A prototypical system to virtually reconstruct high energy fracture events

Waseem G. Shadid* and Andrew Willis

Electrical and Computer Engineering Department, University of North Carolina – Charlotte, Charlotte, NC, USA

(Received 30 June 2014; accepted 22 June 2015)

This paper presents a new computational system that seeks to estimate the dynamics of fracture event from a 3D CT image of the fractured limb and the (assumed) intact contra-lateral limb. A model of the intact limb anatomy prior to injury is estimated from the image of the intact contra-lateral limb, and a model of the fractured limb anatomy after injury is estimated from the image of the fractured limb. Fracture reconstruction software aids in estimating an intact limb model which characterizes the limb in terms of its soft tissues and fractured bone fragments at the time prior to the fracture event. The system then virtually simulates fracture events by impacting the reconstructed limb model using a hypothesized impacting object, i.e. the strike object, and applying a physics-engine to simulate the dynamics of the fracture event. A maximum likelihood estimation framework is applied to score fracture events based on how well they re-create the fracture pattern observed in the fractured limb image. Initial results are shown for one fracture case which demonstrate that solutions to this heretofore unexplored problem are possible using this framework.

Keywords: reconstruction; fracture; bone; 3D CT

1. Introduction

Treatment and long-term care associated with symptoms that arise due to traumatic bone fractures have a significant impact on our health care system (Zhou et al. 2009). Accurate diagnosis of fracture severity and treatment of the injury are key factors in determining the long-term prognosis for traumatic bone fracture patients. Recent work has focused on developing systems that seek to assist in accurately characterizing fractures (Willis et al. 2007). This includes diagnostic systems to assist in assessment of the fracture severity (Liu 2012) and systems that virtually reconstruct complex bone fractures (Chowdhury et al. 2009). These systems focus on documentation and analysis of the post-traumatic condition of the fractured limb. In contrast, the proposed system seeks to estimate the fracture event itself, i.e. the state of the limb from immediately prior to impact to the state of the limb after the injury was sustained. Estimation of fracture events is a challenging task which involves significant challenges in (1) image processing, (2) 3D surface modeling, (3) physics-based simulation, and (4) searching high-dimensional parameter spaces. This article describes a prototypical system for the purpose of understanding complex fracture events. The system takes as input CT images and user interaction and provides as output plausible video reconstructions of how the fracture event may have occurred. The presence of computational systems that enable users to visualize the fracture event simulations can contribute important insights for these difficult cases. For surgical treatment, knowledge of fragment trajectories indicates likely regions of soft tissue trauma. This knowledge may provide trajectories for percutaneous manipulation of bone fragments which encourage the fragment to ‘move back along its trajectory’ to reduce soft tissue trauma during surgical reduction. For forensic applications, users can hypothesize specific objects and points-of-impact to attempt to reconstruct the injurious event after-the-fact which is often of interest as investigative information. Currently, there are no existing approaches that seek to provide this level of understanding of a bone fracture despite their potential helpfulness to orthopaedic surgeons. However, recent research in Willis et al. (2007), Zhou et al. (2009), and Liu (2012) provide key technologies for semi-automatic virtual fracture reconstruction that now enable virtual simulation of the fracture event possible.

The system estimates the geometry and dynamics of a fracture event, i.e. the process of breaking a bone into fragments and moving these fragments away from their anatomic positions, using 3 data-sets as input: (1) a 3D CT image of a fractured limb, referred to as a fracture image, (2) a 3D CT image of an unfractured limb, referred to as an intact image, and (3) a collection of hypothesized objects that could have broken the bone. The outputs of the system are: (1) an estimate of the hypothesized object that broke the bone, referred to as a strike object, and (2) a virtual simulation of how bone fragments moved from their original anatomic positions to their fractured positions as presented by the 3D CT image of the fractured limb, referred to as a fracture simulation. Solutions to this problem require estimating of the state of the limb prior to...
the fracture event and subsequently discovering plausible values for the geometry and material composition of the strike object that explain the data of the fracture image through virtual fracture simulations. Toward this end, the system constructs a virtual model for the limb and the strike object and then computes fracture simulations that virtually collides these objects. User interaction aids the system by providing initial guesses for the unknown fracture event variables. Advantages of this proposed approach come at the added cost of scanning the contra-lateral limb which increases treatment cost and radiation exposure for the patient. The key contribution of this paper is the prototypical system for estimating the fracture event from CT images.

2. Related work
To our knowledge, this is the first system proposed for virtually reconstructing a fracture event. Yet, this system uses two key enabling technologies to complete its task: (1) image segmentation algorithms to identify bone fragments and (2) virtual fracture reduction software to virtually reconstruct the unbroken bone by piecing together extracted bone fragments. Work in Cimerman and Kristan (2007), Chowdhury et al. (2009), Willis et al. (2007), and Liu (2012) propose methods to virtually reduce fractures. Cimerman and Kristan proposed in their work (Cimerman & Kristan 2007) a system for surgical planning that has surgeons interact in a virtual 3D environment to reconstruct acetabular fractures of the pelvis. This system focuses on allowing surgeons to virtually manipulate the bone fragments without any computer-aided guidance for the reduction. The system proposed in Chowdhury et al. (2009) has a goal similar to Cimerman and Kristan (2007), but targets reconstruction of craniofacial fractures with examples that include a multi-fragment fracture of the mandible. The problematic interactive aspects of prior systems are alleviated by computer-assisted surface alignment. Here, broken fragment surfaces are aligned using the Iterative Closest Point (ICP) algorithm (Birkfellner 2014) and a graph-based optimization method (Pulli 1999) is used to align multiple fragments simultaneously.

The proposed system builds on the fracture reconstruction approach initially proposed in Willis et al. (2007) and Zhou et al. (2009) which evolved into the virtual fracture reduction system described in Liu (2012). The approach developed here differs from prior research by using a bone template to drive reconstruction. Work in Cimerman and Kristan (2007), Chowdhury et al. (2009), and Willis et al. (2007) all seek to match broken fragment surfaces to reconstruct the fracture. Unfortunately, these surfaces often consist almost entirely of cancellous bone tissue which is often difficult to differentiate from background soft tissues or fluids which can cause significant errors when the fragments are segmented. These errors can pose significant problems for methods that attempt to match broken surfaces to reconstruct fractures such as inaccurate alignments or incorrect fragment matches. Liu (2012) proposes using a CT of the contra-lateral limb as a template into which the fragment surfaces can be matched. Here, the outer surfaces are used to match the fragments to the intact outer surface of the template bone. Since fragment outer surfaces consist of cortical tissue which is easier to differentiate in CT, surfaces are extracted with less error and a more accurate and robust reconstruction result is possible.

3. Methodology
The process that our system uses to estimate a fracture event consists of two parts: (1) estimate the ‘initial’ state of fractured limb, i.e. the state of the bone at the time the bone fracture fragments were generated, and (2) search for values of the fracture event variables that explain the 3D fracture CT data. The proposed problem presents numerous geometric and physical model challenges which make it both complex, i.e. it depends on a large number of dependent variables, and ill-posed, i.e. it may have many plausible solutions. The system copes with these difficulties by simplifying both the geometric and physical model and by providing a collection of plausible solutions for the user rather than a single solution. We expect users to integrate their domain knowledge (medical or forensic) and other external information to interpret these results to facilitate their analysis.

The system consists of three separate components (Figure 1).

- (1) Estimate the intact limb, i.e. the state of the limb at the beginning of a fracture event,
- (2) Search for the values of the fracture event variables that agree with the data in the fractured limb 3D CT image,
- (3) Display plausible fracture event simulations to the user in a 3D virtual environment for analysis.

To our knowledge, the proposed approach represents the first system to attempt to recreate the fracture event. It includes a novel framework which makes computation on this difficult problem tractable by using innovative technologies. It also provides a rationale for addressing ill-posed inverse problems by allowing users to navigate a collection of plausible solutions rather than displaying a single ‘best’ solution.

3.1. Estimating the intact limb model
The system uses a model of the intact limb to simulate fracture events. While a complete limb model would
include models for numerous body substructures, e.g. bone, muscle, skin layers, connective tissue, tendons, etc., our prototypical system models categorize limb tissue into two basic types: (1) bone tissue and (2) soft tissues (fat and muscle). Since the limb has already been injured, the models for both the intact bone tissue and the intact soft tissue for the intact limb must be estimated.

An estimate of the intact bone tissue for the intact limb is obtained by applying the bone fracture reconstruction system (Liu 2012). This system takes as input the fracture image and an intact image. Typically, the intact image is generated by imaging the undamaged contra-lateral limb of the patient and subsequently transforming the image by negating or 'flipping' the image across the anatomic plane of symmetry. As part of the reconstruction process, the reconstruction system extracts the bone fracture fragments from the fracture image via medical image segmentation (Shadid & Willis 2013). Let \( B_i \) denote the collection of \((x, y, z)\) coordinates identified as members of the \(i\)th bone fragment and let \( B = \bigcup_i B_i \) denote the collection of all bone fragment 3D coordinates in the fracture image.

A segmentation algorithm is also applied to the intact image to extract the intact bone surface. The intact image is then aligned (registered) to the fracture image by aligning the intact bone surface with the surface of the remaining/undisturbed portion of bone tissue from the fracture image, i.e. the bone fragment in the fracture image which remains in correct anatomic position after the fracture has occurred, also referred to as the ‘base fragment’ in Liu (2012) (Figure 2).

The generation reconstruction then proceeds by aligning the surfaces of the bone fragments to the surface of the intact bone. In this way, the intact bone model acts as a template into which the bone fragments are fit to reconstruct a model of the fractured bone from its fragments. Geometric alignments are performed using the ICP algorithm (Birkfellner 2014). As a result, the reconstruction system provides a Euclidean transformation for each bone fragment that repositions the bone fragment from its as-found position in the fracture image to its estimated anatomic position within the fracture image. Let \( T_i \) denote the Euclidean transformation that transforms the points of bone fragment \( B_i \) to their estimated anatomic location. For simplicity, we collect the transformation parameters for a fracture involving \( N \) fragments into a single parameter \( T \) which incorporates 6 variables for each fracture fragment, i.e. it includes \( 6N \) variables. Using this notation we can denote the state of the bone tissue for the intact limb as \( B(t_0) = T(t_0)B \) where \( t_0 \) denotes the initial, i.e. beginning time of the fracture event. We also use this notation for fracture event simulation. Specifically, we denote the position and orientation of the bone fragment data at a generic time, \( t \) as simply \( B(t) \).

An estimate of the soft tissue of the intact limb model is generated by segmenting soft tissues from the registered intact image. While many soft tissues exist, the prototype system characterizes regions of soft tissue generically as either fat or muscle. These tissues are difficult to accurately differentiate in CT images and our approach here seeks to coarsely approximate the

Figure 1. The fracture event analysis system consists of three operations: (1) generate an estimate the intact limb from CT images and user input, (2) perform virtual simulations to find plausible values for fracture event variables, (3) display to the user plausible simulations ranked by their likelihood score for visualization and analysis.
The soft tissue and air. For this reason, the segmentation approach classifies the pixels of the image into three categories: (1) fat pixels, (2) muscle pixels, and (3) non-soft tissue pixels. These regions are estimated using a global threshold where a user specifies three threshold values: (1) a pixel is categorized as fat if its intensity is greater than \( T_{\text{fat}} \), (2) muscle if its intensity is less than \( T_{\text{muscle}} \), and (3) other tissues if its intensity is greater than \( T_{\text{fat}} \) and less than \( T_{\text{muscle}} \). The used threshold values are: \( T_{\text{fat}} = -400 \text{HU}, \ T_{\text{muscle}} = -100 \text{HU}, \ T_{\text{other}} = 100 \text{HU} \) (Terrier et al. 2000). Pixels are then classified by applying these thresholds as follows: (1) a pixel is categorized as fat if its intensity is greater than \( T_{\text{fat}} \) and less than \( T_{\text{muscle}} \), and (2) a pixel is categorized as muscle if its intensity is greater than \( T_{\text{fat}} \) and less than \( T_{\text{other}} \). All other pixels are considered not to be soft tissue. The resulting pixel-wise categorization of the soft tissues are then simplified by applying an octree scheme and averaging categorizations over larger volumes of the image where the larger volumes are labeled according to the majority vote of the categorized pixels that they contain. The final result is a coarse covering of the soft tissue regions within the registered intact image with cubes where each cube includes a categorization of the tissue it contains as either fat (yellow) or muscle (red) (Figure 3). A 3D model of these tissue elements is generated by placing rigid cylinders centered at each cube center (Figure 3). Let \( S_j \) denote the \( j \)th element within the 3D model for the soft tissue and \( S \) denote the collection of all 3D soft tissue model elements. Under simulation, the soft tissue elements will rotate and translate over time due to impact and fragment dispersion associated with the fracture event. Let \( T_j(t) \) denote the transformation for \( S_j \) at time \( t \). Further, for simplicity let \( S(t) = T(t)S \) denote the generic position of all the soft tissue data at time \( t \) where \( T \) denotes the Euclidean transformations for all of the geometric elements (cylinders) of the intact limb soft tissue model.

The end result of the image processing functions and reconstruction system described in this section are geometric estimates for the intact/unbroken bone in terms of its fragments, \( B(t_0) \) and the soft tissue surrounding the intact bone, \( S(t_0) \) which is further subdivided into elements of fat or muscle. These two elements combine to form the estimate of the geometry intact limb needed to simulate the fracture event. The proposed segmentation approach correctly segments much of the cortical and trabecular bone tissue. Yet, some trabecular tissue voxels will assume intensity values that lie below the \( T_{\text{other}} \) threshold, in which case these voxels are not correctly classified. Fortunately, these trabecular tissue regions are associated with low-density cancellous tissue which tears or compacts with relative ease when compared to the more dense cortical bone tissue. As such, for the high-energy fractures of interest, we propose that errors associated with these structurally weak tissues are unlikely to significantly impact the fracture simulation result.

### 3.2. Simulating fracture events

Fracture event simulations seek to approximate the physical process that occurred when the injury was sustained. The simulation component of the system accomplishes this task in three steps: (1) the estimate geometry of the intact limb model is assigned physically meaningful attributes, (2) a strike object is generated whose purpose is to deliver the fracture impact, and (3) a search algorithm searches for values of the fracture event that are plausible. The output of this process is a collection of fracture event simulations where each simulation is ranked by a likelihood score. Each simulation score indicates the likelihood of the fracture image data given the assumed value for the fracture event variables.

The system software is almost entirely specified in Java code (v1.6) as an extension to the source of the fracture reconstruction system from Liu (2012) with the exception of the fracture simulations. Fracture simulations use the Python-based scripting interface to the Blender 3D software package (v2.67) and the Bullet physics engine (v2.82) (Mullen & Coumans 2008). Python code places the 3D bone fragments and strike
object in the virtual space, specifies the surrounding soft
tissue model geometry and connectivity constraints, and
attributes these geometries with the physical simulation
parameters specified in Table 1. Finally, initial values for
the fracture variables \( \Theta \) are specified and the Bullet phys-
ics engine simulates the fracture event. Motion of the
fragments is reported as a sequence of keyframes which
provide a snapshot of the 3D fracture geometry as a col-
clection of 3D transformations that encode the position
and orientation of soft tissue and bone fragment elements
over time.

Estimation of the intact limb model provides a geo-
metric description of the limb at the time immediately
prior to the impact. However, to simulate the fracture
event, these geometric models have to be attributed with
physically meaningful attributes. Physical attributes for
bone tissue are a density, \( \rho = 1.85 \text{ g/cm}^3 \) and a coef-
ficient of friction, \( \gamma = 0.6 \). Constraints are implemented
that specify bone tissue as rigid, i.e. non-deformable,
objects. Physical attributes for soft tissue also include a
density and a coefficient of friction for fat elements
\( (\rho = 0.9 \text{ g/cm}^3, \ \gamma = 0.3) \) and muscle elements
\( (\rho = 1.06 \text{ g/cm}^3, \ \gamma = 0.8) \). These quantities determine the
mass of the 3D objects in the intact limb model and
specify one of the ways that energy is dissipated as these
objects contact with each other due to motion.

The cylindrical elements of the soft tissue model are
also connected by a lattice of ‘breakable constraints.’
Breakable constraints within the bullet engine specify
elastic/spring-mass connections between the soft tissue
elements that constrain their joint movement (Figure 4).
These constraints can ‘break’ when the force exerted on
the connection exceeds the connection bond strength.
For our soft tissue model, the breaking condition serves
to approximate how soft tissue will tear apart when
placed under severe forces. This model is constructed
based on the Mass-Spring Method (MSM) described in
Halic et al. (2009) to represent soft tissues. MSM allows
simulating the tissue physics and behaviors at real time.
In this model, the soft tissue is thought of as discrete
point masses that are connected to each other with linear
springs. So that, employing force to a particle mass

---

Table 1. Shows the value of the parameters for the friction factors and breakable constraint parameters.

| Parameter        | Fat friction | Muscle friction | Bone friction | Damping factor | Linear limit | Angular limit | Stiffness | Breaking threshold |
|------------------|--------------|-----------------|---------------|----------------|--------------|--------------|-----------|--------------------|
| Value            | 0.3          | 0.8             | 0.6           | 0.5            | 5 mm         | 5°           | 0.2       | 1                  |

Notes: The friction value specifies how much velocity is lost when an object collides with another object. The values for breakable constraints specify how the linear springs connecting the virtual soft tissue objects react when a force is applied. The constraints serve to approximate how soft tissue will tear apart when placed under severe forces.
within the spring mesh will move the particular node and propagate the forces across its neighbors. Five properties are needed to specify the breakable non-rigid constraints (Figure 4): (1) damping, (2) linear limit, (3) angular limit, (4) stiffness, and (5) breaking threshold. The first two attributes, damping and stiffness, are typical attributes needed when specifying a second-order damped oscillatory mechanical system. Here, the damping attribute specifies the rate that energy in an oscillatory response will dissipate due to losses in the spring/mass system in terms of percent decrease in peak-magnitude per second. The stiffness attribute specifies the spring constant of the system as represented by Hooke’s Law and is specified in terms of how much force is needed to compress or elongate the spring in N/m. The remaining three attributes are used to specify contexts which will ‘break’ connections in the spring-mass network. The linear limit attribute specifies the maximum the amount of elongation allowed for the spring from its initial rest length before breaking. The angular limit attribute specifies the maximum amount of angular deflection allowed for the spring away from its initially defined coordinate axis before breaking. The stiffness attribute specifies how flexible a connection is in terms of how much force is needed to compress or elongate the spring in N/m. The breaking threshold attribute specifies the instantaneous force required to break a connection in N. These properties collectively determine the deformation, energy dissipation, and tearing characteristics of the soft tissue model. These properties are specified by a user through the system interface to reflect the inner stretching, bending, tension characteristics of soft tissues, because, according to Bhasin and Liu (2006), it is hard to find the corresponding values for these properties for a particular tissue.

A strike object must be created as the object which delivers the traumatic impact to the intact limb model. The strike object geometry and physical attributes are typically unavailable for measurement at the point of treatment. The prototypical system requires the user to specify a geometry, position, orientation, and physical attributes for the strike object. This information provides an initial point within the fracture event space, i.e. this provides sufficient data to perform a fracture simulation. However, the simulation result using these initial values represents only one plausible guess for the fracture event. We consider this initial point as a coarsely specified approximation of the system state immediately before the injury. Some errors present in this initial specification originate from inaccuracies in specifying the pose, velocity, shape, and material of the strike object. We also consider this initial point as one of possibly multiple explanations for the observed fracture data. For this reason, a search procedure is prescribed that varies the user-specified variables of the fracture event to find values for these variables that are supported by the recorded fracture image data. For the prototypical system described here, the fracture event variables are limited to the variables that specify the geometry, position, orientation, direction of travel, and velocity of the strike object at the initial time of the fracture event; denoted Θ. For the results shown, the geometry of the strike object is restricted to be a sphere which eliminates geometry and orientation variables from the vector of unknowns. The orientation variables are eliminated because the fracture event results are invariant with respect to the sphere
orientation at impact. Hence, the resulting vector of unknown fracture event variables is \( \Theta = [v, m, s, p, d]^T \) where \( v \) denotes the strike object velocity, \( m \) denotes the strike object mass, \( s \) denotes the scale factor for the strike object size, \( p \) denotes the initial 3D position of the strike object to hit and \( d \) denotes the direction of travel of the strike object as a \((\phi, \theta)\) unit vector in spherical coordinates. Under these circumstances, \( \Theta \) is a vector of eight variables.

The search procedure attempts to efficiently search the space of all plausible values for the fracture event variables to find those which generate fracture patterns similar to that measured in the fracture image. The search is facilitated by providing an external scoring function which rates the quality of a chosen set of fracture event values. The system applies a maximum likelihood estimation (MLE) framework for scoring specific collections of fracture event variable values. The conceptual goal of the MLE framework is simple, for each guess of the fracture event variables \( \Theta = \Theta_i \), a likelihood value will be computed which reflects the likelihood that the fracture image data given the chosen, \( p(D|\Theta = \Theta_i) \). Values of \( \Theta \) with higher likelihoods are values of the fracture event variables that are better supported by the fracture image data, i.e. they are ‘more likely.’ The likelihood distribution chosen is a Gaussian distribution which expresses the probability of all the data as the sum of the differences between the simulated bone tissue intensities at the final time, \( B(t_f) \) and the bone tissue intensities identified by segmenting the fracture image, \( B \) as shown in Equation (1).

\[
p(D|\Theta) = p(B|\Theta, B(t_f)) = k \times \exp\left(-\frac{1}{2} \|B - B(t_f)\|^2\right)
\]

Since \( B(t_f) = TB \), we can see that Equation (1) seeks values of the fracture event variables that cause the simulated bone fragments final position to coincide with their segmented position in the fracture image. In this case, \( B(t_f) = B \) and the likelihood function is a maximum. The search procedure guesses at values for \( \Theta \) as inputs for simulated fracture events. When the simulation ends, final transformation values for the bone fragment data are provided, denoted \( T_f \). Each of these values for \( T_f \) let us evaluate \( p(D|\Theta = \Theta_i) \) and generate a likelihood score for the guessed \( \Theta \) value. The search procedure assigns each value of \( \Theta_i \) the likelihood score \( p(B|\Theta, B(t_f)) \) and stores the result in a list sorted in order of decreasing likelihood.

The current search procedure takes the user-provided value of the fracture event variables as a point within an eight-dimensional solution space. Each point in this space provides a value for \( \Theta \) and, after simulation, a likelihood score, \( p(B|\Theta, B(t_f)) \). A heterogeneous mixture of multi-scale search and gradient descent is used to find values of \( \Theta \) having high likelihood scores. The multi-scale aspect of the algorithm defines an eight-dimensional volume by bracketing each of the eight components of the fracture variables to the subset of values which are physically meaningful. For example, the direction vector must be constrained to point in the direction of the fracture site and the velocity has to be above a minimum speed and below some maximum speed. Within this eight-dimensional volume samples are taken at locations along a rectangular grid. After sampling at grid locations, gradient descent minimizations are initialized at the most likely, i.e. highest scoring eight locations.

### 3.3. Visualization and analysis of fracture simulations

The search for plausible fracture event variable values generates a list of fracture event variable values which are listed in order of their likelihood score. A visualization system allows users to browse through the computed simulations to explore the space of plausible solutions. Fracture simulation results may be viewed by animating bone fragment surfaces within a virtual 3D environment and analysis tools provide views of the simulated 3D models in situ with the intact and fracture CT images. This allows users to visually approximate the trajectory of the bone fragments through the soft tissue which also provides clues about the location and extent of soft tissue trauma. Visualization of the simulation result for bone fragments shows how these fragments moved over time during the fracture event. The analysis tool for soft tissue damage highlights image areas that correspond to soft tissue where bone fragments passed through during the fracture event. These areas are assumed to correspond to damaged tissue parts in the fractured limb.

To summarize, a flowchart of the algorithm is shown in Figure 5. The algorithm uses the input fracture image, intact image, and the user input to estimate the intact limb, i.e. reconstructed bone fragments, soft tissue, and strike object. Then, the algorithm initializes the search variables, i.e. strike object variables for this prototypical system, and runs a fracture simulation. The algorithm runs simulations by interacting with Bullet engine through Blender using Python script. The error is computed for the simulation result, if the error is high the search variables are updated then the simulation is repeated. If the error is low, the search process stops and the result is displayed to the user.

### 4. Results

The system was used to estimate the fracture event of a clinical tibial plafond case. For this case, the fracture image, the intact image, the bone fracture fragment
surfaces, the intact bone surface, and the reconstructed fracture fragment positions were provided by the work in Liu (2012) which virtually reconstructs highly comminuted tibia fragments from 3D CT images of a fracture case. The fracture case includes six bone fragments and also includes a model for the talus bone.

Figure 5. Shows a flowchart of the algorithm to estimate plausible solutions of the fracture event.

Notes: The algorithm uses the input images to estimate the intact limb. The algorithm runs a search procedure to find plausible solutions for the fracture event variables. Then, the result is displayed to the user.
All experiments were conducted with the following collection of values for different parameters and settings of the system on a PC with 8 GB RAM and Intel Core i5 CPU 750 @ 2.67 GHz × 4. Each experiment takes an average of 8 h to generate results. The settings that are used to generate soft tissue attributed models are: fat density = 0.9 g/cm³ (Farvid et al. 2005), fat friction = 0.3, muscle density = 1.06 g/cm³ (Urbanchek et al. 2001), and muscle friction = 0.8. The breakable constraints settings are: damping factor = 0.5, linear limit = 5 mm, angular limit = 5°, stiffness = 0.2, and breaking threshold = 1. The settings for physical properties of fracture fragment attributed models are: bone density = 1.85 g/cm³ (Yang et al. 2002) and friction = 0.6. The geometry of the strike object was set to be a sphere. Fracture simulations were set to run to a final time of 1.5 s and the maximum number of search iterations allowed to find the best fracture simulation is set to 100. The system solved for likely values of the unknown fracture event variables. These variables control the impact of the strike object and determine its: speed \( v \), mass \( m \), scale \( s \), 3D position \( p \) and direction \( d \). At the end of each experiment, the list of most likely \( \Theta \) values is provided with their error score.

The error score is the difference between the measured positions for the fracture fragments observed in the fracture event \( B \) and the estimated positions of the fracture fragments in a simulation at the final time \( B(t_f|\Theta) = T_i(t_f|\Theta)B \). The error score, denoted as \( \| B - B(t_f|\Theta) \| \), is computed as in Equation (2). \[
\| B - B(t_f|\Theta) \| = \frac{1}{K} \sum_{i=1}^{K} \| B_i - B_i(t_f|\Theta) \| \quad (2)
\]

where \( K \) is the number of bone fragments. Let \( B_i = \{ x_0, x_1, x_2, ..., x_M \} \) be the measured data for the \( i \)th bone fragment and \( x_m \) be the \((x, y, z)\) coordinate in the image for the \( m \)th point in bone fragment \( B_i \). Then, \( \| B_i - B_i(t_f|\Theta) \| \) is computed as the average amount of translation that is needed to displace the transformed points in \( B_i(t_f|\Theta) \) to their measured positions as defined in Equation (3). \[
\| B_i - B_i(t_f|\Theta) \| = \frac{1}{M} \sum_{m=0}^{M} \| x_m - T_i(t_f|\Theta)x_m \| \quad (3)
\]

where \( T_i(t_f|\Theta) \) is the data transformation computed for \( i \)th bone fragment, \( B_i \), by the fracture simulation at the final time for the fracture event. \( \| \cdot \| \) is the norm of a vector. \( T_i(t_f|\Theta)x_m \) is equal to \( x_m \) only and only if the data transformation \( T_i(t_f|\Theta) \) is the identity matrix. So, the desired fracture simulation is the one that is able to move bone fragment pixels from their estimated original anatomic positions at \( t_0 \), i.e. \( T_i(t_0|\Theta)x_m \), and to put them back to their measured positions. Such a simulation has the highest likelihood \( \Theta \).

The experimental result for estimated fracture simulations for a fracture event that are generated using a spherical strike object is provided. The list of likely \( \Theta \) values is shown in Table 2. The values are sorted in an ascending order according to the error value. A snapshot of the fracture simulation from sagittal view using the initial conditions with the lowest error is shown in Figure 6. The contours for fracture fragments are shown in Figure 7. This figure shows contours of the fractured fragments measured from the CT image, reduced fragments, and fracture fragments at the end of the simulation for the first value of \( \Theta \). The contours are shown at two different slices of the CT image: (1) slice 80 that shows the contours for three fragments and (2) slice 108 that shows the contours for all six fragments. This figure is provided to compare visually the positions and orientations of the fragments computed via the simulation with their measured ones within the fracture CT image. A 3D representation of the estimated void region, i.e. the space that is created by the pushed soft tissue due to the movement of the bone fragments during a fracture event, is shown in Figure 8. The surface of the void region is shown from three different view perspectives: axial, coronal, and sagittal. The surface is drawn with respect to other bone fragment surfaces in their estimated original anatomic positions in order to visualize the shape and location of the void region in the fractured ankle.

Another experiment was conducted with different values for the breakable constraint and friction settings of the system. The settings that are used for friction are: fat friction = 0.1, muscle friction = 0.2, and bone friction = 0.9. The breakable constraints settings are: damping factor = 0.2, linear limit = 10 mm, angular limit = 10 degrees, stiffness = 0.1, and breaking threshold = 0.1.

| Table 2 | Shows three values of \( \Theta \) that are minima in the error surface. |
|--------|---------------------------------|
| \( \Theta \) | \( m \) (g) | \( s \) | \( v \) (mm/s) | \( d \) (\( \phi \), \( \theta \)) rad | \( p \) (x, y, z) | Error (mm) |
| 1 | 158 | 1.5 | 111 | (-174, 2.18) | (32, 41, 84) | 60 |
| 2 | 152 | .98 | 107 | (0.0, 2.06) | (35, 44, 88) | 62 |
| 3 | 199 | 1.4 | 72 | (-611, 2.0) | (34, 36, 87) | 67 |

Notes: The values are sorted in an ascending order according to the error value. These values represent plausible solutions of how bone fragments moved from their original positions to their fractured ones over time.
The list of likely $H$ values is shown in Table 3. The values are sorted in an ascending order according to the error value. The error in this experiment is higher than the error in the first experiment that is because the linear and angular limits are higher and the breaking threshold is lower which allow the bone fragments to break through soft tissue easier and move far away from their fracture positions.

The result shows a plausible solution of how the fracture fragments are displaced from their anatomic positions to their fractured ones. The results of fracture simulations for this prototypical system indicate that it is

Figure 6. Shows sagittal view snapshots for the virtual fracture simulation for a human clinical tibial plafond case using the initial conditions with the lowest error at four different keyframes. Notes: This fracture case includes six bone fragments and also includes a model for the talus bone, each fragment is shown in a unique color. The time interval for the fracture simulation is 1.5 s. Figures (a–d) show snapshots of the fracture simulation at 0, 0.5, 1.0, and 1.5 s, respectively.

Figure 7. Shows contours of the six fractured fragments measured from the CT image, reduced fragments, and fracture fragments at the end of the simulation for the first value of $\Theta$. Notes: The contours are shown at two different slices of the CT image: slice 80 (top row) and slice 108 (bottom row). Slice 80 shows contours for three of the six fragments, while slice 108 shows the contours for all six fragments.
possible to construct a virtual dynamics model for the fractured limb, to virtually hit that model with a strike object, and to generate a fracture simulation for a fracture event.

5. Conclusion

Estimation of fracture events is a challenging task which involves significant challenges in (1) image processing, (2) 3D surface modeling, (3) physics-based simulation, and (4) searching high-dimensional parameter spaces. The presented prototypical system represents initial work on this heretofore unsolved and difficult inverse problem. The system utilizes a unique blend of advanced image processing technologies, new fracture reconstruction tools, and commercial virtual physics modeling to approach this otherwise intractable problem. The approach shows promise to successfully provide new insights on fracture analysis and the reduction in complexity afforded by the simplifications allow solutions to be computed within the span of several hours. The system also incorporates methods to cope with the ill-posed nature of the estimation problem by providing scores for estimated fracture events and an interface to navigate through different simulations.

While existing systems focus on documentation and analysis of the post-traumatic condition of the fractured limb, the proposed system seeks to estimate the dynamics of the sustained injury. These dynamics show promise to heighten clinical understanding of the fracture and may allow surgeons to infer regions of soft tissue trauma which are otherwise difficult to perceive in imagery. Other applications include forensic investigation where users may want to recover information about the object and point-of-impact that generated the fracture event.

The proposed approach uses a CT of the injured limb and a presumed intact limb which is undesirable in a clinical circumstance due to the increased cost and radiation exposure associated with scanning the intact limb. Further research explores replacing the model derived from the intact limb scan with a 'nominal' 3D bone model template which can be morphologically fit to the case to achieve the virtual fracture reduction.

The proposed prototype makes a number of simplifications to produce results. Simplifications include omission of the surface fracture-mechanics, the restriction that bone fragments are rigid, exclusion of connective tissue sub-structures such as ligaments and tendons, and an oversimplification of the role that soft-tissue plays in a typical fracture event. These simplifications are likely to be problematic when the system is used to estimate highly complex fractures involving many fragments, large-scale tissue deformations, and large-scale fragment motions. In addition, the system does not use a perfect model of the soft tissue. While similar models

Table 3. Shows two values of $\hat{\Theta}$ that are minima in the error surface.

| $\hat{\Theta}$ | $m$ (g) | $s$ | $v$ (mm/s) | $d$ ($\phi, \theta$) rad | $p$ ($x, y, z$) | Error (mm) |
|----------------|---------|-----|------------|-----------------|----------------|------------|
| 1              | 100     | 1.6 | 50         | (0.0, 3.12)     | (37, 39, 89)   | 118        |
| 2              | 133     | 1.0 | 45         | (0.71, 1.6)     | (37, 53, 34)   | 140        |

Notes: The values are sorted in an ascending order according to the error value. These values represent plausible solutions of how bone fragments moved from their original positions to their fractured ones over time.

Figure 8. Shows a 3D representation of the estimated void region in soft tissue.

Notes: Figures (a–c) show the void region surface in solid light blue from three different view perspectives: axial, coronal, and sagittal, respectively. The void region surface is shown with respect to other bone fragment surfaces (in faded colors) in their estimated original anatomic positions in order to visualize its shape and location in the ankle.
for soft tissue deformation have been used by other researchers (Halic 2009), there remains uncertainty regarding the accuracy of this model for large tissue deformations and tearing as is often the case for high-energy fractures. However, despite these shortcomings, the results for our prototypical system demonstrate that computational analysis of these high-dimensional systems are now becoming feasible due to recent increases computing power and recently developed technologies for virtual fracture analysis.

Disclosure statement
No potential conflict of interest was reported by the author(s).

References
Bhasin Y, Liu A. 2006. Bounds for damping that guarantee stability in mass-spring systems. In: Westwood, JD, Haluck, RS, Hoffman, HM, editors. Medicine Meets Virtual Reality 14. Amsterdam: IOS Press; p. 1–5.
Birkfellner W. 2014. Applied medical image processing, second edition: a basic course. Boca Raton (FL): Taylor & Francis.
Chowdhury A, Bhandarkar S, Robinson RW, Yu JC. 2009. Virtual multi-fracture cranio-facial reconstruction using computer vision and graph matching. Comput Med Imaging Graph.
Cimerman M, Kristan A. 2007. Preoperative planning in pelvic and acetabular surgery: the value of advanced computerised planning modules. Injury. 38:442–449.
Farvid MS, Ng TWK, Chan DC, Barrett PHR, Watts GF. 2005. Association of adiponectin and resistin with adipose tissue compartments, insulin resistance and dyslipidaemia. Diabetes Obes Metab. 7:406–413.
Halic T, Kockara S, Bayrak C, Rowe R, Chen B. 2009. Soft tissue deformation and optimized data structures for mass spring methods. Proceedings of the 2009 ninth IEEE International Conference on Bioinformatics and Bioengineering; Washington, DC: IEEE Computer Society; p. 45–52.
Liu P. 2012. A system for computational analysis and reconstruction of 3D comminuted bone fractures [PhD thesis]. The University of North Carolina at Charlotte.
Mullen T, Coumans E. 2008. Bounce, tumble, and splash! simulating the physical world with blender 3D. Serious skills. Indianapolis (IN): John Wiley & Sons.
Pulli K. 1999. Multiview registration for large data sets. In: Proceedings of the 2nd international conference on 3-D digital imaging and modeling, October 04–08, 1999, Ottawa, Canada; p. 160–168.
Shadid W, Willis A. 2013. Bone fragment segmentation from 3D CT imagery using the probabilistic watershed transform. In: Southeastcon, 2013 Proceedings of IEEE; Jacksonville, FL: IEEE. p. 1–8.
Terrier F, Grossholz M, Becker C. 2000. Spiral CT of the abdomen. Medical radiology: diagnostic imaging and radiation oncology. Berlin: Springer.
Urbanchek MG, Picken EB, Kallianiemi LK, Kuzon WM. 2001. Specific force deficit in skeletal muscles of old rats is partially explained by the existence of denervated muscle fibers. J Gerontol Ser A Biol Sci Med Sci. 56:B191–B197.
Willis A, Anderson D, Thomas T, Brown T, Marsh JL. 2007. 3D reconstruction of highly fragmented bone fractures. Med Imaging. 6512:65121P1–65121P10.
Yang J, Chiou R, Ruprecht A, Vicario J, MacPhail LA, Rams TE. 2002. A new device for measuring density of jaw bones. Dentomaxillofac Radiol. 31:313–316.
Zhou B, Willis A, Sui Y, Anderson D, Thomas T, Brown T. 2009. Improving inter-fragmentary alignment for virtual 3D reconstruction of highly fragmented bone fractures. SPIE Med Imaging.