A SYSTEM FOR COMPUTATIONAL ANALYSIS AND RECONSTRUCTION OF 3D COMMINUTED BONE FRACTURES

by

Pengcheng Liu

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Electrical Engineering

Charlotte

2012

Approved by:

____

Dr. Andrew R. Willis

____

Dr. Thomas P. Weldon

____

Dr. Jiang (Linda) Xie

____

Dr. Anthony B. Wilkinson

____

Dr. Richard M. Souvenir

ABSTRACT

PENGCHENG LIU. A system for computational analysis and reconstruction of 3D comminuted bone fractures. (Under the direction of DR. ANDREW WILLIS)

High energy impacts at joint locations often generate highly fragmented, or comminuted, bone fractures. A leading current approach for treatment requires physicians qualitatively to classify the fracture to one of four possible fracture severity cases. Each case then has a sequence of best-practices for obtaining the best possible prognosis for the patient. It has been observed that *qualitative* evaluation of fracture severity by physicians can vary significantly which can lead to potential mis-classification and mis-treatment of these fracture cases. Major indicators of fracture severity are (i) fracture surface area, i.e., how much surface area was generated when the bone broke apart and (ii) dispersion, i.e., how far the fragments have rotated and translated from their original anatomic positions. Work in this dissertation develops computational tools that solve the bone puzzle-solving problem automatically or semi-automatically and extract previously unavailable quantitative information for these indicators from each bone fragment that are intended to assist physicians in making a more accurate and reliable fracture severity classification. The system applies novel three-dimensional (3D) puzzle-solving algorithms to identify the fracture fragments in the CT image data and piece them back together in a virtual environment. Doing so provides *quantitative* values for both fracture surface area and dispersion that reduce variability in fracture severity classifications and prevent mis-diagnosis for fracture cases that may be difficult to qualitatively classify using traditional approaches. This dissertation describes the system, the underlying algorithms and demonstrates the virtual reconstruction results and quantitative analysis of comminuted bone fractures from six clinical cases.

ACKNOWLEDGMENTS

It would not have been possible to write this doctoral dissertation without support and assistance of people around me, I would like to express my gratitude towards them here. My Ph.D. studies were supported as research-assistantship and as teaching assistantship from UNC Charlotte graduate school Graduate Assistant Support Plan (GASP) and the Electrical and Computer Engineering Department.

Above all, I would like to thank my family for their support and help, especially my wife Xi Zhao and my daughter Grace Zhao Liu. My parents and parents-in-law have given me their support throughout, as always, for which my only expression of thanks likewise does not fulfill.

I must express my extreme gratitude towards my advisor, Dr. Andrew R. Willis, whose expertise and understanding guided me through the completion of my PhD degree. Without his support and patience, this dissertation would not have been possible. I appreciate many things he did to shape me into a better person.

I also thank all of my fellow PhD students who had worked together with me, inspired me, and helped me. I learned a lot from them during my PhD education here.

I'd like to thank the many professors who have taught me in classes, in projects and even in my daily life: Dr. Richard Souvenir, Dr. Thomas Weldon, and Dr. Jiang Xie. Also, I would like to express my gratitude to folks in the University of Iowa who provided me with real clinical data and assistance to help me complete my dissertation.

Lastly, but by no means least, I wish to thank my coworkers in research department of Camstar systems, Rami Locas, Kathleen deValk, Kevin Farmer, and Sandeep Chaturvedi. I would also to thank PJ Brunson for checking my writing. To them, I dedicate this dissertation.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Accurately classifying the severity of highly comminuted bone fractures can be challenging for orthopedic physicians and surgeons. Research in [1] states that accurate determination of the initial fracture severity as afforded by fracture severity classification is the single most important prognostic determinant of long-term joint health subsequent to trauma. Due to the importance of fracture severity classification, many researchers [2, 3, 4, 5, 6] have investigated this problem where their common goal is to define methods capable of predicting fracture severity from quantitative measurements derived from medical image data.

This dissertation defines a new system for 3D image and surface analysis that enables users to perform a virtual 3D reconstruction of a comminuted bone fracture from a 3D CT [1] image of the fractured limb. Our system accomplishes this task as a sequence of three steps:

1. Fragment surfaces are extracted from CT images,
2. Each fragment surface is further decomposed into anatomically meaningful subregions,
3. Fragments are pieced back together in a virtual space with a puzzle-solving algorithm.

Using this process, the system produces a virtual reconstruction of an unbroken bone from the 3D CT of the fractured limb. The work described in this dissertation develops computational tools that solve the puzzle-solving problem automatically and semi-automatically and extracts previously unavailable quantitative information from each

[1] See "CT" in Glossary

bone fragment that is intended to assist physicians in making a more accurate and reliable fracture severity classifications.

Accurate reconstruction of a patient's original bone anatomy is the desired outcome for surgical treatment of a bone fracture. Treatment goals include achieving expeditious reconstruction, and avoiding Post-Traumatic OsteoArthritis (PTOA[2]). When there is involvement of an articulating joint such as the hip, knee, or ankle, accurate reconstruction of the bone joint surface is critical to avoid PTOA. Yet, this task can be quite challenging when dealing with highly comminuted fractures. This is due to the fact that often dozens of individual fragments are involved and they are sometimes displaced significantly from their original anatomic position and scattered in a complex geometric pattern.

This dissertation chooses to focus on clinical cases of tibial plafond fracture[3](see figure 1.1c and figure 1.1d). This type of fracture typically occurs as a result of high-energy trauma such as ballistic penetrations, vehicular accidents, or falls from a height. This dissertation chooses tibia fracture cases for several reasons:

1. The complex characteristics of this kind of fracture can often create difficulties for physicians in making accurate and reliable fracture severity assessments,
2. Tibia fractures often involve the ankle joint which is typically difficult to treat,
3. The quality of reconstruction is a critical factor for good prognosis,
4. PTOA is directly related to the accuracy of reconstruction,
5. Tibia fractures are important in military conflicts because it is a common combat injury.

1.1 Structure of this Dissertation and Terminology

The content of this dissertation is organized as follows: Chapter 1 introduces the dissertation topic, states the problem to be addressed (see 1.4) and describes the

[2] See "PTOA" in Glossary

[3] See "tibial plafond fracture" in Glossary

contribution (see 1.5) of this work to the state-of-the-art. Chapter 1 also introduces important background information on the current treatment of fractures (see 1.2) and fracture severity metrics (see 1.3). Chapter 2 describes the system interface and compares this system with other commercial products for image-based analysis of orthopedic injuries. Chapter 3 details the underlying algorithms that are applied by the system including a detailed description for their implementation and related previous work (see 3.1 - 3.5). Chapter 4 details the implementation of the 3D puzzle solving algorithm, which is the core contribution of this work. Chapter 5 describes 3D reconstruction results produced by using the system for six clinical cases and provides quantitative analysis of these results. Chapter 6 details a user study conducted to evaluate the effectiveness of the current system. Chapter 7 summarizes the work and the current state of the system and details future investigations needed that will further improve the system.

Some terminology used throughout this document is defined here to simplify the discussion. In this dissertation, we denote the 3D CT image obtained from patient's fractured limb as the *fracture CT image*, and the 3D CT image obtained from the patient's healthy limb as the *intact CT image.* We denote the outer surface of bone excluding articular surface as the *periosteal surface,* the smooth surfaces at the end of bones used for joint movement are *articular surface,* and the *fracture surface* of a bone are the surfaces generated when the bone broke apart. Finally, we denote unbroken bone surface extracted from the intact CT image as the *intact template*, and the bone fragment surfaces extracted from the fracture CT image as *bone fragment*s.

1.2 Current Treatment of Bone Fractures

Orthopedic surgery is concerned with conditions involving the musculoskeletal system. Orthopedic surgeons use both surgical and non-surgical means to treat musculoskeletal traumas, sports injuries, degenerative diseases, infections and tumors. In many cases, developments in orthopedic disciplines were brought on by technological advances from other fields including physics, chemistry, mechanical engineering, biological engineering, material engineering, and electrical engineering, etc. Examples of new technologies include X-ray (see figure 1.1d) and computed tomography (CT, MRI[4], PET[5])(see figure 1.1b) which provide means for obtaining an image of a subject's internal anatomy and empower orthopedic surgeons to perform less invasive and more accurate surgical treatment. As imaging technology and computer algorithms developed, computer-aided tools for surgical pre-operative planning became feasible.

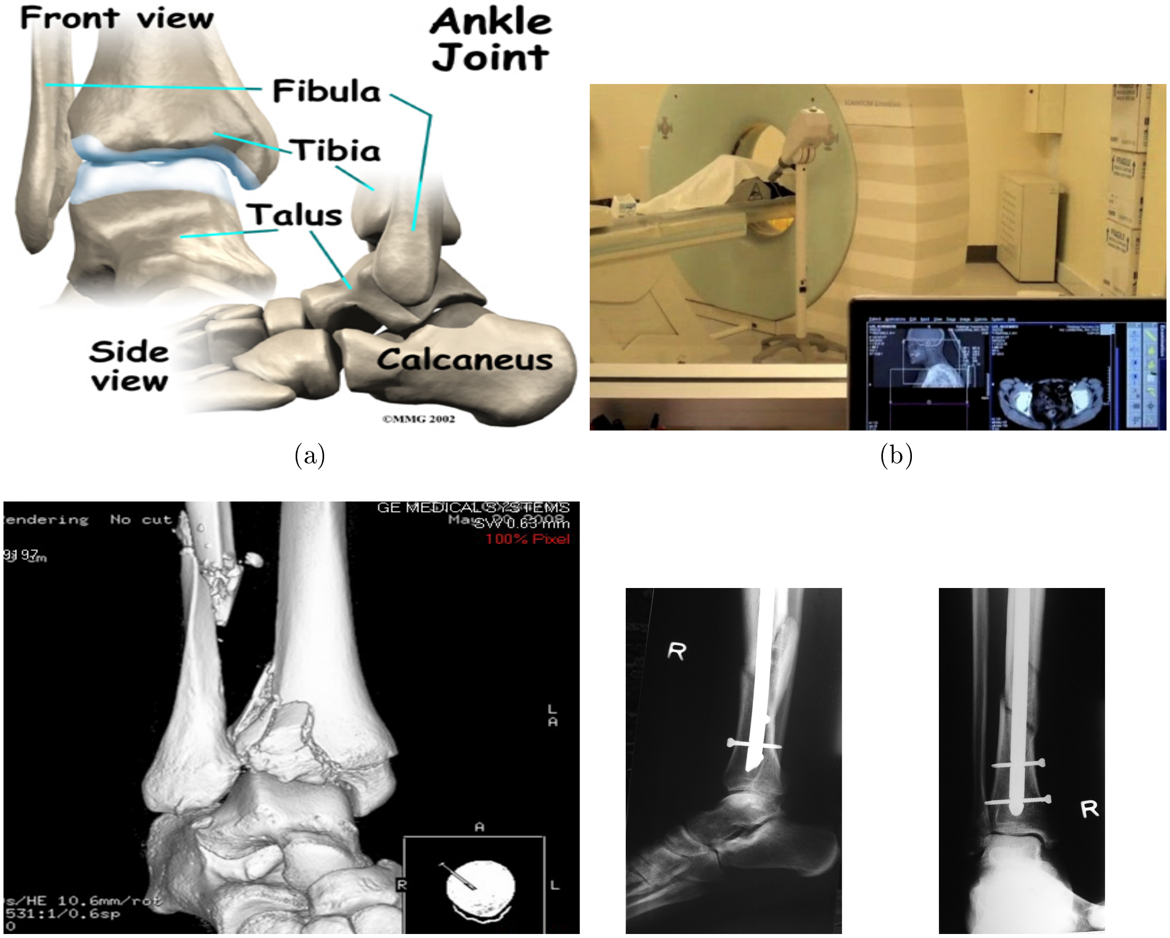
When a patient is treated in the hospital, different imaging equipment such as X-ray, fluoroscopy [6], CT imaging and MRI, are used to obtain a detailed view of the injury. Trained physicians use the image information and their experience to estimate the severity of the fracture. This is accomplished by classifying the fracture to an adopted fracture severity metric system. Given the severity classification, surgical treatment methods are planned and executed by experienced orthopedic surgeons. After the surgery, patients receive physical therapy and rehabilitation to facilitate recovery from the injury. In some cases, post-operative imagery is collected during the recovery period. Surgeons and rehab specialists evaluate the collected information and guide the patient through the healing process.

This dissertation introduces a software system that allows users to virtually reconstruct the patient's original bone anatomy from 3D CT images of the patient's bone fracture. Using digital reconstruction, quantitative values for several key factors directly linked to fracture severity can be computed using the analysis tools of the system. These quantitative values are intended to help physicians improve the objectivity and accuracy their assessments of the fracture severity. The system also provides 3D visualizations of the reconstructed bone anatomy which allows views of the reconstruction not possible from CT images. Possible applications for the system

[4] See "MRI" in Glossary

[5] See "PET" in Glossary

[6] See "fluoroscopy" in Glossary



(a) (b)

(c) (d)

Figure 1.1: (a) A diagram of the ankle joint anatomy. (b) a CT scan of a patient is being performed, an imaging computer reveals a 3D image of the body's interior. (c) is a CT image that shows an example of a tibial plafond fracture. (d) shows two planar (2D) X-rays of a tibial fracture that have been reconstructed.

include fracture severity assessment, fracture surgical planning and virtual training for orthopedic surgeons.

1.3 Fracture Severity Metrics

Several methods exist for estimating fracture severity, their goal is to determine the extent and type of fracture injury. Severity metrics for fractures allow physicians to relate new cases to prior cases having similar attributes. This allows physicians to draw upon past experience to determine a treatment that will provide the best possible prognosis. Most metrics require the physician to classify new cases to a class of similar cases. Using this classification, surgeons determine their treatment plan, estimate the recovery time and attempt to predict the likely level of function for the injured limb after recovery.

Because the fracture severity classification directly influences the surgeon's choice of treatment and the patient's anatomical and functional outcome, quantifying the severity of a fracture becomes a crucial step in the diagnosis and prognosis of an injury. It can alter the surgery and treatment planning and is more likely to be advantageous in difficult cases such as highly comminuted fractures. Due to the importance of the severity metrics, orthopedic surgeons and researchers have been pursuing methods to improve their accuracy [4].

1.3.1 Observer-Based Approaches

This section describes several traditional observer-based approaches for fracture severity classification in detail. Existing observer-based methods for assessing fracture severity consist of four stages:

1. Choosing a fracture classification system.
2. Capturing the images of the fracture.
3. Observing these images.
4. Based on the observations in step 3 and the training and experience of the physician, the best-fit classification for the fracture case is selected.

In practice, severity metrics define distinct fracture classes based on the bone, location (part of the bone), number of fragments, skin wound over the break, and orientation of fracture line. Numerous fracture classification systems have been proposed in orthopedics [7, 8, 9, 10], but only a small number of them have become widely accepted in practice, such as the Müller AO Classification of fractures [11], Salter-Harris classification [12], and Schatzker classification [13]. Yet, none of them have withstood rigorous validation tests [14]. The following sections describe these three most popular classification systems.

1.3.2 Müller AO Classification System

The Müller AO classification system describes the fracture pattern as characterized by the location (bones and segments), degree of comminution and type of displacement. It is thought to be the most comprehensive system for the classification of fractures [11]. The system defines a numbering scheme based on the anatomical location of a fracture and bone segments, i.e, portion of the long bone, as is shown in figure 1.2. Each major long bone and each specific anatomical region, as well as each bone region, are named and numbered. The fractures of each bone segment are then divided into three types (A, B, C) with a further sub-division into three groups and sub-groups, therefore generating a three-tier hierarchical classification system. The classification system consists of 27 sub-groups for each bone segment. Since each long bone consists of three bone segments, this creates a total of 81 subgroups. See figure 1.3 for an example classification of a fracture of the end segment into three fracture groups. Figure 1.4 shows a step-by-step example that demonstrates how a diaphyseal[7] fracture can be classified by answering a sequence of questions.

1.3.3 Salter-Harris Classification System

The Salter-Harris classification system targets growth plate fractures, i.e., fractures that occur at the ends of long bones where cartilage tissue grows. This system includes

[7] See “diaphyseal” in Glossary

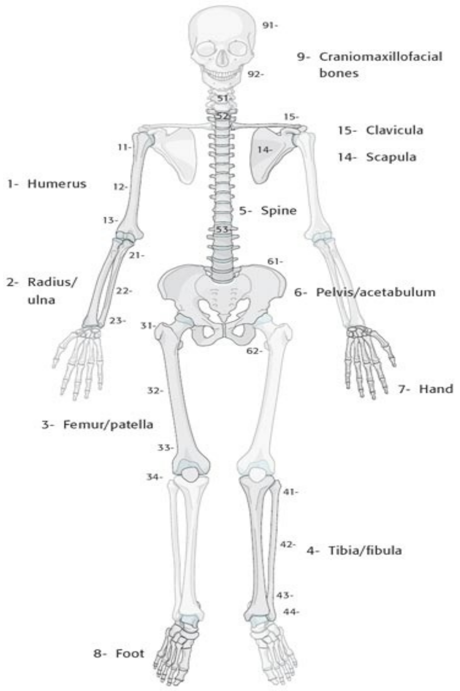


Figure 1.2: AO classification system: the image for numbering the anatomical location of a fracture in three bone segments [11]. (proximal =1, diaphyseal =2, distal = 3)

five injury types (See figure 1.5):

1. Slipped: a fracture through the physis[8] without involvement of the bone of the epiphysis[9] or metaphysis[10].
2. Above: a fracture that lies above the growth plate.
3. Lower: a fracture in the lower part of the growth plate.
4. Through: a fracture that extends through the growth plate.
5. Rammed: a growth plate that has been rammed or crushed.

This classification system has seen widespread use since 1963. It was later refined by physicians when they recognized that the original five types were insufficient to categorize all growth plate injuries. To address this issue the following four additional rare types of growth plate injuries were added to the original five:

[8] See "physis" in Glossary
[9] See "epiphysis" in Glossary
[10] See "metaphysis" in Glossary

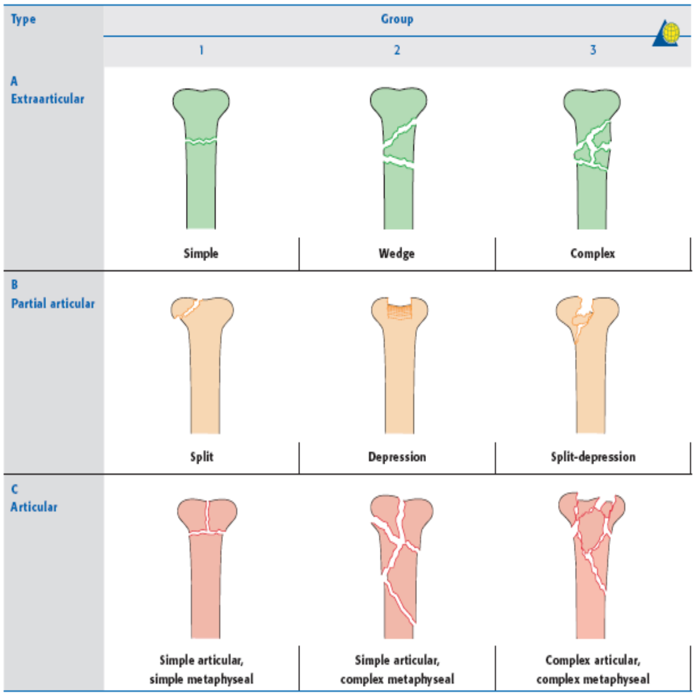
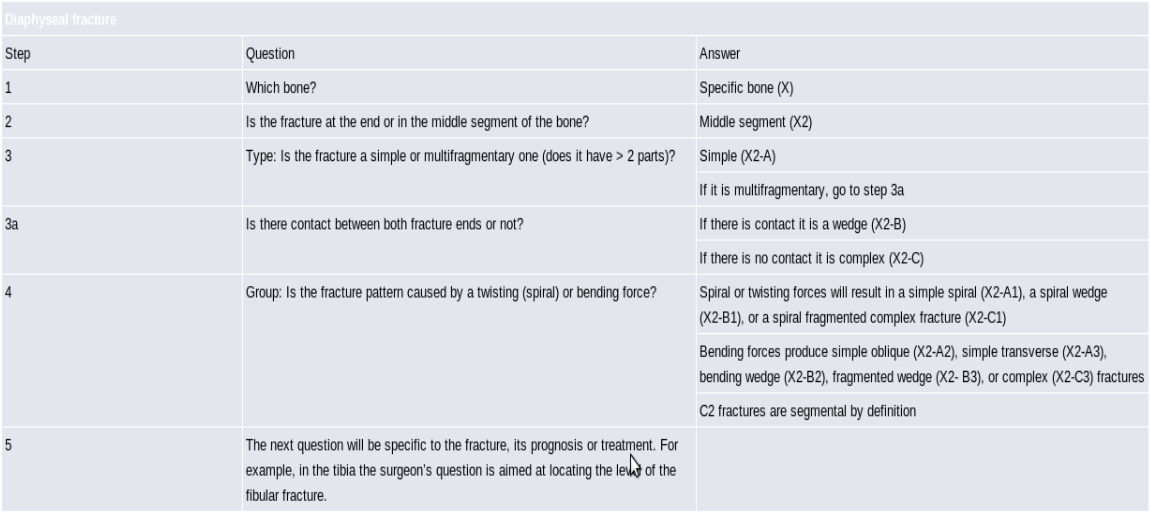

Type
Group
1
2
3
A
Extraarticular
Simple
Wedge
Complex
B
Partial articular
Split
Depression
Split-depression
C
Articular
Simple articular, simple metaphyseal
Simple articular, complex metaphyseal
Complex articular, complex metaphyseal

Figure 1.3: AO classification system: an example of AO classification of the end segment, nine fractures range from easy to difficult are shown in the image. Hence, A1 is considered as the fracture with the best prognosis and the one that is easiest to treat clinically. C3 represents the fracture with the worst prognosis and the one that is typically difficult to manage. The image is taken from [11].



Diaphyseal fracture		
Step	Question	Answer
1	Which bone?	Specific bone (X)
2	Is the fracture at the end or in the middle segment of the bone?	Middle segment (X2)
3	Type: Is the fracture a simple or multifragmentary one (does it have > 2 parts)?	Simple (X2-A)
		If it is multifragmentary, go to step 3a
3a	Is there contact between both fracture ends or not?	If there is contact it is a wedge (X2-B)
		If there is no contact it is complex (X2-C)
4	Group: Is the fracture pattern caused by a twisting (spiral) or bending force?	Spiral or twisting forces will result in a simple spiral (X2-A1), a spiral wedge (X2-B1), or a spiral fragmented complex fracture (X2-C1)
		Bending forces produce simple oblique (X2-A2), simple transverse (X2-A3), bending wedge (X2-B2), fragmented wedge (X2- B3), or complex (X2-C3) fractures
		C2 fractures are segmental by definition
5	The next question will be specific to the fracture, its prognosis or treatment. For example, in the tibia the surgeon's question is aimed at locating the level of the fibular fracture.	

Figure 1.4: AO classification system: a sequence of questions are shown that depict the process of classifying a fracture case using the AO system [11].

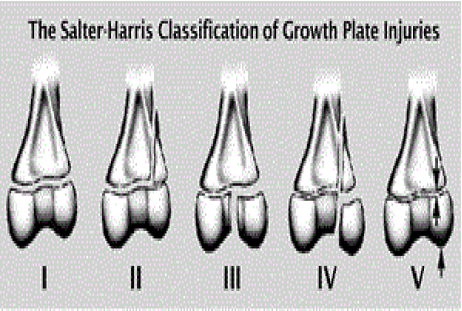


Figure 1.5: This image shows the five fracture types for the Salter-Harris classification system.

1. Injury to the peripheral portion of the physis which may cause the deformation of the articular bone.
2. Isolated injury that occurs on the epiphyseal plate.
3. Isolated injury of metaphysis with possible impairment of *endochondral ossification*, i.e., the formation of bone in which a cartilage template is gradually replaced by a bone.
4. Injury extends to the *periosteum*, i.e., a specialized connective tissue covering all bones and having bone-forming potentialities, which may impair the ossification that occurs in and replaces connective tissue.

1.3.4 Schatzker Classification System

The Schatzker classification system targets tibial plafond fractures and divides these fractures into six different types (see figure 1.6) based on the fracture pattern. Each type is associated with a suggested treatment method and injury-specific difficulties with these treatments. The following list describes each type in detail:

1. Lateral split: A split fracture of the lateral tibial plateau without depression, it often occurs in younger patients with stronger bones Typical treatment for this fracture is *lateral fixation*, i.e., the process of holding or fastening fractures in a fixed position.
2. Split with depression: The most common tibial plateau fracture and often occurs

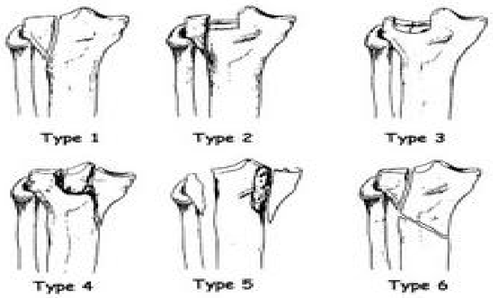


Figure 1.6: Schaztker classification system: the image displays six fracture types (from the Societe Francaise d'Arthroscopie).

in older patients because their bones do not resist depression. The treatment for this fracture is also lateral fixation with the depressed fragments elevated.

3. Pure lateral depression: This fracture is often due to a fall in older patients, the treatment is *lateral internal fixation*, i.e., an orthopedic operation that involves the surgical introduction of implants for the purpose of repairing a bone.
4. Pure medial depression: This fracture is considered severe trauma because it involves the anterior surfaces of the tibia. For this type of injury, simple fixation is not enough. Instead *medial plate fixation* is used where a plate and screws are surgically introduced to hold the fractured bones together.
5. Bicondylar: This fracture is a combination of a split medial and a lateral tibia fracture. This type of injury is usually caused by pure axial impact and considered severe trauma. The treatment for this type is medial plate fixation combined with lateral internal fixation.
6. Split extends to metaphysis: This type of fracture is considered high-energy trauma, and may often involve articular surfaces. The treatment for this type is medial plate fixation combined with lateral internal fixation.

1.3.5 Key factors in Fracture Severity Assessment

The three classification systems discussed above are the most commonly used methods for classifying fractures. The AO classification system is the most comprehensive and refines the process of classification once the bone and the bone segment have been identified. However, Salter-Harris and Schatzker classification systems are more focused on the tibial plafond fractures. There are some common key factors that influence physicians' classification judgment no matter what fracture classification system they are using:

- Fragment displacement,
- Angular dislocation,
- Number of fragments,
- Size of fragments,
- Soft tissue damage,
- Fracture surface area,
- Relative displacement of fragments,
- The volume of displaced soft tissue.

Fracture severity classification systems attempt to integrate these factors into a single metric which is a complicated problem because many different key factors are involved. Each key factor reflects one fracture attribute that is linked to the level of severity.

Fracture severity classifications made by physicians can vary significantly due to the complexity of the fracture process and additional extenuating influences. Extenuating influences that effect physician classifications include the following:

- imperfect imaging data can create blurred images and make classification difficult,
- lack of standardized references, i.e., a fracture severity "gold standard" [14],
- observer-based fracture severity classification depends significantly upon physicians' experience and skills.

This dissertation attempts to address some of the shortcomings of existing methods for severity classification by deriving quantitative values for many of the key factors listed above from 3D CT imagery. It is hoped that physicians can use these values to generate more accurate and objective assessments of fracture severity.

1.3.6 Automatic Classification Approaches

Since traditional observer-based approaches for quantifying fracture severity have been found to depend upon the physician's visual assessment and experience [15], several efforts have been made by researchers to define automated fracture severity classification processes [5, 2, 6]. These research efforts target the key factors previously listed that impact the fracture severity. Work in [5] uses fragment displacement and the volume of displaced soft tissue. Work in [2] uses number of fragments, size of fragments, and fracture surface area. Work in [6] uses soft tissue damage, and fragment displacement, and fracture surface area.

Work in [2] proposed that fracture surface area is a major factor in determining fracture severity. The rationale of this approach relies on fracture mechanics theory for brittle solids which states that a monotonic relationship exists between the energy absorbed by the bone due to the fracture impact and the fracture surface area generated when the bone fragmented. Work in [2] tested this hypothesis by using segments of bovine bone and experimentally fracturing them with a controlled impact where the energy of the impact was measured. Their results demonstrated, for the first time, that fracture severity for impacted cortical bone can be characterized on a continuous scale based on CT-apparent energy absorption. Although the tests in [2] are not solid proof for the clinical domain, this approach introduced a physically justified basis for quantification in an area which has been until now purely the domain of subjectivity.

Work in [5] considered the displaced soft tissue volume as a metric for fracture severity. Here the mirror image of the healthy contra-lateral tibia is aligned with an image of the fractured limb. Once aligned, a score for the fragment displacement was computed by computing the volume of convex hull for the bone fracture. A convex hull is the smallest convex polygon that contains the fracture bone fragment tissue identified within the CT image. The volume difference between the convex hull volume for the fractured bone fragments and convex hull volume for the intact bone is considered to be an indicator for both fragment dispersion and the area of newly generated fragment surface. From this measurement, a quantitative metric for assessing the fracture severity was generated. Three experienced orthopedic clinicians made the severity ranking for 20 clinical cases separately. Their ranking results were compared with the severity ranking automatically made using their CT metric based on fracture energy and displacement respectively. The average *concordance value*, i.e., a quantitative statistical expression that reflects as a score, the percentage of cases have been ranked in the same order by two different methods. In this case the fracture energy based metric was shown to have a 77% agreement with clinician's assessment. Inclusion of a fragment displacement metric offered an improvement to the original metric to have an 86% agreement.

Work in [6] proposed a novel CT-based method to objectively measure fracture energy and evaluated the utility of this measure for predicting fracture severity. In addition to fracture surface area used in [2] and fragment displacement utilized in [5], this work introduced soft tissue swelling as another quantitative variable related to the fracture severity. The degree of fracture-associated soft tissue swelling was quantified by calculating the volume of non-osseous regions in the fractured limb and then subtracting the analogous measure for the intact contra-lateral limb. Fracture energy, fragment displacement, and soft tissue swelling were quantified in 20 tibial plafond fractures by analyzing injury CT scans. Three experienced clinicians then independently performed a rank order analysis of fracture severity. The study described in [6] demonstrates good agreement (fracture energy concordance from 73% to 76%, fragment displacement from 82% to 89%, and soft tissue swelling from 61% to 65%) between the CT-based metric and expert severity assessments. The semi-automated technique physically links an injury severity metric to the energy absorbed in the initial injury. Such work also provides a prototype platform for assessing the relationship between severity of an injury and its eventual clinical outcomes.

1.4 Motivation and Goals

This dissertation attempts to address shortcomings in existing methods for fracture severity assessment. Currently, observer-based fracture classification schemes are used to determine for fracture severity. Yet, their value in guiding treatment has some potential drawbacks [15]. High-energy injuries in which comminution and displacement occur are difficult for physicians to classify accurately. Current classification systems categorize fractures according to the key factors, but objective measures for these key factors do not exist. Also, observer-based severity assessment can vary significantly which can lead to potential mis-classification and mis-treatment of these high-energy fracture cases. Among high-energy fractures, comminuted fractures that extend into an articular joint are especially troublesome due to the difficulty in visualizing and understanding the complex 3D anatomy of the fracture and limited visibility of important anatomic structures within the images [16, 4].

This dissertation introduces a system for computational analysis of comminuted fractures. The system allows for quantitative severity analysis by estimating the key factors that impact severity. Estimates are obtained by generating a complete 3D reconstruction of the unbroken bone from the bone fragments. The reconstruction process computes quantities for several key factors that impact the severity such as fragment displacement, angular dislocation, number and size of fragments, and fracture surface area. These quantities can be used for fracture severity analysis via the system's analysis tools. Physicians may utilize this quantitative information to potentially improve the accuracy of the fracture severity classification which can improve applied treatments and prognosis for these fractures.

1.5 Contribution

The contributions of this dissertation in the area of medical image processing and 3D puzzle-solving are discussed in this section. This dissertation has three main contributions as listed below:

1. A complete fracture analysis system was created that is capable of virtually reconstructing a bone from a 3D CT image of the fractured bone.
2. A fully automatic computational puzzle solving algorithm that was developed that pieces together bone fragment surfaces to reconstruct the unbroken bone.
3. Quantitative information is provided about the bone fracture that is not available from any existing medical image analysis system.

The bone reconstruction system designed in this dissertation provides unique visualization capabilities for fracture management by allowing users to view and edit their fracture cases in both 2D (CT image) and 3D (fragment surface). The system interface provides a structured process for reconstruction that enables users to control and manage the data at each step of the reconstruction process. The system is a unique combination of state-of-art 2D/3D image processing and surface processing algorithms together with an interface appropriate for non-expert users. This integration allows users to solve bone fracture cases from the beginning, i.e., raw CT image data, to the end, i.e., quantitative evaluation information about the fracture severity. The system also provides tools that link the 2D CT imagery with 3D visualizations of the bone surfaces inside a single interface. This helps users better understand both of these representations (see Chapter 5). The 3D puzzle solving algorithm introduced in this dissertation is the first automatic bone reconstruction tool for this purpose. The quantitative fracture analysis is a valuable final product and may provide crucial information for physicians to make fracture severity classification better. Although the system's ability of to assist orthopedic surgeons in pre-operative planning has not been well studied yet, the work in this dissertation provides the first example of a software system that enables physicians to visualize, assess, manipulate and reconstruct bone fragments virtually, and includes a novel 3D puzzle-solving algorithm to automatically reconstruct these bone fragments.

Visualization of both the fractured displaced and virtually reconstructed joints in 3D provides important information for treatment. In comparison to viewing CT images in 2D, the degree of comminution and specific fracture characteristics may be easier to understand using 3D renderings of the bone fragment surfaces extracted from CT imagery. For tibia fractures, the amount of articular disruption usually cannot be judged, since the joint surface is usually hidden by other bones such as the talus. Puzzle solving results resolves this issue by restoring tibia anatomy and color coding, i.e., segmenting the individual bones and bone fragments (see Chapter 5). In addition to providing detailed positioning information, inspection of the 3D reconstruction and alignment data enables one to analyze other fracture characteristics (see case 6 in 5.6).

As shown in the sections from 5.1 to 5.6, without special weighting for articular surface alignment, the puzzle solution achieves sub-millimeter accuracies for non-deformed articular bone fragments (see table 5.2). The effect of virtual bone reconstruction on clinical fracture cases is not studied yet. In this regard, statistically significant conclusions can be made due to the lack of experimental data. However, the successful application of the system to reconstruct a series of clinical comminuted tibia plafond fractures shown in Chapter 5 demonstrate that the 3D puzzle solving system represents a significant advancement towards a clinically practical technology. Injury characteristics such as fracture energy (in terms of fracture surface area), fragment displacement (in terms of fragment rotation and translation), and angular dislocation can now be precisely measured with puzzle-solving (see Chapter 5).

CHAPTER 2: THE SYSTEM INTERFACE

This chapter introduces the graphical interface of the system developed for this dissertation. The system interface is implemented as a Java application that provides 3D visualizations to view results and an interface to control the algorithms that assist in the reconstruction and analysis of the bone fracture (see figure 2.1). The system interface consists of three main windows:

1. The region outlined with a red line in figure 2.1 is the *3D canvas*. The system displays 3D objects here for the user to view and manipulate.
2. The region outlined with a green line in figure 2.1 is referred to as the *image panel*. The system displays intact CT images and fracture CT images here for the user to view and manipulate.
3. The region outlined with a yellow line in figure 2.1 is referred to as the *tree panel*. The system displays a tree representation of objects in 3D canvas here for the user to operate on specific objects or groups of objects.

The 3D canvas provides a virtual environment where the user can transform and view the individual object or group of objects. In addition, the user can also hide objects in the 3D canvas. The image panel displays 2D CT images of the fractured and intact limbs in two separate windows. The displayed 2D CT image is a slice of the complete 3D CT image and different slices can be viewed using a slider bar below each displayed image. The system also supports three different anatomic views for both the fracture and intact CT images: (1) axial, (2) sagittal, and (3) coronal. Fragments shown in the 3D canvas window are displayed as nodes in the tree panel. Each fragment may be partitioned into one or more surface patches. There are four

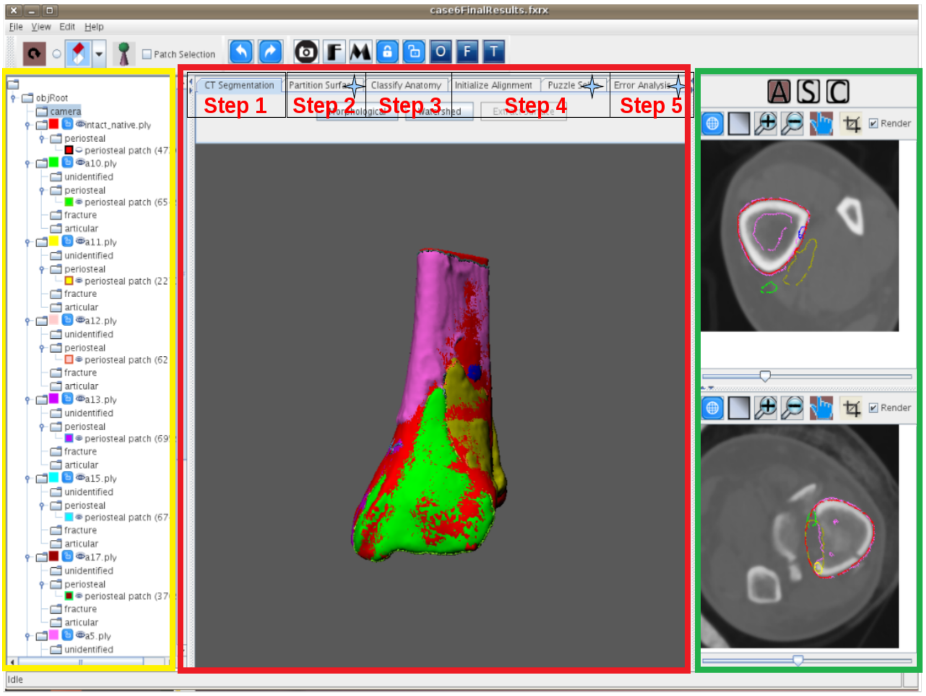


Figure 2.1: A screen capture of the system interface developed in this dissertation is shown. This image was captured after the fracture fragments had been reconstructed. Tabs in red at the top of the window access separate interfaces for each of the five steps required to reconstruct and analyze fractures.

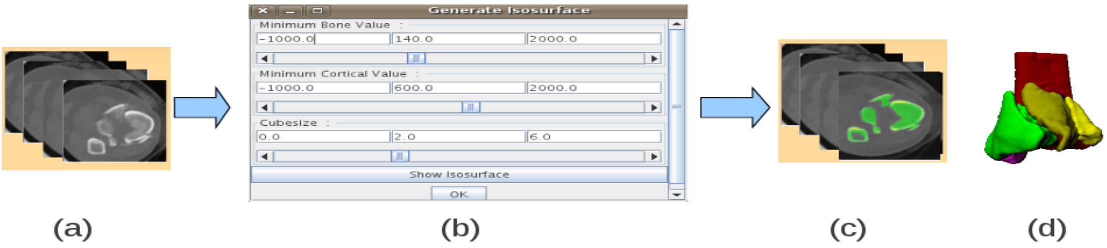
different categories for surface patches: (1) unidentified, (2) fracture, (3) periosteal, and (4) articular. Each category is represented as a child node of the fragment node in the tree panel. The position of a surface patch node in the tree panel indicates the surface category to which it belongs. The category of a surface patch can be manually specified by moving the surface patch object in the tree panel into the appropriate location such that the surface patch is as a child node of one of the four surface types which are indicated as folders in the tree panel. The tree panel also allows users to interactively classify fragment surface patches which is accomplished by selecting objects (fragments or fragment surfaces) in the tree panel and moving these objects to the appropriate folder. Users can also perform a number of useful actions on both fragments and their surface patches such as delete fragment, select fragment, merge and split surface patches, etc.

Fracture reconstruction using the system interface is accomplished by performing a sequence of five interactive steps as follows:

1. Segment the 3D bone surfaces from the intact and fracture CT images to generate an intact template surface and a surface for each bone fragment,
2. Partition the 3D fragment bone surfaces into surface patches,
3. Classify each surface patch to determine its anatomic type and create a model of the outer/periosteal surface of each fragment,
4. Reconstruct the bone by aligning the outer/periosteal surfaces of the bone fragments with corresponding portions of the intact template surface,
5. Analyze the fracture severity by inspecting the values of several key factors that relate to severity provide by the reconstruction.

In figure 2.1, the interface for these steps is shown as five tabs on top of the 3D canvas.

Step 1 – Segmentation of 3D bone surface: This step takes 3D CT images as input (see figure 2.2 (a)) and outputs a collection of 3D bone surface meshes (see figure 2.2 (d)). The parameters for the segmentation algorithm include (1) the minimum bone intensity value, i.e., the minimum intensity value considered as bone tissue, and (2) the minimum cortical intensity value, i.e., the minimum intensity value considered to be cortical bone tissue. After the algorithm finishes segmenting the 3D CT image, the interface displays 3D CT images where pixels associated with each bone fragment are marked with different colors (see figure 2.2 (c)). Bone fragment surfaces extracted from the segmented 3D CT images and are visualized as a collection of 3D surface meshes shown in the 3D canvas. These fragments are also added as objects in the tree panel. A third parameter is the cube size (see figure 2.2 (b)), i.e., the sampling density. The cube size determines the resolution of the extracted 3D surfaces. Surfaces are generated using the marching cubes algorithm on the segmented image data [17]. Alternatively, the user can segment the CT images and extract bone surfaces outside the system and load the resulting surface meshes and medical (DICOM format) images



Minimum Bone Value : -1000.0 | 140.0 | 2000.0

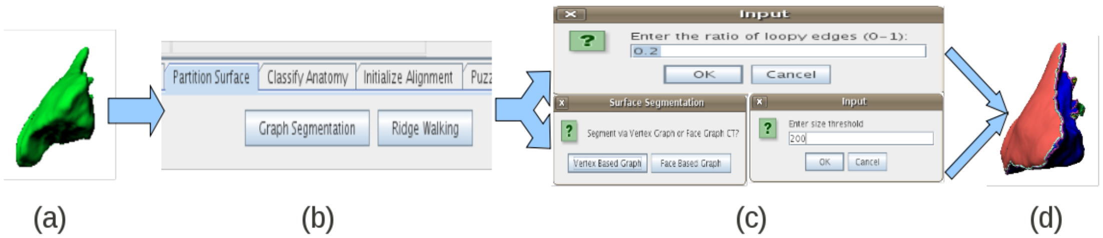
Minimum Cortical Value : -1000.0 | 600.0 | 2000.0

Cubesize : 0.0 | 2.0 | 6.0

Figure 2.2: This image shows the interactions that take place in step one.(a) a collection of medical input images, (b) user specified parameters for bone fragment segmentation, (c) segmented images, and (d) 3D fragment surfaces.

into the system.

Step 2 – Partition bone surface into sub-regions: This step takes a fragment (see figure 2.3 (a)) as input and outputs a collection of surface patches (surface sub-regions) (see figure 2.2 (d)). The user can choose between two algorithms (see figure 2.3 (b)) to partition fragment surfaces: (1) a ridge walking partitioning algorithm (see 3.2.1 for implementation details) or (2) a graph-based partitioning algorithm (see 3.2.2 for implementation details). If the ridge walking method is chosen, the user provides a percentage value (denoted as ρ_{ridge}) between 0 and 1 to the system interface which roughly controls the number of patches generated (higher values = more patches) (see figure 2.3 (c)). If the graph-based method is chosen, the user provides an integer value which roughly controls the number of patches generated (higher values = less patches) (see figure 2.3 (c)). When the partitioning algorithm finishes, each surface patch on each fragment is shown as a separate color and surface patch boundaries are also displayed. Each surface patch is added to the tree panel as a tree node that is a child of the fragment node from which it came. The system interface provides interactive tools to modify the partition results. These tools include a method to subdivide surface patches into smaller patches by repeating the partitioning algorithm on a surface patch (used when sensitivity is low) and a method to merge a collection of surface patches into a single surface patch (used when a single surface is divided into two or more surface patches). These operations are invoked by selecting the



Enter the ratio of loopy edges (0−1): 0.2

Segment via Vertex Graph or Face Graph CT?

Enter size threshold 200

Figure 2.3: This image shows user interactions that take place in step two. (a) an input fragment surface, (b) an interface window to choose a partitioning algorithm, (c) an interface window to specify algorithm parameters, and (d) a partitioned fragment surface.

corresponding tree nodes in the tree panel and running the *split* or *merge* functions. Patches can only be merged when they are adjacent, i.e., they must share at least one mesh edge on the 3D fragment model. Users may also delete surface patches by deleting the corresponding node in the tree panel.

Step 3 – Classify each sub-region into anatomically meaningful regions: This step takes as input a collection of surface patches (see figure 2.4 (a)) and categorizes these surface patches into three different types (see figure 2.4 (d)). The system interface provides an automatic and a manual method to perform this step. There are three stages for the automatic method:

1. Specify training data: Users provide training data to the system by selecting surface patches extracted from the fragment and assigning an anatomic category to the patch such as *fracture*, *periosteal*, *articular*, or *unidentified*. This step can be time-consuming since the user often needs to select many (~8-10) different surface patches on the fragment before there is sufficient data for accurate classification. Figure 2.4 (b) displays the window that allows the user to assign labels to the surface patches.

2. Train classifier: After the training data has been specified, the user runs a training algorithm (see 3.3 for implementation details) which processes the training data and generates a classifier which is stored as a file on disk. (see figure 2.4

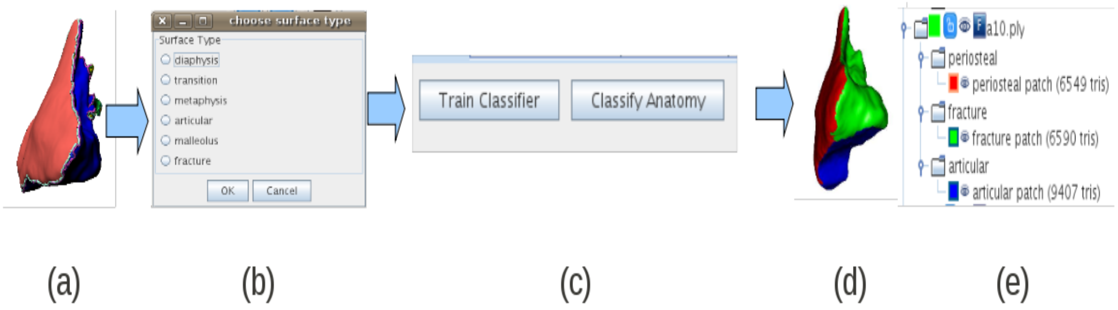


Figure 2.4: This image shows the interactions that take place for step three. (a) a fragment with a collection of partitioned surface patches, (b) an interface window for assigning labels to surface patches, (c) an interface window for training and classifying fragment surface data, (d) a fragment with classified surface patches, and (e) a screen shot of the tree panel showing the classified surface patches.

(c))

3. Classify anatomy: The user loads the classifier generated in step two from disk and applies the classifier to automatically classify all unidentified surface patches to their estimated categories (see 3.3 for details). As a result of this process, tree nodes representing surface patches are moved from the "unidentified" folder to the appropriate category folder.

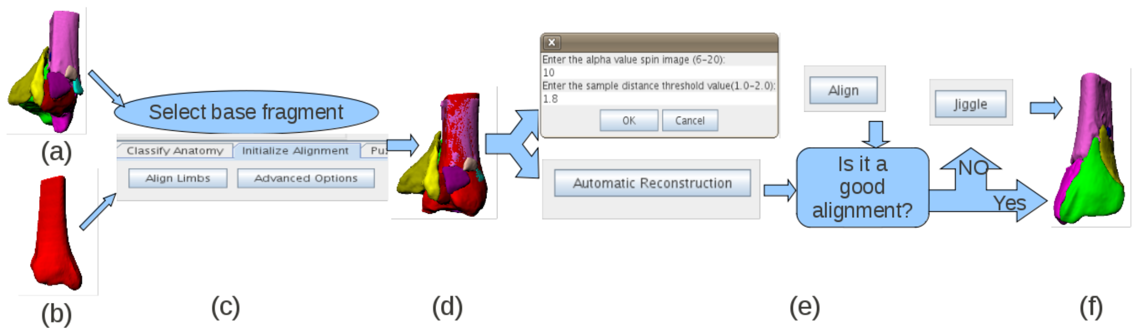
In the current system, the manual method is often more accurate and less time-consuming. Here, users manually move the fragment surface patches into different category folders using the tree panel as the interface. Surface patches under same category folder are then merged into a single patch. The system only allows one surface patch to exist within each category folder. (see figure 2.4 (e))

Step 4 – Reconstruction of the bone by aligning 3D bone surfaces: This step takes classified surface patches (see figure 2.5 (a)) and the intact template (see figure 2.5 (b)) as input and outputs a 3D reconstruction of bone fragments (see figure 2.5 (f)). There are two stages for this step: (1) initialize alignment, and (2) 3D bone fragment reconstruction.

For the first stage, users identify the *base fragment*. This is typically the largest bone fragment in the fracture case and is often the easiest fragment to align during reconstruction. This fragment is used to provide a gross alignment between the intact template and all of the fragments of the fracture. The alignment is generated by aligning the intact template to the base fragment which is accomplished by clicking on the button "Align Limbs" as shown in figure 2.5 (c). (see 3.4 for details). Figure 2.5 (d) shows an alignment result where the intact template has been aligned to the base fragment (the light pink fragment in figure 2.5 (a)).

For the reconstruction stage, the system interface provides an automatic solution and a semi-automatic solution. Using the automatic solution, the user clicks the "Automatic Reconstruction" button (see figure 2.5 (e)) which initiates the 3D puzzle-solving algorithm. If the automatic reconstruction fails to correctly position all of the bone fragments, the user may adjust the solution to improve the alignment of the misaligned fragments. Manually adjusting these fragments is often difficult as they are typically small. A custom alignment algorithm referred to as "jiggling" the fragments is provided to assist users in correctly positioning such fragments (see figure 2.5 (e)). The goal of this algorithm is to assist users in correcting the position of misaligned fragments. Details of its technical implementation are provided in section 4.4.4.

When the automatic reconstruction method fails, users may opt to manually reconstruct the fragments using the system. The manual reconstruction interface allows users to align individual fragments with the intact template. The interface consists of two 3D windows as shown in figure 2.6. There are two interactive modes for this interface: (1) point selection mode, where users specify corresponding points on the two viewed surfaces by clicking on the surface, and (2) viewing mode, where users translate and rotate the objects inside the 3D windows to obtain viewpoints of the objects where correspondences can be specified. These two interactive modes can be toggled by pressing the keyboard keys "A"(view mode) and "S" (selection mode).



Enter the alpha value spin image (6-20):
10
Enter the sample distance threshold value(1.0-2.0):
1.8

Figure 2.5: This image shows the interactions that take place in step four. (a) a collection of fragments, (b) an intact template, (c) an interface window for the first stage of reconstruction, (d) the fragments and the intact template in the aligned position after the initial alignment, (e) interface windows for stage two reconstruction, i.e., automatic reconstruction, and (f) the reconstruction result.

To perform alignment, users can simply right-click mouse which starts the alignment (when in the selection mode). Note that in order to align two surfaces, there should be at least three pairs of corresponding points selected. When the initial rough alignment with selected point pairs is started, the fragment surface in the right 3D window will be moved to the left 3D window and aligned to the intact template. A series of pop-up messages which contain alignment information and questions guides users on methods to refine the alignment. Once an alignment is complete, users can choose to restart the process if they are not satisfied with the result or accept the alignment. If the user accepts the alignment, the manual reconstruction interface will disappear and both surfaces will be moved to the main 3D canvas in their new aligned positions.

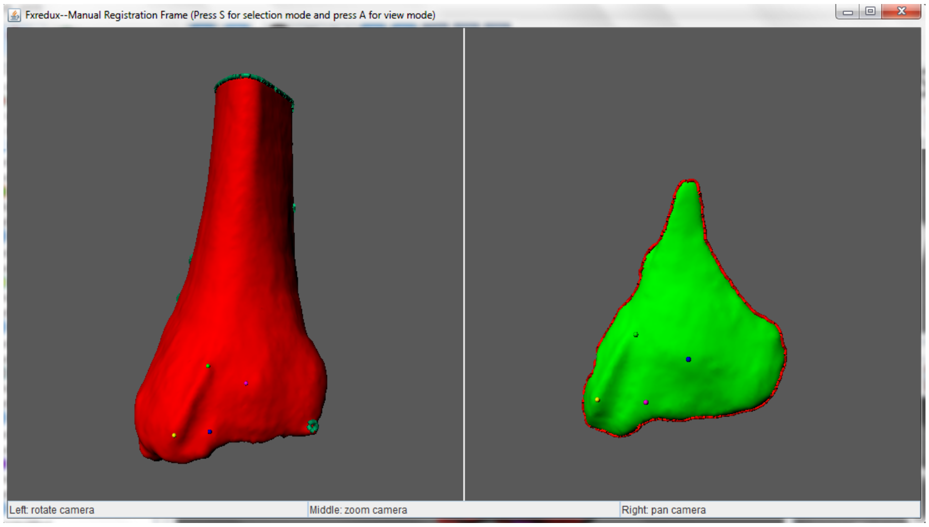

Fxredux--Manual Registration Frame (Press S for selection mode and press A for view mode)
Left: rotate camera
Middle: zoom camera
Right: pan camera

Figure 2.6: This image shows a screen capture of the manual reconstruction interface. The left part of the image displays a window which contains the 3D model of the intact template (red), the right part of the image displays a window which contains a 3D model of a bone fragment (green). On both surfaces, there are four pairs of user-selected corresponding points shown as small 3D spheres on the surface. Each pair of surface points are painted with same color.

Step 5 – Analysis of the fracture severity using quantitative estimates for key fracture severity factors: Two types of alignment analysis information are available after reconstruction. As shown in figure 2.7: (1) overall alignment error between the intact template and the fracture fragments, and (2) a histogram of alignment error between points on each fragment surface and the surface of the intact template. Overall alignment errors may be viewed by clicking the “Periosteal” button under the “Error Analysis” tab. A histogram of local alignment errors may be viewed by selecting target surface patch, clicking the “Periosteal” button and choosing the number of bins for the displayed histogram. Details regarding the computation of these error values are described in 3.5.

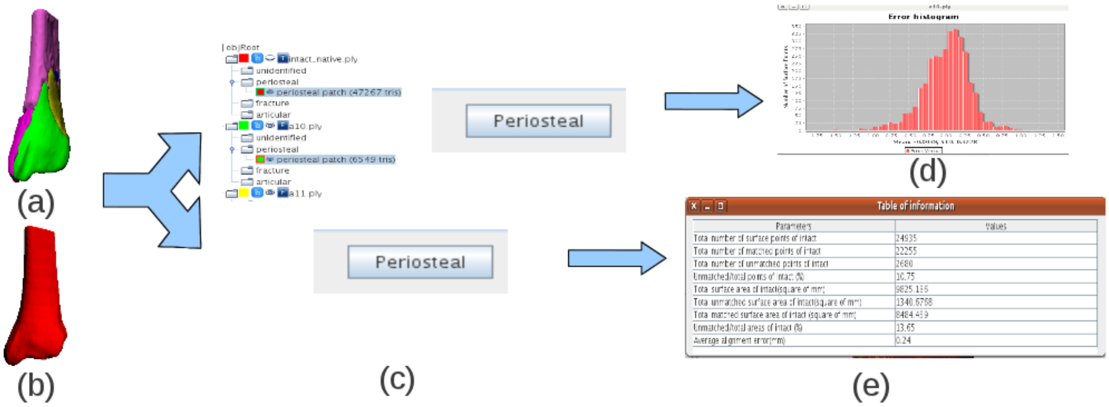
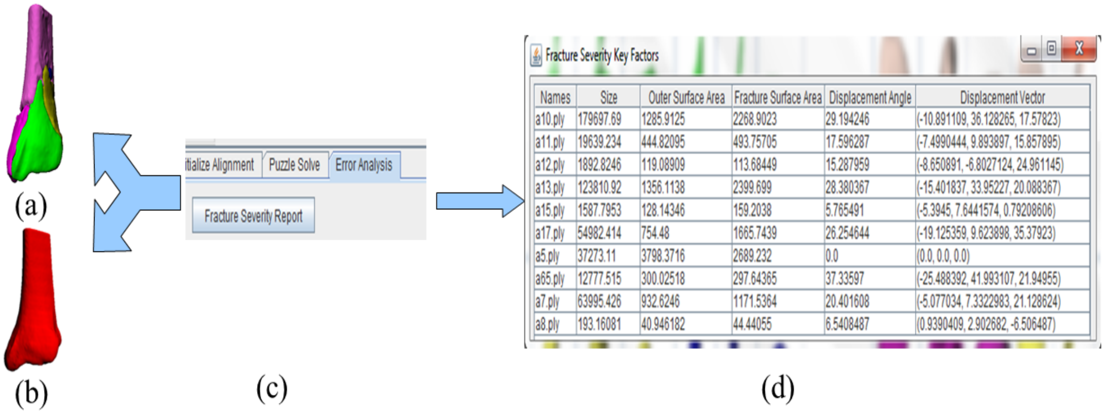


Table of information

Parameters	Values
Tota number of surface points of intact	24935
Tota number of matched points of intact	22255
Tota number of unmatched points of intact	2680
Unmatched/total points of intact (%)	10.75
Tota surface area of intact(square of mm)	9825.136
Tota unmatched surface area of intact(square of mm)	1340.6768
Tota matched surface area of intact (square of mm)	8484.459
Unmatched/total areas of intact (%)	13.65
Average alignment error(mm)	0.24

Figure 2.7: This image shows the interactions that take place in step five. (a) reconstructed fragments, (b) the intact template, (c) a sequence of interface components for reconstruction analysis, (d) a histogram of the alignment error for one fragment, and (e) a table of total alignment error information for a puzzle-solved fracture case.

The system also supports the generation of severity analysis reports which include values for several important key factors that are known to be linked to fracture severity. Surgeons may want this information to assist them in making reliable fracture severity classification. As such, this is an essential part of the entire system. Severity analysis reports include quantitative values for the following key factors: fragment surface area, the total number of fragments, 3D volume of bone fragment surfaces, fragment fracture surface areas, fragment displacements, and fragment angular dislocations for each case. An example of a severity analysis report is shown in figure 2.8. It contains a table having these key factor values and can be generated by clicking the button “Fracture Severity Report” under the “Error Analysis” tab.



Names	Size	Outer Surface Area	Fracture Surface Area	Displacement Angle	Displacement Vector
a10.ply	179697.69	1285.9125	2268.9023	29.194246	(-10.891109, 36.128265, 17.57823)
a11.ply	19639.234	444.82095	493.75705	17.596287	(-7.4990444, 9.893897, 15.857895)
a12.ply	1892.8246	119.08909	113.68449	15.287959	(-8.650891, -6.8027124, 24.961145)
a13.ply	123810.92	1356.1138	2399.699	28.380367	(-15.401837, 33.95227, 20.088367)
a15.ply	1587.7953	128.14346	159.2038	5.765491	(-5.3945, 7.6441574, 0.79208606)
a17.ply	54982.414	754.48	1665.7439	26.254644	(-19.125359, 9.623898, 35.37923)
a5.ply	37273.11	3798.3716	2689.232	0.0	(0.0, 0.0, 0.0)
a65.ply	12777.515	300.02518	297.64365	37.33597	(-25.488392, 41.993107, 21.94955)
a7.ply	63995.426	932.6246	1171.5364	20.401608	(-5.077034, 7.3322983, 21.128624)
a8.ply	193.16081	40.946182	44.44055	6.5408487	(0.9390409, 2.902682, -6.506487)

Figure 2.8: This image shows the interactions necessary in step five to generate a report having values for several fracture severity key factors. (a) reconstructed fragments, (b) the intact template, (c) the interface components used to generate a severity analysis report, and (d) a severity analysis showing values for key factors that impact the fracture severity for one case.

2.1 Previous Work: Digital Image Analysis Tools For Orthopedic Trauma

The literature regarding virtual bone fragment reconstruction systems is relatively sparse in comparison to the very large body of research on bone segmentation, surface alignment and medical image registration.

Work in [18] introduces a fracture surface reconstruction system for reconstructing two-fragment femur bone fractures. In this work, they align the distal and proximal[1] femur fragments based on the *periaxial rotation angle*, i.e., the angle between reference plane and neck axis (see figure 2.9 for a detailed description), computed from both fractured femur and healthy femur. Like the system proposed in this dissertation, they use the patient's symmetric healthy femur as a reference model. Their system is designed to be used during surgery to provide the surgeon real-time values for the periaxial rotation angle difference between fractured femur and the healthy femur. However, their system is intended for use on simple two-fragment fractures and cannot deal with highly comminuted fractures. The impact of this work is diminished by the fact that these fractures tend to have good prognosis without computer aided tools.

[1] See ”proximal” in Glossary

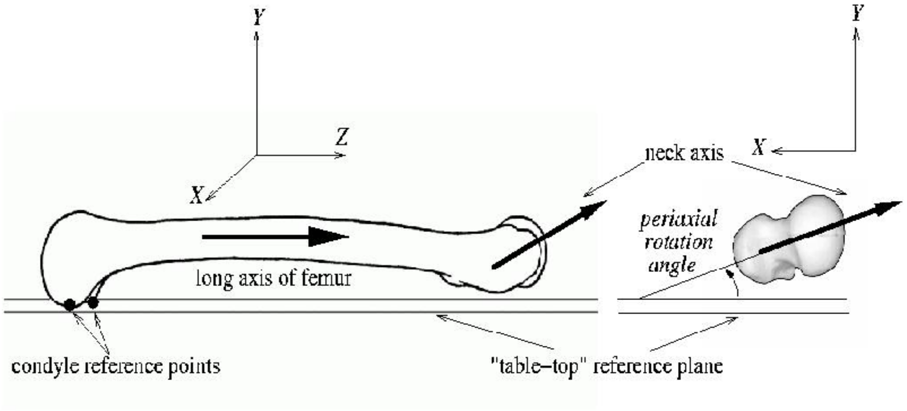


Figure 2.9: Definition of periaxial rotation: lateral (left) and frontal (right) views. The XYZ orthogonal coordinate system in which the angle is measured is defined as follows: the XZ plane is parallel to the table top plane, which is the plane containing the condyle reference points (see the two points in the figure) and parallel to the femur long axis. The Z axis coincides with the long axis of the femur. The Y axis is perpendicular to the XZ plane and is oriented upwards. The periaxial rotation angle is the angle between the projection of the neck axis in the XY plane measured and the XZ (table top) plane [18].

Recent work has focused on image-based reconstruction systems such as [19], or puzzle-solving algorithms such as [20]. The application developed in [20] supports volumetric collision detection in a virtual 3D environment and an optimization process for repositioning the bone fragments. The approach segments bone surfaces out from CT data, then users manually position fragments to be close together. At this point, an automated alignment algorithm refines the fragment positions by moving one fragment to the fixed fragment until a criterion that measures the gap between the two fragments becomes maximum. The criterion is to maximize the number of voxels in *a volume of interest*, i.e., a box-shaped region that has been manually specified to include the overlapping regions of each fragment pair. Since volumes of interest must be manually specified for each match, this system requires a significant amount of time-consuming user interaction to generate a correct reconstruction result. These interactions include identifying matching pairs of fragments, and positioning each pair of fragments to be close together such that their alignment algorithms can be applied.

The system described in this dissertation requires less time and user interactions to generate a correct reconstruction (see table 5.3 in Chapter 5). In particular, there is significant time savings afforded since fragments are aligned automatically and do not need to be manually manipulated.

Recent notable methods for virtually 3D reconstruction of displaced acetabular fractures have been developed to aid surgical interventions [21]. The system introduced in [21] consists of two tools: (1) a 3D viewing tool and (2) a surgical simulation tool. In this case, the 3D CT image data must be manually segmented by expert users. Surgeons then perform virtual reconstruction on the resulting 3D fragments. Unfortunately, this approach also does not include an algorithm to automate or to assist in the difficult process of piecing the bone fragments back together. In order to achieve an acceptable 3D reconstruction result, surgeons must manually make small refinements to fragment positions in their system. This process can be very time consuming and frustrating to the surgeon for even simple fragment fracture cases. In addition, due to the manual nature of the alignment process virtual reconstruction results completed by surgeons in their system are not repeatable.

There are some efforts [22, 23, 24] being made to automatically aid the assembly problem with computer vision algorithms. [22] investigates the specific problem of automated reconstruction of *craniofacial* fractures, i.e., fractures of the head and facial structures. Unfortunately, the method specified in [22] cannot be applied to high-energy comminuted fractures where single fragment surfaces can match to more than one fragment, because their method assumes that each fragment has only one matching fracture surface. Recent work in [23] develops a method for semi-automatic reconstruction of highly comminuted bone fractures to aid in treatment planning. Work in [24] improves upon [23] by refining the alignment algorithm. This is accomplished by emphasizing geometric surface variations (ridges and valleys) during the alignment to more heavily influence the final reconstruction solution, generating more visually encouraging results. However, their algorithm is only tested on simulated bone material fabricated from a specialty high-density polymeric foam. Fracture surfaces generated from these 3D fragments have less variation and are easier to match than the fracture surfaces of fragments extracted from real-world clinical CT images which is the focus of this dissertation.

2.2 Previous Work: Commercial Orthopedic Digital Image Analysis Systems

In addition to the systems generated through research, there are three commercial orthopedic digital image analysis systems capable of analyzing bone fractures: (1) OrthoView (Orthopedic Digital Planning, USA), (2) TraumaCad (VoyantHealth,USA), and (2) Mimics software system (Materialise, Belgium).

OrthoView and TraumaCad are two software systems for digital medical image analysis that are designed especially for orthopedic applications. These systems allow users to carry out virtual preoperative planning sessions on-screen with 2D digital images. Both systems have some common benefits:

1. Users may make measurements of various anatomic features from the digital images.
2. Users can easily access and share these images with PACS[2] .
3. Users can visualize fracture patterns in 2D X-ray images.
4. Users can design custom implants in 2D to use in their surgeries.
5. Users can generate pre-surgical reports that specify the instruments needed and a treatment plan.
6. Users can reconstruct simple fracture cases from 2-D images.
7. Users can plan and manage fracture cases in these systems.

The full spectrum of the OrthoView applications includes five planning modules: (1) joint replacement, (2) limb deformity correction, (3) pediatrics, (4) fracture management, and (5) spine. While TraumaCad consists of nine specialist modules: (1) adult

[2] See “PACS” in glossary.

hip joint replacement, (2) pediatric hip joint, (3) deformity correction, (4) spine, (5) foot and ankle, (6) adult knee joint replacement, (7) upper limbs, (8) trauma, and (9) 3D implant visualization.

The major contribution of these two systems is that they replace traditional X-ray image planning by performing this process digitally. Typically, aspects of this process include editing, marking, and measuring X-ray images which may now be done digitally, using a method is more convenient as one can store, archive and share these results more easily. *Unlike* the system in this dissertation, these applications exclusively use 2D images to help surgeons plan their cases, no 3D objects and reconstruction solutions are supported. The 3D displays are perceived to be more natural and possibly require less mental integration than 2D displays. In some fracture cases such as highly comminuted fractures, the limited view in 2D images can prevent surgeons from identifying fracture patterns and can adversely impact surgical treatment planning. Their systems are not designed for solving these complex fractures.

The Mimics software system (Materialise, Belgium) allows users to process and edit 2D image data and construct 3D models for fractures. The underlying image segmentation tools enable users to segment digital medical images, and take measurements directly on 3D models. From their system, one can export generated 3D data to a wide range of output formats so that downstream applications such as implant design software or surgical simulation software can use the result as input. The goal of the Mimics system is to bridge the gap between 2D image data and these downstream 3D software products. However, while their system can generate 3D models, no further efforts are made to reconstruct these 3D models automatically or semi-automatically. The work described in this dissertation is distinct from the Mimics system in several ways:

1. This dissertation provides a complete analysis and reconstruction process from CT images.

2. This dissertation supports automated 3D virtual fragment reconstruction.
3. This dissertation provides a user-friendly interface to view, modify and manipulate both 2D and 3D data and includes specific tools needed for bone fracture reconstruction.
4. This dissertation integrates several state-of-the-art image and surface segmentation tools into the system that saves users time.

These attributes set the software discussed in this dissertation apart from all existing tools in industry or academia. The specialized tools for bone reconstruction and fracture analysis are unprecedented in either of these contexts and represents a substantial contribution in the area of medical image analysis for complex bone fractures.

The system described in this dissertation includes several improvements over previously proposed systems. It provides unique visualization capabilities for both 2D and 3D fracture management. It supports a structured process for 3D bone reconstruction. It is an unique combination of state-of-art 2D/3D image processing and surface processing algorithms together with an interface that is appropriate for non-expert users. It also enables users to perform a complete analysis of a fracture case from beginning, i.e., the raw CT image data, to the end, i.e., the quantitative severity analysis information for the fracture case. The system also provides tools that link the 2D CT imagery with 3D visualizations of the bone surfaces inside a single interface. This can help users better understand the complex information in these images (see examples in Chapter 5).

CHAPTER 3: ALGORITHMS TECHNICAL SPECIFICATION

Implementation of the bone reconstruction system requires application-specific solutions for five distinct areas of ongoing research on 3D images analysis and 3D surfaces. In some cases, the proposed solutions also have an impact beyond medical image and surface analysis and effect related problems in generic image and surface processing. The five specific research areas of interest for this work are:

1. 3D CT image segmentation (see 3.1 section)
2. **3D surface partitioning (see 3.2 section)**
3. Appearance-based 3D surface classification (see 3.3 section)
4. **3D puzzle-solving (see 3.4 section)**
5. **Post analysis (see 3.5 section)**

These steps also specify the sequence of algorithms through which the data flows as each algorithm requires results computed from the previous one. Users interactively invoke these algorithms through the system interface as described in Chapter 2. *The steps, (2) (4) and (5) (those highlighted in bold), are the focus areas of the research effort described in this dissertation.* In order to provide a complete description of the system, the other two steps, (1) and (3), are briefly discussed. Figure 3.1 shows a diagram of the algorithmic sequence for the system which consists of five steps:

1. 3D CT Image Segmentation: The image segmentation algorithm takes as input 3D volumetric CT images and user-specified parameter values (described in Chapter 2), and provides as output a collection of fragment surfaces. The CT images provide data for both the injured, i.e., fractured, limb and the unbroken or intact limb.

2. 3D Surface Partitioning: The surface partitioning algorithm takes as input 3D surfaces generated from (1) and user-specified parameters (described in Chapter 2), and provides as output a collection of surface patches that describe semantically distinct regions of bone fragment surface needed for the reconstruction.
3. Appearance-Based 3D Surface Classification: This algorithm takes as input a collection of surface patches generated from (2) and a collection of user-specified training data (described in Chapter 2), and provides as output a classification for each surface patch into one of the following anatomically meaningful categories: *{fracture, periosteal, articular}*.
4. 3D Puzzle-Solving: This algorithm takes as input the classified surface patches from (3) and the intact template from (1), and provides as output an alignment between each fragment and the intact template to produce a 3D reconstruction of the unbroken bone from its fragments.
5. Post Analysis: This algorithm evaluates the 3D reconstruction result from (4) and generates quantitative information as a severity report which includes values for several key factors which are known to be linked to fracture severity (listed in 1.3).

Mathematically, We denote generic 3D CT image data as $I(x, y, z)$, fracture CT image data as $I_f(x, y, z)$, and intact CT image data as $I_i(x, y, z)$. The bone fragment surface is denoted as $S(\mathbf{v},\mathbf{g})$, where $\mathbf{v}$ is a collection of 3D points and $\mathbf{g}$ is a collection of polygons. The variable $\mathbf{v}$ is a vector containing surface point coordinates(x, y, z). The variable $\mathbf{g}$ is a vector of polygons (triangles) indicated as a triplet of indices which refer to the three surface points in $\mathbf{v}$ used to construct the triangle. The m^{th} surface point on the fragment is denoted as $\mathbf{v}_{\mathrm{m}}$, the n^{th} polygon of the fragment is denoted as $\mathbf{g}_{\mathrm{n}}$. The intact template is denoted as $S^t(\mathbf{v}, \mathbf{g})$ and the k^{th} fragment is denoted as $S^k(\mathbf{v}, \mathbf{g})$. Each bone fragment surface consists of three types surface patches: the periosteal surface, $S_p(\mathbf{v}, \mathbf{g})$, the articular surface, $S_a(\mathbf{v}, \mathbf{g})$, and the fracture surface,

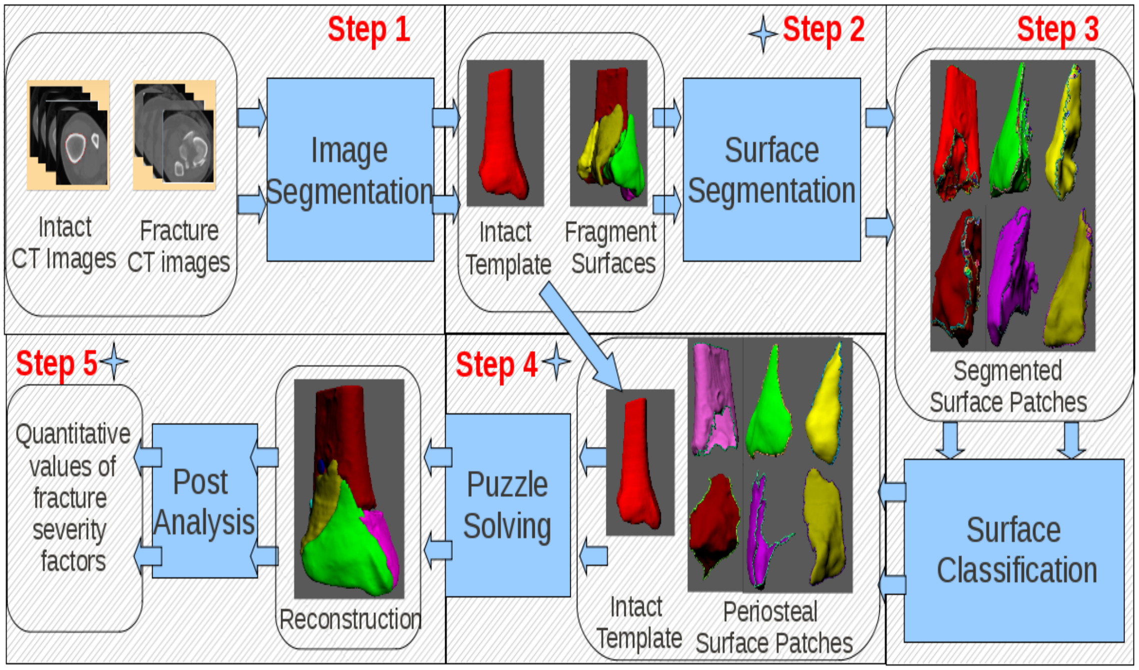


Figure 3.1: A brief overview of a system proposed in this dissertation. This system takes as input a 3D CT image of the fractured limb and a 3D CT image of the undamaged (intact) limb, and provides as output a virtual reconstruction of bone fragments which estimates the anatomy of the patient's original bone. Each block denotes a step and the associate algorithm name is shown inside the block (in blue). Images show how the data has been changed in each step of the 5-step reconstruction process. Gray blocks show interim results at each step and also denote instances where the user can interact with the result as discussed in §2.

$S_f(\mathbf{v}, \mathbf{g})$. In this dissertation, the union of the periosteal surface and the articular surface are referred to as the outer surface and denoted as $S_o(\mathbf{v}, \mathbf{g})$. To simplify the discussion, we refer to the intact and fragment surfaces without $(\mathbf{v}, \mathbf{g})$ parameters as the specific values for the points and polygons are irrelevant to most discussion topics.

3.1 CT Image Segmentation

Reconstruction of a 3D solid from its fragments is a geometric problem where the geometry of the fragments must be known in order to piece them back together. For this reason geometric models for each bone fragment in the 3D CT image must be computed. This is the goal of the CT image segmentation process. The segmentation algorithm used transforms the CT image $I(x, y, z)$ into $I^{'}(x, y, z)$ where image intensity for all pixels in $I(x, y, z)$ are either -1 or 1 where -1 indicates the background and 1 indicates the presence of bone tissue. The goal of the surface segmentation algorithm is to estimate the correct label for every pixel in $I(x, y, z)$. Fragment surfaces may be extracted from the labeled image $I^{'}(x, y, z)$ by estimating the locations where the image changes value from -1 to 1, or equivalently, solving for the locations where $I^{'}(x, y, z) = 0$. This dissertation adopts the approach described in [25] to segment the image which uses a modified watershed algorithm [26]. Bone fragment surfaces are then extracted using the matching cubes algorithm [17]. There are two major challenges for this segmentation problem:

1. Bone fragment boundaries are especially difficult to demarcate when fragments are abutting other fragments.
2. Intensities for some bone tissues, i.e., cancellous tissue, are the same as that for some soft tissues which makes them difficult to identify (see figure 3.2.c).

The watershed segmentation approach can be explained using an analogy between the shape of an image and the topographic relief of terrain. Here, image intensity values define the topographic relief, i.e., the intensity of a pixel is interpreted as the

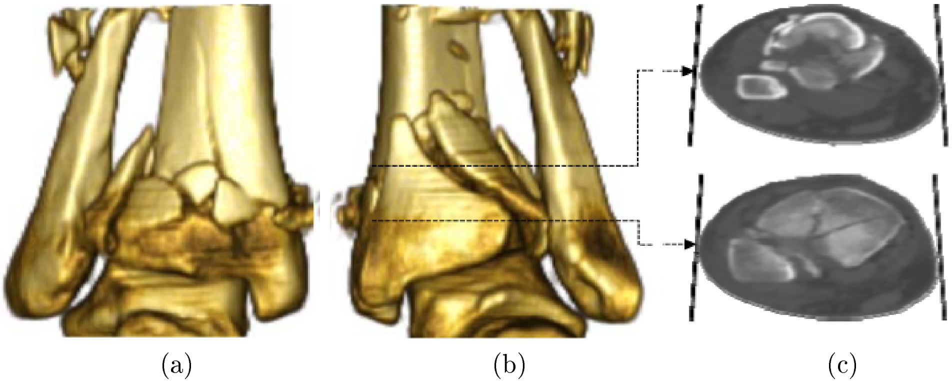


Figure 3.2: Illustrative CT data for a comminuted tibial plafond fracture. (a) a 3D front (coronal) view of tibial plafond fracture. (b) a 3D rear (ventral) view of the tibial plafond fracture. (c) two 2D CT images from two different locations of the fracture as indicated in (b) [25].

altitude of a point. Using this analogy the basins and valleys of the relief correspond to the dark areas of the image, whereas the mountains and crest lines correspond to the bright areas of the image. If one were to pour water onto the topographic relief, the water will collect into the valleys of the relief. The locations where different pools of water settle is referred to as the watersheds of the relief. The watershed algorithm uses this concept as a technique for image segmentation, where watersheds in the image are distinctly segmented objects. Our modified version of the watershed algorithm [25] proceeds as described in the following steps:

1. Define a region of interest within the CT image for each bone fragment.
2. Interactively modify the image data: Each bone and bone fragment in the image is marked to have separate region of interest that is not connected to any other bone or bone fragment region. The intensities of the image are then made to be locally monotonic in the vicinity of each region of interest. This is referred to as imposing a regional minima in each fragment region.
3. Extract the distal tibia from the 3D CT image: A watershed algorithm is applied, and, due to steps 1 and 2, each bone fragment is typically segmented into one or more objects. These objects are automatically classified based on their

locations in the image to be portions of the fibula, talus, or tibia. All objects corresponding to the fibula and talus are removed and the remaining objects (tibia) are merged.

4. Segment the tibia into discrete bone fragments: The watershed algorithm is applied a second time by restarting the algorithm on only the tibia regions, this generates many small regions which cover the tibia bone fragments..
5. Merge adjacent fragment regions to generate bone fragments: In order to identify and correct the over-segmented cortical fragments, an automatic detection and merge algorithm was developed. The algorithm detects regions that should be merged by finding boundaries that are flat/convex.
6. Export meshes: The segmented image volume is processed to generate meshes of the bone fragment surfaces as a collection of triangular facets.

As output, the segmentation process provides bone fragment surface meshes which may be saved to disk in PLY (Stanford's Polygon File Format).

3.1.1 Previous Work

Image segmentation refers to the process of partitioning a digital image into multiple regions. For 2D applications these regions consist of groups of pixels. For 3D applications these regions consist of 3D pixels or voxels. Segmentation algorithms seek to automatically or semi-automatically compute these regions with the intent of generating regions that are semantically meaningful and of interest to the user [27]. In medical imaging, imaging modalities such as MRI and CT provide means for measuring the internal anatomy of a subject. These technologies have become a critical component in diagnosis and treatment planning. Therefore, image segmentation algorithms for the delineation of anatomical structures and other regions of interest within these images are important tools that assist by partially or completely automating these tasks. Segmentation techniques applied on CT images for extracting bone geometries can be categorized into two types: region-based approaches and boundary-based approaches.

Boundary-based segmentation approaches intend to identify bone boundaries in the image. Normal image analysis methods such as active contours [28], graph cut [29], level sets [30], adaptive thresholding [31] and edge detection [32] are adopted by researchers to identify the bone boundaries. Work in [28] modified the traditional snake method for segmentation to segment out tibia and fibula bone structures from a set of CT images. Classical energy functions in the snake model do not hold enough information to tackle the generic bone segmentation problem for CT images. To improve the result a new external energy function was introduced that combines region and gradient information in order to make the snake model more robust to noise. Work in [29] defines a graph over the image by connecting all pairs of neighboring image pixels by weighted edges. With a prior identified object and user specified background seed pixels, the algorithm cuts the edges in the graph with minimum cost so that the objects are separated from the background. Work in [30] adopts the level set segmentation approach for segmenting bone structures from CT scans of the knee joint. Their method is based on intensity profiles along the normal to the evolving contour. The segmentation process is guided by the similarity of image intensity profiles to the manually generated profile model. Work in [31] introduced an automated segmentation algorithm of mid-sagittal brain MR images with two phases of thresholding. The first threshold serves to convert the MR images to binary images which substantially simplifies the subsequent operations. The second threshold is connectivity-based and identifies boundaries between two regions using a path connection algorithm. Work in [32] proposes a multiscale approach to estimate the normal direction of bone edges. The optimal scale at each image location is obtained while estimating the normal direction and then a simple edge detector is applied at this scale for segmentation. Their algorithm improves the segmentation quality by decreasing the number of spurious or noisy edges and decreasing the number of discontinuities for the object contours.

Region-based image segmentation approaches focus on extracting a region of the image that is connected based on some predefined criteria. Region growing is a technique that falls into this category [33]. It requires a seed point that is manually selected by a user and extracts all pixels connected to the initial seed that satisfy a pre-defined compatibility condition. This technique is applied on medical images to delineate small, simple structures such as tumors and lesions [34, 35]. Clustering algorithms are another set of techniques which fall into the region-based category. Three commonly used clustering algorithms are K-means [36], the fuzzy c-mean [37], and the expectation-maximization (EM) [38]. The K-means algorithm [36] clusters data by iteratively computing a mean intensity for each class and segmenting the image by classifying each pixel in the class with the closest mean. The fuzzy c-mean algorithm [37] generalizes the K-means algorithm, allowing for soft segmentation based on fuzzy set theory. The EM algorithm [38] applies the same clustering principles with the underlying assumption that the image intensity data follows a Gaussian mixture model.

The watershed segmentation can be classified as a region-based segmentation approach. It was originally proposed in [39], and later improved in [40]. There are two types of concepts for the watershed idea. One uses a landscape or topographic relief which is flooded by water, watersheds being the dividing lines of the domains of attraction of rain falling over the region [41]. The other uses a landscape immersed in a lake with holes pierced in the local minima. Water fills up the landscape starting at these local minima, and dams are built where the water from different basins meet [26]. Using either approach, the landscape is partitioned into regions by dams and the regions are called watersheds. When simulating this process for image segmentation, these two concepts motivate two different approaches for segmentation: (1) immersion, where one first finds local minima then computes watersheds by taking a set complement [42, 41, 43], and (2) topographical distance, where one computes a complete partition of the image into basins and subsequently computes the watersheds by boundary detection [44, 26, 45, 46]. The detailed discussion of different watershed segmentation algorithms is outside the scope of this dissertation, see the watershed algorithm survey [47] for more details.

3.2 Surface Partitioning

Each segmented bone fragment model must be partitioned into a collection of surface patches to allow portions of the fragment surface to be geometrically matched for 3D reconstruction. There are two approaches to achieve 3D bone reconstruction: (1) matching the outer surface of bone fragments to a model of the intact bone, referred to as the template model, and (2) puzzle-solving by matching the break surfaces, i.e., the surfaces generated when two fragments break apart [23]. Both approaches require the fragment to be sub-divided into anatomically meaningful surface patches. In addition, by dividing the fragment surface into anatomically different regions the amount of searching to find correct surface matches is significantly reduced which improves the matching algorithm performance. The task of surface mesh partitioning is to divide bone fragment meshes, $S^k(\mathbf{v}, \mathbf{g})$, into a collection of surface patches, $\{S_1, S_2 \cdots\}$, such that $S^k = \cup\{S_1, S_2 \cdots\}$. Each of the generated surface patches are intended to consist of surface points from only one anatomical category.

This dissertation introduces two surface partitioning algorithms: (1) a "ridge walking" algorithm [48], and (2) a graph-based algorithm. Due to the complex geometry of human bone structures, current existing surface partitioning algorithms cannot efficiently partition bone surfaces into anatomically meaningful sub-regions. The algorithm heavily used in this dissertation, [48], addresses this problem by dividing the bone surface along 3D contours that traverse high-curvature ridges. The second partitioning algorithm introduced in this dissertation draws inspiration from the idea in [49], which implements an efficient graph-based image segmentation. Specifically, the proposed approach uses the graph idea from [49] to produce a new graph-based 3D surface mesh partition algorithm (see 3.2.2 for implementation details). The overview of the ridge walking algorithm is described in 3.2.1.

3.2.1 Partitioning Surfaces Using The Ridge Walking Algorithm

The ridge walking surface partitioning algorithm [48] works directly from the 3D mesh of the bone surface which consists of 3D surface points and a set of edges that connect these points. The approach defines a graph over the polygonal mesh where graph edges are the mesh edges and graph nodes are the mesh vertices. For each edge of the graph, an attribute is computed that is referred to as the edge salience, $w(\mathbf{o}_{ij})$. The edge salience defines a weight for each graph edge that is large for edges that define lines on the surface that are "salient", and small for all other edges. The salience of an edge is determined by one of three possible functions: (1) a "ridge" salience function, (2) a "valley" salience function and (3) a "curvature " salience function. The different functions are used to partition the surface in different ways. The "ridge" salience function tends to follow convex ridge-like surface edges. The "valley" salience function tends to follow concave valley-like surface edges. The "curvature" salience function tends to follow both convex and concave surface edges. This dissertation uses the "ridge" salience function to partition the bone fragment surfaces. This function is used because different bone anatomical regions are usually separated by convex edges see figure 3.3. The output of the partitioning method is a collection of disjoint surface patches, each of which is a subset of the fragment surface mesh. The system displays these patches as a part of each fragment surface as different colors and colored boundaries and shows them in the tree panel see figure 2.4. Figure 3.3 shows several results generated by the ridge walking algorithm.

3.2.2 Partitioning Surfaces Using A Graph-Based Algorithm

This new partitioning approach represents a fragment surface as a graph and then applies the segmentation criterion from [49] on the resulting graph. This results in

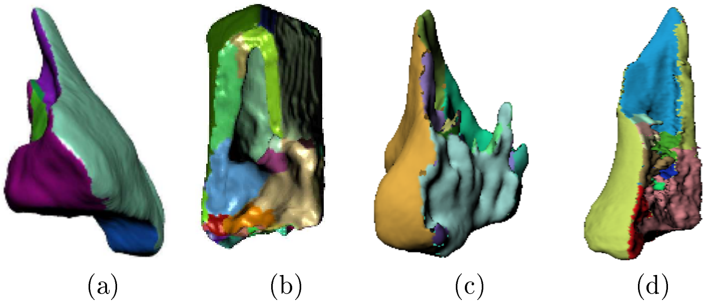


Figure 3.3: Ridge walking surface partition results for four different fragment surfaces.

a partitioning of the surface into different surface patches. This graph-based partitioning algorithm is a 3D generalization of the well known 2D image segmentation approach [49].

The goal of the graph-based image segmentation is to partition the image into different components such that elements in same component are similar and elements in different components are dissimilar. This is achieved by evaluating if there is evidence for a boundary between a pair of components. The approach in [49] constructs the image as graph denoted $G = (V, E)$ where V denotes graph vertices and E denotes graph edges. Each pixel in the image is taken to be a vertex in the graph, $\mathbf{v}_i$, and two neighboring pixels $(\mathbf{v}_i, \mathbf{v}_j)$, form an edge, $\mathbf{o}_{ij}$, in the graph that connects the corresponding vertex pair. The weight of an edge $\mathbf{o}_{ij}$ is denoted $w(\mathbf{o}_{ij})$ and is determined by the dissimilarity of the neighboring two pixels, i.e., intensity difference between the two pixels. A collection of vertices is called a component, C. A score is given for any boundary in the image. It consists of two parts: (1) the *internal difference of a component,* $Int(C)$, which is the largest weight in the minimum spanning tree [1] of the component C, and (2) the *difference between two components,* $Dif(C_1, C_2)$, which is the minimum weight edge connecting the components C_1 and C_2. The task of finding the component boundaries reduces to checking if $Dif(C_1, C_2)$ is large or small relative to the internal differences, $Int(C_1)$ or $Int(C_2)$.

[1] see “minimum spanning tree” in Glossary

The input of the algorithm is a graph $G = (V, E)$, with n vertices and m edges. The output is a segmentation of G into components $S = (C_1, C_2, \ldots C_r)$. This is accomplished by following the steps shown below in [49]:

1. Sort the edges E into π, a collection of m edges, by non-decreasing edge weight.
2. Start with a segmentation S_0, where each vertex $\mathbf{v}_i$ is in its own component.
3. Repeat step 4 for $q = 1, \ldots, m$.
4. Construct S^q given S^{q-1} as follows: Let $\mathbf{v}_i$ and $\mathbf{v}_j$ denote the vertices connected by the q^{th} edge in the ordering, i.e., $\mathbf{o}_q = (\mathbf{v}_i, \mathbf{v}_i)$. If $\mathbf{v}_i$ and $\mathbf{v}_j$ are in disjoint components of S^{q-1} and $w(\mathbf{o}_q)$ is small compared to the internal difference of both those components $(Int(C_1), Int(C_2))$, then merge the two components, otherwise do nothing.
5. Return $S = S^m$.

When components stop merging, each component groups a collection of vertices (image pixels) and forms a segmented region in the image. Vertices in the graph that do not belong to any component are grouped together to form a separate image region referred to as the background.

The new partitioning approach proposed in this dissertation modifies the above approach by replacing the constructed image graph with a graph representation for a 3D surface mesh. Two different methods were explored for constructing the new graph, each of which includes its own dissimilarity functions and provides different partitioning results. One method is referred to as the vertex-based approach which constructs the graph by letting the mesh vertices define the graph nodes, V, and mesh edges define the graph edges, E (see 3.2.2.1 for details). Another method is referred to as the face-based approach which constructs a graph by letting the mesh faces (polygons) define the graph nodes, V, and mesh edges that are shared by two faces define the graph edges, E (see 3.2.2.2 for details).

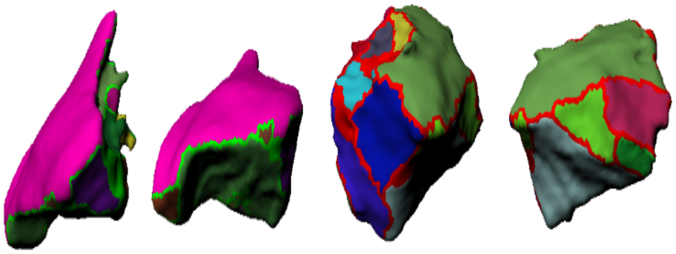


Figure 3.4: Vertex-based graph surface partitionings for two fractured bones in two different views.

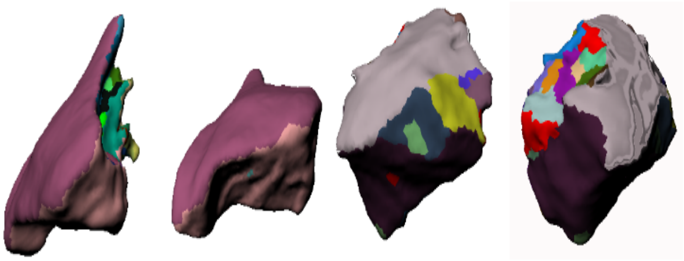


Figure 3.5: Face-based graph surface partitionings for two fractured bones in two different views.

3.2.2.1 Vertex-Based Graph Partitioning Algorithm

The vertex-based graph is constructed by defining each 3D surface point to be a graph node, $\mathbf{v}_i$, and each surface mesh edge connecting vertices $(\mathbf{v}_i, \mathbf{v}_j)$ is defined to be a graph edge, $\mathbf{o}_{ij}$. The edge weight, $w(\mathbf{o}_{ij})$ is defined in equation (3.1):

$$w(\mathbf{o}_{ij}) = \left| \frac{\mathbf{u}_i + \mathbf{u}_j}{|\mathbf{u}_i + \mathbf{u}_j|} \cdot \frac{\mathbf{o}_{ij}}{|\mathbf{o}_{ij}|} \right| |k_{max}| \tag{3.1}$$

Where $\mathbf{u}_i$ is the direction of the minimum principal curvature at $\mathbf{v}_i$ and $\mathbf{u}_j$ is the direction of the minimum principal curvature at $\mathbf{v}_j$. The quantity $\frac{\mathbf{u}_i+\mathbf{u}_j}{|\mathbf{u}_i+\mathbf{u}_j|}$ represents the average minimum curvature direction between the vertex $\mathbf{v}_i$ and $\mathbf{v}_j$, and $\frac{\mathbf{o}_{ij}}{|\mathbf{o}_{ij}|}$ is the mesh edge direction as a unit vector and k_{max} is the average of the maximum curvature values at the vertex $\mathbf{v}_i$ and $\mathbf{v}_j$. Applying the algorithm steps introduced above on this graph partitions the surfaces into different components where each component is a collection of 3D surface points. Figure 3.4 shows bone surface partitioning results for two bone fragments using this algorithm.

3.2.2.2 Face-Based Graph Partitioning Algorithm

The face-based graph is constructed by defining: each 3D surface polygon (triangle) to be a graph node, $\mathbf{v}_i$, and each mesh edge that is shared by two surface polygons, $(\mathbf{v}_i, \mathbf{v}_j)$, is defined to be a graph edge. The edge weight, $w(\mathbf{o}_{ij})$, is defined in equation(3.2):

$$w(\mathbf{o}_{ij}) = \arccos(\mathbf{n}_i \cdot \mathbf{n}_j) \tag{3.2}$$

Where, $\mathbf{n}_i$ and $\mathbf{n}_j$ denote surface normals for polygons $\mathbf{v}_i$ and $\mathbf{v}_j$ respectively. The edge weight is then simply the dihedral angle between the polygons that share the edge. Applying the algorithm steps introduced above on the face-based graph partitions the surfaces into different components where each component is a collection of 3D surface polygons. Figure 3.5 shows bone surface partitioning results for two bone fragments.

There is a difference in the computational logic of these two algorithms. The vertex-based graph partitioning algorithm divides surfaces into different components by grouping together surface points while the face-based algorithm divides surfaces into components by grouping together surface polygons. Therefore, results from vertex-based graphs will result in a collection of surface vertices that don't belong to any components which are used to separate the components (shown in red in figure 3.5). An extra step is needed to classify these vertices which assigns these vertices as members of both of their neighboring components. The performance of graph-based partitioning algorithm is determined by the number of edges in the constructed graph. For the case of a fragment surface graph, the face-based partitioning algorithm tends to have more nodes and edges than those constructed using the vertex-based partitioning algorithm. For this reason, the vertex-based algorithm is often faster than the face-based algorithm when running on the same fragment.

Although two partitioning algorithms are available for partitioning bone fragments, there are cases where both algorithms cannot divide the fragment surface into the desired set of surface patches. This can occur when the bone fragment surface is under-segmented. In this case, partitioned surface patches contain surface geometries having different anatomical types. In these cases, further partitioning of these surface patches is required. In other cases the surface is over-partitioned and includes many small surface patches that share the same anatomical type and must be merged. To overcome these problems, the system provides interactive tools to merge and to subdivide surface patches (see Chapter 2 for details).

3.2.3 Previous Work

As mentioned earlier, a 3D surface mesh partitioning algorithm decomposes a single 3D surface model into a collection of surface patches, where each surface patch is intended to carry some semantic meaning. Recent surveys [50, 51, 52] provide comparative analysis between leading contemporary algorithms. These algorithms [53, 54, 55] are often an extension of well-established algorithms for 2D image segmentation that have been generalized to 3D surfaces. 3D surface partitioning algorithms can be grouped into two general categories: (1) those that group surface regions, and (2) those that divide surfaces via boundaries.

One popular approach from the category that groups surface regions is the region growing approach in [56]. This approach selects a collection of polygons on the surface as “seeds”. A region then forms around each seed by a region merging neighboring polygons into each seeded region using a similarity function and a threshold which jointly define a compatibility criterion that determines if the merge occurs, e.g., the angle between two surface normals. Region growth stops when all neighboring polygons do not satisfy the compatibility criterion. [57] improves the region growing method of [56] by introducing multi-scale surface smoothing, user interactivity and a stochastic filter to speed up growth in flat areas. [58] adapts the well-known watershed segmentation algorithm typically used for images for use on 3D surface meshes. Graph-based cut method described in [59] was also adapted for use on 3D surfaces as described in [60].

Other approaches solve the partitioning problem by solving for surface contours that divide the surface into different regions. The mesh scissoring approach described in [61] generalizes the snake model of [55] such that the boundary curves that divide the surface into parts propagate into concave surface regions. But, as known for snake models in general and also explicitly noted in [50], such methods can sometimes converge to local minima of the curve-evolution performance function which can result in an undesirable surface segmentation. [62] proposes a 3D surface segmentation technique based on level set methods that seek concave surface contours which are solutions to a geodesic curvature flow differential equation. However, user interactions are needed to help the contours to demarcate the surface in a satisfactory way. The randomized cut method described in [60] tries to extract partition boundaries using a statistical approach. First, it generates a random set of mesh segmentations using graph cut algorithms and then measures how often each edge of the mesh lies on a segmentation boundary in the resulting set. The partition function defined on edges seeks to find the set of most consistent cuts which provide a continuous measure of where natural part boundaries might occur in a mesh.

The ridge walking algorithm [48] adopted in this dissertation is related to [61] since they both directly solve for contours on the surface that satisfy salience criteria. The approach in [61] performs well only on CAD models[2] where their geometries follow the minima rule. The criteria for distinguishing anatomic parts of bone fragments are different than those for distinguishing parts of models partitioned using the minima rule. This make the ridge walking algorithm particularly well-suited to the application of bone fragment surface partitioning because one can guarantee that the boundary of the partitioned bone fragment surface patches will satisfy specific salience criterion such as following the convex-like ridges.

[2] see “CAD models” in Glossary

3.3 Appearance-Based 3D Surface Classification

The surface classification process takes as input surface patches extracted by partitioning the bone fragment surfaces and seeks to estimate a single anatomic label for each surface patch. The anatomic labels are critical to reconstruction as they determine which surface patches can be matched and how to match them. In this dissertation, the anatomic labels are (1) "fracture surfaces" (surfaces generated when bone fragment broke apart), (2) "periosteal surfaces" (surfaces that were part of the outer bone surface before the fracture occurred), (3)"articular surfaces" (surfaces of bones that facilitate the articulation of joints). Classification of these surface patches is useful for bone reconstruction for the following reasons: (1) periosteal surfaces can be used to facilitate template-based reconstruction (the approach used in this dissertation), (2) fracture surfaces facilitate reconstruction via fragment matching, (an approach used in [23]), and (3) articular surfaces are sub-regions of the periosteal surface especially important in reconstruction. Accurate reconstruction of the articular surfaces are critical for achieving a good outcome for fractures at joint locations. The work in [1] has shown that high accuracy of the surgical bone reconstruction for the articular surfaces is critical in avoiding future onset of PTOA.

The goal of the surface classification step is to classify each surface patch to one of three anatomic surface categories : (1) the periosteal surface, $S_p(\mathbf{v}, \mathbf{g})$, (2) the fracture surface, $S_f(\mathbf{v}, \mathbf{g})$, and (3) the articular surface, $S_a(\mathbf{v}, \mathbf{g})$. After classification surface patches having the same type are merged into one surface patch. Mathematically, the surface classification algorithm takes in a collection of surface patches $S^k = \cup\{S_1, S_2 \cdots\}$ and outputs a classified bone fragment $S^k = \cup\{S_f, S_p, S_a\}$. In this dissertation, we only use the periosteal surface, S_p, and the articular surface, S_a, for reconstruction which jointly define the outer (intact) surface, S_o. The outer (intact) surface of the k^{th} fragment is denoted as S_o^k.

The surface classification approach is based on the idea that different regions of a

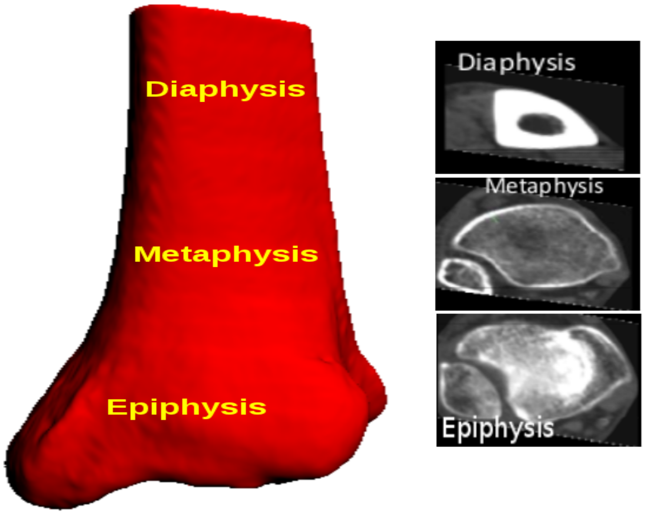
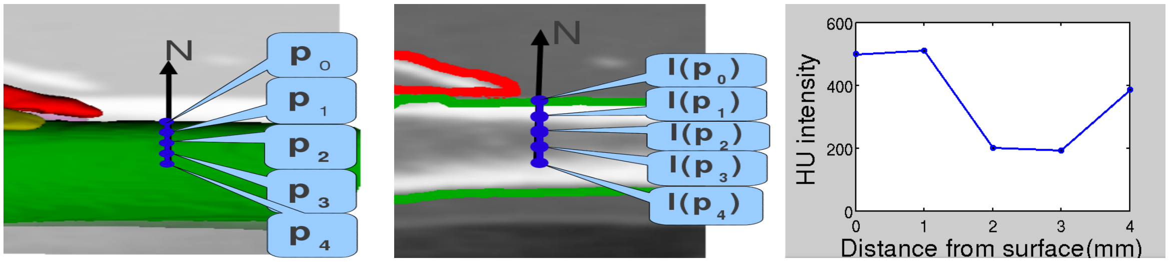


Figure 3.6: CT appearance of the distal tibia anatomy. Relative to the tibia's outer surface, the CT intensities vary with characteristic profiles along the inward surface normals for each anatomic region (diaphysis, metaphysis and epiphsis).

bone consist of different tissues and the appearance of these tissues in medical images have distinct properties as one traverses both up and down (proximally and distally) and as one traverses from the outer bone surface into the interior. As shown in figure 3.6, the diaphysis is made up of solid dense cortical bone having high intensities. The metaphysis is made up of an outer cortical shell having high intensities and a less dense and porous cancellous bone on the interior having lower intensities. The epiphysis is at the end of the long bone in the vicinity of the articular surface and consists of relatively dense subchondral bone having intensities that lie in between cortical and cancellous bone tissues. The proposed approach uses these intensity variations present in the 3D CT images to classify the partitioned surface patches to the different anatomic labels.

Surface patch classification starts by classifying the surface vertices of each patch. This is done by associating each bone fragment surface vertex with a collection of intensities as attributes. Since these intensity attributes are derived from the appearance of the bone tissues in the CT image that are in the vicinity of the surface patch, we refer to these attributes as appearance attributes. Each surface patch vertex is



(a) (b) (c)

Figure 3.7: (a) Shows the process of "drilling down" into the green fragment surface along the negated direction of the surface normal, N, of the surface point, $\mathbf{p}_0$. Note that these positions are actually inside the fragment and surface geometries are registered with CT scans. (b) Shows how to compute the intensity of each position.(c) Plots the computed intensity values as a function of distance along the inward pointing surface normal.

assigned with a sequence of appearance attributes that are taken as the sequence of intensities that occur as one traverses the CT image by following the vertex surface normal from the surface patch vertex into the bone interior. The sequence of the CT intensity values recorded along this path is referred to as a *CT-profile*. Each CT-profile describes a distinct sequence of CT intensity values generated by taking measurements at a set of prescribed distances as one moves at constant increments into the bone interior from the outer surface. It is computed by interpolating the CT intensities at each of these positions, $\mathbf{p}_k$, for each surface point. The sequence starts at the surface vertex itself, $\mathbf{p}_0$, by interpolating the CT intensity at that point, $I(\mathbf{p}_0)$. M additional intensities values are computed by moving along the direction of the negated (inward pointing) surface normal at that point in equal amounts, $\triangle s$, the value of $\triangle s$ is computed from the resolution of the provided CT data. The CT intensity value at the k^{th} position, $I(\mathbf{p}_k)$, is computed by performing trilinear interpolation within the registered 3D CT volume. This creates a vector of $M{+}1$ CT intensity values for each surface patch vertex which is referred to as CT-profile for that surface point. Equation (3.3) describes the CT-profile vector for the i^{th} surface vertex, $\mathbf{v}_i$, and requires the surface normal at that vertex denoted $\mathbf{n}_i$:

$$\mathbf{X} = \left(I\left(\mathbf{p}_0\right), I\left(\mathbf{p}_1\right), I\left(\mathbf{p}_2\right), \cdots, I\left(\mathbf{p}_M\right)\right)^t; \tag{3.3}$$

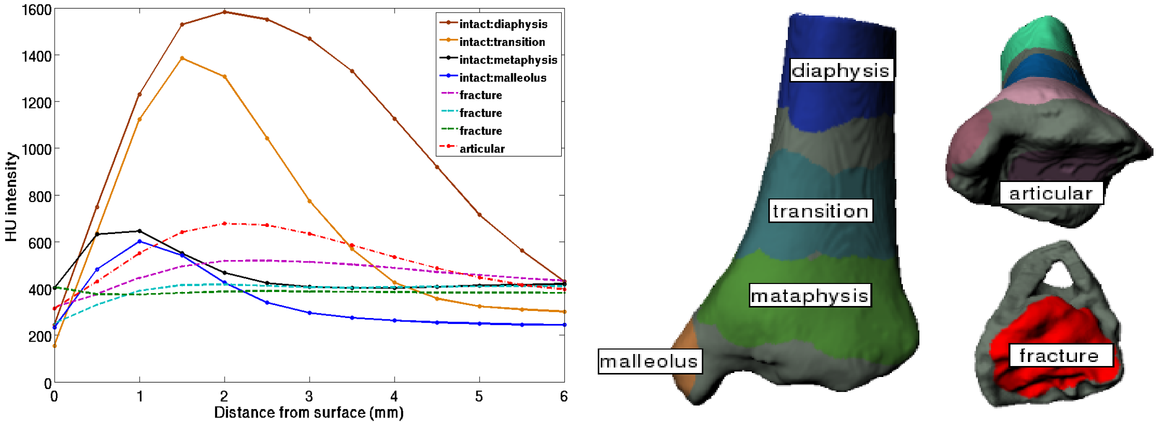
where $\mathbf{p}_k = \mathbf{v}_i - k\triangle s\mathbf{n}_i$ and $k = 0, 1, 2 \ldots M$. Figure 3.7 shows graphically how a CT-profile is computed. For the measurements in the figure $\triangle s = 1mm$ and $M = 4$.

The CT-profile encodes the anatomical tissue variations present in the bone as a function of depth along the surface normal as a sequence of computed intensity values. Physiological variations in typical bone anatomy allow for these intensities to be used to detect the approximate anatomic type and location of the surface vertex within the bone. CT-profiles from the proximal tibia, i.e., the diaphysis, generate curves that have the largest intensity peak which typically occurs about 1-2 mm from the fragment surface. CT-profiles from the medial tibia, i.e., the metaphysis, also typically have a peak after 1-2 mm however this peak is approximately 25% of the magnitude of the diaphyseal intensity peak. CT-profiles from distal tibia, i.e., the articular region, have a profile that typically lies in-between the diaphyseal and metaphyseal profiles. Figure 3.8 shows the behavior of CT-profiles as one traverses from the diaphysis to the articular regions of the bone (from top to bottom).

The specific intensity variations that occur within a given image can vary widely due to several factors: (1) patient age/gender, (2) the imaging equipment, and (3) patient height/weight. In fact, the studies [63, 64] have shown that bone intensity of human beings can significantly vary due to these factors. Typically, men tend to have more dense bone tissues than women, and young people tend to have stronger and harder bones than older people. In addition, bone intensities recorded with different imaging equipment can also vary significantly. Due to these effects, the classification systems must be trained on each patient and CT machine to provide reliable classification outputs.

The classification process is described in the following sequence of steps:

1. The user specifies training data,



(a) (b)

Figure 3.8: (a) Shows the CT-profile appearance models for intact, fracture, and articular surface regions which consist of a collection of eight different CT-profiles. (b) Shows a collection of surface regions interactively selected as training data necessary to form the CT-profile appearance models in (a).

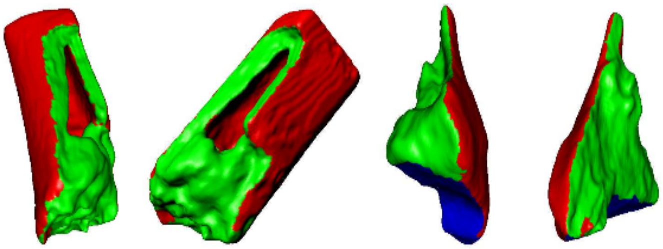


Figure 3.9: Two views of the surface mesh classification results for two fragments. Red surface patches denote the periosteal surface. Green patches denote the fracture surface, and blue patches denote the articular surface.

2. A classifier is constructed from the training data,
3. The classifier classifies the surface points of each surface patch and assigns an anatomic label to each surface patch,
4. Adjacent surface patches having the same anatomic label are merged.

Since the surface classification is not a contribution of this dissertation, the details of its implementation is not discussed here. Figure 3.9 shows classification results for two bone fragments using this approach.

3.3.1 Previous Work

Other appearance-based classification systems [65, 66, 67] are widely used for matching and tracking faces and for medical image segmentation. Work in [65] proposed a generic appearance-based classification algorithm that uses a spectral representation for images. They partition the frequency domain of an image into small disjoint regions and derive a set of filters for these regions. A filter selection algorithm is proposed to maximize the classification performance over the training data. The main idea of this algorithm is to extract translation invariant statistical features using the frequency domain representation which can be used to classify objects in images based on their appearance. Work in [67] introduced a technique for frontal face detection in color images based on facial feature extraction and appearance-based classification. This algorithm first performs skin detection using a statistical skin model to derive a bounding box region for human face area in the image. Then mouth and eye feature points are detected within this region using the statistical, geometrical, and structural properties of eyes and mouth in frontal face images. Their final step of detecting a face in the image uses a multivariate normal distribution as a model for facial features as originally proposed by [68] for face detection in gray-scale images. Work in [66] extends the idea of modeling shape and local appearance to locate flexible objects in the images. This approach proposes a novel method for interpreting images that uses an Active Appearance Model (AAM). During training phase, the relationship between the model parameter displacements and the residual errors between a training image and a synthesized model example is derived. To match to an image they measure the current residuals and use the model to predict changes to the current parameters, leading to a better fit. Their approach can be used to do object tracking and recognition in various applications.

Due to the specificity and difficulty of the problem, there are few examples of standard algorithms that segment and classify fragmented bone surfaces. Existing reconstruction approaches rely on manual segmentation as provided by manually identifying the 2D contour that bounds the bone tissues of each fracture fragment in each slice of a 3D CT image [69]. Work such as that in [23], perform automatic surface extraction by using a threshold to segment bone fragments from the CT images. Unfortunately, such a simple method for segmentation of the bone fragment surfaces fails for almost all fragments especially those fragments which include cancellous or subchondral bone tissue. These fragments have subtle intensity variations and often have intensities similar to other image structures that are not bone tissue such as cartilage and soft tissue. For this reason one cannot use simple thresholding for bone fragment classification.

3.4 3D Puzzle-Solving

The 3D puzzle-solving algorithm takes as input a collection of bone fragments which have been partitioned into surface patches where each surface patch has been classified as an intact surface or a fracture surface. As output, it provides a 3D virtual reconstruction of the bone fragments which approximate the anatomy of the unbroken bone. The intact contra-lateral bone, as represented in the intact CT image, is taken as a reasonable approximation of the unbroken bone. Bone fragment surfaces are matched with this surface to reconstruct the fracture [70]. Specifically, this dissertation assumes that aligning the k^{th} fragment's intact surface, S_o^k, to the intact template, S^t, will restore the k^{th} fragment to its original position in the unbroken bone anatomy. Since the development of the 3D puzzle solving algorithm is the main contribution of this dissertation, the introduction and background details for the 3D puzzle solving algorithm are discussed in Chapter 4.

3.5 Post-Reconstruction Analysis

The post-reconstruction analysis is the final output of the system. The analysis tools are integrated into the system and allow users to analyze the reconstruction result and help users better understand the fracture case. The analysis consists of

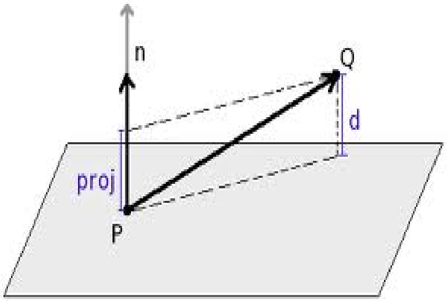
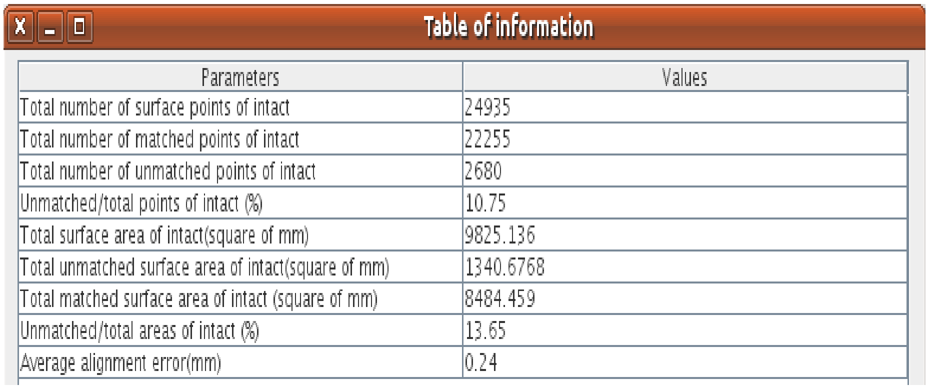


Figure 3.10: This figure describes the computation of the point-to-plane distance. Let Q denote a point on the surface of a fragment. Let P denote the closest point on the surface of the intact template and n denote the surface normal at that point. The figure shows a geometric representation for the point-to-plane distance from Q to P, given as d in the figure.

two components: (a) analysis of the geometric accuracy of the aligned fragments and (b) analysis of the severity of the bone fracture. The first component provides a table containing alignment information and a histogram of alignment error for each fragment. The second component is a fracture severity report which contains quantitative values for several key factors for each fragment which are known to be indicators of fracture severity. Although visual assessment of 3D reconstruction result is valuable to users, these analysis tools provide quantitative information which can help users objectively interpret the fracture case.

The geometric accuracy of the aligned fragments is one aspect of the reconstruction of interest. Analysis of the geometric accuracy of the reconstruction is accomplished by measuring the distance metric between the outer surfaces of the reconstructed fragments and the surface of the intact template. These distances are averaged to compute alignment error for each fragment. Each distance is computed as the point-to-plane distance, i.e., the Euclidean distance from a vertex on the fragment outer surface to the plane of the closest triangle on the surface of the intact template as shown in figure 3.10. The post-reconstruction analysis report consists of a table of global alignment information and a table of histograms that show the distribution of



Parameters	Values
Total number of surface points of intact	24935
Total number of matched points of intact	22255
Total number of unmatched points of intact	2680
Unmatched/total points of intact (%)	10.75
Total surface area of intact(square of mm)	9825.136
Total unmatched surface area of intact(square of mm)	1340.6768
Total matched surface area of intact (square of mm)	8484.459
Unmatched/total areas of intact (%)	13.65
Average alignment error(mm)	0.24

Figure 3.11: The figure shows a table displaying a list of alignment analysis parameters and their values for a specific fracture case.

the point-to-plane of alignment errors for each fragment.

Figure 3.11 shows an example of the global alignment analysis table. In the table, numerical values are given for the following quantities:

1. Number of points on the intact surface model (samples).
2. Number of matched points on the intact surface model (samples).
3. Number of unmatched points on the intact surface model (samples).
4. Percentage of unmatched surface points as a proportion of the total number of points on the intact model (%).
5. Surface area of the intact model (mm^2).
6. Matched surface area of the intact model (mm^2).
7. Unmatched surface area of the intact model (mm^2).
8. Percentage of unmatched surface area as a proportion of the total surface area of the intact model (%).
9. Average global alignment error for this fracture (mm).

These quantities assist users in understanding the reconstruction quality in terms of the geometric accuracy. The first four quantities allow the user to understand the global matching relationships between the intact template and all of the bone fragments. The reconstruction hypothesis assumes the fragment outer surfaces, when reconstructed, will “cover up” the entire surface of the intact template. If true, any surface patches on the intact template left “uncovered” indicate possible locations for fragment mis-alignment, fragment surface deformations or missing fragments. See 4.4.2 for details about the “covered” region on the intact template is computed.

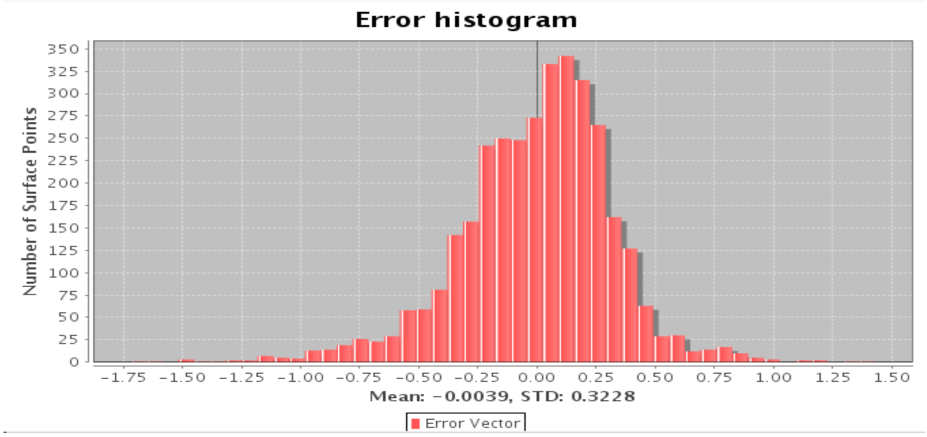
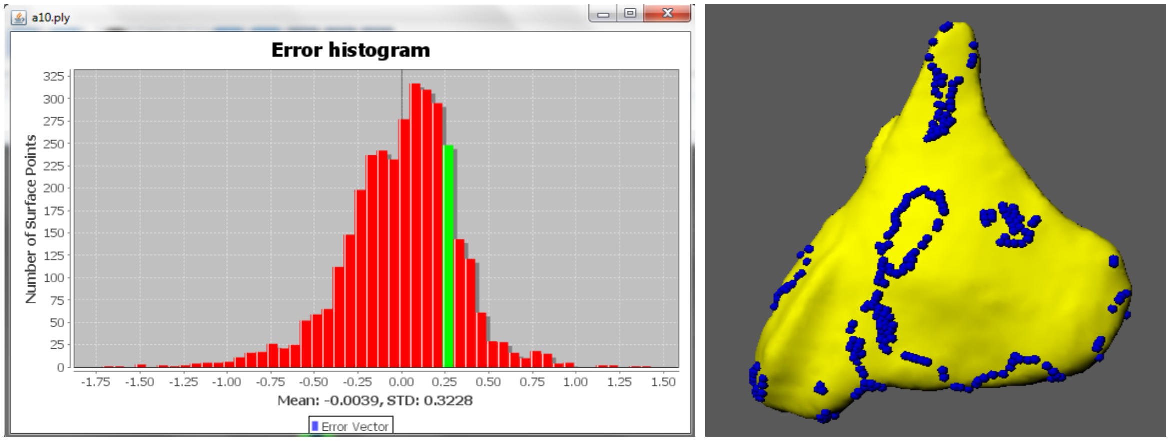


Figure 3.12: The figure shows a histogram of alignment error for one fragment in a fracture case.

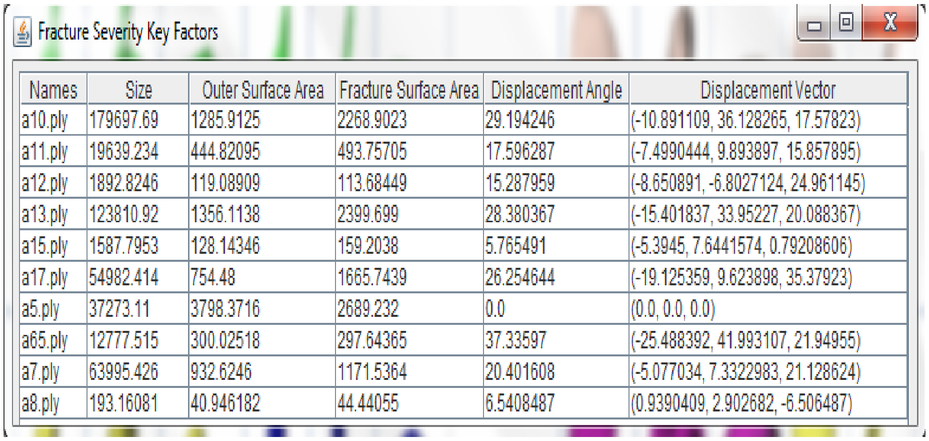


(a)

(b)

Figure 3.13: (a) shows a histogram of alignment errors where the bin of vertices having an error of approximately 0.25 *mm* has been selected (green). (b) shows a visualization of the spatial distribution surface points (blue spheres) which have these errors on the fragment surface.

The figure 3.12 shows an example of an alignment error of the histogram for fragment A10 in fracture case one (see Chapter 5). The value of the x-axis is the observed values of the alignment error (in *mm*) and the value of the y-axis is the



Names	Size	Outer Surface Area	Fracture Surface Area	Displacement Angle	Displacement Vector
a10.ply	179697.69	1285.9125	2268.9023	29.194246	(-10.891109, 36.128265, 17.57823)
a11.ply	19639.234	444.82095	493.75705	17.596287	(-7.4990444, 9.893897, 15.857895)
a12.ply	1892.8246	119.08909	113.68449	15.287959	(-8.650891, -6.8027124, 24.961145)
a13.ply	123810.92	1356.1138	2399.699	28.380367	(-15.401837, 33.95227, 20.088367)
a15.ply	1587.7953	128.14346	159.2038	5.765491	(-5.3945, 7.6441574, 0.79208606)
a17.ply	54982.414	754.48	1665.7439	26.254644	(-19.125359, 9.623898, 35.37923)
a5.ply	37273.11	3798.3716	2689.232	0.0	(0.0, 0.0, 0.0)
a65.ply	12777.515	300.02518	297.64365	37.33597	(-25.488392, 41.993107, 21.94955)
a7.ply	63995.426	932.6246	1171.5364	20.401608	(-5.077034, 7.3322983, 21.128624)
a8.ply	193.16081	40.946182	44.44055	6.5408487	(0.9390409, 2.902682, -6.506487)

Figure 3.14: A severity assessment report is shown which includes values for several quantities that are known to be key factors in determining fracture severity.

number of fragment surface points whose alignment error falls into this range. Below the plot of the histogram, the statistical information indicating the mean and standard deviation of the observed alignment errors are also provided. The histogram of the alignment error provides users more detailed statistical information that relate to the quality of the fragment alignment. Moreover, the system makes it possible to visualize the location and magnitude of the alignment errors as a spatial distribution on the fragment surface. Interactions are available from the histogram plot which allow the user to select a range of error values. Once selected, the errors associated with these values are visualized spatially across the surface of the fragment as shown in figure 3.13. These tools are valuable for understanding the quality of the geometric reconstruction results and understanding the complex geometric inter-dependencies of a highly-fragmented bone fracture.

As mentioned in 1.3, severity assessment for bone fractures is heavily influenced by a number of related key factors. One contribution of this dissertation is to provide quantitative values for some of these key factors which are heretofore unavailable in any other fracture analysis software. Figure 3.14 shows an example of the severity assessment report which includes computed values of key factors for each fragment in case one. The following list describes the values provided in the severity assessment report in detail:

1. Fragment Volume: This is 3D volume of the region enclosed by the fragment surface (shown as "Size" in the severity report table). This information is useful in determining if the fragment is structurally stable and useful in clinical reconstruction, i.e., can it be used for fixation, or is too small to be used in reconstruction.
2. Fragment Fracture Surface Area: The area of the fragment fracture surface is a major factor in determining fracture severity as shown in [2]. As discussed in 1.3, the area of the fracture surface generated is directly related to the energy that the fractured bone absorbed during the fracture event.
3. Fragment Displacement: The fragment displacement is the 3D translation vector that moves the centroid of the fragment, i.e., the average of all surface points, from the fragments original position to its reconstructed position. Fragment displacement indicates how far a fragment has moved or dispersed during the fracture event.
4. Angular Dislocation: The angular dislocation of the fragment is the angle between the principal axis of the bone fragment in its original position and the principal axis of the bone fragment in its reconstructed position.

These quantities are closely linked to fracture severity and the displayed values may allow users to more accurately and objectively estimate fracture severity. Computation of fracture surface areas from 2D or 3D CT images are difficult and the results are often unreliable. Here, the total fracture surface area can be calculated more accurately after the 3D fragment surfaces are segmented and anatomically classified. Physicians have no way to quantitatively estimate fragment displacement and angular dislocation from image data. Current approaches rely upon the physician's visual assessment and experience. For high energy fracture cases, accurately assessment of these values are difficult. Since the proposed system has unique capabilities to compute the original and reconstructed position for each bone fragment, the key factors shown are inaccessible from any other source. Values uniquely available from the proposed software are fragment displacement, angular dislocation, and the surface area for the fragment outer surface and fracture surfaces.

CHAPTER 4: 3D PUZZLE SOLVING

3D computational puzzle solving, as it pertains to this dissertation, describes the generic research problem that seeks to use computer algorithms to facilitate reconstructing 3D broken objects from their fragments. In general, puzzle-solving approaches fall into two categories: (1) boundary matching, i.e., algorithms that match together fragments by comparing their boundaries, and (2) template matching, i.e., algorithms that match fragments into an a-priori known template that is used as a reference shape for the broken fragments. Approaches from both categories require algorithms for curve and surface matching and surface alignment. For boundary matching, boundary curves from the broken fragments are matched to piece together the fragments. For template matching, surfaces on the fragment are matched to the corresponding surfaces on the template so that fragments can be aligned into the template to accomplish reconstruction.

The computational 3D puzzle solving typically consists of three steps: (1) hypothesize geometric matches between fragmented pieces, (2) test these hypotheses by examining how well the geometry of the pieces match, (3) classify these hypotheses and find the best match as the puzzle solving solution. The first step usually involves a searching problem: in order to generate the hypothesized geometric matches, one has to search all geometric shapes in the data. This searching process could be done exhaustively by testing every possible combination of matches. More efficient search methods choose to search matches by selecting pieces that have high probability of generating distinctive matches. The second step typically involves alignment, i.e., the famous alignment algorithm is Iterative Closest Point [71] (ICP [1]). By aligning

[1] See "ICP" in Glossary

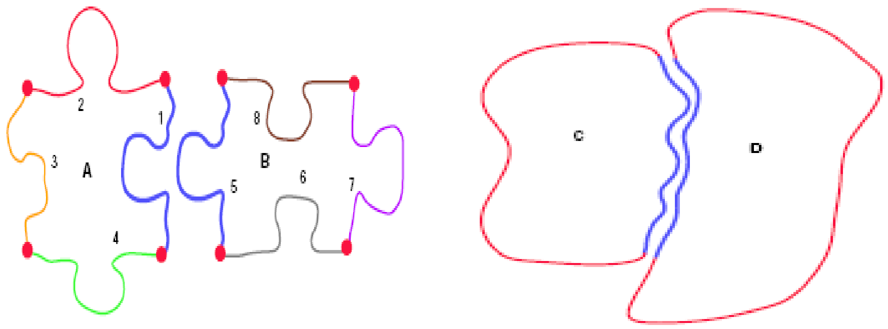
surfaces or boundaries based on the hypothesized geometric match, one can determine whether the hypothesis is true or not. The final puzzle solution is typically the collection of hypotheses that produces the minimum alignment error for all the pieces.

Many researchers have developed algorithms for matching objects based on their boundary shapes to assemble 2D jigsaw puzzles [72, 73]. However, in this case the jigsaw pieces share similar sizes and have distinctly identifiable shapes that need to be matched. In this case one can also assume that all of the jigsaw pieces are complete and intact, and no pieces are missing. Puzzle solving 3D fragments is a much harder problem than the jigsaw re-assembly problem due to the difficulties one faces when identifying and matching the shapes of the fragments. See figure 4.1 for a detailed description.

The puzzle solving approach in this dissertation seeks to piece together bone fragments to reconstruct the original unbroken bone using the second mentioned method for reconstruction: template matching. There are two reasons for choosing this template matching method: (1) boundary matching for 3D bone fragments is a difficult geometric matching problem, (2) the geometry of the template, i.e., the intact bone, is a-priori known. Work in [70] shows that the intact contra-lateral bone, i.e., the unbroken bone in the other limb of the patient, is an adequate template against which to reconstruct the fracture. This template is generated by "mirroring" its geometry, i.e., having its geometry reflected across the plane of symmetry for the human skeletal system. It is assumed that aligning each fragment's outer surface to the surface of the mirrored template, the original bone anatomy will be restored.

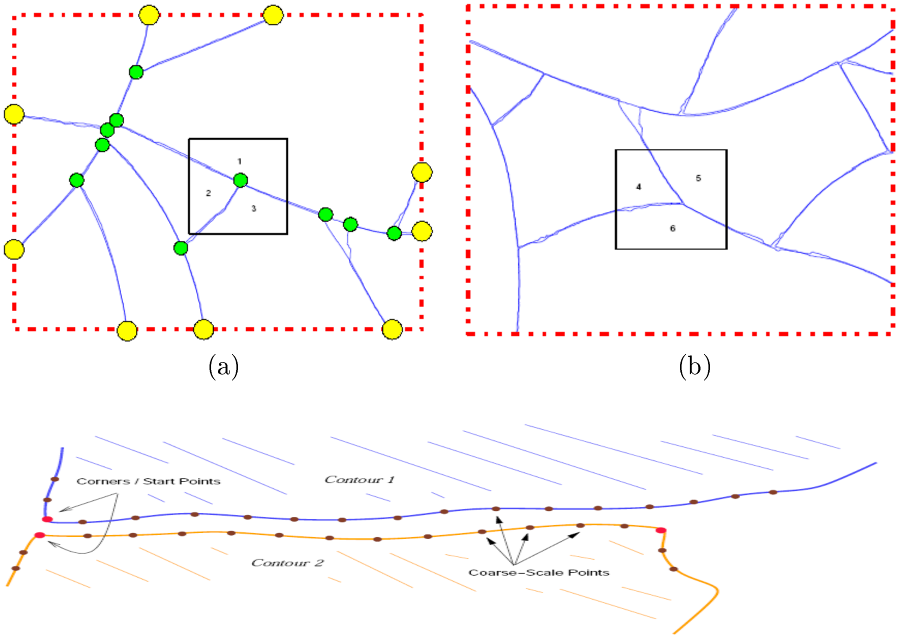
4.1 Previous work

A recent survey [74] discusses systems capable of automatically reconstructing 3D objects from their fragments: The survey categorizes existing puzzle-solving systems into 2D reconstruction approaches [75, 76] and 3D reconstruction approaches



(a) (b)

Figure 4.1: The difference between reassembling commercial jigsaw puzzles and reconstructing broken artifacts. (a) Jigsaw pieces have readily identifiable corners (red dots) allowing programs to easily separate portions of the boundary (shown as different color curves) that will match with some other unique puzzle piece. Additionally, each boundary segment is a smooth planar curve having an isthmus, or neck, that is highly indicative for finding that unique matching puzzle piece. (b) Two hypothetical fragment boundaries. Note here that the problem is made much more difficult as corners are not easily identifiable and may not indicate the beginning or end of a curve that will uniquely match some other fragment. Worse, any portion of a boundary curve may match to any other fragment and the curve itself may match equally well with numerous similar boundaries from other fragments. (Used with permission from [74].)



(a) (b)

(c)

Figure 4.2: In (a), the outer contour of the square ceramic tile, i.e., the intact boundary, Ω_{outer} is shown in red, in this dissertation it is equivalent to ,S_i, the outer surface of bone fragments; fracture boundary, Ω_{inner}, is shown in blue, it is equivalent to ,S_f, the fracture surface of bone fragments; vertices, $\Omega_{inner\perp}$, are shown as green points; and outer vertices, $\Omega_{outer\perp}$, are shown as yellow points. (Used with permission from [74].)

[77, 78, 79, 80]. The survey unifies all types of reconstruction approaches by categorizing the geometric variables that must be used by all geometry-based puzzle-solving algorithms. These variables are grouped into four sets as shown in figure 4.2: (1) Ω_{outer}, variables that characterize the outer boundary of the object before being fractured, i.e., outer boundary, (2) Ω_{inner}, variables that characterize the boundaries generated when the object was fractured, i.e., fracture boundaries, (3) $\Omega_{inner\perp}$, variables that characterize locations where three or more fragment boundaries have broken apart, i.e., fracture junctions, (4) $\Omega_{outer\perp}$, variables that characterize special junctions between outer boundary and the fracture boundary, i.e., outer-surface junctions.

Work in [76, 75] converts the broken 3D objects, i.e., thin stone tablets into 2-D curves by using the tablet fragment outlines as viewed from either top or bottom. Because the tablets are thin, this simplification reduces the computational complexity without significantly impacting the reconstruction results. In [76], curvature signatures are extracted as features from these 2D fragment fracture curves and used to match up pairs of fragments. In contrast to [76], [75] uses fracture curves and vertices, i.e., boundary points lie in locations where multiple fragments were broken apart, to match fracture curve segments. Both [75] and [76] use a robust elastic curve matching instead of traditional cross correlation and least squares techniques, and a multi-scale approach to improve the performance by quickly reducing the search space for compatible matches. The work in [75] uses a best-first strategy which looks at junctions and seeks to match fragment boundaries at open vertices, i.e., where less than three pieces have been matched. Their reconstruction results are promising, but these systems cannot be used to piece together broken bone fragments. This is due to the fact that the geometry of bone fragments are irregular and complex and converting 3D bone fragment surfaces into 2D boundary curves would lose information critical to computing the solution.

[80, 81] attempt to compute the geometry of the pot as a maximum likelihood estimation problem where they seek to find the collection of matched outer surface break curves, i.e., the edges of the surfaces along which the fragments physically broke apart, and global axis/profile-curve geometry that maximizes the probability of all the measured fragment data. Pair of fragments are aligned by aligning vertices, i.e., high curvature points on the fragments outer boundary, of each fragment. The search algorithm starts with all possible pairwise fragment matches and arranges them in a stack sorted by the match probability. Large configurations of fragments such as triplets and quadruplets are built up by merging elements in the stack and inserting the merged objects back into the stack ordered by their probability. However, the computational complexity of the search problem limits this method to problems involving a small number of fragments. [80] modifies this approach by using compact

probability distributions for the fragment geometry rather than repeatedly using the raw data which speeds up the matching process. In addition, [80] aligns axis/profile-curves for groups of fragments instead of aligning break-curve vertices first which is much faster. Unfortunately, these approaches are not feasible for reconstruction of broken bone fragments for two reasons: (1) bone fragment surfaces tend to be more highly irregular and are usually not symmetrical, and (2) fracture surfaces of bone fragments are difficult to accurately measure and reliably match.

[77] uses scanned data points of the fracture surface and a curvature and torsion representation of the outer fracture space curves as features to reconstruct 3D objects. The matching algorithm in [77] uses a similarity matrix whose content at index (i, j) is the sum of squared differences between the curvature and torsion at sample i from one curve and the curvature and torsion at sample j from another curve to identify similar boundary curve segments. Then the algorithm applies a brute force search to find matching break curves and refines fragment alignments by matching fracture surfaces adjacent to these curves. The basic idea of [77] is that, given two 3D models, the best fit is likely to occur when the spatial arrangement of the fragments minimizes the point-to-point distance between the mutually visible, i.e., spatially close, faces of the objects. Towards this end, they introduce an error measure to match two objects, based on this point-to-point distance which is optimized to align fragments. The idea of matching break curves and fracture surfaces is repeated in [23] which puzzle-solves high-energy fracture bone fragments by matching the fracture surfaces of different fragments. However, identification of the fracture surface from CT image data is difficult and often results in inaccurate fragment surfaces which can adversely impact the reconstruction. In addition, the method of [23] relies on a significant amount of user interaction, i.e., one manually selects sub-regions on the fragment surface and matches these surface sub-regions.

Work in [78] uses integral invariant features of fracture surface patches and break curves for reconstruction. These features are computed at a surface point by integrating spatial functions over the region contained within a sphere of radius r centered at the surface point. The matching algorithm in [78] is done by clustering similar integral invariant features for different fracture surface patches and their adjacent break curves over a sequence of scales. Initial pairwise fragment matching is accomplished by performing principal component analysis (PCA) on the augmented feature vector. A second matching criterion checks geometric consistency by ensuring that distances between corresponding points on corresponding surfaces have approximately the same point-to-point distance. This geometric consistency idea is adopted by puzzle-solving in this dissertation to filter generated feature matches as discussed in 4.3.3. After those criteria are checked, a search algorithm computes a graph where each node is a fragment and each edge represents a candidate match between two fragments. The task of the puzzle-solving algorithm is to search for the appropriate sub-graph that specifies the correct collection of fragment matches.

In [82, 83] some efforts are made to improve the matching performance by including information obtained from patterns on the outer fragment surface. They use inpainting and texture synthesis methods to extend the patterns on the boundaries of the outer surface of the fragments to predict those patterns that they expect to find on matching pieces. The mean and variance of the predicted pixel values within small windows around the image boundary are computed as the fragment features and these are used for fragment matching. There are no texture patterns on the bone fragment surface, that can be used as additional features to incorporate in matching.

4.2 Generic template matching as a puzzle solving approach

The template matching puzzle solving approach for bone fracture reconstruction works as follows: (1) hypothesize geometric matches, i.e., each match hypothesizes a point on the surface of the intact template corresponds to a point on the outer surface of a fragment to create a hypothesized pair of points, (2) test these hypotheses, i.e., each hypothesis is tested by computing the alignment error between the two surfaces when matched using the hypothesized point pair, (3) take the collection of hypotheses having smallest alignment error as the final puzzle solution.

The ICP algorithm [71] is a popular non-linear minimization technique used to do generic surface alignment and surface matching. The algorithm takes as input a floating surface and a fixed surface and then iteratively improves the alignment by moving the floating surface to the fixed surface. Each iteration of the ICP algorithm consists of two steps: (1) establish a set of pairwise point correspondences between points on the floating surface and points on the fixed surface, and (2) compute the 3D rigid-body transformation that transforms the floating surface points such that the sum of the squared distances between corresponding points is minimized. Corresponding points at each iteration are taken as the closest point on the other surface. The algorithm recomputes the corresponding point pairs in the beginning of each iteration, and it typically converges to a meaningful alignment position. However, when the floating surface starts in a pose far away from the correct pose the ICP algorithm can converge to local minima and generate a poor alignment result.

The intact template, S^t, defines the surface into which we seek to place the bone fragments. The template-based puzzle solving approach seeks to find a mapping of the bone fragment surfaces onto the surface of the template. The mapping is provided by geometrically matching the k^{th} bone fragment outer surface, S_o^k, to the corresponding part of the intact template, S^t. The surface matching problem requires computation of the Euclidean transformation, $\mathbf{T} = \{\mathbf{R}, \mathbf{t}\}$, where $\mathbf{R}$ is 3x3 matrix that denotes a 3D Euclidean rotation and $\mathbf{t}$ denotes a 3D translation vector. The chosen alignment metric seeks to determine the best value of these variables for minimizing the sum of squared distances between the corresponding points of the bone fragment outer surface and the surface of the intact template. Let $\mathbf{v}_i^k$ denote the i^{th} surface point on the outer surface of the k^{th} fragment, and $\mathbf{v}_j^t$ denote the j^{th} surface point on the intact template, the set χ contains the index pairs (i, j) for the corresponding points pair $(\mathbf{v}_i, \mathbf{v}_j)$. Equation (4.1) computes the alignment error for the alignment given by the transformation $\mathbf{T_k}$:

$$e(\mathbf{T_k}) = \sum_{(i,j)\subset\chi} \left\| \left(\mathbf{T}_k \mathbf{v}_i^k + \mathbf{t}\right) - \mathbf{v}_j^t \right\|^2 \tag{4.1}$$

By minimizing $e(\mathbf{T_k})$ over all values of $\mathbf{T_k}$, we can obtain the best possible alignment.

The brute-force approach for puzzle solving bone fractures is computationally prohibitive. The computational cost is determined by the number of hypotheses (matches) and the computational cost associated with testing or evaluating the likelihood that each hypothesis is true, i.e., this is equal to the computational cost of the ICP algorithm for each such test. Assume n_{icp} denotes the number of ICP iterations to compute a solution, and N_k denotes the number of corresponding pairs for each iteration. Then the computational cost of the ICP algorithm alone is $O(n_{icp}N_k^2)$. The number of corresponding surface point pairs between the fragment surface and the intact template surface may make N_k very large, e.g., $N_k > 20000$. Given the very large number of possible point-pair hypotheses which can be $(20000)^2$ for the earlier example this approach becomes computationally prohibitive for even very small puzzles. The puzzle solution approach introduced in this dissertation addresses this problem by reducing the computational cost for each hypothesis test by replacing the surface matching problem with a less computationally costly surface matching metric.

4.3 Feature-Based Puzzle Solving

Feature-based puzzle solving approaches extract features from surfaces and use these features to build hypothesis matches and to test these matches. The approach to puzzle-solve the k^{th} fragment, S_o^k, with respect to intact template, S^t, consists of five steps:

1. Feature Extraction: This step takes in S_o^k and S^t as input and outputs a collection of features for each point on the fragments and intact template. These features are a compact representation of the geometry of the surface in the

vicinity of each point. (see 4.3.1)

2. Generate Matched Features: This step takes in as input the computed features from (1) and outputs a list of matched feature pairs: $L_{initial}$. (see 4.3.2)
3. Remove Incorrect Matches: This step takes $L_{initial}$ as input and removes suspected false matches from the initial list and outputs a list of candidate matches in a new list: $L_{candidate}$. (see 4.3.3)
4. Test Candidate Matches: This step takes in the previously generated list $L_{candidate}$ and outputs a 3D transformation matrix for each match which aligns the bone fragment to the surface of the intact template.(see 4.3.4)
5. Select The Best Matches: This step determines which of the alignments from the previous step provide the best result and use these to provide the final puzzle-solved solution. (see 4.3.5)

Before we puzzle solve a clinical fracture case, an extra initialization step is required. This step is needed to provide a gross/coarse alignment between the intact surface and the bone fragment. This is accomplished by aligning the intact template to the base fragment, which is usually taken to be the uppermost (proximal) bone fragment in the fracture. The resulting alignment for the base fragment is then applied to all fragments in the fracture case. This initial gross movements of all the fracture fragments serves to bring the surfaces of the fragments which were extracted from the fracture CT images into coarse alignment with the intact surface extracted from the intact CT images. Figure 4.3 shows a graphical overview of the proposed method for puzzle-solving a clinical bone fracture case.

4.3.1 Feature Extraction

This section discusses the features used to represent the geometry of a local surface patch around a chosen surface point. In order to simplify the discussion, let's define some notation. The problem is to match the k^{th} fragment outer surface, S_o^k, to the intact template, S^t. Let $\mathbf{f}_m^t$ denote the feature extracted for m^{th} surface point on the

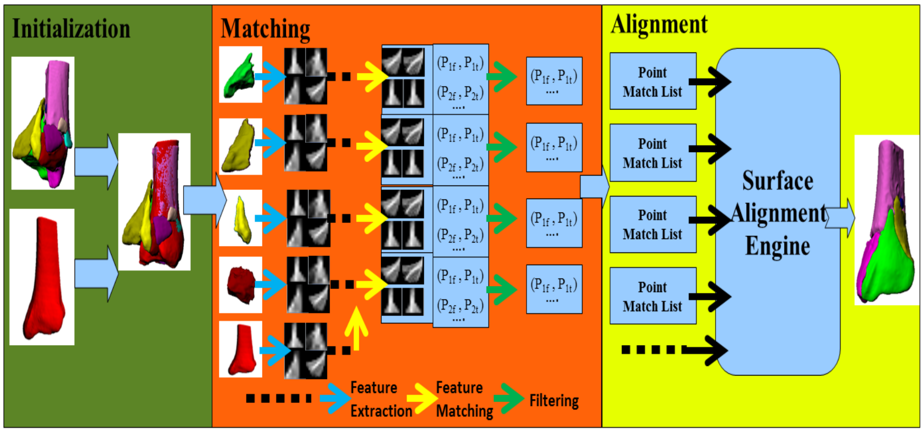
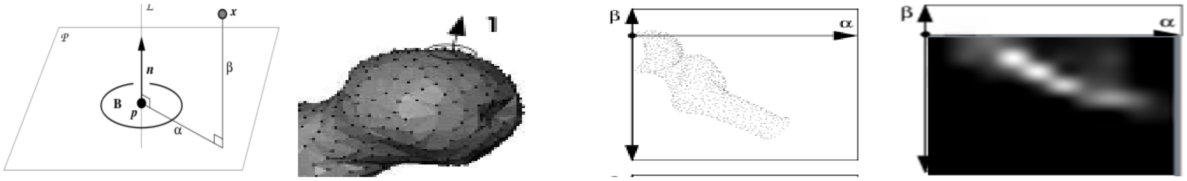


Figure 4.3: Puzzle solving process: initialization, matching and alignment.

intact template. Let $\mathbf{f}_n^k$ denote the feature extracted for the n^{th} surface point on a fragment surface patch. We denote the collection of all features extracted from the intact template as $\mathbf{f}^t$ and the collection of all features extracted from the k^{th} fragment outer surface patch as $\mathbf{f}^k$.

This dissertation adopts the spin image representation [84, 85] as the feature for representing local bone surface geometries around a surface point. The approach converts this geometry into a 2-D image called the spin image. Using this representation two local surface patches with the same geometry produce the same spin images. This reduces the surface matching problem to the less-complex problem of matching similar images. The spin image for an oriented surface point, $\mathbf{p}$, having surface normal $\mathbf{n}$ is depicted in figure 4.4a. Here, the point $\mathbf{x}$ is a point in the local vicinity of $\mathbf{p}$ that also lies on the fragment surface. Spin images are computed by projecting $\mathbf{x}$ into the spin image at the location (α, β) where α is the closest distance between $\mathbf{x}$ and the normal line $\mathbf{n}$, and β is the closest distance between $\mathbf{x}$ and the tangent plane of surface point $\mathbf{p}$. α is always positive and β is positive when $\mathbf{x}$ is above the plane passing through $\mathbf{p}$ with normal $\mathbf{n}$ and negative when $\mathbf{x}$ is below this plane as shown in figure 4.4a. Plotting the (α, β) values of all the points in the vicinity of the point $\mathbf{p}$ generates a 2D image as shown in figure 4.4c. A spin image is generated by imposing a grid over



(a) (b) (c) (d)

Figure 4.4: (a) Oriented surface point **p** with its normal n and the tangent plane p, one of its neighbor point **x** on the fragment surface and (α,β) are computed values for point **x** respect to point **p** ; (b) shows point **p** on the bone surface and its neighbor points; (c) Plotted neighbor points on the (α, β) grid; (d) Computed spin image for out of (c) [84].

this image and setting the spin image intensity to the number of (α, β) points that lie inside each box of the grid. (see figure 4.4d). The spin image is a description of the local shape of the object surface because it is a function of the projection of the relative positions of 3D points that also lie on this surface. Since the results is a 2D image some of the 3D metric information is lost in this transformation. Specifically, the angular position of the 3D surface point relative to the point **p**. Hence, it can be used as a compact, yet, imperfect, representation of the shape of a bone fragment surface which is Euclidean invariant, i.e., the spin image does not change when the object is transformed.

4.3.2 Generate Matched Features

Matching the spin images from $\mathbf{f}^t$ and $\mathbf{f}^k$ generates a list of hypothesized correspondences between the fragment surface points and the surface points of the intact template. Let $L_{initial}$ denote this list of initial hypothesized point pairs. The normalized linear correlation coefficient, $R_{sp}(P, Q)$, is used to compute a score for each hypothesized match where a score of 1 is a perfect match and scores can come from the interval (-1, 1). Given two spin images P and Q with N_{bin} bins (number of pixels in the image), denote $p_i = I(\alpha, \beta)$ as the i^{th} image intensity value for spin image P, similarly, denote $q_i = I(\alpha, \beta)$ as the i^{th} image intensity value for spin image Q. With this notation the linear correlation coefficient, $R_{sp}(P, Q)$, is computed as shown in equation (4.2):

$$R_{sp}(P,Q) = \frac{N_{bin}\sum_i p_i q_i - \sum_i p_i \sum_i q_i}{\sqrt{\left(N_{bin}\sum_i p_i^2 - \left(\sum_i p_i\right)^2\right)\left(N_{bin}\sum_i q_i^2 - \left(\sum_i q_i\right)^2\right)}} \tag{4.2}$$

R_{sp} lies between -1 (negatively-correlated) and 1 (positively correlated), and it provides a basic criteria for the comparison of two spin images. When R_{sp} is close to 1, the images are similar, when R_{sp} is close to -1 the images are different. If we use the normalized correlation to score matched spin images the resulting scores are biased by the overlap of the matched images. For example, there might be a case where two matched spin images have a small area of overlap. The score for such images may indicate that the images are very similar with a high R_{sp} value. On the other hand, two spin images with a large area of overlap may be assigned a lower R_{sp} value but exhibit widespread similarities. To address this problem, we consider the number of overlapping pixels as part of the final matching score. Let's denote $N_{overlap}$ as the number of overlapping pixels, i.e., pixels where spin images P and Q both have values. The adjusted similarity function $C_{sp}(P,Q)$ is introduced by modifying the linear correlation $R_{sp}(P,Q)$ as shown in equation (4.3).

$$C_{sp}\left(P,Q\right) = \left(\arctan\left(R_{sp}\left(P,Q\right)\right)\right)^2 - \lambda\left(\frac{1}{N_{overlap} - 3}\right) \tag{4.3}$$

This similarity function weights the correlation R_{sp} against the variance intrinsic to the correlation coefficient which increases as the amount of overlap in the two spin images increases. The parameter λ is used to control the relative weight of the correlation coefficient and the variance of this statistic to produce a final similarity score. [84, 85] mention that λ controls the point at which the overlap between spin images dominates the value of the similarity metric for two spin images. When the overlap is much larger than λ, the second term in equation (4.3) becomes negligible. In contrast, when the overlap is much less than λ, the second term dominates the similarity measure. Therefore, λ should be the expected overlap between spin images. In this dissertation, λ is automatically computed for each fragment by computing average number of none-black pixels for all the spin images generated from that fragment and setting λ to half of the average value. Note that the parameter λ is estimated dynamically for each fragment surface.

4.3.3 Remove Incorrect Matches

This section describes method using geometric consistency to remove matches in the initial list that are suspected to be incorrect hypothesized surface correspondences. Due to the noise from image data and errors from segmentation and classification algorithms, the initial list of matches $L_{initial}$ often contains many false hypothesized surface point correspondences. Since careful analysis of these matches is done computationally expensive, a quick method for removing these hypotheses is necessary. Assume we have two hypothesized correspondences (matches) $[\mathbf{f}_1^k, \mathbf{f}_1^t]$ and $[\mathbf{f}_2^k, \mathbf{f}_2^t]$ from the list $L_{initial}$ and their corresponding surface points are $[\mathbf{v}_1^k, \mathbf{v}_1^t]$ and $[\mathbf{v}_2^k, \mathbf{v}_2^t]$. If they are both true matches, then the surface point pairs should be geometrically consistent, i.e., the distance between $\mathbf{v}_1^k$ and $\mathbf{v}_2^k$ should be equal to the distance between $\mathbf{v}_1^t$ and $\mathbf{v}_2^t$. We validate this constraint using spin image coordinates in a geometric consistency test. If the two matches satisfy equation (4.4), we consider them to be geometrically consistent matches which means both of them may be true matches.

$$\left|\xi_{\mathbf{v}_1^k}\left(\mathbf{v}_2^k\right) - \xi_{\mathbf{v}_1^t}\left(\mathbf{v}_2^t\right)\right| < D_{gc}, \left|\xi_{\mathbf{v}_2^k}\left(\mathbf{v}_1^k\right) - \xi_{\mathbf{v}_2^t}\left(\mathbf{v}_1^t\right)\right| < D_{gc} \qquad (4.4)$$

$\xi_{\mathbf{v}_1^k}\left(\mathbf{v}_2^k\right)$ represents the 2D coordinates (α_1, β_1) which are computed by considering $\mathbf{v}_1^k$ as oriented point and $\mathbf{v}_2^k$ as its neighbor point, similarly, $\xi_{\mathbf{v}_1^t}\left(\mathbf{v}_2^t\right)$ represents the 2D coordinates (α_2, β_2) which are computed by considering $\mathbf{v}_1^t$ as oriented point and $\mathbf{v}_2^t$ as its neighbor point. This test evaluates the distance between (α_1, β_1) and (α_2, β_2). Here $D_{gc} = 2\gamma_{intact}$ where γ_{intact} is the resolution of the intact template, i.e., average edge length of the edges from the intact template surface mesh.

When the initial match list is constructed the elements are sorted by decreasing similarity score, C_{sp}. Inconsistent elements are removed by splitting the list $L_{initial}$ into two parts at its midpoint. The first list will contain matches that have higher similarity scores and the second will contain matches that have lower similarity scores. One match from each of the two lists respectively is taken from the top of the list to form a collection of two hypothesized correspondences. The geometric consistency condition is evaluated for the pair of correspondences. If they both satisfy the consistency condition, we keep both matches. Otherwise, we keep the match that has a higher similarity score and discard the other match. After evaluating all matches in both lists, the remaining matches are placed in the sorted list referred to as $L_{candidate}$.

4.3.4 Test Candidate Matches

A novel surface alignment technique uses hypothesized surface point matches $M(\mathbf{v}_i^k, \mathbf{v}_j^t)$ to compute the transformation matrix $\mathbf{T^k}$ that aligns the fragment, S^k, to the intact template, S^t. Because the computational cost of the ICP algorithm is high (see 4.2), it is inefficient to use the ICP algorithm to evaluate every match in the list. This dissertation develops a hybrid surface alignment solution. It proceeds by using the hypothesized corresponding surface points to compute an accurate estimate of the initial starting position for the fragment surface. For a given surface point match, the surface-to-surface alignment proceeds in three steps:

1. Initial Surface Alignment: This step takes in a point match $M(\mathbf{v}_i^k, \mathbf{v}_j^t)$, outputs an initial transformation matrix, $\mathbf{T}^{\mathrm{k}}_{\mathrm{init}}$ (see 4.3.4.1).
2. Coarse Surface Alignment : This step takes in the point match $M(\mathbf{v}_i^k, \mathbf{v}_j^t)$ and the initial transformation matrix $\mathbf{T}^{\mathrm{k}}_{\mathrm{init}}$ and outputs a new transformation matrix, $\mathbf{T}^k_{coarse}$, which repositions the fragment to reduce its alignment error (see 4.3.4.2).
3. Refined Surface Alignment: This step takes in as input the surface points of the fragment,$\mathbf{v}^{\mathrm{k}}$, and the surface points of the intact template, $\mathbf{v}^{\mathrm{t}}$, and the transformation $\mathbf{T}^k_{coarse}$ and outputs a final transformation matrix $\mathbf{T}^{\mathrm{k}}_{\mathrm{refine}}$ which aligns the fragment surface with the surface of the intact template (see 4.3.4.3).

Note that the final step uses the ICP algorithm. This allows the points of the initial match to change as the correspondence in ICP is computed as the closest points after each movement of the floating surface. In this way hypotheses can be used that may not be exactly correct and in cases where the hypothesis is "close", the ICP algorithm is still likely to converge to the correct alignment.

4.3.4.1 Initial Surface Alignment

The initial transform matrix $\mathbf{T}^k_{init}$ is computed by moving the surface point of the fragment to coincide with the corresponding surface point on the intact template and by aligning the two surface normal vectors at these points. For correspondences which are nearly or exactly correct, the resulting transform matrix $\mathbf{T}^k_{init}$ typically brings the fragment into good alignment with the intact template. Let the point pair $(\mathbf{v}^k_i, \mathbf{v}^t_j)$ denote the hypothesized corresponding surface points having normals $(\mathbf{n}^k_i, \mathbf{n}^t_j)$. Then the translation vector; $\mathbf{t}_{\text{init}} = \mathbf{v}^t_j - \mathbf{v}^k_i$. The rotation matrix is computed by rotating the fragment around a fixed axis, $\mathbf{n}_{axis}$, by an angle ϕ_r. The axis of rotation, $\mathbf{n}_{axis}$, is the cross product of template surface normal, $\mathbf{n}^t_j$, and the fragment surface normal, $\mathbf{n}^k_i$; $\mathbf{n}_{axis} = \mathbf{n}^t_j \times \mathbf{n}^k_i$. The angle of rotation is the angle between the two normal vectors; $\phi_r = \arccos(\mathbf{n}^t_j \cdot \mathbf{n}^k_i)$. The Rodriguez formula (4.5) illustrates how to compute a 3-dimensional rotation vector from the normal vector, $\mathbf{n}^k_i$, the axis of rotation, $\mathbf{n}_{axis}$, and the rotation angle, ϕ_r. The rotation vector, $\mathbf{v}_{\text{rot}}$, can then be converted to a rotation matrix $\mathbf{R}_{\text{init}}$ using equation (4.6) [86] .

$$\mathbf{v}_{\text{rot}} = \mathbf{n}^k_i \cos\phi_r + (\mathbf{n}_{axis} \times \mathbf{n}^k_i)\sin\phi_r + \mathbf{n}_{axis}(\mathbf{n}_{axis} \cdot \mathbf{n}^k_i)(1 - \cos\phi_r) \tag{4.5}$$

$$\mathbf{R}_{\text{init}} = \mathbf{I}\cos\phi_{\text{r}} + \mathbf{R_x}\sin\phi_{\text{r}} + (1 - \cos\phi_{\text{r}})\,\mathbf{n}_{\text{axis}}\mathbf{n}^{\text{t}}_{\text{axis}} \tag{4.6}$$

Where $\mathbf{I}$ is the 3x3 identity matrix and $\mathbf{R_x}$ is a cross-product matrix for the vector $\mathbf{n}_{\text{axis}} = (x_a, y_a, z_a)$ as shown in equation (4.7).

$$\mathbf{R_x} = \begin{bmatrix} 0 & z_a & y_a \\ z_a & 0 & -x_a \\ -y_a & x_a & 0 \end{bmatrix} \tag{4.7}$$

4.3.4.2 Coarse Surface Alignment

The coarse surface alignment seeks to find the best orientation for the fragment as a rotation around the hypothesized corresponding surface normal and point. As mentioned in 4.3.1 spin images lose this information when they are computed and recovering the value of this unknown angle requires solving an one-dimensional error minimization problem. Consider the transformation matrix $\mathbf{T_r^k}$ which aligns the k^{th} fragment with the intact template. Let $\mathbf{T_r^k}$ be a function of six-parameters transformation vector, $\{x, y, z, \theta, \varphi, \psi\}$. If the hypothesized correspondence between the two surfaces is correct, the transformation matrix $\mathbf{T}_{init}$ computes the correct value for five of the variables in the vector. Three of them, $\{x, y, z\}$, are computed from aligning two point coordinates, i.e., $\mathbf{t}_{\text{init}} = \{x, y, z\}$. Two of remaining three angles, $\{\theta, \varphi\}$, are computed by aligning the two normals $\mathbf{n}_i^k$ and $\mathbf{n}_{axis}$ resulting in the rotation matrix $\mathbf{R}_{init}$, i.e., $\mathbf{R}_{init} = \{\theta, \varphi\}$. The last parameter, the rotation angle ψ, is the free parameter we seek to compute in the one-dimensional error minimization problem. This is accomplished by rotating the floating fragment surface around the aligned surface normals using the corresponding surface points as the origin. The angle, ψ, is taken as the rotation angle that produces the minimum local alignment error between two surfaces.

Let $\mathbf{R}(\psi)$ denote the one parameter rotation around the corresponding surface normal, $\mathbf{n}_j^t$, or equivalently, $\mathbf{R}_{init}\mathbf{n}_i^k$. Since the rotation matrix $\mathbf{R}(\psi)$ is a rotation around this surface normal we can compute it using equation (4.5) and (4.6) and substituting $\mathbf{n}_j^t$ for $\mathbf{n}_{axis}$ and ψ for ϕ_r. Candidate values for the coarse alignment are computed by updating the rotation matrix of the transformation $\mathbf{T}_{init}^{'} = \{\mathbf{R}(\psi)\mathbf{R}_{init}, \mathbf{t}_{init}\}$ for values of $\psi \in [0, 2\pi)$. Estimation then reduces to solving equation (4.8):

$$\hat{\psi} = arg\min_{\psi}(\sum_{(i,j)\in\chi} (\left\| \left(\mathbf{T}'_{init} v_i^k\right) - v_j^t \right\|^2)); \tag{4.8}$$

Where v_i^k denotes the transform fragment point from the the k^{th} fragment. The resulting transformation, $\mathbf{T}_{coarse}$, includes the new transformation matrix, i.e., $\mathbf{T}_{coarse} = \{\mathbf{R}(\hat{\psi})\mathbf{R}_{init}, \mathbf{t}_{init}\}$ computed by minimizing local alignment error between two surfaces and is the output of this step.

4.3.4.3 Refined Surface Alignment

The refinement step utilizes the ICP algorithm on two surfaces to finalize the alignment. Since the fragment is nearly aligned to the intact template: the point-to-plane error metric is used for this step. As stated in [87], the point-to-plane metric tends to converge faster than the point-to-point metric and it is more stable provided that the angular misalignment between the two point sets is small. The transformation matrix $\mathbf{T}^{\mathrm{k}}_{\mathrm{refine}}$ is computed by the ICP algorithm and is considered to be final solution for the k^{th} fragment alignment.

4.3.5 Select The Best Matches

Figure 4.5 illustrates how the alignment process uses a hypothesized surface correspondence to align a fragment to the intact template. For a given fragment, each hypothesized surface correspondence is evaluated in the sequence given by the match score from (4.3.3) starting at the highest score. Each evaluation aligns a fragment to the intact template using the three alignment steps specified in the three previous section. However, if the algorithm goes through all three steps for every match, the reconstruction process is very time-consuming. To reduce computation each alignment step includes pre-defined conditions to determine whether the system should continue to evaluate the match or discard the match and try a new hypothesized match. The local alignment error is used to control the process. Because the computational cost increases in each step, the alignment error threshold values are smaller (stricter) for each step. In this dissertation, the threshold values are $4\gamma_{intact}$ for step one, $2\gamma_{intact}$

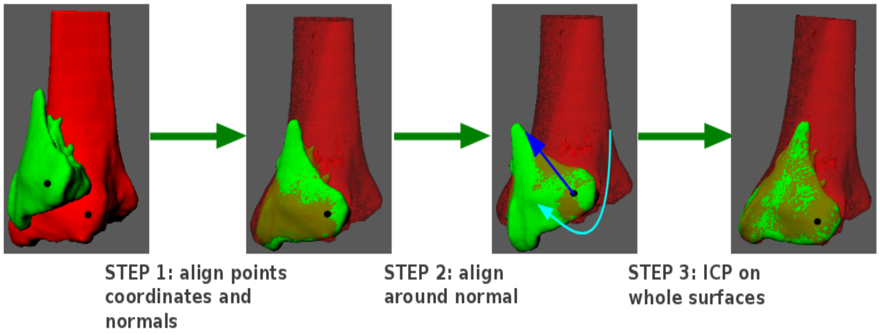

STEP 1: align points coordinates and normals
STEP 2: align around normal
STEP 3: ICP on whole surfaces

Figure 4.5: An example of alignment using the surface alignment engine. The red bone is the intact template, the green bone is one of the fracture fragment. The two black points on both surfaces are matched surface points.

for step two, and γ_{intact} for step three. Finally, when one match is accepted by all three steps the output position from step three is considered as the final alignment for the fragment.

4.4 System Enhancements

Several software enhancement tools are introduced to help improve the quality of the 3D puzzle-solved solutions and to help reduce the time necessary to compute these solutions. These enhancement tools include:

1. Surface Sampling.
2. Mean Curvature Histogram Biased Search.
3. Using Occupied Regions.
4. Improving Fragment Alignment By “Jiggling”.

The following sections ((4.4.1), 4.4.2, 4.4.3, 4.4.4) discuss each tool in detail.

4.4.1 Surface Sampling

The intact template and the fragment surfaces often consists of a large number of surface points. Computing spin images for every point on these surfaces is a time-consuming task. In order to improve the speed of the puzzle-solving algorithm, uniform sub-sampling is applied on both the intact template and the fragment surfaces. Surface points are randomly selected on the fragment with a constraint that the distance between any two sampled surface points are greater than a pre-defined sampling distance $\triangle s$. The larger $\triangle s$ value computes fewer sampled points on the fragment surface and smaller value computes more sampled points on the fragment surface. Spin images are only computed for those sampled surface points on the intact template and the fragment surfaces. In this dissertation, the sub-uniform sampling distance for each fragment is set to $\triangle s = 1.5\gamma_{intact}$, and γ_{intact} is the average edge length of edges on the intact template.

4.4.2 Using Occupied Regions

The concept of occupied regions allows for significant performance improvements. The intact template usually contains a large number of surface points. It is time-consuming to compute spin images for each of these surface points on the intact template. Since part of the intact template surface has already been aligned to the base fragment, these points should be excluded from the matching. We mark these surface points on the intact template surface as "occupied", and spin images are only computed for surface points inside the "unoccupied" regions of the intact template. This significantly reduces the search space for the matching process. Occupied points on the intact template are flagged using the following condition: the intact surface point, $\mathbf{v}_{\mathrm{j}}^{\mathrm{t}}$, is marked as "occupied" if its distance to the closest aligned fragment surface is less than $2\gamma_{intact}$.

4.4.3 Mean Curvature Histogram Biased Search

To improve the performance of the spin image matching, an approach called " Mean Curvature Histogram Biased Search" is used which reduces the number of spin images computed for both the intact template surface and the fragment surfaces. This significantly reduces the computational cost of searching for the correct spin image matches. Intuitively, for the bone fragment matching problem, points from planar surfaces should not be considered as candidate feature points for matching, while distinctive points such as those on ridges or alleys should be selected as feature points.

Likewise, surface regions around high curvature points have more distinctive geometric shape than other more typically surface regions. As such these surfaces are less common and more easily identified and matched. Information theory supports this intuition as it assigns events with low probability (high entropy) a higher information value (in bits). The goal in this approach is to identify the surface points on each surface that encode more information (distinguishable geometric shape). These points should have fewer candidate correspondences and should provide more accurate alignments.

To accomplish this goal, a histogram of mean curvature values is computed that includes curvatures for all points on the intact template (see figure 4.6 for an example). For each bin in the histogram, a probability value is computed by dividing the number of points falling into the bin by the total number of points on the intact template to generate an "empirical" probability distribution. The distribution provides a probability value for each bin. Points falling into bins with probability lower than a defined threshold are selected as feature points. Using this enhancement, spin images are computed for surface points only if when their associated probability in the empirical curvature distribution is less than 0.005. This drastically reduces the number of spin images computed for both the intact template and the fragment and results in significant improvements in the speed of the automatic reconstruction algorithm.

The search space for surface matching is further reduced using a simple curvature-based matching technique. When matching a spin image from each fragment with spin images from the intact template, the algorithm first computes the surface mean curvatures at the surface point and then finds surface points on the intact template having similar mean curvature values. The fragment spin image is only compared with those spin images on the intact template found to have similar surface curvature. This enhancement enforce the constraint that two surface points are true matched points only if their surface mean curvature values are also similar. This improvement greatly reduces the computational cost for generating spin images by limiting the total number of spin images computed and stored for both fragment surfaces and the intact template surface, and also reduces the number of hypothesized surface correspondences by limiting the hypotheses to small number of candidate matches based on the surface mean curvature values at the hypothesized corresponding points.

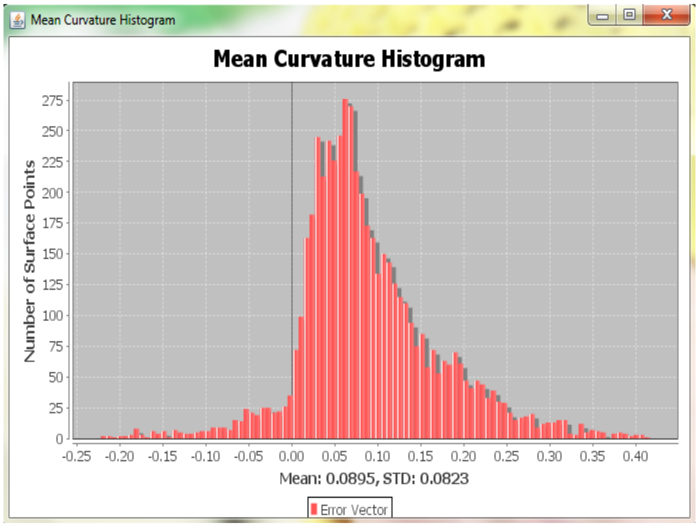


Figure 4.6: This figure shows an example of a mean curvature histogram of the intact template of case one.

4.4.4 Improving Fragment Alignment By “Jiggling”

The jiggling algorithm is used to correct the position of misaligned fragments in the final 3D bone reconstruction results. After automatic reconstruction finishes, users visually assess the reconstruction results and can choose fragments that they consider to be mis-aligned. For these fragments, the user may apply the “jiggling” algorithm on those fragments. The jiggling algorithm relies on the unoccupied regions of the intact template to assist in improving the alignment of the misaligned fragments.

The “jiggling” algorithm consists of three steps: (1) determining the neighboring fragments for the fragment which needs to be “jiggled”, (2) determining the jiggling distance between the target fragment and the neighboring fragment, (3) “jiggling” the fragment to find the best position which is defined to be the position that minimizes the "unoccupied" region on the intact template as defined in 4.4.2.

Two fragment surfaces are considered neighbors if they share common boundaries, i.e., some portion of their boundaries lie close to each other. The boundary of a fragment is the set of surface points that lie on the edge of fragment surface meshes. Let $\mathbf{v}^{bk}$ denote the boundary point set from one fragment surface and let $\mathbf{v}^{bt}$ denote the boundary point set from another fragment surface. If the minimum Euclidean distance between these two point sets is less than $2\gamma_{obj}$, where γ_{obj} is the resolution of one of the fragments, then these two fragments are neighbors, otherwise they are not neighbors.

The "jiggling" distance between two neighboring fragments is computed as the average distance between the two shared boundary point sets. Consider a point, $\mathbf{v}_1^{bk}$, from the point set, $\mathbf{v}^{bk}$, if the boundary point set on another fragment surface, $\mathbf{v}^{bt}$, contains a point that its distance to $\mathbf{v}_1^{bk}$ is less than $4\gamma_{obj}$, then the boundary point, $\mathbf{v}_1^{bk}$, is considered as a shared boundary point in the point set $\mathbf{v}^{bk}$. Identifying all the shared boundary points on both point sets, $\mathbf{v}^{bk}$ and $\mathbf{v}^{bt}$ to form two subset, $\mathbf{v}^{sk}$, and $\mathbf{v}^{st}$. Then establish point pairs between those two sets using closest point-to-point distance metric. The "jiggling distance" is computed as the average distance between these point pairs.

For each neighboring fragment, the target fragment moves toward to that neighboring fragment by the "jiggling" distance and then starts the ICP algorithm to further refine the alignment. When the fragment is aligned, the total surface area of the "unoccupied" region on the intact template is computed. The final solution for the "jiggled" fragment is the aligned position which produces the minimum "unoccupied" region on the intact template.

CHAPTER 5: RESULTS

The bone reconstruction system was used to reconstruct six clinical fracture cases which range from low energy fracture events such as 1.5 foot fall, to high energy fracture events such as a 50 mph car accident. The patient data, injury cause, and the Orthopedic Trauma Association (OTA) classification for each case is shown in Table 5.1. Since these are real clinical cases, the patient names have been removed to protect their privacy. All of the six cases were assigned a numerical severity score ranging from (1-100) by three well-trained surgeons based on their personal experience and subjective inference shown in column C_1,C_2, and C_3 of Table 5.1.

The following sections (from 5.1 to 5.6) discuss the reconstruction results for each case. The discussion for each case follows the steps of the interactive reconstruction process discussed in the Chapter 2. For all six cases, the fragment and intact surfaces were segmented by the Orthopedic Biomechanics Laboratory of The University of Iowa outside the system using the method introduced in [88] and which is also discussed in 3.1. The fragment surfaces are then partitioned by the ridge walking algorithm as described in 3.2.1 and in [48]. All of the surface patches are manually

Table 5.1: This table shows patient data, injury cause, OTA classification, and severity scores by three surgeons for each case. The higher severity scores indicate higher fracture severity.

Case #	Sex	Age	OTA classification	Injury mechanism	C1	C2	C3	Avg
1	F	38	C32	MVA (50 mph)	60	55	60	58
2	M	21	B13	Fall (30 ft)	50	60	58	56
3	F	42	C21	MVA (30 mph)	62	80	79	74
4	M	20	C13	ATV	6	15	32	18
5	M	24	C23	Fall (12 ft)	55	57	62	59
6	M	34	C11	Fall (18 ft)	70	65	77	71

classified by users during the reconstruction as described in Chapter 2. Users also performed a manual initial alignment of the base fragment to the intact template as described in 3.4.

Each case is discussed in a separate section. These sections describe the unique aspects of each case and adhere to a fixed structure for clarity. This structure consists of six figures, one table, and a discussion. The discussion details some information in these figures and tables and also mentions the distinctive aspects of the case and its reconstruction using the system. The following list defines the elements in each case analysis section in detail:

1. CT Images: This figure shows a set of six 2D images from axial, coronal and sagittal views. They have been selected to depict distinctive aspects of the geometry of the fracture fragments and intact template.
2. Segmented Intact & Fracture Surfaces: This figure shows the segmented fragment surfaces and the surface of the intact template from the axial, coronal and sagittal views.
3. Fragment Views: This figure shows each fracture fragment and its outer surface from axial, coronal, and sagittal views.
4. Fragment Outer Surfaces: This figure shows the outer surface patch for each fracture fragment from axial, coronal, and sagittal views.
5. Fragment Mesh Analysis Table: This table provides quantitative values for each fragment mesh and the fragment outer surface patch mesh.
6. Mixed Image & Surface Data: This figure shows two screen captures of the fracture case, one before reconstruction and one after reconstruction in both 2D and 3D. These images show how the fragments move due to reconstruction within a single 2D slice of the 3D CT data and also as a collection of 3D surfaces.
7. Fracture Severity Analysis Report: This figure shows a table containing quantitative values for key factors that impact the fracture severity for each fragment.

These values include the following attributes for each fragment: size, outer surface area, fracture surface area, displacement angle, and displacement vector (see 3.5 for details of these attributes).

8. A discussion accompanies each collection of the figures and tables that discusses this information and the distinctive aspects of the case and its reconstruction using the system.

The Mixed Image & Surface component to each case analysis section requires further explanation. Each instance of this figure contains four images which are taken from windows within the system interface before and after the case was reconstructed. Image (a) is a screen capture of the fracture image window before reconstruction, the colored fragment boundary lines show the boundaries of each fragment in the 2D image. Image (b) is a screen capture of the 3D fragment surfaces before reconstruction. The 2D CT image in figure (a) is also shown in (b) as an image in a 2D plane which intersects with the 3D fragment surfaces. Images (c) and (d) show the same information as that within (a) and (b), but, in this case, the fragment boundaries and 3D fragment surfaces are shown in their reconstructed positions. This provides a visual representation depicting the fragments and how they were moved to reconstruct the bone.

Figure 5.1 shows the displaced positions and three different views of reconstructed fragments for all six clinical fractures. By visual assessment of the reconstruction results, all cases were reconstructed successfully with the exception of case three. Some large fragments such as fragment A17 in case one and fragment A8 in case six are mis-aligned by the automatic puzzle-solving algorithm and had to be subsequently fixed by the “jiggling” algorithm. Some small fragments in some of the cases such as fragment A8, A12 in case one and fragment A5, A3 in case three were not correctly puzzle-solved by the automatic algorithm and the “jiggling” algorithm. Manually repositioning these fragments is required. See following sections for detailed discussion

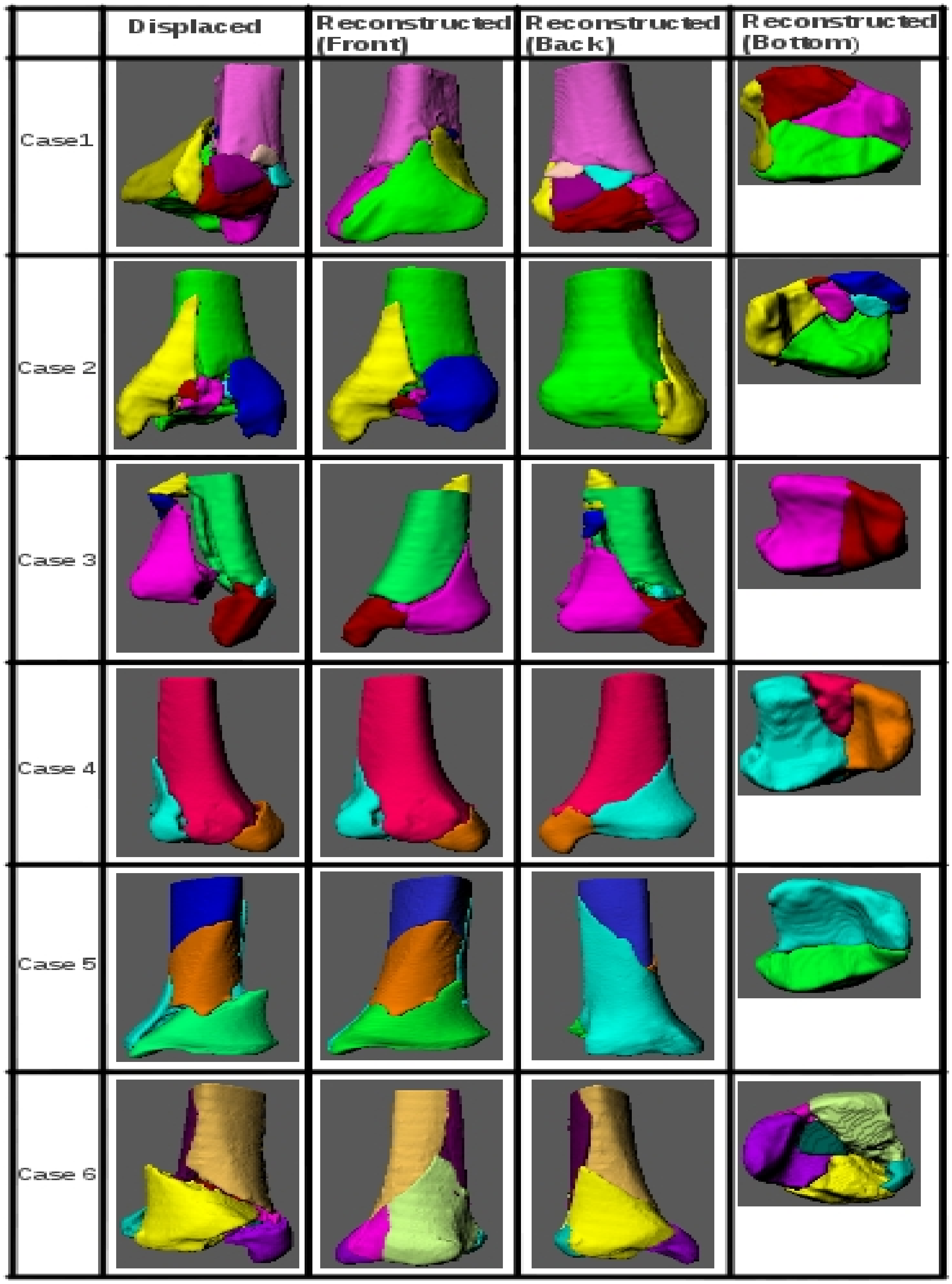


Figure 5.1: Six clinical tibia plafond fractures are puzzle-solved. The original fractured positions for the fragments are shown in the left column and three different views of reconstructed fragments are shown in the remaining columns.

Table 5.2: Global alignment error, i.e., the average local alignment error of all fragments, is recorded for each case in this table.

Case ID	Global Alignment Error (mm)
1	0.23
2	0.27
3	0.32
4	0.34
5	0.33
6	0.27

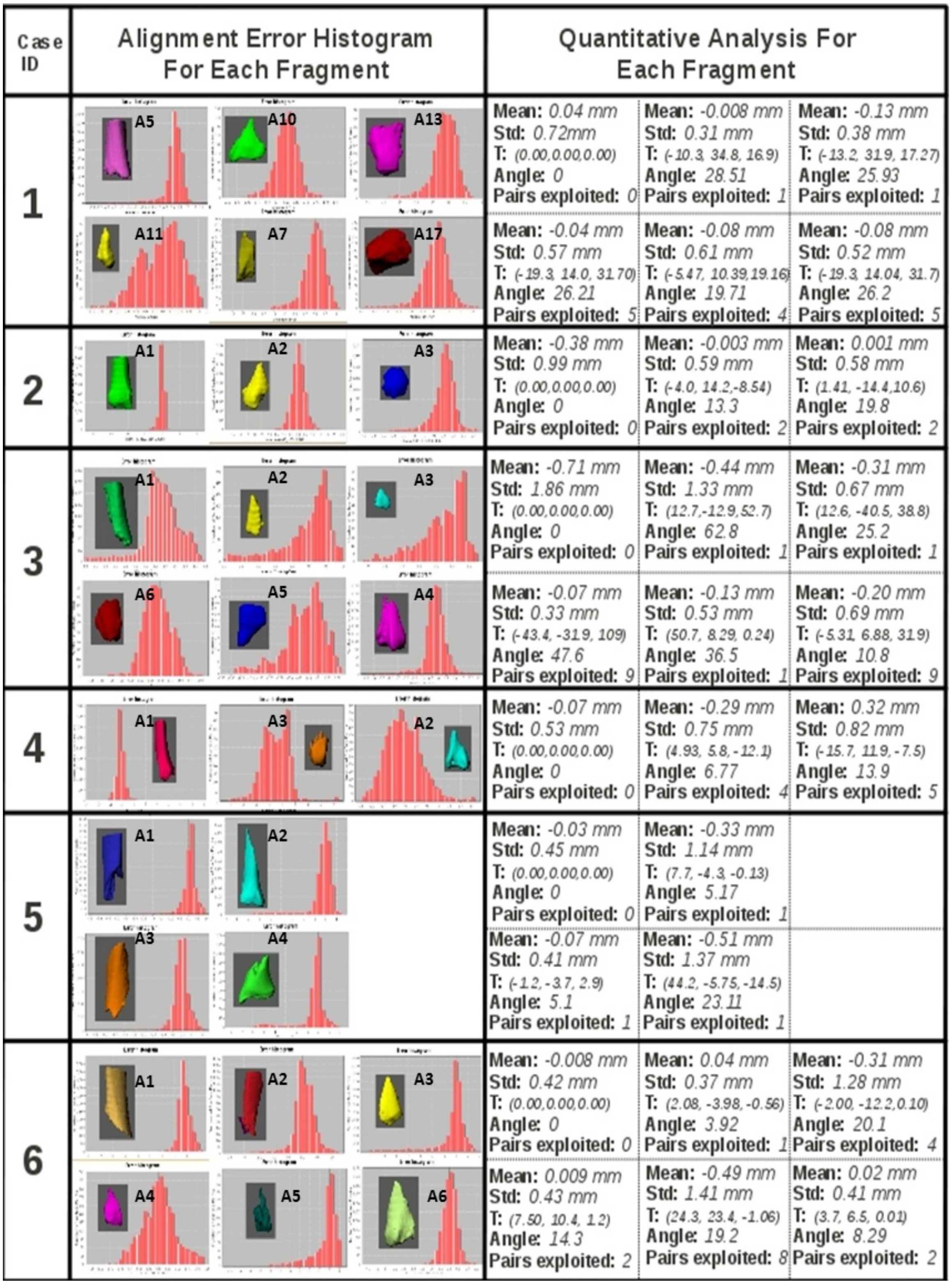
Table 5.3: Puzzle solving performance: the time needed to run puzzle-solving algorithm for the fracture case. This includes time for computing spin images on the intact template and on the fragment surface, time spent matching and filtering spin images, and time spent aligning fragment surfaces.

Case ID	Completion time (sec)	Number of points on intact
1	140	24935
2	220	45529
3	272	50539
4	90	33630
5	430	68160
6	650	117549

about the reconstruction results.

Figure 5.2 shows alignment error histograms for individual fragments and statistical error analysis for all six cases. Local alignment error histograms for individual fragments in each case are shown in left column. Statistical error analysis for these errors are shown in right column which include the mean and standard deviation of each histogram. Also provided are each fragment's displaced distance as an (x, y, z) vector, $T(mm)$, and the angular displacement as a single angle (degrees) between broken position and the reconstructed position. The pairs exploited value indicates how many point pairs were tried before the accepted "correct" match was found.

Table 5.2 summarizes the global alignment errors for all six cases. From the table, we can see that global alignment error after the construction for all six cases are relatively small ($<$ 1mm) and they are considered to be successful. Table 5.3 shows



Case ID	Alignment Error Histogram For Each Fragment			Quantitative Analysis For Each Fragment		
1	A5	A10	A13	Mean: 0.04 mm Std: 0.72mm T: (0.00,0.00,0.00) Angle: 0 Pairs exploited: 0	Mean: -0.008 mm Std: 0.31 mm T: (-10.3, 34.8, 16.9) Angle: 28.51 Pairs exploited: 1	Mean: -0.13 mm Std: 0.38 mm T: (-13.2, 31.9, 17.27) Angle: 25.93 Pairs exploited: 1
	A11	A7	A17	Mean: -0.04 mm Std: 0.57 mm T: (-19.3, 14.0, 31.70) Angle: 26.21 Pairs exploited: 5	Mean: -0.08 mm Std: 0.61 mm T: (-5.47, 10.39,19.16) Angle: 19.71 Pairs exploited: 4	Mean: -0.08 mm Std: 0.52 mm T: (-19.3, 14.04, 31.7) Angle: 26.2 Pairs exploited: 5
2	A1	A2	A3	Mean: -0.38 mm Std: 0.99 mm T: (0.00,0.00,0.00) Angle: 0 Pairs exploited: 0	Mean: -0.003 mm Std: 0.59 mm T: (-4.0, 14.2,-8.54) Angle: 13.3 Pairs exploited: 2	Mean: 0.001 mm Std: 0.58 mm T: (1.41, -14.4,10.6) Angle: 19.8 Pairs exploited: 2
3	A1	A2	A3	Mean: -0.71 mm Std: 1.86 mm T: (0.00,0.00,0.00) Angle: 0 Pairs exploited: 0	Mean: -0.44 mm Std: 1.33 mm T: (12.7,-12.9,52.7) Angle: 62.8 Pairs exploited: 1	Mean: -0.31 mm Std: 0.67 mm T: (12.6, -40.5, 38.8) Angle: 25.2 Pairs exploited: 1
	A6	A5	A4	Mean: -0.07 mm Std: 0.33 mm T: (-43.4, -31.9, 109) Angle: 47.6 Pairs exploited: 9	Mean: -0.13 mm Std: 0.53 mm T: (50.7, 8.29, 0.24) Angle: 36.5 Pairs exploited: 1	Mean: -0.20 mm Std: 0.69 mm T: (-5.31, 6.88, 31.9) Angle: 10.8 Pairs exploited: 9
4	A1	A3	A2	Mean: -0.07 mm Std: 0.53 mm T: (0.00,0.00,0.00) Angle: 0 Pairs exploited: 0	Mean: -0.29 mm Std: 0.75 mm T: (4.93, 5.8, -12.1) Angle: 6.77 Pairs exploited: 4	Mean: 0.32 mm Std: 0.82 mm T: (-15.7, 11.9, -7.5) Angle: 13.9 Pairs exploited: 5
5	A1	A2		Mean: -0.03 mm Std: 0.45 mm T: (0.00,0.00,0.00) Angle: 0 Pairs exploited: 0	Mean: -0.33 mm Std: 1.14 mm T: (7.7, -4.3, -0.13) Angle: 5.17 Pairs exploited: 1	
	A3	A4		Mean: -0.07 mm Std: 0.41 mm T: (-1.2, -3.7, 2.9) Angle: 5.1 Pairs exploited: 1	Mean: -0.51 mm Std: 1.37 mm T: (44.2, -5.75, -14.5) Angle: 23.11 Pairs exploited: 1	
6	A1	A2	A3	Mean: -0.008 mm Std: 0.42 mm T: (0.00,0.00,0.00) Angle: 0 Pairs exploited: 0	Mean: 0.04 mm Std: 0.37 mm T: (2.08, -3.98, -0.56) Angle: 3.92 Pairs exploited: 1	Mean: -0.31 mm Std: 1.28 mm T: (-2.00, -12.2,0.10) Angle: 20.1 Pairs exploited: 4
	A4	A5	A6	Mean: 0.009 mm Std: 0.43 mm T: (7.50, 10.4, 1.2) Angle: 14.3 Pairs exploited: 2	Mean: -0.49 mm Std: 1.41 mm T: (24.3, 23.4, -1.06) Angle: 19.2 Pairs exploited: 8	Mean: 0.02 mm Std: 0.41 mm T: (3.7, 6.5, 0.01) Angle: 8.29 Pairs exploited: 2

Figure 5.2: (left column) Local alignment error histograms for individual fragments in each case are shown. (right column) Statistical error analysis for these errors are shown which include the mean and standard deviation of each histogram. Also provided are each fragment's displaced distance as an (x,y,z) vector, $T(mm)$, and the angular displacement as a single angle (degrees) between broken position and the reconstructed position. The pairs exploited value indicates how many point pairs were tried before the accepted "correct" match was found.

the time needed to run the puzzle-solving algorithm. Also shown are the number of surface points in the intact surface mesh for each case which effects this time as shown in figure 5.39c. The reconstruction time recoded for each case includes time spent for computing spin images, matching spin images, and aligning fragment surfaces. The time spent for surface partitioning, surface patch classification, and initialization of puzzle-solving are not include here, see Chapter 6 for these system performance variables. From the table we can see that as the number of points on intact template increases more completion time of reconstruction. This is reasonable because more points result in more hypotheses in the puzzle solving process. Indeed, the recorded reconstruction time is influenced by several variables such as the number of spin images computed from the intact template and the fragment surfaces, number of fragments, number of iterations of the ICP algorithms, etc. See 5.7 for more detailed discussion.

5.1 Case 1 Reconstruction Analysis

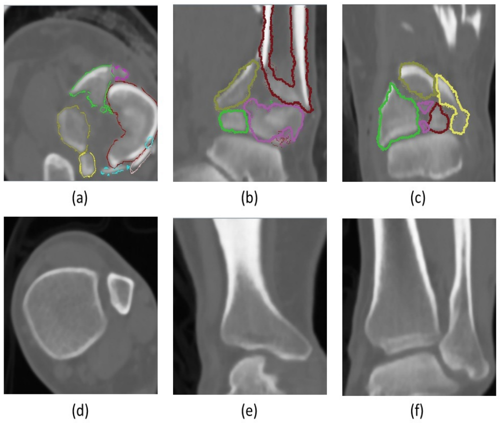


Figure 5.3: Case 1: CT images: (a, b, c) show CT images of the fractured limb from the axial, sagittal, and coronal views after segmentation. The fragment surface boundaries are shown as different colors in each image. (d, e, f) show CT images of the intact limb from the axial, sagittal, and coronal views.

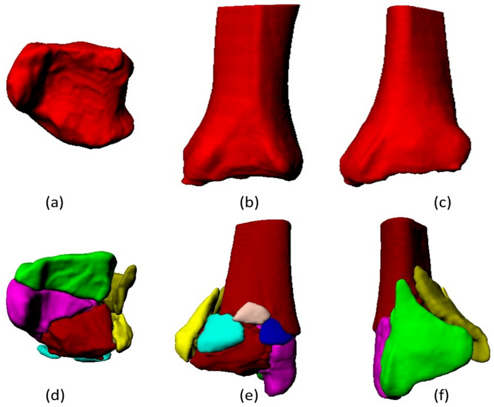


Figure 5.4: Case 1: Segmented Intact & Fragment Surfaces: (a, b, c) show axial, sagittal, and coronal views of the intact template bone surface. (d, e, f) show axial, sagittal, and coronal views of the fragment surfaces before reconstruction.

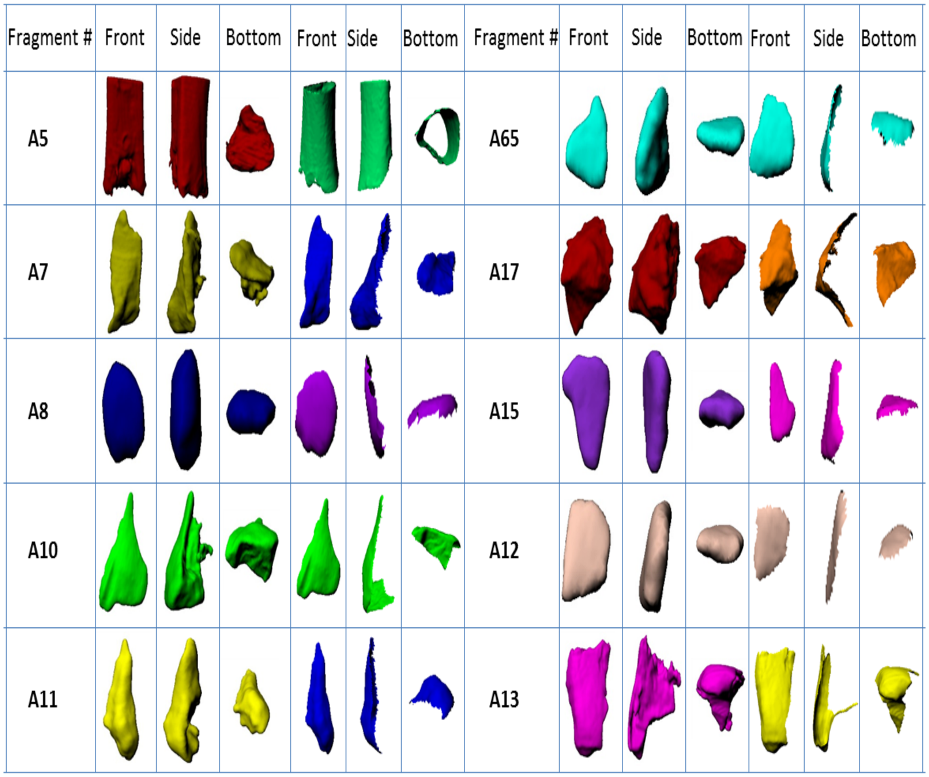


Figure 5.5: Case 1: Fragment Overviews: Naming conventions for each case fragment and three views of each fragment and the outer surface of each fragment.

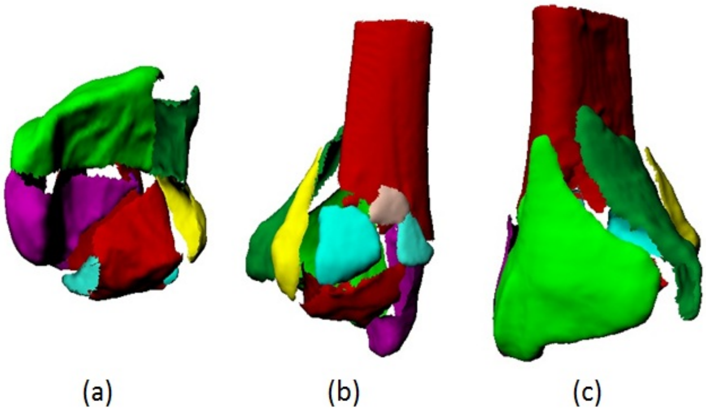


Figure 5.6: Case 1: Fragment Outer Surfaces: (a, b, c) axial, sagittal, and coronal views of the segmented fragment outer surfaces.

Table 5.4: Case 1: Fragment Mesh Analysis Table: This table shows the number of points, triangles, and the total surface area of each fragment and each fragments outer surface. The ratio of the area of the outer surface to the entire fragment surface area is also provided.

Fragment #	fragment			fragment outer surface			% outer area
	# points	# triangles	area (mm^2)	# points	# triangles	area (mm^2)	
A5	16359	32718	6487.6	9336	18293	3524.7	35.9%
A7	5312	10620	2104.2	2602	4927	990.1	10.1%
A8	223	442	85.4	65	132	34.6	0.4%
A10	8987	17970	3554.8	4395	8481	1688.6	17.2%
A11	2373	4742	938.6	1183	2196	428.5	4.3%
A65	1515	3026	597.6	796	1483	292.4	3%
A17	6186	12340	2420.22	1607	3011	616.4	6.3%
A15	732	1460	287.3	406	726	139.1	1.4%
A12	594	1184	232.77	594	1184	232.77	2.7%
A13	9595	19160	3755.81	3263	6241	1216.5	12.4%
Intact	24935	49367	9825.1	24935	49367	9825.1	100%

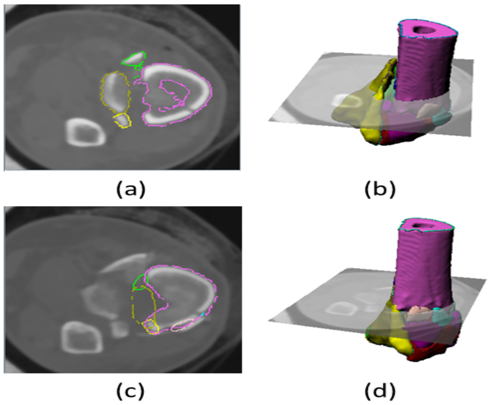
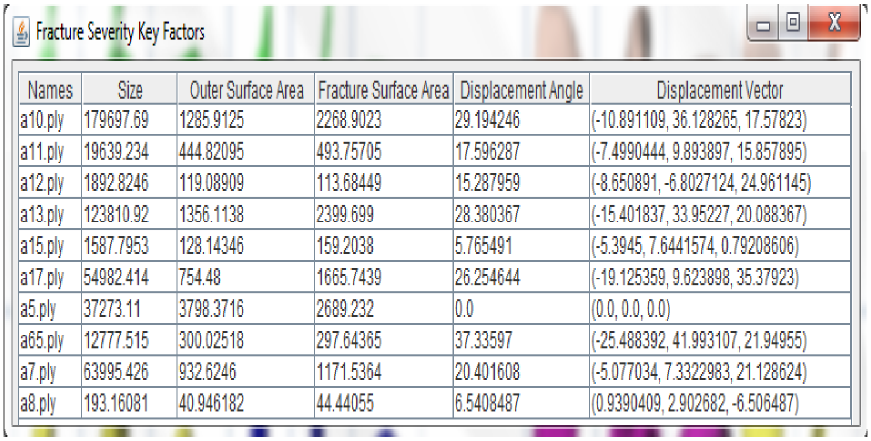


Figure 5.7: Case 1: Mixed Image & Surface Data: (a, b) show the fracture case before reconstruction. (a) shows a CT image of the fracture fragments (outlined in color). (b) shows the CT image from (a) in 3D together with the fragment surfaces. (c, d) show the fracture case after reconstruction. (c) shows the CT image from (a) where the fracture fragment (outlined in color) are shown in their *reconstructed* positions. (d) shows the same CT image from (c) in 3D together with the *reconstructed* fragment surfaces.



Names	Size	Outer Surface Area	Fracture Surface Area	Displacement Angle	Displacement Vector
a10.ply	179697.69	1285.9125	2268.9023	29.194246	(-10.891109, 36.128265, 17.57823)
a11.ply	19639.234	444.82095	493.75705	17.596287	(-7.4990444, 9.893897, 15.857895)
a12.ply	1892.8246	119.08909	113.68449	15.287959	(-8.650891, -6.8027124, 24.961145)
a13.ply	123810.92	1356.1138	2399.699	28.380367	(-15.401837, 33.95227, 20.088367)
a15.ply	1587.7953	128.14346	159.2038	5.765491	(-5.3945, 7.6441574, 0.79208606)
a17.ply	54982.414	754.48	1665.7439	26.254644	(-19.125359, 9.623898, 35.37923)
a5.ply	37273.11	3798.3716	2689.232	0.0	(0.0, 0.0, 0.0)
a65.ply	12777.515	300.02518	297.64365	37.33597	(-25.488392, 41.993107, 21.94955)
a7.ply	63995.426	932.6246	1171.5364	20.401608	(-5.077034, 7.3322983, 21.128624)
a8.ply	193.16081	40.946182	44.44055	6.5408487	(0.9390409, 2.902682, -6.506487)

Figure 5.8: Case 1: Fracture Severity Analysis Report: A screenshot of the table containing values for several key factors computed for each fragment from the 3D reconstruction. The units for the key factors shown in the table are as follows: mm^3 for size, mm^2 for surface area, degrees for the displacement angle, and mm for the displacement vector.

This case contains 10 different fragments whose sizes range from large pieces like fragment A5 and A7 to small pieces like fragment A8. Since bone fragments and the intact template bone segmentations are performed outside the system, the time required for this step is not discussed. After each fragment was partitioned, generated surface patches were manually merged and classified using the system interface. Some fragments such as A5 and A10 were easy to process, i.e., outer surfaces were partitioned quickly and correctly without any further user interaction necessary. Fragments A17 and A13 were problematic and required multiple runs of the surface partitioning algorithm with different parameters and additional user interaction to cope with over-segmentation and under-segmentation. All of the fragments were partitioned using the default parameter value ($\rho_{ridge} = 0.2$) for the ridge-walking algorithm as discussed in 3.2.1 with the exception of fragment A13 whose parameter value was $\rho_{ridge} = 0.4$.

The 3D reconstruction results for case one is shown in row one of figure 5.1 and the statistical analysis of the surface alignment errors for each fragment are shown in row one in figure 5.2. The reconstruction for this case used the automatic puzzle solving algorithm introduced in Chapter 4. The fragments were assembled successfully by algorithm with the exception of fragment A17 which required the use of the jiggle algorithm to fix mis-alignment. The total time spent to complete the automatic fragment alignment algorithm for case one is 140s as recorded in row one in table 5.3. Fragments, A11, A7, A17 required the majority of the processing time to align as shown by the Pair Exploited values of 5, 4, 5, for these fragments shown in row one of figure 5.2. At least four matched pairs were required for all fragments before the good alignment positions were discovered by the puzzle solving algorithm. The average global alignment error was 0.23 mm for case one as shown in table 5.2, which is relatively small and is considered to be a successful reconstruction.

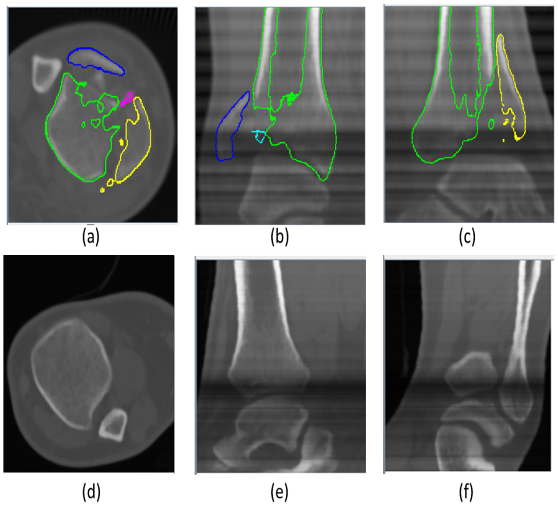


Figure 5.9: Case 2: CT images: (a, b, c) show CT images of the fractured limb from the axial, sagittal, and coronal views after segmentation. The fragment surface boundaries are shown as different colors in each image. (d, e, f) show CT images of the intact limb from the axial, sagittal, and coronal views.

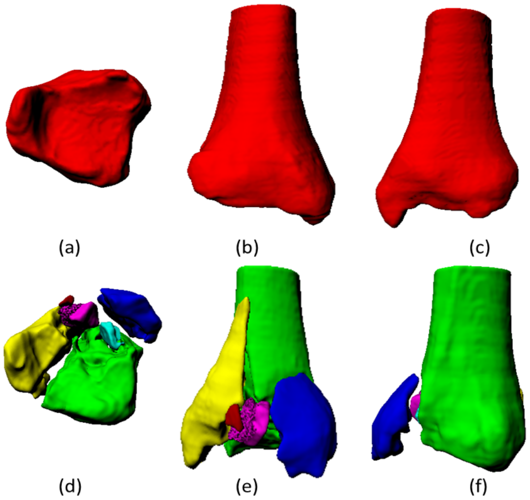


Figure 5.10: Case 2: Segmented Intact & Fragment Surfaces: (a, b, c) show axial, sagittal, and coronal views of the intact template bone surface. (d, e, f) show axial, sagittal, and coronal views of the fragment surfaces before reconstruction.

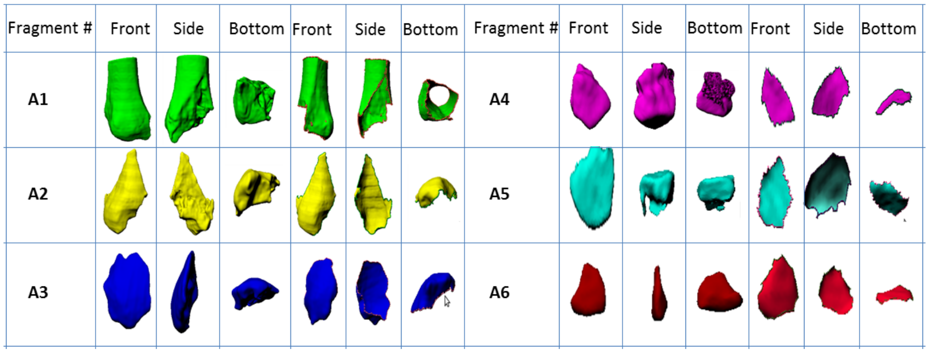


Figure 5.11: Case 2: Fragment Overviews: Naming conventions for each case fragment and three views of each fragment and the outer surface of each fragment.

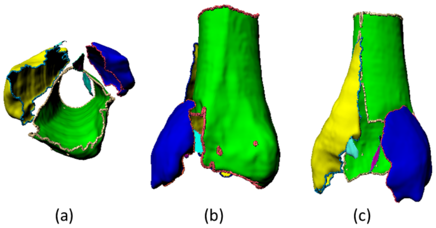


Figure 5.12: Case 2: Fragment Outer Surfaces: (a, b, c) axial, sagittal, and coronal views of the segmented fragment outer surfaces.

Table 5.5: Case 2: Fragment Mesh Analysis Table: This table shows the number of points, triangles, and the total surface area of each fragment and each fragments outer surface. The ratio of the area of the outer surface to the entire fragment surface area is also provided.

Fragment #	fragment			fragment outer surface			% outer area
	# points	# triangles	area (mm^2)	# points	# triangles	area (mm^2)	
A1	87845	175318	11709.4	42296	83427	5721.8	46.3%
A2	34120	68200	4389.7	18813	36993	2348.1	19%
A3	17971	35936	2277.7	10701	21052	1318.3	10.7%
A4	2717	5428	338.8	640	1178	103	0.8%
A5	4300	8594	754.2	646	1171	73.6	0.6%
A6	326	646	138.9	121	200	44.8	0.5%
Intact	45530	91098	18081.2	30903	61619	12367.2	100%

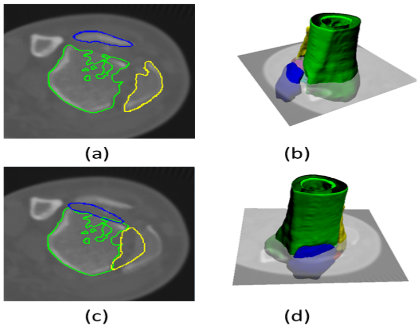
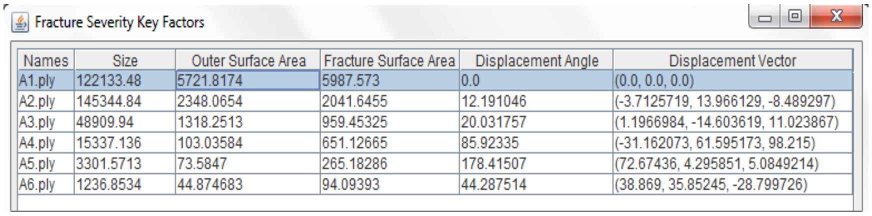


Figure 5.13: Case 2: Mixed Image & Surface Data: (a, b) show the fracture case before reconstruction. (a) shows a CT image of the fracture fragments (outlined in color). (b) shows the CT image from (a) in 3D together with the fragment surfaces. (c, d) show the fracture case after reconstruction. (c) shows the CT image from (a) where the fracture fragment (outlined in color) are shown in their *reconstructed* positions. (d) shows the same CT image from (c) in 3D together with the *reconstructed* fragment surfaces.



Names	Size	Outer Surface Area	Fracture Surface Area	Displacement Angle	Displacement Vector
A1.ply	122133.48	5721.8174	5987.573	0.0	(0.0, 0.0, 0.0)
A2.ply	145344.84	2348.0654	2041.6455	12.191046	(-3.7125719, 13.966129, -8.489297)
A3.ply	48909.94	1318.2513	959.45325	20.031757	(1.1966984, -14.603619, 11.023867)
A4.ply	15337.136	103.03584	651.12665	85.92335	(-31.162073, 61.595173, 98.215)
A5.ply	3301.5713	73.5847	265.18286	178.41507	(72.67436, 4.295851, 5.0849214)
A6.ply	1236.8534	44.874683	94.09393	44.287514	(38.869, 35.85245, -28.799726)

Figure 5.14: Case 2: Fracture Severity Analysis Report: A screenshot of the table containing values for several key factors computed for each fragment from the 3D reconstruction. The units for the key factors shown in the table are as follows: mm^3 for size, mm^2 for surface area, degrees for the displacement angle, and mm for the displacement vector.

This case contains six different fragments with widely ranging sizes including large pieces like fragment A1 to small pieces like fragment A6. Some fragments such as A2 and A3 were easy to process, i.e., outer surfaces were partitioned quickly and correctly without any further user interaction necessary. Fragments A1 and A4 were problematic and required multiple runs of the surface partitioning algorithm with different parameters and additional user interaction to cope with over-segmentation and under-segmentation. All of the fragment surfaces were partitioned using the default value $\rho_{ridge} = 0.2$, with the exception of fragments A1 and A4. Their best outcome values were obtained using $\rho_{ridge} = 0.4$ and $\rho_{ridge} = 0.5$ respectively.

The 3D reconstruction results for case two are shown in row two in figure 5.1 and the statistical analysis of the surface alignment errors for each fragment are shown in row two of figure 5.2. The reconstruction for this case used the automatic puzzle solving algorithm introduced in Chapter 4. Most of the fragments were assembled successfully by the automatic algorithm with the exception of fragments A4 and A5 which required the use of the jiggle algorithm. The jiggle algorithm also fails to align these two fragments, and manual positioning is required to reconstruct them. Fragments A4 and A5 have outer surfaces that are articular surfaces. These surfaces can be difficult to match because they contain many planar surface areas. These fragments are also far away from their original positions. These two factors make fragments A4 and A5 difficult to accurately reconstruct using the proposed reconstruction algorithm. The manual reconstruction interface as described in Chapter 2 is necessary to address fragments such as these efficiently. The total time spent by the automatic algorithm for case two is 220s as recorded in row one of Table 5.3. Most of this time is spent building the spin image stack of the intact template. As shown in Table 5.5, fragments in this case contain many surface points. Although it only has six fragments, the high density of the surface points on these fragments make the automatic surface alignment process a time-consuming task. However, as shown in row two of figure 5.2 fragments A2 and A3 exploited only two pairs before finding a successful match. This indicates that time spent to align fragment surfaces for these fragments is less. The global alignment error for this fracture is an average of 0.27 mm for case two as shown in table 5.2.

5.3 Case 3 Reconstruction Analysis

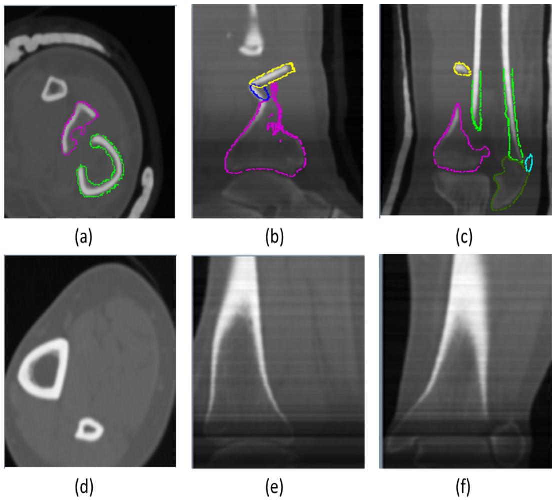


Figure 5.15: Case 3: CT images: (a, b, c) show CT images of the fractured limb from the axial, sagittal, and coronal views after segmentation. The fragment surface boundaries are shown as different colors in each image. (d, e, f) show CT images of the intact limb from the axial, sagittal, and coronal views.

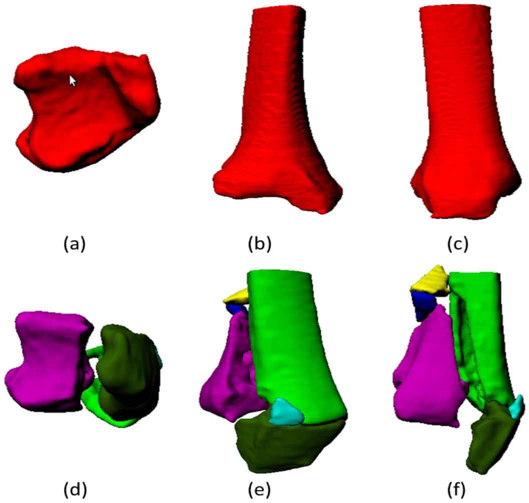


Figure 5.16: Case 3: Segmented Intact & Fragment Surfaces: (a, b, c) show axial, sagittal, and coronal views of the intact template bone surface. (d, e, f) show axial, sagittal, and coronal views of the fragment surfaces before reconstruction.

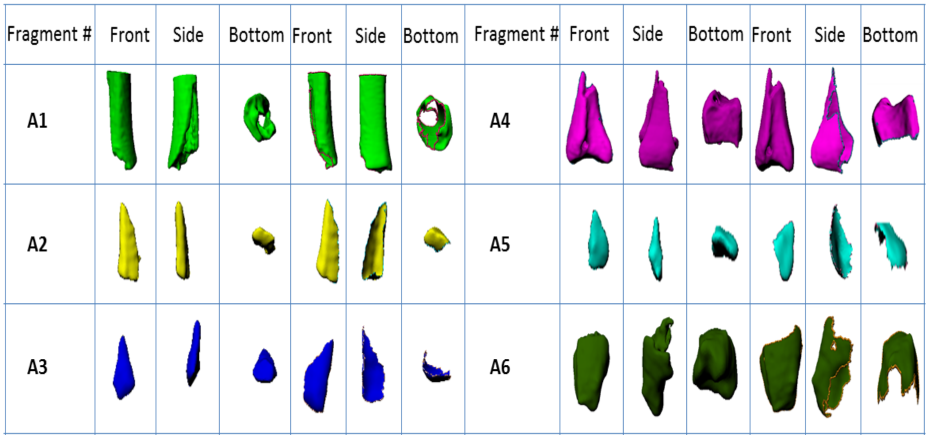


Figure 5.17: Case 3: Fragment Overviews: Naming conventions for each case fragment and three views of each fragment and the outer surface of each fragment.

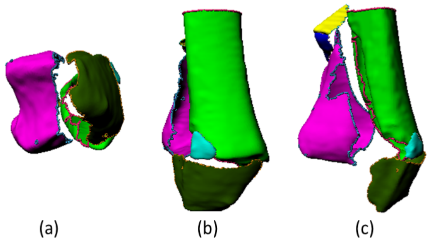


Figure 5.18: Case 3: Fragment Outer Surfaces: (a, b, c) axial, sagittal, and coronal views of the segmented fragment outer surfaces.

Table 5.6: Case 3: Fragment Mesh Analysis Table: This table shows the number of points, triangles, and the total surface area of each fragment and each fragments outer surface. The ratio of the area of the outer surface to the entire fragment surface area is also provided.

Fragment #	fragment			fragment outer surface			% outer area
	# points	# triangles	area (mm^2)	# points	# triangles	area (mm^2)	
A1	29007	57837	8193.2	17853	35152	5001.2	33.9%
A2	3106	6206	844.7	1765	3384	463.9	3.1%
A3	956	1906	244.3	781	1448	196.6	1.3%
A4	23780	47558	6419.9	14672	28895	3966.4	26.9%
A5	946	1886	252.7	483	873	119.4	0.8%
A6	11653	23288	3181.7	4674	9034	1233.7	8.3%
Intact	71696	139984	20136.5	37043	73873	14755.5	100%

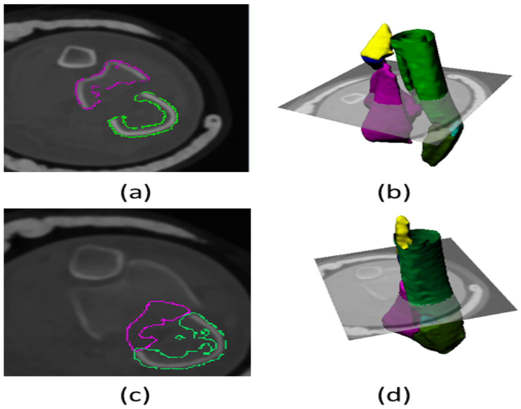
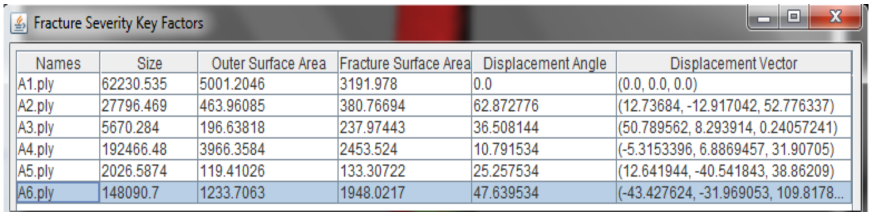


Figure 5.19: Case 3: Mixed Image & Surface Data: (a, b) show the fracture case before reconstruction. (a) shows a CT image of the fracture fragments (outlined in color). (b) shows the CT image from (a) in 3D together with the fragment surfaces. (c, d) show the fracture case after reconstruction. (c) shows the CT image from (a) where the fracture fragment (outlined in color) are shown in their *reconstructed* positions. (d) shows the same CT image from (c) in 3D together with the *reconstructed* fragment surfaces.



Names	Size	Outer Surface Area	Fracture Surface Area	Displacement Angle	Displacement Vector
A1.ply	62230.535	5001.2046	3191.978	0.0	(0.0, 0.0, 0.0)
A2.ply	27796.469	463.96085	380.76694	62.872776	(12.73684, -12.917042, 52.776337)
A3.ply	5670.284	196.63818	237.97443	36.508144	(50.789562, 8.293914, 0.24057241)
A4.ply	192466.48	3966.3584	2453.524	10.791534	(-5.3153396, 6.8869457, 31.90705)
A5.ply	2026.5874	119.41026	133.30722	25.257534	(12.641944, -40.541843, 38.86209)
A6.ply	148090.7	1233.7063	1948.0217	47.639534	(-43.427624, -31.969053, 109.8178...

Figure 5.20: Case 3: Fracture Severity Analysis Report: A screenshot of the table containing values for several key factors computed for each fragment from the 3D reconstruction. The units for the key factors shown in the table are as follows: mm^3 for size, mm^2 for surface area, degrees for the displacement angle, and mm for the displacement vector.

This case consists of six different fragments whose sizes range from large pieces like fragment A1 to small pieces like fragment A3 and A5. The surfaces of all fragments were partitioned using the ridge walking algorithm with default value, $\rho_{ridge} = 0.2$, which generated a segmentation of the outer surfaces quickly and correctly without any additional need for user interaction.

The 3D reconstruction results for case three are shown in row three of figure 5.1 and the statistical analysis of the surface alignment errors for each fragment are shown in row three of figure 5.2. As shown in the figure 5.2, fragments A4 and A6 required 9 different pairs to be tried before a successful match was found. Fragments such as these are considered difficult for the puzzle-solving algorithm to correctly match. One interesting aspect of these two fragments is that they both include high curvature ridges and valleys on their outer surfaces. Such structures are typically distinctive and often are quickly matched by the puzzle-solving algorithm. This is caused by the rough surfaces on the intact template (CT image segmentation errors) which makes surface matching step finding many false matches. The puzzle-solving algorithm matched together fragments A2 and A3 after trying only one pair. In this case, the match was incorrect and as a result, these fragments were mis-aligned in the automatic reconstruction result. These two fragments are difficult to reconstruct because their outer surfaces are non-distinctive and flat. The total time spent by the automatic algorithm for case three is 272s as recorded in row three of Table 5.3. Most of this time was spent aligning the difficult fragments: A4 and A6. The average global alignment error for this fracture was 0.32 mm as shown in table 5.2.

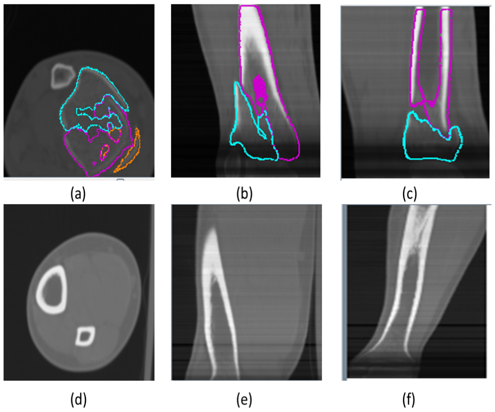


Figure 5.21: Case 4: CT images: (a, b, c) show CT images of the fractured limb from the axial, sagittal, and coronal views after segmentation. The fragment surface boundaries are shown as different colors in each image. (d, e, f) show CT images of the intact limb from the axial, sagittal, and coronal views.

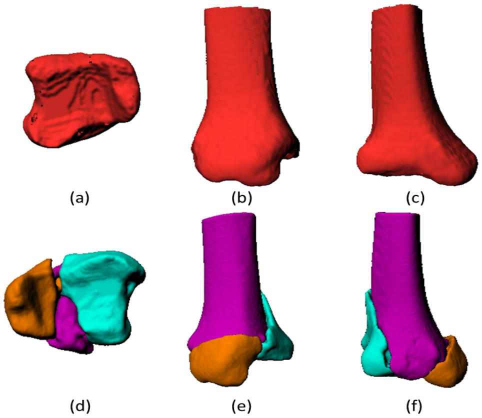


Figure 5.22: Case 4: Segmented Intact & Fragment Surfaces: (a, b, c) show axial, sagittal, and coronal views of the intact template bone surface. (d, e, f) show axial, sagittal, and coronal views of the fragment surfaces before reconstruction.

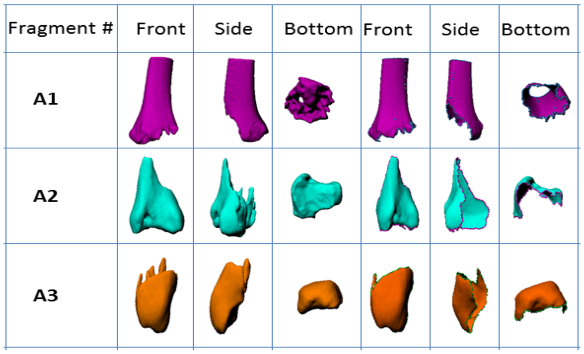


Figure 5.23: Case 4: Fragment Overviews: Naming conventions for each case fragment and three views of each fragment and the outer surface of each fragment.

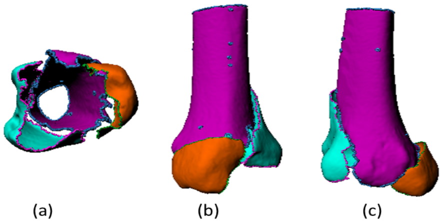


Figure 5.24: Case 4: Fragment Outer Surfaces: (a, b, c) axial, sagittal, and coronal views of the segmented fragment outer surfaces.

Table 5.7: Case 4: Fragment Mesh Analysis Table: This table shows the number of points, triangles, and the total surface area of each fragment and each fragments outer surface. The ratio of the area of the outer surface to the entire fragment surface area is also provided.

Fragment #	fragment			fragment outer surface			% outer area
	# points	# triangles	area (mm^2)	# points	# triangles	area (mm^2)	
A1	36046	72100	14311.2	16536	32407	6713.7	E4.5%
A2	15194	30384	5991.5	8649	16915	3272.5	26.6%
A3	8324	16644	3287.2	4729	9245	1810.3	14.7%
Intact	89863	179710	35940.4	23408	46618	12324.5	100%

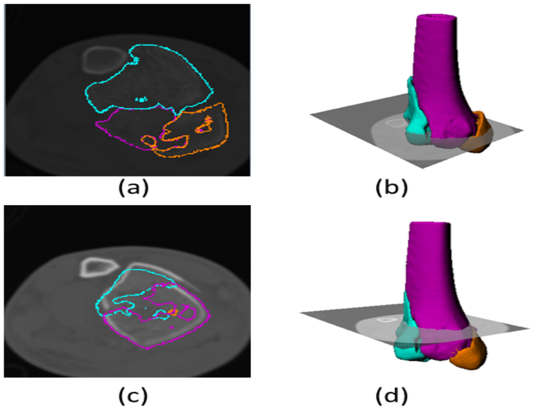
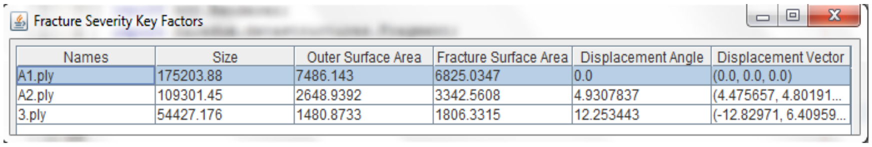


Figure 5.25: Case 4: Mixed Image & Surface Data: (a, b) show the fracture case before reconstruction. (a) shows a CT image of the fracture fragments (outlined in color). (b) shows the CT image from (a) in 3D together with the fragment surfaces. (c, d) show the fracture case after reconstruction. (c) shows the CT image from (a) where the fracture fragment (outlined in color) are shown in their *reconstructed* positions. (d) shows the same CT image from (c) in 3D together with the *reconstructed* fragment surfaces.



Names	Size	Outer Surface Area	Fracture Surface Area	Displacement Angle	Displacement Vector
A1.ply	175203.88	7486.143	6825.0347	0.0	(0.0, 0.0, 0.0)
A2.ply	109301.45	2648.9392	3342.5608	4.9307837	(4.475657, 4.80191...
3.ply	54427.176	1480.8733	1806.3315	12.253443	(-12.82971, 6.40959...

Figure 5.26: Case 4: Fracture Severity Analysis Report: A screenshot of the table containing values for several key factors computed for each fragment from the 3D reconstruction. The units for the key factors shown in the table are as follows: mm^3 for size, mm^2 for surface area, degrees for the displacement angle, and mm for the displacement vector.

This case contains three large fragments. Because of the small number of fragments and the large size for these fragments, this case is considered to be a low energy fracture case that is easy to puzzle-solve automatically. The 3D reconstruction results for this case are shown in row four of figure 5.1 and the statistical analysis of the surface alignment errors for each fragment are shown in row four of figure 5.2. Figure 5.2 shows that the rotation angle and translation vector for each of the two fragments are relatively small which supports the hypothesis that this is a low energy fracture case. All the fragments were assembled successfully by the automatic puzzle-solving algorithm. The total time spent by the automatic algorithm to reconstruct case four is 90s as recorded in row four in table 5.3. Figure 5.2 shows that fragments A2 and A3 required at least 4 matches before they were correctly aligned. Despite the fact that only two fragments were puzzle-solved their large size and high point density make this reconstruction task relatively time-consuming. This is due to the fact that the individual fragment matching algorithm is not optimized for performance. The average global alignment error for this fracture was 0.34 mm as shown in table 5.2.

5.5 Case 5 Reconstruction Analysis

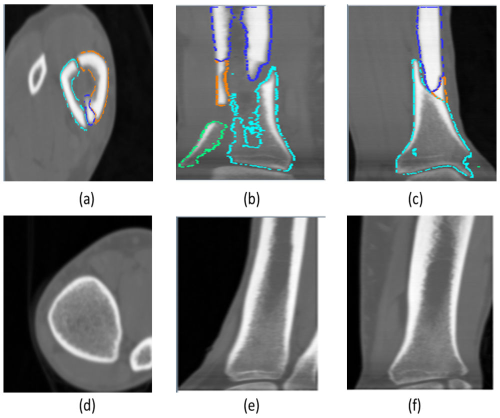


Figure 5.27: Case 5: CT images: (a, b, c) show CT images of the fractured limb from the axial, sagittal, and coronal views after segmentation. The fragment surface boundaries are shown as different colors in each image. (d, e, f) show CT images of the intact limb from the axial, sagittal, and coronal views.

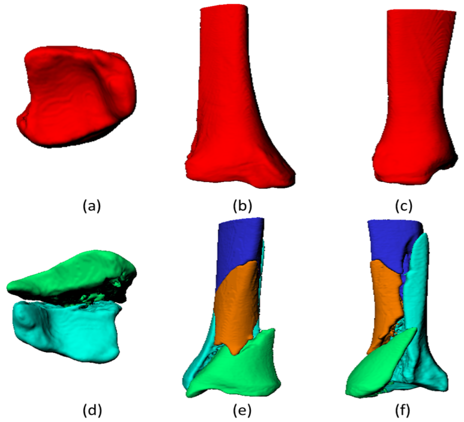


Figure 5.28: Case 5: Segmented Intact & Fragment Surfaces: (a, b, c) show axial, sagittal, and coronal views of the intact template bone surface. (d, e, f) show axial, sagittal, and coronal views of the fragment surfaces before reconstruction.

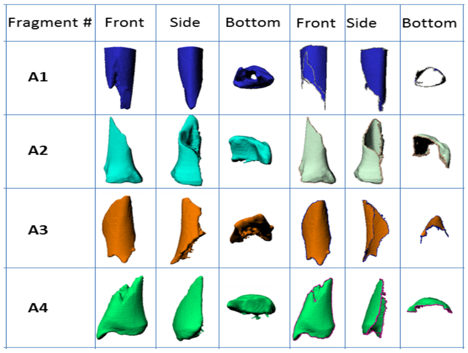


Figure 5.29: Case 5: Fragment Overviews: Naming conventions for each case fragment and three views of each fragment and the outer surface of each fragment.

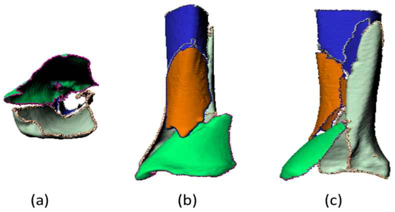


Figure 5.30: Case 5: Fragment Outer Surfaces: (a, b, c) axial, sagittal, and coronal views of the segmented fragment outer surfaces.

Table 5.8: Case 5: Fragment Mesh Analysis Table: This table shows the number of points, triangles, and the total surface area of each fragment and each fragments outer surface. The ratio of the area of the outer surface to the entire fragment surface area is also provided.

Fragment #	fragment			fragment outer surface			% outer area
	# points	# triangles	area (mm^2)	# points	# triangles	area (mm^2)	
A1	20299	40598	8054.9	11064	21440	4337.2	23.2%
A2	36385	72834	14352.9	20109	39441	7796.7	41.7%
A3	14388	28780	5679.9	7027	13559	2688.4	14.4%
A4	18255	36422	7090.2	8145	15854	3101.8	16.6%
Intact	68160	136240	26833.5	47411	94351	18692.9	100%

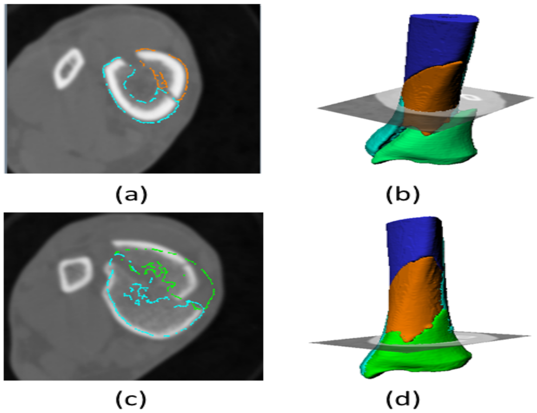
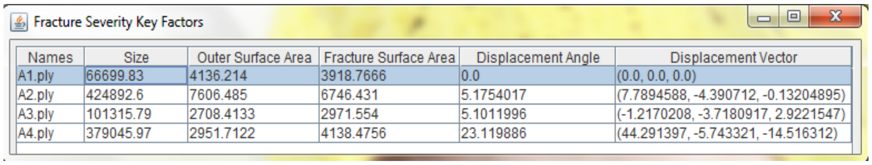


Figure 5.31: Case 5: Mixed Image & Surface Data: (a, b) show the fracture case before reconstruction. (a) shows a CT image of the fracture fragments (outlined in color). (b) shows the CT image from (a) in 3D together with the fragment surfaces. (c, d) show the fracture case after reconstruction. (c) shows the CT image from (a) where the fracture fragment (outlined in color) are shown in their *reconstructed* positions. (d) shows the same CT image from (c) in 3D together with the *reconstructed* fragment surfaces.



Fracture Severity Key Factors

Names	Size	Outer Surface Area	Fracture Surface Area	Displacement Angle	Displacement Vector
A1.ply	66699.83	4136.214	3918.7666	0.0	(0.0, 0.0, 0.0)
A2.ply	424892.6	7606.485	6746.431	5.1754017	(7.7894588, -4.390712, -0.13204895)
A3.ply	101315.79	2708.4133	2971.554	5.1011996	(-1.2170208, -3.7180917, 2.9221547)
A4.ply	379045.97	2951.7122	4138.4756	23.119886	(44.291397, -5.743321, -14.516312)

Figure 5.32: Case 5: Fracture Severity Analysis Report: A screenshot of the table containing values for several key factors computed for each fragment from the 3D reconstruction. The units for the key factors shown in the table are as follows: mm^3 for size, mm^2 for surface area, degrees for the displacement angle, and mm for the displacement vector.

This case contains four different large fragments. All of the fragments were partitioned using the ridge walking algorithm with the default value, $\rho_{ridge} = 0.2$, for this algorithm. The resulting outer surfaces were correctly partitioned by the system. The 3D reconstruction results for this case are shown in row five of figure 5.1 and the statistical analysis of surface alignment errors for each fragment are shown in row five of figure 5.2. All four fragments were successfully reconstructed by the automatic algorithm, no jiggle algorithm or manual positioning was needed. The total time spent by the automatic algorithm to reconstruct case five was 430s as recorded in row five in table 5.3. There were only four fragment pieces to align for this case and all of them were correctly aligned after trying only one pair. However, since each fragment has a large number of surface points (see table 5.8), the reconstruction was time-consuming. The average global alignment error was 0.33 mm for this fracture as shown in table 5.2.

5.6 Case 6 Reconstruction Analysis

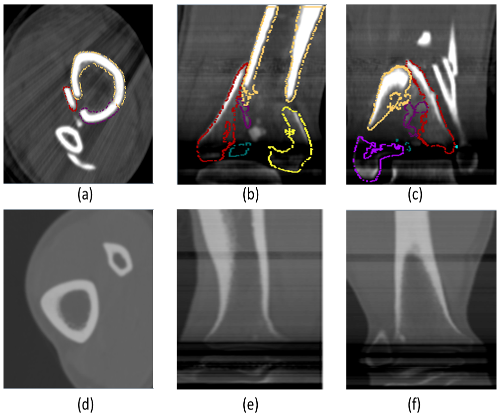


Figure 5.33: Case 6: CT images: (a, b, c) show CT images of the fractured limb from the axial, sagittal, and coronal views after segmentation. The fragment surface boundaries are shown as different colors in each image. (d, e, f) show CT images of the intact limb from the axial, sagittal, and coronal views.

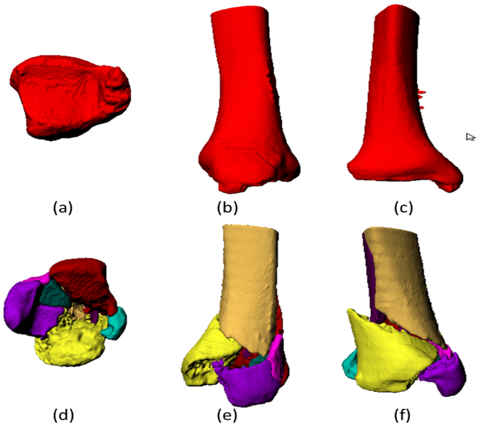


Figure 5.34: Case 6: Segmented Intact & Fragment Surfaces: (a, b, c) show axial, sagittal, and coronal views of the intact template bone surface. (d, e, f) show axial, sagittal, and coronal views of the fragment surfaces before reconstruction.

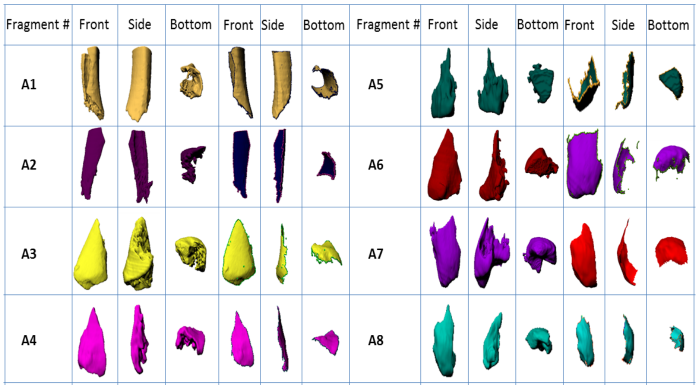


Figure 5.35: Case 6: Fragment Overviews: Naming conventions for each case fragment and three views of each fragment and the outer surface of each fragment.

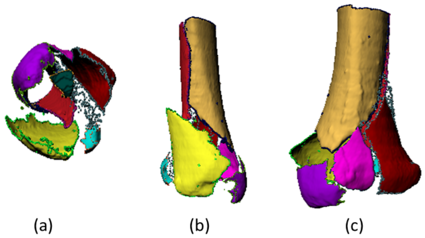


Figure 5.36: Case 6: Fragment Outer Surfaces: (a, b, c) axial, sagittal, and coronal views of the segmented fragment outer surfaces.

Table 5.9: Case 6: Fragment Mesh Analysis Table: This table shows the number of points, triangles, and the total surface area of each fragment and each fragments outer surface. The ratio of the area of the outer surface to the entire fragment surface area is also provided.

Fragment #	fragment			fragment outer surface			% outer area
	# points	# triangles	area (mm^2)	# points	# triangles	area (mm^2)	
A1	39160	78344	9817.7	18052	35019	4693.9	23.7%
A2	20307	40614	5333.5	7380	14248	2026.1	10.2%
A3	33586	67044	7695.9	9196	17732	2182.6	11%
A4	7045	14086	1638.9	2434	4632	559.9	3%
A5	5548	11092	1229.2	892	1645	132.7	0.6%
A6	30430	60886	6898.8	14198	27768	3025.4	15.2%
A7	16476	32956	3590.4	7587	14823	1659.7	8.3%
A8	4305	8606	964.2	679	1238	144	0.7%
Intact	117549	235106	30450	75974	151575	19835.1	100%

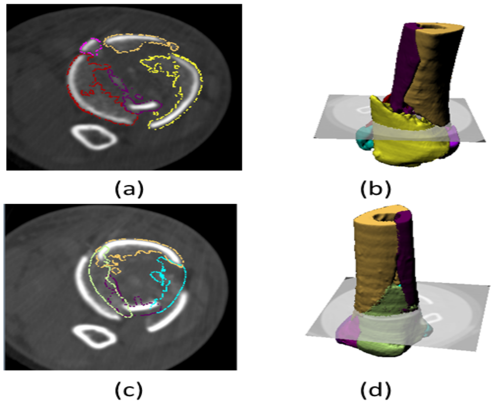
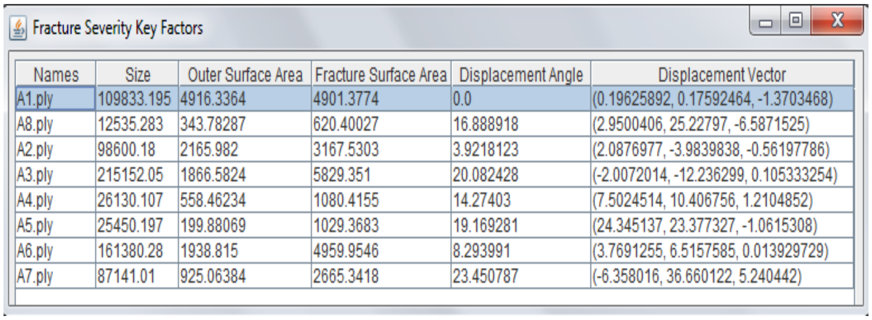


Figure 5.37: Case 6: Mixed Image & Surface Data: (a, b) show the fracture case before reconstruction. (a) shows a CT image of the fracture fragments (outlined in color). (b) shows the CT image from (a) in 3D together with the fragment surfaces. (c, d) show the fracture case after reconstruction. (c) shows the CT image from (a) where the fracture fragment (outlined in color) are shown in their *reconstructed* positions. (d) shows the same CT image from (c) in 3D together with the *reconstructed* fragment surfaces.


Fracture Severity Key Factors

Names	Size	Outer Surface Area	Fracture Surface Area	Displacement Angle	Displacement Vector
A1.ply	109833.195	4916.3364	4901.3774	0.0	(0.19625892, 0.17592464, -1.3703468)
A8.ply	12535.283	343.78287	620.40027	16.888918	(2.9500406, 25.22797, -6.5871525)
A2.ply	98600.18	2165.982	3167.5303	3.9218123	(2.0876977, -3.9839838, -0.56197786)
A3.ply	215152.05	1866.5824	5829.351	20.082428	(-2.0072014, -12.236299, 0.105333254)
A4.ply	26130.107	558.46234	1080.4155	14.27403	(7.5024514, 10.406756, 1.2104852)
A5.ply	25450.197	199.88069	1029.3683	19.169281	(24.345137, 23.377327, -1.0615308)
A6.ply	161380.28	1938.815	4959.9546	8.293991	(3.7691255, 6.5157585, 0.013929729)
A7.ply	87141.01	925.06384	2665.3418	23.450787	(-6.358016, 36.660122, 5.240442)

Figure 5.38: Case 6: Fracture Severity Analysis Report: A screenshot of the table containing values for several key factors computed for each fragment from the 3D reconstruction. The units for the key factors shown in the table are as follows: mm^3 for size, mm^2 for surface area, degrees for the displacement angle, and mm for the displacement vector.

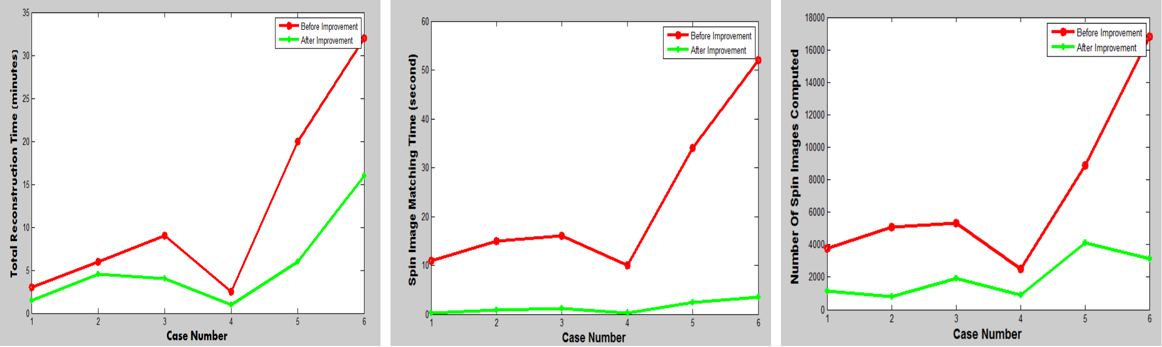
This case contains eight different fragments whose sizes range from large pieces like fragment A1 and A3 to small pieces like fragment A5. Fragments A1 and A2 were easy to partition using the ridge walking algorithm with the default parameter, $\rho_{ridge} = 0.2$. Yet fragment A5 was problematic and required multiple runs of the ridge

Table 5.10: This table shows quantitative values for the performance improvements observed for each case. Using the system enhancements discussed in §4.4.3.

Case	t_{total} before (min)	t_{total} after (min)	Avg matching before (sec)	Avg matching after (sec)	n_{intact} before	n_{intact} after	# Points On Template
1	3	1.5	11	0.3	3750	1125	24935
2	6	4.5	15	0.8	5065	802	45529
3	8	5	16	1.2	5320	1913	50539
4	2.5	1	10	0.2	2509	897	33630
5	20	6	34	2.4	8890	4135	68160
6	31	16	52	3.5	16829	3120	117549

walking algorithm with different parameters and additional user interaction to cope with over-segmentation and under-segmentation to generate an acceptable surface partitioning was produced for this fragment.

The 3D reconstruction results for case six are shown in row six of figure 5.1 and the statistical analysis of surface alignment errors for each fragment are shown in row six of figure 5.2. Most of the fragments were assembled successfully by the automatic puzzle-solving algorithm with the exception of fragments A5 and A8. Fragments A5 and A8 have outer surfaces that are articular surfaces. These surfaces can be difficult to match because they contain small planar surface areas. This make these fragments difficult to accurately reconstruct using the proposed reconstruction algorithm. The total time spent by the automatic algorithm for case six was 650s as recorded in row six of table 5.3. This case has a lot of densely sampled fragments and the matching and alignment of these fragments was time-consuming. The average global alignment error was 0.27mm for this fracture as shown in table 5.2. There are also voids in the solution, i.e., empty regions where no fragment has been matched, as shown in the bottom view of this reconstruction in figure 5.1. It is suspected that this voids is a missing fragment or from fragment deformation.



(a) (b) (c)

Figure 5.39: (a,b,c) show three metrics related to the computational cost of the puzzle-solving algorithm. For each metric, the value of the metric is shown with and without the enhancement discussed in §4.4.3 for each of the six clinical cases (the x-axis). The metrics are as follows: (a) shows improvements of the total reconstruction time for the puzzle-solving algorithm, (b) shows improvements of the average matching time for the puzzle-solving algorithm, and (c) shows the number of spin images computed on the intact template.

5.7 Performance Improvement Results

As mentioned in 4.4.3, the mean curvature histogram biased search algorithm significantly improves the speed of the automatic puzzle-solving algorithm by reducing the number of spin images computed for both the intact template surface and the fragment surfaces. The puzzle-solving algorithm is a complex process which consists of many steps, and the time spent for the reconstruction is affected by several sub-steps such as computing the spin images for the intact template and the fragment surfaces, matching spin images, and aligning fragment surfaces to the intact template. In order to better understand the improvements, the following equation (5.1) details the time spent for reconstruction for each step.

$$t_{total} = t_{overhead} + n_{intact} t_{spin} + \left(t_{spin} n_{frag} + t_{matchspin} + t_{filter} + t_{align}\right) M_{frag} \qquad (5.1)$$

In this equation, $t_{overhead}$ denotes the time spent for processing the intact template before computing the spin images on the intact template such as surface sampling, computing occupied regions and computing the mean curvature histogram for the intact template. The term $n_{intact} t_{spin}$ denotes the time spent to identifying feature points on the intact template and compute their spin images. The term $t_{spin}n_{frag}$ denotes the time spent identifying feature points on the fragment surfaces and computing their spin images. The term $t_{matchspin}$ denotes time spent generating hypothesized surface correspondences which is affected by n_{frag} and n_{intact}. The term t_{filter} denotes the time spent removing false matches. The term t_{align} denotes time spent aligning the fragment surface to the intact template which is affected by number of hypothesized correspondences being tested. The term M_{frag} denotes the number of fragments in the fracture case. The major factors that impact the total reconstruction time are n_{frag}, the number of spin images computed on the fragment surface, and n_{intact}, the number spin images computed on the intact template. The mean curvature histogram approach in 4.4.3 reduces the total reconstruction time by reducing both n_{intact} and n_{frag} significantly as shown in figure 5.39c. Table 5.10 shows the quantitative values for total reconstruction time, t_{total}, average time spent for matching spin images and filtering matches, $t_{matchspin} + t_{filter}$, and n_{intact} for each case. Figure 5.39 shows plots that graphically depict the performance improvements for each of these factors. Note that all reconstruction times were running on a laptop computer with 2.4GHz dual core CPU with 4GB memory.

5.8 Summary of Results

Figure 5.1 shows the reconstructed results for the six clinical cases using the automated puzzle-solving algorithm. Visually, one can see that the articular surfaces of the reconstructed fractures are smooth across fragment boundaries with the exception of case two and case six. Other errors can be seen in some reconstructions such as the yellow and the blue fragments of case three which are mis-aligned by the algorithm. This can be explained, in part, by their large initial displacement and flat and indistinct outer surfaces. Matching flat surfaces such as these is difficult as there are many candidate matches which have similar geometric alignment error. This fact, in conjunction with the large initial displacement of some fragments present difficulties to the puzzle-solving algorithm as it searches for the true correspondences within the large group of similar candidates. From table 5.2, it can be seen that the overall global alignment error is small. This can be explained by the fact that the majority of the bone fragments are not plastically deformed and they can be accurately fit into the intact template and to other adjacent fragments. Figure 5.2 provides more information and some important quantitative values for each individual fragment in each case. An alignment error histogram for each fragment is given and the statistical mean and standard deviation of each histogram are shown. The provided quantitative information for each fragment's translation and rotation is hoped to provide users an objective measure for the fragment dispersion and displacement which may improve the accuracy and reliability for estimating fracture severity.

CHAPTER 6: USER STUDY

Because the system has not been used by anyone other than its developer, the usability of it by a generic user is still unknown. This chapter discusses a user study conducted to evaluate the usability of the system. The user study examines how several bone reconstruction tasks are accomplished by a select group of untrained users. Measurements from the study and user feedback allow a better understanding of key issues regarding the effectiveness of the system. The user study seeks to answer the following questions: (1) does the automatic reconstruction tools benefit users? If yes, how do users benefit from it, and which aspects are most beneficial? (2) Is there any difference between the reconstruction results using manual fragment alignment and automatic fragment alignment? If yes, how significant is the difference? (3) Is the software interface acceptable to users? and (4) What things can be done to improve the system interface?

The content of this chapter is organized as following: 6 describes the experimental set up for the user study, including user selection, scope of the study, and specific design tasks; 6.1 shows the results and feedback from each user; 6.2 draws conclusions from the gathered results and feedback.

User Study Experimental Set Up

This section discusses the set up for the user study experiments. There are four major aspects addressed: (1) selection of users, (2) methodology, (3) tasks, and (4) results. The following paragraphs will describe each aspect in details.

Selection of users: Ideally, these users should be trained radiological technicians. However, such expertise was unavailable for user studies. Three untrained users were selected for this user study. Users are referred as user A, user B, and user C. User A is a stay-at-home mother with no prior virtual 3D experience. User B is a software engineer having significant prior experience with virtual 3D environments. User C is an electrical engineering student with limited experience on 3D virtual environments.

Methodology: Two clinical cases were selected from Chapter 5: case one (a difficult case to puzzle-solve) and case four (an easy case to puzzle-solve). Users were given a brief introduction to the software. Then the less difficult case (case four) was reconstructed for them by the dissertation author as an example of how to perform 3D bone reconstruction using the software. After the demonstration, users were asked to perform the bone reconstruction on the same case. Once they finished, they were asked to reconstruct the more difficult case (case one). After completing these tasks, users were asked to complete the survey (shown in figure 6.1). This survey collected the user's feedback on specific functions, interactions, interface design and additional comments or suggestions about the software system.

Tasks: The bone reconstruction process consists of four main tasks and each task is broken into smaller sub-tasks in the following list:

1. Surface partition and classification:

   (a) Load intact CT and fracture CT images.

   (b) Load intact template and fracture 3D fragment surfaces.

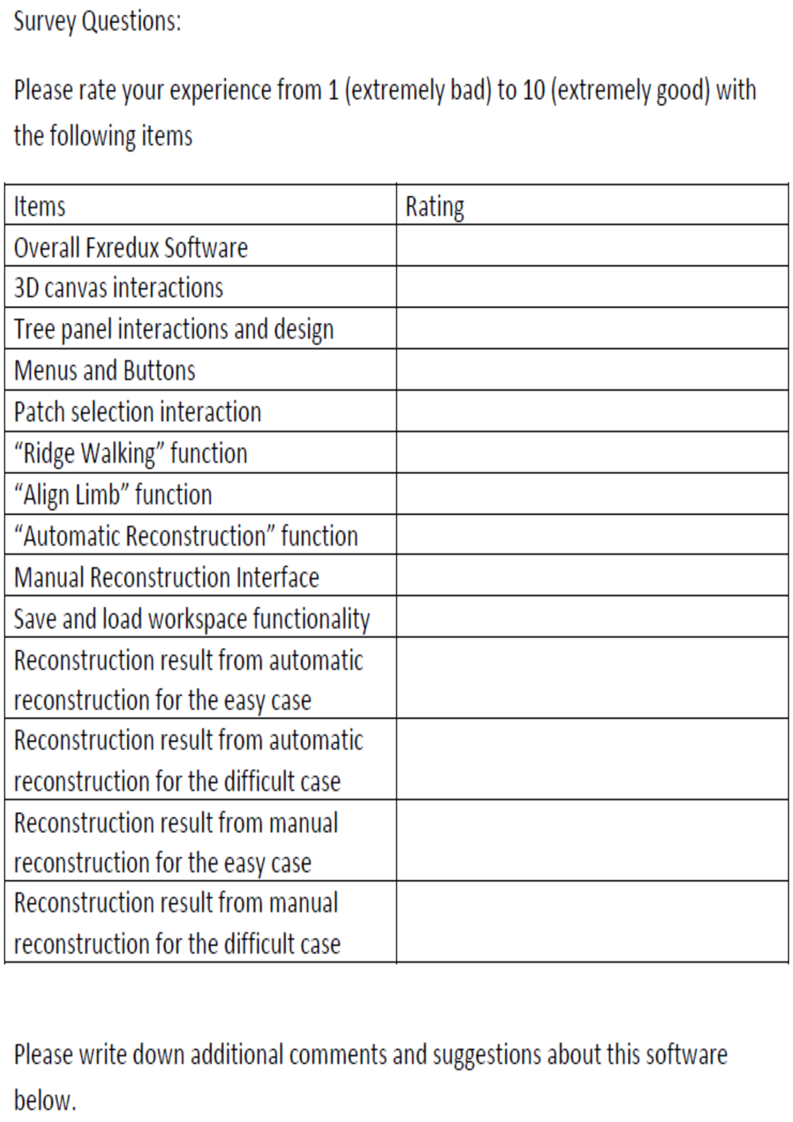
   (c) Run the "Ridge Walking" algorithm which partitions the 3D fragment surfaces.

   (d) Manually classify the resulting 3D fragment surface patches as periosteal, fracture, or articular surfaces.

   (e) Save workspace.

2. Initialization:

   (a) Select periosteal patches from the intact 3D bone model and a periosteal patch from the base fragment.



Survey Questions:

Please rate your experience from 1 (extremely bad) to 10 (extremely good) with the following items

Items	Rating
Overall Fxredux Software	
3D canvas interactions	
Tree panel interactions and design	
Menus and Buttons	
Patch selection interaction	
“Ridge Walking” function	
“Align Limb” function	
“Automatic Reconstruction” function	
Manual Reconstruction Interface	
Save and load workspace functionality	
Reconstruction result from automatic reconstruction for the easy case	
Reconstruction result from automatic reconstruction for the difficult case	
Reconstruction result from manual reconstruction for the easy case	
Reconstruction result from manual reconstruction for the difficult case	

Please write down additional comments and suggestions about this software below.

Figure 6.1: This figure shows a screen capture of the survey.

   (b) Run the “Align Limb” function.

   (c) Save workspace.

3. Automatic reconstruction:

   (a) Run the “Automatic Reconstruction” algorithm.

   (b) Run the “Key Severity Factors ” function to generate a severity report.

   (c) Run the “Error Analysis”.

   (d) Save workspace.

4. Manual reconstruction:

   (a) Load workspace saved at the end of step 2.

   (b) Manually align each fragment to the 3D surface of the intact template.

   (c) Run the “Key Severity Factors ” function to generate a severity report.

   (d) Run the “Error Analysis” function.

   (e) Save workspace.

Results: The results of the user study include 3D reconstruction results, feedback and notes. The results are as follows:

1. Time spent accomplishing task 1- 4 for each case.
2. Visual and quantitative measures for automatic reconstruction results for each case.
3. Visual and quantitative measures for manual reconstruction results for each case
4. Survey feedback.
5. Notes recording any unusual activities, failure or system crash.

Experimental values were recorded and analyzed to answer the questions posed as the goals of the user study. Since users were untrained, if they could not proceed because of some unexpected errors or confusion, assistance was provided. In each case where assistance was necessary, the context and reason was recorded. During the experiments users did not receive any assistance in solving the fracture case and made their own decisions on how to execute the reconstruction process. All experiments were executed on a laptop computer with 2.4GHz dual core CPU with 4GB memory.

6.1 User Study Results

The study results are presented and discussed in this section. The following paragraphs show the compiled results from all three users. Each figure is followed by a short explanation. Additional notes taken during the experiments are discussed in section 6.1.1, 6.1.2, and 6.1.3.

Table 6.1: This table shows the time spent by each user on each of the four main tasks for each case.

Users	Easy Case (minutes)				Difficult Case (minutes)			
	Task 1	Task 2	Task 3	Task 4	Task 1	Task 2	Task 3	Task 4
User A	20	7	2	7	40	4	2	16
User B	15	3	1.5	3	25	3	2	27
User C	5	4	1	7	20	4	1.5	20
Avg	13.3	4.67	1.5	5.67	28.3	3.67	1.8	21

Table 6.1 shows the time spent by each user on each of the four main tasks for each case. From the table, Task 1 and Task 4 are the most time-consuming tasks for all users since these tasks involve a lot of user interaction. Task 2 and Task 3 are the least time consuming tasks for all users, since automatic algorithms did most of the computation and decisions. From this table, we can find that tasks involving significant user interaction tend to be more time-consuming, and the time spent on tasks can vary significantly between users. User C outperforms user B and user A on most tasks. User A is slow on Task 1 for both cases (surface partition and classification) and User B is slow on Task 4, the Difficult Case. Based on the time table, it is clear that the automatic algorithm saves a lot of time for users. Total time saved by the automatic puzzle-solving algorithm for three users are 12.9 minutes for case four and 57.5 minutes for case one.

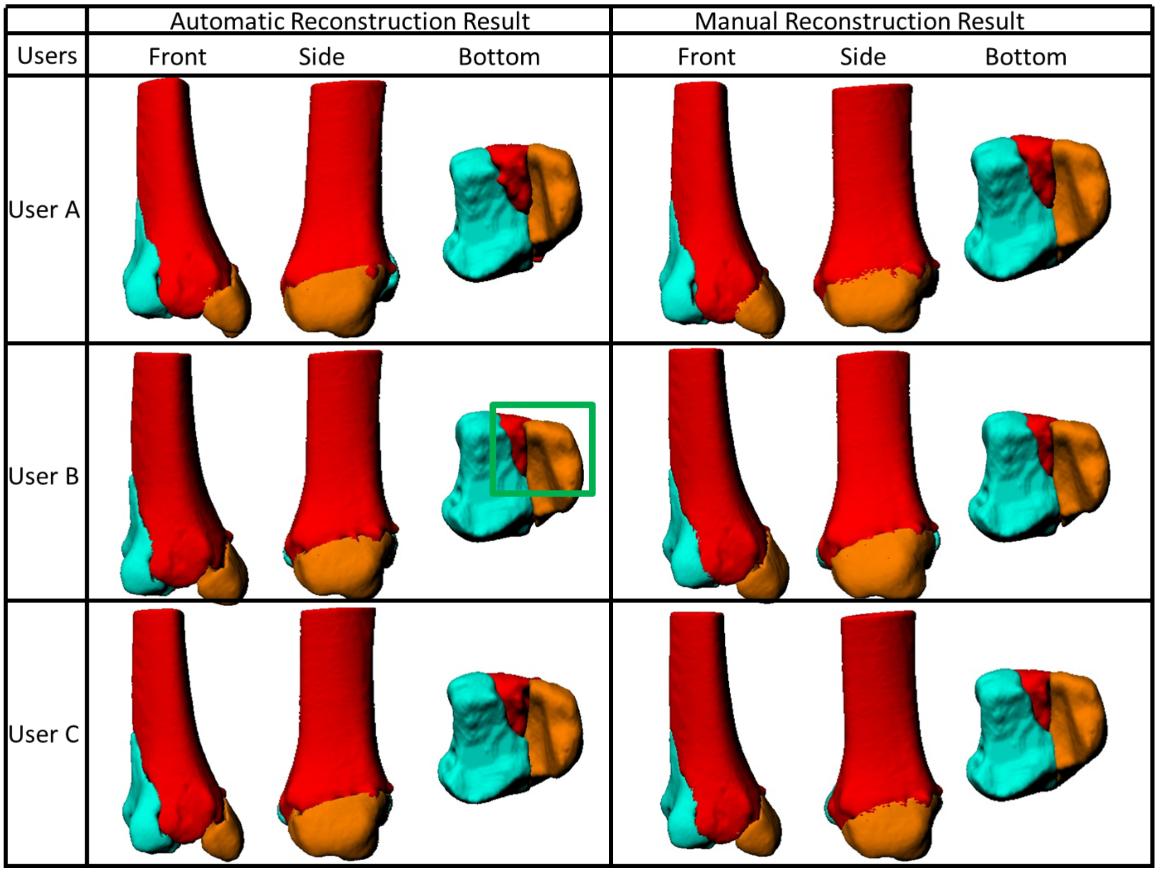


Figure 6.2: Three views of the automatic and manual reconstruction results for the easy case are shown for each user.

Figure 6.2 shows three views of reconstructed bone for the easy case by the three users. Automatically reconstructed solutions are shown in the left column and manually reconstructed solutions are shown in the right column. The figure shows that reconstructed results between different users are very similar, and for each user there are noticeable differences between the automatic approach and manual approach for reconstruction. As marked with the green box in figure 6.2, the reconstructed result using the automatic approach by User B is slightly different than other users. This is due to the fact that User B generated different periosteal surfaces in Task 2 than other users. In this case, the periosteal patch generated for the dark yellow fragment by User B represents only a part of the periosteal surface for that fragment. This adversely impacts User B's reconstruction result using the manual approach and results in a slight mis-alignment for this fragment.

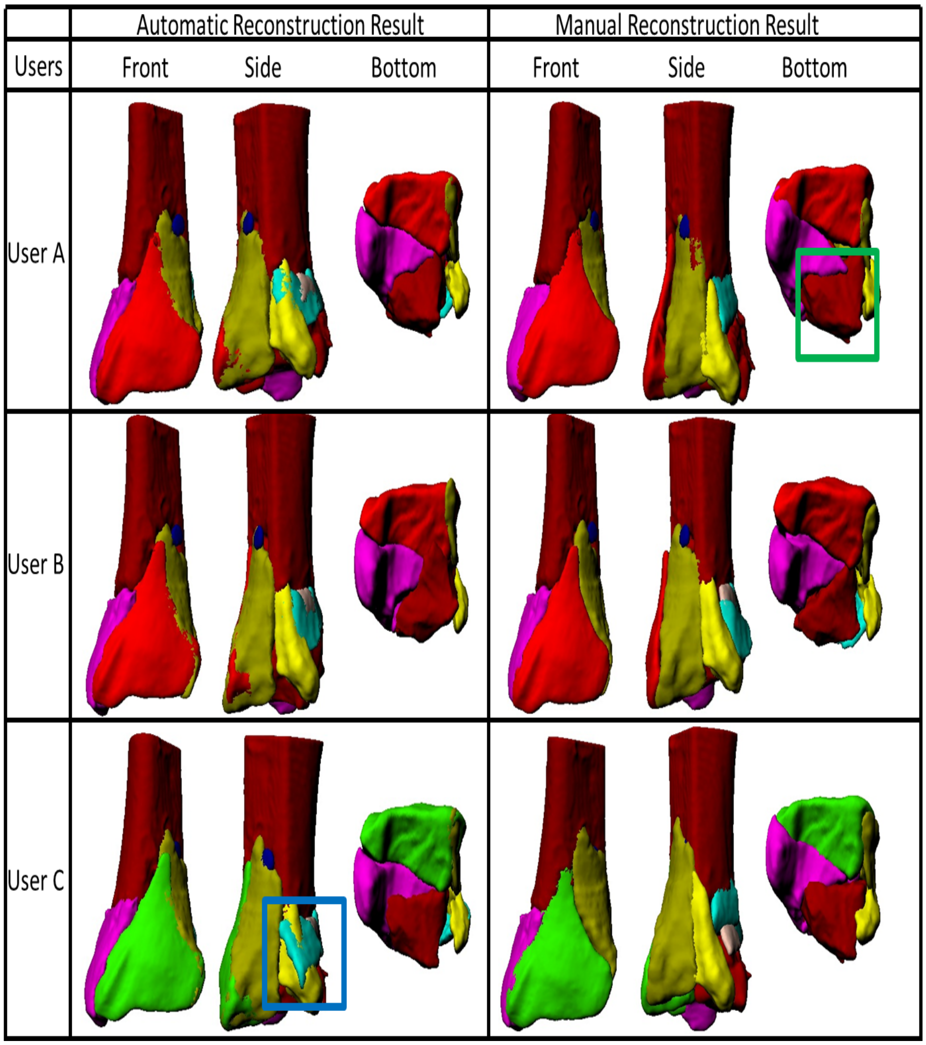


Figure 6.3: This figure shows three anatomic views of the users automatic and manual reconstruction results for the difficult case.

Figure 6.3 shows three views of the users automatic and manual bones reconstructions for the difficult case. One can see that the reconstruction results are quite different for both automatic and manual solutions. A fragment marked with the green box in the figure denotes one problematic fragment that is the most difficult fragment to align for both automatic and manual approach. This fragment is called A13 and is shown in dark red in the green box. A13 is difficult to align because its periosteal surface has a lot of noise and segmentation error. The "Ridge Walking" algorithm could not successfully segment the correct periosteal surface for this fragment. As a result, users had to manually select the periosteal patch using on their own judgment and without any assistance. In the experiment, users created different periosteal surface patches for the A13 fragment and this resulted in different automatic reconstruction results. This also posed problems during manual reconstruction. Because A13 has a bumpy and rough periosteal surface, users had difficulty in identifying corresponding points between the A13 fragment and the intact template. This caused reconstructed position for this fragment to be quite different between users when using the manual reconstruction approach. There are also some problematic flat surface fragments in this fracture case as shown in the blue box of figure 6.3. All users failed to accurately align these fragments using either the automatic or manual approach.

Table 6.2: This figure shows the average alignment error for the bone fragments of the easy and difficult cases.

Users	Easy Case (mm)		Difficult Case (mm)	
	Automatic	Manual	Automatic	Manual
User A	0.38	0.21	0.27	0.37
User B	0.28	0.31	0.40	0.65
User C	0.28	0.26	0.28	0.42

Table 6.3: This figure shows the percentage of unmatched area on the intact template. High numbers indicate less surface area was matched between the template and fragments. This often implies that higher number are associated with less-accurate solutions.

Users	Easy Case (%)		Difficult Case (%)	
	Automatic	Manual	Automatic	Manual
User A	4.99	6.39	14.67	29.20
User B	4.61	6.40	19.34	38.57
User C	3.82	5.41	20.22	23.32

Table 6.2 shows the average or global alignment error for each case and Table 6.3 shows percentage of unmatched area on the intact template. The above statistics confirm the observations made previously and support the conclusion that the

reconstruction results of the easy case are generally similar in accuracy, while the reconstruction results of the difficult case are significantly different in accuracy. Note that the intact template used in the difficult case contains a limb part that does not belong to any of the fragments. Even if all fragments are perfectly aligned with the intact template, the unmatched area percentage will not be zero.

Table 6.4: This table shows the result of survey from User A, B, and C. Users rate the software for a list of items using a 1-10 scale with 1 = poor and 10 = excellent.

Items	User A	User B	User C
Overall Fxredux Software	8	8.5	7.5
3D canvas interactions	5	7.5	9.5
Tree panel interactions and design	10	8	8
Menus and Buttons	8	7	9
Patch selection interaction	10	8	8
“Ridge Walking” function	7	9	9
“Align Limb” function	7	8.5	8.5
“Automatic Reconstruction” function	8	7.5	8.5
Manual Reconstruction Interface	10	8.5	8.5
Save and load workspace functionality	10	7	8
Results from automatic approach for easy case	8	7.5	7.5
Results from manual approach for easy case	8	8	7
Results from automatic approach for difficult case	10	9	8.5
Results from manual approach for difficult case	6	6	7.5

Table 6.4 shows the survey results from the three users. From the survey, we found that the software meets the usability requirement. All users gave scores above 7 for most interface actions. However, there are a few exceptions. For example, User A gave 5 rating for the 3D canvas interactions because she was not able to rotate fragments to the desired position. She was also not comfortable with her manual reconstruction results as reflected by the score of 6 for the manual reconstruction results for the difficult case (see 6.1.1 for additional details).

6.1.1 Additional Notes For User A

During the experiment, User A had a lot of failures. In the reconstruction experiment of the easy case, User A had trouble during the first 15 minutes with the 3D interactions, and found it difficult to move objects in 3D to a desired view. Afterward, she started to gain better control of the 3D objects using the interface. She was also confused by the differences between camera selection and object selection at first, and often forgot to select fragments from the tree panel to run algorithms. User A also made some interesting interactions during the experiments. She attempted to view selected fragments by double-clicking the fragment item from the tree panel. She also attempted to use the manual reconstruction interface to align the intact bone model to the base fragment model instead of using the initialization step for this purpose. However, within a few minutes, she abandoned this idea because it is difficult to define correspondences between these two objects. When User A was performing the last main task, manual reconstruction, she was very precise with the correspondence selection and repeated the refine alignment (ICP) to get a smaller alignment error. Every time she repeated the refinement she increased the number of sample points, and she figured out that using more sampling points on the surface would give her better results. This is why it took about 11 minutes to align only two fragments.

In the reconstruction experiment of the difficult case, User A became familiar with the software and started to identify some bugs or places to need improvement. She found the redo and undo functions were only applicable to fragment movements and complained about it. She also found that if she ran the "Ridge Walking" algorithm again, she had to delete previously generated patches manually, which was annoying to her. She suggested these issues should be resolved by the system. When she performed manual reconstruction, the time spent on viewing and learning the geometric shape of both the fragment and the intact template to identify correspondence surfaces was significantly increased. She found it very difficult to make a decision on selecting correspondences. For fragment A17 in the difficult case, she claimed that there was no correspondence between this fragment and the intact template. In this case, external help was provided that allowed the user to use the relative positions of all fragments to infer the correct surface correspondences necessary to complete the task.

For additional comments and suggestions, User A suggested that the view selected fragments function should be added as a part of the tool bar since it was a frequently used function. She also commented on the rotation interaction in 3D canvas that the overall control of object was acceptable but the precise rotational movement are difficult. She liked the workspace functionality and the usability of the manual reconstruction interface. She also thought the patch selection interface was user friendly with the exception of the visible selection which is too slow.

6.1.2 Additional Notes For User B

In the reconstruction experiment for the easy case, User B expected a visible response from the 3D canvas when he made selections on the tree panel. He also attempted to move items in the tree panel by dragging them. During Task 1, User B refused to use the surface selection tool because of the slow speed and long wait for selections which can take more than ten seconds. When the user performed the last main task in manual reconstruction, he was very quick with the correspondence selection and didn't use refine alignment (ICP) to get a smaller alignment error. The method by which he selected pairs of points on surfaces was quite different from User A. He proceeded by selecting all points on one surface then selecting their corresponding points on the other surface. He attempted to slide middle mouse button to zoom in and out, although he had been instructed to press down the middle mouse button. After he failed several times, he thought the zoom in and out was very hard to use because he was not accustomed to using the zoom operation in this way.

In the reconstruction experiment of the difficult case, User B became familiar with the software, and started to find ways to reduce the time needed for each task. Since he decided not to use the surface patch selection tool, he reduced the time spent on Task 1 by running the "Ridge Waking" algorithm for all fragments at the same time. Afterward, he searched the fragments for periosteal surface patches one by one.

He found that small patches generated by the algorithm always need to be deleted manually, and he thought this issue should be resolved by the algorithm automatically. When he was performing manual reconstruction (Task 4), he accidentally closed the software, and lost all previous aligned results. He had to start over again, and this is why his time spent on this task was longer than other users though he didn't use any refinement.

For additional comments and suggestions, User B suggested that when he selected item on the tree panel, something should appear in 3D canvas indicating the object selected in the tree panel. He also commented that a progress bar for the "Ridge Walking" algorithm should be added, otherwise he had no idea what was happening when this algorithm was working. Regarding the menu items and buttons, he suggested that these items should be ordered alphabetically or properly categorized so that users can easily find their desired functions. He thought the automatic reconstruction was helpful and would be his first choice to reassemble the fracture case.

6.1.3 Additional Notes For User C

During the experiment, User C was very comfortable using the 3D interactions such as rotation, zoom in and out, which allowed her to finish all tasks quickly. In the reconstruction experiment of the easy case, User C often forgot to select tree items before clicking the right mouse key, but she managed to change that habit when solving the next case. Because User C had great control over the 3D interactions, she could view and understand the fracture case better than the other two users. However, she had trouble making decisions on whether the alignment was good or not. Also she was confused between the selection of a fragment and a patch. In the reconstruction experiment of the difficult case, User C faced the same difficulty the other users faced as she could not use the manual reconstruction interface to align small flat fragments. However, she found a work-around quickly by using the original 3D canvas as a reference window and utilizing the relative position between fragments and the intact template to locate the corresponding surface areas. For additional comments and suggestions, User C suggested that the manual reconstruction interface should include another reference window where users could view all fragments simultaneously along with the intact template. She thought this improvement would benefit users when specifying corresponding points during the manual alignment.

6.2 User Study Conclusions

From this user study, we not only get answers to the set of questions put forward in the beginning of the Chapter, but also discover unexpected interesting things. From the reconstruction results shown in Figure 6.2 and Table 6.2, for the easy case, reconstruction results are quite similar between different users and different approaches, but for the difficult case, the results are quite different. Deviation of results between different users is attributed to different levels of understanding for the fracture case and different levels of abilities for handling the 3D interactions. The results also show that manual reconstruction doesn't provide better results than automatic reconstruction in terms of quantitative alignment error and visual assessment. All results from the manual approach seem inferior to the automatic approach. There are several reasons for this: (1) User B didn't fully utilize the functions provided by the manual interface, (2) User A didn't understand the fracture case, and (3) User A and User C were limited by the current functionality of the manual reconstruction interface. From the time recorded in table 6.1, it is clear that Task 4 (manual reconstruction) is the most time-consuming task. It is also clear that the automatic puzzle-solving algorithm saves a lot of time. Among all three users, User C outperformed other two users because she had better control over the 3D interactions and she could study and learn the fracture geometry pattern quicker. We believe that with proper training, users will achieve better performance and reconstruction results.

In conclusion, the findings of the user study show that the automatic reconstruction tool benefits users for both time efficiency and spatial accuracy. Comparative results between manual reconstructions and automatic reconstructions are quite different for the difficult case and similar for the easy case. The user's different levels of understanding about fracture cases and different abilities for controlling the system have the major impact on the variation of results. Overall, the usability of the software interface is acceptable to users. However, several parts of the interface can be further improved for a better user experience. For example, providing a reference 3D window in the manual reconstruction interface, categorizing the menu items and organizing them in alphabetical order, adding a progress bar to “Ridge Walking” algorithm.

CHAPTER 7: CONCLUSIONS & FUTURE WORK

The proposed system is capable of virtually reconstructing broken bone fragments for complex bone fracture cases, which is heretofore unsolved problem in automatic puzzle-solving algorithms and difficult to achieve using manual methods. The bone reconstruction system designed in this dissertation enables users to understand fracture cases from both 2D (CT image) and 3D (fragment surface) imagery. The system represents a unique combination of state-of-the-art 2D/3D image processing and surface processing algorithms. The software is a comprehensive reconstruction tool that guides users from the first step, i.e., segmenting raw CT image data, to the last step, i.e., generating quantitative evaluation information about the fracture's severity. Finally, 3D visualization of fragments surface can provide important information for surgical treatment, especially for articular fractures which often have a poor prognosis.

While it is intuitive that detailed pre-operative reconstruction plans provided software like this could improve surgical treatment, this has not been proven as a clinically effective tool. [88] efforts to design and execute a surgical simulation experiment for quantifying the utility of this tool in a well-controlled environment. However, due to the limited number of experiments, solid conclusive statements about clinical utility of the puzzle-solving software cannot be made. Both the puzzle-solving algorithm and the software system are significant advancements toward improving the treatment of comminuted tibial plafond fractures. The computational 3D puzzle solving framework provides a heretofore unavailable patient-specific blueprint for fracture reconstruction planning. Having a suitable blueprint for restoring the original anatomy, it becomes possible for the surgeon to pre-operatively explore less extensive surgical approaches, and to attempt new intra-operative approaches and may eventually result in improving how surgeons determine fracture severity and they puzzle-solve bone fractures.

7.1 Future Work

Although the proposed 3D puzzle solving application and algorithms offer a powerful new tool for improving surgical reconstruction of complex tibial plafond fractures, there are some limitations that need to be taken into account for future research before the system can be used in a clinical setting. The most important limitation is the assumption that a patient's healthy bone template is available as a reference. In reality we cannot always have a healthy bone template. This may occur when only the fractured limb is scanned to reduce cost or when the patient has broken both of their legs. In order for this technology to be clinically embraced, future work might investigate how to automatically generate a generic 3D template bone for each fracture case. The generated templates will need to take biological information of the patient such as age, sex, height and weight into account. Another limitation of the current system is that it does not consider the soft tissue around the bone in reconstruction process. Consideration of this information is complicated by the fact that soft tissue is difficult to segment within CT images. As a result, the system cannot provide quantitative information about several severity key factors that relate to soft tissue such as soft tissue damage and displaced soft tissue volume. In order to solve this problem, future work would seek to develop methods for segmenting soft tissues within CT images. Additionally, the puzzle-solving algorithm only provides the final aligned positions for each bone fragment, while physicians are also interested in knowing the specific trajectory each fragment must take in order to arrive at the final reconstructed position. The system will be more valuable to physicians and surgeons when these trajectories for each fragment can be computed such that the soft tissue damage due to reconstruction is minimized. Finally, survey feedback and suggestions for improving the system interface from the user study should be taken into consideration as the system evolves to suit these needs.

GLOSSARY

CAD models: Computer-Aided Designed is the use of computer technology for the process of design and design-documentation. The CAD models refers to 2-D and 3D models that are desinged using this technology.

CT: Computed Tomography, a medical imaging method employing tomography created by computer processing.

diaphyseal: Relating to a shaft of a long bone.

epiphysis: The end of long bone that is originally separated from the main bone by a layer of cartilage but that later becomes united to the main bone through ossification.

fluoroscopy: Fluoroscopy is an imaging technique commonly used by physicians to obtain real-time moving images of the internal structures of a patient through the use of a fluoroscope. In its simplest form, a fluoroscope consists of an X-ray source and fluorescent screen between which a patient is placed. However, modern fluoroscopes couple the screen to an X-ray image intensifier and CCD video camera allowing the images to be recorded and played on a monitor.

ICP: Iterative Closest Point is an popular algorithm employed to minimize the difference between two clouds of points. ICP is often used to reconstruct 2-D or 3D surfaces from different scans, to localize robots and achieve optimal path planning, to co-register bone models, etc.

metaphysis: The zone of growth between the epiphysis and diaphysis during development of a bone.

minimum spanning tree: In graph theory, a spanning tree of a graph is a tree composed of all the vertices and some of the edges pf the graph. A minimum spanning tree is minimum weighted edges that forms the spanning tree.

MRI: Magnetic Resonance Imaging is a medical imaging technique used in radiology to visualize detailed internal structures.

PACS: Picture Archiving and Communication System is a medical imaging technology which provides economical storage of images from multiple modalities.

PET: Positron Emission Tomography is nuclear medicine imaging technique that produces a three-dimension image or picture if functional processes in the body.

physis: The segment of a bone that is responsible for lengthening.

proximal: segments of long bone close to the middleline of body.

PTOA: Post-Traumatic OsteoArthritis which is the increased wear that occurs in a joint as the direct and indirect result of injury. Arthritis is a general term for painful inflammation of a joint. Osteoarthritis is the most common form and is due to wear of the joint surface. It is a process more than a disease and takes place slowly in all joints in all of us. It is complex in its details but the overall concept is simple.

tibial plafond fracture: A tibial plafond fracture occurs at the end of the shin bone and involves the ankle joint.

REFERENCES

[1] J. Marsh, J. Buckwalter, and R. Gelberman, “Articular fractures: Does an anatomic reduction really change the result?,” The Journal of Bone Joint Surgery, vol. 84-A(7), pp. 1259–71, 2002.

[2] C. Beardsley, D. Anderson, L. Marsh, and T. Brown, “Interfragmentary surface area as an index of comminution severity in cortical bone impact,” Journal of Orthopaedic Research, vol. 23, no. 3, pp. 680–690, 2005.

[3] C. Charalambous, M. Tryfonidis, F. Alvi, M. Moran, C. Fang, R. Samaraji, and P. Hirst, “Inter- and intra-observer variation of the schatzker and AO OTA classifications of tibial plateau fractures and a proposal of a new classification system,” The Journal of Bone and Joint Surgery, vol. 89, no. 4, pp. 400–404, 2007.

[4] R. Brumback and A. Jones, “Interobserver agreement in the classification of open fractures of the tibia. The results of a survey of two hundred and forty five orthopaedic surgeons,” The Journal of Bone and Joint Surgery, vol. 76, no. 8, pp. 1162–1166, 1994.

[5] T. P. Thomas, D. D. Anderson, L. J. Marsh, and T. D. Brown, “Displaced soft tissue volume as a metric of comminuted fracture severity,” in Annual Meeting of the American Society of Biomechanics, vol. 87, p. 309, August 2007.

[6] D. Anderson, T. Mosqueda, and T. Thomas, “Quantifying tibial plafond fracture severity: Absorbed energy and fragment displacement agree with clinical rank ordering,” Journal of Orthopaedic Research, vol. 26, no. 8, pp. 1046–1052, 2008.

[7] C. Rockwood, D. Green, and R. Bucholz, Rockwood and Greens Fractures in Adults. New York: Lippincott-Raven, 1996.

[8] B. Browner, J. Jupiter, and A. Levine, Skeletal Trauma Fractures Dislocations, Ligamentous Injuries. 1998.

[9] J. Bernstein, B. Monaghan, and J. Silber, “Taxonomy and treatment, a classification of fracture classifications,” Journal of Bone Joint Surgery, vol. 79, no. 5, pp. 706–707, 1997.

[10] M. Muller, S. Nazarian, and P. Koch, The Comprehensive Classification of Fractures of Long Bones. Berlin, Heidelberg, New York: Springer-Verlag, 1994.

[11] T. P. Ruedi, R. E. Buckley, and C. G. Moran, AO Principles of Fractures Management. 2007.

[12] R. Salter and W. Harris, “Injuries involving the epiphyseal plate,” The Journal of Bone and Joint Surgery, vol. 45, no. 3, pp. 587–622, 1963.

[13] J. Schatzker, R. McBroom, and D. Bruce, "The tibial plateau fracture. The Toronto experience," Clinical Orthopaedics Related Research, vol. 138, pp. 94–104, 1979.

[14] L. Audigi, M. Bhandari, and J. Kellam, "How reliable are reliability studies of fracture classifications? A systematic review of their methodologies," Acta Orthopaedica Scandinavica, vol. 75, no. 2, pp. 184–194, 2004.

[15] D. Dirschl and G. Adams, "A critical assessment of factors influencing reliability in the classification of fractures, using fractures of the tibial plafond as a model," Journal Orthop Trauma, vol. 11, no. 7, pp. 471–476, 1997.

[16] J. Martin, "Assessment of the AO/ASIF fracture classification for the distal tibia," Journal Orthopaedic Trauma, vol. 11, no. 7, pp. 477–483, 1997.

[17] W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface construction algorithm," Computer Graphics, vol. 21, no. 4, pp. 163–169, 1987.

[18] O. Ron, L. Joskowicz, C. Milgrom, and A. Simkin, "Computer-based periaxial rotation measurement for aligning fractured femur fragments from CT: a feasibility study," Computer Aided Surgery, vol. 7, no. 6, pp. 332–341, 2002.

[19] M. Harders, A. Barlit, and C. Gerber, "An optimized surgical planning environment for complex proximal humerus fractures," MICCAI Workshop on Interaction in Medical Image Analysis and Visualiszation, pp. 104–110, 2007.

[20] M. Scheuering, C. Rezk-Salama, C. Eckstein, K. Hormann, and G. Greiner, "Interactive repositioning of bone fracture segments," in Proceedings of the Vision Modeling and Visualization Conference 2001, VMV '01, pp. 499–506, 2001.

[21] M. Cimerman and A. Kristan, "Preoperative planning in pelvic and acetabular surgery: The value of advanced computerised planning modules," Injury, vol. 38, no. 4, pp. 442–449, 2007.

[22] A. Chowdhury, S. Bhandarkar, and R. Robinson, "Virtual multi-fracture craniofacial reconstruction using computer vision and graph matching," Computerized Medical Imaging and Graphics, vol. 33, no. 5, pp. 333–342, 2009.

[23] A. Willis, D. Anderson, T. Thomas, T. Brown, and J. L. Marsh, "3D reconstruction of highly fragmented bone fractures," in Proceedings of SPIE Medical Imaging, vol. 6512, pp. 1–10, 2007.

[24] B. Zhou, A. Willis, Y. Sui, D. D.Anderson, T. Thomas, and T. D. Brown, "Improving inter-fragmentary alignment for virtual 3D reconstruction of highly fragmented bone fractures," SPIE Medical Imaging, vol. 10, no. 2, pp. 341–349, 2009.

[25] T. P. Thomas, D. D.Anderson, A. Willis, P. Liu, M. Frank, J. Marsh, and T. D.Brown, "A computational/experimental platform for investigating three-dimensinal puzzle solving of comminuted articular fractures," Computer Methods in Biomechanics and Biomedical Engineering, vol. 14, no. 3, pp. 263–270, 2010.

[26] F.Meyer, "Topographic distance and watershed lines," Signal Processing, vol. 38, no. 1, pp. 113–125, 1994.

[27] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration," in Proceedings of the 7th European Conference on Computer Vision-Part III, ECCV '02, (London, UK, UK), pp. 408–422, Springer-Verlag, 2002.

[28] X. Pardo, M. Carreira, A. Mosquera, and D. Cabello, "A snake for CT images segmentation integrating region and edge information," Image and Vision Computing, vol. 19, no. 3, pp. 461–475, 2001.

[29] Y. Boykov and M.-P. Jolly, "Interactive organ segmentation using graph cuts," in Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI '00, (London, UK, UK), pp. 276–286, Springer-Verlag, 2000.

[30] G. Impoco, "Level set segmentation of knee bones using normal profile models," in Proceedings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques, (MIRAGE), (Berlin, Heidelberg), pp. 195–206, Springer-Verlag, 2009.

[31] C. Lee, S. Huh, T. A. Ketter, and M. Unser, "Unsupervised connectivity-based thresholding segmentation of midsagittal brain MR images," Computers in Biology and Medicine, vol. 28, no. 3, pp. 309–338, 1998.

[32] W. Yao, P. Abolmaesumi, M. Greenspan, and R. Ellis, "An estimation/correction algorithm for detecting bone edges," IEEE Transactions on Medical Images, vol. 24, no. 8, pp. 997–1010, 2004.

[33] R. Haralick and L. Shapiro, "Image segmentation techniques," Computer Vision, Graphics, and Image Processing, vol. 29, no. 1, pp. 100–132, 1985.

[34] P. Gibbs, D. Buckley, S. Blackband, and A. Horsman, "Tumour volume determination from MR images by morphological segmentation," Physics in Medicine and Biology, vol. 41, no. 11, pp. 2437–2446, 1996.

[35] S. Pohlman, K. Powell, N. Obuchowski, W. Chilcote, and S.Grundfest-Broniatowski, "Quantitative classification of breast tumores in digitized mammograms," Medicine and Physics, vol. 23, no. 8, pp. 1337–1345, 1996.

[36] S. Thilagamani and N. Shanthi, "A survey on image segmentation through clustering," International Journal of Research and Reviews in Information Sciences, vol. 1, no. 1, pp. 14–17, 2011.

[37] J. Bezdek, L. Hall, and L. Clarke, "Review of MR image segmentation techniques using pattern recognition," Medicine and Physics, vol. 20, no. 4, pp. 1033–1048, 1993.

[38] T. Lei and W. Sewchand, "Statistical approach to X-Ray CT imaging and its applications in image analysis part ii: A new stochastic model-based image segmentation technique for X-Ray CT image," IEEE Transactions on Medical Imaging, vol. 11, no. 1, pp. 62–69, 1992.

[39] C. Lantuejoul, Skeletonization and its application to topological measures mosaics polycrystalline. PhD thesis, Ecole des Mines, 1978.

[40] S. Beucher and C. Lantuejoul, "Use of watersheds in contour detection," in International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, Rennes, France, September 1979.

[41] J. Serra, Image Analysis and Mathematical Morphology. Orlando, FL, USA: Academic Press, 1983.

[42] J. Roerdink and Meijster, "Segmentation by watersheds: definition and parallel implementation," Advances in Computer Vision, vol. 3, no. 1, pp. 21–30, 1997.

[43] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 6, pp. 583–598, 1991.

[44] S. Beucher and F. Meyer, "The morphological approach to segmentation: the watershed transformation," Mathematical Morphology in Image Processing, vol. 12, no. 4, pp. 433–481, 1993.

[45] A. Rosenfeld and J. Pfaltz, "Distance functions on digital pictures," Pattern Recognition, vol. 1, no. 1, pp. 33–61, 1968.

[46] F. Meyer and S. Beucher, "Morphological segmentation," Visual Communication and Image Representation, vol. 11, no. 3, pp. 21–45, 1990.

[47] J. B. Roerdink and A. Meijster, "The watershed transform: definitions, algorithms and parallelization strategies," Fundamenta Informaticae, vol. 41, no. 1-2, pp. 187–228, 2000.

[48] A. Willis and B. Zhou, "Ridge walking for 3D surface segmentation," in Fifth International Symposium on 3D Data Processing Visualization and Transmission (3DPVT), (Paris, France), May 2010.

[49] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," International Journal of Computer Vision, vol. 59, no. 2, pp. 167–181, 2004.

[50] E. Agathos, I. Pratikakis, S. Perantonis, N. Sapidis, and P. Azariadis, "3D mesh segmentation methodologies for CAD applications," Computer-Aided Design And Applications, vol. 4, no. 6, pp. 827–841, 2007.

[51] A. Shamir, "Segmentation and shape extraction of 3D boundary meshes," in State-of-the-Art Report, Proceedings Eurographics 2006, pp. 137–149, 2006.

[52] A. Shamir, "A survey on mesh segmentation techniques," Computer Graphics, vol. 27, no. 6, pp. 1539–1556, 2008.

[53] P. J. Besl and R. Jain, "Segmentation through variable-order surface fitting," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 2, pp. 167–192, 1988.

[54] R. Hoffman and A. K. Jain, "Segmentation and classification of range images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, no. 5, pp. 608–620, 1987.

[55] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," IEEE Transactions on Image Processing, vol. 7, no. 3, pp. 359–369, 1998.

[56] M. Vieira and K. Shimada, "Surface mesh segmentation and smooth surface extraction through region growing," Computer Aided Geometric Design, vol. 22, no. 8, pp. 771 – 792, 2005.

[57] Y.-K. Lai, S.-M. Hu, R. R. Martin, and P. L. Rosin, "Rapid and effective segmentation of 3D models using random walks," Computer Aided Geometric Design, vol. 26, no. 6, pp. 665 – 679, 2009. Solid and Physical Modeling 2008, ACM Symposium on Solid and Physical Modeling and Applications.

[58] A. P. Mangan and R. T. Whitaker, "Partitioning 3D surface meshes using watershed segmentation," IEEE Transactions on Visualization and Computer Graphics, vol. 5, no. 4, pp. 308–321, 1999.

[59] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888–905, 1997.

[60] A. Golovinskiy and T. Funkhouser, "Randomized cuts for 3D mesh analysis," ACM Transactions on Graphics (Proc. SIGGRAPH ASIA), vol. 27, no. 5, pp. 145–151, 2008.

[61] Y. Lee, S. Lee, A. Shamir, D. Cohen-Or, and H. P. Seidel, "Mesh scissoring with minima rule and part salience," Computer Aided Geometric Design, vol. 22, no. 5, pp. 444–465, 2005.

[62] J. Zhang, C. Wu, J. Cai, J. Zheng, and X. cheng Tai, “Mesh snapping: Robust interactive mesh cutting using fast geodesic curvature flow,” Computer Graphics, vol. 29, no. 2, pp. 1467–8659, 2010.

[63] A. C. Looker, L. J. Melton, T. Harris, L. Borrud, J. Shepherd, and J. McGowan, “Age, gender, and race/ethnic differences in total body and subregional bone density,” Osteoporosis International, vol. 20, no. 7, pp. 1141–1149, 2009.

[64] D. T. Felson, Y. Zhang, M. Hannan, and J. Anderson, “Effects of weight and body mass index on bone mineral density in men and women, the framingham study,” Journal of Bone and Mineral Research, vol. 8, no. 5, pp. 567–573, 1993.

[65] X. Liu and A. Srivastava, “A spectral representation for appearance-based classification and recognition,” in Proceedings of the 16th International Conference on Pattern Recognition, vol. 1 of ICPR, (Washington, DC, USA), pp. 10037–10040, IEEE Computer Society, 2002.

[66] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 484–498, 1998.

[67] S. H. Cooray and N. E. O’Connor, “Facial features and appearance-based classification for face detection in color images,” in International Workshop on Systems, Signals and Image Processing, September 2004.

[68] C. Liu, “A bayesian discriminating features method for face detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 6, pp. 725–740, 2003.

[69] S. M. Bhandarkar, A. S. Chowdhury, Y. Tang, J. C. Yu, and E. W. Tollner, “Computer vision guided virtual craniofacial reconstruction,” Computer Medical Imaging Graph, vol. 31, no. 6, pp. 418–427, 2007.

[70] J. H. Plochocki, “Bilateral variation in limb articular surface dimensions,” American Journal of Human Biology, vol. 16, no. 3, pp. 328–333, 2004.

[71] Z. Zhang, “Iterative point matching for registration of free-form curves and surfaces,” International Journal of Computer Vision, vol. 13, no. 2, pp. 119–152, 1994.

[72] G. L. Freeman H, “Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition,” IEEE Transactions on Electronic Computers, vol. EC-13, no. 2, pp. 118–127, 1964.

[73] H. Wolfson, E. Schonberg, A. Kalvin, and Y. Lamdan, “Solving jigsaw puzzles by computer,” Annals of Operations Research, vol. 12, no. 1-4, pp. 51–64, 1988.

[74] A. Willis and D. B. Cooper, “Computational reconstruction of ancient artifacts,” IEEE Signal Processing Magazine, vol. 25, no. 4, pp. 65–83, 2008.

[75] J. McBride, “Archaeological fragment reassembly using curve matching,” Master’s thesis, Brown Univeristy, Providence, RI, September 2002.

[76] J. Stolfi and H. Leitao, “A multiscale method for the reassembly of two-dimensional fragmented objects,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, pp. 569–578, 2002.

[77] G. Papaioannou, E. Karabassi, and T. Theoharis, “Reconstruction of three-dimensinal objects through matching of their parts,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp. 114–121, 2002.

[78] Q. Huang, S. Floy, N. Gelfand, M. Hofer, and H. Pottmann, “Reassembling fractured objects by geometric matching,” ACM Transactions on Graphics, vol. 25, no. 3, pp. 569–578, 2006.

[79] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in International Conference On 3D Digital Imaging And Modeling, pp. 145–152, 2001.

[80] A. R. Willis and D. B. Cooper, “Estimating a-priori unknown 3-D axially symmetric surfaces from noisy measurement of their fragments,” in 3-D Data Processing, Visualization, and Transmission (3DPVT), pp. 334–341, 2006.

[81] A. R. Willis and D. B. Cooper, “Bayesian assembly of 3D axially symmetric shapes from fragments,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. I, no. 1, pp. 82–89, 2004.

[82] M. S. Sagiroglu, “A texture based approach to reconstruction of archaeological finds,” in The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST), pp. 137–142, 2005.

[83] M. S. Sagiroglu, “A texture based matching approach for automated assembly of puzzles,” in International Conference on Pattern Recognition, vol. 3, pp. 1036–1041, 2006.

[84] A. E. Johnson and M. Hebert, “Surface registration by matching oriented points,” in Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, (Washington, DC, USA), pp. 121–128, IEEE Computer Society, 1997.

[85] A. Johnson and M. Hebert, “Recognizing objects by matching oriented points,” Tech. Rep. CMU-RI-TR-96-04, Robotics Institute, Pittsburgh, PA, May 1996.

[86] J. E. Mebius, “Derivation of the euler-rodridues formula for three-dimensinal rotations from the general formular for four-dimensional rotations,” General Mathematics, vol. 1, pp. 759–766, 2007.

[87] K. L. Low, “Linear least-squares optimization for point-to-plane ICP surface registration,” Tech. Rep. TR04-004, University of North Carolina at Chapel Hill, Raleigh, NC, May 2004.

[88] T. P. Thomas, Virtual Pre-Operative Reconstruction Planning for Comminuted Articular Fractures. PhD thesis, The University of Iowa, 2010.