Computer Vision for Reconstructive Plastic Surgery

S. M. Bhandarkar[1], A. S. Chowdhury[1], Yarong Tang[1], Jack Yu[2,4], E.W. Tollner[3]
(suchi@cs.uga.edu, ananda@cs.uga.edu, yrtang@uga.edu, jyu@mail.mcg.edu, btollner@engr.uga.edu)

[1]Dept. of Computer Science
[3]Dept. of Biological & Agricultural Engineering
The University of Georgia
Athens, Georgia 30602-7404, USA

[2]Dept. of Plastic Surgery
[4]Dept. of Oral & Maxillofacial Surgery
The Medical College of Georgia
Augusta, Georgia 30912-4080, USA

***Abstract* – High energy traumatic impact of the craniofacial skeleton is an inevitable consequence of today's fast paced society. The work presented in this paper leverages recent advances in computer vision, computer visualization and computer aided design/manufacturing to reduce the fractures and reconstruct the craniofacial skeleton and reduce the fractures *in silico.* More specifically, the work has two objectives. The first is to design surface matching algorithms to reduce the distances separating the two opposable fracture surfaces both in 2D and 3D when imaged using Computer Tomography (CT). The second is to develop a Graphical User Interface (GUI) that will permit a plastic surgeon to perform the surgery in a virtual environment i.e., *in silico* before performing it on a physical patient. The resulting software could be potentially used as a tool for *ex vivo* surgical planning as well as an instructional tool for training surgery residents and students.**

***Index Terms* – Surface Matching, Computer Vision, Graphical User Interface, Virtual Surgery.**

I INTRODUCTION

High energy traumatic impact of the craniofacial skeleton is an inevitable consequence of today's fast paced society. The plastic surgeon restores the form 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. However, there are several critical, inherent and intrinsic limitations of this current, standard approach. To visualize the fragments in order to reduce them necessitates their exposure which consequently reduces the attached blood supply. 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 mal-alignments of the bone fragments. The present paper seeks to solve the above dilemma by developing an enabling technology that leverages recent advances in computer vision, computer visualization and computer-aided design and manufacturing (CAD/CAM) to reconstruct the craniofacial skeleton and reduce the fractures *in silico.* The work presented in the paper thus has two primary objectives, namely (a) To develop a Graphical User Interface (GUI) to allow the surgeon to manually identify corresponding landmarks on opposable fracture surfaces imaged using Computer Tomography (CT), both in 2D and 3D. The surgeon should be also able to perform basic geometric operations like rotation, translation etc. on the broken fragments, and, (b) To implement computer vision algorithms to reduce the distances separating the two opposable fracture surfaces both in 2D and 3D using free-form surface matching algorithms.

II GENERAL PROCEDURE

The system for computer-aided surgery can be viewed as a synergetic combination of computer-vision based automation and GUI-based human (surgeon) intervention. The input to the system is a sequence of 2D grayscale images of a fractured human mandible, generated via Computer Tomography (CT). Each of the slices shown in Fig.1 has dimensions 150 mm x 150 mm and has 8-bit color depth. As far as automation is concerned, various computer vision algorithms are used to obtain geometric transformations between opposable fracture surfaces of the available broken fragments of the lower human mandible which have been imaged using CT. Simultaneously, using the interactive GUI, a surgeon can provide landmark points for the surface matching algorithm and can drag or rotate a specific broken fragment. The software performing the various tasks is written in the form of plug-ins for a JAVA-based Image Processing package called ImageJ, developed by the National Institutes of Health (NIH), Bethesda, Maryland USA.

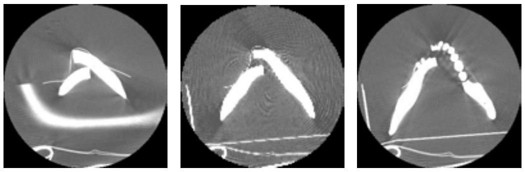


Fig.1: A typical sequence of 2D CT Images

III GUI DESIGN

For the front-end, we have designed a simple but compact GUI consisting of buttons for performing various functions. Good interface design principles [1] were followed in the development process. Subsequently, considerable importance was given to various factors such as the orientation of buttons, size of buttons, font-size on 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 be using it. Each button is entrusted with a specific function such as rotation, translation, surface matching etc. Thus a surgeon has to click a specific button to perform the desired operation. For translation, the user can provide horizontal and vertical translation parameters (for any fragment) from the keyboard. Alternatively, the user can simply select a specific fragment and drag it with a mouse to its new position. When performing a rotation the user can choose the pivot point (i.e. the center of rotation) with a mouse click or can input the coordinates from the keyboard. Furthermore, via mouse clicks, the user can specify the corresponding landmark points on the two fractures. In this case the final matching is done on the basis of landmarks, provided by the user (i.e., surgeon). Fig. 2(a) and 2(b) illustrate our design of the GUI. The top level interface is shown in Fig. 2(a). Here the label on each button indicates the task it is designated to perform. For example, in order to rotate a fragment, the user can click on the ‘’Rotate” button, upon which the menu screen depicted in Fig. 2(b) pops up. At this point the user may input the degree of rotation from the keyboard, in which case the rotation centre will be assumed to be the geometrical centre of the selected object. The user can also click on a specific point on the object to designate it as the pivot point for rotation and then specify the angle of rotation. Furthermore the user can perform the rotation or translation operation for a particular slice or for all the slices in a given stack. In the latter case it becomes a 3D operation.

2(a)

2(b)

Fig.2: A 2-level User Interface. When one clicks on the “Rotate” button in 2(a), the screen in 2(b) pops up.

IV. COMPUTER VISION TASKS

For each button in the front-end GUI, there is a dedicated back-end function which can perform *geometric transformations* [2] such as *rotation* or *translation*, or can do *image registration* tasks such as *contour matching* for a single slice (a CT image) or *surface matching* for a set of such slices.

A. Geometric transformations

For 2D Rotation we used a standard 2x2 orthonormal matrix representation. The rotation is performed in the x-y plane about the geometrical centre of the object or about the chosen pivot point, as the case may be. The 2D translation is similarly represented by the standard 2x1 column vector. Fig. 3(a) shows the selected component in an original image on which the transformations are applied. The results of rotation (clockwise by 30^o) and translation (20 pixel units along x-direction and 20 pixel units along y-direction) are shown in Fig. 3(b) and Fig. 3(c) respectively.

(a) (b) (c)

Fig.3: (a) is the original image, (b) the bottom fragment rotated by 7^0, (c) same translated by 20 pixel units both along x and y direction

B. Contour Matching

Contour matching in 2-D in order to bring two opposable fracture surfaces in registration was achieved using the *Singular Value Decomposition (SVD)* algorithm [2]. Some image pre-processing tasks were performed prior to contour matching which include *binarization* followed by *Connected Component Labeling* (CCL). For the given set of CT images, the bright components represent the broken mandible fragments and the dark areas represent soft tissues. Hence, the threshold for the binarization is not difficult to select. Based on prior knowledge, we classify a pixel with gray-scale value above 250 to belong to the above object of interest and represent it using the color black (as shown in Fig. 4b). Thus we have a binary image B(i, j) for a gray scale CT image slice G(i, j) given by

$$B(i, j) = \begin{cases} 0 & \text{if } G(i, j) > 250 \\ 1 & \text{otherwise} \end{cases} \qquad (1)$$

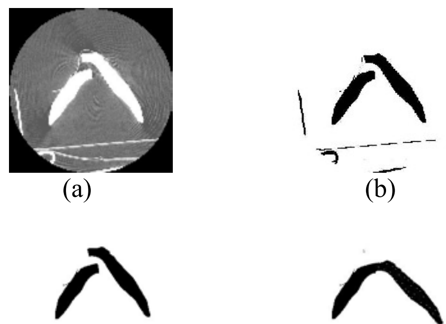
However, binarization by itself cannot distinctly represent the two fractured fragments since we also need to filter out some undesired artifacts. CCL in conjunction with an area filter is used to remove unwanted artifacts (which are typically small in size). The threshold value for the area filter is chosen to be 1000. Connected components with area less than the threshold value are deleted. The result is shown in Fig. 4(c).

After binarization and CCL, the user is required to click on the corresponding landmark points on the two fragments in a particular CT slice. The two sets of corresponding N landmark points $((a_1 \ldots a_N)$ and $(b_1 \ldots b_N))$ from the two fragments along the fracture contours constitute the input to the SVD algorithm. In principle, 2 pairs of corresponding points from the 2 fracture contours are sufficient to yield the parameters t_x, t_y and θ. However, the SVD algorithm is able to handle an over-constrained system of equations in order to determine the 2D transformation (both rotation $R = R(\theta)$ and translation $t = [t_x \; t_y]$) required for matching these contours. This is achieved by minimizing the weighted sum of the residual errors ε^2:

$$\varepsilon^2 = \sum_{i=1}^{N} w_i \left\| b_i - (Ra_i + t) \right\|^2 \qquad (2)$$

where weights satisfy the following conditions $w_i \geq 0$, $1 \leq i \leq N$ and $\Sigma w_i = 1$. Using an over-constrained system of equations to solve for the transformation parameters makes the parameter estimation robust to additive Gaussian noise.

In our case, we choose $\Sigma w_i = 1/N$. The b_i's and a_i's are 2x1 column vectors, each having its own x and y coordinates. The result of the matching 2D contours is shown in Fig. 4(d).



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

Fig.4: Various Stages of 2D Matching
(a) Original CT image, (b) After thresholding
(c)After CCL & filtering, (d) Matched contours

C. Surface Matching

The most challenging task is to compute the 3D transformation for matching two opposable fracture surfaces for a set of CT slices. There are various existing algorithms for performing this task. The SVD algorithm mentioned above can be extended for achieving this matching in 3D. However SVD can be only applied when the cardinality of the two data sets to be matched are the same, which is difficult to satisfy in the present application. Hence an alternative algorithm called the ICP (*Iterative Closest Point*) algorithm is used [3, 4]. Before using the ICP algorithm, binarization and CCL are performed on all the slices under consideration. In addition, the task of interactive contour detection is performed on each of the 2D image slices comprising the 3D image stack. The interactive contour detection algorithm requires the user to click on the end points of the fracture contour in each of the CT slices. The intervening contour points are automatically generated using a contour tracing algorithm. The contour points from the CT image stack are assembled to form the 3D surface point data set. A 3D surface point data set is generated for each fracture surface. The ICP algorithm intelligently determines the initial pairs of matched points called the *Closest Set.* It then iteratively evaluates the 3D rigid body transformation (3D translation and 3D rotation) to bring the two surfaces into registration until a specific error convergence has been reached. There are many interesting aspects to the ICP algorithm. The rotation is computed using the theory of *Quaternions* [5]. Although the original two sets to be matched have different cardinalities but the *Closest Set*, determined at each iteration, has the same number of points as the source set. To compute the *Closest Set*, which is the most crucial step in the ICP algorithm, the matching pairs are determined using *Bipartite Graph Matching* [6], [7]. For the *Bipartite Graph* G(V, E) [8], the two 3D data sets correspond to the two subsets (V1, V2) into which the original Vertex Set V is partitioned. 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 3-D rigid body transformation. Thus, the edge weights are given by:

$$W_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2\right]^{1/2} \quad (3)$$

We have employed the *Maximum Cardinality Minimum Weight Bipartite Matching* algorithm [8] based on the Hungarian method proposed by Kuhn [9] to determine the matched pairs in the *Closest Set*. ICP performs the registration task by minimizing the Mean Square Error (MSE) (between source set and goal set) objective function. The expression for the MSE is similar to the one in equation (2), the only difference being that the rotation matrix R is now R(θ, φ, χ) and the translation vector t is $t[t_x\ t_y\ t_z]$; where θ, φ, χ represent the rotation angles about the x-axis, y-axis and z-axis respectively and t_x , t_y , t_z represent translations along these axes. Obviously, each b_i and a_i is a 3x1 column vector, with its own x, y and z coordinates.

V CONCLUSIONS AND FUTURE WORK

The first version of the GUI has been tested by the plastic surgeons at the Medical College of Georgia and has been approved. The surface matching algorithms, have so far resulted in a MSE of the order of 0.15 – 0.35 (measured in mm^2). Initially, the *Closest Set* computation in ICP was performed in a purely greedy fashion. However the current approach to *Closest Set* computation using the Maximum Cardinality Minimum Weight Bipartite Graph Matching algorithm has yielded much superior results. In addition to yielding a lower MSE, the improved ICP converges within fewer than 5 iterations. Visual verification of the results indicate that the fractures were well matched. One serious limitation of the ICP algorithm is that it is purely data driven. Since the ICP algorithm performs a local search, the 3D transformation obtained represents a locally optimal solution. A 3D transformation that preserves the global 3D shape of the human mandible is desirable. Future enhancements to the ICP algorithm will incorporate a model-driven search which will achieve both the objectives i.e. matching the fracture surfaces to a high degree of accuracy and preservation of the global 3D shape of the human mandible. This would entail significant changes to the variational principle within the ICP algorithm (note that the current variational principle is based on the MSE only).

ACKNOWLEDGEMENT

The research was supported in part by a research grant from the University of Georgia - Medical College of Georgia Joint Research Program administered by the Biomedical and Health Sciences Institute at the University of Georgia and by an Engineering Research Grant by the Faculty of Engineering at the University of Georgia. The authors also sincerely acknowledge the help of Prof. Robert W. Robinson, Department of Computer Science, The University of Georgia for explaining the intricate details of the Bipartite Graph Matching algorithm.

REFERENCES

[1] Y. Rogers, H. Sharp and J. Preece, *Interaction Design beyond human-computer interaction*. John Wiley & Sons. 2002, ch 7.

[2] R.M. Harlick and L.G. Shapiro, *Computer and Robot Vision, Vol.II.* Addison-Wesley, 1993. pp. 160 – 167.

[3] P.J. Besl and N.D. McKay, "*A Method for Registration of 3-D Shapes",* IEEE Transaction on Pattern Analysis and Machine Intelligence. Vol. 14, Feb. 1992. pp 239 – 255.

[4] N. Nikolaidis and I.Pitas, *3-D Image Processing Algorithms,* John. Wiley & Sons. 2001. pp. 114 -117.

[5] J.B. Kuipers, *Quaternion and Rotation Sequences,* Princeton University Press, 1999. ch.3.

[6] N. Christofides, *Graph Theory An Algorithmic Approach.* Academic Press, 1975. pp. 372 – 379.

[7] S. Ranade and A. Rozenfield," *Point Pattern Matching by Relaxation*", Pattern Recognition. Vol.12, 1980. pp 269-275.

[8] W. Kim and A.C. Kak, *"3-D Object Recognition Using Bipartite Matching Embedded in Discrete Relaxation",* IEEE Transaction on Pattern Analysis and Machine Intelligence. Vol. 13, March. 1991. pp 224 – 251.

[9] H.W. Kuhn, "*The Hungarian method for the assignment problem*", Nav. Res. Log. Quart. 2, pp. 83.