and c in the
least-squares sense by
        % using the backslash operator
        abc=[x y ones(length(x),1)]\[-(x.^2+y.^2)];
        a = abc(1); b = abc(2); c = abc(3);

        % calculate the location of the center and the radius
        xcen = -a/2;
        ycen = -b/2;
        center = [xcen; ycen];
        startpt = [contour(1,2); contour(1,1)];
        endpt =
[contour(max(size(contour)),2);contour(max(size(contour)),
1)];
        phi = ((startpt-center)' *(endpt - center))/
(norm(startpt - center) * norm ( endpt - center));
        phi=acos(phi);
        if startpt(1) > endpt(1)
            phi = 2*pi - phi;
        end

        plot([center(1),startpt(1)], [center(2),startpt(2)], 'r-
,')
        plot([center(1),endpt(1)], [center(2),endpt(2)], 'r-'

```

```

        radius = sqrt((xcen^2+ycen^2)-c);
        %residual(index) = norm([x y ones(length(x),1)]* abc -
[-(x.^2+y.^2)]);
        if radius < tol2
            col = col-1;
        end
        if phi > tol
            num_points = num_points + 5;
        end
    else
        col = col -1;
        row = row +1;
    end
end
end
xa=contour(:,2);
ya=contour(:,1);
xc=xcen;
yc=ycen;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

```

```

%find inner circle

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

```

```

rowi = round(yc);%start at center of previous circle
coli = round(xc);

```

```

while BW(rowi,coli) < 1 , %its in the black region
    rowi = rowi - 1; % pick new points straight above
    %rowi = rowi - 2;
    %coli = coli+1;
    BW(rowi,coli)
    %coli = coli + 1;
end

```

```

num_circ = 10;
xca = zeros(num_circ,1); %initialize
yca = zeros(num_circ,1);
radiusa = zeros(num_circ,1);
residual = zeros(num_circ,1);
num_points = 100;%number of points to be fitted
phi = 2*pi;
tol = pi;%maximum sector missing from inner circle
tol2= 10;% minimum radius in pixel

```

```

radiusi = 0;
lastsize =10;
while (phi > tol | radiusi < tol2)
    contour = bwtraceboundary(BW, [rowi ,coli ], 'N',
connectivity, num_points, 'clockwise');
    plot(coli,rowi,'y+');
    if max(size(contour)) <= lastsize
        rowi = rowi +1 ;
        while BW(rowi,coli) < 1,
            coli = coli +1;
            plot(coli,rowi,'y+');
        end
        lastsize= max(size(contour));
        contour=[];
    end

    if (~isempty(contour)) %check for contour
        lastsize= max(size(contour));
        x = contour(:,2);
        y = contour(:,1);
        % Mathematical Derivation of Least Square formula for
Circle
        % (x-xcen)^2 + (y-ycen)^2 = r^2
        % -(x^2+y^2) = (-2xcen)x + (-2ycen)y + (xcen^2 +
ycen^2 - r^2)
        % -(x^2+y^2) = ax + by + c
        % solve for parameters a, b, and c in the least-
squares sense by
        % using the backslash operator
        abc=[x y ones(length(x),1)]\[-(x.^2+y.^2)];

        a = abc(1); b = abc(2); c = abc(3);

        xcen = -a/2;
        ycen = -b/2;
        center = [xcen; ycen];
        radiusi = sqrt((xcen^2+ycen^2)-c);
        % determine size of missing sector
        startpt = [contour(1,2); contour(1,1)];
        endpt =
[contour(max(size(contour)),2);contour(max(size(contour)),
1)];
        phi = ((startpt-center)' *(endpt - center))/
(norm(startpt - center) * norm ( endpt - center));
        phi=acos(phi);
        if startpt(1) > endpt(1)
            phi = 2*pi - phi;

```

```

end

    if radiusi < tol2 %check to see if the result is good,
if not move and try again
        rowi = rowi - 1;
    end
    if phi > tol
        num_points = num_points + 10;%changed to 2
    end
    else
        rowi = rowi -1;
    end
end
xci=xcen;
yci=ycen;
xai=contour(:,2);
yai=contour(:,1);

    %imshow(BW);
    imshow(RGB);

    hold on;

plot(xa,ya,'y-','LineWidth',1);
plot(xai,yai,'m-','LineWidth',1);

% display the calculated center

plot(xc,yc,'yx','LineWidth',2);
plot(xci,yci,'m+','LineWidth',2);

% plot the entire circle
theta = 0:0.01:2*pi;

% use parametric representation of the circle to obtain
coordinates
% of points on the circle
Xfit = radius*cos(theta) + xc;
Yfit = radius*sin(theta) + yc;
Xfiti = radiusi*cos(theta) + xci;
Yfiti = radiusi*sin(theta) + yci;

plot(Xfit, Yfit);
plot(Xfiti, Yfiti);
distance = norm([xc,yc] - [xci,yci]) * 10/(2*radius)

```

```
TF_offset = (xc - xci) * 10/(2*radius)
ZF_offset = (yc - yci) * 10/(2*radius)
if TF_offset > 0
    disp('table is near to the camera and TF_offset value
need to be subtracted from the current position');
else disp('Table is far from the camera and TF_offset
value need to be added to the current position');
end
    if ZF_offset > 0
        disp('table is higher than the beam and ZF_offset
value need to be subtracted from the current position');
    else disp('Table is lower than the beam and ZF_offset
value need to be added to the current position');
    end
end
```


APPENDIX B
ORTHOGONAL TRANSFORMATION

1.1 Method used for Orthogonal Transformation:

In the following discussion, the superscript (g) indicates global coordinates and the superscript (l) indicates local coordinates. In general, the coordinates $p_{1,i}^{(l)}$, $p_{2,i}^{(l)}$, $p_{3,i}^{(l)}$ ($i = 1-3$) of three distinct markers in the local system will also be known in the global system, where they are called $p_{1,i}^{(g)}$, $p_{2,i}^{(g)}$, $p_{3,i}^{(g)}$. All coordinate systems considered here are right-handed.

Consider the triangle $P_1^{(l)}$, $P_2^{(l)}$, $P_3^{(l)}$ in the local coordinate system, which is formed by the three known markers (Figure 1). Let $p_1^{(l)}$, $p_2^{(l)}$, and $p_3^{(l)}$, denote the position vectors pointing from the origin of the local reference system to the central point of each marker. Note that lower-case bold letters are used here to denote vectors, and upper-case bold letters to denote matrices. The corresponding position vectors to the triangle $P_1^{(g)}$, $P_2^{(g)}$, $P_3^{(g)}$ in the global reference system are called $p_1^{(g)}$, $p_2^{(g)}$, and $p_3^{(g)}$. One may obtain the clearest perception of the rotations and translation involved in the coordinate transformation between the two reference system by assuming that the origins and axes of both coordinate systems coincide, and that the vectors $p_1^{(l)}$, $p_2^{(l)}$, $p_3^{(l)}$ and $p_1^{(g)}$, $p_2^{(g)}$, $p_3^{(g)}$ represent two different marker sets. Then, the task to find a coordinate transformation between the

two coordinate systems is identical to finding the transformation that maps the local marker set onto the global marker set.

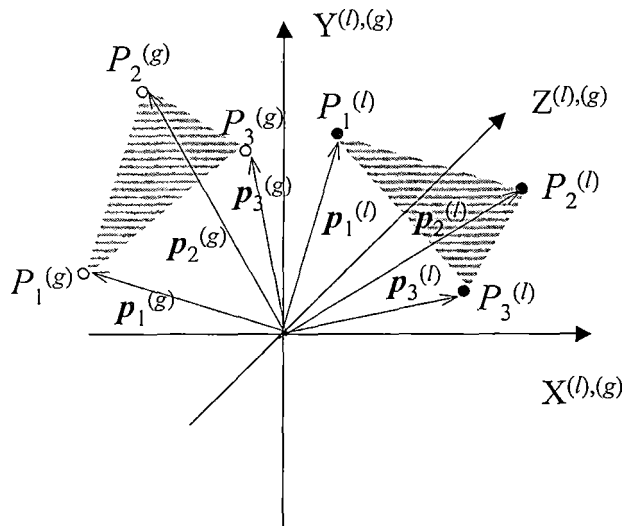


Figure 1. Conceptual view of the two markers sets in the local and global reference systems.

In general, the transformation equation, which maps corresponding l points onto g points, can be expressed as follows:

$$p_n^{(g)} = M_B \cdot M_A \cdot p_n^{(l)} + t \quad (n = 1 - 3)$$

where M_A and M_B are 3×3 matrices representing proper rotations. The matrix M_A corresponds to a rotation that makes the plane formed by the l marker set parallel to the plane formed by the g marker set. The matrix M_B

corresponds to an "in-plane" rotation, which aligns corresponding triangle sides with respect to each other. After performing these two rotations on the l triangle, the vector t corrects for the residual translational difference between l points and corresponding g points.

1.1.1 Rotation of a Vector about a Non-collinear Vector

We now derive a useful equation for the matrix describing the rotation of a vector about another non-collinear vector. Consider a unit vector v , which we want to rotate around a unit vector o by an angle ϕ to form the vector v' . Note that the angle θ between v and o is given by $\cos(\theta) = v \cdot o$. We perform this rotation in a Cartesian coordinate system formed by the three orthogonal vectors o , $p = (v \times o)/\sin(\theta)$, and $q = [o \times (v \times o)]/\sin(\theta)$, where the factor $1/\sin(\theta)$ is required to assure unit length. The rotated vector v' can then be expressed in terms of these

three unit vectors as follows:

$$v' = (v \cdot o) o + \sin(\theta) \sin(\phi) p + \sin(\theta) \cos(\phi) q$$

By substituting the expressions for p and q in terms of o and v , and by taking into account that

$$o \times (v \times o) = v - o (v \cdot o),$$

we find that

$$v' = v \cos(\phi) + o(v \cdot o) [1 - \cos(\phi)] + (v \times o)$$

$\sin(\phi)$

This equation can also be expressed in matrix form as $v' = M v$, where the rotation matrix M is explicitly given by

$$M = \begin{pmatrix} \cos(\phi) + o_1^2(1 - \cos(\phi)) & o_3 \sin(\phi) + o_1 o_2(1 - \cos(\phi)) & -o_2 \sin(\phi) + o_1 o_3(1 - \cos(\phi)) \\ -o_3 \sin(\phi) + o_1 o_2(1 - \cos(\phi)) & \cos(\phi) + o_2^2(1 - \cos(\phi)) & -o_1 \sin(\phi) + o_2 o_3(1 - \cos(\phi)) \\ o_2 \sin(\phi) + o_1 o_3(1 - \cos(\phi)) & -o_1 \sin(\phi) + o_3 o_2(1 - \cos(\phi)) & \cos(\phi) + o_3^2(1 - \cos(\phi)) \end{pmatrix}$$

1.1.2 Derivation of the Matrix M_A

To find the mathematical expression for the matrix M_A , which transforms the l triangle into one that is coplanar with the g triangle, we first determine the unit normal vector of the l triangle, $n^{(l)}$, and the unit normal vector of the g triangle, $n^{(g)}$. The two unit vectors can be calculated by forming and normalizing the vector products $(p_3^{(l)} - p_1^{(l)}) \times (p_2^{(l)} - p_1^{(l)})$ and $(p_3^{(g)} - p_1^{(g)}) \times (p_2^{(g)} - p_1^{(g)})$, respectively (Fig. 2a).

The matrix M_A corresponds to a rotation of the vector unit $n^{(l)}$ about the orthogonal vector $n_A = (n^{(l)} \times n^{(g)})$ by the angle α , where $\cos(\alpha) = n^{(l)} \cdot n^{(g)}$ (Fig. 2b).

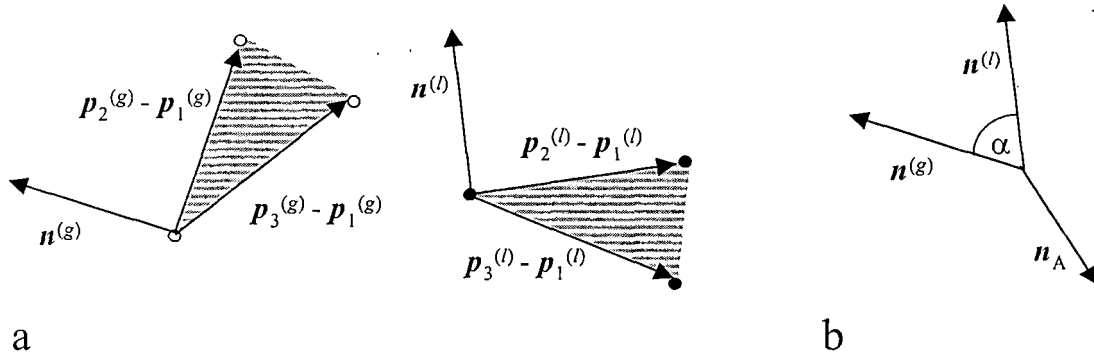


Figure 2 (a) Definition of the normal vectors $n^{(1)}$ and $n^{(g)}$, and (b) rotation performed by matrix M_A .

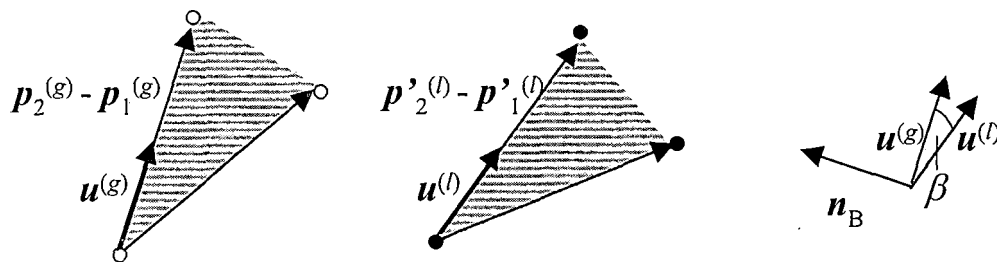
By normalizing the vector n_A to $o_A = n_A / \sin(\alpha)$, and by using the expression for the rotation matrix M derived above, we obtain the following expression for the matrix M_A :

$$M_A = \begin{pmatrix} \cos(\alpha) + o_{A1}^2(1 - \cos(\alpha)) & n_{A3} + o_{A1}o_{A2}(1 - \cos(\alpha)) & -n_{A2} + o_{A1}o_{A3}(1 - \cos(\alpha)) \\ -n_{A3} + o_{A2}o_{A1}(1 - \cos(\alpha)) & \cos(\alpha) + o_{A2}^2(1 - \cos(\alpha)) & n_{A1} + o_{A2}o_{A3}(1 - \cos(\alpha)) \\ n_{A2} + o_{A3}o_{A1}(1 - \cos(\alpha)) & -n_{A1} + o_{A3}o_{A2}(1 - \cos(\alpha)) & \cos(\alpha) + o_{A3}^2(1 - \cos(\alpha)) \end{pmatrix}$$

Note that in this expression the terms $o_{Ai} \sin(\alpha)$ have been replaced by n_{Ai} ($i = 1-3$).

1.1.3 Derivation of the Matrix M_B and the Vector t

Multiplication of the local position vectors $p_1^{(1)}$, $p_2^{(1)}$, and $p_3^{(1)}$ by matrix the M_A yields new vectors $p'_1^{(1)}$, $p'_2^{(1)}$, and $p'_3^{(1)}$ which form a triangle that is now coplanar with that formed by the global position vectors $p_1^{(g)}$, $p_2^{(g)}$, and $p_3^{(g)}$. To obtain the rotation matrix M_B , we normalize the triangle vectors $(p'_2^{(1)} - p'_1^{(1)})$, and $(p_2^{(g)} - p_1^{(g)})$, which yields the non-collinear unit vectors $u^{(1)}$ and $u^{(g)}$, respectively (Fig. 3a). The matrix that aligns unit vector $u^{(1)}$ with unit vector $u^{(g)}$ represents a rotation of the vector $u^{(1)}$ about the orthogonal vector $n_B = (u^{(1)} \times u^{(g)})$ by the angle β where $\cos(\beta) = (u^{(1)} \cdot u^{(g)})$ (Fig. 3b). By normalizing the vector n_B to $o_B = n_B / \sin(\beta)$ the matrix M_B can be expressed as



a

b

Fig. 3 (a) Definition of the normal vectors $u^{(l)}$ and $u^{(g)}$, and (b) rotation performed by matrix M_B .

$$M_B = \begin{pmatrix} \cos(\beta) + o_{B1}^2(1 - \cos(\beta)) & n_{B3} + o_{B1}o_{B2}(1 - \cos(\beta)) & -n_{B2} + o_{B1}o_{B3}(1 - \cos(\beta)) \\ -n_{B3} + o_{B2}o_{B1}(1 - \cos(\beta)) & \cos(\beta) + o_{B2}^2(1 - \cos(\beta)) & n_{B1} + o_{B2}o_{B3}(1 - \cos(\beta)) \\ n_{B2} + o_{B3}o_{B1}(1 - \cos(\beta)) & -n_{B1} + o_{B3}o_{B2}(1 - \cos(\beta)) & \cos(\alpha) + o_{B3}^2(1 - \cos(\beta)) \end{pmatrix}$$

Multiplication of the local position vectors $p'_{1(1)}$, $p'_{2(1)}$, and $p'_{3(1)}$ by matrix M_B yields new vectors $p''_{1(1)}$, $p''_{2(1)}$, and $p''_{3(1)}$, which makes the l triangle identical in orientation with respect to the g triangle. Finally we translate $p''_{1(1)}$ into $p_1^{(g)}$ by adding the vector $t = p_1^{(g)} - p''_{1(1)}$. If no systematic or random error is involved the triangles should now exactly superimpose.

The two rotations involved in the transformation can be combined into one rotation by calculating the matrix $M_{AB} = M_B \cdot M_A$. We then have

$$v^{(g)} = M_{AB} \cdot v^{(1)} + t$$

for transformations of any vector v from the local to the global coordinate system. Since the rotation matrix can be inverted, we can also transform in the opposite direction:

$$v^{(l)} = M_{AB}^{-1} (v^{(g)} - t)$$

This inverse transformation can be used to transform any vector from the global coordinate system into a local coordinate system.

1.2 Routines for Orthogonal Transformation

```
*****
COORDINATE TRANSFORMATION FOR FUNCTIONAL RADIOSURGERY
*****
% This program Computes the distance between the CONE AXIS
% and a selected Phantom Base Target marker based on an
% Orthogonal Transformation from CONE REFERENCE (Local) to
% the Stereotactic Reference System(Global) system.

% Input:
      Observed Caddy and Cone markers coordinate,
%Image offset values, Calibration offset values,
% referenced Caddy and Cone markers values, Phantom base
% markers location.
% OUTPUT:
%       X, Y and Z axis alignment errors
%
% Reading reference and measured Coordinate Data From
files%

%Reference DATA for the Caddy Markers

%[m n]= size(Acaddy);
%Reference DATA for the Caddy Markers

global Ecaddy Econe AcaddyDist AconeDist McaddyDist
MconeDist Etotal_Caddy
global MidCaddy Mno_Caddy Acaddy Bcaddy Mab_cone Mab_caddy
Tsum_cone Tsum_cone

Acaddy = CaddyReference();%Fetching the Stereotactic
Coordinate values for Caddy Markers
Acone = ConeReference();% Fetching the Stereotactic
Coordinate Values for Cone Markers

Bcaddy = CaddyActual();%Fetching the Observed Coordinate
Values for Caddy Markers
Bcone = ConeActual();%Fetching the Observed Coordinate
Values for Cone Markers

%Asking for the total number of marker visible to the
CAMERA
```

```

Mno_Caddy = input('How many markers are visible for CADDY
\n');
Mno_Cone = input('How many markers are visible for
CONE\n');
%k=zeros(Mno_Caddy,1);
%j=zeros(Mno_Cone,1);
MidCaddy=[];
MidCone=[];
fid=fopen('Orthogonal_output.xls','a+');
display('Visible markers for Caddy are:');
MidCaddy = Bcaddy(:,1);
disp(MidCaddy);
%Asking for the markers Visible to the CAMERA in the
CADDY
%for(i= 1:1:Mno_Caddy)
%   sprintf('%d Marker',i);
% MidCaddy(i,1)=str2num(input('', 's'));
% end

%Asking for the Markers Visible to the CAMERA in the CONE

display('Visible markers for Cone are:');
MidCone= Bcone(:,1);
disp(MidCone);
%for(i= 1:1:Mno_Cone)
%   sprintf('%d Marker',i) ;
%MidCone(i,1)=str2num(input('', 's'));
%end

%Initializing the Vectors

AcaddyDist=zeros(Mno_Caddy, Mno_Caddy);
AconeDist=zeros(Mno_Cone, Mno_Cone);
McaddyDist=zeros(Mno_Caddy,Mno_Caddy);
MconeDist=zeros(Mno_Cone,Mno_Cone);

%Calling the Corresponding function to load the file into
respective vector

AcaddyDist
=distance(Acaddy(:,2:end));%Actual(Reference)Distance
Between Caddy markers
AconeDist = distance(Acone(:,2:end));%Actual(Reference)
Distance Between Cone markers

```

```

McaddyDist =distance(Bcaddy(:,2:end));%Measured Distance
Between Caddy Markers
MconeDist = distance(Bcone(:,2:end));%Measured Distance
Between Cone Markers

% |-----DATA QUALITY CHECK SECTION-----
----|
%Calling the Scaling_Factor funtion which calculates the
Scaling Factor for
%Caddy and Cone distance Error.
[SF_Caddy,Ecaddy,SF_Cone,Econe]=
Scaling_Factor(AcaddyDist,McaddyDist,AconeDist,MconeDist);
%SF_Caddy = 0.995;
%SF_Cone = 0.998;
McaddyDist = McaddyDist*SF_Caddy;
MconeDist = MconeDist*SF_Cone;

Efinal_Caddy = McaddyDist-AcaddyDist;
Efinal_Cone = MconeDist - AconeDist;

%[EStdev_Caddy,Emean_Caddy] = stdev(Efinal_Caddy);
EStdev_Caddy = std(Efinal_Caddy(:))/sqrt(2);

EMin_Caddy = min(Efinal_Caddy(:));
EMax_Caddy = max(Efinal_Caddy(:));
Emean_Caddy = mean(Efinal_Caddy(:));
% [EStdev_Cone,Emean_Caddy] = stdev(Efinal_Cone);
EStdev_Cone = std(Efinal_Cone(:))/ sqrt(2);
EMin_Cone = min(Efinal_Cone(:));
EMax_Cone = max(Efinal_Cone(:));
Emean_Cone = mean(Efinal_Cone(:));

%Outputing the Calculated Value for Scaling
Factor,Standard Deviation and
% Distance Errors for Caddy and Cone.....
table1 = [ SF_Caddy' EStdev_Caddy' Emean_Caddy'
EMin_Caddy' EMax_Caddy'];
templ = {'For CADDY',' ',' ', datestr(now) , ' ';
'Scaling Factor', 'Standard Deviation', 'Mean
Error', 'Minimum Error', 'Maximum Error';
SF_Caddy, EStdev_Caddy, Emean_Caddy, EMin_Caddy,
EMax_Caddy};

%xlswrite('Quality_Check.xls','For CADDY');
xlswrite('Quality_check.xls',templ, 'A2:E4');
templ =[];

```

```

table2 = [SF_Cone' EStdev_Caddy' Emean_Cone' EMin_Caddy'
EMax_Caddy'];
temp2 = {'For CONE',' ',' ', datestr(now), ' ';
'Scaling Factor', 'Standard Deviation', 'Mean
Error', 'Minimum Error', 'Maximum Error';
SF_Cone, EStdev_Cone, Emean_Cone, EMin_Cone,
EMax_Cone};
%xlswrite('Quality_Check.xls','For Cone', 'A10');
xlswrite('Quality_check.xls', temp2,'A7:E9');
temp2 = [];

%Calling the Function which Calculates the Unitery
Transformation for
%CADDY
Bcaddy = [Bcaddy(:,1) Bcaddy(:,2:end).*SF_Caddy];
%Bcone = [Bcone(:,1) (Bcone(:,2:end).*SF_Cone)];
disp(Bcaddy);
disp(Bcone);
[Error_caddy_triangle1,Error_caddy_triangle2,Mab_caddy,
Tsum_caddy] = UT_Caddy(MidCaddy,Mno_Caddy,Acaddy,Bcaddy);

[Error_cone_triangle1,Error_cone_triangle2,Mab_cone,Tsum_c
one] = UT_Cone(SF_Cone,MidCone,Mno_Cone);

%Transformation From the CONE to the CADDY System
% The transformation from the cone to the global system
is described by
% the matrix Mab_cone and the vector Tsum_cone. The
transformation from
% the global to the caddy system is the inverse of the
transformation
% calculated above and therefore described by the matrix
Mab_caddy' and
% vector Mab_caddy'Tsum_cone. The combination of these
transformation
% yeilds:

%Mcc = inv(Mab_caddy)*Mab_cone;
Mcc = Mab_caddy\Mab_cone;
%Mcc = [ 0.01919 0.01211 0.9974 ;
% 0.0392 0.99915 -0.01285;
% -0.99905 0.03944 0.01867];

%Total_error =inv(Mab_caddy)*(Tsum_cone - Tsum_caddy);

```

```

Total_error = Mab_caddy \ (Tsum_cone - Tsum_caddy);
% Total_error = [613.5968; 39.58582;39.81147];

%Distance Between CONE Axis and Target Marker
%the distance between a selected target marker of the
Phantom base and the
%Cone Axis. The general strategy is to describe the
equation of the cone
%axis in the stereotactic coordinate system and then to
calculate the
%shortest vector between target marker and cone
axis,which is
%perpendicular to the axis.
M_pbase = [];
P_target = [];
disp('Calculating Distance between Cone axis and Target
Marker');
Aphbase=Phantom_Base();
M_pbase =input('Please Enter the hole/pin combination of
the selected phantom base marker: ');

P= Aphbase(ismember(Aphbase(:,1),M_pbase),:);
P_target = P(:,2:end)';

%The Cone Axis intersects the origin of the cone
reference system and the
%unit vector parallel to the axis points in Z-directions
r0 = [0;0;0];
U_cone = [0;0;1];

%the cone axis is given by

%r =r0 + (Lambda)*U_cone;

%The Transformation of
%this to the stereotactic system yeilds

% rs = r0s +(Lambda)*U_cones;

r0s = Mcc* r0 + Total_error;
U_cones = Mcc * U_cone;

disp(r0s);
disp(U_cones);

%The Value of Lambda that corresponds to the endpoint of
the shortest

```

```

%vector between target and axis is given by:

Lambda_T = dot((P_target - r0s),U_cones);

%the position vector of the endpoint of the shortest
vector

r_T = r0s + (Lambda_T)*U_cones;

%Shortest vector between marker and axis,E_ta

E_ta = P_target - r_T;

%Correct for any actual offset + vertical offset = marker
above axis +
%horizontal offset = marker inferior(toward halo) from
axis)and offset(in mm) (x,y,z)

M_off = (Offset())';
x_off = M_off(1);
y_off =M_off(2);
z_off =M_off(3);
Image_off = Image_offset();
v_off = Image_off(1);%The Vertical Offset Calculated in
the Image processing program
h_off = Image_off(2);%The horizontal Offset Calculated in
the Image processing program
E_ta_1 = E_ta(1)-x_off;
E_ta_2 = E_ta(2)- v_off - y_off;
E_ta_3 = E_ta(3)- h_off - z_off;
E_TA = [ E_ta_1 ; E_ta_2; E_ta_3];
Length_mm=norm(E_TA);
%OUTPUT SECTION
fid=fopen('Orthogonal_output.xls','a+')
fprintf(fid,'The Scaling Factor for the Caddy is =
%0.5g\n ',SF_Caddy);
fprintf(fid,'\n');
fprintf(fid,'The Scaling Factor for the Cone is =
%0.5g\n ',SF_Cone);
fprintf(fid,'\n');
fprintf(fid,'The Error in X direction is
%0.5g\n',E_ta_1);
fprintf(fid,'\n');
fprintf(fid,'The Error in Y direction is
%0.5g\n',E_ta_2);
fprintf(fid,'\n');

```

```

fprintf(fid,'The Error in Z direction is
%0.5g\n',E_ta_3);
fprintf(fid,'The normalized error in mm is
%0.5g\n',Length_mm);
fprintf(fid,'\n');
fclose(fid);
clear fid;

```

1.3 Scaling Factor Calculation routine

```

%This Function calculates the Scaling Factor for both the
Caddy and Cone distances.

```

```

% input Variables: AcaddyDist -> Actual (Reference)
%                 Distance between Caddy Markers
%                 McaddyDist -> Measured (Observed)
%                 Distance between Cone Markers
%                 AconeDist -> Actual (Reference)
%                 Distance between Cone Markers
%                 MconeDist -> Measured (Observed)
%                 Distance between Cone Markers
%
% Returning Variables: SF_Caddy -> Scaling factor for
%                       Caddy Markers Distances
%                       SF_Cone -> Scaling factor for Cone
%                       Markers Distances

```

```

function[SF_Caddy, Ecaddy, SF_Cone, Econe]=Scaling_Factor(AcaddyDist, McaddyDist, AconeDist, MconeDist)

```

```

global AcaddyDist AconeDist McaddyDist MconeDist
Ecaddy = McaddyDist - AcaddyDist;
Econe = MconeDist - AconeDist;
[m n] = size(Ecaddy);
[x y]=size(Econe);
%Berror = [1: m*n];
u = (m*(m-1))/2; %temporary variable to hold total area of
lower triangular
l =u;
Berrl = zeros (u, 1);%we are looking for the lower
triangles for CADDY
v = (x*(x-1))/2;
w=v;

```



```

Berr2 = zeros(v,1);%we are looking for the lower triangles
for CONE
Bte1=zeros(u,1);
Bte2=zeros(v,1);

%Coloumnizing the Error and the Distance in single column

for( j= m-1:-1:1)
    Berr1(1:u,1)= Ecaddy(j+1:m,j);%Columnized Temporary
Variable for Caddy

    Bte1(1:u,1)=AcaddyDist(j+1:m,j);%Columnized Temporary
Variable for Caddy
    u=1-1;
    l=u+j-m;

end
for( k= x-1:-1:1)
    Berr2(w:v,1)= Econe(k+1:x,k);% Columnizing Temporary
Variable for CONE
    Bte2(w: v,1)=AconeDist(k+1:x,k);%Columnizing Temporary
Variable for CONE
    v = w-1;
    w = v + k-x;
end

%Appending '1' With the Coloumnized Vector

Aone1=[ones(size(Bte1)),Bte1];%Appending the ones with the
Actual Caddy Distance
Aone2 =[ones(size(Bte2)),Bte2];%Appending the ones with
the Actual Caddy Distance

z1 = Aone1\Berr1; %Using the least Square Fitting
technique to find the slope for Caddy
z2 = Aone2\ Berr2; %Using the least Square Fitting
technique to find the slope for Cone
S1= z1 (2); %Slope of the line for Caddy
S2 = z2(2); %Slope of the line for Cone
% now calculates the slope and intercept.

SF_Caddy= 1/(1+S1) % Scaling factor for the Distance
error of CADDY
SF_Cone=1/(1+S2) %Scaling factor for the Distance error
of CONE
%returns SF_Caddy, Ecaddy, SF_Cone, and Econe;

```

```

%Routine to calculate Orthogonal transformation for each
% Each Cone triangles.

%UT transformation function calculates the transformation
% equation for the Cone triangles.
% Input - Scaling factor, reference and observed
coordinates values for
% selected markers and marker number.
% Output - Error present in both triangle transformation
and transformation matrices.

Function
[Error_cone_triangle1,Error_cone_triangle2,Mab_cone,Tsum_c
one]= UT_Cone(SF_Cone,MidCone,Mno_Cone)
Global MidCone Mno_Cone Acone Mab_cone Tsum_cone fid

SF_Cone = 0.99836;
MidCone = [ 2;3;4;5;6;7;8;9];
Mno_Cone = 8;
Acone = ConeReference();
Bcone1 = ConeActual();
%Bcone_old = Bcone;
%Bcone_temp = Bcone1(:,1:end).*SF_Cone;
Bcone_temp = Bcone1(:,2:end).*SF_Cone;
Bcone1 = [Bcone1(:,1) Bcone_temp];
disp('the selected Marker for Cone');
disp(MidCone);

%The criteria for the selection of the Triangles are
those symmetrical

disp('please Enter the Markers for First Triangle');
for(i= 1:Mno_Cone/2-1)
sprintf('%d Marker',i) ;
M1_triangle(i,1)=str2num(input('', 's'));
end

disp('please Enter the Markers for Second Triangle');
for(i= 1:Mno_Cone/2-1)
sprintf('%d Marker',i) ;
M2_triangle(i,1)=str2num(input('', 's'));
end
%visible markers are 1,7,9,13,17,22

```

```

%Assigning vector for the Local Steoreotactic
system(reference) from Acone
matched_marker = Acone(ismember(Acone(:,1),Bcone1),:);
Ltriangle1 = Acone(ismember(Acone(:,1),M1_triangle),:);
L_first = Ltriangle1(:,2:end);
Ltriangle2 = Acone(ismember(Acone(:,1),M2_triangle),:);
%Ltriangle2 =
matched_marker(ismember(M2_triangle(:,1),M2_triangle),:);
L_second= Ltriangle2(:,2:end);
%Assigning Vector for Global from Bcone(observed
coordinates)
% the order of matrix is 1, 7, 9, 13, 17, 22
%Bcone = [Bcone(:,1) Bcone(:,2:end).*SF_Cone];
Gtriangle1 = Bcone1(ismember(Bcone1(:,1),M1_triangle),:);
G_first = Gtriangle1(:,2:end);
Gtriangle2 = Bcone1(ismember(Bcone1(:,1),M2_triangle),:);
G_second= Gtriangle2(:,2:end);
[Error_cone_triangle1,Mab_cone_triangle1,T_triangle1_cone]
= first_triangle1(G_first,L_first);
[Error_cone_triangle2,Mab_cone_triangle2,T_triangle2_cone]
= second_triangle2(G_second,L_second);

%Calculation of the combined Transformation Matrix and
Vectors in a
%Single Transformation
%Error_cone_triangle2 = 0.9938;
Esum_cone =
1/Error_cone_triangle1+1/Error_cone_triangle2;

Mab_cone = (((1/Error_cone_triangle1)*
Mab_cone_triangle1) +((1/Error_cone_triangle2)*
Mab_cone_triangle2))/Esum_cone;

Tsum_cone = (((1/Error_cone_triangle1)*
T_triangle1_cone)+ ((1/Error_cone_triangle2)*
T_triangle2_cone))/Esum_cone;

!

disp('The Combined(Averaged) Error in the Cone
is::::');
disp(Mab_cone);
%Transformation Quality Check with Reference Marker

```

```

% The quality of the caddy transformation can be checked
by transforming
% the global coordinates of the phantom base reference
marker, which was
% captured with the Vicon coordinates of the Phantom
base reference
% marker, which was captured with the Vicon cameras,
into local
% coordinates, which are then compared to the
stereotactic coordinates
% measured by the DIL. The coordinates of the reference
marker are in the
% remarkers file.

```

```

disp('Performing the Quality Check of the Cone
Transformation with the REFERENCE MARKER');
disp('Absolute Error in the transformation should be
less than 0.5');
disp('the absolute error is:');
Rm = [9999 9999 9999
      9999 9999 9999
      9999 9999 9999];

```

```

R_PB_g = Rm(:,1);

```

```

% The global coordinate vector of the reference marker
is now transformed
% into local coordinates

```

```

%R_PB_l = Mab_cone'* R_PB_g - Mab_cone'*Tsum_cone;
R_PB_l = Mab_cone\ R_PB_g - Mab_cone \Tsum_cone;
%The DIL coordinates of the reference marker(phantom
base marker 34) are
%looked up next
%Phantom = Phantom_base();
% M_pbase = 10;
%R_target =Phantom(ismember(Phantom(:,1),M_pbase),:);
%R_target = R_target(:,2:end)';
R_target = [-0.066;-0.1676;-356.6439];
%Finally the error of the transformation is defined as
the difference
%between observed and expected coordinates
diff = R_PB_l - R_target;
Error_transformation_Cone = norm(R_PB_l - R_target);
disp(Error_transformation_Cone);

```

```

    if Error_transformation_Cone > 0.5
        disp('The absolute Error present in the Cone
transformation is greater than the 0.5');
        disp('The Error present in the Cone Transformation
contain Systematic Error');
    else disp('the Error in the Cone transformation is
similar to the expected value which is less than the
0.5');
    end
    fid=fopen('Orthogonal_output.xls','a+');
    fprintf(fid,'\n');
    fprintf(fid,'The Error in first Triangle in the Cone is
= %0.5g \n',Error_cone_triangle1);
    fprintf(fid,'The Error in Second Triangle in the Cone
is = %0.5g\n ',Error_cone_triangle2);

    fprintf(fid,'The transformation Error for the Cone is
= %0.9g\n', Error_transformation_Cone);
    fprintf(fid,'\n');
    fclose(fid);
    clear fid;

```

```

%ROUTINE TO CALCULATE TRANSFORMATION ERROR FOR THE CADDY
%TRANSFORMATION

```

```

%this is the Function performs Orthogonal Transformation
for the Selected
%Triangles for Caddy
function
[Error_caddy_triangle1,Error_caddy_triangle2,Mab_caddy,Tsu
m_caddy]= UT_Caddy(MidCaddy,Mno_Caddy,Acaddy,Bcaddy)
global MidCaddy Mno_Caddy Acaddy Bcaddy Mab_caddy
Tsum_caddy
%SF_Caddy = 0.9954641;
%MidCaddy = [ 1;7;9;13;17;22];
%Mno_Caddy = 6;
%Acaddy = CaddyReference();%Reference Caddy Coordinates
%Bcaddy = CaddyActual();%Observed Caddy Coordinates
%Bcaddy = [Bcaddy(:,1) Bcaddy(:,2:end)].*SF_Caddy];
disp('the selected Marker for Caddy');
disp(MidCaddy);

```

```

%The criteria for the selection of the Triangles are
those symmetrical

disp('please Enter the Markers for First Triangle');
for(i= 1:Mno_Caddy/2)
    sprintf('%d Marker',i) ;
    M1_triangle(i,1)=str2num(input('', 's'));
end

disp('please Enter the Markers for Second Triangle');
for(i= 1:Mno_Caddy/2)
    sprintf('%d Marker',i) ;
    M2_triangle(i,1)=str2num(input('', 's'));
end

%visible markers are 1,7,9,13,17,22
%Assigning vector for the Local Steoreotactic
system(reference) from Acaddy
matched_marker = Acaddy(ismember(Acaddy(:,1),Bcaddy),:);
Ltriangle1 = Acaddy(ismember(Acaddy(:,1),M1_triangle),:);
L_first = Ltriangle1(:,2:end);
Ltriangle2 = Acaddy(ismember(Acaddy(:,1),M2_triangle),:);
%Ltriangle2 =
matched_marker(ismember(M2_triangle(:,1),M2_triangle),:);
L_second= Ltriangle2(:,2:end);
%Assigning Vector for Global from Bcaddy(observed
coordinates)
% the order of matrix is 1, 7, 9, 13, 17, 22

Gtriangle1 = Bcaddy(ismember(Bcaddy(:,1),M1_triangle),:);
G_first = Gtriangle1(:,2:end)
Gtriangle2 = Bcaddy(ismember(Bcaddy(:,1),M2_triangle),:);
G_second = Gtriangle2(:,2:end)
[Error_caddy_triangle1,Mab_caddy_triangle1,T_triangle1]=fi
rst_triangle1(G_first,L_first);
[Error_caddy_triangle2,Mab_caddy_triangle2,T_triangle2]=se
cond_triangle2(G_second,L_second);
disp('Error in first Triangle of the Caddy is:');
disp(Error_caddy_triangle1);
disp('Error in Second Triangle of the Caddy is:');
disp(Error_caddy_triangle2);

%Calculation of the combined Transformation Matrix and
Vectors in a
%Single Transformation
Esum_caddy =
1/Error_caddy_triangle1+1/Error_caddy_triangle2;

```

```

Mab_caddy = (((1/Error_caddy_triangle1)*
Mab_caddy_triangle1) +((1/Error_caddy_triangle2)*
Mab_caddy_triangle2))/Esum_caddy;

Tsum_caddy = (((1/Error_caddy_triangle1)* T_triangle1)+
((1/Error_caddy_triangle2)* T_triangle2))/Esum_caddy;

disp('The Combined(Averaged) Error in the Caddy
is::::');
disp(Mab_caddy);
%Transformation Quality Check with Reference Marker
% The quality of the caddy transformation can be checked
by transforming
% the global coordinates of the phantom base reference
marker, which was
% captured with the Vicon coordinates of the Phantom
base reference
% marker, which was captured with the Vicon cameras,
into local
% coordinates , which are then compared to the
stereotactic coordinates
% measured by the DIL. The coordinates of the reference
marker are in the
% remarkers file.

disp('Performing the Quality Check of the Caddy
Transformation with the REFERENCE MARKER');
disp('Absolute Error in the transformation should be
less than 0.5');
disp('the absoulte error is:');
Rm = [9999 9999 9999
      9999 9999 9999
      9999 9999 9999];

R_PB_g = Rm(:,1);

% The global coordinate vector of the reference marker
is now transformed
% into local coordinates

%R_PB_l = Mab_caddy'* R_PB_g - Mab_caddy'*Tsum_caddy
R_PB_l = Mab_caddy \ R_PB_g - Mab_caddy \ Tsum_caddy;
%The DIL coordinates of the reference marker(phantom
base marker 34) are
%looked up next

```

```

Phantom = Phantom_Base();
M_pbase = 34;
R_target = Phantom(ismember(Phantom(:,1),M_pbase),:);
R_target = R_target(:,2:end)';
%Finally the error of the transformation is defined as
the difference
%between observed and expected coordinates
diff = R_PB_1 - R_target;
Error_transformation_Caddy = norm(R_PB_1 - R_target);
disp(Error_transformation_Caddy);
if Error_transformation_Caddy > 0.5
    disp('The absolute Error present in the Caddy
transformation is greater than the 0.5');
    disp('The Error present in the Caddy Transformation
contain Systematic Error');
else disp('the Error in Caddy transformation is similar
to the expected value which is less than the 0.5');

end
fid=fopen('Orthogonal_output.xls','a+');
fprintf(fid,'THE Experiment was performed at %s
\n',datestr(now)) ;
fprintf(fid,'\n');
fprintf(fid,'The Error in first Triangle in the Caddy is
= %0.5g \n',Error_caddy_triangle1);
fprintf(fid,'The Error in Second Triangle in the Caddy
is = %0.5g \n',Error_caddy_triangle2);
%fprintf(fid,'The transformation Error for the Caddy is
= %0.9g\n', Error_transformation_Caddy);
fprintf(fid,'\n');
fclose(fid);
clear fid;

%function to fetch the Observed Coordinates Values for the
Caddy Markers

function [Table]= CaddyActual()

fprintf('Reading a file for Captured Coordinates For CADDY
markers \n');
fprintf('=====');
fprintf('\n');
filename='c:/Caddy_actual.txt';
%[filename,pname] = uigetfile('*.*', 'Select Input File');
u = fopen(filename,'r'); %open input file
fprintf('The file name is %s',filename);
Table = []; %initialize the empty matrix

```



```

while 1
    line = fgetl(u);          %read line
    if ~ischar(line)
        break,
    end
    Table = [Table; str2num(line)]; %convert to number and
add to matrix
end
fprintf('\n');
fclose(u);

```

```

%Function to Read the Coordinates Data for the Cone
Markers from a File
function [Table]= ConeActual()
fprintf('Reading a file for Captured Coordinates For CONE
markers \n');
fprintf('=====');
fprintf('\n');
filename='c:/Cone_actual.txt';
u = fopen(filename,'r');    %open input file
fprintf('The file name is %s',filename);
Table = []; %initialize the empty matrix
while 1
    line = fgetl(u);          %read line
    if ~ischar(line)
        break,
    end
    Table = [Table; str2num(line)]; %convert to number and
add to matrix
end
fprintf('\n');
fclose(u);

```

% Reading reference Coordinate Data From files for the
Caddy Markers%

```

%This function can be used to read text data from a file
function [Table]=CaddyReference()
%function readtable(filename)
fprintf('Reading a file for the Stereotactic Reference
System For CADDY \n');
fprintf('=====');

```

```

fprintf('\n');
fprintf('\n');
filename='c:/caddy_reference.txt';
%[filename,pname] = uigetfile('*.*', 'Select Input File');
u = fopen(filename,'r'); %open input file
fprintf('The file name is %s',filename);
Table = []; %initialize the empty matrix
while 1
    line = fgetl(u); %read line
    if ~ischar(line)
        break,
    end %end when no more lines available
    Table = [Table; str2num(line)]; %convert to number and
add to matrix
end
fprintf('\n');
fclose(u);

```

% Reading reference Coordinate Data From files for Cone%

```

function [Table]= ConeReference()
%function readtable(filename)
fprintf('Reading a file for the Stereotactic Reference
System For CONE \n');
fprintf('=====\n');
fprintf('\n');
fprintf('\n');
filename='c:/cone_reference.txt';
%[filename,pname] = uigetfile('*.*', 'Select Input File');
u = fopen(filename,'r'); %open input file
fprintf('The file name is %s',filename);
Table = []; %initialize the empty matrix
while 1
    line = fgetl(u); %read line
    if ~ischar(line)
        break,
    end %end when no more lines available
    Table = [Table; str2num(line)]; %convert to number and
add to matrix
end
fprintf('\n');
fclose(u);

```

%Calculation of the Distance between the Visible markers

```
function[Adist]= distance(A)
%Acaddy=CaddyActual()
%A=Acaddy(:,2:end)
[m n]=size(A);

%end
Adist=zeros(m);
for i=1:m
    for j=1:m
        Adist(i,j)=sqrt(sum((A(j,:)-A(i,:)).^2));
    end
end
end
```

%Function to read the file for the image offset calculated by the Image

%processing program

```
function [Table]= Image_offset()
```

```
fprintf('Reading a file for Phantom Base \n');
```

```
fprintf('=====');
```

```
fprintf('\n');
```

```
filename='C:\Documents and
```

```
Settings\raj\Desktop\data2\image_offset.txt';
```

```
u = fopen(filename,'r'); %open input file
```

```
fprintf('The file name is %s',filename);
```

```
Table = []; %initialize the empty matrix
```

```
while 1
```

```
    line = fgetl(u); %read line
```

```
    if ~ischar(line)
```

```
        break,
```

```
    end
```

```
        %end when no more lines available
```

```
    Table = [Table; str2num(line)]; %convert to number and
```

```
add to matrix
```

```
end
```

```
fprintf('\n');
```

```
fclose(u);
```

%Function to Read the Coordinates Data for the Cone Markers from a File

```
function [Table]= Offset()
```

```
fprintf('Reading a file for Captured Coordinates For CONE markers \n');
```

```

fprintf('====');
fprintf('\n');
filename='C:\Documents and
Settings\raj\Desktop\data2\Offset.txt';
u = fopen(filename,'r');    %open input file
fprintf('The file name is %s',filename);
Table = [];    %initialize the empty matrix
while 1
    line = fgetl(u);        %read line
    if ~ischar(line)
        break,
    end
    %end when no more lines available
    Table = [Table; str2num(line)]; %convert to number and
add to matrix
end
fprintf('\n');
fclose(u);

```

APPENDIX C
LEAST SQUARE BASED TRANSFORMATION

2.1 Least Square based Transformation

Least Square Problem Solution is a mathematical optimization technique to find an approximate solution for a system of linear equations that has no exact solution. For our application, given is a matrix A in which the i^{th} column corresponds to the three coordinates of the i^{th} marker in the local reference system and another matrix B in which the i^{th} column corresponds to the three coordinates of the i^{th} marker in the global reference system. We search for a linear transformation, represented by a 3×3 matrix X, that transforms matrix A into matrix B:

$$AX = B \quad (1)$$

Provided data on at least three markers are available, in which case the matrices A and B are also 3×3 matrices, equation (1) can be solved

$$X = A \setminus B \quad (2)$$

In our case A holds the reference coordinates of the markers while B holds observed Vicon coordinates.

This equation gives the transformation equation required to transform the local coordinates into the global coordinates.

In case of transformation from global to the local coordinates the following equation can be used:

$$Y = B \setminus A \quad (3)$$

Hence the transformation matrix produced by equation 2 and 3 further refined and used to determine the alignment error.

2.2 Subroutine to perform least Square based transformation

```
%This function calculates the distance between the central  
% cone beam axis and the phantom base target marker based  
% on a Least Squares Transformation from the Local  
% Coordinate System to the Global Coordinates.  
% This function takes Scaling Factor for Caddy and Cone  
% along with the Marker offset and the Calibration Offset  
% as input and yields x, y and X axis net error in the  
% transformation.
```

```
Function [E_ta1, E_ta2, E_ta3]=  
LS_transform(SF_Caddy,SF_Cone)  
%function [Xcaddy]=LS_transform(SF_Caddy,SF_Cone)  
global Ecaddy Econe AcaddyDist AconeDist McaddyDist  
MconeDist Etotal_Caddy  
global MidCaddy Mno_Caddy Acaddy Bcaddy Mab_cone Mab_caddy  
Tsum_cone Tsum_cone
```

```
Acaddy1 = CaddyReference();%Fetching the Stereotactic  
Coordinate values for Caddy Markers  
Aconel = ConeReference();% Fetching the Stereotactic  
Coordinate Values for Cone Markers
```

```
Bcaddy1 = CaddyActual();%Fetching the Observed Coordinate  
Values for Caddy Markers
```

```

Bcone1 = ConeActual();%Fetching the Observed Coordinate
Values for Cone Markers
%SF_Cone = 0.998;
Mno_Caddy = input('How many markers are visible for CADDY
\n');
Mno_Cone = input('How many markers are visible for
CONE\n');
%k=zeros(Mno_Caddy,1);
%j=zeros(Mno_Cone,1);
MidCaddy=[];
MidCone=[];
fid=fopen('Least_Square_output.xls','a+');
display('Visible markers for Caddy are:');
MidCaddy = Bcaddy1(:,1);
disp(MidCaddy);
%Asking for the markers Visible to the CAMERA in the
CADDY
%for(i= 1:1:Mno_Caddy)
%   sprintf('%d Marker',i);
%   MidCaddy(i,1)=str2num(input('', 's'));
% end

%Asking for the Markers Visible to the CAMERA in the CONE

display('Visible markers for Cone are:');
MidCone= Bcone1(:,1);
disp(MidCone);
%for(i= 1:1:Mno_Cone)
%   sprintf('%d Marker',i) ;
%MidCone(i,1)=str2num(input('', 's'));
%end

%Initializing the Vectors

AcaddyDist=zeros(Mno_Caddy, Mno_Caddy);
AconeDist=zeros(Mno_Cone, Mno_Cone);
McaddyDist=zeros(Mno_Caddy,Mno_Caddy);
MconeDist=zeros(Mno_Cone,Mno_Cone);
matched_Caddy = Acaddy1(ismember(Acaddy1(:,1),Bcaddy1),:);
matched_Cone = Acone1(ismember(Acone1(:,1),Bcone1),:);
%Calling the Corresponding function to load the file into
respective vector

AcaddyDist
=distance(matched_Caddy(:,2:end));%Actual(Reference)Distan
ce Between Caddy markers

```



```

AconeDist =
distance(matched_Cone(:,2:end));%Actual (Reference)
Distance Between Cone markers
McaddyDist =distance(Bcaddy1(:,2:end));%Measured Distance
Between Caddy Markers
MconeDist = distance(Bcone1(:,2:end));%Measured Distance
Between Cone Markers
[SF_Caddy,Ecaddy,SF_Cone,Econe]=
Scaling_Factor(AcaddyDist,McaddyDist,AconeDist,MconeDist);

Bcaddy =[Bcaddy1(:,1) Bcaddy1(:,2:end).*SF_Caddy]; %
Elemental Multiplication of the Scaling factor with the
Observed Caddy
matched_Caddy =[];
matched_Cone = [];
Bcone = [Bcone1(:,1) Bcone1(:,2:end).*SF_Cone];

matched_Caddy = Acaddy1(ismember(Acaddy1(:,1),Bcaddy),:);
[m n]=size(matched_Caddy);
matched_Cone = Acone1(ismember(Acone1(:,1),Bcone),:);

Acaddy_new = [matched_Caddy(:,2:end) ones(m,1)];
Acone_new =[ matched_Cone(:,2:end)
ones(size(matched_Cone,1),1)];

%CADDY Transformation
t1=[Bcaddy(:,2:end) ones(size(Bcaddy,1),1)];
Xcaddy = Acaddy_new \ t1;
%Xcaddy= Xcaddy(1:end-1,2:end)
Ycaddy = t1 \ Acaddy_new;

%det(Xcaddy);

Ecaddy = (Acaddy_new * Xcaddy -t1); %%%%Error
Generated by the Caddy

%NORM(X,'fro') is the Frobenius norm,
sqrt(sum(diag(X'*X))).

SSR_Caddy = norm(Ecaddy,'fro')^2;

%CONE TRANSFORMATION
t2=[Bcone(:,2:end),ones(size(Bcone,1),1)];
Xcone = Acone_new \ t2;

```

```

Ycone = t2 \ Acone_new;

Eccone = (Acone_new * Xcone -t2); % Error Generated by the
Cone

SSR_Cone = norm(Eccone,'fro')^2;

mpbase = input('Enter the Phantom Base Pin for Least
Square transformation ');
Aph_base = Phantom_Base();
matched_phantom =
Aph_base(ismember(Aph_base(:,1),mpbase),2:end);

phantom_target = [matched_phantom(:)
ones(size(matched_phantom(:)),1)];
phantom_target = [phantom_target(:,1)' 1];

%target point in global Coordinates
T_4_g = phantom_target * Xcaddy;

P_04_l=[0 0 -200 1];%Point on the Central Beam Axis 20 cm
from the cone origin

U_4_l=[0;0;1;1];

P_04_g = P_04_l*Xcone;
U_4_g=U_4_l'*Xcone- Xcone(4,:);

Tg= T_4_g(:,1:end-1)';

P0_g = P_04_g(1:end-1)';

U_g = U_4_g(:,1:end-1)';

t_g = P0_g- Tg;

E_ta = -dot(t_g,U_g)*U_g + t_g;

%Correction for any actual offset + vertical offset =
marker above axis +
%horizontal offset = marker inferior(toward halo) from
axis) and
%calibration offset (in mm) (x,y,z);

moff = Image_offset();
voff = moff(1);

```

```

hoff = moff(2);

Cali_off = Offset();
x_off = Cali_off(1);
y_off = Cali_off(2);

z_off = Cali_off(3);

E_ta_1=E_ta(1)-x_off;
E_ta_2 = E_ta(2)-voff-y_off;
E_ta_3 = E_ta(3)-hoff-z_off;
E_total = [E_ta_1 E_ta_2 E_ta_3];
E_norm = norm(E_total);

fid = fopen('Least_Square_output.xls','a+');
fprintf(fid,'THE Experiment was performed at %s
\n',datestr(now)) ;
fprintf(fid,'\n');
fprintf(fid,'The SSRcaddy is %0.5g\n',SSR_Caddy);
fprintf(fid,'The SSRcone is %0.5g\n',SSR_Cone);
fprintf(fid,'The Error in X direction is
%0.5g\n',E_ta_1);
fprintf(fid,'The Error in Y direction is
%0.5g\n',E_ta_2);
fprintf(fid,'The Error in Z direction is
%0.5g\n',E_ta_3);
fprintf(fid,'The Normalized Error in is %0.5g\n',E_norm);
fprintf(fid,'\n');
fprintf(fid,'\n');
fclose(fid);
)

```

APPENDIX D
CONSTRAINED LEAST SQUARE BASED TRANSFORMATION

3.1 Constrained Least Square Based Transformation

This method uses the same process as the Least Square Based transformation but it is constrained. The starting point for this method is to set the initial value equal to that generated by the Least Square based transformation. Once the transformation matrix is calculated it is used to calculate the overall alignment errors.

3.2 Subroutines for Constrained Least Square Based Transformation

```
%This is the constrained Least Square based Coordinate
transformation routine.
%This function calculates the distance between the central
%
%cone beam axis and the phantom base target marker based %
%on a Constrained Least Squares Transformation from the
Local
% Coordinate System to the Global Coordinates.
% This function takes Scaling Factor for Caddy and Cone
% along with the Marker offset and the Calibration Offset
%
%as input and yields x, y and X axis net error in the
% transformation.
%Produces output as alignment errors

function [E_ta1, E_ta2, E_ta3]= LS_Constrained()
global E_ta1 E_ta2 E_ta3 SF_Caddy SF_Cone M_off Cal_off
Acaddy_new t1
global Ecaddy Econe AcaddyDist AconeDist McaddyDist
MconeDist Etotal_Caddy
global MidCaddy Mno_Caddy Acaddy Bcaddy Mab_cone Mab_caddy
Tsum_cone Tsum_cone Acone_new t2

%SF_Caddy = 0.9947;
Acaddy1 = CaddyReference();%Fetching the Stereotactic
Coordinate values for Caddy Markers
Acone1 = ConeReference();% Fetching the Stereotactic
Coordinate Values for Cone Markers

Bcaddy1 = CaddyActual();%Fetching the Observed Coordinate
Values for Caddy Markers
Bcone1 = ConeActual();%Fetching the Observed Coordinate
Values for Cone Markers
%SF_Cone = 0.998;
```

```

Mno_Caddy = input('How many markers are visible for CADDY
\n');
Mno_Cone = input('How many markers are visible for
CONE\n');
%k=zeros(Mno_Caddy,1);
%j=zeros(Mno_Cone,1);
MidCaddy=[];
MidCone=[];
fid=fopen('LS_CONSTRAINED_output.xls','a+');
display('Visible markers for Caddy are:');

MidCaddy = Bcaddy1(:,1);
disp(MidCaddy);
%Asking for the markers Visible to the CAMERA in the CADDY
%for(i= 1:1:Mno_Caddy)
%   sprintf('%d Marker',i);
% MidCaddy(i,1)=str2num(input('', 's'));
% end

%Asking for the Markers Visible to the CAMERA in the CONE

display( 'Visible markers for Cone are:');
MidCone= Bconel(:,1);
disp(MidCone);

%Used Cone markers are
%Used_Cone_markers = [M1_triangle;M2_triangle]
%for(i= 1:1:Mno_Cone)
%   sprintf('%d Marker',i) ;
%MidCone(i,1)=str2num(input('', 's'));
%end

%Initializing the Vectors

AcaddyDist=zeros(Mno_Caddy, Mno_Caddy);
AconeDist=zeros(Mno_Cone, Mno_Cone);
McaddyDist=zeros(Mno_Caddy,Mno_Caddy);
MconeDist=zeros(Mno_Cone,Mno_Cone);
matched_Caddy = Acaddy1(ismember(Acaddy1(:,1),Bcaddy1),:);
matched_Cone = Aconel(ismember(Aconel(:,1),Bconel),:);
%Calling the Corresponding function to load the file into
respective vector

```

```

AcaddyDist
=distance(matched_Caddy(:,2:end));%Actual(Reference)Distance
Between Caddy markers
AconeDist =
distance(matched_Cone(:,2:end));%Actual(Reference)
Distance Between Cone markers
McaddyDist =distance(Bcaddy1(:,2:end));%Measured Distance
Between Caddy Markers
MconeDist = distance(Bcone1(:,2:end));%Measured Distance
Between Cone Markers
[SF_Caddy,Ecaddy,SF_Cone,Econe]=
Scaling_Factor(AcaddyDist,McaddyDist,AconeDist,MconeDist);

Bcaddy =[Bcaddy1(:,1) Bcaddy1(:,2:end).*SF_Caddy]; %
Elemental Multiplication of the Scaling factor with the
Observed Caddy
matched_Caddy = [];
matched_Cone = [];
Bcone = [Bcone1(:,1) Bcone1(:,2:end).*SF_Cone];

matched_Caddy = Acaddy1(ismember(Acaddy1(:,1),Bcaddy),:);
[m n]=size(matched_Caddy)
matched_Cone = Acone1(ismember(Acone1(:,1),Bcone),:);

%Acaddy_new = [matched_Caddy(:,2:end) ones(m,1)];
Acaddy_new = [matched_Caddy(:,2:end)
ones(size(matched_Caddy),1)];
Acone_new =[ matched_Cone(:,2:end)
ones(size(matched_Cone,1),1)];
Econe = [];
Ecaddy = [];
%CADDY Transformation
t1=[Bcaddy(:,2:end) ones(size(Bcaddy,1),1)];
%t1 =Bcaddy(:,2:end);

Xcaddy = Acaddy_new \ t1;%its equivalent to
||Acaddy_new.Xcaddy - t1||
% such that ||B.Xcaddy - B.t1|| <=
alpha(tolerance)
%t1=t1(:);
%Acaddy
x0 = Xcaddy;
%x0 = 0;
options = [];

```

```

[Xcaddy_constrained,fval]=
fmincon(@objective_function_x_caddy,x0,[],[],[],[],[],[],[],@
constrained_function,options);
%fmincon(@objective_function_x_caddy,x0,[],[],[],[],[],[],@
@constrained_function,options)
Ycaddy = t1 \ Acaddy_new; %its equivalent to ||t1.Ycaddy -
Acaddy_new||
%
% such that ||B.Ycaddy - B.t1||
<=alpha(tolerance)
y0 = Ycaddy;
%y0=0;
options = [];
[Ycaddy_constrained,fval]=
fmincon(@objective_function_y_caddy,y0,[],[],[],[],[],[],@
constrained_function,options);

%Xcaddy_constrained
%Ycaddy_constrained

Ecaddy_constrained = (Acaddy_new * Xcaddy_constrained -
t1); %Error Generated by the Caddy

%NORM(X,'fro') is the Frobenius norm,
sqrt(sum(diag(X'*X))).

SSR_Caddy_constrained = norm(Ecaddy_constrained,'fro')^2;

%CONE TRANSFORMATION

t2=[Bcone(:,2:end),ones(size(Bcone,1),1)];
Xcone = Acone_new \ t2;
%[m n]= size(matched_Cone(1:end-2,2:end))
%Acone_new = [ matched_Cone(1:end-2,2:end)
ones(size(matched_Cone,1)-2,1)];
%t2=[Bcone(1:end-2,2:end),ones(size(Bcone,1)-2,1)];
%x1=0;
x1 = Xcone;
%Xcone_constrained = Xcone;
%Acone_new = Acone_new(1:end-2,:);
%t2 = t2(1:end-2,:);
options = [];
[Xcone_constrained,fval]=
fmincon(@objective_function_x_cone,x1,[],[],[],[],[],[],@
onstrained_function,options);

```



```

Ycone = t2 \ Acone_new;
y1 = Ycone;
%y1 =0;
%Xcone_constrained = qr(Xcone');
options = [];
[Ycone_constrained,fval]=
fmincon(@objective_function_y_cone,y1, [], [], [], [], [], [], [], @c
onstrained_function,options);
%Ycone_constrained = Ycone;
%Ycone_constrained = qr(Ycone');
Econe_constrained = (Acone_new * Xcone_constrained -t2); %
Error Generated by the Cone
%generating 3*3 matrix
%Econe_constrained = Econe_constrained(:,1:end-1);
SSR_Cone_constrained = norm(Econe_constrained,'fro')^2;
%Calculating Alignment Errors
mpbase = input('Enter the Phantom Base Pin for Constrained
Least Square transformation ');
Aph_base = Phantom_Base();
matched_phantom =
Aph_base(ismember(Aph_base(:,1),mpbase),2:end);

phantom_target = [matched_phantom(:)
ones(size(matched_phantom(:)),1)];
phantom_target = [phantom_target(:,1)' 1];

%target point in global Coordinates
T_4_g = phantom_target * Xcaddy_constrained;

P_04_l=[0 0 -200 1];%Point on the Central Beam Axis 20 cm
from the cone origin

U_4_l=[0;0;1;1];

P_04_g = P_04_l* Xcone_constrained;
U_4_g=U_4_l'*Xcone_constrained - Xcone_constrained(4,:);

Tg= T_4_g(:,1:end-1)';

P0_g = P_04_g(1:end-1)';

U_g = U_4_g(:,1:end-1)';

t_g = P0_g- Tg;

E_ta = -dot(t_g,U_g)*U_g + t_g;

```

```

%Correction for any actual offset + vertical offset =
marker above axis +
%horizontal offset = marker inferior(toward halo) from
axis) and
%calibration offset (in mm) (x,y,z);

moff = Image_offset();
voff = moff(1);
hoff = moff(2);

Cali_off = Offset();
x_off = Cali_off(1);
y_off = Cali_off(2);

z_off = Cali_off(3);

E_ta_1=E_ta(1)-x_off;
E_ta_2 = E_ta(2)-voff-y_off;
E_ta_3 = E_ta(3)-hoff-z_off;
E_total = [E_ta_1 E_ta_2 E_ta_3];
E_norm = norm(E_total);

fid = fopen('LS_CONSTRAINED_output.xls','a+');
fprintf(fid,'THE Experiment was performed at %s
\n',datestr(now)) ;
fprintf(fid,'\n');
fprintf(fid,'The SSRcaddy is
%0.5g\n',SSR_Caddy_constrained);
fprintf(fid,'The SSRcone is
%0.5g\n',SSR_Cone_constrained);
fprintf(fid,'The Error in X direction is
%0.5g\n',E_ta_1);
fprintf(fid,'The Error in Y direction is
%0.5g\n',E_ta_2);
fprintf(fid,'The Error in Z direction is
%0.5g\n',E_ta_3);
fprintf(fid,'The Normalized Error in is %0.5g\n',E_norm);
fprintf(fid,'\n');
fprintf(fid,'\n');
fprintf(fid,'\n');
fprintf(fid,'\n');
fclose(fid);

```

```

%Routine to set the objective function

%For caddy

% X function

function f = objective_function_x_caddy(x)
global Acaddy_new t1
%disp(Acaddy_new);
%disp(x);
%disp(t1);
f = norm(Acaddy_new*x - t1);%first one to calculate Xcaddy

%Y Function

function f = objective_function_y_caddy(y)
global Acaddy_new t1
f = norm (t1*y- Acaddy_new);

%Objective Function required for Cone

%For X

function f = objective_function_x_cone(x)
global Acone_new t2
f = norm(Acone_new*x - t2);%first one to calculate Xcaddy
%disp(f);

% For Y

function f = objective_function_y_cone(y)%equation for
Ycaddy
%buit new function
global Acone_new t2
%disp(Acone_new);
%disp(t2)
f = norm(t2*y' - Acone_new);

%Constrained Function

function [C,Constrained_Cost]= constrained_function(x)
x_bar = [ x(1:3,1:3)];%We just need 3*3 matrix
%Constrained_cost function
Constrained Cost = norm(x_bar'*x_bar - eye(3));
C=0;

```

REFERENCES

1. Arun, K., et al, "Least-square fitting of two 3D point sets", IEEE Transactions Pattern Analysis and Machine Intelligence Vol.9, No.5, June 1987, pp.698-700
2. Ford, R., et al, "Effects of Coordinate Transformation and Differencing Schemes on Matrix Structure and Convergence History", Applied Mathematics and Computation, Vol.77, June 1996, pp.53-66
3. Chatterjee S., and Price B., Regression Analysis by Example, Second Edition, John Wiley and Sons, Inc, 1991, pp.123-126
4. Leaver, D., "Stereotactic Radiosurgery," Radiation Therapist, Vol.14, No.1, Spring 2005
5. Cantarella, Jason and Piatek, M., "tsnnls: A solver for large sparse least squares problems with non-negative variables," www.ada.math.uga.edu/research/, Aug 11, 2005.
6. Sachs, J., "Digital Image Basics," Digital Lights and colors, 1999.
7. Loan, C. V., "A Survey of Matrix Computations," Handbooks in OR and MS, Vol.3, 1992.
8. Golub, G.H., "An Analysis of the Total Least Squares Problem," Vol. 17, No.6, December 1980.
9. Zhu, Y. M., "Recursive least squares with linear constraints," Proceedings of the 38th IEEE Conference on Decision and Control, 1999.
10. Weaver, K., et al, "A Ct-Based Computerized Treatment Planning System For I-125 Stereotactic Brain Implants", Int J. Radiation Oncology Biology Physics Vol.18, pp. 445-454
11. Golub, G.H., Van Loan, C.F., "Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)," Third Edition, 1996