

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Ercan AVŞAR

**SIMULATION OF A RING FIXATOR SYSTEM WITH PATIENT-SPECIFIC
BONE MODELING**

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA, 2016

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

**SIMULATION OF A RING FIXATOR SYSTEM WITH PATIENT-SPECIFIC
BONE MODELING**

Ercan AVŞAR

PhD THESIS

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Doctor of Philosophy by the board of jury on 22/06/2016.

.....
Asst. Prof. Dr. M. Kerem ÜN
SUPERVISOR

.....
Prof. Dr. İ. Deniz AKÇALI
MEMBER

.....
Prof. Dr. Ulus ÇEVİK
MEMBER

.....
Asst. Prof. Dr. Mehmet K. BARAN
MEMBER

.....
Asst. Prof. Dr. Hüseyin MUTLU
MEMBER

This PhD Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.

Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

Not: The usage of the presented specific declarations, tables, figures, and photographs either in this thesis or in any other reference without citation is subject to "The law of Arts and Intellectual Products" number of 5846 of Turkish Republic

ABSTRACT

PhD THESIS

SIMULATION OF A RING FIXATOR SYSTEM WITH PATIENT-SPECIFIC BONE MODELING

Ercan AVŞAR

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Supervisor : Asst. Prof. Dr. M. Kerem ÜN
Year: 2016, Pages: 119
Jury : Asst. Prof. Dr. M. Kerem ÜN
: Prof. Dr. İbrahim Deniz AKÇALI
: Prof. Dr. Ulus ÇEVİK
: Asst. Prof. Dr. Mehmet Kadir BARAN
: Asst. Prof. Dr. Hüseyin MUTLU

External ring fixators are widely used in orthopaedics for the purposes of fracture fixation, bone lengthening and deformity correction. In these fixators, the clinician typically brings the bone fragments to an anatomically desired position by changing the lengths of the rods connecting the fixator rings. This task is accomplished by the clinician based on experience and expertise. As an alternative, commercial systems exist where the same task is automatized with the help of an accompanying software that implements a mathematical model of the fixator. Proper visualization tools, that will allow the clinician to foresee the outcome of the treatment, are not present in the currently available software systems. In this thesis, we have developed methods to convert the model of a healthy bone, called the *canonical model*, to the patient-specific pathological (such as deformed) bone model. A graphical user interface (GUI) has been programmed with an advanced visualization and simulation tool that can automatically create the patient-specific bone model based on this approach. The bone modeling algorithm has been evaluated on a total of 19 different tibia and femur models and found to be effective. Furthermore, the visualization tool has been tested in the simulation of four orthopaedic procedures. In all examples, the visualization tool has provided a realistic depiction of the bone-fixator system and the treatment procedure. We believe that the developed GUI equipped with the visualization module could be a useful clinical tool where the clinician can visualize the applied treatment or evaluate different treatment scenarios *a priori* per patient.

Key Words: External Fixation, Graphical User Interface, 3D Simulation, Bone Model Generation, Patient-Specific Model.

ÖZ

DOKTORA TEZİ

HASTAYA ÖZEL KEMİK MODELLEMESİ İLE BİR HALKA FİKSATÖR SİSTEMİNİN SİMÜLASYONU

Ercan AVŞAR

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Yrd. Doç. Dr. M. Kerem ÜN
Yıl: 2016, Sayfa: 119
Jüri : Yrd. Doç. Dr. M. Kerem ÜN
: Prof. Dr. İbrahim Deniz AKÇALI
: Prof. Dr. Ulus ÇEVİK
: Yrd. Doç. Dr. Mehmet Kadir BARAN
: Yrd. Doç. Dr. Hüseyin MUTLU

Harici fiksatorler ortopedi alanında kırık tespiti, boy uzatma ve deformite düzeltme amaçları için yaygın olarak kullanılmaktadır. Bu fiksatorlerde tipik olarak klinisyen, fiksator halkalarını birbirine bağlayan çubukların boylarını değiştirerek kemik parçalarını anatomik olarak istenen pozisyona getirir. Bu işlem klinisyen tarafından, onun tecrübe ve uzmanlığına dayanılarak gerçekleştirilir. Alternatif olarak, fiksatorün matematiksel hesaplarını yapan bir yazılım yardımıyla aynı işlemlerin otomatize edildiği ticari sistemler bulunmaktadır. Klinisyenin tedavinin sonucunu önceden görmesine olanak sağlayacak uygun görselleştirme araçları günümüzde kullanılan yazılım sistemlerinde mevcut değildir. Bu tezde *kanonik model* ismindeki sağlıklı kemik modelini hastaya özel patolojik (örneğin deforme) kemik modeline dönüştüren yöntemler geliştirdik. Hastaya özel kemik modellerini otomatik olarak oluşturan, görselleştirme ve simülasyon araçlarına sahip bir kullanıcı arayüzü programlanmıştır. Kemik modelleme algoritması 19 farklı kemik modeli üzerinde değerlendirilmiştir ve algoritmanın efektif olduğu sonucuna varılmıştır. Buna ek olarak, görselleştirme aracı dört ortopedik uygulamanın simülasyonu şeklinde test edilmiştir. Örneklerin tamamında görselleştirme aracı kemik-fiksator sistemini ve tedavi sürecini gerçekçi olarak göstermiştir. Klinisyenin uygulanan tedaviyi görselleştirmesi ve farklı tedavi senaryolarını önceden değerlendirmesine olanak sağlayan görselleştirme aracı işle donatılmış bir arayüzün klinik olarak faydalı bir araç olacağına inanmaktayız.

Anahtar Kelimeler: Harici Fiksasyon, Kullanıcı Arayüzü, 3B Simülasyon, Kemik Modeli Oluşturma, Hastaya Özel Model.

ACKNOWLEDGEMENTS

I would like to present my endless gratitude to my supervisor Asst. Prof. Dr. Kerem ÜN for his support throughout this PhD journey. He has made a lot of valuable contributions to my academic knowledge and experience as well as to my intellectual life. I feel very lucky to have the chance to work with him.

I wish to express my special thanks to Prof. Dr. İbrahim Deniz AKÇALI. I think that he is one of the rare academicians bound to his work with passion. I am thankful to Asst. Prof. Dr. Ahmet AYDIN for sharing his knowledge and being more than a colleague.

I also want to thank to Asst. Prof. Dr. Ömer Sunkar BİÇER and Dr. Ömer Faruk EREN for their valuable help in the medical context of the thesis.

I would like to thank to Prof. Dr. Ulus ÇEVİK, Asst. Prof. Dr. Mehmet K. BARAN, and Asst. Prof. Dr. Hüseyin MUTLU for their valuable suggestions and comments to my thesis.

This thesis is one of the outcomes of a larger scientific project and I want to thank all project members for their contribution during this time.

I thank to my family for helping and supporting me throughout my entire life.

Finally, I can not express my gratitude enough to those people who deceived me, left me alone when I needed them, and made me learn how long a night can be. I have realized how powerful I am thanks to them.

The financial support of the Scientific and Technological Research Council of Turkey (TÜBİTAK) during my Ph.D. is gratefully acknowledged.

CONTENTS	PAGE
ABSTRACT	I
ÖZ	II
ACKNOWLEDGEMENTS	III
CONTENTS	IV
LIST OF TABLES	VI
LIST OF FIGURES	VII
LIST OF ABBREVIATIONS	XII
1. INTRODUCTION	1
1.1. Ring Fixators	2
1.2. Computer Based Systems	3
1.2.1. Taylor Spatial Frame	4
1.2.2. Smart Correction	5
1.2.3. TrueLok Hexapod System	6
1.2.4. Ortho-SUV	7
1.3. Modeling of Patient-Specific Bones and Simulation External Ring Fixators..	8
1.4. Objective and Contribution of the Thesis	12
1.5. Outline of the Thesis	13
2. BACKGROUND	15
2.1. CORA Method and Surgery Planning	15
2.2. Free Form Deformation	22
3. GENERATION OF PATIENT-SPECIFIC BONE MODELS	25
3.1. Model Creation	25
3.2. Model Manipulation for Undeformed Bones	27
3.2.1. Model Manipulation for Undeformed Bones with Linear Mapping	28
3.2.2. Model Manipulation for Undeformed Bones with FFD	37
3.3. Model Manipulation for Deformed Bones	39
3.3.1. Scaling the Canonical Model in Axial and Transverse Directions	39
3.3.2. Deforming the Canonical Model According to Bone Image	42
3.4. Splitting the Final Model and Modeling of Callus Formation	48

3.4.1. Splitting the Final Model	48
3.4.2. Modeling of Callus Formation.....	51
3.5. Error Calculation	53
4. EXTERNAL FIXATOR TYPES	57
4.1. Gough – Stewart Platform as External Fixator	57
4.1.1. Inverse Kinematics	57
4.1.2. Direct Kinematics	63
4.2. The Lambda Fixator	66
4.2.1. Structure of the Lambda Fixator	67
4.2.2. Inverse Kinematics	69
4.2.3. Direct Kinematics	70
5. GRAPHICAL USER INTERFACE	73
5.1. GUI for SP-type External Fixator	73
5.1.1. The <i>Treatment</i> Tab.....	75
5.1.2. The <i>Checking X-Ray Data</i> Tab-.....	77
5.1.3. The <i>Simulation</i> Tab.....	77
5.2. GUI for Lambda Fixator	78
5.3. Computer Implementation.....	80
6. BONE MODELING EXAMPLES.....	81
6.1. Examples of Undeformed Bone Modeling.....	81
6.2. Examples of Deformed Bone Modeling.....	85
7. DEMONSTRATION OF THE GUI.....	91
7.1. Example 1: Femur Fracture Fixation with SP-type Fixator	91
7.2. Example 2: Tibia Fracture Fixation with SP-type Fixator	98
7.3. Example 3: Deformity Correction with SP-type Fixator	103
7.4. Example 4: Tibia Fracture Fixation with Lambda Fixator.....	104
8. CONCLUSION	109
8.1. Concluding Remarks and Discussion.....	109
8.2. Future Work	110
REFERENCES.....	113
BIOGRAHPY	119

LIST OF TABLES

PAGE

Table 4-1. Actuation functions and associated constraints. Here, i and j represent numbers for rods ($i = 1-6$) and treatment steps ($j = 1-n$), respectively.....	65
Table 6.1. Average and normalized errors calculated for the generated patient-specific undeformed bone models (Linear mapping).....	84
Table 6.2. Average and normalized errors calculated for the generated patient-specific undeformed bone models (FFD).....	85
Table 6.3. Average and normalized errors calculated for the generated patient-specific deformed bone models (fourth order polynomial fit)	89
Table 6.4. Average and normalized errors calculated for the generated patient-specific deformed bone models (Cubic splines fit).....	90

LIST OF FIGURES

PAGE

Figure 1.1. Illustrations of pin (a) and ring (b) fixators (adopted from http://www.bonfixator.com/).....	2
Figure 1.2. Taylor Spatial Frame	4
Figure 1.3. Visualization of bone-fixator system in TSF software in the initial (a) and final (b) positions	5
Figure 1.4. Frame of Smart Correction system (a) and the associated visualization tool (b).....	6
Figure 1.5. Frame of TL-HEX system (a) and the associated visualization tool (b)	7
Figure 1.6. Frame of ortho-SUV structure (a) and bone edges drawn on the X-ray image (b)	8
Figure 1.7. External fixation system (left), X-ray image of the system taken from AP view (center) and generated bone-fixator system according to the X-ray image (right).....	13
Figure 2.1. Frontal plane anatomic and mechanical axes for tibia (a) and femur (b). Blue dots represent the joint center points.....	16
Figure 2.2. Frontal plane anatomic axis planning on tibia. The mid-diaphyseal line for proximal and distal parts are shown in red and yellow, respectively (a). The same planning performed on femur. The mid-diaphyseal line for proximal and distal parts are shown in red and blue, respectively (b). (from (Paley, 2005))	18
Figure 2.3. Frontal plane anatomic axis planning in tibia. Extra anatomic axis line referenced to the ankle joint orientation line is shown in blue (a) Similar planning performed on femur (b). (from (Paley, 2005)).....	19
Figure 2.4. Tibial anatomic axis planning on AP (a) and L (b) views of X-ray images.....	20
Figure 2.5. Different osteotomy options in bone deformity correction: (Top to bottom) opening, neutral and closing wedge angulations. The red dots indicate the rotation axes (Paley, 2005). The circled case,	

where no bone removal is performed, is the preferred case and the algorithm described in this work is based on this case.	21
Figure 2.6. A 3D object before (a) and after (b) FFD. (from (Sederberg and Parry, 1986)).....	22
Figure 2.7. A cylinder (a) bent with FFD (b) by changing the locations of control points (white spheres)	24
Figure 3.1 Original (a) and canonical (b) tibia models.....	27
Figure 3.2. The projection of canonical model in AP (a) and L (b) directions with model vertices shown as red dots.....	29
Figure 3.3. AP X-ray of tibia (a). Output of canny edge detection (b). Edges after manual intervention (c).	30
Figure 3.4. Lateral (L) X-ray of tibia (a). Output of canny edge detection (b). Edges after manual intervention (c).	31
Figure 3.5. Illustration of points P_1 and P_2 on bone edges (left) and R_1 and R_2 on the canonical model profile (right).....	33
Figure 3.6. Actual bone and the corresponding canonical bone.....	34
Figure 3.7. Cross-sections where thicknesses and scaling factors are calculated	36
Figure 3.8. The canonical bone model before (a) and after (b) transverse scaling together with the control points in their original positions (magenta dots) and displaced positions (blue dots).	38
Figure 3.9. AP X-ray of tibia (a). Output of canny edge detection (b). Edges after manual intervention (c).	40
Figure 3.10. Joint orientation lines, proximal and distal anatomic axes (red lines) (a) and segments where thickness is measured (blue lines) (b).	40
Figure 3.11. The canonical bone model before (a) and after (b) transverse scaling together with the control points in their original positions (magenta dots) and displaced positions (blue dots).	41
Figure 3.12. The anatomic axes and joint orientation lines (red lines), and the fitted fourth order polynomial (blue curve).....	42
Figure 3.13. The anatomic axes and joint orientation lines (red lines), and the fitted cubic splines (green and blue curves)	44

Figure 3.14. Functions obtained with fourth order polynomial fit (a) and cubic splines fit (b) superposed on the edge images.....	46
Figure 3.15. Final canonical bone models after deformation according to the fourth order polynomial (a) and the cubic splines (b).....	48
Figure 3.16. Vertices of triangles of interest (v) and detected intersection points (p)	50
Figure 3.17. Generated new triangles representing the proximal (blue) and distal (orange) parts, respectively.	50
Figure 3.18. New triangles representing callus (black) are introduced between proximal and distal parts.	51
Figure 3.19. Detected intersection points (red dots).....	52
Figure 3.20. Splitting the bone into two pieces	52
Figure 3.21. Callus formation.....	53
Figure 3.22. Superposed images of actual bone and canonical bone in AP view before (left) and after scaling (right). The error in the matching procedure is measured using the distances l_i	55
Figure 4.1. The biomechanical system and the locations of the coordinate systems	59
Figure 4.2. View – position relations of the biomechanical system in AP-view.....	60
Figure 4.3. View – position relations of the biomechanical system in L-view	61
Figure 4.4. Relative angular orientation of the rings and the bone fragments.....	62
Figure 4.5. Angular positions of the joints at the proximal (a) and the distal (b) rings.....	62
Figure 4.6. Sketch (a) and picture (b) of a lambda fixator	67
Figure 4.7. Sketch (a) and picture (b) of a lambda module	67
Figure 4.8. Naming convention of the lambda module	68
Figure 4.9. Eight possible connection types for lambda module.....	69
Figure 4.10. Possible lambda fixator configuration types: (3-6)(a), (6-3)(b), (4-5)(c), (5-4)(d).	71
Figure 5.1. Main window of the GUI for SP-type external fixator	73
Figure 5.2. The <i>Treatment</i> tab	75

Figure 5.3. Menu for actuation functions	76
Figure 5.4. The <i>Checking X-ray Data</i> tab.....	77
Figure 5.5. The <i>Simulation</i> tab	78
Figure 5.6. Main window of the GUI for lambda fixator	79
Figure 6.1. Superposed AP (a) and L (b) images of actual and canonical bone projections of Tibia #1 (Linear mapping)	82
Figure 6.2. Superposed AP (a) and L (b) images of actual and canonical bone projections of Tibia #1 (FFD)	83
Figure 6.3. Superposed AP (a) and L (b) images of actual and canonical bone projections of Femur #2 (Linear mapping)	83
Figure 6.4. Superposed AP (a) and L (b) images of actual and canonical bone projections of Femur #2 (FFD)	84
Figure 6.5. AP (a) and L (b) views of a healthy canonical femur model	86
Figure 6.6. AP (a) and L (b) views of a preprocessed canonical femur model	86
Figure 6.7. Superposed AP (a) and L (b) images of actual and canonical bone projections of Tibia #6 (Fourth order polynomial fit).....	87
Figure 6.8. Superposed AP (a) and L (b) images of actual and canonical bone projections of Tibia #6 (Cubic splines fit)	88
Figure 6.9. Superposed AP (a) and L (b) images of actual and canonical bone projections of Femur #4 (Fourth order polynomial fit).....	88
Figure 6.10. Superposed AP (a) and L (b) images of actual and canonical bone projections of Femur #4 (Cubic splines fit)	89
Figure 7.1. Example 1: Main window of the GUI for SP-type external fixator with input and output values.	92
Figure 7.2. Example 1: Instant displacement values	93
Figure 7.3. Example 1: Relative displacement of fragment ends	93
Figure 7.4. Example 1: Distal fragment end coordinates	94
Figure 7.5. Example 1: Rod lengths	94
Figure 7.6. Example 1: Rod lengths graphs.....	95
Figure 7.7. Example 1: Graph of Jacobian determinant	95
Figure 7.8. Example 1: Rotation matrices	96

Figure 7.9. Example 1: Bone – fixator system in initial (a) and final (b) configurations.....	97
Figure 7.10. Example 1: Evolution of the rod lengths for “Quadratic-1” actuation functions (a) and the corresponding fixation path (b).....	97
Figure 7.11. Example 1: Evolution of the rod lengths for “Sinusoidal-2” actuation functions (a) and the corresponding fixation path (b)	98
Figure 7.12. Example 2: AP (a) and L (b) view images of the bone-fixator system in the initial configuration.....	99
Figure 7.13. Example 2: AP (a) and L (b) view images of the bone-fixator system.....	100
Figure 7.14. Example 2: Axial image of the bone-fixator system.....	100
Figure 7.15. Example 2: Main window of the GUI.....	101
Figure 7.16. Example 2: Bone – fixator system in initial (a) and final (b) configurations.....	102
Figure 7.17. Example 2: AP (a) and L (b) view images of the bone-fixator system in the final configuration.....	103
Figure 7.18. Example 3: Bone – fixator system in the initial (a) and final (b) configurations.....	104
Figure 7.19. Example 4: L view image of the bone-fixator system in the initial configuration	105
Figure 7.20. Example 4: X-ray images of the bone-fixator system are taken in AP (a) and L (b) directions.....	105
Figure 7.21. Example 4: Input and output values shown on the main window of the GUI for lambda fixator.....	106
Figure 7.26. Example 4: Bone – lambda fixator system in the initial (a) and final (b) configurations	107

LIST OF ABBREVIATIONS

AP	: Anterior-Posterior
CORA	: Center of Rotation of Angulation
CT	: Computed Tomography
DRR	: Digitally Reconstructed Radiograph
FFD	: Free-Form Deformation
GUI	: Graphical User Interface
L	: Medial-Lateral
MPPM	: Morpho-Realistic Personalized Parametric Model
MRI	: Magnetic Resonance Imaging
PCA	: Principal Component Analysis
PDM	: Point Distribution Model
SERF	: Simulation Environment of a Robotic Fixator
SP	: Gough – Stewart Platform
SPPM	: Simplified Personalized Parametric Model
TL-HEX	: TrueLok Hexapod System
TSF	: Taylor Spatial Frame
2D	: Two Dimensional
3D	: Three Dimensional

1. INTRODUCTION

As the world population grows steadily, maintaining the quality of life of people requires new approaches. Among other things, computers are widely used to achieve this objective. For example, stock tracking, air traffic control, database management, biometric identification, etc. involve processing of massive amounts data and are all realized via computer applications. Consequently, computer applications are rapidly becoming an integral part of our daily life. Therefore, developing efficient computer applications is one of the main concerns of scientific research today.

As in many other fields, computers are utilized in different branches of medicine for a variety of purposes. Development of medical imaging modalities, computer-assisted surgery and diagnosis, generation of patient-specific tissue models are among the first examples that come to mind regarding computer usage in medicine.

In orthopaedics, conditions such as bone deformities and limb length inequality (i.e. bone shortness) typically require longer treatment durations where the relative position of bone fragments has to be constantly adjusted during the course of the treatment. In the conventional treatment of these pathologies, at least two X-ray images are taken from different directions and a treatment course is prescribed by the clinician according to information deduced from X-ray images. In other words, the clinician has to grasp the anatomy and the geometry of the pathology by observing the X-ray images, and decide how the relative position of bone fragments has to be adjusted during the entire course of the treatment. This is a subjective process whose success depends highly on the experience of the clinician. By taking X-rays periodically, the state of the treatment is checked, and modifications may be proposed if necessary.

This type of procedures are usually accomplished with adjustable ring-shaped external fixators (see Section 1.1). Due to the clinical complexity of these procedures, computer-based support systems have been developed to assist the clinician in planning a proper treatment course.

1.1. Ring Fixators

Fixators are devices that are commonly used in orthopaedic practice. According to the type and localization of the pathology to be treated, fixation may be internal or external. While internal fixators are (almost entirely) buried in the tissue, external fixators reside outside the body such that fixation process can be controlled externally.

The external fixator can be roughly categorized as pin fixators and ring fixators (Figure 1.1). While pin fixators allow bone motion only in axial direction, ring fixators have evolved to provide a higher degree of freedom of relative motion of the attached bone fragments which makes them much more flexible in clinical usage. In a typical ring fixator, bone fragments are fixed to the rings using stressed pins and the rings are connected to each other by a number of rods with adjustable lengths. Consequently, changing the rod lengths changes the relative position of the fragments. Therefore, a controlled motion of bone fragments in space may be achieved by controlling the rod lengths. Due to their flexibility, ring fixators are widely used in the treatment of bone deformities and bone shortness where the relative position of bone fragments has to be constantly adjusted during the course of the treatment.

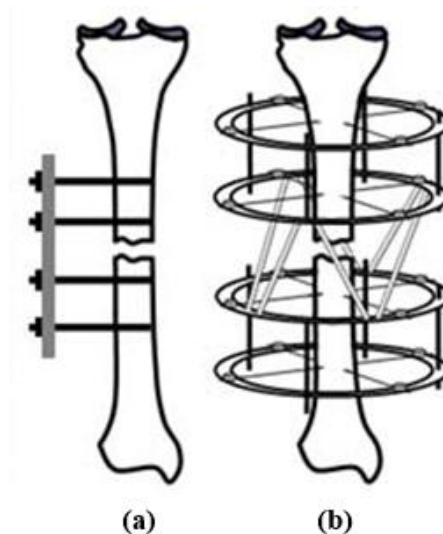


Figure 1.1. Illustrations of pin (a) and ring (b) fixators (adopted from <http://www.bonefixator.com/>).

Although the concept of external fixation goes back to Hippocrates' era (Adams, 1849), the first modern fixator was developed in 1840 by a French surgeon by the name Francois Malgaigne in order to treat a patellar fracture. Since then until 1950s, the external fixation apparatuses and the related procedure have advanced with the contributions of many others, including Rigaud, Parkhill, Lambotte, Codvilla, Hoffman, Vidal, and Charnley (Lawrence A. DiDomenico 2012). In 1950s, Gavril A. Ilizarov has developed the first ring fixator and successfully applied it to a patient with a tibial non-union (Spiegelberg et al., 2010). With its successful application to various cases (i.e. treatment of non-unions, bone lengthening, correction of deformities), this device has gained great reputation in time. Due to the pioneering role of Ilizarov, the device and the related method were named after him. An important recent advancement in external fixation with a ring fixator has been done by Charlie Taylor who developed a highly adjustable circular frame, called *Taylor Spatial Frame*TM (TSF) (Taylor and Taylor, 1999).

1.2. Computer Based Systems

Computer based fixation systems process the data obtained from X-ray images as well as the clinical data determined by the clinician. Rod lengths for each treatment step are calculated by the accompanying software and provided to the patient. Yet, there are several constraints associated with these systems. For example, almost in all systems some input data are collected manually which increases risk for human error. X-rays must be taken parallel to the plane of one of the rings in certain systems such that the ring appears as a straight line in the X-ray image. Also, in many systems connection points of the rods on the rings are fixed spots which diminishes the flexible usage of the fixator.

Another coming shortcoming of these systems is that they do not provide any visualization tools that animate the treatment process, probably, due to the difficulties associated with modeling the bone-fixator system accurately. On the other hand, simulation of external fixation has found limited interest in the literature.

The lack of visual support in the existing systems was one of the motivations for the research performed in this work.

In this section, commercially available computer based systems are briefly described along with their particular shortcomings.

1.2.1. Taylor Spatial Frame

TSF is the predecessor of many commercial and computer-based external fixator systems available today. It is used together with an accompanying software that generates a treatment plan automatically based on the parameters measured on the X-ray images of the treated bone. The system is in the form of Stewart Platform (SP) structure (Stewart, 2009) which consists of two rings connected with six telescopic rods (Figure 1.2). One of the rings is taken as reference and the positioning of the other ring relative to the reference ring is controlled by changing the rod lengths. Length of each rod can be changed independently, hence the structure can move with six degrees of freedom, corresponding to rotations about and translations along the three coordinate axes (Smith&Nephew, 2016).



Figure 1.2. Taylor Spatial Frame

In TSF, the reference ring should appear as a straight line on X-ray images taken in the anterior-posterior (AP) and medial-lateral (L) directions for a proper course of treatment to be calculated by the software. Also, the bone fragment on the

reference ring side should be connected to the frame such that its axis is perpendicular to the ring plane. Another constraint associated with this system is that the rods should be connected to specified holes on the rings.

The visualization tool of the bone-fixator system in TSF is relatively simple. It involves the sketch of a generic bone attached to the frame to provide some guidance to the user (Figure 1.3). Furthermore, animation of the fixation procedure is not available.

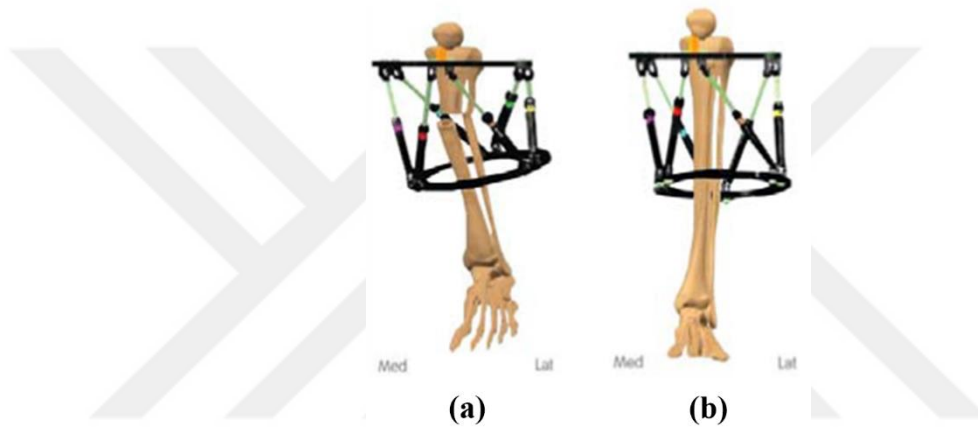


Figure 1.3. Visualization of bone-fixator system in TSF software in the initial (a) and final (b) positions

1.2.2. Smart Correction

Another commercially available computer-assisted fixator system is *Smart Correction* (Smart Correction, 2016) whose frame has a similar physical structure to TSF (Figure 1.4a). The system relaxes some constraints associated with the TSF, while imposing other constraints specific to itself.

In this system, the holes on the rings are consecutively numbered. For a proper calculation of the treatment course, the attachment of the fixator should be done in such a way that the first hole of the proximal ring should face the anterior side of the bone as much as possible. Although not absolutely necessary, placing the proximal ring perpendicular to the axis of the bone is recommended. Consequently, the first and second rods should be attached to the first and second holes, respectively. The other four rods are only allowed to be connected to a certain set of

holes. While neighboring rods are allowed to be attached up to two or three holes apart, they should be ideally attached to adjacent holes. It is also recommended that the rod pairs be equally spaced around the *proximal* (closer to the torso) and *distal* (further from torso) rings. When taking X-ray images, the system requires the X-ray source to be located over the mid-point of the frame. For visual validation, a simple simulation is provided within the software where an animated image of the fixator system is superposed with X-ray images (Figure 1.4b). In this animation, the bone fragments are represented with colored straight lines.

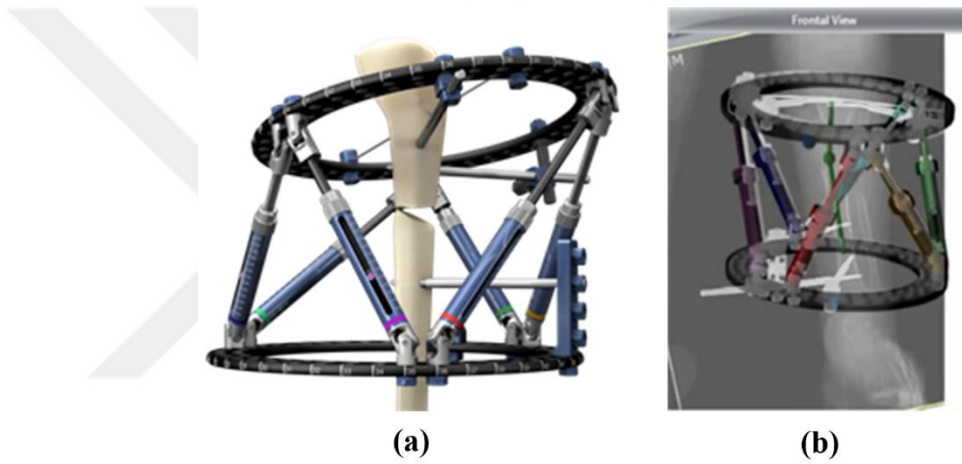


Figure 1.4. Frame of Smart Correction system (a) and the associated visualization tool (b)

1.2.3. TrueLok Hexapod System

TruLok Hexapod System (TL-HEX) is a less commonly used system (Figure 1.5a) (Orthofix, 2016) subject to similar constraints as the other systems. For instance, rods are, again, to be connected to specified holes on the rings and the reference ring is, again, to appear as a straight line segment on X-ray images. A graphical representation tool is present in the accompanying software that lacks any simulation capabilities. The tool only generates images of the system from AP, L and axial views while showing the bone fragments as straight lines as in Smart Correction (Figure 1.5b).

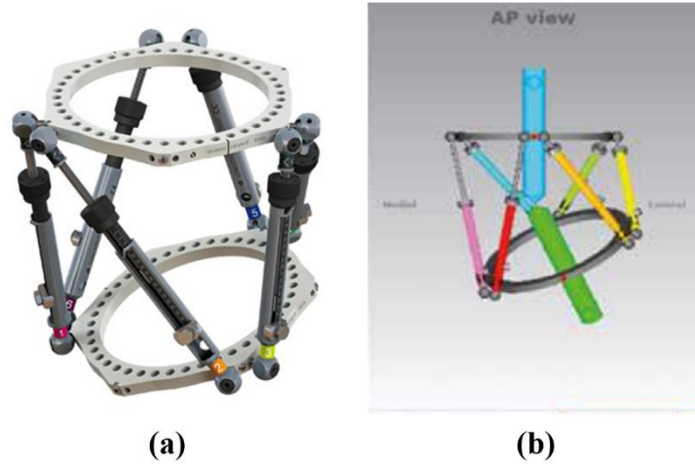


Figure 1.5. Frame of TL-HEX system (a) and the associated visualization tool (b)

1.2.4. Ortho-SUV

With differently designed rods and rings, the structure of Ortho-SUV differs from other three systems (Figure 1.6a) (Solomin et al., 2012). Most constraints associated with other systems do not exist in Ortho-SUV. On the other hand, a marker indicating the X-ray beam center is placed onto the film cassette. This marker should not be blocked by any other radiopaque materials of the frame. Although there are no restrictions regarding the connection points of the rods to the rings, interconnection of cascaded rods is constrained. Also, visual confirmation of the fixation is done by the user by drawing the bone edges on the X-ray images (Figure 1.6b).

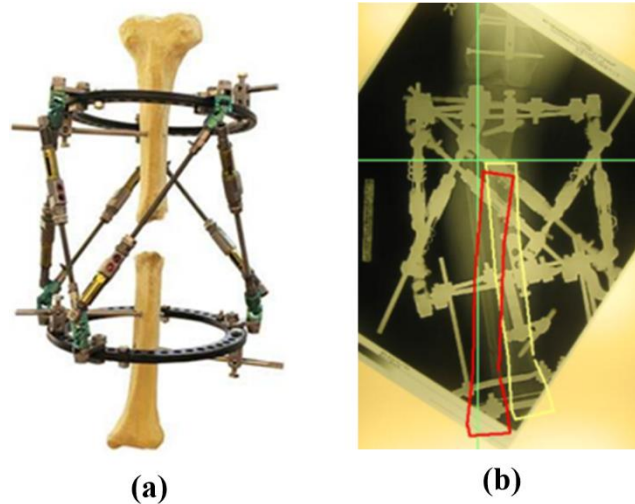


Figure 1.6. Frame of ortho-SUV structure (a) and bone edges drawn on the X-ray image (b)

1.3. Modeling of Patient-Specific Bones and Simulation External Ring Fixators

The main objective of this thesis involves 3D reconstruction of patient-specific bone models and graphical simulation of a ring fixator. In this section, we review the relevant literature.

To our knowledge, only two early studies exist in the literature that deal with the simulation of bone-ring fixator systems. In the first study, a bone surface model is constructed with two X-ray images taken in perpendicular directions using a back-projection algorithm (Caponetti and Fanelli, 1993). The study involves the combination of the bone model with the model of an Ilizarov apparatus, and an interactive GUI, although the mathematical and computing details of the work remains largely unexplained. In the second study, a simulation software called “Simulation Environment of a Robotic Fixator” (SERF) is developed for treatment visualization (Viceconti et al., 1993). Both studies has produced relatively primitive models with limited user-interaction capability.

One way of 3D reconstruction is using a number of computed tomography (CT) or magnetic resonance images (MRI) which provide cross-sectional views of the anatomical structure. Processing and segmenting a sequence of these images allow them to be combined to a 3D surface model (e.g. Marching Cubes Algorithm)

(Lorensen and Cline, 1987) and many such approaches based on cross-sectional images have been reported in the literature (Viceconti et al., 1999; Birkfellner et al., 2003; Viceconti et al., 2004; Yosibash et al., 2007). With sufficiently large number of slice images gathered, this approach is useful for generating a patient-specific bone model accurately. But these methods have certain drawbacks such as the patient being exposed to a relatively high dosage of radiation (in case of CT) and high financial cost of the imaging. As a result, for orthopaedic treatments requiring fixation, X-ray is the preferred imaging method and slice imaging using CT or MR is usually not performed. It should be noted that 3D reconstruction from slice data is computationally more demanding as well. There are commercial software packages, such as MIMICS (MIMICS, 2016), available to accomplish this task whose performance is dependent on the quality of the input data.

Researchers approach the problem of 3D reconstruction of femur or tibia in numerous ways. Parametric models (Baudoin et al., 2008; Chaibi et al., 2012; Quijano et al., 2013), statistical shape models (Fleute and Lavallee, 1999; Rajamani et al., 2007; Baka et al., 2011; Zhu and Li, 2011; Blanc et al., 2012), and generic models of the bone surface (Laporte et al., 2003; Gamage et al., 2011; Koh et al., 2011; Karade and Ravi, 2015) are used as input and the model is deformed according to various measurements obtained from either X-ray or CT images.

There are many parameters affecting the shape of the resulting bone model and the computational burden of the proposed methods are usually high. Therefore, purpose-specific reconstruction methods rather than general ones may be more appropriate for practical purposes. For example, the works focusing on reconstruction of proximal femur (Dong and Zheng, 2006; Zheng, 2006; Zheng et al., 2006; Zheng and Schumann, 2008; Zheng and Schumann, 2009; Galibarov et al., 2010) are important for medical applications such as total hip replacement and implant manufacturing. Therefore, emphasis is given to accurately represent the head and neck of the femur and other relevant parts. Similarly, distal femur reconstruction bears importance for navigation surgeries and *in vivo* knee kinematic studies (Zhu and Li, 2011).

In orthopaedic applications, bone is typically visualized through X-ray images and an alternative way for 3D model reconstruction is to utilize the 2D biplanar X-ray images along with a statistical model of the anatomical structure to be reconstructed. This is a challenging task, because, to generate a statistical model, a large number of bones (either cadaveric (Chaibi et al., 2012) or *in vivo* (Zheng and Schumann, 2009)) has to be scanned and geometrically analyzed. Furthermore, accuracy of these methods cannot be always properly evaluated due to limited number of available X-ray images for testing.

In numerous works of Zheng et. al., various methods for 3D bone model reconstruction are proposed. These methods include image-to-model correspondence building using edge pixels detected from the 2D images and apparent contours extracted from the 3D model (Zheng, 2006), and a non-rigid 2D point matching process which iteratively uses a symmetric injective nearest neighbor mapping operator (Zheng and Nolte, 2006; Zheng et al., 2006; Zheng and Schumann, 2008). As an extension to one of their previous methods, whose convergence relies on a proper initialization, a fully automated initialization method is proposed (Dong and Zheng, 2006). This approach uses a particle filter based inference algorithm to obtain the geometric parameters of proximal femur from fluoroscopic images. Using point distribution models (PDM), Baka et. al proposes a method for pose estimation and shape reconstruction of 3D bone surfaces from at least two X-ray images. The method requires a single manual initialization and optimizes the shape and pose of the model based on the Canny edge map of the X-ray images (Baka et al., 2011).

Galibarov et. al. proposes a fully automated three step method for 3D reconstruction of proximal femur surfaces. It begins with contour extraction from X-ray images using an algorithm which utilizes the bone anatomy for improved edge detection. Then, the model closest to bone at hand is selected from a generic model library. Finally, this model is warped to fit the extracted contour of the femur (Galibarov et al., 2010). Laporte et. al. have developed a method called non-stereo corresponding contour (NSCC) which is based on contours identification from biplanar radiographs. Anatomical regions are defined on a generic model and extracted 2D contours are associated with these regions. (Laporte et al., 2003).

Cresson et. al. report a method to estimate a 3D personalized shape of bone models obtained from two planar X-ray images, using a deformation method based on moving least squares optimization (Cresson et al., 2010). Gamage et. al. propose a similar method in which the 2D X-ray images are processed to detect edges of femurs. Then, a non-rigid robust registration is performed between these edges and contours of a generic model. These point correspondences are used to determine the deformations to be applied to the 3D generic model (Gamage et al., 2011). The method proposed by Mahfouz et. al. utilizes a statistical atlas of bone, based on principal component analysis (PCA) and biplanar X-ray images, to fully reconstruct a patient-specific 3D bone model. In this method, automated morphometric and surgical axis measurements are performed on the biplanar images to estimate the shape and the size of the bone. Then, digitally reconstructed radiographs (DRRs) of the bone atlas are generated for different shape parameters, translations and rotations to determine the best bone model. (Mahfouz et al., 2006). In the work of Gunay et. al., hierarchical free-form deformation is used to scale and deform a 3D template model of a clinically normal bone. Subdividing the control lattice of the free-form deformation is divided into subregions and further deformation is applied in the subregions where 2D error is high. The procedure is continued until the projection of the deformed shape matches with the X-ray images (Gunay et al., 2007). In order to address the issue of generating femur models with severe rotational deformities, Koh et. al. proposes a method that uses two X-ray and three CT images, which are used to identify relevant anatomic axes at different levels of the bone. In this method, a 3D template of the femur model is deformed with free-form deformation (FFD) algorithm until the projection of the deformed model matches the given X-ray images and its cross-sectional contours match the given CT slices. They report that introducing CT images improves the reconstruction accuracy (Koh et al., 2011).

In the method proposed by Karade et. al., Kohonen self-organizing maps are used to find 2D-3D correspondences between X-ray images and the template model and deformation is carried out with Laplacian surface deformation (Karade and Ravi, 2015). In the work of Chaibi et. al., 3D models are reconstructed based on parametric and statistical data obtained from biplanar X-ray images. Their method consists of

two main steps: The first one involves generating a 3D simplified personalized parametric model (SPPM). This model is then used for the calculation of the most important parameters, which form the input for the next step, namely generating a full 3D morpho-realistic personalized parametric model (MPPM). Generation of this final model includes projecting the MPPM on radiographs and iteratively adjusting the model according to the incoherence between MPPM landmarks and bone contours (Chaibi et al., 2012). This methods has been later improved by Quijano et. al. (Quijano et al., 2013).

Some of these bone modeling techniques proposed in the previous work may be suitable to be used in the simulation of an orthopaedic external fixation process. However, most of them contain details too irrelevant for the purpose of fixation. and come with a high computational cost. Also, non-automatic reconstruction procedures will definitely burden the clinician with an extra non-clinical task of data gathering/handling. Therefore, the need for a specific method to generate patient-specific 3D bone models, to be used merely in simulation of orthopaedic treatment routines (e.g. deformity correction, fracture fixation), is obvious.

1.4. Objective and Contribution of the Thesis

As mentioned earlier, there are numerous constraints associated with the computer-based systems. Moreover, these systems all lack a proper visualization tool which would allow the user to observe the proposed treatment plan in a realistic manner.

In a simulation environment, a realistic depiction of the system, consist of the patient-specific bone and the fixator frame, is important. So, in this thesis, a computationally efficient method for generation of patient-specific bone models is proposed. This method uses the data drawn from X-ray images by othopaedists when planning a surgery and does not require collection of large amounts of statistical data. A graphical user interface (GUI) featuring a simulation tool for bone-fixator systems is also developed. Generated patient-specific 3D bone models are to be within this simulation environment together with an accurate model of the fixator

frame. The GUI implements a mathematical theory that generates a treatment plan automatically using the relevant input parameters. The simulation tool visualizes this treatment plan (Figure 1.7).

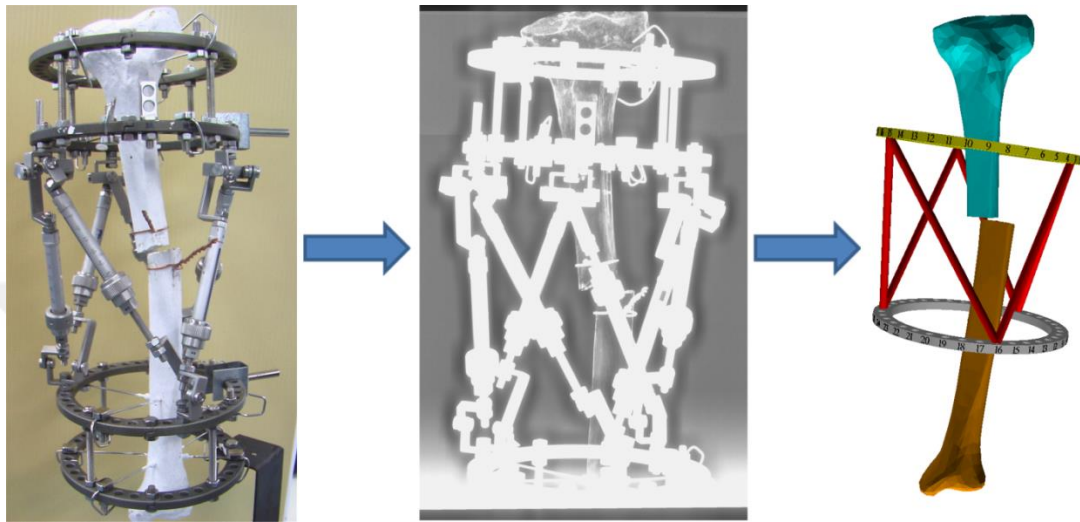


Figure 1.7. External fixation system (left), X-ray image of the system taken from AP view (center) and generated bone-fixator system according to the X-ray image (right).

The simulation tool associated with the developed GUI allows the user to visualize the entire fixation process in computer environment. This original feature of the GUI is not present in the currently available computer-based systems. Patient-specific bone models are utilized in the simulation and hence an enhanced visualization is achieved. Generation of patient-specific bone models is accomplished by a novel method. When the GUI is used for deformity correction or bone lengthening purpose, the formation of the healing bone tissue (called *callus*) between the bone fragments is simulated and visualized as well.

1.5. Outline of the Thesis

This thesis involves realistic simulation of an orthopaedic process. Therefore, in Chapter 2, some background information for the methodologies used in the thesis is presented.

Methods, developed for generating patient-specific bone models from X-ray images of undeformed or deformed bones, are described in Chapter 3.

The obtained patient-specific bone models are appropriately placed into the simulation environment of the two different external ring fixator systems, which are described in Chapter 4. Brief information about their associated inverse and direct kinematic calculations are given.

The developed GUIs implement the kinematic theories associated with these fixator systems and described in detail in Chapter 5. These GUIs perform the mathematical computations, output detailed computational results and contain the simulation tool.

In Chapter 6, the performance of patient-specific bone model generation methods are evaluated according to the defined error measure.

The capabilities of the GUIs are illustrated using four different examples in Chapter 7.

Concluding remarks are presented in Chapter 8

2. BACKGROUND

The bone reconstruction method we propose in this work utilizes the geometrical parameters that the clinicians define/measure on the X-ray images as part of surgery planning. Hence, some basic information about surgery planning is provided here as medical background.

In this work, a method called Free Form Deformation is used to deform models of healthy bones to patient-specific pathological bones, as explained in detail in Chapter 3. This method is introduced also in the current chapter.

2.1. CORA Method and Surgery Planning

One common application in orthopaedic practice is correction of deformities of long bones (e.g. femur or tibia). In order to identify the deformity and make accurate surgery planning, a point called *center of rotation of angulation* (CORA) of is located on the X-ray images. For this purpose, either anatomic or mechanical axis of the treated bone is appropriately drawn through both proximal and distal end of the bone on the X-ray images. In a bone with deformity, these two lines will intersect to form an angle. CORA is defined as the point where the lines representing the proximal and distal axes intersect (Paley, 2005).

The force acting on a joint is typically a distributed force. Mechanically, this distributed force can be considered as a concentrated single force with the same magnitude, acting at a point called the *joint center*. The mechanical axis of a bone is defined as the straight line connecting the joint centers of the proximal and distal side. The anatomic axis is the line that approximately bisects the *diaphysis*, (i.e. the bone shaft), called the *mid-diaphyseal* line (Paley, 2005). In tibia, in the frontal plane corresponding to AP view, the mechanical and anatomic axes are parallel and very close to each other. But in femur, they deviate significantly from each other as the load-bearing mechanical axis passes through the center of femoral head (Figure 2.1). A similar procedure is followed when determining the anatomic and mechanical axes in sagittal plane. The deviation between mechanical and anatomic axes brings about

some complexity to surgery planning. Therefore, it is more convenient to implement computer algorithms when the CORA is defined using anatomic axes. Consequently, surgery planning based only on the anatomic axes has been considered in this study.

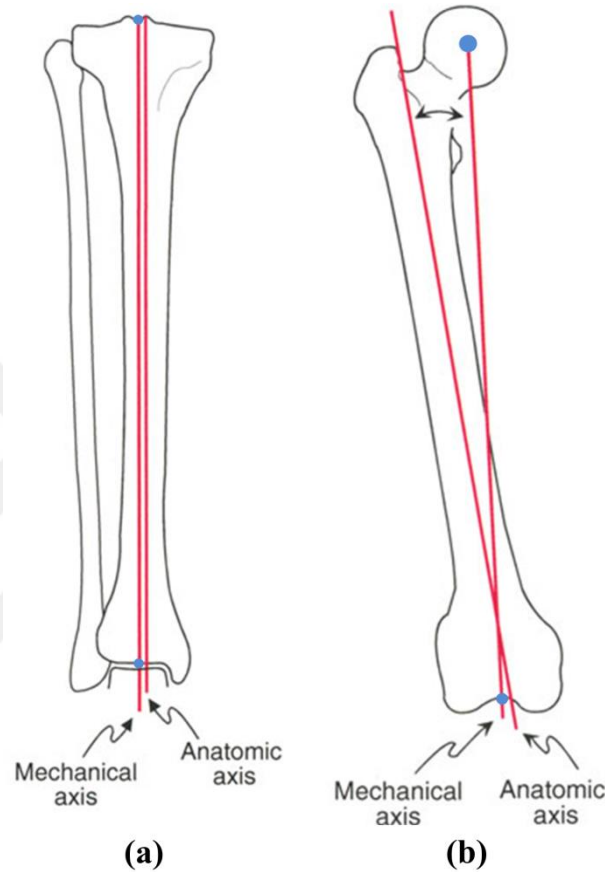


Figure 2.1. Frontal plane anatomic and mechanical axes for tibia (a) and femur (b). Blue dots represent the joint center points.

It is possible that a deformity exists in both planes at the same time or only on one plane. In case of a single-plane deformity, the CORA can be located only on the plane of deformity. From the other view, the bone would look healthy.

In each of the frontal plane and the sagittal plane, the procedure for determining the position of CORA in tibia and femur on the X-rays involves a sequence of steps. Regardless of the bone type and the plane, the procedure follows identical steps which may be summarized as follows:

- 1) Draw the mid-diaphyseal lines from the proximal and distal side of the bone on the X-rays. Each mid-diaphyseal line segment is the anatomic axis line for that segment of bone (Figure 2.2).
- 2) Check whether the angle between proximal (and distal) mid-diaphyseal lines and related *joint orientation line* (i.e. line that defines the loadbearing plateau of the joint) is normal (Figure 2.2). In proximal tibia, medial proximal tibial angle (MPTA); in distal tibia, lateral distal tibial angle (LDTA) are measured. Similarly, in case of femur, in proximal femur, medial proximal femoral angle (MPFA) and in distal femur, anatomic lateral distal femoral angle (aLDFA) are measured. The accepted normal range for these angles are 85° - 90° for MPTA, 86° - 92° for LDTA, 80° - 89° for MPFA and 79° - 83° for aLDFA.
- 3) If both distal and proximal angles of the bone in consideration are normal, there is only one CORA at the intersection point of the lines drawn in Step 1 (Figure 2.3b). Otherwise, the surgeon defines multiple CORAS which is beyond the scope of this work. If the CORA found in this step is not at the obvious apex (or outside the bone), there is either a second CORA or a translation deformity, which are, again, not covered in this work.

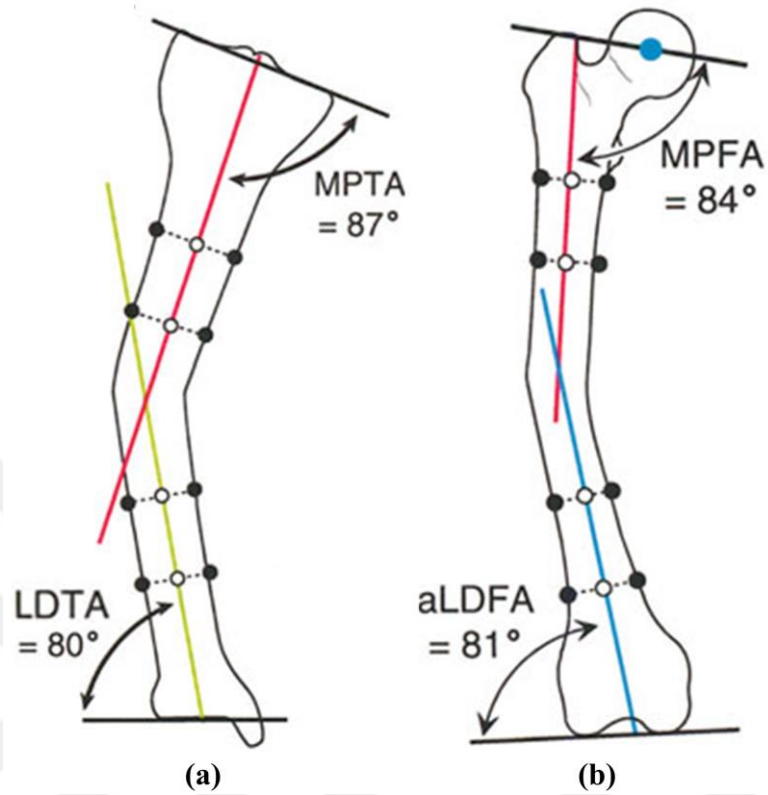


Figure 2.2. Frontal plane anatomic axis planning on tibia. The mid-diaphyseal line for proximal and distal parts are shown in red and yellow, respectively (a). The same planning performed on femur. The mid-diaphyseal line for proximal and distal parts are shown in red and blue, respectively (b). (from (Paley, 2005))

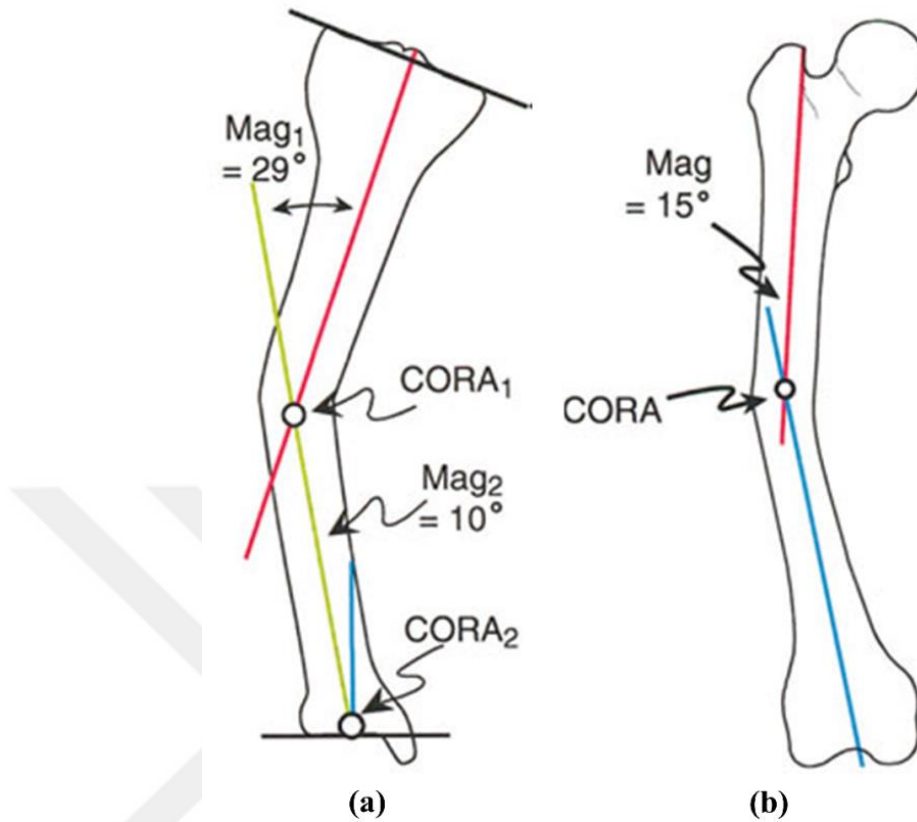


Figure 2.3. Frontal plane anatomic axis planning in tibia. Extra anatomic axis line referenced to the ankle joint orientation line is shown in blue (a) Similar planning performed on femur (b). (from (Paley, 2005))

This planning scheme requires that only mid-diaphyseal lines and joint orientation lines be drawn on X-ray images and the CORA (C in Figure 2.4) be located. For our purposes, the intersection points of mid-diaphyseal lines with joint orientation lines (P and D in Figure 2.4) also have to be marked. Consequently, a method to generate 3D model of the bone using only the information conveyed by these lines is certainly more practical as the clinician does not have to make any other measurements on X-rays.

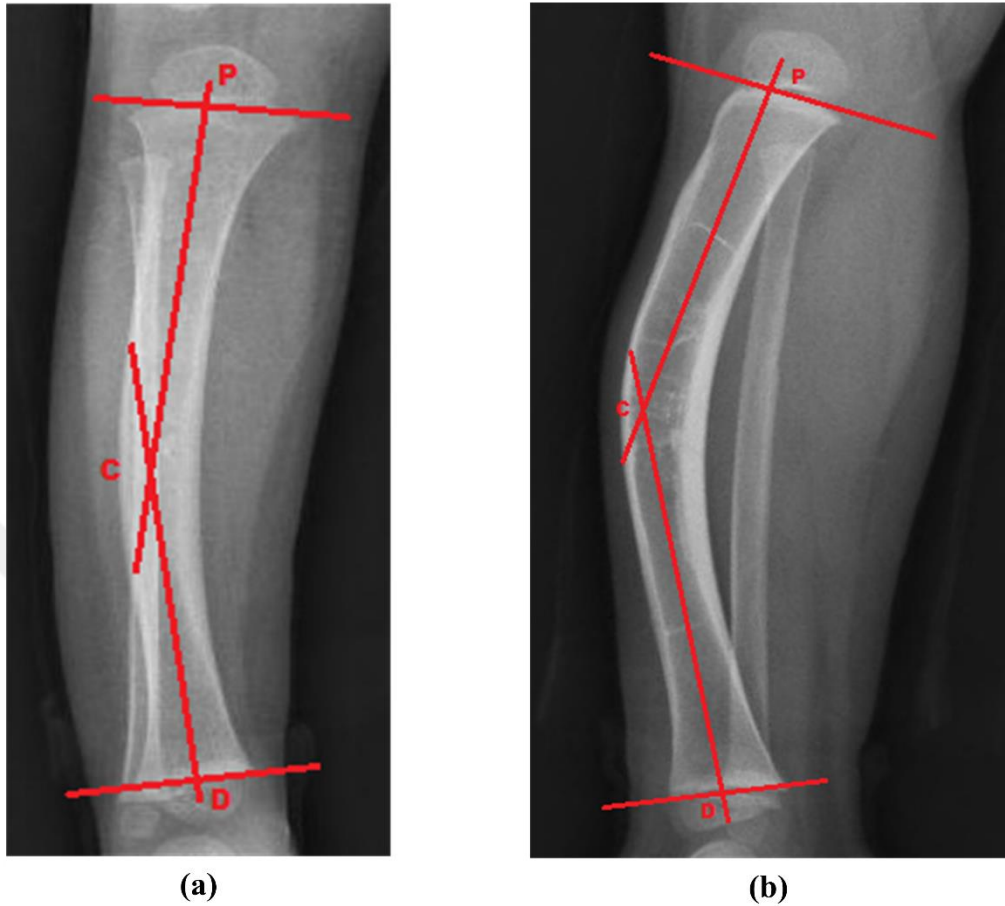


Figure 2.4. Tibial anatomic axis planning on AP (a) and L (b) views of X-ray images.

Since surgery planning relies mostly on the location of the CORA in deformity correction procedures, it is important to locate the CORA accurately on the X-rays. Once the CORA is located, an *osteotomy* is performed where the bone is divided into two fragments at the level of the CORA with possible removal of some bone material.

Angulation-only deformities are treated by defining a rotation axis at the same level with CORA and perpendicular to the image plane. Therefore, rotation around this axis produces opening and/or closing wedge angulation at the osteotomy site (Figure 2.5).

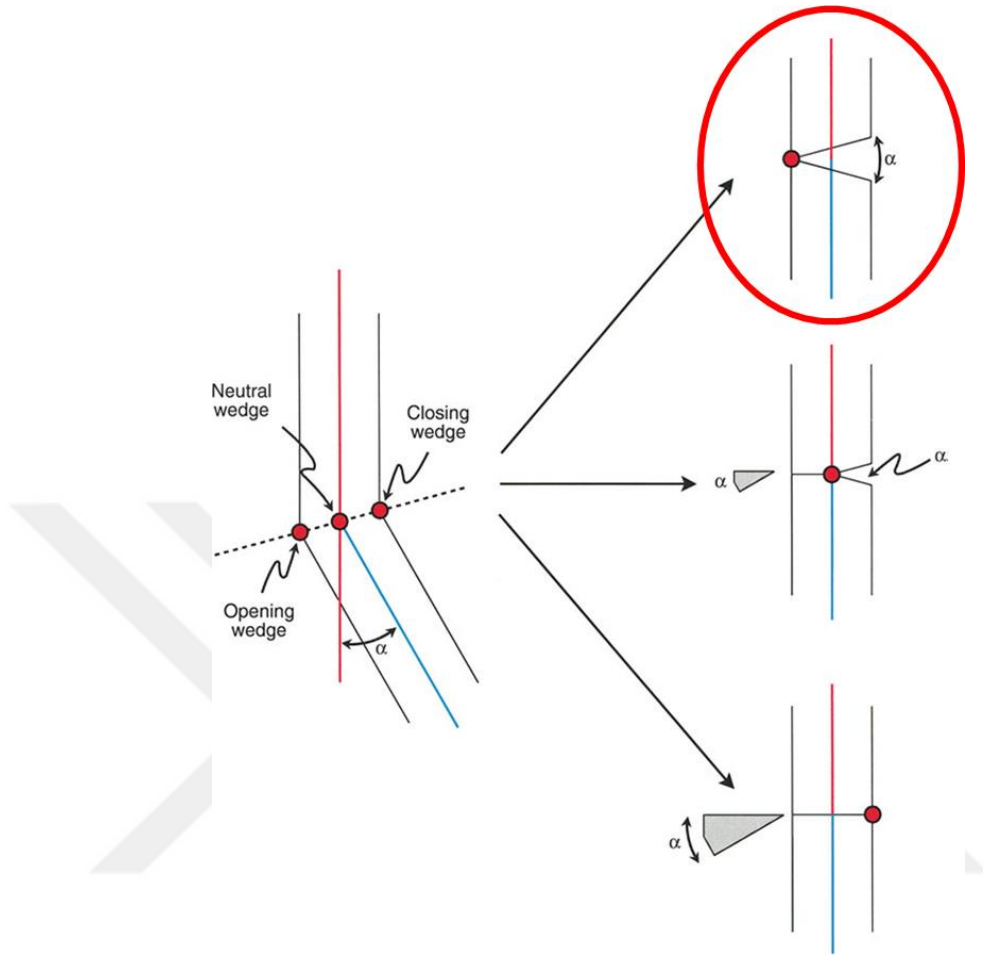


Figure 2.5. Different osteotomy options in bone deformity correction: (Top to bottom) opening, neutral and closing wedge angulations. The red dots indicate the rotation axes (Paley, 2005). The circled case, where no bone removal is performed, is the preferred case and the algorithm described in this work is based on this case.

Deformity correction can be realized in different ways by the surgeon by choosing the location of rotation axis and osteotomy site differently. This choice is based on the physical/clinical status of the patient, treatment objectives and the experience of the clinician. However, orthopaedists usually avoid removal of bone material in correction surgery. Therefore, the simulation tool has been developed to cover correction of angulation-only deformities in open wedge sense (shown in red circle in Figure 2.5).

2.2. Free Form Deformation

To obtain deformed bone models we have used an algorithm called *Free Form Deformation* (FFD). The concept of FFD is often intuitively explained with a plastic object analogy where the object(s) to be deformed are embedded in a transparent plastic block. A change in shape of this plastic block will change the shape of the object as well. So, the control over the plastic block allows the control of the embedded objects indirectly (Figure 2.6).

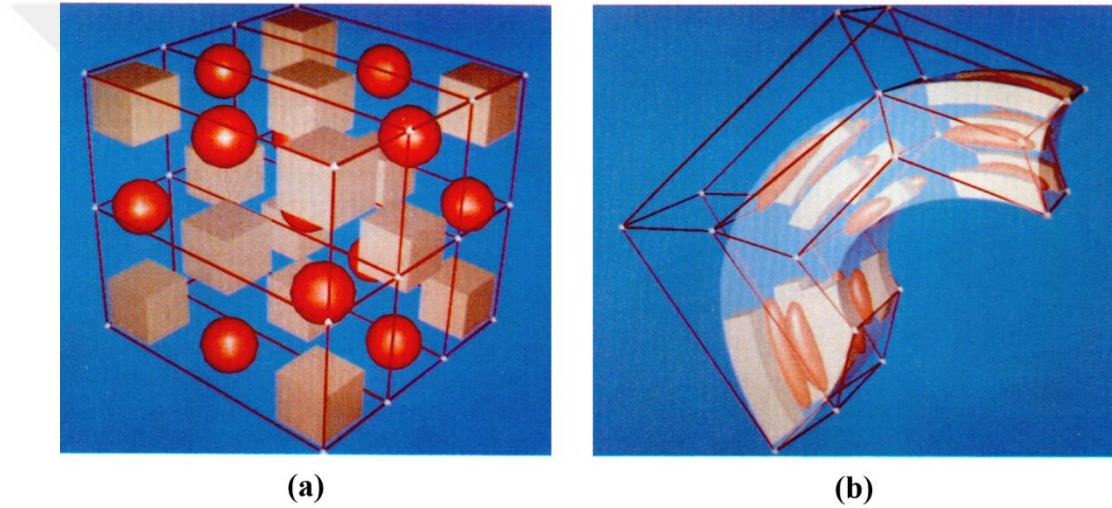


Figure 2.6. A 3D object before (a) and after (b) FFD. (from (Sederberg and Parry, 1986))

Mathematically, FFD is a mapping from R^3 to R^3 through a trivariate tensor product of Bernstein polynomial (Sederberg and Parry, 1986). A parametric (s, t, u) coordinate system is introduced such that $0 \leq s \leq 1$, $0 \leq t \leq 1$, $0 \leq u \leq 1$ to enclose the entire region to be deformed. Then, $l+1, m+1, n+1$ ($l, m, n \in Z^+$) control points along s, t, u directions, respectively, are defined to form a grid enclosing the region. (The number of control points in each direction is determined by the user.) The deformed position of an arbitrary point $\tilde{Q}(s, t, u)$, in the object is calculated by

$$\tilde{Q}(x, y, z) = \sum_{k=0}^n \sum_{j=0}^m \sum_{i=0}^l B_k^n(s) B_j^m(t) B_i^l(u) \tilde{P}_{ijk} \quad (2.1)$$

where $\tilde{P}_{ijk}$ is an array containing the x,y,z coordinates of the control points, $\tilde{Q}(x,y,z)$ is a vector containing x,y,z coordinates of the deformed point and $B_k^n(s)$, $B_j^m(t)$, $B_i^l(u)$ are Bernstein polynomials of order n, m, l , respectively, defined according to

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, 1, \dots, n \quad (2.2)$$

FFD produces a deformed object with a certain level of smoothness, which is an important requirement in this work to obtain proper deformed bone models. An example, where a cylindrical object is bent using FFD, is provided in Figure 2.7.

In the developed bone reconstruction algorithm, FFD is utilized for two different purposes: Scaling the canonical bone model in transverse direction (thickness matching) and deforming the canonical bone model.

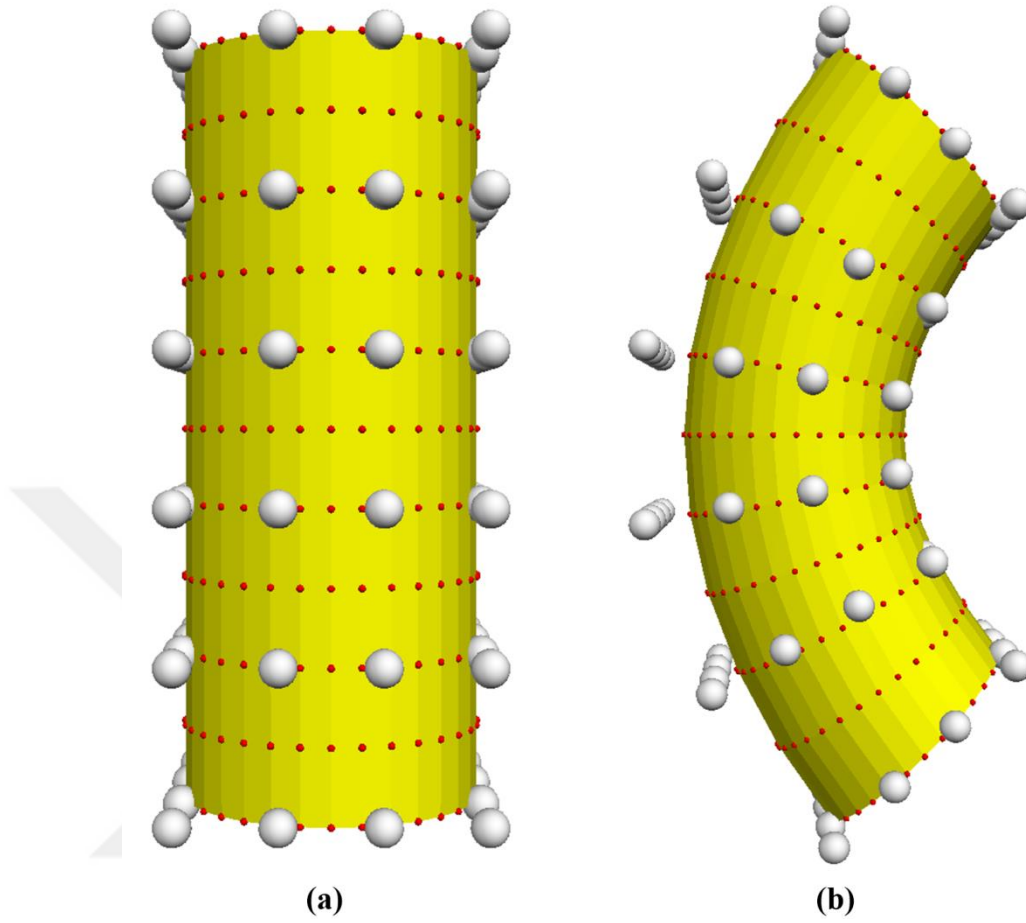


Figure 2.7. A cylinder (a) bent with FFD (b) by changing the locations of control points (white spheres)

3. GENERATION OF PATIENT-SPECIFIC BONE MODELS

3.1. Model Creation

Computer-assisted systems have been commonly used in external fixation procedures. In the existing systems, visualization tools either display a simplified picture of the fixation frame where bone fragments are represented with either cylinders (Orthofix, 2016) or simplified bone images (Smith&Nephew, 2016); or superpose a sketch of the fixator on the bone X-ray image (Smart Correction, 2016) (see Section 1.2). To our knowledge, visualization tools, where fixation procedure is animated, are not present in the existing systems.

A proper 3D visualization tool that will accompany a computer-assisted fixation system should fulfill certain conditions. In particular, it should:

- 1) reflect the bone-fixator system realistically.
- 2) run reasonably fast.
- 3) be easy to handle by the user.

For a proper depiction of the bone-fixator system, the treated bone has to be modelled realistically. Software tools exist to create a solid model of the tissue of interest from radiological slice data (MRI or CT), yet, in orthopaedic treatment, slice data for the entire bone is rarely available. Moreover, model creation from slice data is not always a straightforward process (Viceconti et al., 1999). Simulation involving such a bone model has to handle a large amount of data and is likely to run slow within the GUI. As a solution to this problem, we propose to create a 3D model of the bone by manipulating an available model of generic bone model, referred to as *canonical model* throughout this thesis. Clearly, the shape and size of the treated bone change from one patient to the other. In the proposed approach, the canonical bone model is exploited to match the shape and size of the treated bone. In this case, the resulting model can be easily handled within the visualization tool as will be explained in the upcoming sections.

The canonical bone models can be obtained through different means. For example, the tibia model used in this study originates from the MR scan of a cadaver. The resulting set of slice images is combined in CATIA software (Vélizy-Villacoublay, France) into a surface model. The femur model, on the other hand, has been created by scanning a cadaveric bone via David 3D Scanner system (Bonn, Germany). These surface models basically consist of a set of vertices with triangular connectivity.

In both cases, the resulting model files are large, containing a total number of vertices in the order of 10000's. For practical purposes, these large models are downscaled to smaller ones using MeshLab software (MeshLab, 2016). The typical downscaled model contains a total number of vertices in the order of 1000's and can be easily handled within the GUI (Figure 3.1).

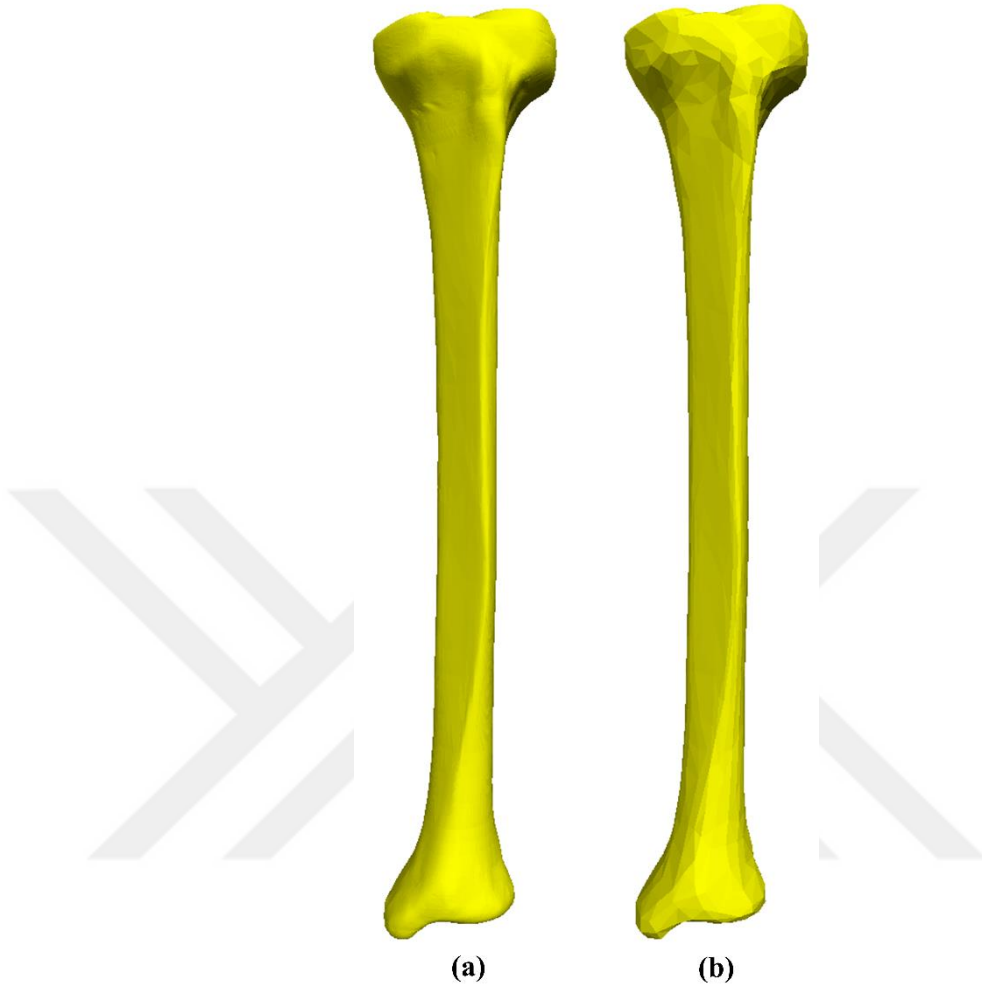


Figure 3.1 Original (a) and canonical (b) tibia models

3.2. Model Manipulation for Undeformed Bones

The canonical bone model and the bone image at hand do not match in size and orientation, in general. They are brought to the same size and orientation by manipulating the canonical bone model. For this purpose, two different methods based on linear mapping and FFD have been proposed. These methods are detailed below.

3.2.1. Model Manipulation for Undeformed Bones with Linear Mapping

Linear mapping approach consists of four steps that are applied separately to AP and L views: (i) Generating the projection of the canonical bone model. (ii) Detection of bone edges on the X-ray images. (iii) Determining the relative orientation of the bone image in the X-ray and canonical bone model. (iv) Scaling and matching of the canonical model to the bone image.

3.2.1.1. Projection of Canonical Model in Anatomical Directions

In the first step, the orthographic projection of the canonical model is performed in the AP and L directions. Since the canonical model is defined as a collection of vertices, the projection images also consists of vertices (Figure 3.2).

The projection images serve as input to the steps (iii) and (iv).

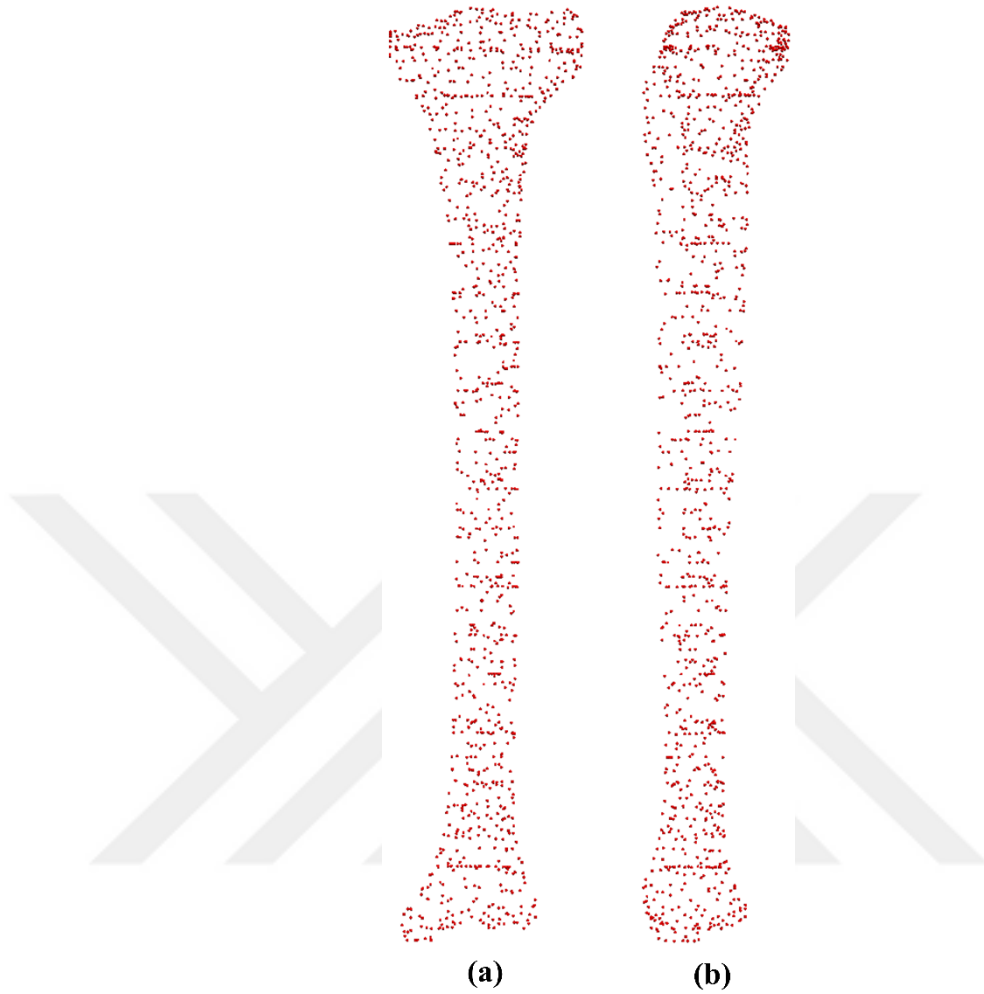


Figure 3.2. The projection of canonical model in AP (a) and L (b) directions with model vertices shown as red dots

3.2.1.2. Detection of Bone Edges on the X-ray Images

The second step involves the extraction of the bone profile from the X-ray images. This is accomplished on the AP and L images through canny edge detection algorithm (Canny, 1986). Some false edges detected by the algorithm are removed from the image manually (Figures 3.3 and 3.4).

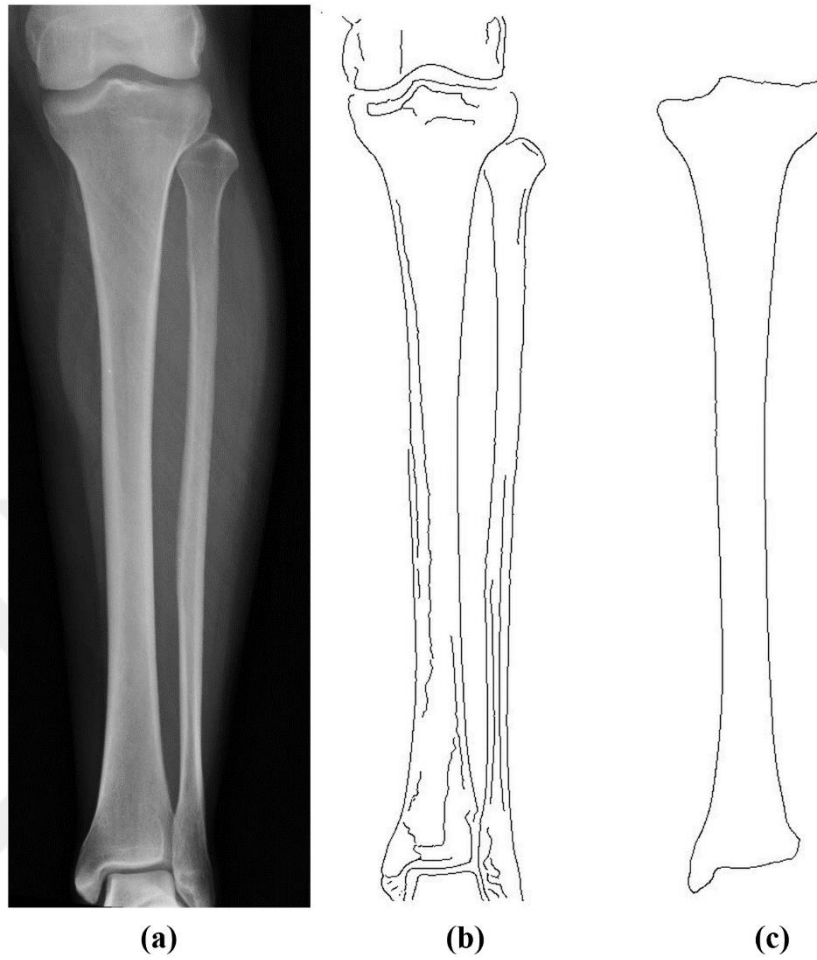


Figure 3.3. AP X-ray of tibia (a). Output of canny edge detection (b). Edges after manual intervention (c).

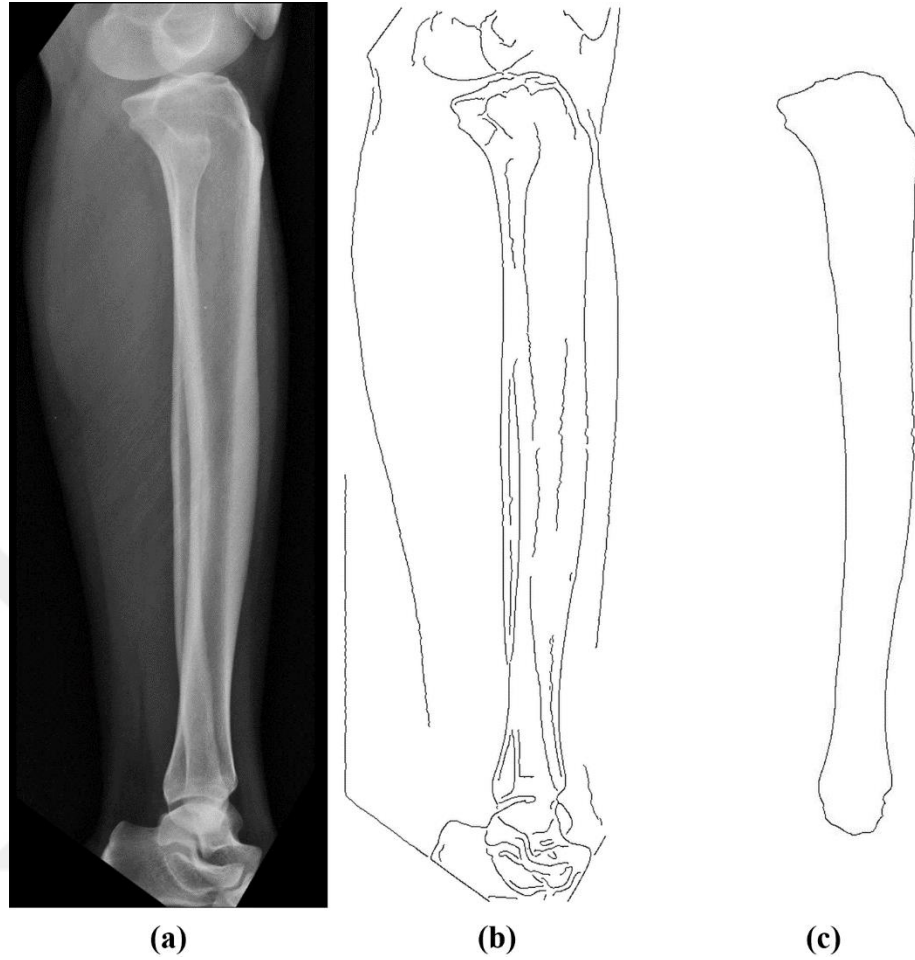


Figure 3.4. Lateral (L) X-ray of tibia (a). Output of canny edge detection (b). Edges after manual intervention (c).

3.2.1.3. Determining the Relative Orientation of the Bone Image and Canonical Model

In the third step, the bone image on the X-ray and the projected images of the canonical bone are brought to the same orientation.

The longitudinal axis of bone may not be aligned with the frame of the X-ray image. In clinical practice, this is sometimes done on purpose to be able to fit the entire bone to the X-ray frame. To bring the canonical bone to the same orientation with the bone image, we first determine a coordinate origin C on both images. With respect to image coordinate system, the y -coordinate of C is the average of the y -coordinates of the top-most and bottom-most points on the bones. If a horizontal line

is drawn at that level, it will intersect the bone image. The x-coordinate of middle point in the intersection segment with the bone is taken as the x-coordinate of C (Figure 3.5).

The points P_1 and P_2 are defined as the middle points of the bone cross-sections that are equidistant (d_1) from the top-most (P_{top}) and bottom-most (P_{bottom}) points of the bone profile. The line connecting these two points indicates the local y-axis in the image (Figure 3.5). The vector from P_1 to P_2 is easily calculated as:

$$\vec{V}_b = (P_{2x} - P_{1x})\vec{i} + (P_{2y} - P_{1y})\vec{j} \quad (3.1)$$

An almost identical procedure is followed on the projected images of the canonical bone model to locate the points R_1 and R_2 (Figure 3.5). To locate R_1 and R_2 , the bone cross-sections are defined equidistant (d_2) from the top-most vertex (R_{top}) and the bottom-most vertex (R_{bottom}) of the canonical model profile. Since the canonical model is a collection of vertices, a strip of thickness ε is defined about the top and bottom cross-sectional lines. The average of the x-coordinates of right-most and left-most vertices in these strips is taken as the x-coordinates of R_1 and R_2 .

The line connecting R_1 and R_2 indicates the direction of the local y-axis in the canonical bone image (Figure 3.5). The vector from R_1 to R_2 can be written as:

$$\vec{V}_c = (R_{2x} - R_{1x})\vec{i} + (R_{2y} - R_{1y})\vec{j} \quad (3.2)$$

The points, denoted as C in Figure 3.5, bisect the $\vec{V}_b$ and $\vec{V}_c$, and are defined as the center for both actual and canonical bone images.

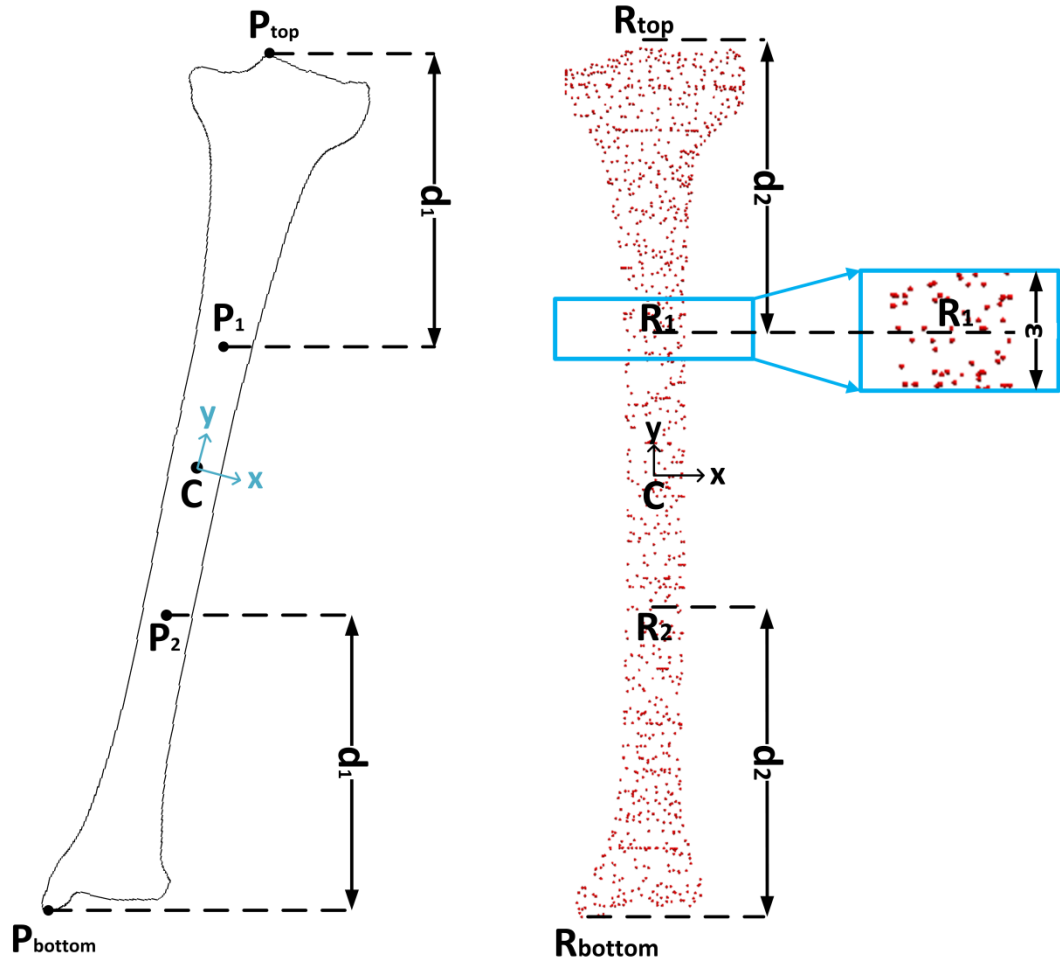


Figure 3.5. Illustration of points P_1 and P_2 on bone edges (left) and R_1 and R_2 on the canonical model profile (right)

The images are superposed by overlapping the center points C . The angle between the vectors $\vec{V}_b$ and $\vec{V}_c$ is the relative orientation of actual and canonical bone images, and can be calculated using the dot product as

$$\theta = \cos^{-1} \left(\frac{\vec{V}_b \cdot \vec{V}_c}{\|\vec{V}_b\| \|\vec{V}_c\|} \right) \quad (3.3)$$

The way the vectors are defined will ensure an orientation angle θ between -90° and 90° , although, in practice, the angle will usually have a small magnitude.

The actual bone image is rotated by the angle θ about the common center C and brought to the same orientation as the canonical bone image. A correction is also applied to the θ value in the upcoming step.

3.2.1.4. Model Scaling and Matching

Initially, the actual bone and the corresponding canonical bone have different scales (Figure 3.6). The canonical bone model is scaled independently in the axial and transverse directions to obtain a model that represents the actual bone. The related scaling factor is calculated from the images of the canonical bone and actual bone in both AP and L views.



Figure 3.6. Actual bone and the corresponding canonical bone

Scaling in axial direction is performed by mapping the extreme y-coordinates of canonical bone image $(R_{top_y}, R_{bottom_y})$ to those of the actual bone image

$(P_{top_y}, P_{bottom_y})$. Accordingly, all vertices of canonical model are mapped linearly.

In that sense, the canonical model is “stretched” in the axial direction.

The variation of the bone thickness along the bone axis will be in general different for the actual bone and the canonical bone. Hence, a variable scaling factor is applied in the transverse direction across the bone axis. We have determined three different transverse scaling factors at three different transverse cross-sections of the bone images. Given the thickness values of the actual bone (a_1, a_c, a_2 measured at P_1, C, P_2 , respectively), and canonical bone (b_1, b_c, b_2 measured at R_1, C, R_2 , respectively) the corresponding scaling factors are (Figure 3.7):

$$r_1 = \frac{b_1}{a_1}, \quad r_c = \frac{b_c}{a_c}, \quad r_2 = \frac{b_2}{a_2} \quad (3.4)$$

The scaling factor r is assumed to change bilinearly in axial direction; i.e. it has two independent linear forms for proximal and distal halves of the canonical model, given as:

$$r(y) = \begin{cases} \frac{r_1 - r_c}{R_{1y}} y + r_c, & \text{for proximal half } (y > 0) \\ \frac{r_2 - r_c}{R_{2y}} y + r_c, & \text{for distal half } (y \leq 0) \end{cases} \quad (3.5)$$

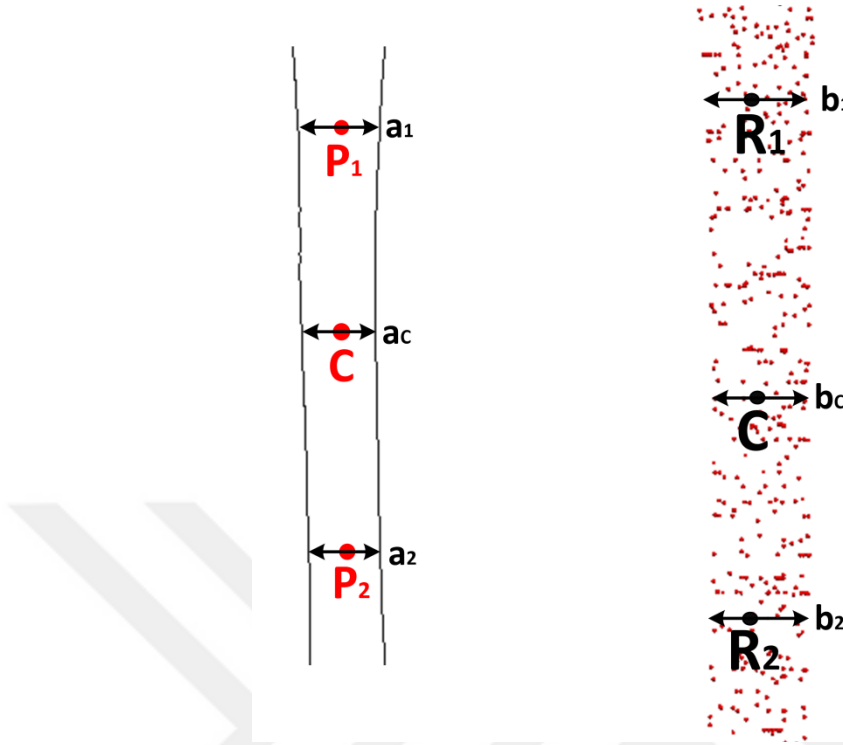


Figure 3.7. Cross-sections where thicknesses and scaling factors are calculated

To apply the transverse scaling, the canonical bone image is again processed by dividing it into a certain number of strips. The left-most vertex in each strip is found and mapped to the appropriate point located at the same level on the actual bone image. Each vertex is then shifted to the right according to the scaling factor calculated from its y-coordinate per Eq.(3.5). This simple linear mapping procedure is performed separately in AP and L views.

To summarize, considering a general vertex P with coordinates p_x, p_y, p_z , axial scaling will map the coordinate p_y to a value q_y , while transverse scaling will map the coordinates p_x and p_z to q_x and q_z , respectively. The transformed model, which consists of the transformed vertices, should match reasonably well with the actual bone in size and shape. As the final computation step, the angle θ , calculated and applied previously, is fine-tuned by a correction angle δ . The angle δ is changed in small increments and the scaled model is rotated about the center C by this angle to find the optimal value of δ which results to the smallest matching error. The used error measure is explained in detail in Section 3.5.

3.2.2. Model Manipulation for Undeformed Bones with FFD

The method based on FFD also consists of four steps: (i) Generating the projection of the canonical bone model. (ii) Detection of bone edges on the X-ray images. (iii) Determining the relative orientation of the bone image in the X-ray and canonical bone model. (iv) Scaling and matching of the canonical model to the bone image. The first three steps are identical to those in linear mapping method, explained in Section 3.2.1. Therefore, only step (iv) is explained here.

3.2.2.1. Model Scaling and Matching

Scaling in axial direction is performed as explained in section 3.2.1.4, and for transverse scaling, three different thickness values and corresponding scaling factors are calculated using Eqs (3.4) and (3.5). The procedure that differs from linear mapping is that the canonical bone model is now placed into a FFD grid. Then the control points constituting the grid are displaced in transverse direction according to the scaling factor calculated from y-coordinates of the control points per Eq. (3.5) (Figure 3.8).

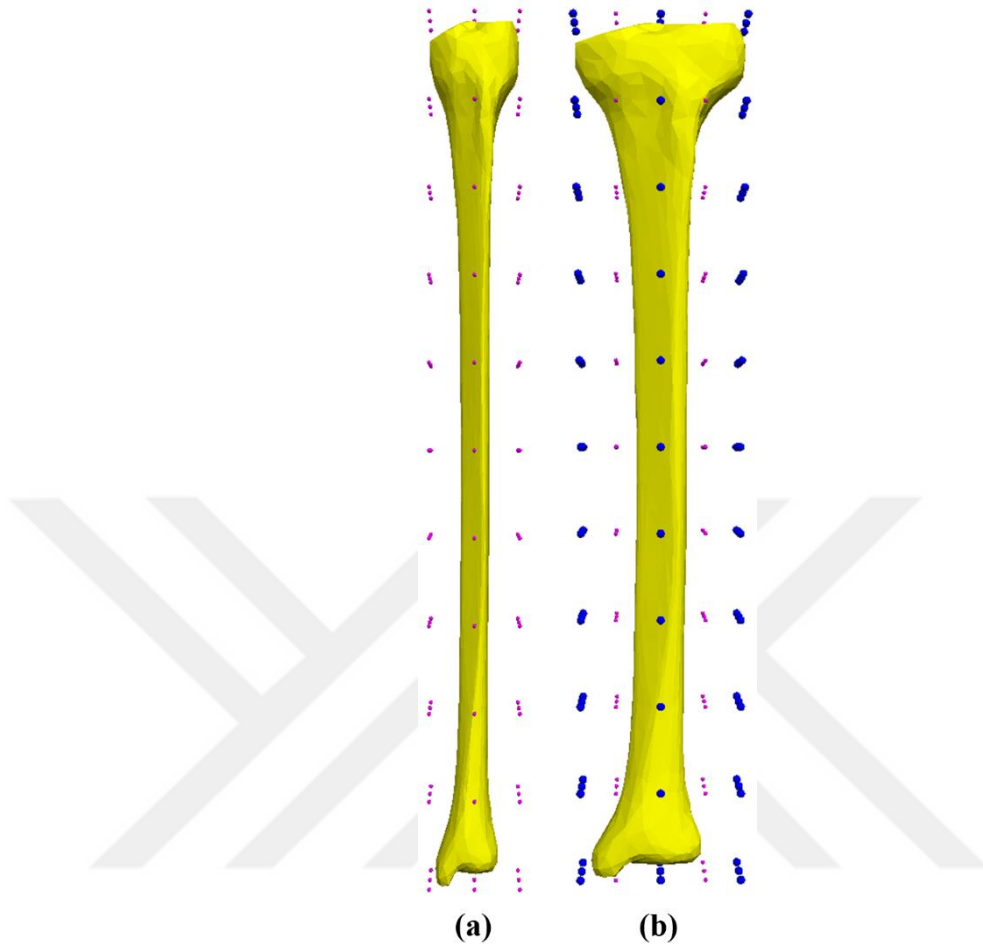


Figure 3.8. The canonical bone model before (a) and after (b) transverse scaling together with the control points in their original positions (magenta dots) and displaced positions (blue dots).

Generally, a vertex in the bone model does not deform as much as a FFD control point. Hence, the transverse scaling applied to a control point is not exactly reflected to a vertex. Because of that, scaling to the bone model through FFD is applied multiple times. After each scaling procedure a new undeformed grid is placed to enclose the bone model and scaling is applied again. This is repeated until control points and vertices are scaled by the same factor, i.e until transverse scaling of the bone model converges to the desired value.

3.3. Model Manipulation for Deformed Bones

Generation of the deformed bone model differs significantly from the generation of undeformed bone model, although there are some similar steps involved. First, it should be noted that, for deformed bone modeling, it is not necessary to bring the canonical and actual bone images to the same orientation. The undeformed model is created by following four steps that are applied separately to AP and L views: (i) Generating the projection of the canonical bone model, (ii) Detection of bone edges on the X-ray images, (iii) Scaling the canonical model along axial and transverse directions, (iv) Deforming the canonical model according to the bone image.

Steps (i) and (ii) are similar to those in the creation of undeformed bone models and explained in Sections 3.2.1.1 and 3.2.1.2. Steps (iii) and (iv) are detailed below.

3.3.1. Scaling the Canonical Model in Axial and Transverse Directions

Scaling the undeformed bone model in axial direction is performed with linear scaling as described in Section 3.2.1.4. However, the transverse scaling differs in the way we calculate the thickness values using the actual bone images.

After the bone edges are detected from the X-ray images (Figure 3.9), two bone axes are drawn at the distal and proximal parts of the bone image. Thickness values are calculated at proximal and distal parts and at CORA level. The thickness measurement is done along lines perpendicular to the related anatomic axes at proximal and distal parts. These levels are chosen such that the lengths of the segments $|\overline{PP'}|$ and $|\overline{DD'}|$ are equal to 10% of the associated fragment lengths, $|\overline{PC}|$ and $|\overline{DC}|$, respectively (Figure 3.10).

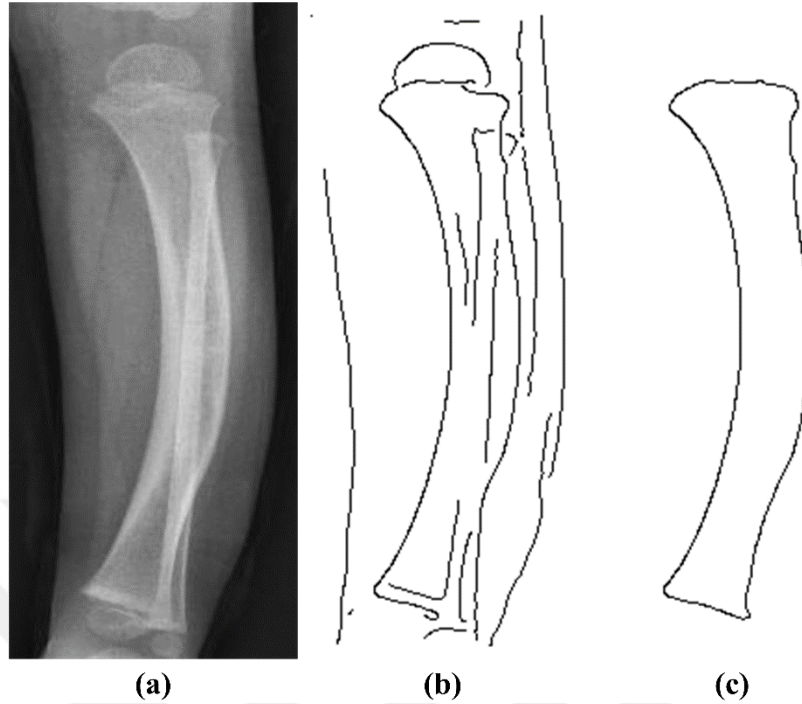


Figure 3.9. AP X-ray of tibia (a). Output of canny edge detection (b). Edges after manual intervention (c).

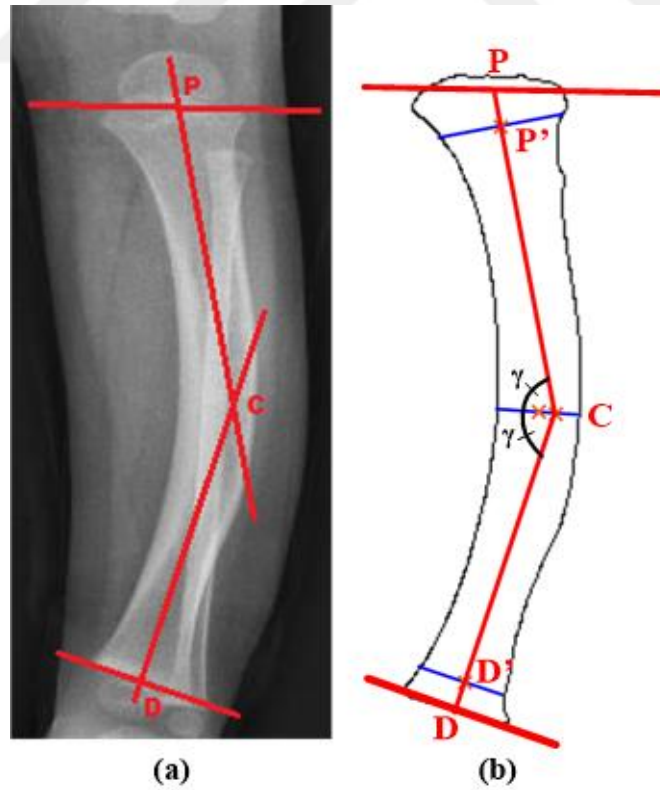


Figure 3.10. Joint orientation lines, proximal and distal anatomic axes (red lines) (a) and segments where thickness is measured (blue lines) (b).

Once these thickness values are obtained, the transverse scaling is performed with the method introduced in Section 3.2.2.1. Since the deformed bones are examples of pathologic cases, their calculated thickness values, and hence the associated scaling factors, may possess a higher variance throughout the bone axis when compared to undeformed bones. For instance, depending on the initial dimensions of the canonical model and the severeness of the pathology of the actual bone, the calculated scaling factors may entail the bone thickness be decreased at proximal and distal parts and increased at CORA level in the canonical model (Figure 3.11). This transverse scaling process is also applied repeatedly until the correct scaling is imposed on the bone model (See Section 3.2.2.1).

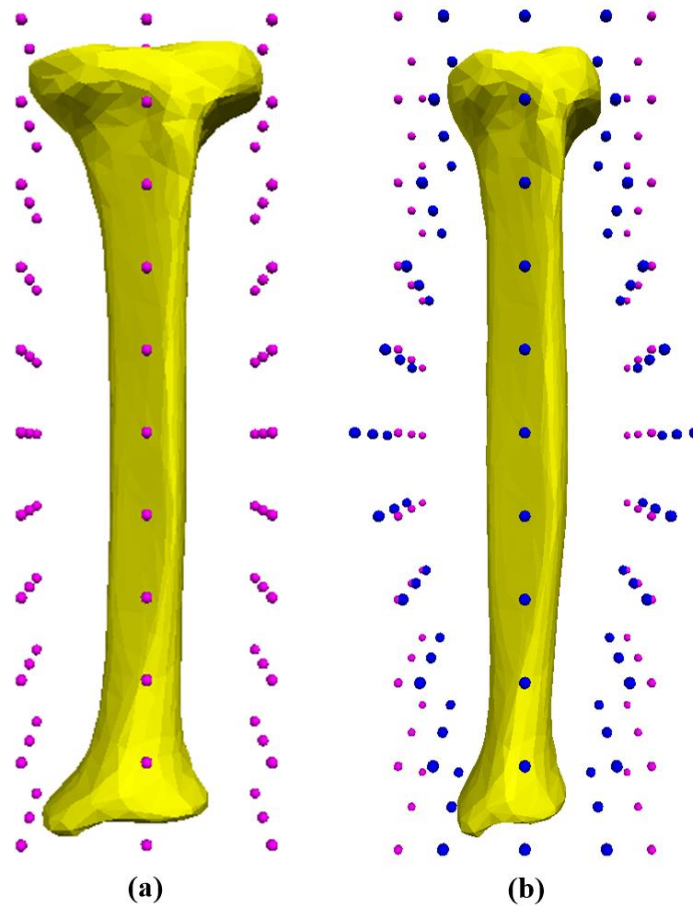


Figure 3.11. The canonical bone model before (a) and after (b) transverse scaling together with the control points in their original positions (magenta dots) and displaced positions (blue dots).

3.3.2. Deforming the Canonical Model According to Bone Image

The deformation of the canonical bone is performed for matching the shape of the canonical bone with the actual bone on X-ray images. The process begins with curve fitting to the midline of the bone on the X-ray images, using either a fourth order polynomial or a cubic spline, which is done separately in AP and L views. The fitted curves are expected to overlap with the midline of the deformed bone in both views while satisfying certain constraints at the bone ends. The curves provide a quantification of the bone shape which is then used in FFD to deform the canonical bone model to the actual bone.

3.3.2.1. Fourth Order Polynomial Modeling

In this method, the path between the points P and D is modeled with a fourth order polynomial passing through point, located on the line segment that bisects the angle PCD (point e). (Figure 3.12).

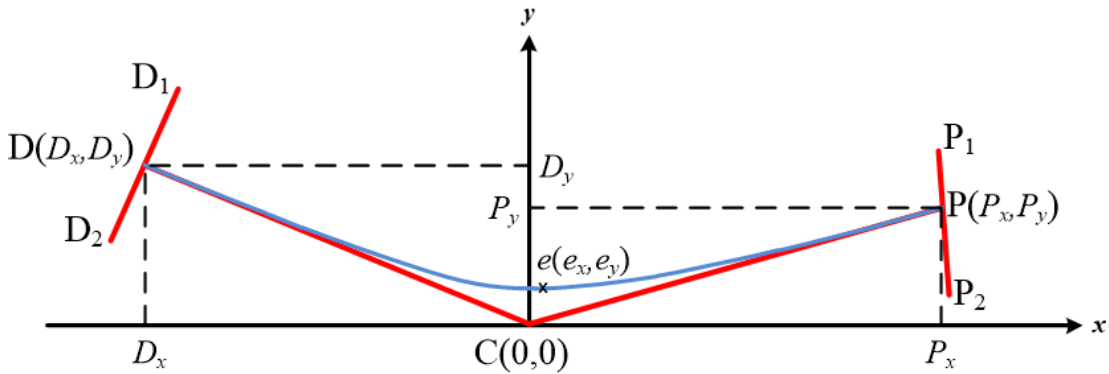


Figure 3.12. The anatomic axes and joint orientation lines (red lines), and the fitted fourth order polynomial (blue curve).

The origin of the 2D coordinate system is placed at the CORA and the bone is assumed to lay horizontally in this system. The five coefficients of the fitted general fourth order polynomial in one variable x , namely

$$p(x) = a_1x^4 + b_1x^3 + c_1x^2 + d_1x + e_1 \quad (3.6)$$

can be uniquely determined by using the constraints:

- i) $p(D_x) = D_y$
- ii) $p(P_x) = P_y$
- iii) $p(e_x) = e_y$
- iv) $p'(D_x) = -1/m_d$
- v) $p'(P_x) = -1/m_p$

The first three constraints ensure that the fitted curve passes through the points D, P and e . The values m_p and m_d are the slopes of the joint orientation lines P_1P_2 and D_1D_2 , respectively. Therefore, constraints (iv) and (v) control the slopes of the polynomial at the bone ends, which are set to be perpendicular to the joint orientation lines.

To find the coefficients, a set of five equations in five unknowns are solved which can be written in matrix form as:

$$\begin{pmatrix} D_x^4 & D_x^3 & D_x^2 & D_x & 1 \\ P_x^4 & P_x^3 & P_x^2 & P_x & 1 \\ e_x^4 & e_x^3 & e_x^2 & e_x & 1 \\ 4D_x^3 & 3D_x^2 & 2D_x & 1 & 0 \\ 4P_x^3 & 3P_x^2 & 2P_x & 1 & 0 \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \\ e_1 \end{pmatrix} = \begin{pmatrix} D_y \\ P_y \\ e_y \\ -1/m_d \\ -1/m_p \end{pmatrix} \quad (3.7)$$

3.3.2.2. Cubic Splines Modeling

In this method, the path between the points P and D is modeled with a cubic spline.

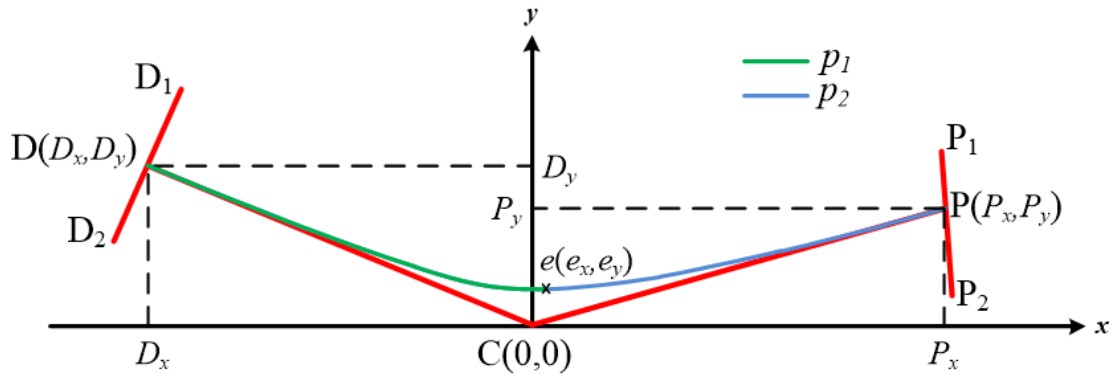


Figure 3.13. The anatomic axes and joint orientation lines (red lines), and the fitted cubic splines (green and blue curves)

Two subintervals are defined between point pairs D-e and e-P. The corresponding cubic interpolating polynomials are defined, respectively, as:

$$p_1(x) = a_1x^3 + b_1x^2 + c_1x + d_1 \quad (3.8)$$

$$p_2(x) = a_2x^3 + b_2x^2 + c_2x + d_2 \quad (3.9)$$

whose coefficients are to be determined according to the constraints:

- i) $p_1(e_x) = e_y$
- ii) $p_2(e_x) = e_y$
- iii) $p_1(D_x) = D_y$
- iv) $p_2(P_x) = P_y$
- v) $p_1'(e_x) = p_2'(e_x)$
- vi) $p_1''(e_x) = p_2''(e_x)$
- vii) $p_1'(D_x) = -1/m_d$
- viii) $p_2'(P_x) = -1/m_p$

The first four constraints ensure that the spline passes through the points P, D and e . As in the case of polynomial curve fit, m_p and m_d are the slopes of the lines P_1P_2 and D_1D_2 , respectively. Therefore, constraints (vii) and (viii) control the slope of the spline at the first (point D) and the last (point P) node which are set to be again perpendicular to the joint orientation lines. Constraints (v) and (vi) ensure the continuity of the first and second derivatives of the spline across the transition node e .

The resulting equation system in eight unknowns can be written in the following matrix form:

$$\begin{pmatrix} e_x^3 & e_x^2 & e_x & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e_x^3 & e_x^2 & e_x & 1 \\ D_x^3 & D_x^2 & D_x & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P_x^3 & P_x^2 & P_x & 1 \\ 3e_x^2 & 2e_x & 1 & 0 & -3e_x^2 & -2e_x & -1 & 0 \\ 6e_x & 2 & 0 & 0 & -6e_x & -2 & 0 & 0 \\ 3D_x^2 & 2D_x & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3P_x^2 & 2P_x & 1 & 0 \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \\ a_2 \\ b_2 \\ c_2 \\ d_2 \end{pmatrix} = \begin{pmatrix} e_y \\ e_y \\ D_y \\ P_y \\ 0 \\ 0 \\ -1/m_d \\ -1/m_p \end{pmatrix} \quad (3.10)$$

3.3.2.3. Canonical Model Deformation According to the Functional Models

The deformation algorithm matches the midline of the canonical bone with the midline of the deformed bone both in AP and L views. Furthermore, it is made sure that every transverse plane in the canonical bone model transforms to a plane that is perpendicular to the fitted curve in the deformed bone model. This involves a rotation and a translation to be applied to the canonical bone geometry using FFD. The procedure is performed separately in the AP and L planes.

After the functional model of the bone midline is obtained with one of the methods explained in Sections 3.3.2.1 (Figure 3.14a) and 3.3.2.2 (Figure 3.14b), control points of a FFD space enclosing the canonical bone model are displaced according to the functional models.

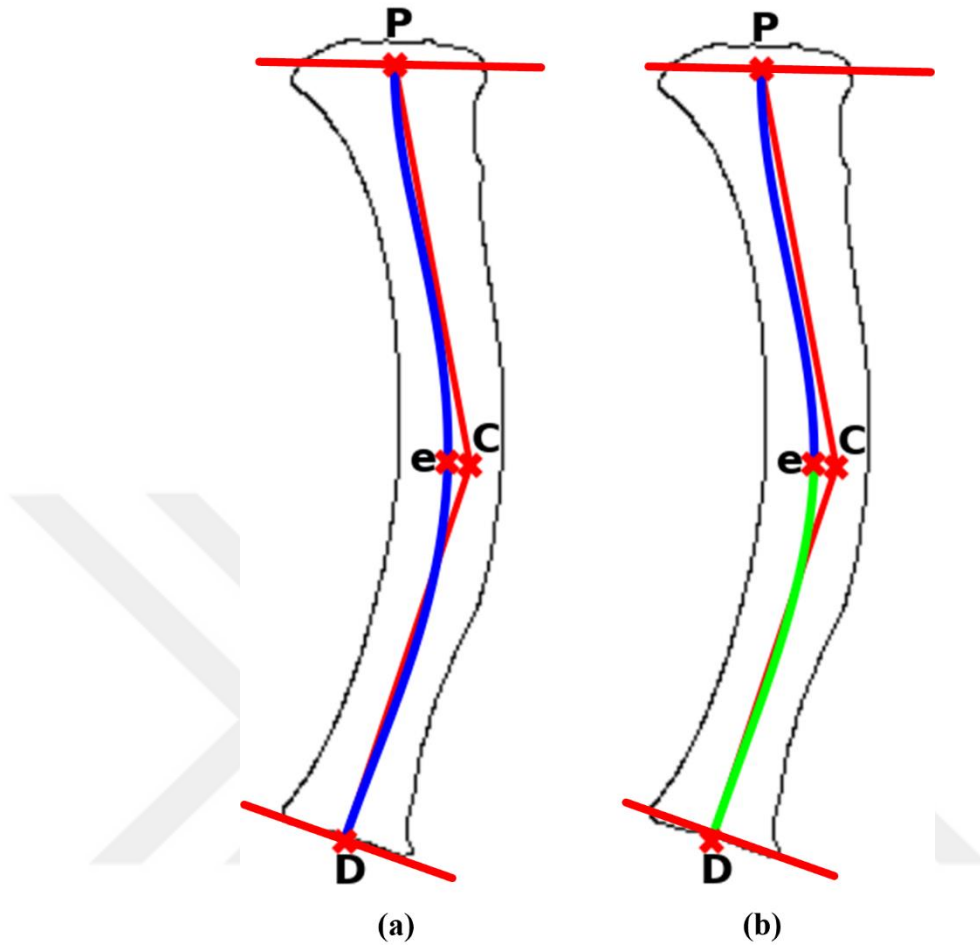


Figure 3.14. Functions obtained with fourth order polynomial fit (a) and cubic splines fit (b) superposed on the edge images

Magnitudes of the rotations and translations are determined with the help of the curve fitted to the bone midline. The value of $p(x)$ prescribes how much a general control point $A(x_A, y_A, z_A)$ is translated in x -direction. The transverse plane that passes through $A(x_A, y_A, z_A)$ projects as the line $x=x_A$ on AP and L views and this line is rotated such that it becomes perpendicular to the fitted curve at $x=x_A$ after deformation.

For practical purposes, rotation is performed first, followed by translation. The rotation angle ϕ in the considered plane can be found using $p'(x)$ value. For the polynomial case, the rotation angle at a general x level is defined by

$$\phi = \tan^{-1} \left(\frac{-1}{p'(x)} \right) \quad (3.11)$$

For cubic splines, ϕ is determined by the respective spline valid at the considered x level:

$$\phi = \begin{cases} \tan^{-1} \left(\frac{-1}{p'_1(x)} \right) & \text{if } x < e_x \\ \tan^{-1} \left(\frac{-1}{p'_2(x)} \right) & \text{if } x \geq e_x \end{cases} \quad (3.12)$$

In AP and L views, the calculated rotation is performed about the point $(x, 0)$. Following that, all control points on that rotated line (plane) are translated by the amount $p(x)$ for the polynomial fit, or $p_1(x)$ or $p_2(x)$ for the spline fit. A sample deformed model obtained through this procedure is given in Figure 3.15.

The deformation procedure is done first in AP and then in L views. In other words, the canonical bone is deformed first in AP, then in L planes. Note that the rotation operation, in general, is mathematically not commutative. Theoretically, there may be a difference in the outcome if the rotation is performed first in L, then in AP view. However, since rotation angles involved in the deformation procedure are relatively small, the difference is negligible.

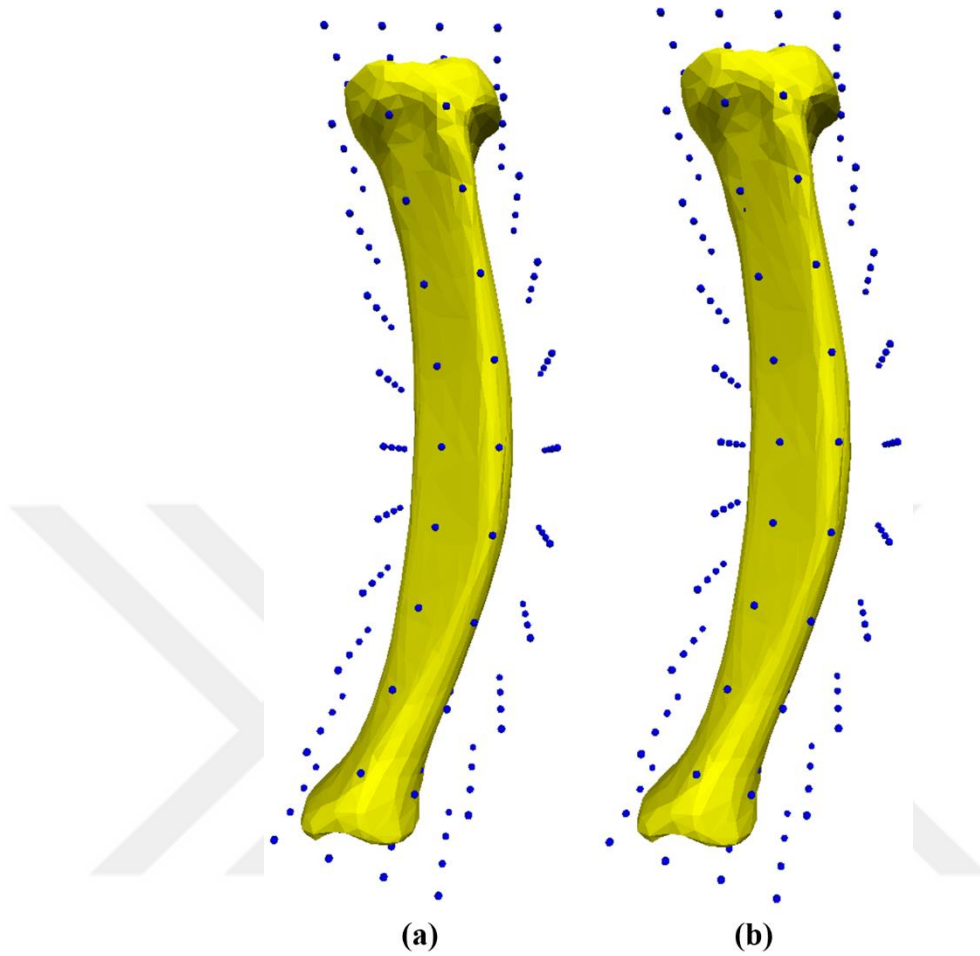


Figure 3.15. Final canonical bone models after deformation according to the fourth order polynomial (a) and the cubic splines (b)

3.4. Splitting the Final Model and Modeling of Callus Formation

The final canonical bone model should be further processed in order to enable a more realistic treatment simulation. This involves splitting the bone at CORA level, as done in an actual osteotomy operation, and visualizing the formation of the *callus* tissue, a precursor tissue in the generation of new bone material.

3.4.1. Splitting the Final Model

Dividing the final model to two submodels, corresponding to proximal and distal bone fragments, is necessary for generating the patient-specific bone-fixator

system. In the visualization tool, these submodels are attached to the fixator model according to data obtained from X-ray images.

The splitting of the bone is performed along the bisector line drawn through the CORA (See Figure 3.10). With respect to the image frame, the osteotomy site may not be coincident with the xz -plane. For convenience, the model is rotated such that the bisector line coincides with this plane. It is important to know which triangles in the original model will be divided by the splitting procedure. Accordingly, new vertices and associated connectivities need to be defined to create the submodels, as well as to model the callus formation. The proposed algorithm is detailed below:

Step 1: Rotating the model

To ensure that the bisector line $\overrightarrow{Ce}$ lies on the xz plane, the model is rotated about the z -axis (passing through C) by an angle of $\tan^{-1}(e_y / e_x)$

Step 2: Detecting the triangles of interest

All triangles intercepted by the the xz -plane are marked as the triangles of interest (Figure 3.16).

Step 3: Detecting intersection points

For each triangle of interest, intersection points of triangle edges with the xz -plane are obtained through simple algebra (Figure 3.16).

Step 4: Generating new triangles

Using the detected intersection points, new triangles are defined belonging to proximal (above the xz -plane) and distal (below the xz -plane) parts (Figure 3.17). Completeness of the submodels are ensured with these newly formed triangles.

Step 5: Rotating the submodels back

The final submodels are brought to their initial orientation by rotating them in the opposite direction by the amount calculated in Step 1.

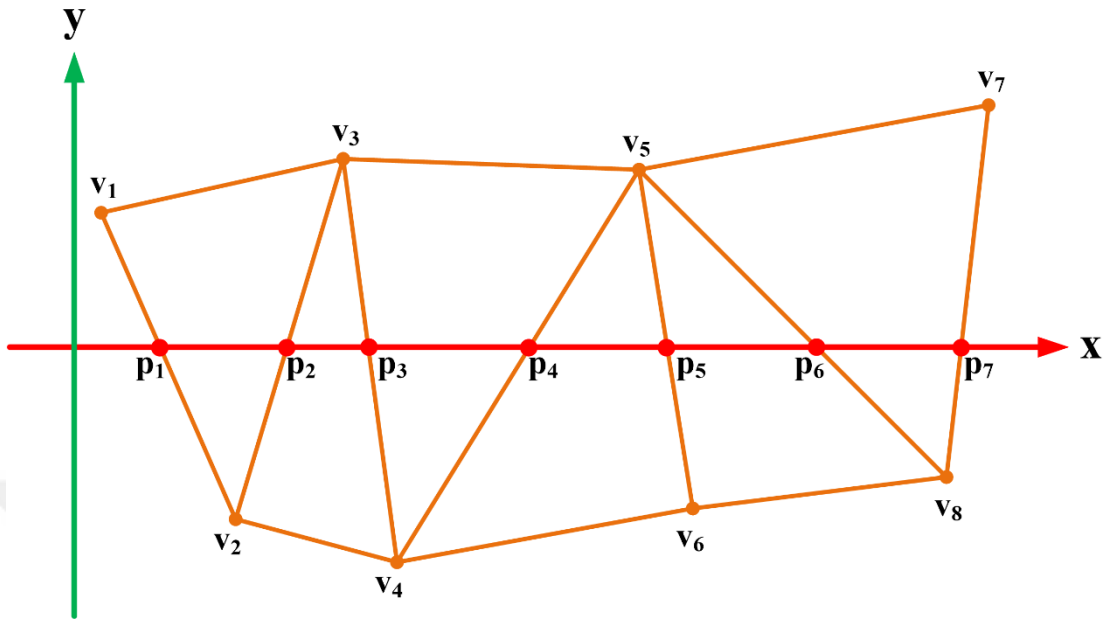


Figure 3.16. Vertices of triangles of interest (v) and detected intersection points (p)

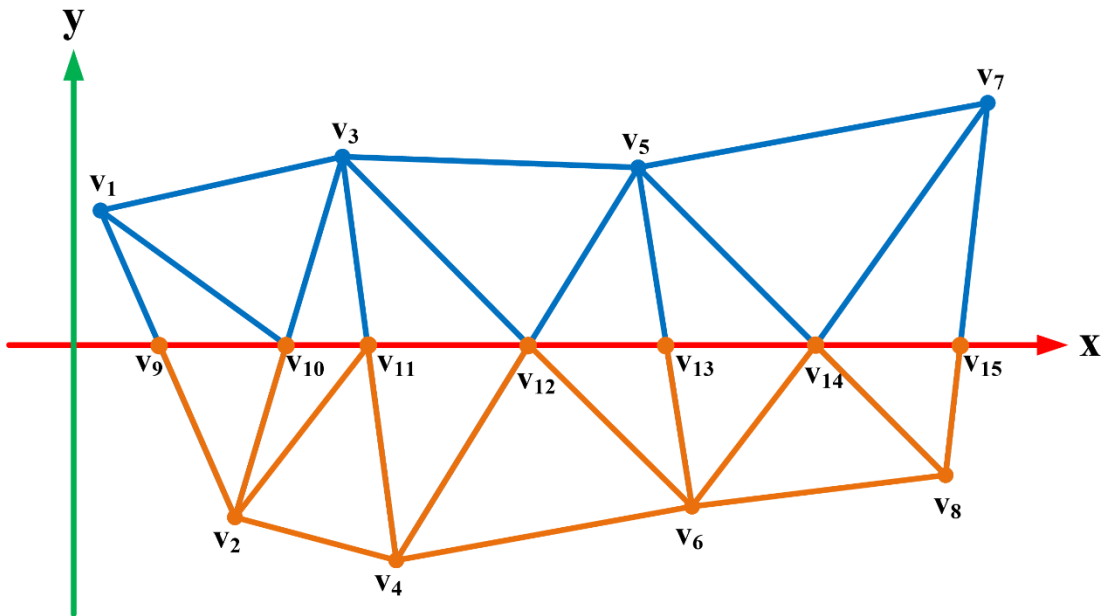


Figure 3.17. Generated new triangles representing the proximal (blue) and distal (orange) parts, respectively.

3.4.2. Modeling of Callus Formation

In bone lengthening or deformity correction procedures, the gap between the bone fragments is filled by callus tissue. The modeling of callus is accomplished by keeping the connectivity between the proximal and distal fragments after model splitting is done.

First, the located intersection points are ordered according to their angular positions around the bone circumference. A complete callus surface is generated by defining the intersection points on the distal and proximal side as vertices, and assigning them an appropriate connectivity (Figure 3.18).

Steps of the bone splitting and the corresponding callus modeling are illustrated in Figures 3.19, 3.20, and 3.21.

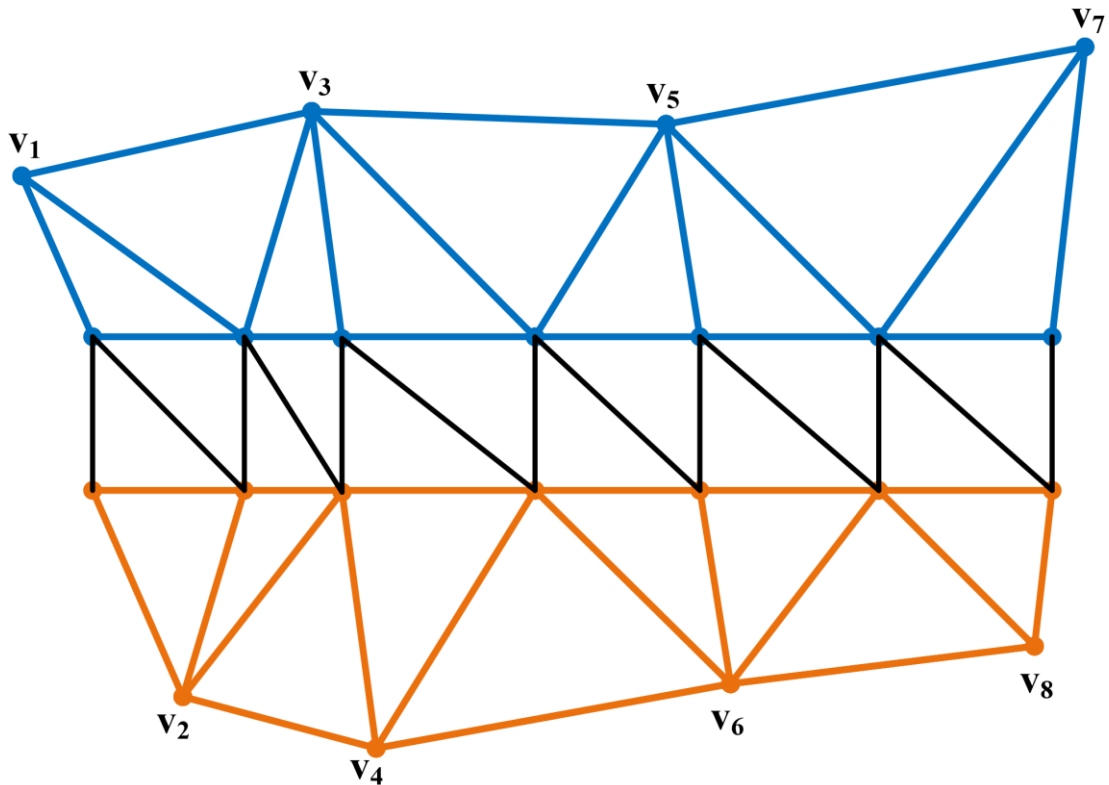


Figure 3.18. New triangles representing callus (black) are introduced between proximal and distal parts.

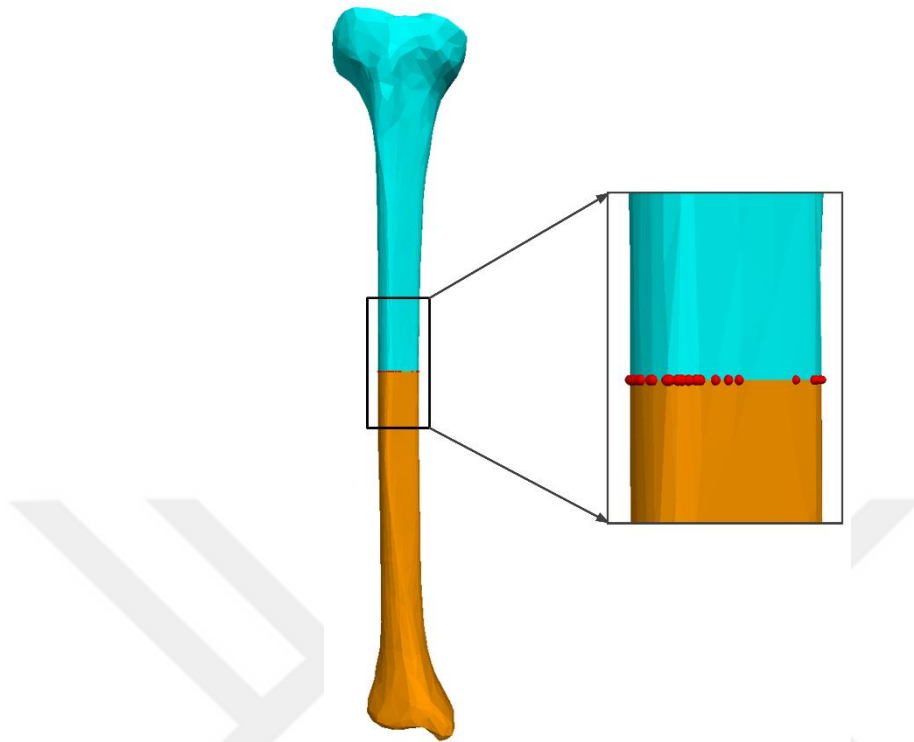


Figure 3.19. Detected intersection points (red dots)

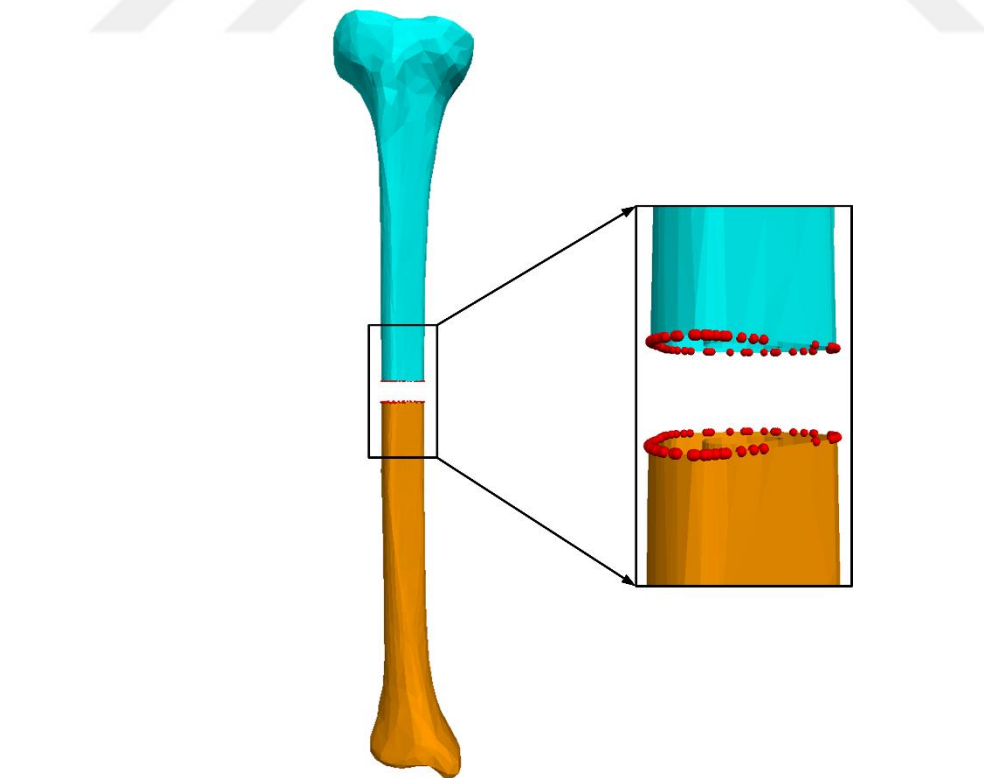


Figure 3.20. Splitting the bone into two pieces

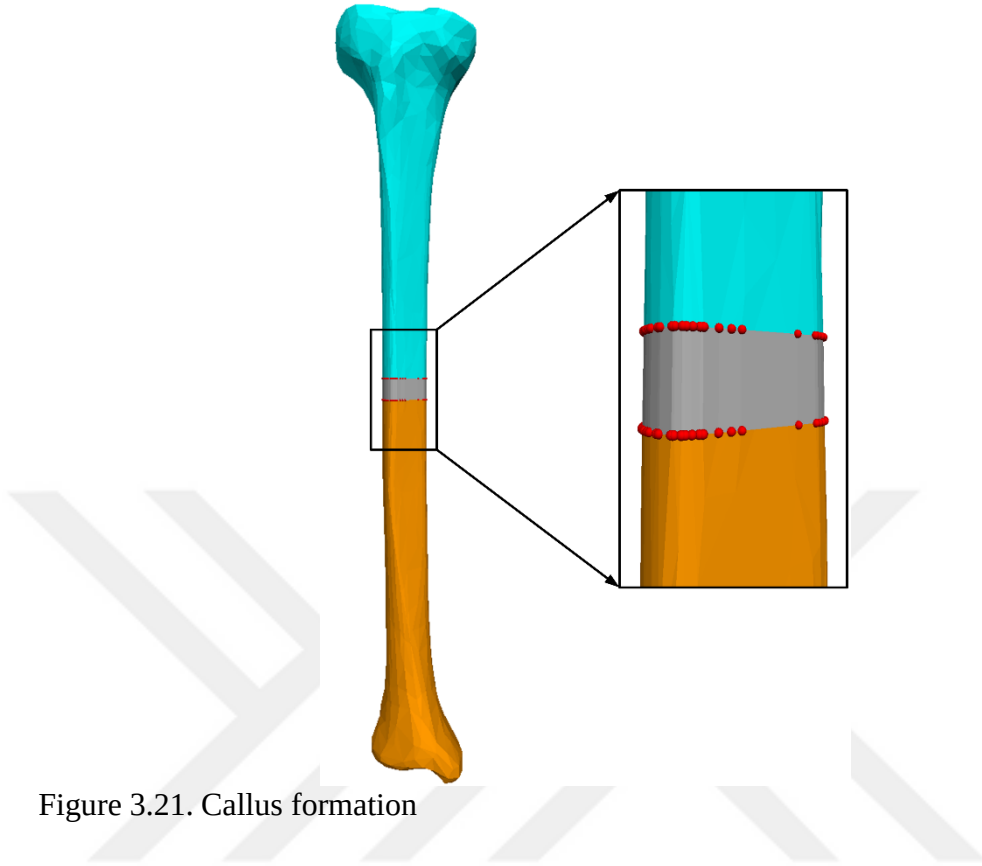


Figure 3.21. Callus formation

3.5. Error Calculation

In this section, a simple metric is proposed to quantify the error made when matching the canonical model to the actual bone. After the canonical bone is transformed, the Euclidian distance (in pixel dimension) between each edge point of the transformed canonical bone image and the corresponding point on the edge of the actual bone image is measured at several cross sections (Figure 3.22). The sum of these distances, E_{total} , given by:

$$E_{tot} = \sum_{i=1}^n l_i \quad (3.13)$$

is averaged over the number of measurement points n to obtain the average error E_{ave} :

$$E_{ave} = \frac{E_{total}}{n} \quad (3.14)$$

Since the bone size and the X-ray scale may differ from one patient to the other, E_{ave} is normalized by dividing it to the maximum width W , measured on the actual bone image, to obtain the *normalized error*, E_{norm} :

$$E_{norm} = \frac{E_{ave}}{W} \quad (3.15)$$

In that sense, an E_{norm} value of, say, 0.1 will correspond to a hypothetical situation where all measurement points of the transformed canonical bone image lie $0.1 \cdot W$ (10 % of the maximum width) away from the corresponding point on the actual bone image.

Note that the same error measure is also used to find the optimal angle δ to correct the orientation angle θ . δ is chosen such that E_{total} , attains a minimum for the set of discrete δ values tried (See Section 3.2.1.4).

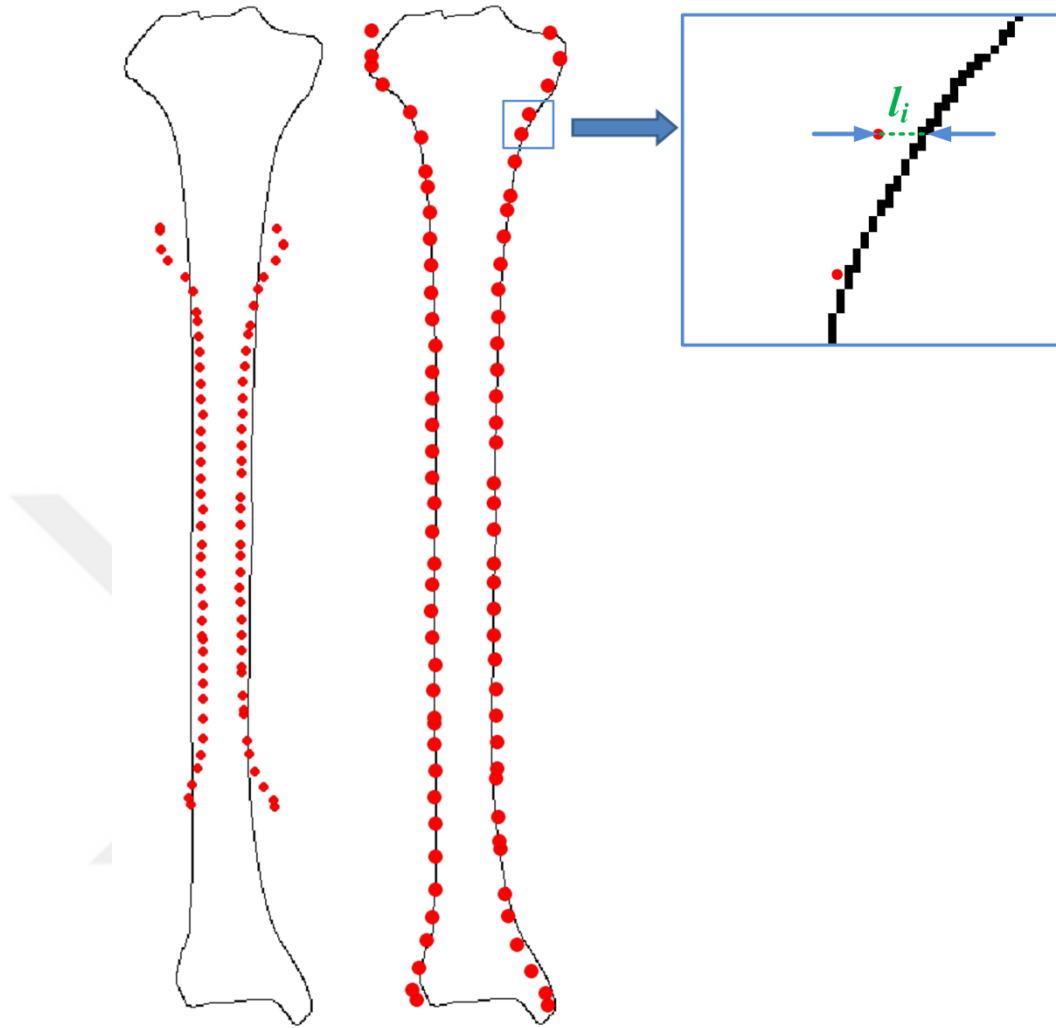


Figure 3.22. Superposed images of actual bone and canonical bone in AP view before (left) and after scaling (right). The error in the matching procedure is measured using the distances l_i



4. EXTERNAL FIXATOR TYPES

In this chapter, two different external fixator types are presented: *Gough-Stewart Platform* (SP) and *Lambda Fixator*. Two distinct GUIs are developed for each external fixator type. The associated mathematical theories are implemented within the GUIs whose usage is explained in Chapter 5. Development of the mathematical theories, which involve only kinematics for fracture fixation, is not in the scope of this thesis. For sake of completeness, inverse and direct kinematics of these external fixator types are summarized here.

4.1. Gough – Stewart Platform as External Fixator

4.1.1. Inverse Kinematics

The mathematical theory related with usage of (3-3) type and (6-6) type SP as an external fixator has been reported previously (Mutlu et al., 2006; Akçali, 2010).

The external fixator of SP type basically consists of a proximal ring (top ring), a distal ring (bottom ring) and six rods connecting these rings (Figure 4.1). After the fixator rings are attached to the bone fragments, two X-ray images are taken in perpendicular directions (conventionally, AP and L views). Then, certain quantities defining the bone/fixator geometry are measured on these X-ray images (Figures 4.2 and 4.3).

The mathematical theory defines a global coordinate system at the proximal ring center (x,y,z) and a relative coordinate system at the distal ring center (u,v,w) . Additionally, two extra coordinate systems are defined in the proximal (x',y',z') and distal (u',v',w') ring planes where the bone fragments intersect the related planes (Figure 4.1). Orthogonal transformations are obtained between these coordinate systems using the measured geometric entities as well as the clinically known data provided by the clinician (Figures 4.4 and 4.5). With these transformations, a vector defined in one coordinate system can be expressed in the other coordinate system. Therefore, rod lengths L_i for the initial fixator configuration can be computed as:

$$L_i = \left| \overrightarrow{GA_i} - \overrightarrow{GB_i} \right|_0 \quad i = 1-6 \quad (4.1)$$

where $\overrightarrow{GA_i}$ and $\overrightarrow{GB_i}$ are the vectors from the proximal ring center to the connection points on the proximal and distal rings, respectively. Since the positions of the bone fragments relative to the associated rings do not change while the fixator is advanced to its final configuration, the transformations between (x,y,z) and (x',y',z') , and between (u,v,w) and (u',v',w') system stay the same. Using this property, transformation between (x,y,z) and (u,v,w) for final configuration is calculated, as well as the final rod lengths L_i that will bring the bone fragments to the desired anatomical position:

$$L_{i_n} = \left| \overrightarrow{GA_i} - \overrightarrow{GB_i} \right|_n \quad i = 1-6 \quad (4.2)$$

where n is the number of proposed treatment steps by the clinician.

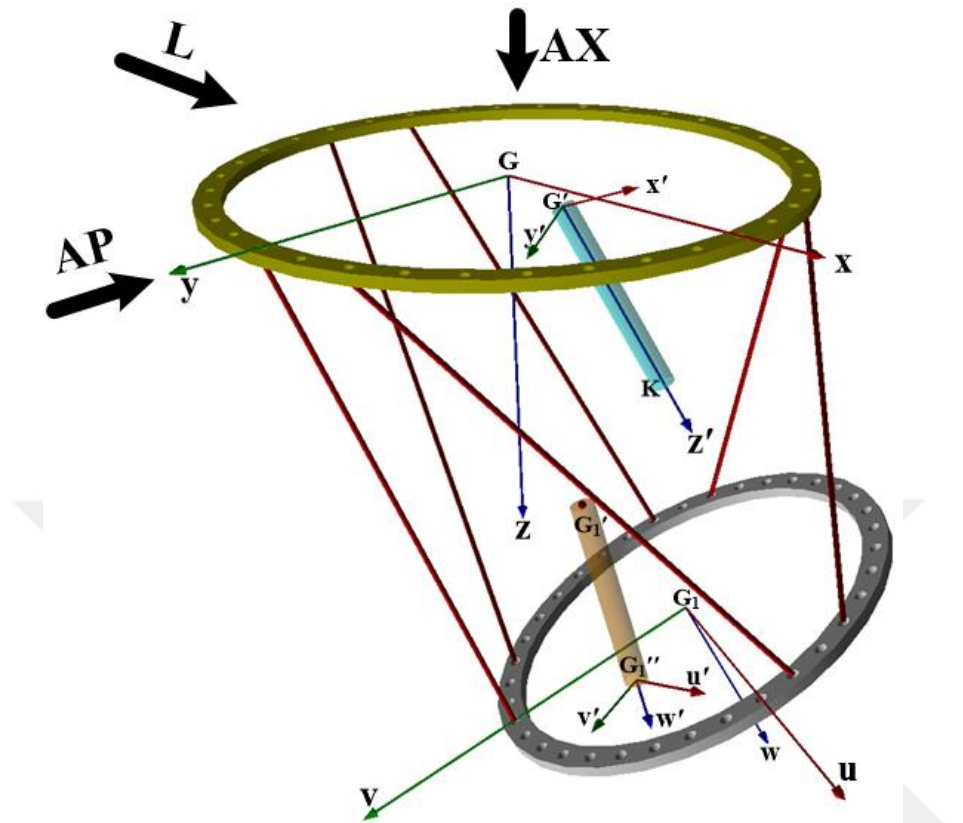


Figure 4.1. The biomechanical system and the locations of the coordinate systems

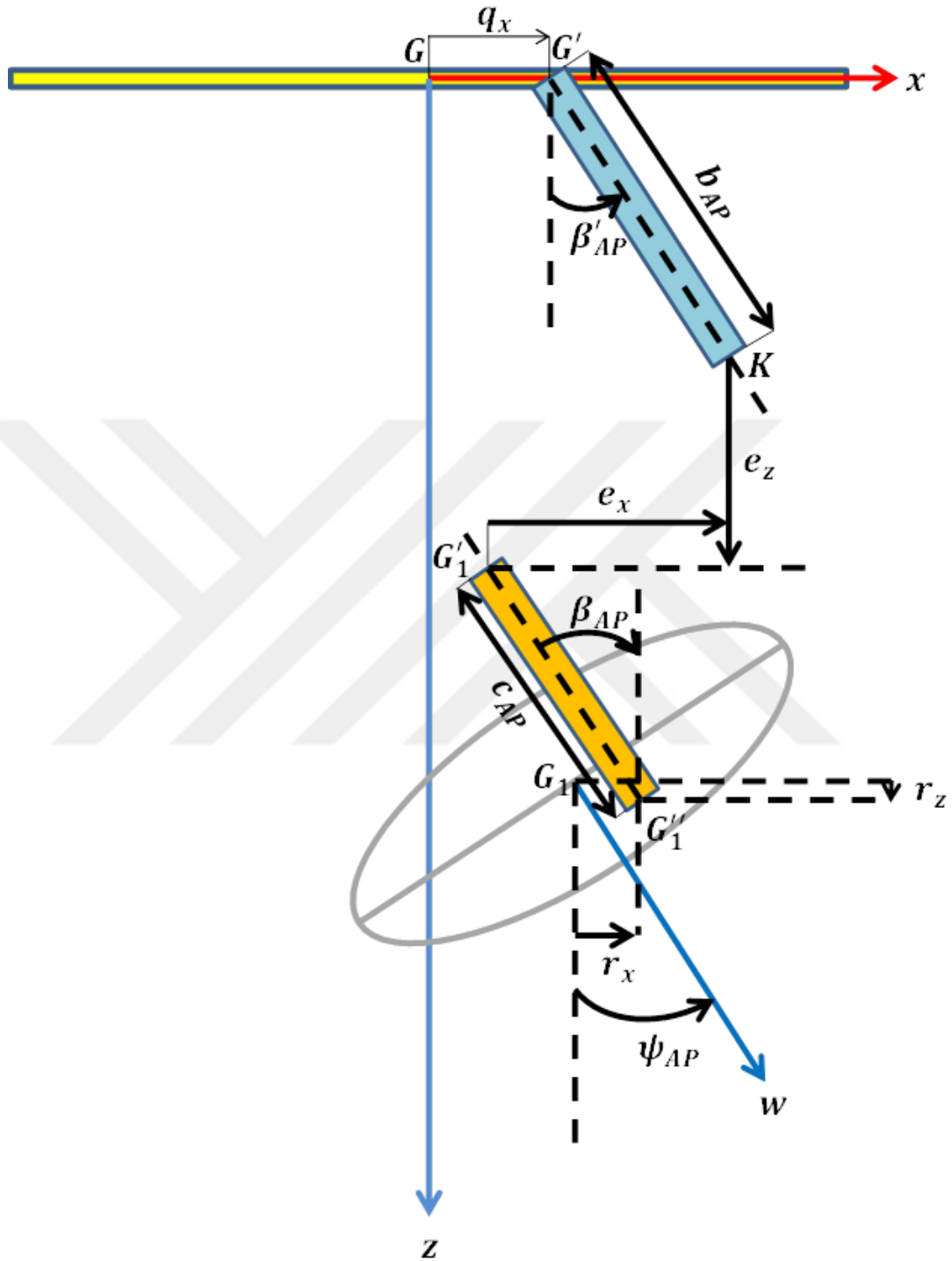


Figure 4.2. View – position relations of the biomechanical system in AP-view

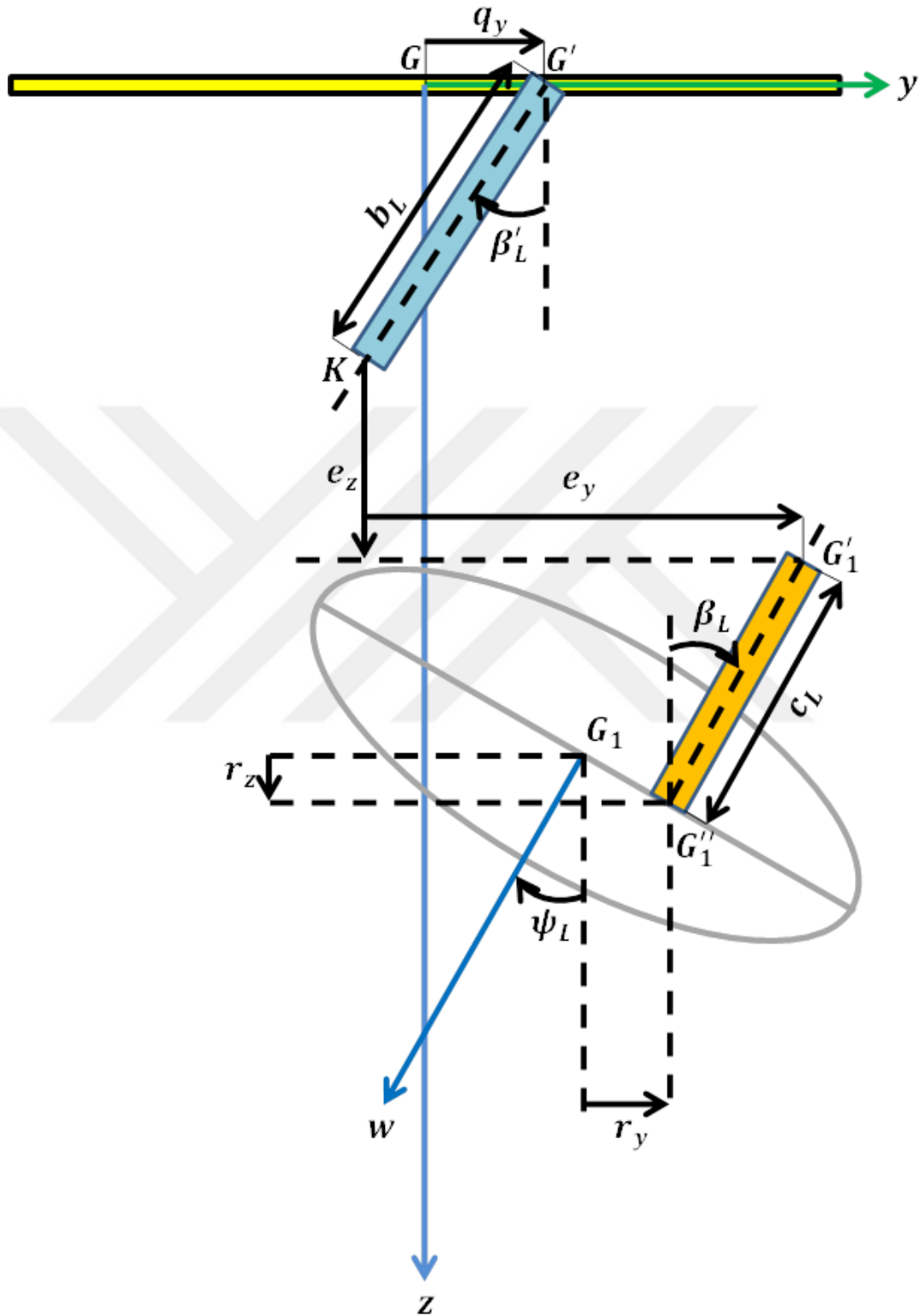


Figure 4.3. View – position relations of the biomechanical system in L-view

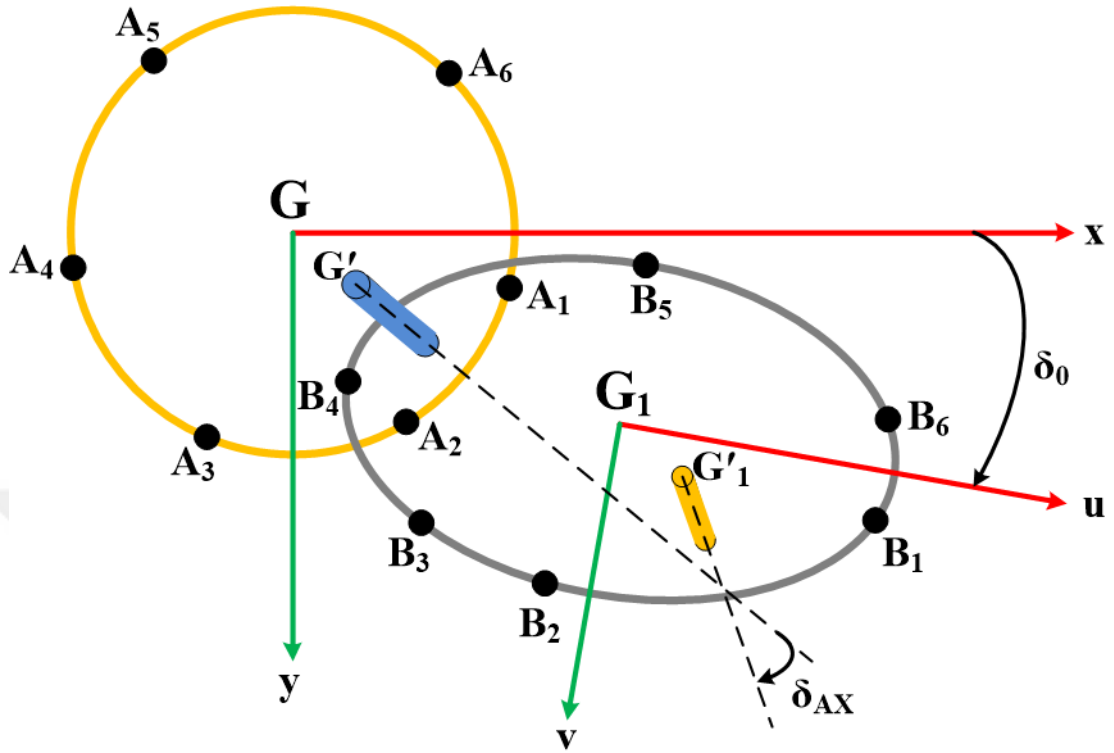


Figure 4.4. Relative angular orientation of the rings and the bone fragments

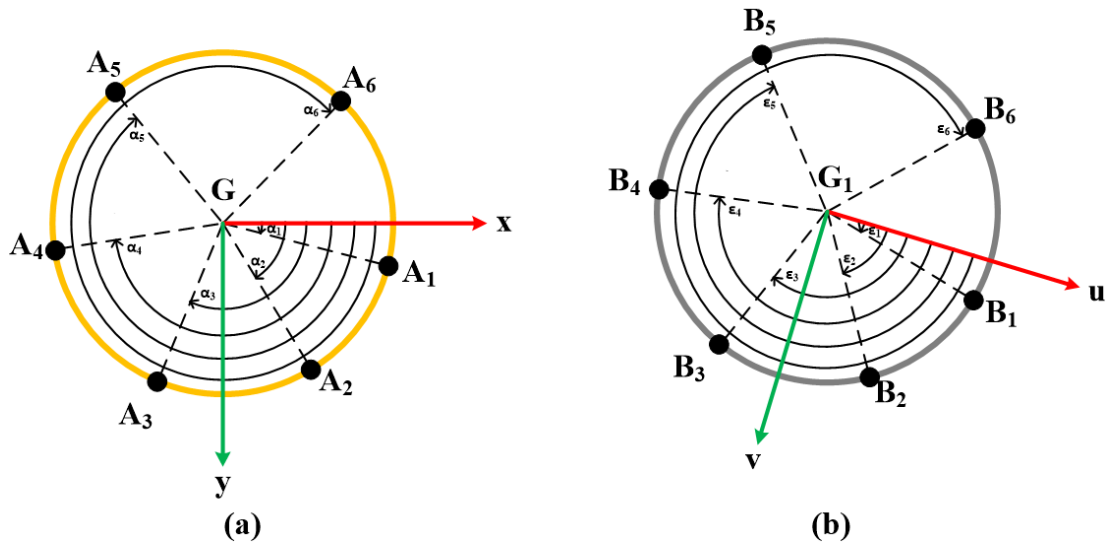


Figure 4.5. Angular positions of the joints at the proximal (a) and the distal (b) rings.

4.1.2. Direct Kinematics

In deformity correction and bone lengthening, the rod lengths are changed gradually, and the target fixation is achieved in n steps. This comes up with the necessity of determining and analyzing the intermediate fixator configurations which is nothing else than a direct kinematic problem.

Rod lengths of the biomechanical system both for initial and final configurations, rotations of the distal ring for the initial configuration, $(\theta_x, \theta_y, \theta_z)_0$, as well as the rotation matrices between the coordinate systems are the output of the inverse kinematics and these data serve as the input for direct kinematics. As the first step of direct kinematic calculations, translations of the distal ring, $(x_1, y_1, z_1)_0$, are calculated and the solution vector for the initial configuration, $(x_1, y_1, z_1, \theta_x, \theta_y, \theta_z)_0$, is obtained.

Once the initial solution vector as well as associated quantities are obtained, successive calculations of direct kinematics, involving suitable changes in rod lengths between initial and final configurations, are performed to find the intermediate configurations of the biomechanical system during the treatment process. These changes are determined according to a number of actuation functions which are explained in Section 4.1.2.1.

Referring to the geometry of SP (Figure 4.1), the distance between the relevant proximal and distal joints (amounting to the rod lengths $(L_i, i=1-6)$), is written to lead to the following displacement function $(f_i, i=1-6)$.

$$f_i = f_{i_0} - L_i^2 = 0 \quad (4.3)$$

f_{i_0} is a function of translations and rotations of distal ring relative to the proximal ring which are denoted by $(x_1, y_1, z_1, \theta_x, \theta_y, \theta_z)$. Knowing the rod lengths for the next treatment step and $(x_1, y_1, z_1, \theta_x, \theta_y, \theta_z)$ for the current step, we can

calculate translations for the next treatment step. This calculation is an iterative approach involving determinant of Jacobian matrix

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial \theta_x} & \frac{\partial f_1}{\partial \theta_y} & \frac{\partial f_1}{\partial \theta_z} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial \theta_x} & \frac{\partial f_2}{\partial \theta_y} & \frac{\partial f_2}{\partial \theta_z} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial y_1} & \frac{\partial f_3}{\partial z_1} & \frac{\partial f_3}{\partial \theta_x} & \frac{\partial f_3}{\partial \theta_y} & \frac{\partial f_3}{\partial \theta_z} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial y_1} & \frac{\partial f_4}{\partial z_1} & \frac{\partial f_4}{\partial \theta_x} & \frac{\partial f_4}{\partial \theta_y} & \frac{\partial f_4}{\partial \theta_z} \\ \frac{\partial f_5}{\partial x_1} & \frac{\partial f_5}{\partial y_1} & \frac{\partial f_5}{\partial z_1} & \frac{\partial f_5}{\partial \theta_x} & \frac{\partial f_5}{\partial \theta_y} & \frac{\partial f_5}{\partial \theta_z} \\ \frac{\partial f_6}{\partial x_1} & \frac{\partial f_6}{\partial y_1} & \frac{\partial f_6}{\partial z_1} & \frac{\partial f_6}{\partial \theta_x} & \frac{\partial f_6}{\partial \theta_y} & \frac{\partial f_6}{\partial \theta_z} \end{pmatrix} \quad (4.4)$$

4.1.2.1. Actuation Functions for the Rods

The rod lengths for each treatment step are to be determined according to the total number of treatment steps (n) decided by the clinician. In order to satisfy certain medical requirements, number of treatment steps may be re-selected. Traditionally, the clinicians tend to make equal sized changes in rod lengths during the whole treatment process. However, if the clinician wishes, adjustments in rod lengths can be made in such a way that changes have a time-dependent functional form, referred to here as *actuation function*. This allows different relative motion possibilities for the bone fragments. The actuation functions implemented within the GUI are given in Table 4-1 along with their associated constraints.

Table 4-1. Actuation functions and associated constraints. Here, i and j represent numbers for rods ($i=1-6$) and treatment steps ($j=1-n$), respectively.

	Model Function	Constraints	Coefficients
Linear	$L_{ij} = a_0 + a_1 j$	$j=0 \Rightarrow L_{ij} = L_{i0}$ $j=n \Rightarrow L_{ij} = L_{in}$	$a_0 = L_{i0}$ $a_1 = (L_{in} - L_{i0}) / n$
Quadratic #1	$L_{ij} = a_0 + a_1 j + a_2 j^2$	$j=0 \Rightarrow L_{ij} = L_{i0}$ $j=n \Rightarrow L_{ij} = L_{in}$ $(dL_j / dj)_{j=0} = 0$	$a_0 = L_{i0}$ $a_1 = 0$ $a_2 = (L_{in} - L_{i0}) / n^2$
Quadratic #2	$L_{ij} = a_0 + a_1 j + a_2 j^2$	$j=0 \Rightarrow L_{ij} = L_{i0}$ $j=n \Rightarrow L_{ij} = L_{in}$ $(dL_j / dj)_{j=n} = 0$	$a_0 = L_{i0}$ $a_1 = 2(L_{in} - L_{i0}) / n$ $a_2 = -(L_{in} - L_{i0}) / n^2$
Sinusoidal #1	$L_{ij} = a_0 + a_1 j + a_2 \cos \frac{\pi}{n} j$	$j=0 \Rightarrow L_{ij} = L_{i0}$ $j=n \Rightarrow L_{ij} = L_{in}$ $(dL_j / dj)_{j=n/2} = 0$	$a_0 = \frac{(\pi-1)L_{i0} - L_{in}}{\pi-2}$ $a_1 = \frac{L_{in} - L_{i0}}{n} \cdot \frac{\pi}{\pi-2}$ $a_2 = \frac{L_{in} - L_{i0}}{\pi-2}$
Sinusoidal #2	$L_{ij} = a_0 + a_1 j + a_2 \sin \omega j$	$j=0 \Rightarrow L_{ij} = L_{i0}$ $j=n \Rightarrow L_{ij} = L_{in}$ $(dL_j / dj)_{j=n/2} = 0$ $(d^2 L_j / dj^2)_{j=n/2} = 0$	$a_0 = L_{i0}$ $a_1 = (L_{in} - L_{i0}) / n$ $a_2 = (L_{in} - L_{i0}) / 2\pi$ $\omega = 2\pi / n$
Sinusoidal #3	$L_{ij} = a_0 + a_1 \sin \omega_1 j + a_2 \cos \omega_2 j$	$j=0 \Rightarrow L_{ij} = L_{i0}$ $j=n \Rightarrow L_{ij} = L_{in}$ $(dL_j / dj)_{j=0} = 0$ $(d^2 L_j / dj^2)_{j=n} = 0$	$a_0 = (L_{in} + L_{i0}) / 2$ $a_1 = 0$ $a_2 = -(L_{in} - L_{i0}) / 2$ $\omega_2 = \pi / n$
Cubic Polynomial	$L_{ij} = a_0 + a_1 j + a_2 j^2 + a_3 j^3$	$j=0 \Rightarrow L_{ij} = L_{i0}$ $j=n \Rightarrow L_{ij} = L_{in}$ $(dL_j / dj)_{j=n/2} = 0$ $(d^2 L_j / dj^2)_{j=n/2} = 0$	$a_0 = L_{i0}$ $a_1 = 3(L_{in} - L_{i0}) / n$ $a_2 = -6(L_{in} - L_{i0}) / n^2$ $a_3 = 4(L_{in} - L_{i0}) / n^3$

4.1.2.2. Singularity of the Fixator Frame

The configuration where fixator loses its stability is called a *singular* position. In order to ensure physical security, the fixator frame should be stable throughout the entire treatment. Therefore, in applications, it should be checked whether the configuration is stable at initial, final and intermediate steps.

Singularity of the frame is avoidable by means of checking the sign of Jacobian matrix's determinant (Eq. (4.4)). The direct kinematics involves taking inverse of Jacobian matrix, thus, a Jacobian matrix having a zero determinant is non-invertible. Such a configuration is an example of a singular position. Furthermore, a sign change between two consecutive steps means that determinant of Jacobian becomes zero at some point when moving from one position to next one.

In case of encountering a singular position within the proposed treatment scheme, location of one of the joints is changed and calculations are repeated. Another way of avoiding singularities is to change the actuation function for at least one of the rods. In this case, the fixation path of distal fragment changes as well. Therefore, this new path should be visually checked and approved by the clinician prior to initializing the treatment.

The developed GUI performs singularity analysis and warns the user in case of a singular position. Entire capabilities of the GUI are demonstrated in Chapter 5.

4.2. The Lambda Fixator

Shortcomings of the SP, such as risk of encountering a singular position and bone images being blocked by rods on X-ray films, have led to the development of a new external fixator design, referred to as *lambda fixator*. The mathematical theory behind the lambda fixator and its GUI are described elsewhere (Aydin, 2015). In this work, new features are added to the GUI of this fixator type. Hence, the mathematical theory implemented in the GUI is summarized here.

4.2.1. Structure of the Lambda Fixator

In the lambda fixator structure (Figure 4.6), connection between the rods is achieved by three lambda modules. A lambda module consists of two rods, one of which is longer than the other, and the short one is connected to the body of the long one at an appropriate location, resulting a shape that resembles the Greek letter “ λ ” (Figure 4.7).

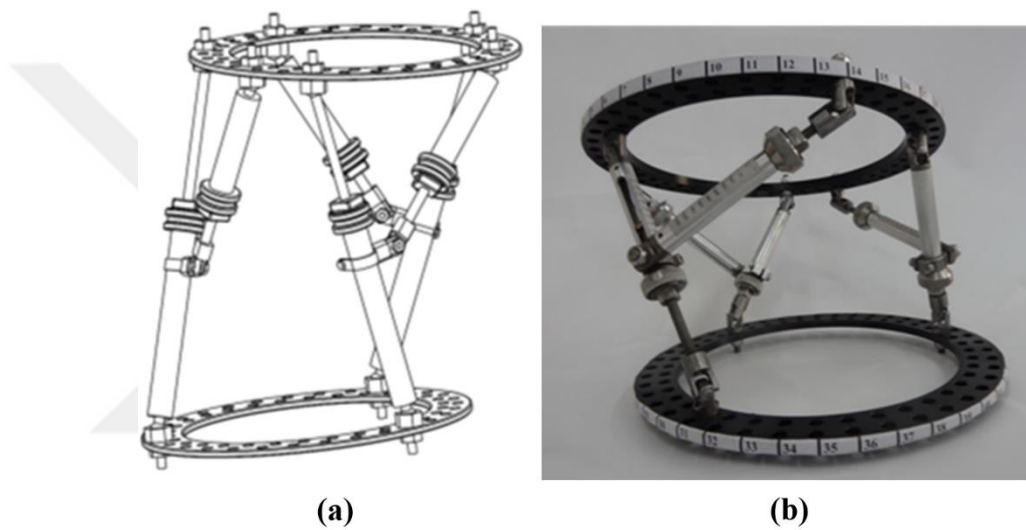


Figure 4.6. Sketch (a) and picture (b) of a lambda fixator

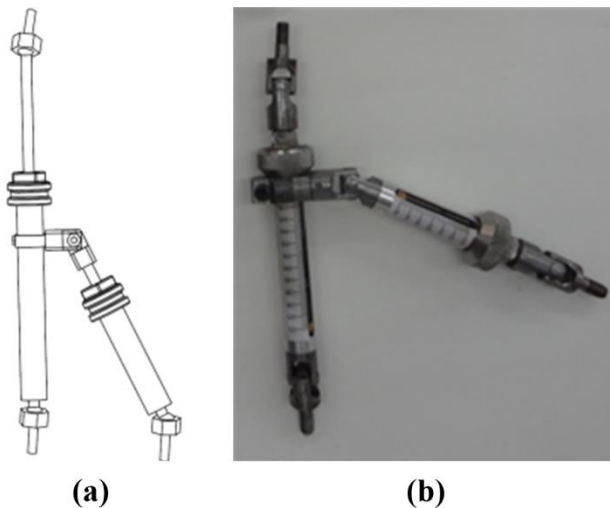


Figure 4.7. Sketch (a) and picture (b) of a lambda module

The quantities measured on the long rod of the lambda module are constant (s) and variable (l) length components, the sum of which constitutes its total length (L'). Lengths of connection element (e) and short rod (L'') are also measured to characterize the shape of the module. Additionally, location of the lambda module on the proximal and distal rings is defined by the connection points (A_i) and (B_i), respectively (Figure 4.8).

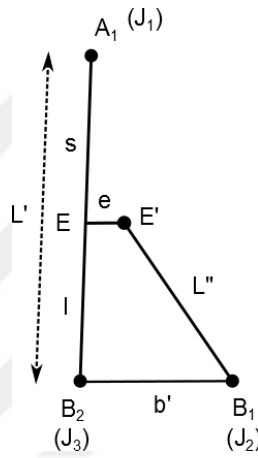


Figure 4.8. Naming convention of the lambda module

Note that the lambda module can be connected to the rings in eight different ways (Figure 4.9). For a fixator with three modules, this gives rise to $8^3 = 512$ different possible configurations.

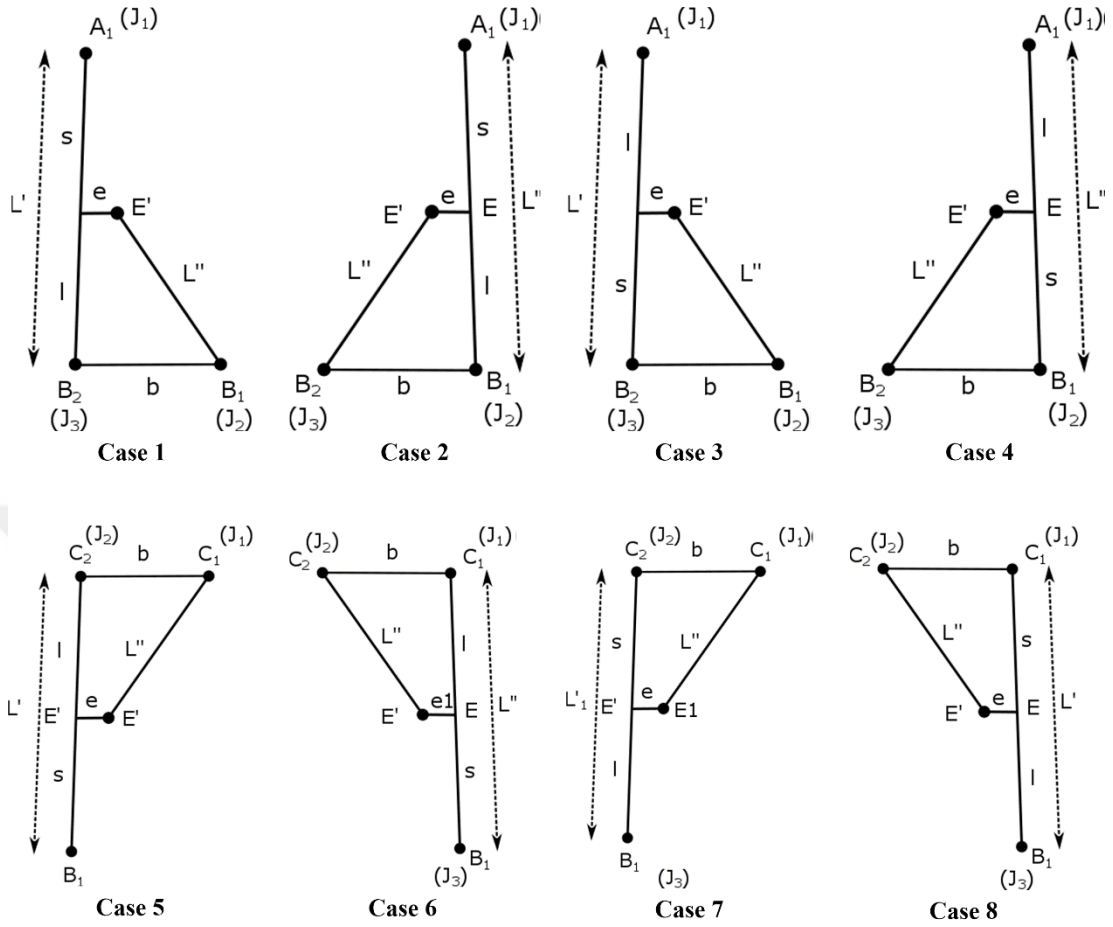


Figure 4.9. Eight possible connection types for lambda module

4.2.2. Inverse Kinematics

Similar to the SP based fixator system, inverse kinematics of lambda fixator involves finding transformations between the coordinate systems defined on ring centers and bone fragments and hence calculating the rod lengths.

The transformations allow the coordinates of a connection point on the distal ring, $J_d(u_d, v_d, 0)$, to be written in terms of the components of the global coordinate system as (x_d, y_d, z_d) . Since a connection point on the proximal ring, $J_p(x_p, y_p, 0)$, is already defined on global coordinate system, the length between two points, L' , is the magnitude of the vector

$$\vec{L}' = \begin{bmatrix} x_d - x_p \\ y_d - y_p \\ z_d - 0 \end{bmatrix} \quad (4.5)$$

In order to find the length of short rod, L'' , four vectors are calculated with simple vector algebra. Three of these vectors ($\vec{l}, \vec{s}, \vec{e}$) are along the l , s , and e components of the module. The other vector ($\vec{b}$) is between the connection point of the short rod to the ring, and the connection point of the long rod to the same ring the short rod is connected to. Depending on the configuration of the module, L'' is calculated with one of the equations below:

$$L'' = |\vec{b} + \vec{s} - \vec{e}| \quad (4.6)$$

or

$$L'' = |\vec{b} + \vec{l} - \vec{e}| \quad (4.7)$$

4.2.3. Direct Kinematics

Short rod of a lambda module can be connected to either proximal or distal ring. Therefore, in a lambda fixator with three lambda modules, three, two, one or none of the short rods may be connected to proximal ring. This results in presence of four main types of lambda fixator types. When none of the short rods is connected to the proximal ring, there are three and six connection points at proximal and distal rings, respectively. Thus, this connection is called (3-6) type (Figure 4.10a). Similarly, other connection types are named as (6-3), (5-4) and (4-5) using the number of connection points at proximal and distal rings. (Figure 4.10)

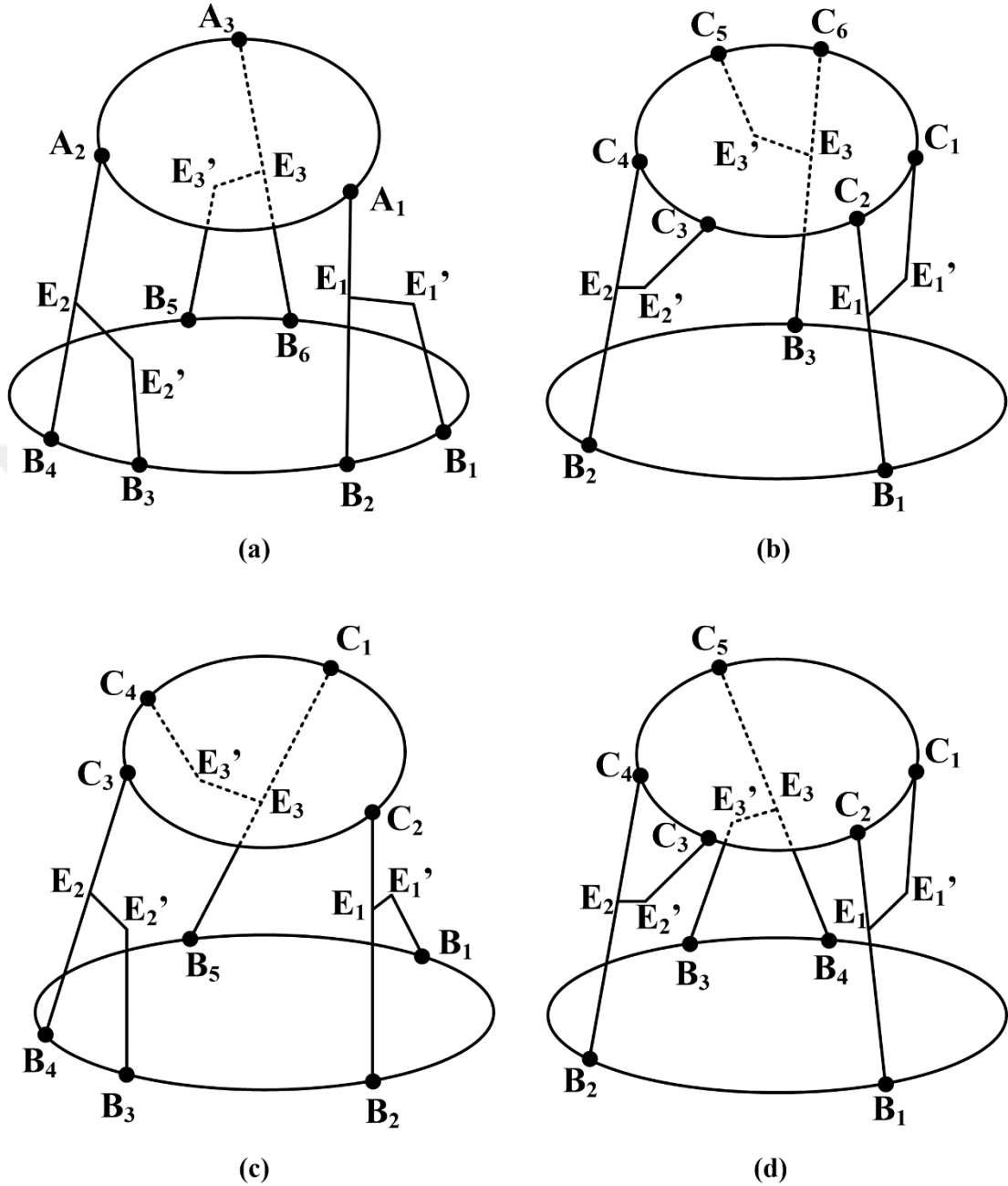


Figure 4.10. Possible lambda fixator configuration types: (3-6)(a), (6-3)(b), (4-5)(c), (5-4)(d).

Even though the inverse kinematic calculations are generalized for all lambda fixator types, this does not hold for direct kinematics. But the idea behind the calculations for each type is based on transforming the geometric shape, formed by the rod connection points on the rings (pentagon, hexagon etc.), into a triangle. Details of the direct kinematic calculations can be found in (Akçali, 2016).



5. GRAPHICAL USER INTERFACE

Two different GUIs for SP and lambda fixators, employing the related mathematical theories, introduced in Chapter 4 are presented in this chapter.

5.1. GUI for SP-type External Fixator

The main window of the GUI for SP fixator involves 38 text boxes for various parameters (Figure 5.1).

Figure 5.1. Main window of the GUI for SP-type external fixator

The data under the titles *AP Data*, *L Data*, *q Data*, *r Data* and *e Data* are obtained through image processing and describe the relative position of the bone fragments with respect to the rings. From AP-view and L-view X-ray images, eight

$(\Psi_{AP}, \beta_{AP}, \beta'_{AP}, c_{AP}, b_{AP}, r_x, e_x, q_x)$ and eleven $(\Psi_L, \beta_L, \beta'_L, c_L, b_L, r_y, r_z, e_y, e_z, q_y, q_z)$ parameters are acquired, respectively.

Parameters under the title *Ring Parameters* describe the exact configuration of the fixator frame. In particular, the connection points of the rods to the proximal (α_{1-6}) and distal (ε_{1-6}) rings are expressed in terms of angle values under the title *Angles* which involves twelve text boxes. The radii of the *Proximal* (R) and *Distal* (R_1) rings are input under the *Radius* title.

Rotation Parameters describe relative rotational position of the bone fragments (δ_{AX}) and the rings (δ_0) in the initial configuration. It should be noted that both of these rotation parameters are determined by the clinician based on experience.

Between the initial and final configurations, the GUI calculates as many intermediate configurations as prescribed by the parameter *Number of Treatment Steps*.

The threshold value for the iterations in direct kinematics is determined by the parameter *stopping criteria*. Some of the computational results are saved to a text file whose name is specified by *file name*.

Actuation Functions for Rods enables the user to select a certain functional form from the drop-down list and hence makes it possible to change the rod lengths in different increments during different stages of the treatment (as described in Section 4.1.2.1).

Additionally, for the previously determined data to be entered to text-boxes conveniently, one more drop-down list (*Choose Problem No*) is present.

By default, actual fragment lengths are calculated using *L Data*. For the cases when the bone fragments are partially or fully blocked by the fixator frame (i.e. if it is not possible to gather relevant data from L-view X-ray accurately), the checkbox above the *Start* button is clicked to make this calculation using *AP data*.

The *Start* button initiates the computation. Parameters regarding the translations and rotations of distal ring, $(x_1, y_1, z_1, \theta_x, \theta_y, \theta_z)$, at every step and result

of singularity analysis (see section 4.1.2.2) are reported under the *Output* title of the main window.

The numerical values of all calculated quantities can be reached through “*Treatment*” and “*Checking X-Ray Data*” tabs. The simulation is initialized using the button under “*Simulation*” tab.

5.1.1. The *Treatment* Tab

The *Treatment* tab contains the buttons allowing the user to display computationally significant results. Although some of the results are not relevant for the end user (e.g. clinicians), they are of high importance for developers to ensure that these calculations are performed correctly.

There are seven buttons available under this tab (Figure 5.2). *Instant Displacement Values* button shows the translations and rotations of the distal ring, $(x_1, y_1, z_1, \theta_x, \theta_y, \theta_z)$ relative to the proximal ring for each treatment step. These data are also saved to a text file for possible needs in future and to be used for system simulation.

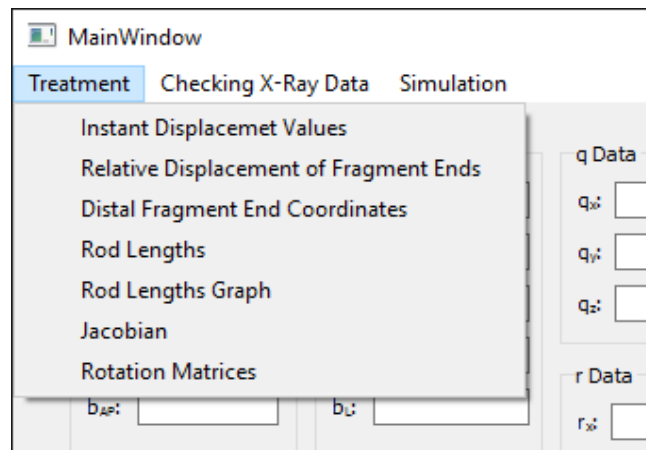


Figure 5.2. The *Treatment* tab

Relative Displacement of Fragment Ends button shows the vectors going from the distal fragment end to the proximal fragment end. A zero vector at the final step is an indicator for perfect alignment.

Distal Fragment End Coordinates button shows the coordinates of distal fragment end at each treatment step. In practice, a large distance covered by the distal fragment end between two consecutive steps may harmfully strain nerve fibers and blood vessels in the surrounding soft tissue. Hence, it should be checked if this covered distance is within acceptable limits or not. Also, in bone lengthening, the bone gap should be left at a small enough rate to allow callus formation (usually 1 mm/day for adults and 2 mm/day for children). For these reasons, this distance value is shown in this window as well. If any of the distance values observed throughout the proposed treatment is too large, the user may simply replan the treatment by increasing the number of treatment steps.

With *Rod Lengths* and *Rod Lengths Graph* options, the rod lengths at each treatment step are displayed and a corresponding rod length graph as a function of treatment step is sketched.

Evolution of rod lengths constitutes the basis for the treatment recipe, which is applied either by the clinician or by the patient himself/herself. The functional forms of these length values are determined by selecting an actuation function for each rod in the drop-down menu titled *Actuation Functions for Rods* (Figure 5.3). This feature may be relevant, if the treatment requires, for instance, relatively high step sizes at the early treatment stages.

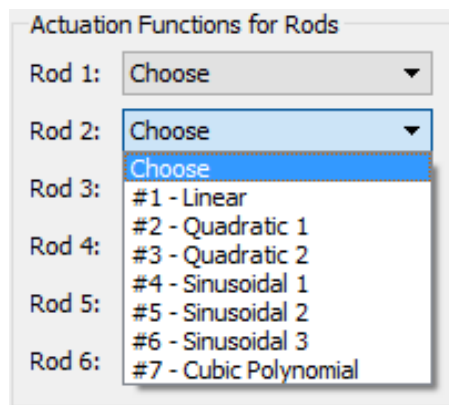


Figure 5.3. Menu for actuation functions

The Jacobian determinant at each treatment step is displayed as a graph when *Jacobian* button is clicked. If a sign change in the Jacobian value is observed, the fixator is said to encounter a singular configuration and information regarding the singularity status is displayed. With the help of this feature, the user can avoid singularities before the treatment process begins.

Rotation Matrices button shows six 3×3 rotation matrices that define the transformation between the defined coordinate systems (xyz , $x'y'z'$, uvw , $u'v'w'$) (Figure 4.1). Indices “0” and “n” denote the initial and final configurations, respectively.

5.1.2. The *Checking X-Ray Data* Tab-

Some geometric parameters in the fixator system are redundant, such that the same calculations in the inverse kinematics can be performed with a slightly different parameter set. The reliability of the results can be checked by performing the same calculations using these two differing parameter sets and see if both results agree. The only button under the *Checking X-Ray Data* tab (Figure 5.4) is *Re-Calculation of Image Values* which performs this recalculation. This type of reliability check is important for software packages utilized in computer-assisted medicine.

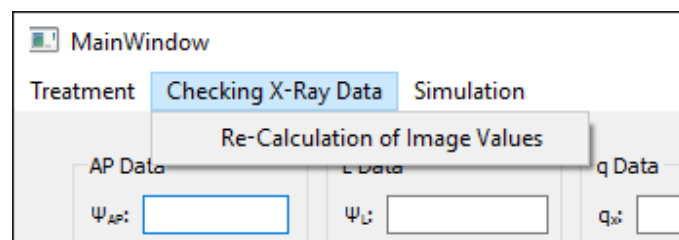


Figure 5.4. The *Checking X-ray Data* tab

5.1.3. The *Simulation* Tab

A simulation window displaying the 3D bone-fixator model is opened via *Simulation* tab of the GUI (Figure 5.5). Upon activation of the simulation window, animation of the fixation process is initialized automatically. Rod lengths, and hence

the bone-fixator system configuration, evolve according to the computed data. During the simulation, the position of the distal fragment end is indicated with a red dot at each intermediate treatment stage to allow the user to visualize the treatment course.

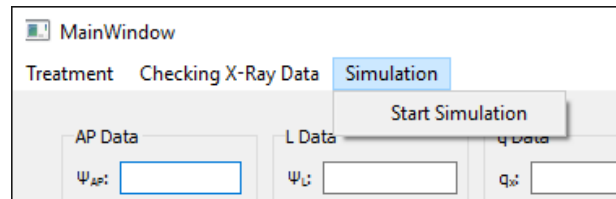


Figure 5.5. The *Simulation* tab

5.2. GUI for Lambda Fixator

The GUI developed for lambda fixator was already introduced in a previous work (Aydin, 2015). In this work, some features have been added to this existing GUI with the addition of the tabs *Simulation* and *Treatment*. Similar to the GUI for SP fixator, the former one is related with simulating and visualizing the fixator system, while the later include some functionalities related with direct kinematic calculations of the fixator system (Figure 5.6).

Figure 5.6. Main window of the GUI for lambda fixator

Most of the input fields of this GUI are also similar to those on the GUI for SP-type external fixator and described in Section 5.1. The only difference is the *Lambda Constraints* field which involves the constraints specific to lambda fixator; namely, constant lengths associated with each module (s and e), angular positions of the connection points ($J1$, $J2$, $J3$) and a drop-down list to identify the connection type of each module as listed in Figure 4.9.

After all input fields are filled appropriately, the calculations are initialized by pressing *CALC* button. Then, the results of inverse kinematic calculations are displayed on the main window and further data can be reached through the menu on top.

In Chapter 7, usage of both GUIs and the capabilities of the visualization tool are demonstrated via various examples.

5.3. Computer Implementation

The coding of the program is accomplished with Python Programming Language (version 2.7.6) in Windows environment. The Python binding of the Qt application framework, namely PyQt4 (version 4.9.6), is used to program the corresponding GUI. The simulation tool is implemented using the VPython library (version 6.05) that includes a 3D graphics module. GUI with the visualization tool use approximately 180 MB of random access memory (RAM) on a Windows 10 Home operating system.



6. BONE MODELING EXAMPLES

In this chapter, the performance of patient-specific bone model generation method (as described in Chapter 3) is evaluated.

The proposed algorithms for the creation of the actual bone model from the canonical model have been tested on 14 tibia and five femur image sets obtained from Orthopaedics and Traumatology Department, Faculty of Medicine, Cukurova University. Nine of the tibia and two of the femur images are deformed bone images and the rest are undeformed bone images.

The canonical bone models have 1138 vertices for femur and 1135 vertices for tibia.

6.1. Examples of Undeformed Bone Modeling

In case of undeformed bone modeling, the canonical models are manipulated either with linear mapping or FFD (as described in Sections 3.2.1 and 3.2.2). For linear mapping, the image of the canonical bone has been divided into 40 slices in axial direction. Numerical experimentation has shown that the transverse scaling results to a better match with the actual bone when d_1 and d_2 are approximately equal to 40% of the bone length. For FFD approach, the model is placed in a $3 \times 21 \times 3$ (along x , y and z directions) grid of control points. The related thickness values are calculated with d_1 and d_2 values equal to 5% of the bone length.

Samples of superposed projections of the actual and transformed canonical bone models for tibia and femur in AP and L directions are given in Figures 6.1, 6.2, 6.3 and 6.4. For all undeformed bone images analyzed, the error done in matching the canonical bone model to the actual bone in both AP and L views are listed in Tables 6.1 and 6.2.

In general, for undeformed bones, the average error is lower in the AP image compared to the L image for both types of bones with both types of methods (linear mapping and FFD). The highest normalized error occurs in a femur sample and is equal to 0.045 and 0.108 in the AP and L views, respectively. In general, the method

performs better with tibia samples. It is expected that the operation on femur images is more error prone due to the relatively complex geometry of this bone. Yet, the analysis reveals that normalized error is relatively low for both types of bone indicating the appropriateness of the proposed algorithm. It is also notable that there is not a significant difference in the performances of linear mapping and FFD.

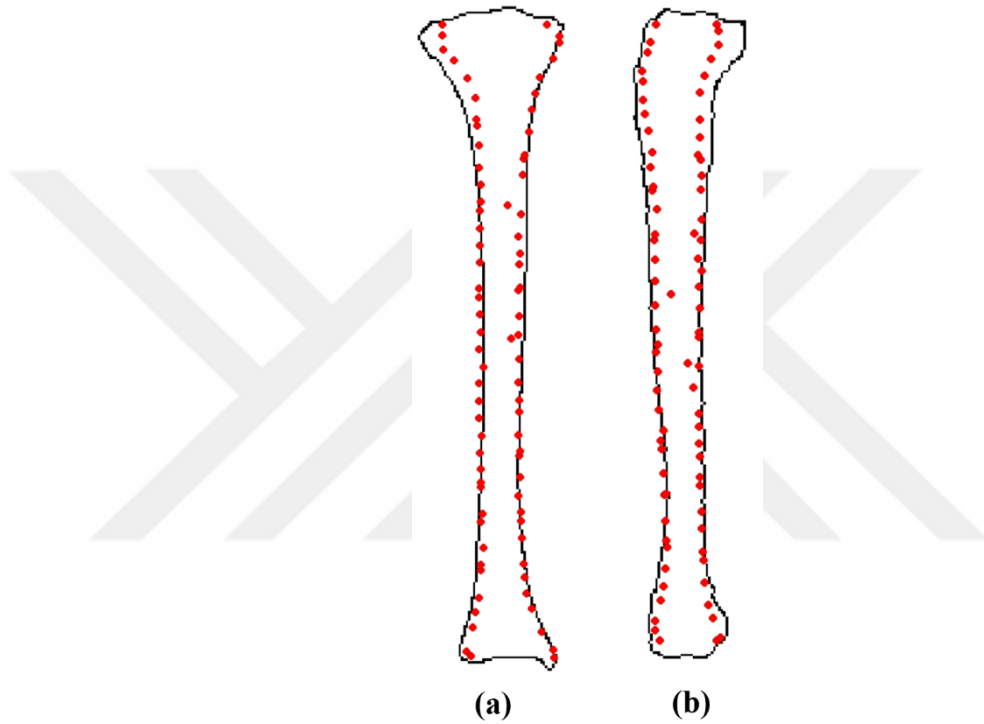


Figure 6.1. Superposed AP (a) and L (b) images of actual and canonical bone projections of Tibia #1 (Linear mapping)

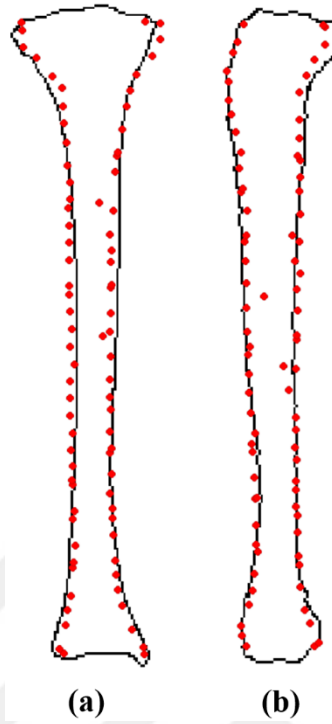


Figure 6.2. Superposed AP (a) and L (b) images of actual and canonical bone projections of Tibia #1 (FFD)

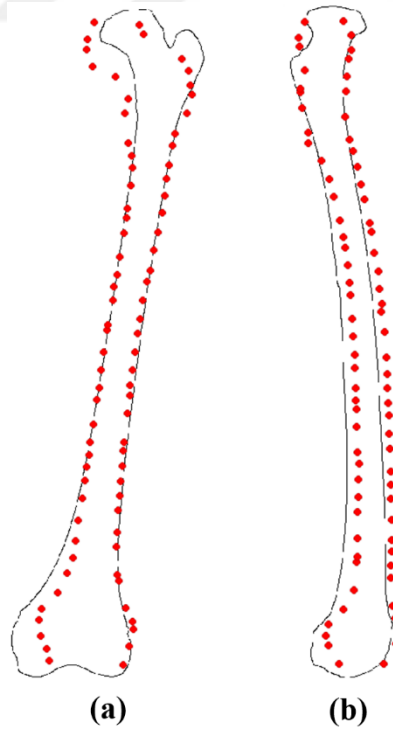


Figure 6.3. Superposed AP (a) and L (b) images of actual and canonical bone projections of Femur #2 (Linear mapping)

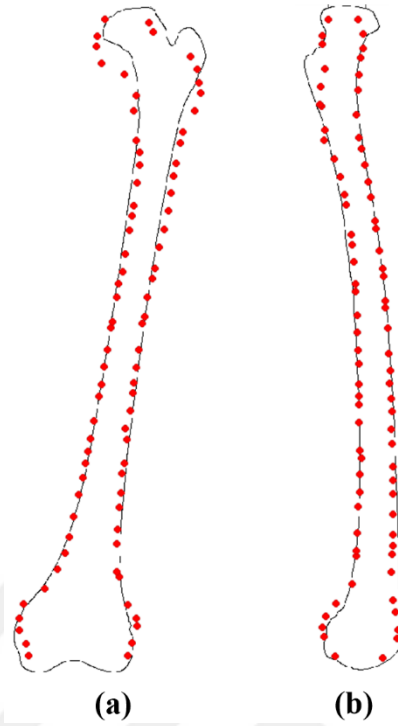


Figure 6.4. Superposed AP (a) and L (b) images of actual and canonical bone projections of Femur #2 (FFD)

Table 6.1. Average and normalized errors calculated for the generated patient-specific undeformed bone models (Linear mapping).

Samples	AP image		L image	
	E_{ave} (pixels)	E_{norm}	E_{ave} (pixels)	E_{norm}
Femur #1	3.613	0.030	8.285	0.089
Femur #2	7.258	0.045	11.081	0.108
Femur #3	2.441	0.027	4.377	0.074
Tibia #1	1.678	0.028	2.277	0.055
Tibia #2	2.206	0.016	5.227	0.055
Tibia #3	3.129	0.034	2.622	0.038
Tibia #4	3.697	0.019	7.842	0.052
Tibia #5	4.346	0.026	11.836	0.077

Table 6.2. Average and normalized errors calculated for the generated patient-specific undeformed bone models (FFD)

Samples	AP image		L image	
	E_{ave} (pixels)	E_{norm}	E_{ave} (pixels)	E_{norm}
Femur #1	6.569	0.054	6.753	0.073
Femur #2	6.306	0.039	3.535	0.035
Femur #3	2.584	0.029	5.978	0.101
Tibia #1	1.733	0.029	1.458	0.035
Tibia #2	2.218	0.017	3.070	0.032
Tibia #3	2.185	0.024	1.775	0.026
Tibia #4	3.293	0.017	3.157	0.021
Tibia #5	5.379	0.032	8.210	0.054

6.2. Examples of Deformed Bone Modeling

Before moving to processing for deformed bones, an exception related with canonical femur models should be explained here. The L view of a healthy femur bone is not straight like tibia (Figure 6.5). In this view, a healthy femur has an angle of about 10° between the proximal and distal mid-diaphyseal lines (Paley, 2005). Since the proposed algorithm is based on a straight bone axis, this angulation in the sagittal plane should be fixed before the deformed femur model can be created. While other means are possible, we have straightened the canonical femur model using an online tool for simplicity (Figure 6.6).

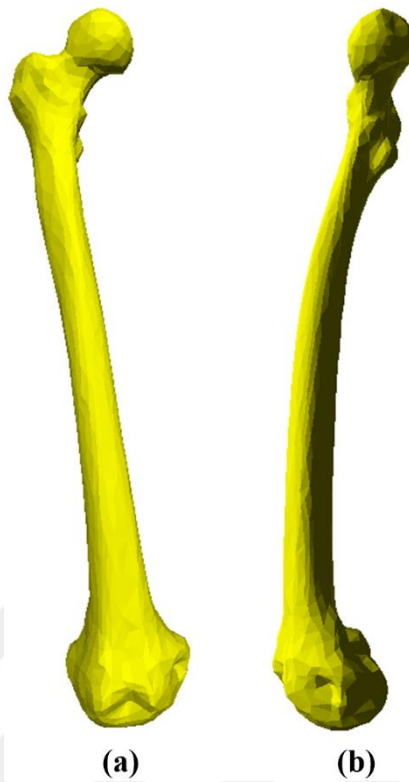


Figure 6.5. AP (a) and L (b) views of a healthy canonical femur model



Figure 6.6. AP (a) and L (b) views of a preprocessed canonical femur model

The FFD grid, utilized to deform the bone models according to fitted curve, has $3 \times 21 \times 3$ (along x , y and z directions) control points.

Samples of superposed projections of the actual and deformed canonical bone models for tibia and femur in AP and L directions are given in Figures 6.7, 6.8, 6.9 and 6.10. For all deformed bone images analyzed, the error done in matching the canonical bone model to the actual bone in both AP and L views are listed in Tables 6.3 and 6.4.

For deformed bones, the average error is lower in the AP image for femur and L image for tibia with both types of methods (polynomial and cubic spline fit). Due to its complex shape, the deformation algorithm performs not as well in femur compared to tibia. The error in both AP and L views of tibia samples is generally low (less than 10% of the maximum width). This shows that the proposed deformation algorithm is appropriate for tibia modeling while it may require some further improvement in femur modeling.

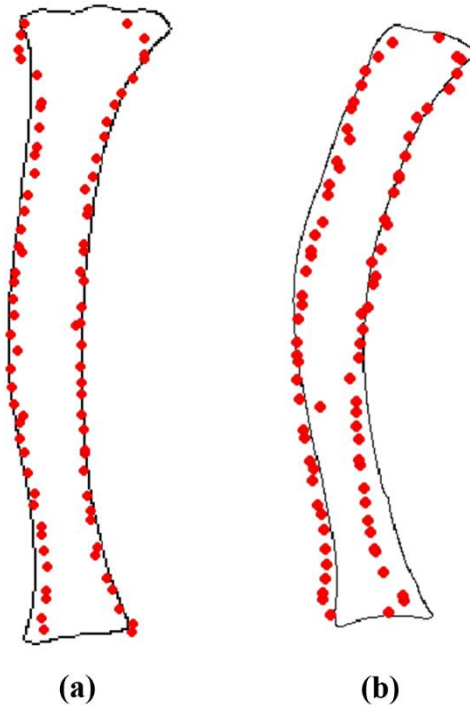


Figure 6.7. Superposed AP (a) and L (b) images of actual and canonical bone projections of Tibia #6 (Fourth order polynomial fit)

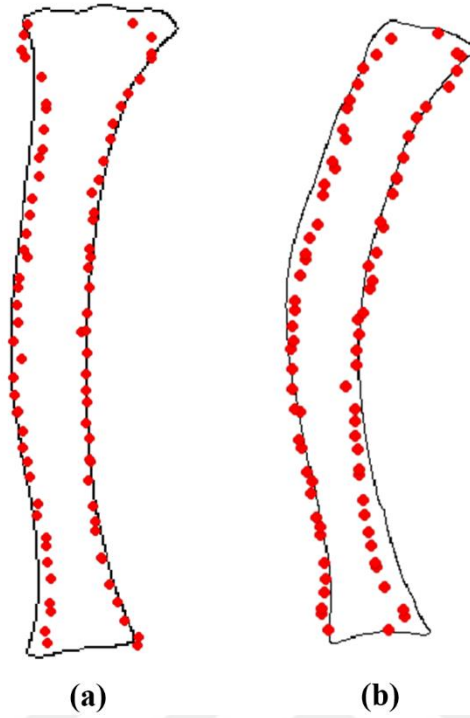


Figure 6.8. Superposed AP (a) and L (b) images of actual and canonical bone projections of Tibia #6 (Cubic splines fit)

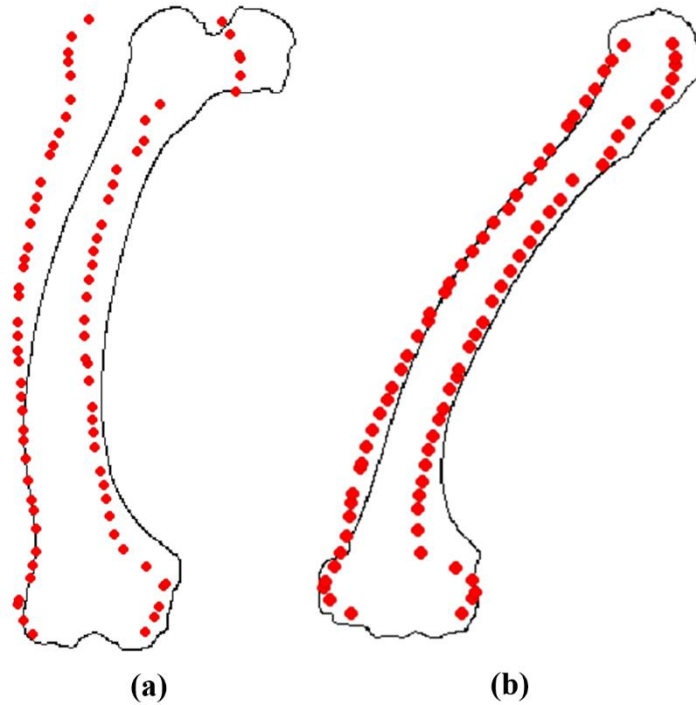


Figure 6.9. Superposed AP (a) and L (b) images of actual and canonical bone projections of Femur #4 (Fourth order polynomial fit)

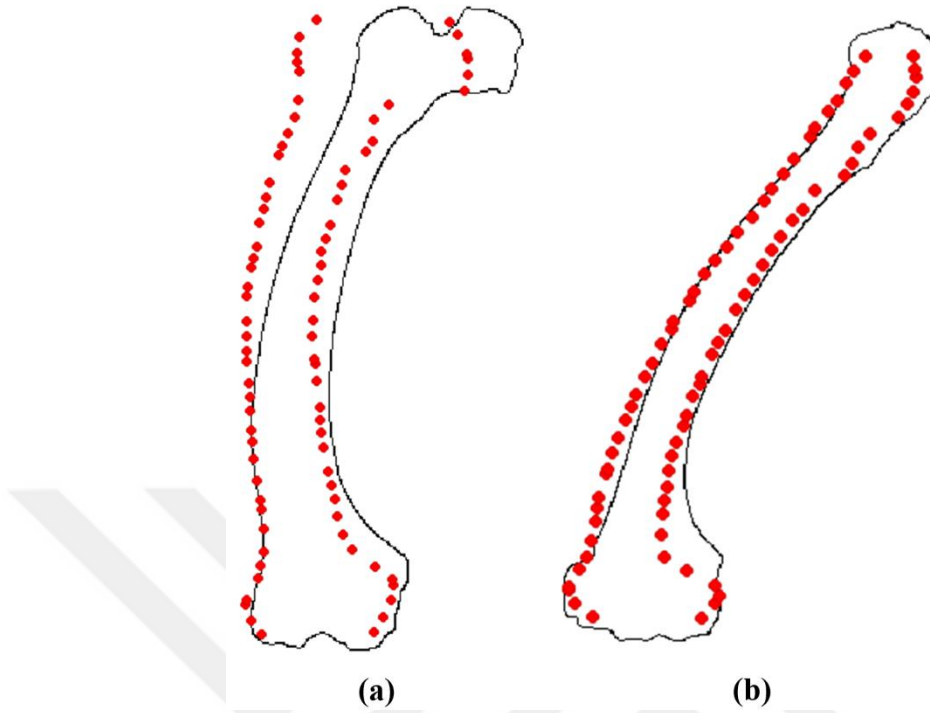


Figure 6.10. Superposed AP (a) and L (b) images of actual and canonical bone projections of Femur #4 (Cubic splines fit)

Table 6.3. Average and normalized errors calculated for the generated patient-specific deformed bone models (fourth order polynomial fit)

Samples	AP image		L image	
	E_{ave} (pixels)	E_{norm}	E_{ave} (pixels)	E_{norm}
Femur #4	13.004	0.137	6.030	0.065
Femur #5	11.664	0.134	5.864	0.066
Tibia #6	2.266	0.033	6.701	0.089
Tibia #7	3.104	0.045	6.857	0.089
Tibia #8	2.729	0.032	3.295	0.058
Tibia #9	2.730	0.033	3.697	0.066
Tibia #10	7.263	0.043	11.802	0.111
Tibia #11	8.378	0.097	5.602	0.067
Tibia #12	8.621	0.074	9.874	0.098
Tibia #13	9.068	0.075	11.576	0.118
Tibia #14	5.061	0.050	5.263	0.082

Table 6.4. Average and normalized errors calculated for the generated patient-specific deformed bone models (Cubic splines fit)

Samples	AP image		L image	
	E_{ave} (pixels)	E_{norm}	E_{ave} (pixels)	E_{norm}
Femur #4	12.803	0.134	5.884	0.064
Femur #5	10.968	0.126	5.751	0.064
Tibia #6	2.346	0.034	5.683	0.075
Tibia #7	3.408	0.049	5.880	0.076
Tibia #8	2.909	0.035	3.295	0.058
Tibia #9	2.822	0.034	3.696	0.066
Tibia #10	8.701	0.051	10.940	0.103
Tibia #11	6.675	0.077	4.775	0.057
Tibia #12	10.691	0.091	15.324	0.151
Tibia #13	8.019	0.067	11.621	0.118
Tibia #14	5.196	0.051	4.989	0.078

7. DEMONSTRATION OF THE GUI

In this chapter, the capabilities of the GUIs, described in Chapter 5, are presented with four different examples.

7.1. Example 1: Femur Fracture Fixation with SP-type Fixator

In this example, capabilities of the GUI for SP-type external fixator are presented by using artificially generated input data for a femur fracture fixation procedure. The motivation behind using artificial data is to be able to check the computational results more efficiently and provide a visual example that is more illustrative.

The analysis is started after all the input fields are filled properly on the main window of GUI (Figure 7.1). Results are displayed for the desired quantity by making the appropriate choice from the *Treatment* tab, which may be instant displacement values (Figure 7.2), relative displacement of fragment ends (Figure 7.3), distal fragment end coordinates (Figure 7.4), rod lengths (Figure 7.5), rod lengths graphs (Figure 7.6), determinant of Jacobian graph (Figure 7.7), and rotation matrices (Figure 7.8).

Using the *Simulation* tab, 3D model of the bone-fixator system is generated according to input data (Figure 7.9a) and fixation procedure is simulated on this model (Figure 7.9b).

For distinguishability, a coloring scheme is used for different elements of the bone-fixator system; in particular, red for rods, white for distal ring, yellow for proximal ring, orange for distal fragment, cyan for proximal fragment. A small red sphere is added to the position of distal fragment end at each treatment step to highlight its trace during the treatment (Figure 7.9b). It is possible to change the form of this trace by changing actuation functions for rods in main window. For example, all actuation functions may be selected as *Quadratic-1* to have smaller steps at the beginning of treatment (Figure 7.10). Similarly, when the actuation functions are set

to *Sinusoidal-2*, the rate of change in rod lengths is smaller in midstages of the treatment compared to beginning and end stages (Figure 7.11).

MainWindow

Treatment Checking X-Ray Data Simulation

AP Data

Ψ_{AP} : 5.17768394
 β_{AP} : 6.590723
 β'_{AP} : 0.54221511
 c_{AP} : 6.52
 b_{AP} : 5.15

L Data

Ψ_L : 5.7413026
 β_L : -4.888039077
 β'_L : 1.085795599
 c_L : 6.553564941
 b_L : 5.120097763

q Data

q_0 : -0.297161
 q_1 : -0.290852
 q_2 : 0

r Data

r_0 : -0.457457
 r_1 : 0.7004543
 r_2 : 0.0414522

e Data

e_0 : 3.9218093
 e_1 : 3.638719
 e_2 : 3.404525

Actuation Functions for Rods

Rod 1: #1 - Linear
 Rod 2: #1 - Linear
 Rod 3: #1 - Linear
 Rod 4: #1 - Linear
 Rod 5: #1 - Linear
 Rod 6: #1 - Linear

Number of Treatment Steps: 10
 stopping criteria (%) 5
 File Name: or4 .txt
 or4

Ring Parameters

Angles

Proximal: α_1 : 80, α_2 : 100, α_3 : 190, α_4 : 210, α_5 : 320, α_6 : 340
 Distal: ϵ_1 : 0, ϵ_2 : 100, ϵ_3 : 130, ϵ_4 : 200, ϵ_5 : 220, ϵ_6 : 340

Radius

Proximal: R: 7.25
 Distal: R: 7.25

Rotation Parameters

δ_{AP} : -4.15
 δ_L : 41.5

☐ Calculate for AP

Output

No Singularity

x_1: 1.89409
 y_1: 0.26047
 z_1: 13.30045
 theta_x: 8.08606
 theta_y: -5.08362
 theta_z: 44.02839

*****station 7*****
 x_1: 1.4692
 y_1: -0.13104
 z_1: 12.90701
 theta_x: 8.09924
 theta_y: -5.92374
 theta_z: 44.36708

*****station 8*****
 x_1: 1.06549
 y_1: -0.49864
 z_1: 12.48818
 theta_x: 8.111
 theta_y: -6.76577
 theta_z: 44.68199

*****station 9*****
 x_1: 0.68298
 y_1: -0.84221
 z_1: 12.04562
 theta_x: 8.12671
 theta_y: -7.6128
 theta_z: 44.9727

*****station 10*****
 x_1: 0.3217
 y_1: -1.16158
 z_1: 11.58084
 theta_x: 8.15228
 theta_y: -8.46825
 theta_z: 45.23858

Start

Figure 7.1. Example 1: Main window of the GUI for SP-type external fixator with input and output values.

Instant Displacement Values						
	x1	y1	z1	theta_x	theta_y	theta_z
1	4.885	3.10881	15.01198	7.7078	0.07136	41.5
2	4.33429	2.5747	14.81614	7.82995	-0.81493	41.98154
3	3.80436	2.06437	14.58088	7.92209	-1.68669	42.43848
4	3.2953	1.57779	14.30915	7.9891	-2.54674	42.87135
5	2.80719	1.11493	14.00361	8.03569	-3.39781	43.28051
6	2.3401	0.67582	13.66664	8.06646	-4.24255	43.66617
7	1.89409	0.26047	13.30045	8.08606	-5.08362	44.02839
8	1.4692	-0.13104	12.90701	8.09924	-5.92374	44.36708
9	1.06549	-0.49864	12.48818	8.111	-6.76577	44.68199
10	0.68298	-0.84221	12.04562	8.12671	-7.6128	44.9727
11	0.3217	-1.16158	11.58084	8.15228	-8.46825	45.23858

Figure 7.2. Example 1: Instant displacement values

Relative Displacement of Fragme...			
	x	y	z
1	3.92181	3.63872	3.40453
2	3.44098	3.16415	3.26796
3	2.97473	2.71836	3.02405
4	2.52942	2.29488	2.74489
5	2.10512	1.89421	2.4331
6	1.70184	1.51685	2.09109
7	1.3196	1.16334	1.72108
8	0.95837	0.83424	1.32513
9	0.61809	0.53017	0.90514
10	0.29868	0.25182	0.46288
11	-0.0	-1e-05	2e-05

Figure 7.3. Example 1: Relative displacement of fragment ends

Distal Fragment End Coordinates				
	x	y	z	distance
1	3.67309	3.25084	8.5237	
2	3.19227	2.77628	8.38713	0.68924
3	2.72602	2.33048	8.14323	0.68964
4	2.28071	1.907	7.86407	0.67496
5	1.8564	1.50633	7.55227	0.66166
6	1.45313	1.12897	7.21026	0.64961
7	1.07089	0.77546	6.84026	0.63873
8	0.70966	0.44636	6.44431	0.62894
9	0.36938	0.14229	6.02432	0.62019
10	0.04996	-0.13605	5.58206	0.61245
11	-0.24872	-0.38788	5.11919	0.6057

Figure 7.4. Example 1: Distal fragment end coordinates

Rod Lengths						
	L1	L2	L3	L4	L5	L6
1	17.54128	15.98532	17.36504	16.58314	14.51582	17.50894
2	17.15535	15.77182	16.79773	16.08225	14.28143	17.05876
3	16.76942	15.55832	16.23041	15.58136	14.04705	16.60858
4	16.3835	15.34482	15.6631	15.08046	13.81266	16.1584
5	15.99757	15.13132	15.09579	14.57957	13.57827	15.70822
6	15.61164	14.91782	14.52847	14.07868	13.34389	15.25804
7	15.22571	14.70432	13.96116	13.57779	13.1095	14.80786
8	14.83978	14.49082	13.39384	13.0769	12.87511	14.35768
9	14.45385	14.27732	12.82653	12.57601	12.64073	13.9075
10	14.06792	14.06382	12.25922	12.07512	12.40634	13.45732
11	13.68199	13.85032	11.6919	11.57423	12.17195	13.00714

Figure 7.5. Example 1: Rod lengths

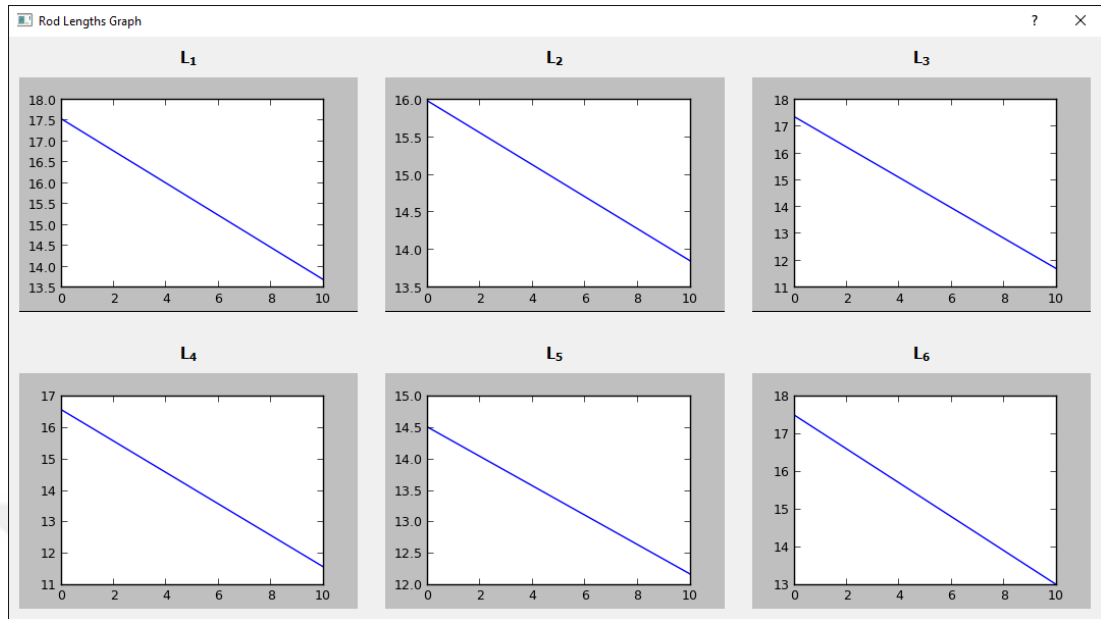


Figure 7.6. Example 1: Rod lengths graphs

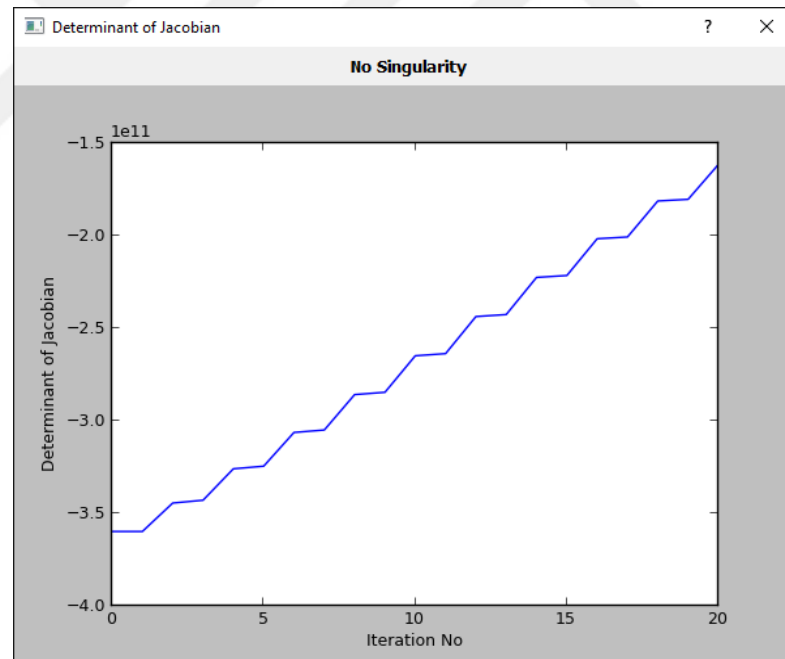


Figure 7.7. Example 1: Graph of Jacobian determinant

The screenshot shows a MATLAB-style GUI window titled "Rotation Matrices". It contains six panels, each for a 3x3 matrix. The panels are arranged in a 3x2 grid. The first panel, $[A_{uvw_xyz}]_0$, has its first cell selected. The other panels are $[A_{uvw_xyz}]_n$, $[A_{u'v'w'}_{xyz}]_0$, $[A_{u'v'w'}_{xyz}]_n$, $[A_{u'v'w'}_{uvw}]$, and $[A_{x'y'z'}_{xyz}]$.

$[A_{uvw_xyz}]_0$			$[A_{uvw_xyz}]_n$		
0.74329	-0.6641	0.08065	0.69648	-0.71757	-0.00196
0.66882	0.73513	-0.1107	0.7023	0.68221	-0.20336
0.01422	0.13622	0.99058	0.14726	0.14026	0.9791

$[A_{u'v'w'}_{xyz}]_0$			$[A_{u'v'w'}_{xyz}]_n$		
0.99312	0.05453	0.10367	0.99996	0.00018	0.00946
-0.06235	0.99532	0.07379	0.0	0.99982	-0.01895
-0.09916	-0.07974	0.99187	-0.00946	0.01895	0.99978

$[A_{u'v'w'}_{uvw}]$			$[A_{x'y'z'}_{xyz}]$		
0.69505	0.70509	0.14051	0.99996	0.00018	0.00946
-0.71887	0.68462	0.12051	0.0	0.99982	-0.01895
-0.01123	-0.18477	0.98272	-0.00946	0.01895	0.99978

Figure 7.8. Example 1: Rotation matrices

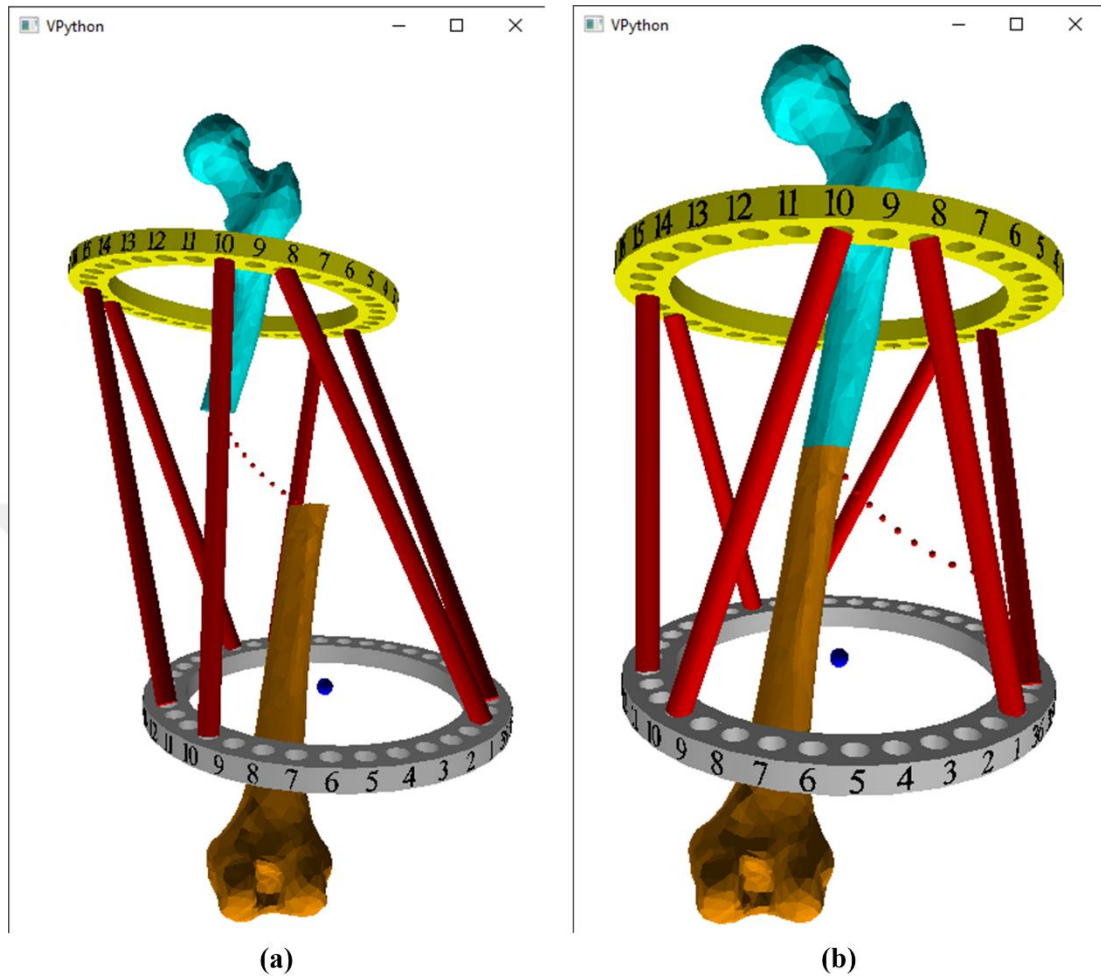


Figure 7.9. Example 1: Bone – fixator system in initial (a) and final (b) configurations

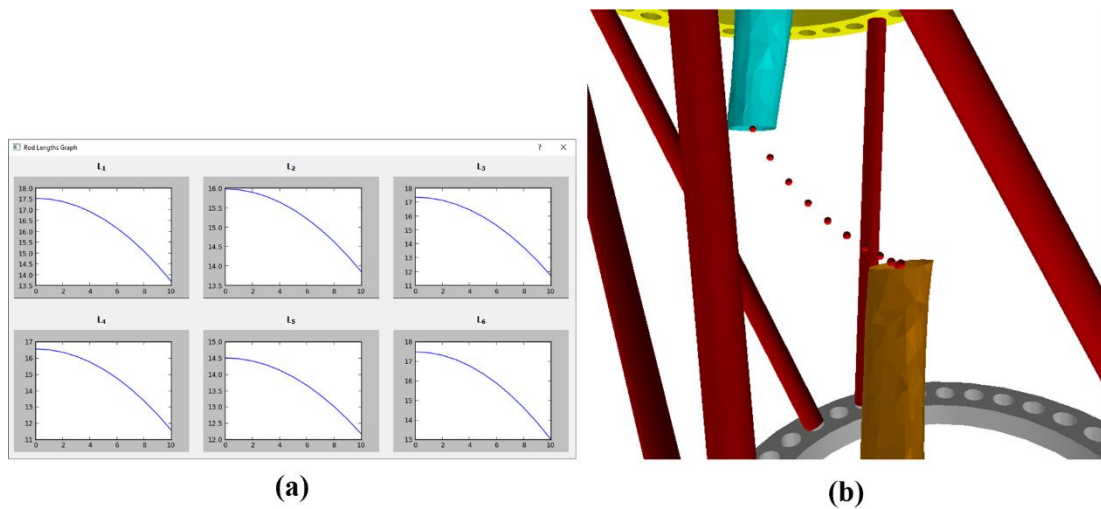


Figure 7.10. Example 1: Evolution of the rod lengths for “Quadratic-1” actuation functions (a) and the corresponding fixation path (b)

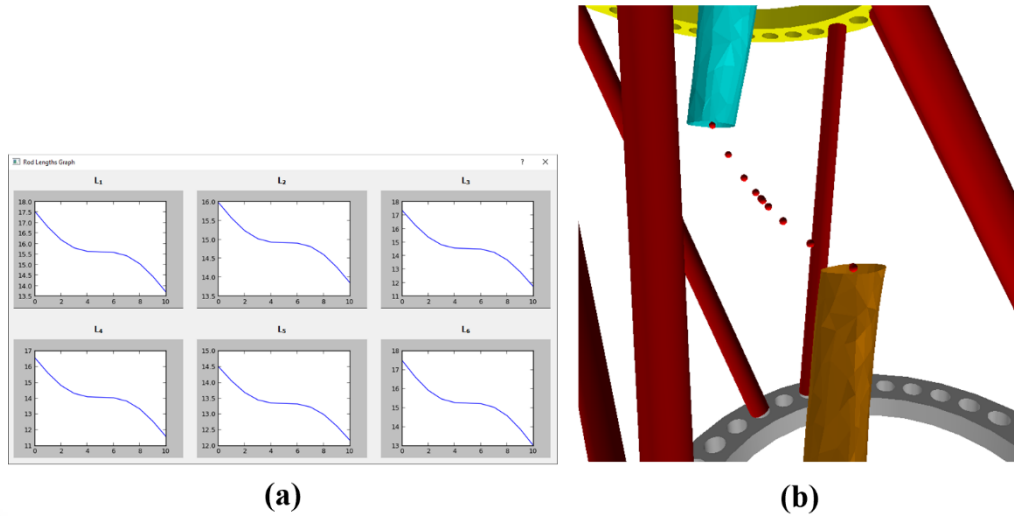


Figure 7.11. Example 1: Evolution of the rod lengths for “Sinusoidal-2” actuation functions (a) and the corresponding fixation path (b)

7.2. Example 2: Tibia Fracture Fixation with SP-type Fixator

Example 2 concerns a tibia fracture fixation process with a SP-type fixator. Input parameters of this example are obtained from an experimental setup consisting of fragments of an artificial bone attached to a SP-type external fixator (Figure 7.12). After constructing the setup, two perpendicular X-ray images are taken from AP and L views (Figure 7.13). Geometric data regarding the bone-fixator system are obtained by appropriately processing the X-ray images as explained elsewhere (Aydin, 2009; Aydin et al., 2011). Furthermore, data from the axial image are obtained by visually inspecting the system in axial direction (Figure 7.14) and clinical data are determined under the supervision of an orthopaedist. After all relevant data is entered to the GUI, these input parameters are input to the main window of the GUI (Figure 7.15), a set of results can be obtained and the simulation of the fixation process can be visualized (Figure 7.16) as shown previously in Section 7.1. When rod lengths in the experimental setup are set to those values obtained via inverse kinematic calculations, it is observed that the bone fragments are aligned correctly (Figure 7.17).



(a)



(b)

Figure 7.12. Example 2: AP (a) and L (b) view images of the bone-fixator system in the initial configuration

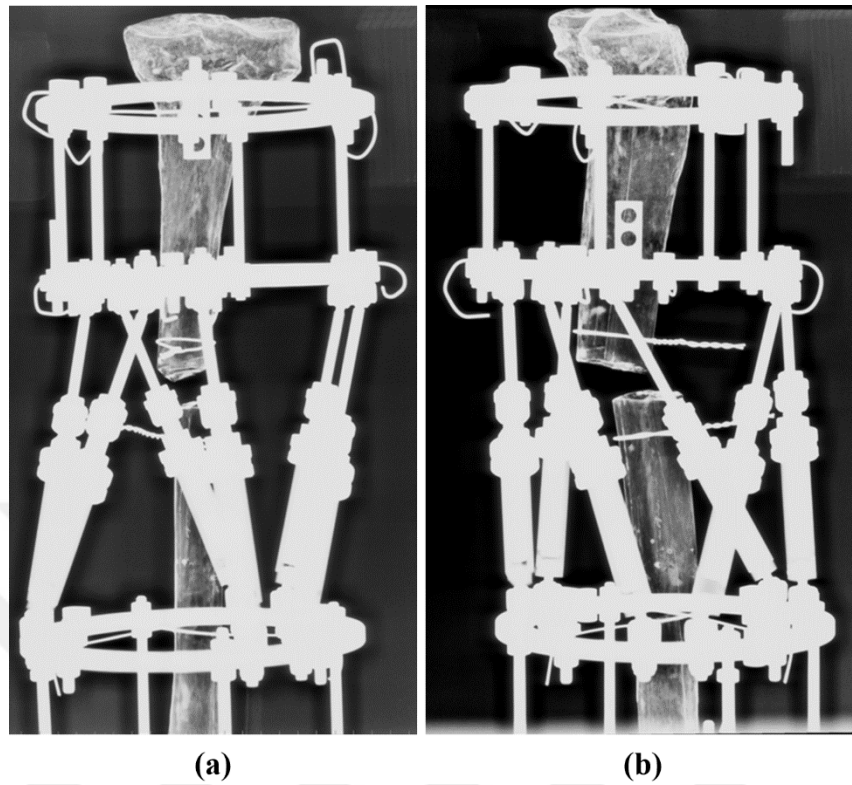


Figure 7.13. Example 2: AP (a) and L (b) view images of the bone-fixator system

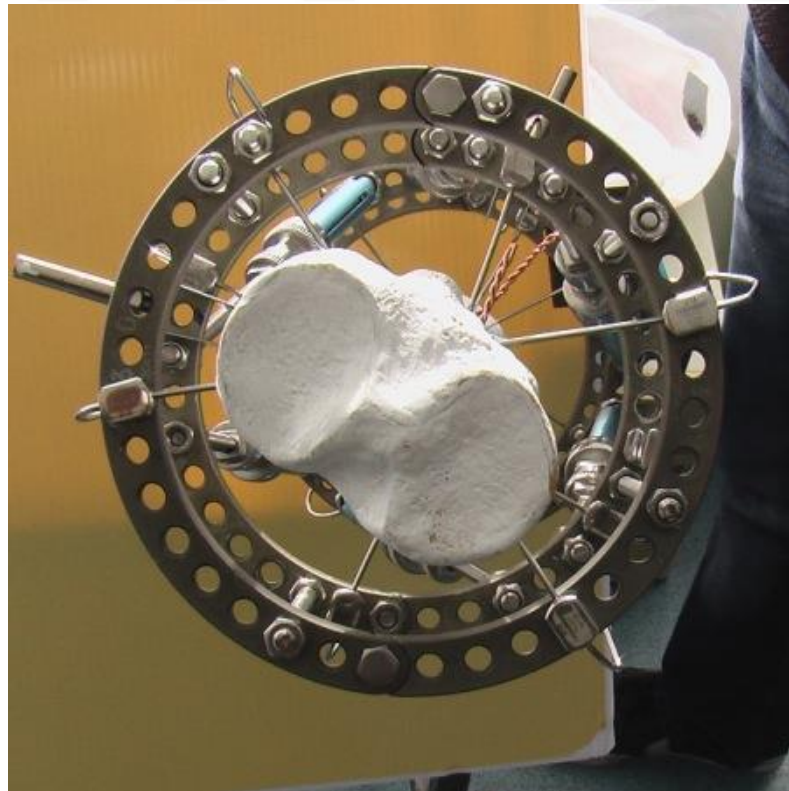


Figure 7.14. Example 2: Axial image of the bone-fixator system

MainWindow

Treatment Checking X-Ray Data Simulation

AP Data

ψ_{AP} : 0.30

β_{AP} : -2.48

β'_{AP} : -6.21

c_{AP} : 10

b_{AP} : 4

L Data

ψ_L : -0.43

β_L : -6.90

β'_L : 5.1

c_L : 9.82165

b_L : 5.4564

q Data

q_s : -1.042

q'_s : -0.678

q_z : 0

r Data

r_s : 0.2789

r'_s : 0.2653

r_z : 0.0015

e Data

e_s : 0.7927

e'_s : 1.6361

e_z : 1.47395

Actuation Functions for Rods

Rod 1: #1 - Linear

Rod 2: #1 - Linear

Rod 3: #1 - Linear

Rod 4: #1 - Linear

Rod 5: #1 - Linear

Rod 6: #1 - Linear

Ring Parameters

Angles

Proximal	Distal
α_1 : 90	ϵ_1 : 0
α_2 : 110	ϵ_2 : 100
α_3 : 210	ϵ_3 : 120
α_4 : 230	ϵ_4 : 220
α_5 : 330	ϵ_5 : 240
α_6 : 350	ϵ_6 : 340

Radius

Proximal	Distal
R : 7.25	R_s : 7.25

Rotation Parameters

δ_{AX} : 22

δ_o : 65.64

Number of Treatment Steps: 10

stopping criteria (%) 5

File Name: example2 .txt

1

☐ Calculate for AP

Start

Output

No Singularity

```

x_1: -2.49931
y_1: -0.80629
z_1: 15.87199
theta_x: 1.46384
theta_y: -7.7779
theta_z: 52.34342

*****station 7*****
x_1: -2.64307
y_1: -1.15606
z_1: 15.69701
theta_x: 2.08383
theta_y: -9.07366
theta_z: 50.21178

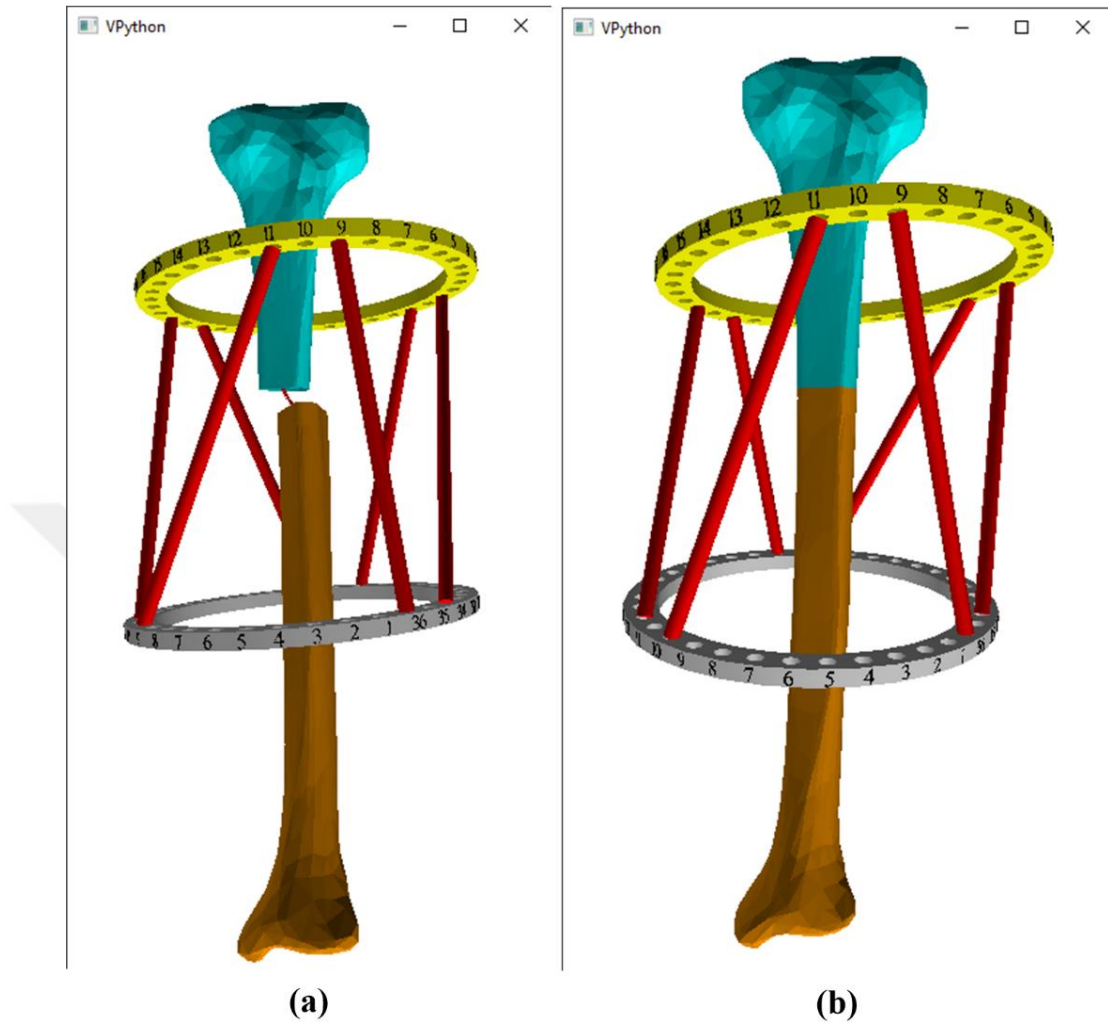
*****station 8*****
x_1: -2.78207
y_1: -1.50023
z_1: 15.51021
theta_x: 2.81024
theta_y: -10.33365
theta_z: 48.08225

*****station 9*****
x_1: -2.91622
y_1: -1.83846
z_1: 15.31188
theta_x: 3.64194
theta_y: -11.55396
theta_z: 45.9479

*****station 10*****
x_1: -3.04536
y_1: -2.17038
z_1: 15.10226
theta_x: 4.57806
theta_y: -12.73065
theta_z: 43.80222

```

Figure 7.15. Example 2: Main window of the GUI



(a) (b)
Figure 7.16. Example 2: Bone – fixator system in initial (a) and final (b) configurations

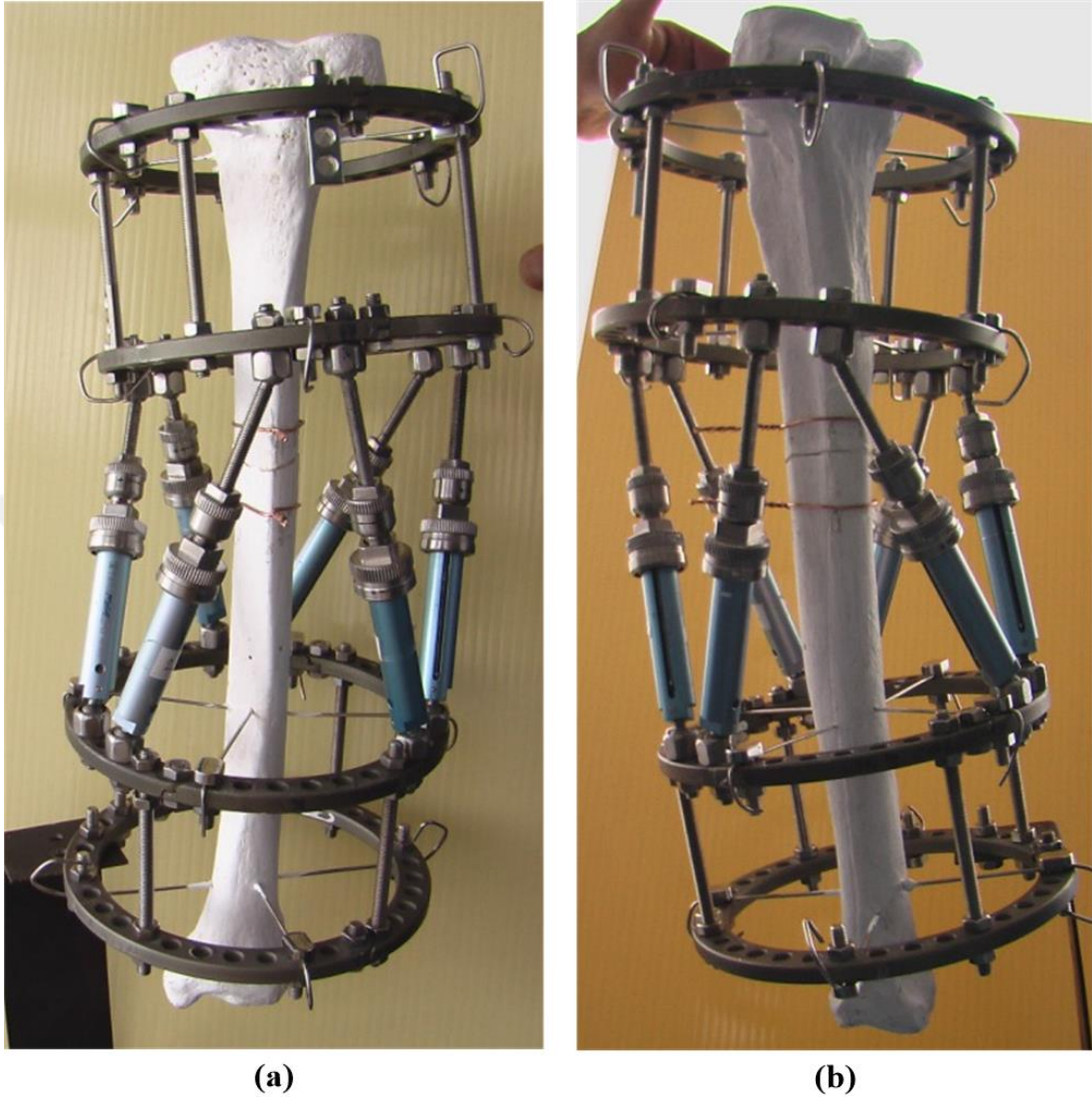


Figure 7.17. Example 2: AP (a) and L (b) view images of the bone-fixator system in the final configuration

7.3. Example 3: Deformity Correction with SP-type Fixator

The next example concerns the simulation of a deformity correction process. The utilized deformed bone model is created as explained in Section 3.3. The input data are generated artificially in such a way that proximal and distal bone fragments are anatomically aligned in final treatment step (Figure 7.18). As explained in Chapter 4, the mathematical theory does not handle the deformity correction kinematics at this point. In order to illustrate the callus formation feature of the

visualization tool, the input data and positioning of the bone fragments are adjusted manually in this example.

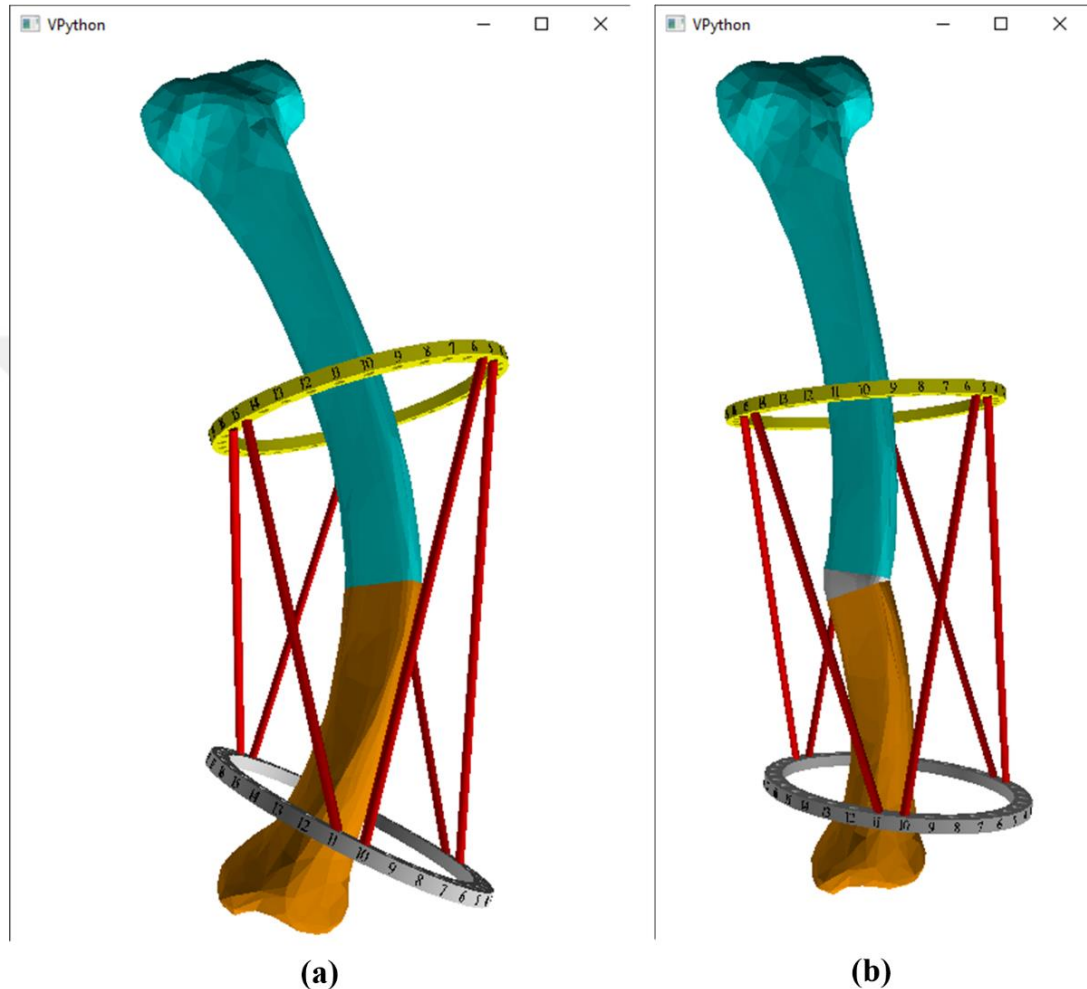


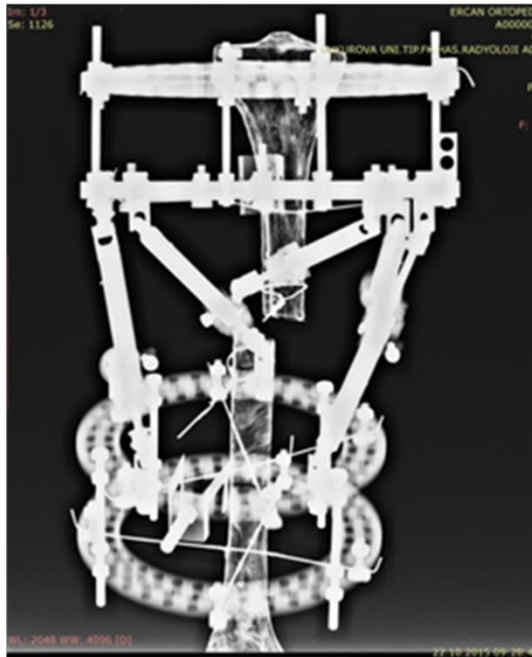
Figure 7.18. Example 3: Bone – fixator system in the initial (a) and final (b) configurations

7.4. Example 4: Tibia Fracture Fixation with Lambda Fixator

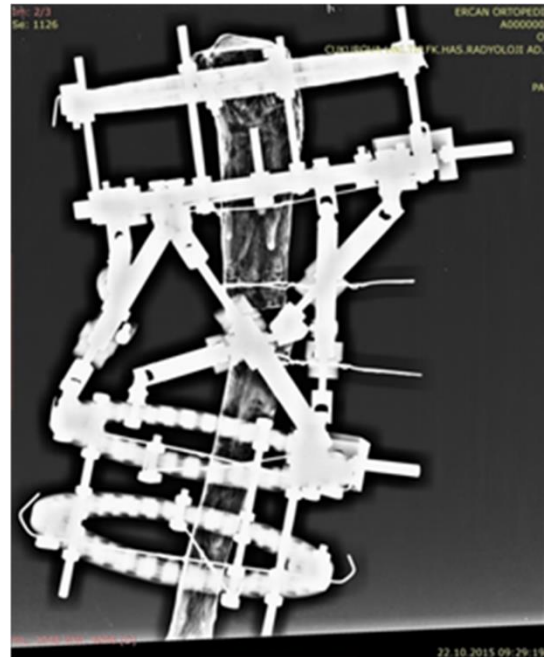
This example demonstrates a fracture fixation procedure with a (5-4) type lambda fixator. Input parameters are obtained from an experimental setup consisting of a lambda fixator and attached bone fragments (Figure 7.19). First, two perpendicular X-ray images of the bone-fixator system are taken in AP and L views (Figure 7.20) and then relevant input parameters are measured on the X-ray images as explained in (Aydin, 2015).



Figure 7.19. Example 4: L view image of the bone-fixator system in the initial configuration



(a)



(b)

Figure 7.20. Example 4: X-ray images of the bone-fixator system are taken in AP (a) and L (b) directions

The measured quantities as well as the clinical data are input to the main window of the GUI developed for the lambda fixator (Figure 7.21). Various computational results are reached through *Treatment* tab as described earlier.

The screenshot displays the 'Lambda Fixator' software interface with the 'Treatment' tab selected. The interface is organized into several sections for data entry and results display.

AP Data:

- Ψ_{AP} : -0.203979162494
- β_{AP} : 1.52409340299
- β'_{AP} : 0.126561768709
- C_{AP} : 41.1432189057
- b_{AP} : 49.5640716964

L Data:

- Ψ_L : 19.6882904498
- β_L : 21.6769711158
- β'_L : 8.47743757626
- C_L : 50.9172392751
- b_L : 51.0744124903

Theta Values:

	θ_x	θ_y	θ_z
Proxi Bone	8.477417194	0.126561768	0.0
Distal Bone	20.85794076	6.201804435	-12.0
Distal Ring	19.46782791	3.002943201	-9.0

e Data:

- e_x : 16.64404744
- e_y : 4.929147955
- e_z : 13.07764697

r Data:

- r_x : -12.02812592
- r_y : 19.070416181
- r_z : 5.8238007961

Control Parameters:

- r_u : -15.2003806432
- r_v : 17.9963231391
- $GG1_0$: 22.7350977118
- $GG1_n$: 9.28593431256

Fragment Lengths:

- c : 50.9328005621
- b : 51.0745343869

q Data:

- q_x : -7.30759758
- q_y : -3.24335447
- q_z : 0

Rotation Parameters:

- δ_{Ax} : -12
- δ_0 : -9
- δ : 0

Lambda Constants:

Modul	s	e	J1	J2	J3	Case
Modul 1	75	20	135	207	140	Case 5
Modul 2	75	20	234	324	270	Case 7
Modul 3	75	20	27	340	70	Case 1

Ring Parameters:

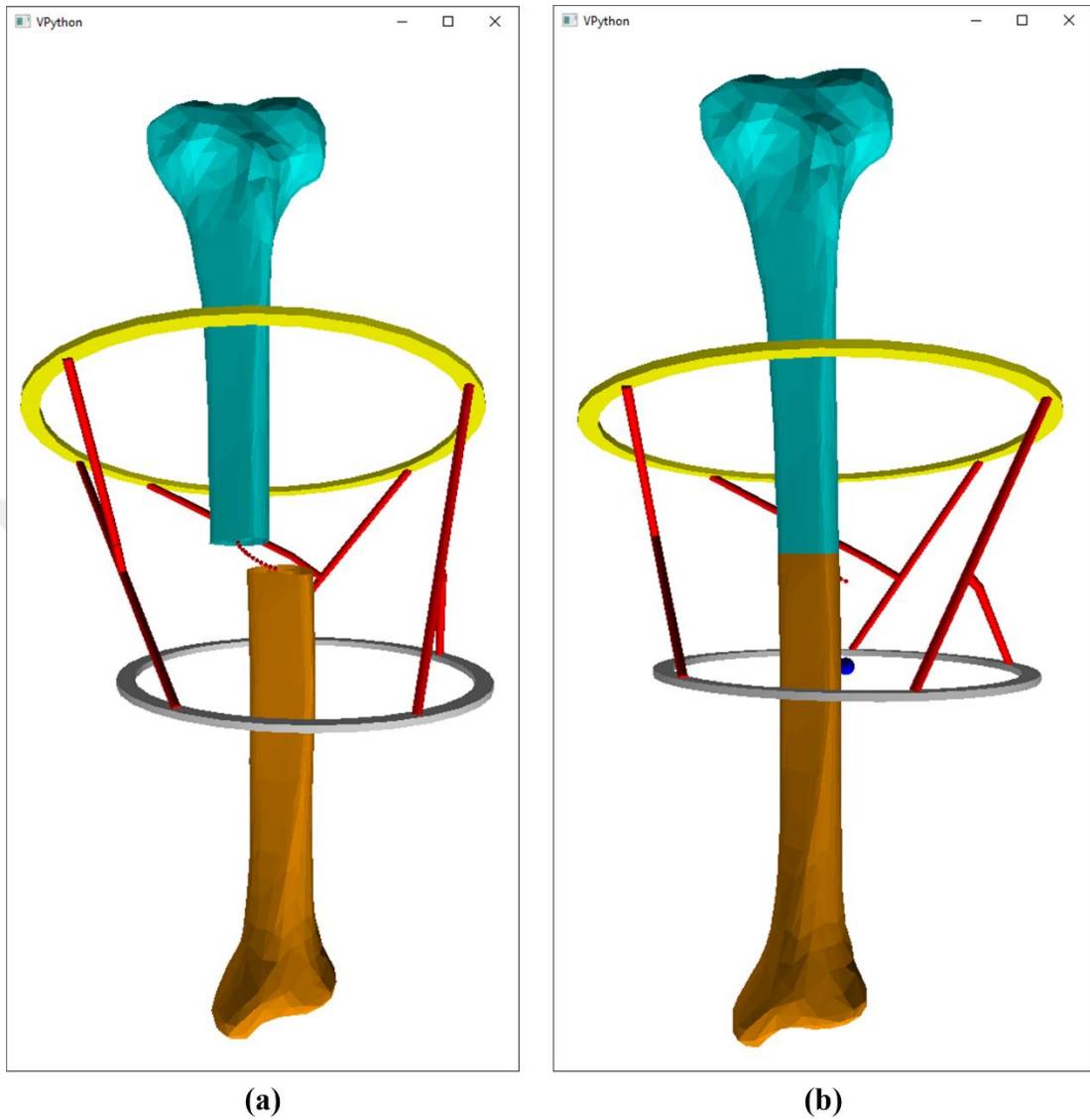
- Proximal Ring: 97.5
- Distal Ring: 82.5

Result Box:

	Initial	Final
L1'	149.439865812	121.739605845
L1''	85.2213910226	79.6681979457
L2'	121.266455873	125.25960814
L2''	86.1049978484	92.5209651639
L3'	132.135534812	122.563614069
L3''	98.8202323537	92.709167773

At the bottom, there is a dropdown menu set to '5-4 problem' and a checkbox for 'Calc for AP' which is currently unchecked. A 'CALC' button is located at the bottom right of the interface.

Figure 7.21. Example 4: Input and output values shown on the main window of the GUI for lambda fixator



(a) (b)

Figure 7.22. Example 4: Bone – lambda fixator system in the initial (a) and final (b) configurations



8. CONCLUSION

8.1. Concluding Remarks and Discussion

In this work, we have developed a simulation tool as a feature of a user friendly GUI for two different external ring fixator systems (SP-type and lambda fixators). To ensure that the simulation tool depicts the bone fixator system realistically, we have developed model generation methods for bone. To achieve patient-specific bone modeling, we have proposed separate algorithms for undeformed and deformed bones. The algorithms entirely rely on the X-rays taken in the AP and L directions of the bone and simple presurgery measurements done by the clinician on the X-ray images. The basic idea behind the algorithm is to transform the model of a healthy bone to the patient-specific bone model using the data gathered from the X-rays. This approach is found to be computationally efficient and has performed well in the several tibia and femur sample images tested.

One of the main motivations for this work has been the fact that a proper 3D simulation tool with patient-specific bone modeling capabilities does not exist in any of the existing computer-assisted systems. The simulation tool embedded to the GUI has a potential to be used clinically. Part of this work has already been published in an international journal and the comments, made by a reviewer with clinical background, highlights the importance of the work and has been encouraging for us: *“I am a clinician and not an engineer or mathematician. The methods are beyond my understanding but the idea behind this work is really great as the current systems do not fulfil the needs to be able to predict what will happen with a deformity.”*

Existing methods for bone reconstruction from X-rays rely on more complex and computationally demanding mathematical approaches (See Section 1.3). The method proposed in this work utilizes mostly a simpler set of data already-generated during surgery planning. As a result, model creation is usually a rapid computational process that is usually completed in less than a minute. In that sense, our method seems to be appropriate for instant visualization and simulation of the bone-fixator system.

Modeling the deformed femur bone seems to be more challenging due to the unusual geometry of the proximal femur. On the other hand, the proposed deformation method has performed particularly well when modeling deformed tibia bones. Nevertheless, shape and geometry can vary tremendously in deformed bones. A surgeon may define two CORAs if the deformity is too severe or some deformities may be too irregular to be categorized. In these cases, the surgeon may have to use unconventional correction practices and computer-assisted systems may be inadequate to support such operations.

8.2. Future Work

In this work, we have tried to develop an algorithmic approach to create the model of the patient-specific bone and fixator frame ,which is accomplished almost automatically within the GUI. The one step that is performed manually is the extraction of bone profile from the X-ray images with canny edge detection algorithm, in particular, the removal of false edges. In order to fully automatize the model generation procedure, new bone-specific image processing algorithms could be developed to eliminate this manual step. These algorithms may use statistical data associated with the shape of the bone as well as the pixel intensity information.

As mentioned earlier, modeling of deformed femur is more challenging and error prone. Hence, the algorithm to create deformed bones can be further improved to handle femur samples. It is also possible to develop bone-specific algorithms to model deformed femur samples.

When working on bones with diaphyseal deformities, generating the diaphysis of the bone model is clinically more important than distal and proximal ends. In that sense, it is possible to separately evaluate the error associated with different parts of the deformed bone model. This may provide a more objective measure for the performance of the modeling approach.

The mathematical theories implemented within the GUIs involve the kinematics of fracture fixation. For bone lengthening and deformity correction, the kinematic theory has to be modified and implemented within the GUIs. (Remember

that artificially generated data are used in the deformity correction example presented in Section 7.3) Once this feature is available, a full-scale simulation of deformity correction and bone lengthening can be achieved through the GUI.





REFERENCES

- ADAMS, F. (1849). The genuine works of Hippocrates, London Printed for the Sydenham society.
- AKÇALI, İ. D., (2010). "Fixatör Otomasyonu." Technical Report, TÜBİTAK, Grant No: 106M466.
- AKÇALI, İ. D., (2016). "Yeni Bir Fiksator Teknolojisi." Technical Report, TÜBİTAK, Grant No: 112M406.
- AYDIN, A., (2009). "Applications of Medical Image Processing Techniques on a Biomechanics System." MSc Thesis, Çukurova University.
- AYDIN, A., (2015). "Development of a New External Fixator System for Automated Bone Fixation: Theory and Implementation." PhD Thesis, Çukurova University.
- AYDIN, A., İBRIKÇİ, T. and AKÇALI, İ. D., (2011). "A hybrid image processing system for X-ray images of an external fixator." Computer Methods in Biomechanics and Biomedical Engineering 15(7): 753-759.
- BAKA, N., KAPTEIN, B. L., DE BRUIJNE, M., VAN WALSUM, T., GIPHART, J. E., NIESSEN, W. J. and LELIEVELDT, B. P. F., (2011). "2D-3D shape reconstruction of the distal femur from stereo X-ray imaging using statistical shape models." Medical Image Analysis 15(6): 840-850.
- BAUDOIN, A., SKALLI, W., DE GUISE, J. A. and MITTON, D., (2008). "Parametric subject-specific model for in vivo 3D reconstruction using bi-planar X-rays: application to the upper femoral extremity." Medical & Biological Engineering & Computing 46(8): 799-805.
- BIRKFELLNER, W., WIRTH, J., BURGSTALLER, W., BAUMANN, B., STAEDLE, H., HAMMER, B., GELLRICH, N. C., JACOB, A. L., REGAZZONI, P. and MESSMER, P., (2003). "A faster method for 3D/2D medical image registration - a simulation study." Physics in Medicine and Biology 48(16): 2665-2679.
- BLANC, R., SEILER, C., SZEKELY, G., NOLTE, L. P. and REYES, M., (2012). "Statistical model based shape prediction from a combination of direct

- observations and various surrogates: Application to orthopaedic research." *Medical Image Analysis* 16(6): 1156-1166.
- CANNY, J., (1986). "A Computational Approach to Edge-Detection." *Ieee Transactions on Pattern Analysis and Machine Intelligence* 8(6): 679-698.
- CAPONETTI, L. and FANELLI, A. M., (1993). "Computer-Aided Simulation for Bone Surgery." *Ieee Computer Graphics and Applications* 13(6): 86-92.
- CHAIBI, Y., CRESSON, T., AUBERT, B., HAUSSELLE, J., NEYRET, P., HAUGER, O., DE GUISE, J. A. and SKALLI, W., (2012). "Fast 3D reconstruction of the lower limb using a parametric model and statistical inferences and clinical measurements calculation from biplanar X-rays." *Computer Methods in Biomechanics and Biomedical Engineering* 15(5): 457-466.
- CRESSON, T., BRANCHAUD, D., CHAV, R., GODBOUT, B. and DE GUISE, J. A., (2010). "3D shape reconstruction of bone from two x-ray images using 2D/3D non-rigid registration based on moving least-squares deformation." *Medical Imaging 2010: Image Processing* 7623.
- DONG, X. and ZHENG, G. Y., (2006). "Determining geometrical parameters by particle filter for automatic reconstruction of surface model of proximal femur from biplanar calibrated fluoroscopic images." *2006 9th International Conference on Control, Automation, Robotics and Vision, Vols 1- 5: 1923-1928.*
- FLEUTE, M. and LAVALLEE, S., (1999). "Nonrigid 3-D/2-D registration of images using statistical models." *Medical Image Computing and Computer-Assisted Intervention, Miccai'99, Proceedings* 1679: 138-147.
- GALIBAROV, P. E., PRENDERGAST, P. J. and LENNON, A. B., (2010). "A method to reconstruct patient-specific proximal femur surface models from planar pre-operative radiographs." *Medical Engineering & Physics* 32(10): 1180-1188.
- GAMAGE, P., XIE, S. Q., DELMAS, P. and XU, W. L., (2011). "Diagnostic radiograph based 3D bone reconstruction framework: Application to the femur." *Computerized Medical Imaging and Graphics* 35(6): 427-437.

- GUNAY, M., SHIM, M. B. and SHIMADA, K., (2007). "Cost- and time-effective three-dimensional bone-shape reconstruction from X-ray images." *International Journal of Medical Robotics and Computer Assisted Surgery* 3(4): 323-335.
- KARADE, V. and RAVI, B., (2015). "3D femur model reconstruction from biplane X-ray images: a novel method based on Laplacian surface deformation." *International Journal of Computer Assisted Radiology and Surgery* 10(4): 473-485.
- KOH, K., KIM, Y. H., KIM, K. and PARK, W. M., (2011). "Reconstruction of patient-specific femurs using X-ray and sparse CT images." *Computers in Biology and Medicine* 41(7): 421-426.
- LAPORTE, S., SKALLI, W., DE GUISE, J. A., LAVASTE, F. and MITTON, D., (2003). "A biplanar reconstruction method based on 2D and 3D contours: application to the distal femur." *Comput Methods Biomech Biomed Engin* 6(1): 1-6.
- LAWRENCE A. DIDOMENICO , B. H. Z., LAURENCE ZACHARY CANE (2012). *The Use of External Fixation in the Lower Extremity. International Advances in Foot and Ankle Surgery.* D. A. Saxena, Springer London: pp 439-452.
- LORENSEN, W. E. and CLINE, H. E., (1987). "Marching cubes: A high resolution 3D surface construction algorithm." *SIGGRAPH Comput. Graph.* 21(4): 163-169.
- MAHFOUZ, M., BADAWI, A. M., FATAH, E. E. A., KUHN, M. and MERKL, B. (2006). *Reconstruction of 3D Patient-Specific Bone Models From Biplanar X-Ray Images Utilizing Morphometric Measurements.* IPCV.
- MESHLAB, (2016). "MeshLab." retrieved 30.04.2016 from <http://meshlab.sourceforge.net/>.
- MIMICS, (2016). "MIMICS." retrieved 15.06.2016 from <http://biomedical.materialise.com/mimics>.

- MUTLU, H., AKCALI, I. D. and GULSEN, M., (2006). "A mathematical model for the use of a Gough-Stewart Platform mechanism as a fixator." *Journal of Engineering Mathematics* 54(2): 119-143.
- ORTHOFIX, (2016). "Truelok Hexapod System." Retrieved 30.04.2016 from <http://web.orthofix.com/sites/Country/Intl/Products/Pages/TL-HEX.aspx?catid=29>.
- PALEY, D. (2005). *Principles of Deformity Correction*, Springer.
- QUIJANO, S., SERRURIER, A., AUBERT, B., LAPORTE, S., THOREUX, P. and SKALLI, W., (2013). "Three-dimensional reconstruction of the lower limb from biplanar calibrated radiographs." *Medical Engineering & Physics* 35(12): 1703-1712.
- RAJAMANI, K. T., STYNER, M. A., TALIB, H., ZHENG, G. Y., NOLTE, L. P. and BALLESTER, M. A. G., (2007). "Statistical deformable bone models for robust 3D surface extrapolation from sparse data." *Medical Image Analysis* 11(2): 99-109.
- SEDERBERG, T. W. and PARRY, S. R., (1986). "Free-form deformation of solid geometric models." *SIGGRAPH Comput. Graph.* 20(4): 151-160.
- SMART CORRECTION, L., (2016). "The Smart Correction® Computer-Assisted Circular External Fixator System." Retrieved 30.04.2016 from <http://www.smartcorrection.com/>.
- SMITH&NEPHEW, (2016). "What is the Taylor Spatial Frame." Retrieved 30.04.2016 from <http://www.smith-nephew.com/patient/treatments/limb-restoration/what-is-the-taylor-spatial-frame-----/>.
- SOLOMIN, L. N., UTEKHIN, A. I. and VILENSKY, V. A. (2012). *Deformity Correction and Fracture Treatment Using the Software-Based Ortho-SUV Frame. The Basic Principles of External Skeletal Fixation Using the Ilizarov and Other Devices*. L. Solomin. Milano, Springer Milan: 705-803.
- SPIEGELBERG, B., PARRATT, T., DHEERENDRA, S. K., KHAN, W. S., JENNINGS, R. and MARSH, D. R., (2010). "Ilizarov principles of deformity correction." *Ann R Coll Surg Engl* 92(2): 101-105.

- STEWART, D., (2009). "A Platform With Six Degrees of Freedom (Reprinted from vol 180, 1965)." *Proceedings of the Institution of Mechanical Engineers Part C-Journal of Mechanical Engineering Science* 223(1): 266-273.
- TAYLOR, J. C. and TAYLOR, H. S. (1999). Orthopaedic fixation plate, Google Patents.
- VICECONTI, M., DAVINELLI, M., TADDEI, F. and CAPPELLO, A., (2004). "Automatic generation of accurate subject-specific bone finite element models to be used in clinical studies." *Journal of Biomechanics* 37(10): 1597-1605.
- VICECONTI, M., SUDANESE, A., TONI, A. and GIUNTI, A., (1993). "A software simulation of tibial fracture reduction with external fixator." *Computer Methods and Programs in Biomedicine* 40(2): 89-94.
- VICECONTI, M., ZANNONI, C., TESTI, D. and CAPPELLO, A., (1999). "CT data sets surface extraction for biomechanical modeling of long bones." *Computer Methods and Programs in Biomedicine* 59(3): 159-166.
- YOSIBASH, Z., TRABELSI, N. and MILGROM, C., (2007). "Reliable simulations of the human proximal femur by high-order finite element analysis validated by experimental observations." *Journal of Biomechanics* 40(16): 3688-3699.
- ZHENG, G. and NOLTE, L.-P. (2006). Surface Reconstruction of Bone from X-ray Images and Point Distribution Model Incorporating a Novel Method for 2D-3D Correspondence. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, IEEE Computer Society*: 2237-2244.
- ZHENG, G. Y., (2006). "Reconstruction of patient-specific 3D bone model from biplanar X-ray images and point distribution models." *2006 IEEE International Conference on Image Processing, Icip 2006, Proceedings*: 1197-1200.
- ZHENG, G. Y., BALLESTER, M. A. G., STYNER, M. and NOLTE, L. P., (2006). "Reconstruction of patient-specific 3D bone surface from 2D calibrated fluoroscopic images and point distribution model." *Medical Image*

Computing and Computer-Assisted Intervention - Miccai 2006, Pt 1 4190: 25-32.

ZHENG, G. Y. and SCHUMANN, S., (2008). "3-D Reconstruction of A Surface Model of the Proximal Femur from Digital Biplanar Radiographs." 2008 30th Annual International Conference of the Ieee Engineering in Medicine and Biology Society, Vols 1-8: 66-69.

ZHENG, G. Y. and SCHUMANN, S., (2009). "3D reconstruction of a patient-specific surface model of the proximal femur from calibrated x-ray radiographs: A validation study." Medical Physics 36(4): 1155-1166.

ZHU, Z. L. and LI, G. A., (2011). "Construction of 3D human distal femoral surface models using a 3D statistical deformable model." Journal of Biomechanics 44(13): 2362-2368.

BIOGRAHPY

Ercan Avşar was born in Adana, Turkey in 1985. He received his BSc. from Electrical and Electronics Engineering Department of Çukurova University, Adana, Turkey, in 2007 and MSc. from Biomedical Engineering program of Istanbul Technical University, Istanbul, Turkey, in 2009, respectively. He worked as a research assistant at Osmaniye Korkut Ata University for eight months starting from August 2010. Now, he has been working as research assistant in Electrical and Electronics Engineering Department of Çukurova University since 2012. His research interests include computer graphics, 3D modeling and machine learning.