# COMPUTER VISION-BASED RECONSTRUCTIVE PLASTIC SURGERY

by

YARONG TANG

(Under the Direction of SUCHENDRA M. BHANDARKAR)

## ABSTRACT

High energy traumatic impact of the craniofacial skeleton is an inevitable consequence of today's fast paced society. The work in the thesis leverages recent advances in computer vision, computer visualization and computer aided design/manufacturing to reduce the fractures and reconstruct the craniofacial skeleton *in silico*. A graphical user interface is designed to help the surgeon to perform the surgery in a virtual environment. Surface matching algorithms such as the Iterative Closest Point (ICP) and the Data-Aligned Rigidity-Constrained Exhaustive Search (DARCES) are applied separately to the problem of craniofacial reconstruction by matching the fragment surfaces. A synergetic combination of the DARCES and ICP algorithms where the output of the DARCES algorithm is fed as input to the ICP algorithm results in an improved performance in terms of both reconstruction accuracy and execution time.

INDEX WORDS: Computer Vision, Graphical User Interface, Virtual Surgery, Surface Matching, Computer-aided Reconstructive Surgery, Iterative Closest Point (ICP), Data-Aligned Rigidity-Constrained Exhaustive Search (DARCES)

# COMPUTER VISION-BASED RECONSTRUCTIVE PLASTIC SURGERY

by

YARONG TANG

B.E., Beijing Institute of Clothing Technology, P.R.China, 1994

M.E., Beijing Institute of Clothing Technology, P.R.China, 1997

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2003

# COMPUTER VISION-BASED RECONSTRUCTIVE PLASTIC SURGERY

by

YARONG TANG

| | |
|---|---|
| Major Professor: | Suchendra M. Bhandarkar |
| Committee: | Walter D. Potter<br>Khaled Rasheed |

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2003

# DEDICATION

This thesis is dedicated to my parents and my husband, who have always loved and supported me in all that I have done, and have stood by me whenever I have needed them the most.

# ACKNOWLEDGEMENTS

I would like to offer thanks to all the people who contributed their ideas, suggestions, expertise, and good will through the completion of my thesis research. First, I wish to express my sincerest gratitude to my major professor, Dr. Suchendra M. Bhandarkar, for his invaluable guidance, constant support and great patience. Without his sage advice, this work would not have been completed. I would also like to thank Ananda Chowdhury, Xingzhi Luo, Xiaochuan Yi, Yong Wei and other members in Dr. Bhandarkar's research group for their suggestions and inspiring discussion in the research. Additionally, I thank Congzhou He for proofreading my thesis. I would also like to thank Dr. Walter D. Potter and Dr. Khaled Rasheed for their support and for being members of my thesis committee. I also thank all the professors and students in the Artificial Intelligence Center for their help and support. A special thanks to Dr. Jack Yu and his group at the Medical College of Georgia for providing the CT image data for this research and valuable suggestions as end-users. Finally, I would like to thank my family for their love and concerns, and for their continuous encouragement and unfailing support.

## TABLE OF CONTENTS

# LIST OF TABLES

Page

## LIST OF FIGURES

# CHAPTER 1

## PROJECT OBJECTIVE

High energy traumatic impact of the craniofacial skeleton is an inevitable consequence of today’s fast moving society. The surgeons restore the configuration and function of the fractured bone elements in the craniofacial skeleton typically by first exposing all the fragments, then returning them to their normal configuration, and finally maintaining these reduced bone pieces with rigid screws and plates. There are several important, inherent and intrinsic limitations of this current, standard approach. Sufficient blood supply and relative stability are two cardinal requirements for the fracture healing. To visualize the fragments in order to reduce them necessitates their exposure, which consequently reduces the attached blood supply required for the healing to occur. To improve the blood supply, one can decrease the extent of dissection. However this means not being able to visualize the entire fracture, which could lead to potential malalignments of the bone fragments. Union without proper alignment is called malunion. The younger the individual, the more robust the healing and remodeling potential and thus better the tolerance to malunion. In the case of the fracture of the upper and lower jaws, an inherent dilemma exists. Because the maxilla and mandible house the 32 permanent teeth which must fit precisely in a particular orientation, known as position of intercuspation, malunion is very poorly tolerated. So to avoid this, extensive dissection must be done to visualize all the fractures. However, this stripping of the periosteum inevitably reduces the amount of blood flow to the bone fragments.

This project seeks to solve the above dilemma by developing an enabling technology that leverages recent advances in computer-aided design (CAD), computer-aided manufacturing (CAM), computer vision and computer visualization to reconstruct the craniofacial skeleton and reduce the fractures *in silico.* The result is a computer-generated, accurate, life-sized, three-dimensional reduced craniofacial skeletal model which can be physically realized using laser-polymerizable epoxy. Using this model, rigid titanium mini-plates and screws are adapted to brace the reduced fractures. The number, locations, and lengths of the screws can be optimally determined *ex vivo,* thus avoiding critical neural and vascular structures such as the inferior alveolar nerve and artery. The significance and advantages of this enabling technology include the following: (1) several-fold reduction in the time required in the operating room (2) increased accuracy of fracture reduction, plate adaptation, and screw placement, and (3) improved blood supply to the bone fragments due to limited dissection.

# CHAPTER 2

# INTRODUCTION

## 2.1 Current Status

Bone is a unique structure found only among members of the Phylum Chordata. It is the only living material in the entire animal kingdom that possesses a compressive modulus in the range of 1 x $10^{10}$ N/$m^2$ with a volumetric range from $10^{-9}$ to 10 $m^3$ [1]. Bone is the rigid element in the body which resists deformation, allows for transmission of forces and protects the internal organs. The craniofacial skeleton consists of high stress-bearing buttresses and low stress-bearing curve planes. When external loads are applied, the craniofacial skeleton undergoes strain. When the strain exceeds the ultimate strain limit, which is about 1.0 x $10^4$ microstrains, failure occurs [1]. This loss of continuity is known clinically as fracture and causes pain, disfigurement, and functional impairments due to the disruption of force transmission. But unlike any man-made material, the bone being living tissue will heal. For this healing to occur, there are two cardinal requirements: sufficient blood supply and relative stability. In clinical treatment of fractures, the realignment of the fragments must be achieved prior to fracture healing. Union without proper alignment is called malunion. The younger the individual, the more robust the healing and remodeling potential and thus better the tolerance to malunion. In the case of the fracture of the upper and lower jaws, an inherent dilemma exists. Because the maxilla and mandible house the 32 permanent teeth which must fit precisely in a particular orientation, known as position of intercuspation, malunion is very poorly tolerated. So to avoid this, extensive dissection must be done to visualize all the fractures. However, this stripping of the periosteum inevitably reduces the amount of blood flow to the bone fragments. Due to the 3D geometry of the masticatory system, a $1^o$ angular malalignment in the anterior region of the mandible will result in 8 to 10 mm transverse displacement at the area of the second molars. This severely disrupts the intercuspation and is a dreaded post-operative condition known as malocclusion. In some cases, the mandibular width can be so altered that the condyles are no longer within the glenoid fossas of the temporal mandibular joints.

Due to the elaborate structure of facial skeleton, the standard approach for severely fractured facial skeleton reconstruction is to dissect and expose all the fragments, and then realign them to the normal configuration and fix them by rigid screws and plates. In order to achieve good alignment, it is unavoidable to prolong the exposure time of the incision, and hence, slow down the healing process. Since 2001, Dr. Jack Yu's group at the Medical College of Georgia (MCG) has been developing an improved method for achieving good bone alignment with minimum periosteal stripping. To achieve this, the bone fragments are aligned *ex vivo*. The axial Computer Tomography (CT) scan pictures are digitized as JPEG files and imported into a Power Point Presentation. The fragments are traced and manipulated into their pre-injured position using the mouse. The symmetry is then verified and the scale adjusted to life-size. A three-dimensional print is then made of the "realignment" and the titanium miniplates are adapted to the correct configuration. Thus during the actual surgery, much less time is required for plate adaptation.

Developments in computer imaging, computer vision, endoscope technology, and surgical instrumentation predict a new revolution in the field of surgery. Computer-assisted surgery is no longer a product of science fiction. In a recent report by the American Medical

Association, the major break-throughs in the field of surgery for the first half of the twenty-first century will be in the area of surgical robotization and automation. Minimally invasive surgery and robotic surgery have gained momentum and attracted intense investigation, and have made it possible to perform a surgical procedure without directly visualizing or touching the structure being operated upon. Advances in computer-assisted surgery and other relevant enabling technologies would allow surgeons to do what could not be done before, and also to do it with greater efficiency and precision. The new paradigm shift reduces the patient's pain, and discomfort, and gives the patient a quicker return to functional health and a productive life [2].

This research is built on Dr. Jack Yu's basic approach by exploiting the current advances in computer-aided design (CAD), computer-aided manufacturing (CAM), computer vision, virtual reality and computer visualization technologies. On completion of this project, a patient with severe facial fractures will have Computer Tomography (CT) images loaded into a program which will automatically recognize the fracture planes/surfaces, reduce the fracture surfaces by realigning the fragments, and feed the results of the *in silico* reconstruction to a laser-3D lithographic printer which will generate the reconstructed, life-sized epoxy model. The plate manufacturers, using this epoxy model, will machine the plates to best brace the fractures, select screws of the optimum size and length and select the positions on the model where the screws can be placed in an anatomically safe and mechanically advantageous manner. The entire construct will be sent back to the surgeon in the operating room and the surgery will be carried out in a much shorter time frame with vastly improved accuracy. In addition, this technology will also allow the fixation device to be deployed using endoscopic or minimally invasive surgical procedures.

Regarding the other currently existing devices or methods, in our extensive survey of the literature, only Kobayshi [3] adopted CAD/CAM, computer vision and computer visualization technologies to produce customized bone substitute for large skull defects. While there exists work published in the literature dealing with simulation of mandibular fractures [4,5] and simulation of dental implantalogy [6], there is little reported by way of computer-aided surgical reconstruction of the craniofacial skeleton. There has been some recent progress in the design of thoracolumbosacral orthosis braces for postoperative patients needing back correction and/or support using CT and Magnetic Resonance (MR) images of the spinal region [7]. However the techniques described exhibit limited automation and entail substantial intervention by the surgeon and radiologist [7]. This reinforces our belief that the work presented here is at the cutting edge of computer-assisted surgery.

There are three phases to the research project. Phases 1 and 2 constitute the preclinical portion of the proposed research and comprise of the study of feasibility and development of a proof-of-concept prototype of the enabling technologies. In Phase 1, a polymerized epoxy phantom of the lower mandible is subject to fractures simulating a high-energy traumatic impact. An artificial model of the craniofacial skeleton, specifically, the lower mandible, is used to simulate fractures. Algorithms for craniofacial reconstruction are designed and tested on this simulated data. The goal of Phase 1 is to demonstrate the feasibility of applying techniques and algorithms from CAD, CAM, computer vision, computer visualization and virtual reality to the problem of craniofacial reconstruction *in silico*. Phase 2 represents the transition from simulated data to real data from CT files of patients with severe facial fractures. Phase 2 concludes with the ability to produce rapid prototype plates for use in craniofacial reconstruction and thus completes the study of feasibility started in Phase 1. Phase 3 comprises of performance of clinical trials necessary for testing the enabling technologies, including steps towards obtaining FDA approval, and steps towards eventual commercialization of the developed technologies. The work presented here is the result of Phase 1 as described above.

There are two targets for Phase1 of the project: first, to develop a Graphical User Interface (GUI) to allow the surgeon to manually identify corresponding landmarks on opposable fracture surfaces on the simulated data (both in 2-dimensional and 3-dimensional space), and second, to develop algorithms to reduce the distances separating the two opposable fracture surfaces and move them together by translation and rotation such that the fragments are properly realigned.

To a first-degree approximation, the facial skeleton can be viewed as a rigid object and the distortion can be ignored. Therefore, the same transformation can be applied to the entire fracture surface and will not cause any shape distortion. Two techniques are employed to estimate the transformation between the two opposable fragments. One is the Iterative Closest Point (ICP) algorithm [14] augmented with the bipartite graph matching algorithm [15], and the other is the Data-Aligned Rigidity-Constrained Exhaustive Search (DARCES) algorithm [8], which is based on the Random Sample Consensus (RANSAC) algorithm [9].

## 2.2 The ICP and Bipartite Graph Matching Algorithms

The registration of two 3D data sets entails the calculation of the geometric transformation that should be applied to one of the data sets in order to match it to the second data set. The Iterative Closest Point (ICP) algorithm proposed by Besl and McKay [14] is the most widely used method for 3D rigid data registration. It is a general-purpose, representation-independent registration method. The ICP algorithm can handle multiple representations of geometric data: point sets, line segment sets (polylines), implicit/parametric curves, triangle sets (faceted surfaces), and implicit/parametric surfaces. It requires no preprocessing for feature extraction and no correspondence establishment between the points in the two data sets. However, it does require a procedure to find the closest point data set from one point data set corresponding to the points in the other data set.

Given two 3D point data sets A = {$a_i$, i = 1,…, m} and B = {$b_i$, i = 1,…,n} for registration, where m and n may be different integers such that $m \leq n$ (without any loss of generality), the ICP algorithm solves the problem using three basic steps. The first step is to evaluate the correspondence between the two data sets, i.e. to determine the closest set for A by pairing each point in A to the closest point in B based on the Euclidean distance between this point and the data set B. The next step is to compute the transformation using a quaternion-based method that minimizes the mean squared error (MSE) between the paired points. The last step is to apply the transformation to data points in set A and calculate the MSE between the updated version of set A and the already evaluated closest set. The above three steps are repeated until the MSE falls below some specified threshold value. The ICP algorithm is proven to converge in terms of the MSE-based criterion; actually it always converges monotonically to the nearest local minimum. Experience has shown that the rate of convergence is rapid during the first few iterations.

The main problems for the ICP algorithm are the convergence to a local minimum, susceptibility to gross statistical outliers and intensive computation needed in order to find the closest set. An easy way to deal with the local minimum problem is to initialize the ICP algorithm with a few different starting registration vectors (rotation + translation), and choose the best resulting solution.

A Bipartite Graph [21] is a special kind of graph where the original vertex set V can be partitioned into two disjoint sets V1 and V2 such that each edge has one vertex in V1 and the other vertex in V2. Such a partition (V1, V2) is also called a bipartition. A complete bipartite graph is a simple bipartite graph with bipartition (V1, V2) in which each vertex of V1 is joined to each vertex of V2; if |V1| = m and |V2| = n, such a graph is denoted by $K_{m, n}$. Bipartite Graph Matching is a very well-known problem in graph theory. The general idea is to find a complete matching of the original bipartite graph G such that each vertex in V1 is matched to one and only one vertex in V2 where $m \leq n$. There are many different variations of the original bipartite graph matching problem such as finding the perfect matching problem [22], and the optimal assignment problem using the Kuhn-Munkres algorithm [17, 23]. Literature citing matrix-based approaches for determining the solution to the optimal assignment problem for a complete bipartite graph is also quite common [15]. Bipartite graph matching plays an important role in computer vision and pattern recognition. Specifically in complex tasks such as 3D object recognition, graph theoretic approaches seem quite popular. Kim and Kak [16] used maximal bipartite graph matching along with discrete relaxation techniques to achieve large efficiencies in 3D model-based vision. Hamdan *et al.* [24] use a maximum weighted bipartite graph matching technique for fast 3D object reconstruction using trinocular vision and structured light. Along similar lines, Cheng *et al.* [25] employ maximum weight bipartite graph matching for image feature matching. For our present problem, we used a Maximum Cardinality Minimum Weight Bipartite Matching algorithm (which is the solution to an optimal assignment problem) for determining the Closest Set in the Iterative Closest Point (ICP) algorithm as outlined in [15].

## 2.3 The DARCES and RANSAC Algorithms

Chen *et al.* proposed a Data-Aligned Rigidity-Constrained Exhaustive Search (DARCES) algorithm [8] to solve the partially overlapping 3D registration problem efficiently and reliably. This algorithm requires no preprocessing or local feature detection and no initial transformation estimation for the matching of the two 3D data sets. The DARCES algorithm is different from any feature-based approach or iterative approach for the 3D point registration problem. It is a good choice for data sets with no prominent or salient local features. In the noiseless case, the DARCES algorithm can guarantee that the solution it finds is the true one, on account of its exhaustive search for all possible data-alignments of the two 3D data sets.

Given two partially overlapping data sets, namely the sample data set and model data set, the registration problem can be solved as follows: first, a set of reference points is selected from the sample data set. The requirement for the reference points is that they must lie on the overlapping region of the two data sets. From these reference points, a number of control points are chosen, and their corresponding points in the model data set are found using the geometry constraints posed on the control points. This search process can be depicted as the sliding of the control points on the model surface. It is possible to have more than one solution to the matching of the control points since only the geometric constraints are used. The corresponding pairs of control and model points are used to compute the transformation. This resulting transformation is then applied to all the reference points (except the control points) to get the corresponding transformed points. If the distance between a transformed point and the model surface is smaller than a prespecified threshold, then this reference point is counted as having been successfully aligned on the model surface. The transformation with the largest overlapping number is considered to be the solution to the registration problem.

Usually, three control points are selected so that they form an acceptable triangle. If a smaller triangle is used, then a faster search speed can be achieved. However, if the triangle is too small, then the computed rigid transformation will be very sensitive to noise. In the case where more than three control points are used, the optimal configuration of the control points is quite data dependent. Hence, it is important to choose a good number of control points having a good distribution.

The Random Sample Consensus (RANSAC) algorithm [9] is used for robust fitting of models in the presence of outliers or noisy data. For parameter estimation, classical techniques assume that the gross errors can always be smoothed out by the good values in the data set, and hence they have no internal mechanisms to detect and reject the outliers. The RANSAC algorithm handles this problem with a different approach. Instead of using as much data as possible, the RANSAC algorithm starts with a small initial data set, which is randomly selected and is just enough to solve the problem (i.e., fit the model), and grows this set with inclusion of consistent data when possible.

The DARCES algorithm can be used in conjunction with the RANSAC algorithm to get rid of the noisy data. The first (primary) control point is selected randomly from the sample data set, and then the basic DARCES algorithm is run. A number of such random trials are performed until the solution is found or the stopping criterion is reached.

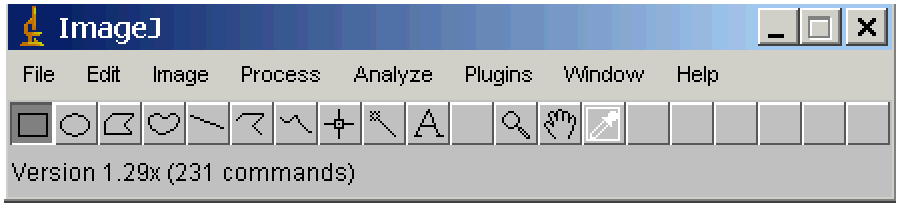
# CHAPTER 3

# GRAPHICAL USER INTERFACE

The system for computer-aided surgery can be viewed as a synergetic combination of computer-vision based automation and graphical user interface (GUI) based human (surgeon) intervention. The input to the system is a sequence of Computer Tomography (CT) slices of a fractured human mandible, and the computer output is the sequence of images of the recovered mandible structure. This makes image display and visualization an important part in this project. Beyond that, it is necessary for surgeons to get involved with the system to make sure the job is done right. Surgeons communicate interactively with the system by providing the corresponding landmark points, classifying confusing data or performing some other operations helpful for the broken mandible reconstruction. Hence, a good interface is desirable for the success of the project.

ImageJ [10] is a public domain Java-based image processing program, developed by the National Institutes of Health (NIH) Bethesda, Maryland. It has several features that make it appropriate for this project. Specifically, it supports the “stacks” idea, that is, a series of images of the same type in the same folder can be opened in a single window. This feature is very important and useful for the mandible reconstruction since one needs to work on a series of CT images at the same time. ImageJ is designed with an open architecture and its functionality can be expanded via Java plugins, hence the users not only have a display tool with a group of embedded functions, but can also add new functions required by a project very conveniently.

Some other features such as displaying the16-bit CT images, supporting multiple data types and their conversions, and saving the results back as a series of images make it a good choice for medical image processing. Fig. 3.1 shows the interface of ImageJ.



Version 1.29x (231 commands)

Fig. 3.1 The interface of ImageJ

For the surgeon’s convenience, the new functions developed are incorporated into a front-end system. For the front-end system, a simple but compact GUI consisting of buttons for performing various functions was designed. Good interface design principles [11] were followed throughout the development process. Considerable attention was given to various factors such as the orientation of buttons, size of buttons, font-size for the titles of the buttons, etc. The use of color in the GUI design was intentionally restricted only to gray and black considering the average age group of the users (surgeons in the present case) who will use it. Each button is entrusted with a specific function such as rotation, translation, surface matching, etc. The label on each button indicates the task it is designated to perform. Thus a surgeon can perform the desired operation just by clicking the appropriate button. Fig. 3.2 illustrates the front-end design for a typical surgeon.

To increase a surgeon's flexibility for the GUI operations, some GUI functions are offered with multiple choices. In such cases, a second level interface will pop up with all the options when the user clicks on the button. For example, for the rotation operation, the user may want to rotate the object by an angle about its center, or the user may want to specify an arbitrary rotation center. In order to rotate a fragment, the user can click on the *Rotate* button on the first level interface (see Fig. 3.2), upon which a submenu screen pops up. Fig. 3.3 displays the pop-up menu with the various options for the rotation operation. At this point the user may choose the button entitled *2D-Mouse*. Once this button is clicked the user will be asked to input the angle of rotation. The user can then use the mouse to specify the center of rotation, which is also known as the pivot point for the rotation operation.

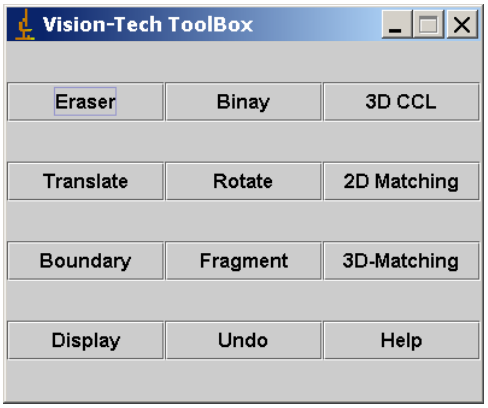


Fig. 3.2 The pop-up menu with new functions for a surgeon

The GUI functions can be applied to an area or a component. Some functions are able to operate on the whole stack, but not others. For example, the main purpose of the *Eraser* function is to separate two fragments that are connected but which the surgeons think should be separate.

Under such a situation, the *Eraser* function is used to do the preprocessing for data retrieval: either to cut the false connection or to create a small gap between the joined fragments, or to remove any noisy artifacts. Usually, the location of the false connection on the first slice is not necessarily the location of the false connection on the next slice, so it is only appropriate for such a function to work on a single slice each time.

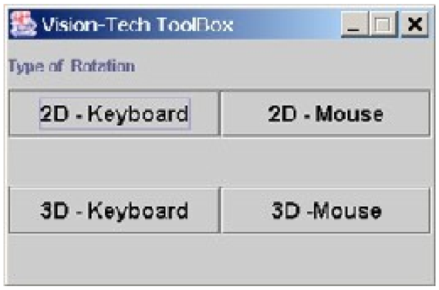


Fig. 3.3 The pop-up submenu for the rotation operation

Listed below is an overview of the currently implemented functionalities (i.e., operations) in the user interface:

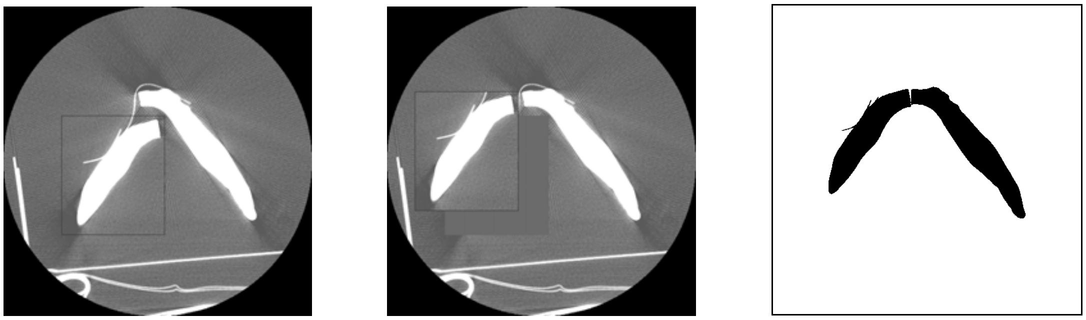
- *Eraser* manually separates two fragments or removes noise on a 2D grayscale CT image slice.
- *Binary* thresholds the grayscale CT images into binary images using *a-priori* knowledge to decide the threshold value. It works on both a single image and a stack of images, and converts the image from 16-bit to 8-bit.
- *3D CCL* identifies 3D objects using the results of the 2D Connected Component Labeling (CCL) propagated along the z dimension. It is equivalent to 3D CCL using 6 neighbors.

- *Rotate* rotates the chosen area or component by a specified angle, both in 2D and 3D. The user can specify the rotation pivot optionally.
- *Translate* translates (or drags) the chosen area or component by specified x, y values. It works for a single image or a stack of images, but is not allowed to translate a component along the z-axis.
- *2D Matching* matches two fragments on a 2D slice. The user marks or specifies the correspondence between a number of pairs of contour points (landmark points) on the two fragments. The two fragments are brought together via rotation and translation. The transformation is obtained by minimizing the mean squared error (MSE) between the landmark points [12]. The Singular Value Decomposition (SVD) algorithm [26] is used to compute the transformation. This requires high accuracy of correspondence between the landmarks and usually works on binary images after CCL.
- *Boundary* interactively retrieves the contour data consisting of the fragment surface from each slice for each fragment. The user has to specify the two end points of the desired portion of the contour part in a binary image.
- *Fragment* interactively retrieves all the data points belonging to one 3D fragment. The user specifies the data area via a selection tool. The 3D rigid-body transformation is applied to the fragment data to perform the mandible reconstruction. A binary image is required for this operation.
- *3D Matching* is the central function of the system. It registers the two fracture surfaces and uses the resulting transformation to reduce the fragments back to their right positions. Two algorithms are implemented: The Iterative Closest Point (ICP) algorithm and the Data-Aligned Rigidity-Constrained Exhaustive Search (DARCES) algorithm. The

DARCES algorithm in turn uses two different approaches to compute the rigid body transformation.

- *Display* is a utility function used by all 3D matching algorithms to display the resulting CT image stack after applying the computed transformation from the 3D surface matching algorithm to the original stack.
- *Undo* is developed to enable the surgeon to easily go back to a previous state. Currently, only the translation and rotation operations are eligible for this operation.
- *Help* documents all the functions.

The figures below show the operations of translation, rotation, and 2D contour matching on CT images of the human mandible:



(a) (b) (c)

Fig. 3.4 Translation. (a) Slice with rectangular area selected for translation. (b) The rectangular area in (a) moves along x-axis by 50 pixels, along y-axis by 40 pixels. (c) Drag the left component in (a) to a new position after binarization, size filtering and 2D CCL.

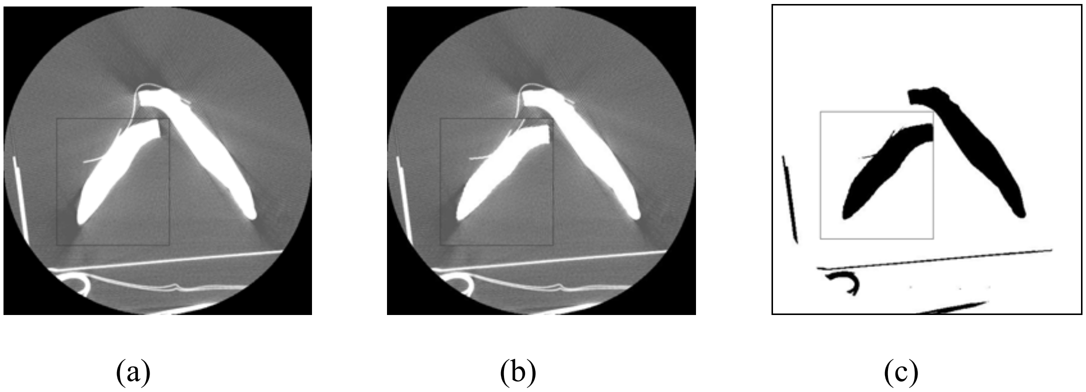


Fig. 3.5 Rotation. (a) Original image with area selected for rotation. (b) The selected area in (a) rotated about its center by 7°, in the clockwise direction. (c) Similar rotation of 6° on a binary image.

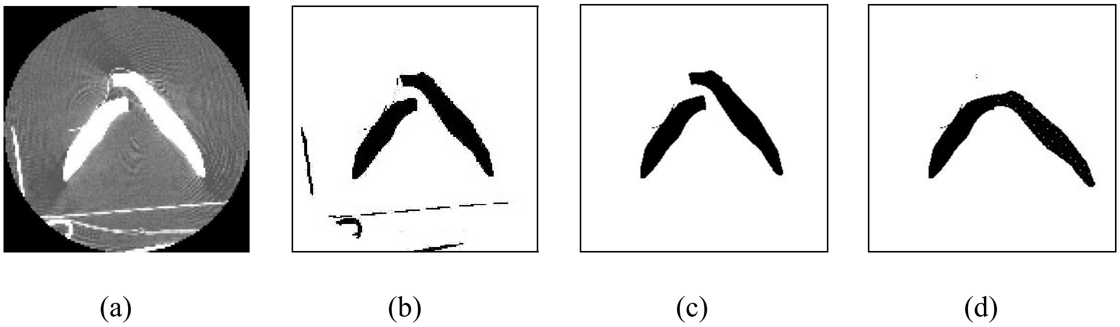


Fig. 3.6 Contour matching for a 2D slice. (a) Original CT image. (b) Binary image. (c) After size filtering and 2D CCL. (d) Contour matched image.

The first version of the GUI front-end design has been tested by the surgeons, but the current version with extra functions added still waits for further user-evaluation and will be updated according to the surgeon's preference. Surgeons may prefer to put all the functions along with the available options on the same interface, or they may like the interface pictured as a layered tree with the options as sub-nodes. Some functions that are currently helpful may not be so valuable in the future when a more automated system is available as the research progresses. However, functions such as landmark identification, correspondence specification and confusing data classification, can assist the surgeon to supervise the reconstruction process and give valuable information to the program. The currently displayed set of functions is what we think are useful to the surgeon as of now. This set of functions can always be updated according to the need of the surgeons performing the surgical operations. With the available plugin feature in the ImageJ package, it is possible to implement the appropriate function for the image processing task at hand.

# CHAPTER 4

# INTERACTIVE DATA ENTRY

## 4.1 Image Data Description

Two series of 16-bit (unsigned short: 0 - 65535) grayscale CT slices are provided by the Plastic Surgery Department of the Medical College of Georgia (MCG). The CT slice thickness is 1.25 mm. Each slice has 512 x 512 pixels, but the resolution may differ for different data groups. The two series of CT images represent two groups of data. The first group has 120 slices for a complete picture of the phantom head and each CT slice covers an area of 266 x 266 mm. The second group has 105 slices for the broken phantom mandible representing an area of 150 x 150 mm. A few slices from each group are illustrated in Fig. 4.1 and Fig. 4.2, respectively. The two groups of data are different in terms of image quality. The second group of images has much more noise. Typically, a computer monitor can show only 256 shades of gray. Therefore, the data are mapped to an 8-bit representation for the display purpose via windowing. The window defines the range of gray values displayed: values below the window are made black, whereas values above the window are made white. Although the image display is remapped, the underlying image data value is not changed.

When stacking all the CT slices, the following coordinate system is assumed: the default 2D image coordinate system is used for each slice where the x-axis is from left to right, the y-axis from top to bottom, and the z-axis refers to the slice number. The z-value is 0 for the first slice in a stack, and is increased by 1 for the next slice in the same stack. The z-value is the same for all points in the same slice. The x, y-values are given in pixels, but z-value is measured in terms of slice thickness. Thus the x, y-values need to be converted into metric measurements before the transformation computation for registration, and converted back into pixels for image display after the transformation computation.

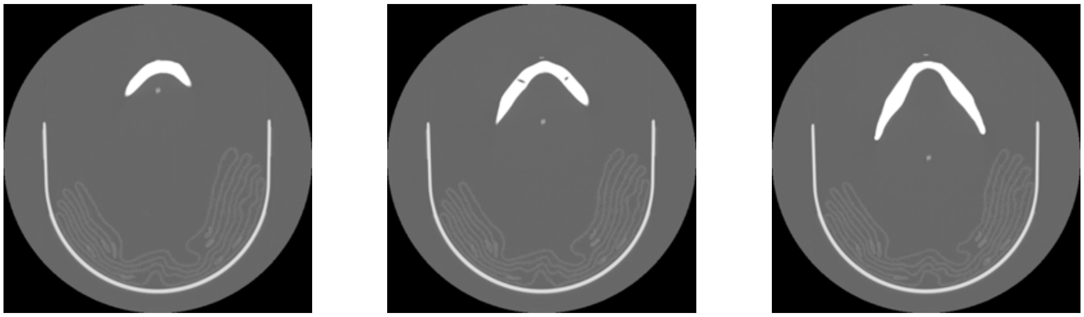


Fig. 4.1 CT slices from the first data group: the complete phantom mandible structure.

Image dimension: 512 x 512 pixels, covered area: 266 x 266 mm, slice thickness 1.25 mm.

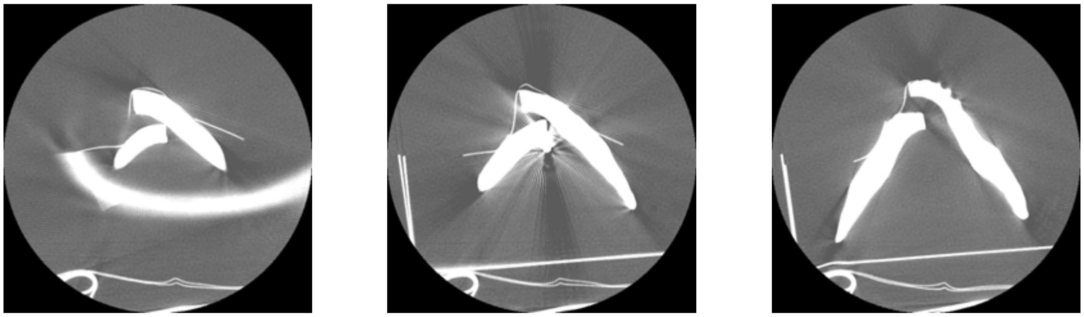


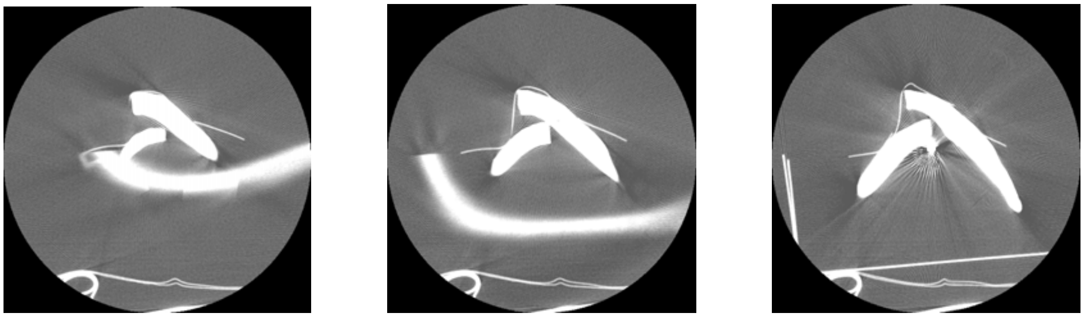
Fig. 4.2 CT slices from the second data group: a phantom representing a fractured mandible.

Image dimension: 512 x 512 pixels, covered area: 150 x 150 mm, slice sickness 1.25 mm.

In order to reconstruct the broken mandible fragments and return them back to their right positions, the image data has to be first retrieved. A binary image is desired since only the bone fragments are the objects of interest. The data retrieval process is divided into a few steps: preprocessing to remove noise, binarization to get binary images, size filtering to filter out small noisy areas, connected component labeling to identify the fragments, contour data retrieval and fragment data retrieval.

## 4.2 Image Preprocessing

In the mandible CT images, the bright parts represent the broken mandible fragments and the dark areas represent the soft tissues. Usually, the two parts are very well separated. However, it is not so for the second group of the given data derived from a phantom. The pictures in Fig. 4.3 illustrate a few cases which make it necessary to do preprocessing before binarization. Compared to the images of the unbroken mandible on Fig. 4.1, some of bright areas are found to be the noisy data. As informed by the surgeon, these white areas are introduced when the data are created using a phantom. Before giving a traumatic impact to the phantom, the surgeon wrapped some white tape around it in order to keep the phantom from completely shattering. A screw is also used in the process. However, these items have high gray values just as bone structures do. These noisy data are not only classified falsely as bone fragments, but also make the connected component labeling (CCL) in 3D impossible to perform on this data set, because the false connection between two actually independent fragments misleads the computer into considering the two fragments as one fragment. The usual preprocessing techniques such as filtering, and smoothing, cannot help to remove such noise, as showed in the next section. The noisy part is removed manually by an *eraser* operation. The unwanted area with an arbitrary shape can be selected and after clicking the *Eraser* button on the interface, the area will be replaced by background gray values, and then after binarization, it turns to white background.



(a) (b) (c)

Fig. 4. 3 Noisy data in the CT slices

The *eraser* function can be used to manually separate two fragments with neighboring pixels. Fig. 4.3 (b) illustrates the case where the two fragments touch each other, which make CCL unable to identify the two fragments as two independent components in 3D. The final transformation cannot be applied to one of the pieces in such cases since they are considered to be one fragment. Therefore, the *eraser* function is used to create a very small gap between the two touching fragments in order to separate them.

## 4.3 Binary Image Generation

The second group of slices deals with the broken mandible fragments, and various algorithms are applied to solve the registration problem. The first step is to generate the binary images and to separate the fragments from the background. ImageJ provides a *Binary* thresholding function (where the threshold level is based on image histogram analysis, an iterative threshold selection method is used), but the result is unsatisfactory, as shown in Fig. 4.4. It works well on image 4.4(d), where the slice is from the first group of data, but poorly on other three images derived from CT image slices from the second group of data which contains noise and has highly non-uniform texture. Filtering and smoothing techniques such as median filtering, adaptive smoothing [21], were used to improve the image quality before binarization. The results are shown in Fig. 4.5. The noise is largely suppressed compared to Fig. 4.4, but the major problem with these techniques is that they will modify the contour data, as illustrated in the last two magnified images in Fig. 4.5. The surgeon's knowledge indicates that the bones have a much higher density than soft tissues, that they are much brighter than other parts and their gray values are above 34000 in the CT images. This *a-priori* knowledge is applied to choose a simple threshold for binarization, and a size filter is then used to remove the components of small size. The threshold value for the size filtering depends on the user's definition of the desired component size, for example, a tooth size. The binary images of a few slices are shown in Fig. 4.6 where this *a-priori* knowledge is used to threshold the CT image. The 2D CCL results are propagated along the z-axis to identify 3D fragments but the result is not usable. If incoming new data are free of such noisy artifacts caused by white tape, screws, false connections etc., then the 2D CCL results propagated along the z-axis can be used to detect fragment automatically, or at least semi-automatically with manual to classify the most confusing part. In this way, a lot of labor involved in data retrieval can be saved.

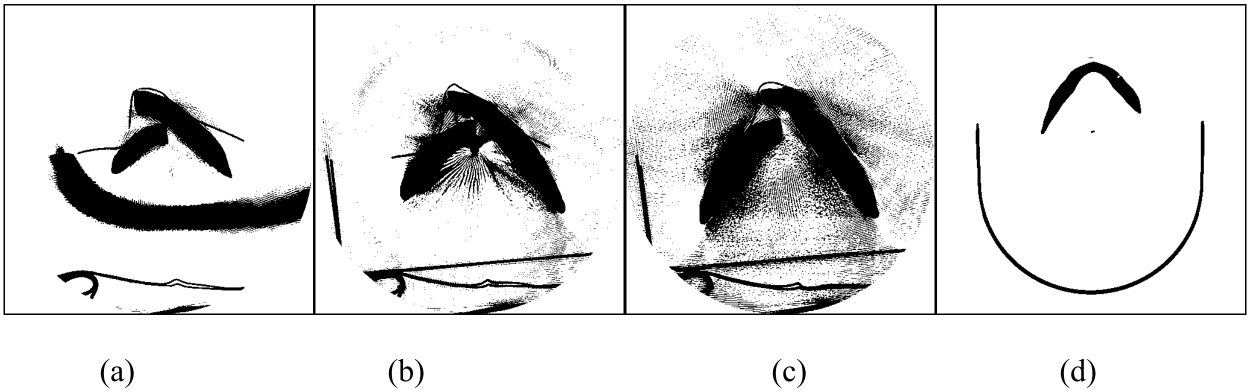


Fig. 4.4 Binarization result using ImageJ's autothreshold binarization function. (a)-(c): images from the second group of data. (d) is the image slice taken from the first group of data.

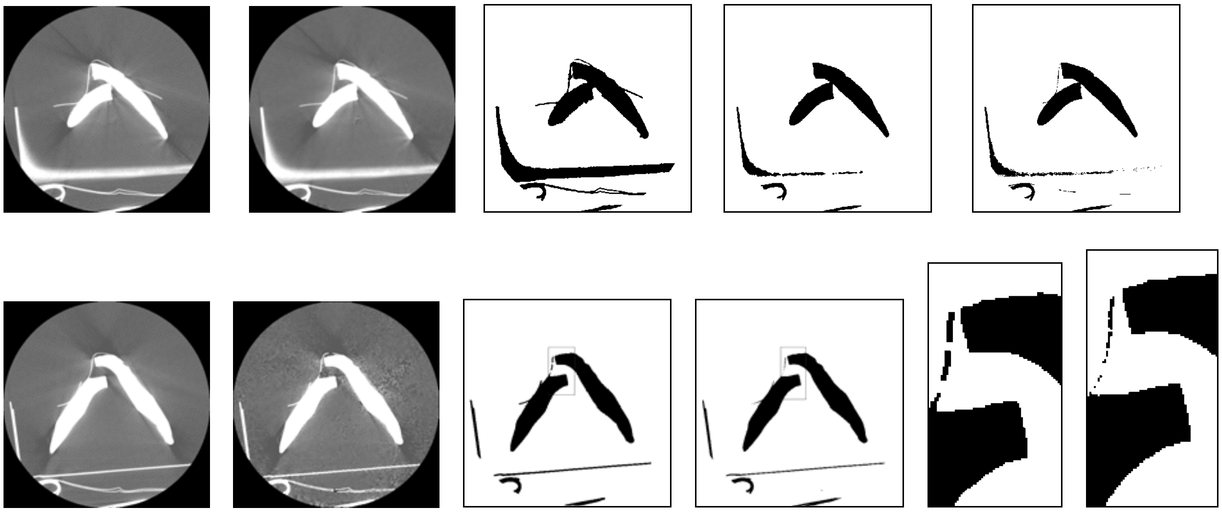
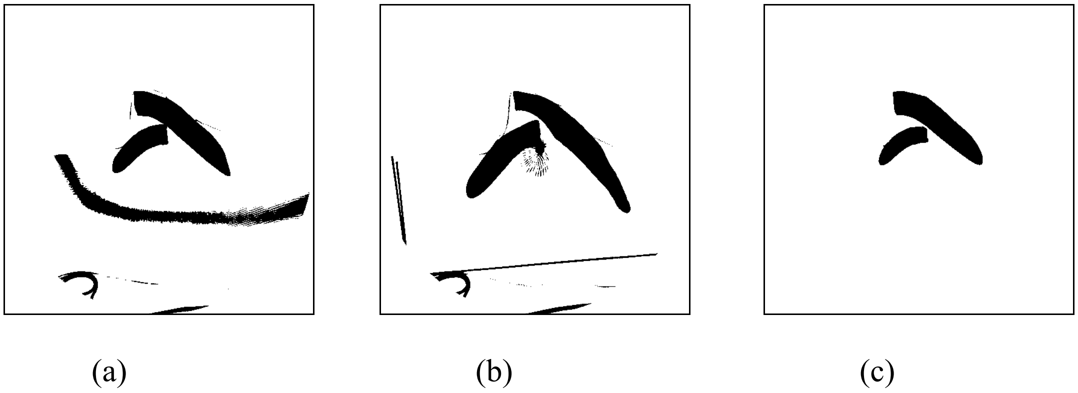


Fig. 4.5 Results of filtering and smoothing techniques before binarization. Upper row (median filter result): the 1st picture is the original image; the 2nd picture is the image after median filter with radius 2 pixels; the 3rd picture is the binary image of the 2nd using ImageJ binary function; the 4th picture is the binary image of the 2nd image using *a-priori* knowledge to select the threshold; the 5th picture is the binary image of the 1st image using *a-priori* knowledge to select the threshold. Bottom row (the adaptive smoothing result): the 1st picture is the original image; the 2nd picture is the result of adaptive smoothing on picture 1 with k = 6 (k: the smoothing parameter); the 3rd picture is the binary image of the 2nd image

using *a-priori* knowledge for threshold selection; the 4th picture is the binary image of the 1st image using *a-priori* knowledge for threshold selection; the 5th and 6th pictures are the magnified versions of the two selected frames to compare the smoothing effect in the 3rd and 4th pictures, respectively.



(a) (b) (c)

Fig. 4.6 Binarization with threshold, threshold selection using *a-priori* knowledge

### 4.4 Fragment Surface Data

The originally intact phantom of the mandible will form a few new fracture surfaces once it is broken. The broken mandible reconstruction is based on the fracture surface matching of the two broken fragments. Surface data are represented by the contours along the fracture surface on each CT slice. When binary images are ready, it is easy to retrieve these edge data along the fragment contour, which, in turn, are assembled to form the 3D fragment surface data set.

A range of slices that form the lower mandible structure were taken from the second group of data to construct a new working data set, where the slices range from slice 8 to slice 43 in the original data set. The fragment surface data are collected for all fragments one by one. For each fragment, all of the component slices for the target fragment are traced to retrieve the contour data using a boundary following algorithm. Button *Boundary* on the interface performs this function interactively. The user needs to tell the program three things: the approximate position of the two end points for the interesting portion of the contour, the upward or downward direction to reach the topmost or bottommost point along the contour, and the edge tracing direction, then the program will get the edge data and write them to a file for later use. This method will introduce a small number of false edge data and miss some real edge data close to the two end points, but such errors can be minimized by scaling the image and then choosing the right end point positions for contour tracing. Furthermore, the contour data close to its two end points on each slice are not used for the surface matching to avoid the error from such data. Fragment surfaces on most slices have apparent joint points to indicate the accurate terminal points of the desired contour portion, but some slices are smooth around the corner, as in Fig. 4.7. On the other hand, neither of the two algorithms chosen (the ICP algorithm and the DARCES algorithm) to compute the 3D transformation requires feature correspondence, so feature detection is not required. Thus the method outlined above is a practical and fast means to get the surface data. Surface data are used as the input data for surface matching by determining the correspondence between the two surface data sets and solving for the rigid body transformation.

## 4.5 Volume Fragment Data

The button *Fragment* invokes the function to retrieve data points belonging to one volume fragment. For the target fragment, the program goes through all the slices consisting of the fragment, and the user uses the selection tool to specify the interesting area for recording the fracture surface data. Usually, the fragment area is easy to define, but the user has to be careful when the small gap between two fragments is created manually. In future, this task can be done automatically if 3D CCL can be applied on the input data. Once the volume fragment data are retrieved (including the surface data), they are saved in a file and used to move the whole fragment to the correct location after rigid transformation is computed by means of surface matching.

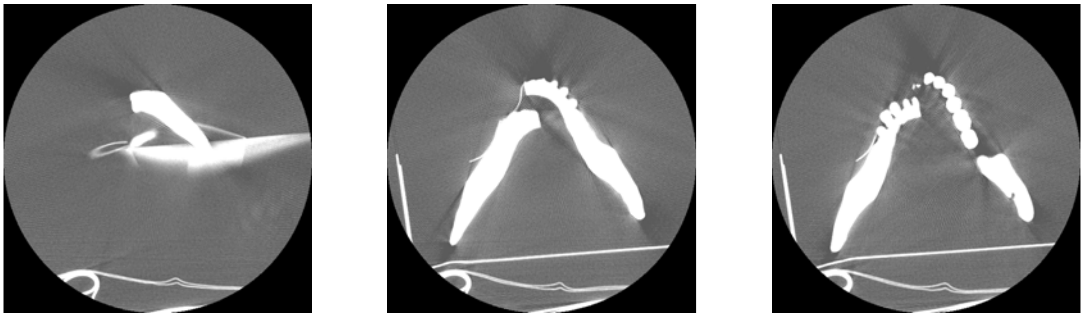


Fig. 4.7 CT slices without prominent features along the fragment surface

# CHAPTER 5

# THE ICP ALGORITHM

## 5.1 Introduction

A key concept in the discussion of registration fundamentals is that of correspondence. We can consider two separate 3D point sets A = $\{a_1,\ldots, a_N\}$ and B = $\{b_1,\ldots, b_N\}$ with equal number N of 3D points. If it is known *a-priori* that each point $a_i$ corresponds to a physical point $b_i$, then the two sets are said to be in correspondence. This knowledge makes the registration problem simple and a closed form solution can be found easily. But in many cases, this knowledge of correspondence is difficult to obtain beforehand. So, in these cases we are left with a 2-fold task, the first is to establish a correspondence between the two data sets and the second task is to compute the rigid body transformation that will bring the two data sets into registration.

The task of the Iterative Closest Point (ICP) algorithm [14] is to establish a correspondence between two sets to be matched as well as to compute the 3D rigid body transformation that will bring the two sets into registration. More often, the cardinality of the two data sets to be matched need not be the same. Let us denote the two sets of points to be matched as A and B, but with different cardinality. So, say A has n points and B has k points. Without any loss of generality, we can assume that $k \geq n$. Let us regard B as the *model* shape to which the points of A, termed as the *sample* shape are to be aligned. The algorithm works in an iterative fashion in which at each iteration the set A is rotated and translated a little closer towards the model set B. The iteration is initialized by setting the following parameters: $A_0 = A$, the rotation matrix R is initialized to an identity matrix I and the translation vector T is initialized to null. Basically the registration vector is represented as $q = [q_R \mid q_T]$ where $q_R$ represents the quaternion due to the rotation and $q_T$ represents the translation portion. So, the initial $q = [1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$.

## 5.2 The ICP Algorithm

The steps of the algorithm are as follows:

(a) Compute the set C which is a subset of B, such that for each point $a_i$ ε A, there is a corresponding point $c_i$ ε C. The set C is called a *Closest Set.* It is interesting to note that although the cardinalities of the two sets to be originally matched i.e. A and B are not the same, the cardinalities of the sets A and C are the same.

(b) Compute the centroids of the two data sets $\mu_A$ and $\mu_C$.

(c) Compute the co-variance matrix $\Sigma_{AC}$.

(d) Compute a 4x4 real symmetric matrix Q from $\Sigma_{AC}$.

(e) The unit eigenvector that corresponds to the maximum eigenvalue of Q is the optimal rotation quaternion $q_R$.

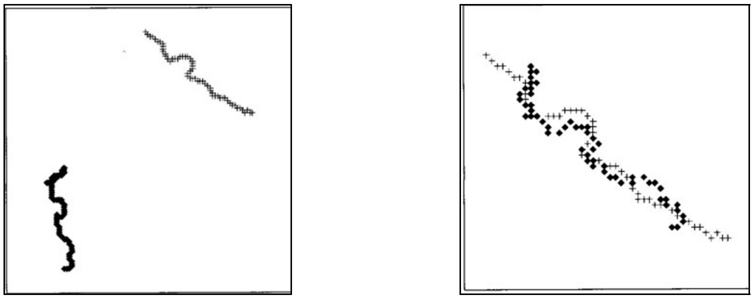
(f) From $q_R$, we can compute the rotation matrix R. Then we can evaluate the translation vector T by: $T = \mu_C - R\mu_A$.

(g) Next apply the translation and rotation to $A_0$ to get $A_1 = q(A_0)$ .

(h) Compute the Mean Squared Error (MSE) between C and $A_0$.

(i) Terminate if the error falls below a certain specified threshold. Otherwise go to step (a) with the updated value of A, as obtained from step (g) and follow all the steps from (a) to (g).

The ICP algorithm has been proven to converge monotonically to a local minimum. Fig. 5.1 [19] illustrates the result of registration of two data sets using the ICP algorithm.



(a) (b)

Fig. 5.1 Two point sets before and after registration by the ICP algorithm

(a) Two point data sets before the ICP algorithm; (b) Same data sets after the ICP algorithm.

### 5.3 Implementation Issues

The ICP algorithm exhibits many diverse and interesting aspects, which are very important to know as far as implementation of the algorithm is concerned. Some of them are discussed below:

#### 5.3.1 The Closest Set Computation in the ICP Algorithm

To compute the *Closest Set*, which is the most crucial step in the ICP algorithm, the matching pairs are determined using the *Maximum Cardinality Minimum Weight Bipartite Matching* algorithm based on the Hungarian method proposed by Kuhn [17]. Fig. 5.2 portrays a typical bipartite graph. In the present case, for the *Bipartite Graph* G(V, E), the two 3D data sets (namely the sample and model point sets) correspond to the two disjoint vertex sets (V1, V2).

The *edge-weight* ($W_{ij}$ ε E) between any two nodes i and j (such that i ε V1 and j ε V2) is the Euclidean distance between them. Note that the Euclidean distance is invariant to a 3D rigid body transformation. Thus, the edge weights are given by:

$$W_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2\right]^{1/2}$$

where $x_i$, $y_i$, $z_i$ represent the x, y, z coordinates of any node i in the vertex set V1 and similarly $x_j$, $y_j$, $z_j$ represent the x, y, z coordinates of any node j in the vertex set V2.

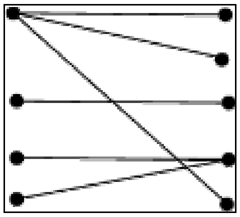


Fig. 5.2 A typical Bipartite Graph

When the *Closest Set* is computed in a greedy fashion, the MSE for the matched surfaces turns out to be quite large. Moreover, the number of iterations of the ICP algorithm required to achieve convergence is also quite large. With the inclusion of bipartite graph matching, the MSE for the matched surfaces turns out to be quite small and the convergence of the ICP algorithm is achieved in far fewer iterations. The constraint of *maximum matching* is enforced because we are trying to match all the points in the sample set to some points in the model set. On the other hand, we also need to incorporate the constraint of *minimum weight* because we are trying to match the points of the two vertex sets in such a fashion that the sum of the weights for all the pairs is minimized.

### 5.3.2 Solving for an Optimal Quaternion derived from a Real Symmetric Matrix

As mentioned in the step (e) of the ICP algorithm, the unit eigenvector that corresponds to the maximum eigenvalue of Q (a 4x4 real symmetric matrix) is the optimal rotation quaternion $q_R$. In order to compute this eigenvector, the well-known Jacobi Transformation [26] is applied to the Q matrix. For a real symmetric matrix, the eigenvectors are real and orthonormal. The Jacobi Transformation basically consists of a sequence of orthogonal similarity transformations.

### 5.3.3 Quaternion Representation of the Rotation Matrix

Quaternions are a generalization of complex numbers and can be used to succinctly represent 2D and 3D rotations. Though a rotation in $\mathbf{R}^3$ can be represented by a 3x3 orthonormal matrix or by means of a rotation angle and an axis of rotation, the third option of representing it by means of quaternions is very useful in the image registration context. A quaternion rotation is represented by a 4-dimensional vector $q_R$ such that $q_R = [q_0\, q_1\, q_2\, q_3]$. Note that $q_R$ interestingly has a scalar portion $q_0$ and a vector portion consisting of $q_1, q_2, q_3$. Formally we can express $q_R$ as:

$q_R = q_0 + q_1\,\mathbf{i} + q_2\,\mathbf{j} + q_3\,\mathbf{k}$

where **i, j, k** are the three unit vectors along x, y and z axes, respectively.

Rotations are represented by a special class of quaternions having the property:

$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$

i.e. rotations are exactly the *unit quaternion* that may be seen as points on the surface of a unit sphere in $\mathbf{R}^4$. Rotation by an angle φ about an unit vector $(q_1, q_2, q_3)$ can be represented by a quaternion $q_R$ as follows:

$q_R = \cos(\varphi/2) + q_1 \sin(\varphi/2)\,\mathbf{i} + q_2 \sin(\varphi/2)\,\mathbf{j} + q_3 \sin(\varphi/2)\,\mathbf{k}$

The corresponding rotation matrix R is given by:

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}$$

This is important because the rotation matrix cannot be directly represented in the ICP algorithm. In fact the rotation matrix is only computed from a quaternion which is essentially the largest eigenvalue of a 4x4 real symmetric matrix.

# CHAPTER 6

# THE RANSAC AND DARCES ALGORITHMS

## 6.1 Introduction

The ICP algorithm is a popular and widely used iterative method to register two data sets, but it is easily misled by outliers and requires a good initial estimation to guarantee that the correct solution is found fairly rapidly. An alterative approach to the registration problem was investigated in this thesis. The CT images displayed in Chapter 4 show that some of the fracture surfaces of the broken fragments have no prominent features, and hence a registration method that does not require feature extraction is needed. The Data-Aligned Rigidity-Constrained Exhaustive Search (DARCES) algorithm [8] is the other method used to solve the 3D transformation by matching the two opposable fracture surfaces for a set of CT slices. The DARCES algorithm can solve the partially-overlapping 3D registration problem efficiently and reliably without requiring any feature extraction and any initial estimation of the rigid-body transformation. Moreover, it exhibits robust performance even when outliers are present.

The bone fracture under traumatic impact will break into a few bone fragments. For each fragment, we have a 3D fracture surface at the fracture site. Two fracture surfaces are considered for surface matching each time, no matter how many fragments there are. After the fragments are identified, the surgeon can help to decide which two fragments are candidates for surface matching in order to avoid the random matching between any two surfaces. Although it is possible to lose some data at the fracture site, it is also possible to introduce some noisy data during the data acquisition process. Hence, it is reasonable to treat the two target surfaces as two partially-overlapping surfaces to which the DARCES method can be applied.

For the two surfaces to be matched, the underlying surface data can be retrieved as described in Chapter 4. The two surfaces are termed as the sample surface and model surface, or sample shape and model shape respectively. The goal of surface matching is to align the sample shape to the model shape. The DARCES algorithm chooses a number of control points on the sample surface and slides these points on the model surface to find a solution with the largest number of overlapping surface points. The DARCES algorithm with three control points is used in this research to form a regular triangle, and an improvised Random Sample Consensus (RANSAC) algorithm [9] is used for the triangle selection. Note that at least three pairs of corresponding points are needed to determine a unique 3D rigid body transformation.

In theory, there is an exact transformation to match each point on the sample surface to the model surface since they were a complete piece before the fracture. This suggests that this is a totally-overlapping registration problem, that is, each point on the sample surface should find its matching point on the model surface. This would be true if the mandible breaks into only two pieces such as in our case. Actually, it is not clear whether all of the data points represent true surface data even in this case, because some data from the inner bone that is away from the surface may appear to be surface data when there is loss of bone at the fracture site. Also, the binarization process and surface data retrieval process may misclassify some non-surface data as surface data. All overlapping data points on the two surfaces to be matched form a reference data set for the control point selection and transformation evaluation. In this case, the reference data set is the sample data set. However, the CT slices of the broken mandible (see the binary images in Chapter 7, Fig. 7.1) show that the left fragment has a much smaller surface contour in the earlier CT slices that becomes progressively larger towards the end of the CT image stack, but the right fragment exhibits the reverse behavior. It is not clear whether the slices chosen for reconstruction have included all surface data since displacement along the z-axis is possible between the fragments. In order to have better control over the surface matching and to take into account the possible errors introduced in the input data near the two end points of the contour in this case, a separate reference data set is generated from the sample surface data to select the control points. However, the whole sample data set is used for transformation evaluation since the two surfaces should have a complete match. For the sample surface data on each CT slice, 5-10% (the actual percentage depends on data reliability) of the contour data from the beginning and end is excluded, and the rest of the contour data is added to the reference set. Thus, the reference data set has about 80-90% of the original data from the sample surface. Thus, most data in reference data set are accurate, and give several choices for the selection of the control points. Furthermore, the RANSAC algorithm will generate many control point triangles with different sizes to give the best solution.

## 6.2 The DARCES Algorithm Implementation

Three control points are selected from the reference data set. We call the three points the primary control point $S_p$, the secondary control point $S_s$, and the auxiliary control points $S_a$. Their corresponding points in the model data set are called $M_p$, $M_s$, $M_a$, respectively. In this case, the three control points form an approximately equilateral triangle, namely, the control triangle (see Fig. 6.1 (a)), although strictly, the equilateral triangle condition is not required.

Suppose such a control triangle is already available. The working of the DARCES algorithm is somewhat like sliding of the control points on the model surface to find the corresponding model points $M_p$, $M_s$, and $M_a$. Since the only information available is the edge lengths of the triangle, there are many possible similar triangles on the model surface that match the control triangle. A systematic, efficient method is used to find all such triangles.

The DARCES algorithm searches the whole model data set for the matching of the control triangle on the sample surface. Each point on the model data set is a potential corresponding point for the primary control point $S_p$. Suppose point $M_p$ in the model data set is such a matching point for point $S_p$. Then, the next step is to find the candidates corresponding to the secondary control point $S_s$. Assume that the distance between $S_p$ and $S_s$ is $d_{ps}$, then the corresponding distance between $M_p$ and $M_s$ should also be $d_{ps}$, which means that $M_s$ must lie on the surface of a sphere $C_s$ whose center is $M_p$ and the radius is $d_{ps}$ (Fig. 6.1 (b)); that is $C_s = \{P = (x, y, z) \mid ||P - M_p|| = d_{ps}\}$. Once points $M_p$ and $M_s$ are hypothesized to correspond to points $S_p$ and $S_s$ respectively, then the same strategy can be followed to find the candidates corresponding to the auxiliary control point $S_a$. The match candidates for $M_a$ should lie within the intersection of two surfaces of spheres $C_{pa}$ and $C_{sa}$, where $C_{pa} = \{P = (x, y, z) \mid || P - M_p|| = d_{pa}$, where $d_{pa}$ is the distance between $S_p$ and $S_a\}$, $C_{sa} = \{ P = (x, y, z) \mid || P - M_s|| = d_{sa}$, where $d_{sa}$ is the distance between $S_s$ and $S_a\}$. Alternatively, $S_a$ can be found as follows: do the orthogonal projection of $S_a$ onto the line segment $\mathbf{S_pS_s}$ (Fig. 6.2 (a)), which is point $S_{q,}$ and $d_{qa}$ is the distance between $S_q$ and $S_a$. The candidates of $M_a$ must lie on the 3D circle $C_a$ centered at point $M_q$ with radius $d_{qa}$, where $M_q$ corresponds to $S_q$ on the model surface (Fig. 6.2 (b)). That is, $C_a = \{P = (x, y, z) \mid ||P - M_q|| = d_{qa}$ and $M_aM_q$ is perpendicular to $\mathbf{M_pM_s}\}$.

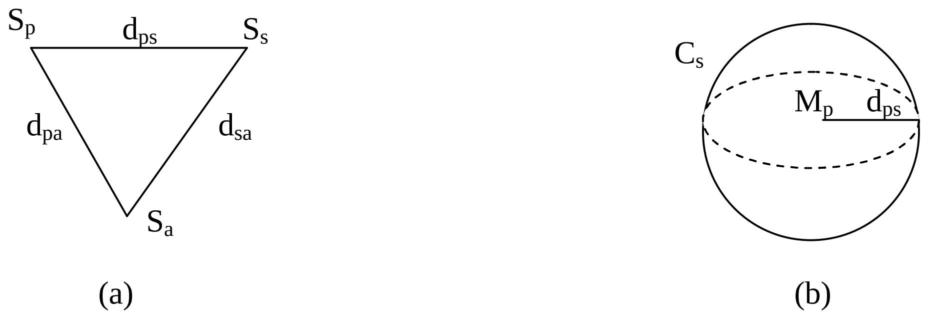


Fig. 6.1 (a) The triangle formed by three control points chosen from the sample surface.

(b) The search region of control point $S_s$ in the model surface is the surface of a sphere.

The data used are discrete pixels, and they are converted into metric measurements based on the image resolution. To simplify the search process, the DARCES algorithm exploits the fact that a range image can be treated as the projection of the 3D points onto an index plane. A sphere will become a circle after such projection. To further simplify the implementation, the smallest square containing the circle is used to replace the original circle for searching (Fig. 6.3). In this case, the y-z plane is chosen to be the index plane. To search for candidates of $M_s$, the user first collects all model data points whose projections onto the index plane fall into the square search region, that is, the square area containing the 2D points defined by the y, z-coordinate values of the model points. For each of these candidate points, we compute its distance to point $M_p$. If the distance is approximately $d_{ps}$, then this point is recorded as the matching candidate of $S_s$. A 3D circle usually will become an ellipse after projection (Fig. 6.4 (a)). It is easier to search for the intersection of the corresponding square regions of spheres $C_{pa}$ and $C_{sa}$ projected onto the y-z plane, where $C_{pa}$ and $C_{sa}$ are defined above (Fig. 6.4 (b)).

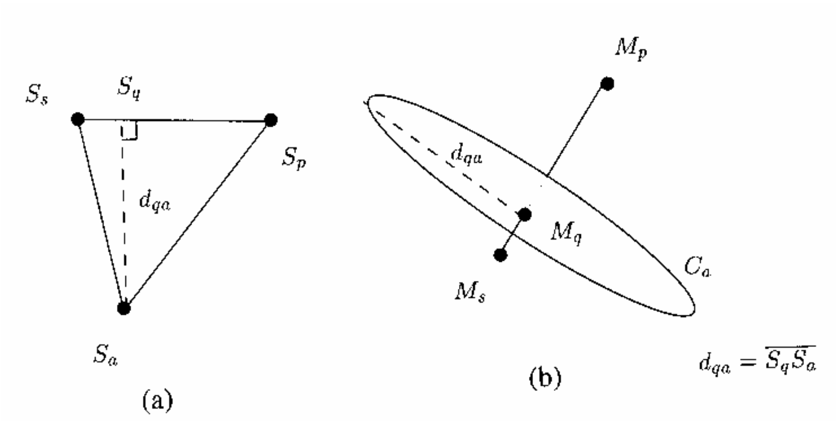


Fig. 6.2 (a) The projection of $S_a$ onto line segment $\mathbf{S_pS_s}$ intersecting at point $S_q$.

(b) The search region for point $S_a$ in the model data set is restricted to the contour of a 3D circle.

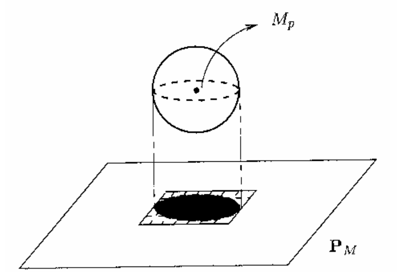


Fig. 6.3 The projection of a sphere onto the index plane forms a 2D circle, but a bounding square is used as the search region for $M_s$ corresponding to the control point $S_S$.

The points $M_p$, $M_s$, and $M_a$ form a triangle, on the model surface, and its computed similarity to the control triangle on the sample surface depends on the predefined threshold. The smaller the threshold value, the more similar the two triangles. However, the threshold value can not be smaller than the voxel resolution. The threshold value does not affect the accuracy of the algorithm, but definitely affects the speed. With a smaller threshold value, the program will jump out earlier and start a new search once the distance difference is above the threshold. Moreover, the smaller the threshold value, the fewer the number of similar triangles that are found, which speeds up the search process. With a larger threshold value, more similar triangles are generated and the subsequent testing of reference points consumes more time.

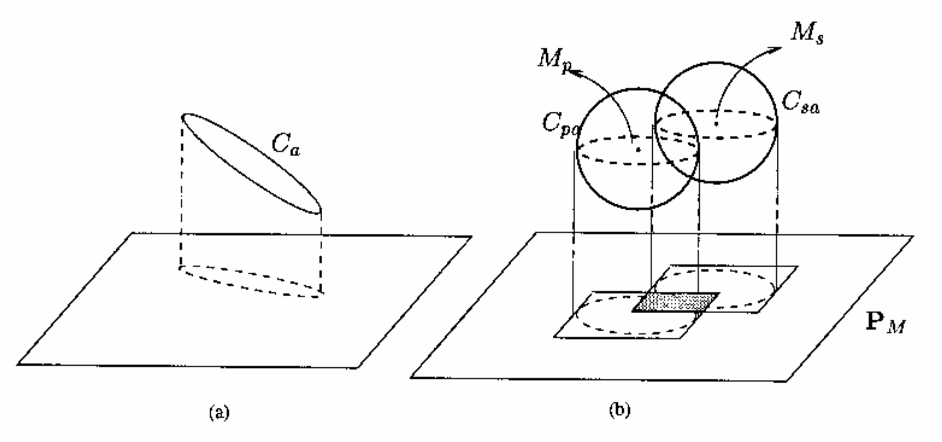


Fig. 6.4 (a) The 2D ellipse of projection of the 3D circle onto the index plane. (b) The search for $M_a$ is performed in the intersection of the two square regions.

The points $M_p$, $M_s$, and $M_a$ form a triangle, on the model surface, and its computed similarity to the control triangle on the sample surface depends on the predefined threshold. The smaller the threshold value, the more similar the two triangles. However, the threshold value can not be smaller than the voxel resolution. The threshold value does not affect the accuracy of the algorithm, but definitely affects the speed. With a smaller threshold value, the program will jump out earlier and start a new search once the distance difference is above the threshold. Moreover, the smaller the threshold value, the fewer the number of similar triangles that are found, which speeds up the search process. With a larger threshold value, more similar triangles are generated and the subsequent testing of reference points consumes more time.

Once a similar triangle on the model surface is found, a unique rigid-body transformation, namely $T_c$, can be determined by using the three pairs of corresponding points: ($S_p$, $M_p$), ($S_s$, $M_s$), and ($S_a$, $M_a$). The reference points are used to evaluate the transformation $T_c$. In this case, the whole sample data is used instead of the reference data set as described before. The transformation $T_c$ is then applied to each sample point $S_i$ other than the three control points resulting in a new point $M_i^{'}$, and the distance between $M_i^{'}$ and the model surface is measured. If the distance is smaller than the alignment threshold, then the point $S_i$ is said to be successfully aligned on the model surface. Three alignment threshold values are used to evaluate the alignment along the x, y, and z-directions based on to the voxel resolution. Along each of the x, y, and z-directions, the threshold values are simply chosen to be equal to the respective voxel resolutions. The number of such successful alignments is accumulated, and is called the overlapping number $N_o$ for transformation $T_c$. This evaluation process is repeated for each similar triangle on the model surface corresponding to the control triangle, and an overlapping number for each pair of triangles is computed. Finally, the rigid-transformation with the largest overlapping number is selected as the solution of the registration task under the noiseless case.

Two methods are experimented with to determine the transformation given three pairs of points with correspondence: ($S_p$, $M_p$), ($S_s$, $M_s$), and ($S_a$, $M_a$). The first is a least-squares-fitting method involving the Singular Value Decomposition (SVD) algorithm to solve the transformation proposed by Arun et al. [13]. The other approach, proposed by Paul [19], is based on determining the axis of rotation and the rotation angle about this axis. For convenience, we refer to these as the SVD approach and Paul’s approach, respectively.

## 6.3 The RANSAC-Based DARCES Algorithm

Proper control triangle selection is very important in the DARCES algorithm. If a smaller triangle is used for searching in the DARCES algorithm, then a faster search speed is achieved. However, if the triangle is too small, then the computed rigid-body transformation is very sensitive to noise. Unfortunately, the proper size for the control triangle is not known before hand. When noisy data are present, the problem is even more complex if one noisy point is unfortunately included in the control points or if the corresponding point for a given control point is a noisy point. To get a better solution, an improvised RANSAC-based sampling method is used along with the DARCES algorithm to solve the triangle selection problem.

In order to generate the most proper control triangles, different triangle sizes in terms of their edge lengths are experimented with. The range of the edge length is decided according to the properties of the data set. In this research, the maximum length, say, $L_{max}$, between two points in the sample data set is determined. These two data points should be located on the boundary of the sample surface. Then, the differences in the x, y, and z-values among the data set, termed as ?x, ?y and ?z respectively, are identified. The maximum edge length $d_{max}$ = min (?x, ?y and ?z) since an equilateral triangle is used for convenience. The edge length d can have values ranging from some small percentage of $L_{max}$ to $d_{max}$ increased by a small step value in each iteration. For each edge length value, instead of randomly generating one triangle or a number of triangles, all triangles satisfied with the edge length condition in the reference data set are produced. The process is similar to finding the corresponding points on the model surface for the chosen control points.

The optimal answer to the registration problem is deemed to the transformation with the largest overlapping number $N_0$ among all the control triangles. The theoretical support for the above conjecture is that the correct transformation should be able to align the most sample points on the model surface.

The pseudo-code for the RANSAC-based DARCES algorithm using three control points is listed below:

```
for triangle edge length d from $d_{max}$ to $d_{max}$ with step size ?d {
 for each triangle with control points $S_p$, $S_s$, and $S_a$ {
   for $M_p$: all points in model data set {
     for $M_s$: all points whose projection is contained in the projected search region of sphere $C_s$ {
       if the distance between $M_p$ and $M_s$ is approximate to $d_{ps}$ {
         for $M_a$: all points whose projection falls into the projection area of 3D circle $C_a$ {
           if the distance between $M_a$ and $M_q$ is approximate to $d_{qa}$ {
             compute the transformation $T_c$ from ($S_p$, $M_p$), ($S_s$, $M_s$), and ($S_a$, $M_a$)
             for each reference point $S_r$ other than $S_p$, $S_a$, and $S_a$ {
               compute $M_{r\ =}\ T_cS_r$, if the distance between $M_r$ and the model data set is smaller than the
               alignment threshold, then count $M_r$ successfully aligned
             }
             record the overlapping number for this model triangle into $N_{Tc}$
           }
         }
       }
     }
   }
 }
 record the largest overlapping number for this control triangle into $N_0$
}
output the transformation $T_c$ with the largest overlapping number $N_0$ among all the control triangles
```

# CHAPTER 7

# EXPERIMENTAL RESULTS AND DISCUSSION

The images showing the reconstruction of the broken mandible by the ICP and DARCES algorithms are shown in Fig. 7.1. The binary images of the broken mandible resulting from the source CT images used in the reconstruction process are also displayed for comparison. Both the ICP and DARCES algorithms showed good surface matching results for the broken mandible fragments. The ICP algorithm achieved an MSE of 0.91 $\text{mm}^2$ for the corresponding pairs of points derived from the sample and model data sets. The DARCES algorithm yielded a large overlapping number between the sample and model surface shapes.

The ICP algorithm converges to a locally minimal MSE within 5 iterations using an initial estimation of [1 0 0 0 0 0 0] for the rotation quaternion and translation vector. The MSE is $0.91\text{mm}^2$, but the bipartite graph matching, the most computational expensive part in this algorithm, takes a long time to compute the closest set.

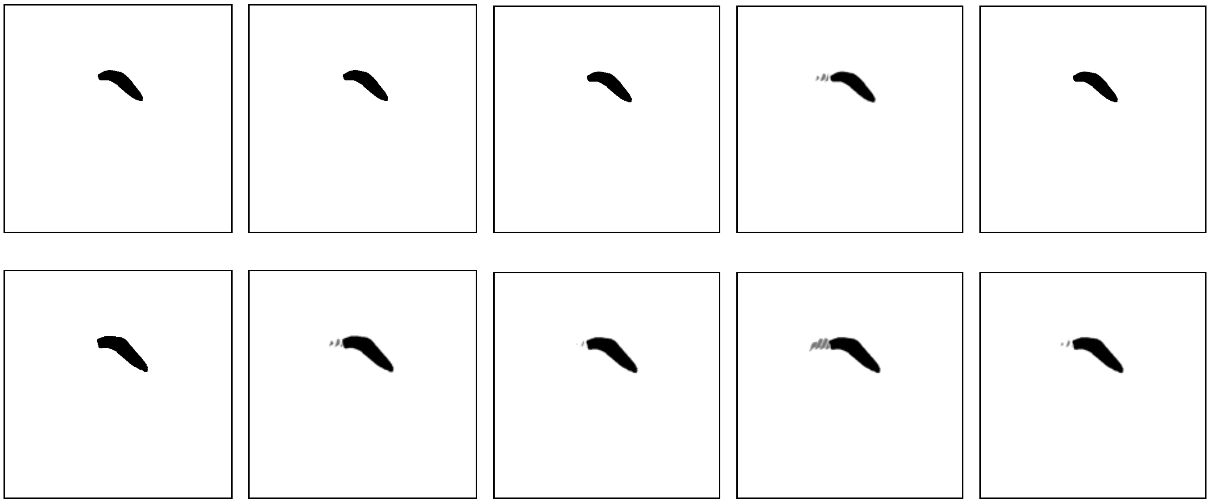
The reference data set in the DARCES algorithm directly affects the selection of the control triangles and the final result. Initially, the algorithm was tested on a small part of the mandible data, but the reconstruction result for that part was not satisfactory; the two fracture surfaces were joined, but the tail part of the sample fragment was skewed out of the normal range. With too small a reference data set it is hard to select a control triangle with proper size and exclude the noisy points. On the other hand, the overlapping number generated from such a small data set may not correctly reflect the true transformation status, that is, to say the transformation corresponding to the largest overlapping number may not be the solution to the registration problem. Two reference data sets each containing a different number candidate points for control point selection are used test the DARCES algorithm. The two sets are composed of the middle 80% and 90% of the contour data on each slice respectively, that is, they contain the most central 80% and 90% of the sample fragment surface data respectively. The SVD method and Paul's method are each chosen to separately solve the transformation problem in the DARCES algorithm for each reference data set once the correspondence between pairs of control points is established. The experimental results showed the two reference data sets were large enough to produce satisfactory results. The reconstructed pictures did not show any apparent difference between the two reference data sets for both, the SVD method and Paul's method. Actually, the SVD method found exactly the same best control triangle and the matching triangle on the model surface, and hence the same transformation for the reconstruction problem in the two reference data sets. The transformation resulting from Paul's method was a little different from the one resulting from the SVD method. The overlapping number for each case is listed in Table 7.1.

The computation of the basic DARCES algorithm was very fast compared to other registration algorithms. It took about 25 seconds when it was used with the SVD algorithm, and approximately 50 seconds when is was used with Paul's method, to exhaustively search the model space for the chosen control triangle, depending on the size of the triangle in this case. The speed can be further improved by relaxing the exhaustive search constraints. Paul's method has no mechanism to decide whether the resulting rotation or its transpose is the correct answer to given problem, hence both the rotation and its transpose have to be tested on the reference data set, which consumes a lot of time.

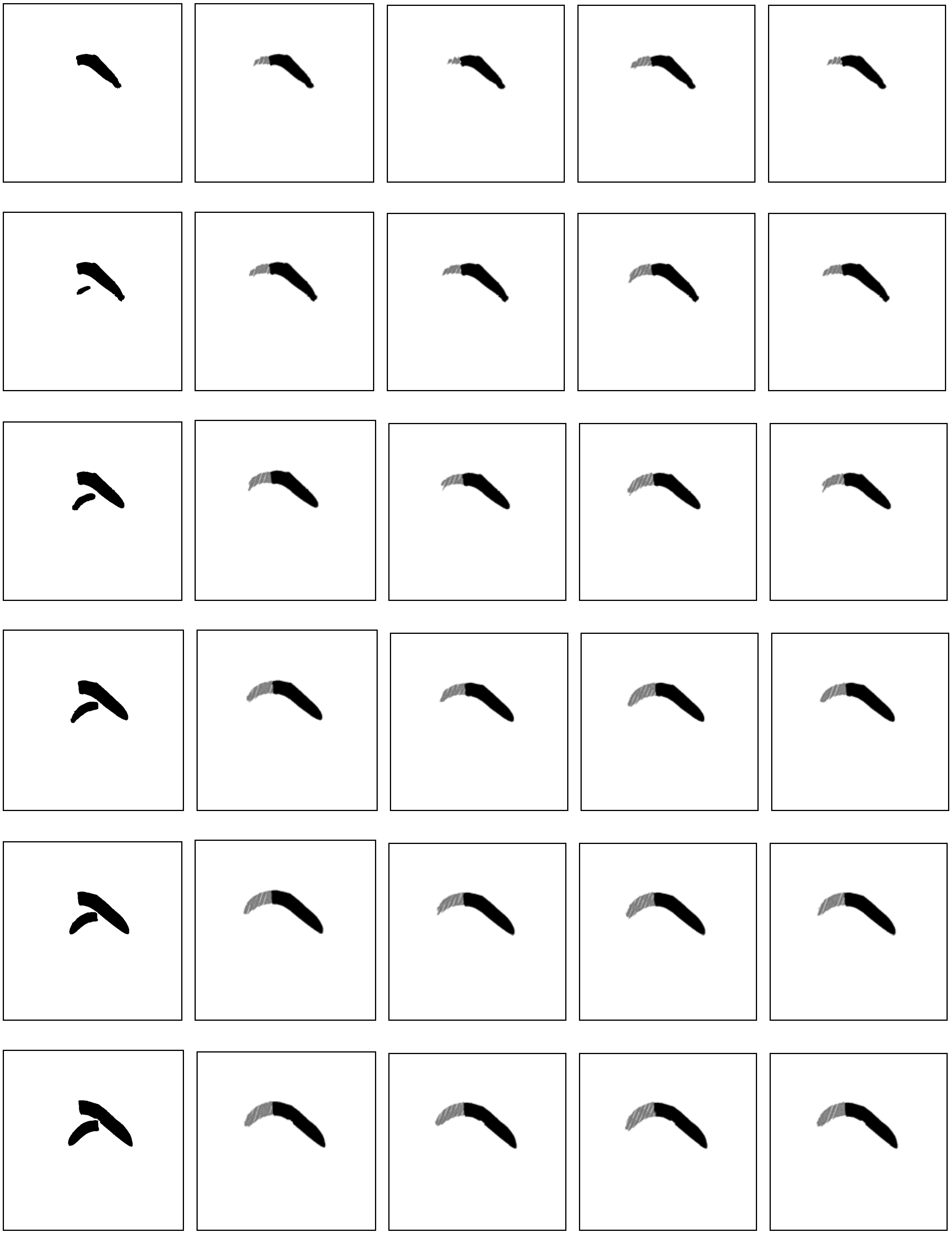
The ICP algorithm and the DARCES algorithm have different error metrics (namely MSE in the case of the ICP algorithm and alignment threshold in the case of the DARCES algorithm) to measure their performances. In order to compare their performances using the same error metric namely the MSE, bipartite graph matching is applied to the DARCES algorithm output to establish the correspondence between the calculated model data set (resulting from the application of the transformation computed by the DARCES algorithm to the sample data set) and the actual model data set. The result illustrates that the two metrics are comparable and have the same level of efficiency to measure performance, and that a good result under one metric will also be good under another metric. The error measurement results are shown in Table 7.2.

Both, the ICP and DARCES algorithms have their own advantages. The DARCES algorithm is an exhaustive search algorithm with no initial transformation estimation requirement and has good performance even when the outliers are present in the input data. The ICP is an iterative algorithm, which reduces the MSE gradually at each iteration, but is sensitive to the presence of outliers and requires good initial estimation of the transformation to get the final result.

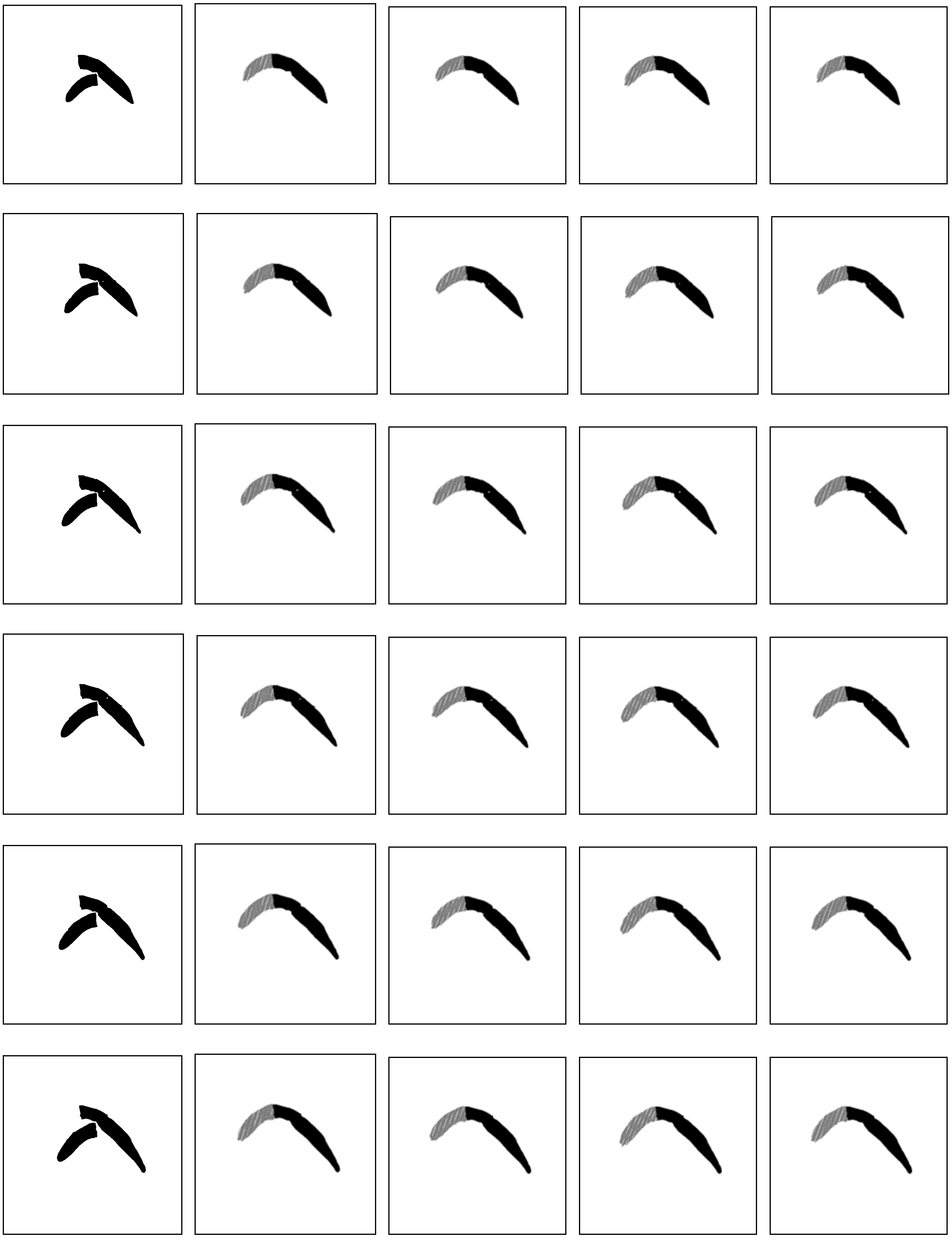
It is expected that the combination of the two algorithms will produce some interesting results. The ICP algorithm will get a very good starting point by using the resulting transformation from the DARCES algorithm. Moreover, the pairs of matched points from the DARCES algorithm speeds up the Bipartite Graph Matching algorithm used in the ICP algorithm to find the closest set in two ways. Firstly, the DARCES output implies potentially correct correspondences between the two data sets, since the sample point is counted as successfully aligned only if the distance between the point and the model surface is below the alignment threshold. This implicit correspondence is close to the true correspondence and it is easier for bipartite graph matching algorithm to establish the correct correspondence between the two data sets without the effect of outliers. Secondly, by removing the outlier data points, the DARCES output reduces the cardinality of the two original data sets to be matched. Therefore, the new bipartite graph has fewer nodes and edges, and the subsequent Maximum Cardinality Minimum Weight Matching algorithm has a reduced computational complexity. Thus the synergetic combination of the DARCES and ICP algorithms where the output of the DARCES algorithm is fed as input to the ICP algorithm, results in an improved surface matching algorithm where the reconstruction accuracy and execution time are both improved. The reconstructed mandible images are displayed in Fig. 7.1. The MSE resulting from the ICP, DARCES and the combination of the ICP and DARCES algorithms is tabulated in Table 7.2. The maximum allowed bone gap for the mandible is 0.5 mm after reconstruction, and the MSE from the combination of the DARCES and ICP algorithms is in this range.



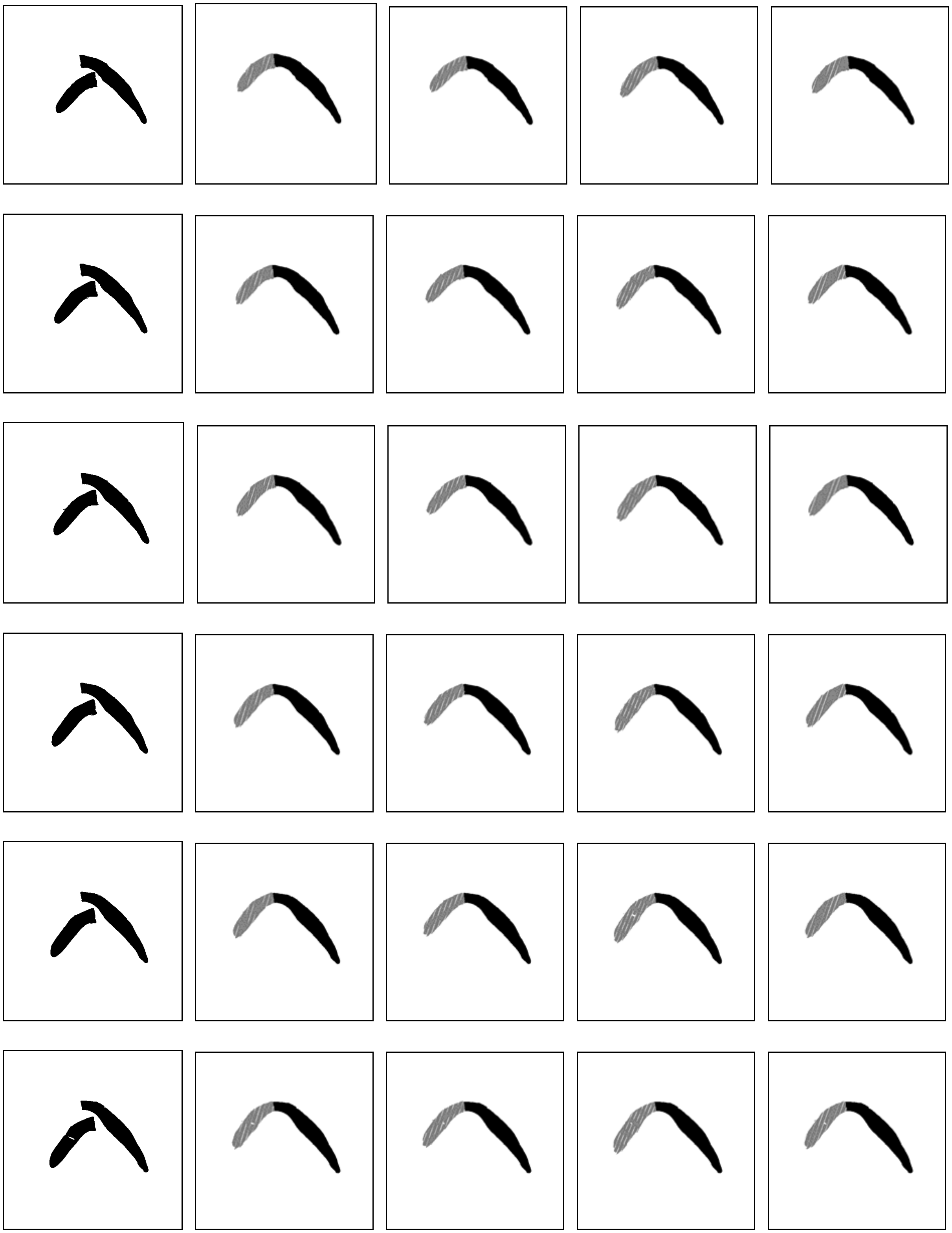
(continued on next page)



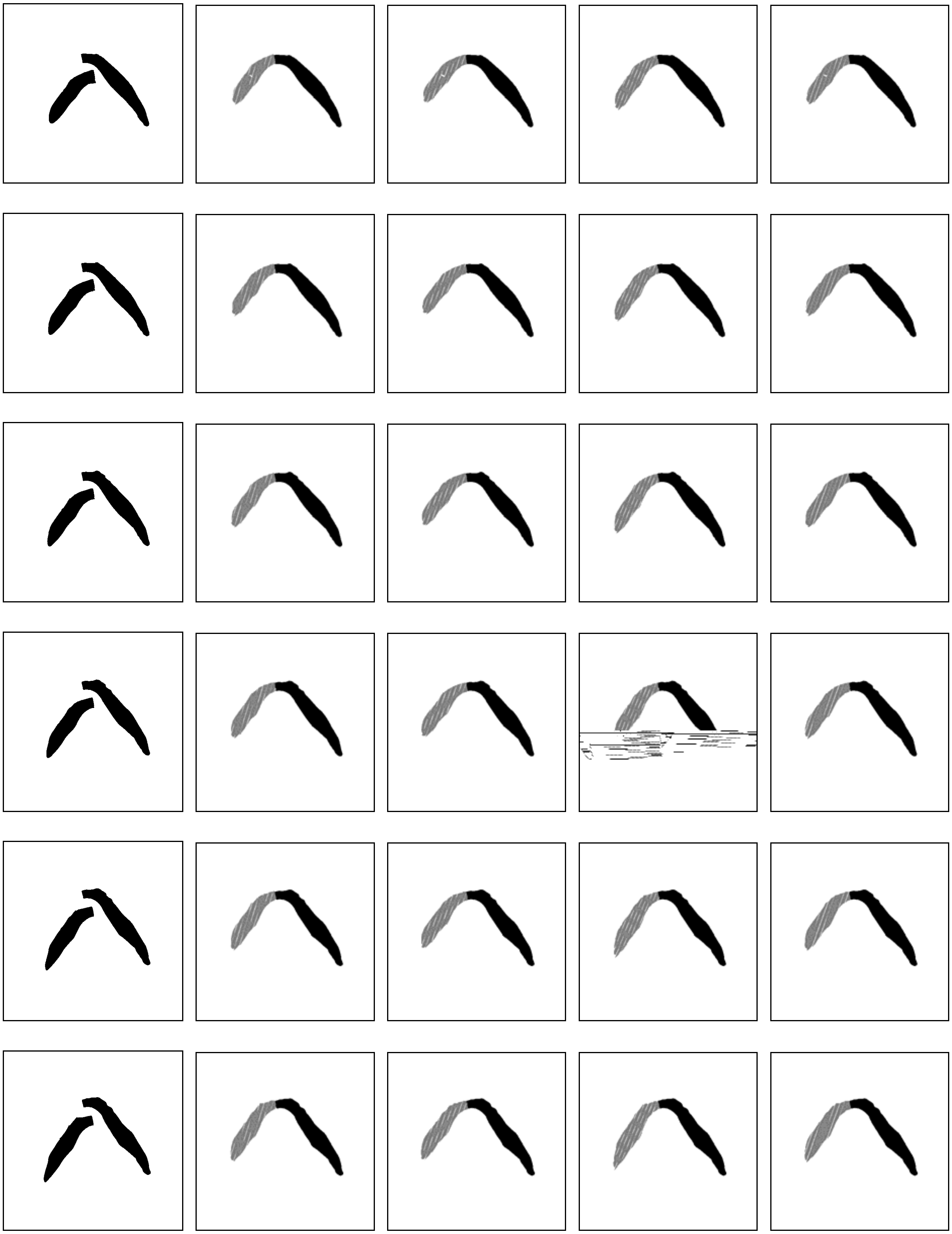
(continued on next page)



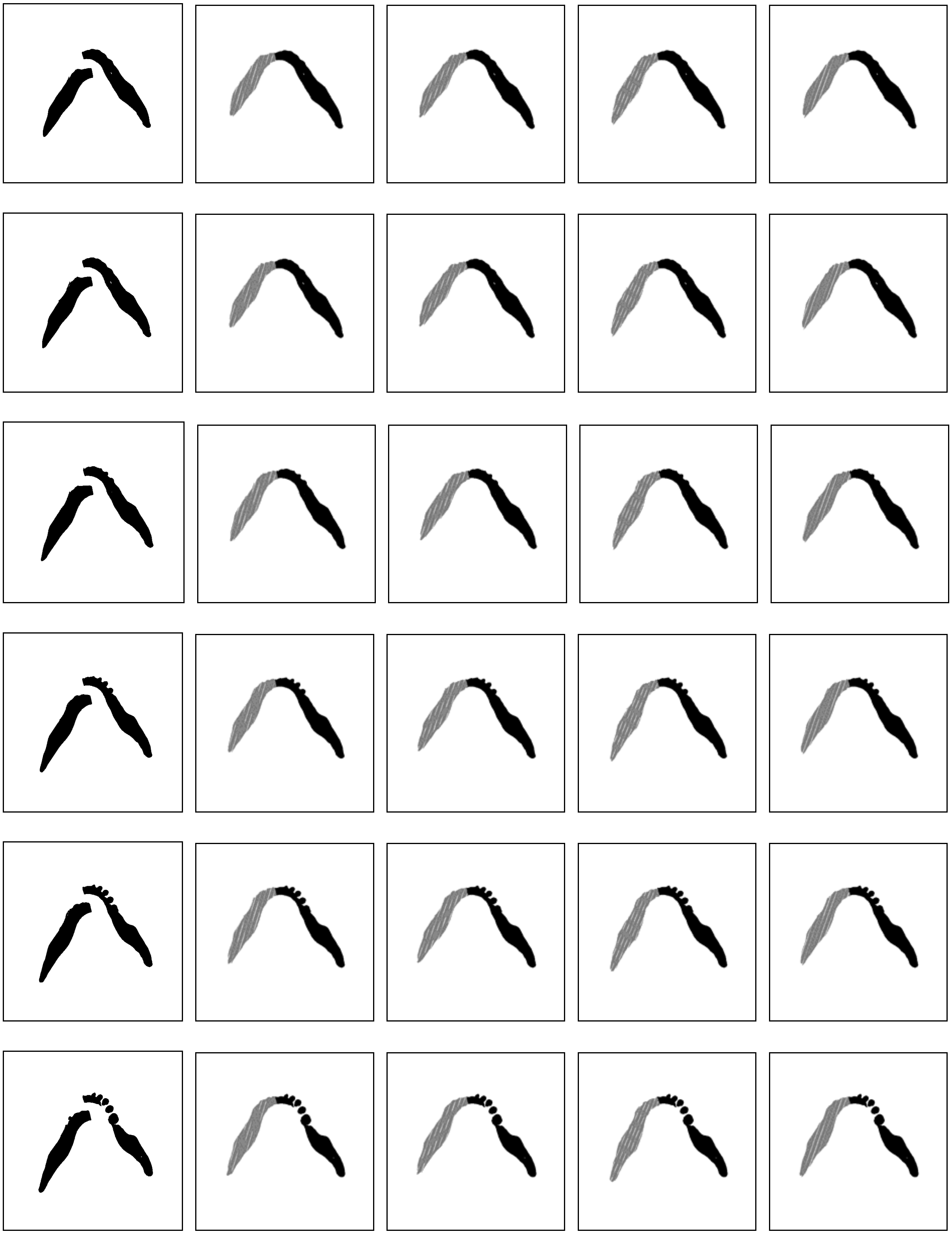
(continued on next page)



(continued on next page)



(continued on next page)



(continued on next page)

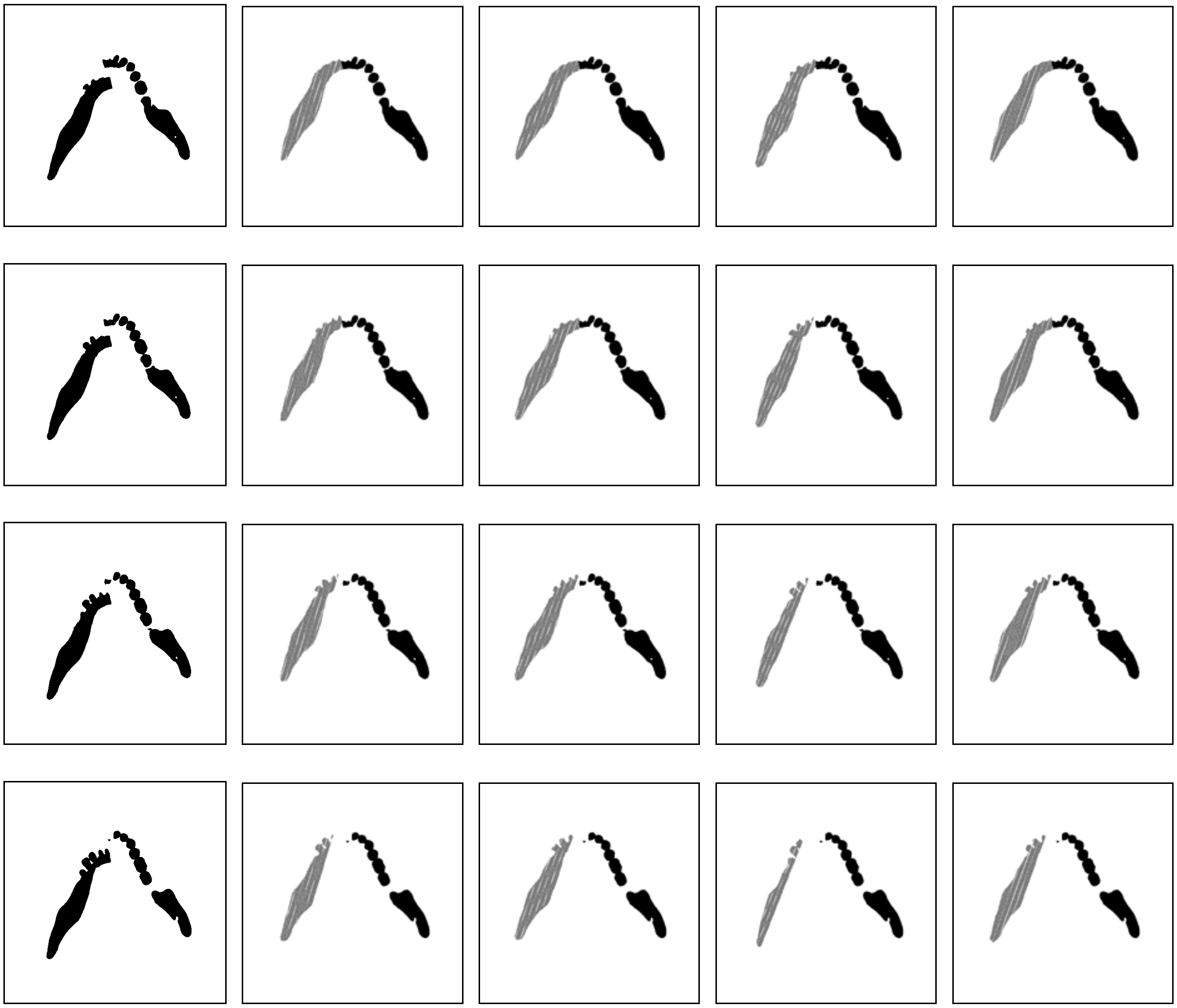


Fig. 7.1 The images of the broken mandible after reconstruction. The first column contains all the binary images of 2D CT slices in the order acquisition, representing the broken mandible fragments to be re-aligned. The right fragment is fixed. The second column represents the surface matching result of the ICP algorithm. The third and fourth columns represent the surface matching results of the DARCES algorithm (with 80% sample data), using the SVD method and Paul's method to compute the 3D rigid-body transformation, respectively. The fifth column represents the reconstruction result of the combined DARCES-ICP algorithm.

Table 7.1 The result of the DARCES algorithm: 80%, 90% are the percentages of the sample data set used to create the reference data set; SVD and Paul's are the two algorithms used in the DARCES algorithm to compute transformation, respectively; $N_{tc}$ is the largest overlapping number corresponding to the solution.

| | 80% of sample | | 90% of sample | |
|---|---|---|---|---|
| | SVD | Paul's | SVD | Paul's |
| $N_{tc}$ | 877 | 869 | 877 | 911 |

Table 7.2 MSE of the ICP, DARCES and DARCES-ICP combined algorithms

| Algorithm | ICP | DARCES | DARCES-ICP |
|---|---|---|---|
| MSE ($mm^2$) | 0.91 | 0.33 | 0.25 |

# CHAPTER 8

# CONCLUSIONS AND FUTURE WORK

## 8.1 Conclusions

The first version of the front-end GUI design has been tested by the plastic surgeons at the Medical College of Georgia and has been approved. The fired format and functions still need further evaluation from the end-users. The fragments of the broken mandible are realigned via rotation and translation. The ICP algorithm and the DARCES algorithm are applied to reconstruct the broken mandible by matching the fracture surfaces. The ICP algorithm is an iterative method that reduces the MSE and refines the solution gradually. However, it is easily affected by noise and outliers, prone to get trapped in a local minimum and the computation to determine the closest set is intensive. The DARCES algorithm uses an exhaustive search to find the solution with the largest overlap number between the two overlapping surfaces. It has good performance even when there are outliers present. The computation is fast and can be even faster if the exhaustive search restriction is relaxed. However, unlike the ICP algorithm, the DARCES has no embedded mechanism such as iteration for solution refinement. The combination of the DARCES algorithm and the ICP algorithm takes advantage of both algorithms and results in a better performance. The three algorithms, the ICP, DARCES and DARCES-ICP, can be ranked in that order based on the MSE for surface matching. The images of the reconstructed mandible indicate that the fragment surfaces have been well matched.

## 8.2 Future Work

Neither the ICP algorithm nor the DARCES algorithm gives a perfect solution to the mandible reconstruction problem. Both algorithms are data driven and focus on finding an optimal local solution for the two fracture surfaces based on available surface data. However, a transformation that also preserves the global 3D shape of the human mandible is desirable. Global constraints, such as the symmetry of the mandible structure, will be used to guide the search for a globally optimal transformation. Future enhancements will incorporate a model-driven search that will achieve the matching of the fracture surfaces to a higher degree of accuracy (as determined by the MSE) as well as ensure the preservation of the global 3D shape of the human mandible.

# REFERENCES

1. Pashley, D.H., J.L. Borke and J.C. Yu. Biomechanics and craniofacial morphogenesis in Craniofacial surgery, *Science and Surgical Technique*. (Lin, K.Y., Ogle R.C., and Jane, J.A. eds.), W. B. Saunders Co., Philadelphia, pp. 84-100, 2002.
2. Michael J. Mack. Minimally Invasive and Robotic Surgery, *Journal of the American Medical Association*, Vol. 285, No. 5, pp. 568-572, 2001
3. Kobayashi, M., T. Fugino, T. Kanejo and H. Chiyokura. The virtual reality technique in simulation surgery – a mandibular fracture model. *Keio Jour. Medicine*, Vol. 50, Suppl. 2, pp. 129-134, March, 2001
4. Kurihara, T. The fourth dimension in simulation surgery for craniofacial surgical procedures. *Keio Jour. Medicine*, Vol. 50, Suppl. 2, pp. 155-165, March, 2001.
5. Nakajima, H., T. Kaneko, T. Kurihara and T. Fujino. Craniofacial surgical simulation in the 3-dimensional CT SurgiPlan system. *Keio Jour. Medicine*, Vol. 50, Supl. 2, pp. 95-102, March, 2001.
6. Siessegger, M., B.T. Schneider, R.A. Mischkowski, F. Lazar, B. Krug, B. Klesper and J. E. Zoller. Use of an image-guied navigation system in dental implant surgery in anatomically complex operation sites. *Jour. Cranio-Maxillo-Facial Surgery*, Vol. 29, No. 5, pp. 276-281, Oct., 2001.
7. Eldeeb, H. N., Boubekri, S. Asfour, T. Khalil and A. Finnieston. Design of Thoracolumbosacral orthosis (TLSO) braces using CT/MR. *Jour. Computer-Assisted Tomography*, Vol. 25, No. 6, pp. 963-970, Nov/Dec. 2001.

8. C.S. Chen, Y.P. Hung, and J.B. Cheng. RANSAC-Based DARCES: A new approach to fast automatic registration of partially overlapping range images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 11, pp. 1229-1234, Nov. 1999.
9. M. A. Fischler and R. C. Bolles. Random Sample Consensus, A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Communications of ACM*, Vol. 24, No. 6, pp. 381-395, 1981.
10. http://rsb.info.nih.gov/ij
11. Y. Rogers, H. Sharp and J. Preece, *Interaction Design beyond Human-Computer Interaction*. John Wiley & Sons. 2002, ch 7.
12. R. M. Harlick and L. G. Shapiro, *Computer and Robot Vision*, Vol.II. Addison-Wesley, pp. 160-167, 1993.
13. K. S. Arun, T. S. Huang, and S. D. Blostein. Least-Squares Fitting of Two 3-D Point Sets, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 5, pp. 698-700, Sep., 1987.
14. P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, pp. 239-255, Feb. 1992.
15. N. Christofides. *Graph Theory An Algorithmic Approach*, Academic Press, 1975, pp. 372-379.
16. W. Kim and A. C. Kak. 3-D Object Recognition Using Bipartite Matching Embedded in Discrete Relaxation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 3, pp. 224-251, 1991.

17. H. W. Kuhn. The Hungarian Method for the Assignment Problem, *Nav. Res. Log. Quart.* 2, pp.83 – 97, 1955.

18. R. P. Paul. *Robot Manipulators: Mathematics, Programming and Control*, MIT Press, Cambridge, MA, 1981.

19. G. Lohmann, *Volumetric Image Analysis*, Wiley and Teubner, 1998, pp. 210-211

20. P. Saint-Marc and J. S. Chen. Adaptive Smoothing: A General Tool for Early Vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 6, pp.514 – 529, Jun. 1991.

21. J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*, The Macmillan Press LTD, 1976, Chapter 5.

22. L. Lovasz. Three Short Proofs in Graph Theory, *J. Combinatorial Theory*, B, 19, pp. 111-113, 1975.

23. J. Munkres. Algorithms for the Assignment and Transportation Problems*, J. Soc. Indust. Appl. Math.*, 5, pp. 32 – 38, 1957.

24. H. M. Hamdan, E. E. Hemayed and A. A. Farag. A Fast 3D Object Reconstruction Using Trinocular Vision and Structured Light, *Proc. of SPIE Conf.*, 3522, pp. 444 – 454, Nov. 1998.

25. Y. Q. Cheng, V. Wu, R. T. Collins, A. R. Hanson, E. M. Riseman. Maximum-weight Bipartite Matching Technique and Its Applications in Image Feature Matching, *SPIE Conference on Visual Communication and Image Processing*, 1996.

26. W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling. *Numerical Recipes in C*, Cambridge University Press, 1988.