12-2013

Automated Fragmentary Bone Matching

Ali Saad Mustafa
University of Tennessee - Knoxville, ali.mustafa@utk.edu

To the Graduate Council:

I am submitting herewith a thesis written by Ali Saad Mustafa entitled "Automated Fragmentary Bone Matching." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Biomedical Engineering.

Mohamed R. Mahfouz, Major Professor

We have read this thesis and recommend its acceptance:

Richard D. Komistek, William R. Hamel

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Automated Fragmentary Bone Matching

A Thesis Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Ali Saad Mustafa

December 2013

DEDICATION

I dedicate my work to my father, mother, sister, brothers, spouse, and kids for their encouragement and continuous support

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Mohamed Mahfouz, for his guidance and continued support of my research. I would like to extend my gratitude to Emam Abdel Fatah, Lyndsay Bowers and Rebecca Robertson. I am also very appreciative of the assistance extended to me by my colleagues, Hatem El-Dakhakhni and Nicholas Battaglia.

I am particularly grateful for all my committee members' willingness to participate in acquiring this honored degree, including Dr. Richard Komistek, Dr. William Hamel and Dr. Emam Abdel Fatah.

After all of these recognitions, my efforts were encouraged by my father, mother, spouse, sister and brothers and I will be forever thankful for their commitment to me.

ABSTRACT

Identification, reconstruction and matching of fragmentary bones are basic tasks required to accomplish quantification and analysis of fragmentary human remains derived from forensic contexts. Appropriate techniques for three-dimensional surface matching have received great attention in computer vision literature, and various methods have been proposed for matching fragmentary meshes; however, many of these methods lack automation, speed and/or suffer from high sensitivity to noise. In addition, reconstruction of fragmentary bones along with identification, in the presence of reference model to compare with, in an automatic scheme have not been addressed. In order to address these issues, we used a multi-stage technique for fragment identification, matching and registration.

The study introduces an automated technique for matching of fragmentary human skeletal remains for improving forensic anthropology practice and policy. The proposed technique involves creation of surfaces models for the fragmentary elements which can be done using computerized tomographic scans followed by segmentation. Upon creation of the fragmentary elements models, the models go through feature extraction technique where the surface roughness map of each model is measured using local shape analysis measures. Adaptive thresholding is then used to extract model features. A multi-stage technique is used to identify, match and register bone fragments to their

corresponding template bone model. First, extracted features are used for matching with different template bone models using iterative closest point algorithm with different positions and orientations. The best match score, in terms of minimum root-mean-square error, is used along with the position and orientation and the resulting transformation to register the fragment bone model with the corresponding template bone model using iterative closest point algorithm.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Motivation

Mass disasters, cremation litigation, human rights investigations and many other types of modern cases raise questions such as "How many individuals are represented in a group of remains?" and "How can remains of single individual be identified within collection of remains from multiple individuals?" [1]. Cases and questions involving commingling issues are highly valuable and answering to these questions is highly valuable too. From a forensic anthropologist's perspective, commingled remains confound the application of standard methods for the determination of the minimum number of individuals, MNI, and complicate the accurate assessment of the biological profile [2]. The refinement of methods and development of new tools focusing on fragmentary and commingled human remains are critical to enable forensic scientists to deal with individual fragmentary cases, issues surrounding mass disasters, and commingled mass graves. Figure 1-1 shows bone samples from James Mellaart excavation of skeletal remains of around 500 individuals at Çatalhöyük, Turkey [3].

Figure 1-1: Bone samples from James Mellaart excavation, [3]

Since the quantification, analysis, and identification of fragmentary human remains derived from forensic contexts is of that importance, the reconstruction and matching of fragmentary human remains is the primary step to accomplish. Appropriate techniques for three-dimensional surface matching have received much attention in computer vision literature, and various methods have been proposed for matching fragmentary meshes; however, many of these methods lack automation, speed and suffer from high sensitivity to noise. In addition, reconstruction of fragmentary bones along with identification in the presence of reference model to compare with in an automatic scheme is not addressed, to the best of our knowledge. In order to address these issues, we used a multi-stage technique for fragment identification, matching and registration.

1.2 Background

In the area of three-dimensional surface matching, different techniques have been proposed for matching fragmentary meshes. Although most of the feature-based methods follow the same pipeline to achieve the goal, they have different methodologies. Generally, in order to solve the 3D surface matching problem using feature-based approaches, a feature extraction step, including detection and/or description, follows the data segmentation step. These features are then used to match

surfaces either in pair-wise or multi-piece matching step. Figure 1-2 shows the general outline of feature-based algorithms, [4].

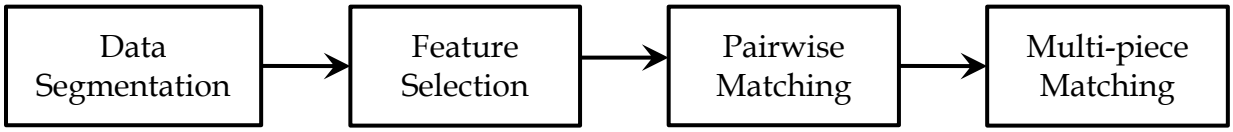

Figure 1-2: A high level overview of feature-based algorithms, [4]

A geometric matching algorithm is proposed to reassemble generic fractured objects [4]. The algorithm first segments the fragments into a set of surfaces bounded by sharp curves using multi-scale edge extraction which is constrained to return cycles of edges. The algorithm then uses graph cut algorithm to partition the set of surfaces into original and fracture surfaces. Integral invariant descriptors are used to produce a set of fracture surface features (volume and volume distance) and a set of fracture edge features (deviation). In order to allow for fast retrieval of matching parts a relatively low dimension of features is used for fracture surface features as well as fracture edge features. The used features in [4] are position, principal components, principal directions and integral invariant descriptor at the position of the feature cluster. These features are calculated to represent fracture surface features, and then used for rough and fine registration of surface features. For pairwise matching, shape and topological pruning is used based on the features parameters to discard redundant and false

correspondences and to verify the correct ones. Forward search method is then used to find possible matches between fracture surfaces. Since forward search method needs a noise-free initial subset, some consistency tests are performed on pairs of correspondences including geometric and registration consistency. The set of possible matches between the fragments along with a quality rating for each match are used to iteratively compute a global multi-piece matching, perform a local multi-piece registration and merge matched fragments until the object is reassembled.

Fractured surfaces matching algorithm is proposed to reassemble broken solids [5]. The algorithm uses multi-scale integral invariants of the fractured surface points to pick surface feature points. The matching process then is performed in two stages. An initial matching is done by comparing the multi-scale integral invariants of the surface feature points using distance metric. As a result, an initial matching point pairs are established. To increase the accuracy and speed up the matching process, some spatial compatibility constraints are used as surface similarity measures to discard the outliers and select an effective matching point set. In the second stage, hierarchal alignment greedy algorithm [6] is used to coarsely align the two fractured surfaces using the effective matching point sets. ICP is then used for fine alignment as the final step in the registration process.

A similar problem was addressed in the context of solving three-dimensional puzzle of comminuted articular fractures [7]. The algorithm is developed to be used as a pre-operative planning procedure for reconstructing highly comminuted articular fractures in order to reduce the difficult surgical challenge of solving a complicated three-dimensional puzzle. A semi-automatic fragment reconstruction approach is used to solve the puzzle problem. The algorithm first segment fracture surfaces into discrete patches using region growing algorithm based on surface normal direction that uses input seed points to propagate patches up to boundaries of high curvature. A facet whose normal deviated from the seeded region by greater than a threshold angle is classified as an edge, and not analyzed in future iterations. An iterative registration function is used to bring fragments surfaces into rough alignment with template, by aligning centers of mass and principal axes. The platform allows the user to adjust the alignment manually if the alignment algorithm fails. As a final step, ICP is used for fine alignment.

A semi-automatic algorithm is proposed for bone fracture reconstruction [8]. The algorithm enables the user to specify matching surface regions between fragment pairs, and initiate a pairwise and global fragment alignment. The algorithm then merges iteratively individual fragments into a group of aligned fragments using graph-based model, where the graph nodes correspond to fragments and the edges represent the user-specified surface match or correspondence between fragments. A variant of ICP

algorithm is used to align fragments; where (1) the error metric is defined as the sum of point-to-plane distances instead of point-to-point distances and (2) the surfaces are selectively subsampled using significant geometric surface variations.

In the area of quantification of bone fragments, fragmented remains are quantified in [9] using Geographic Information System (GIS) software to derive minimum number of elements (MNE) values and MNI estimates. The system takes the bone fragments, which are manually identified by experts, and then generate a map that can provide estimate for the element under investigation. The system is time consuming and is dependent on the observer in the manual identification of fragments.

However, the majority of the presented methods lack either speed, sensitivity to noise, automation in reconstruction, or identification of pieces relative to a reference model. The core of the presented work is to introduce an automated method for identification, matching and registration of fractured pieces of fragmentary bones with the corresponding template model. Using statistical bone atlas, missing parts can further be interpolated to generate the complete bone [10] that can be used for extracting measurements and assessment of biological profile

1.3 Overview

The purpose of this research is designed to work on fragments of four skeletal elements, femur, humerus, pelvis and skull bones. The first step involves generation of surfaces models for the fragmentary elements. This can be done using CT or laser scanner. In the case of CT scanning, a segmentation step is performed to generate the surface models. Upon creation of the fragmentary elements models, the next step is to extract features from each bone fragment by measuring surface roughness. Surfaces points with roughness above certain specified threshold values are denoted as feature points. A multi-stage technique is then used to identify, match and register these bone fragments to their corresponding template bone model. For each bone fragment, the features are used to be matched with template bone features using iterative closest point (ICP) algorithm. As long as it has been proven that ICP algorithm converge to a local minimum with respect to the mean-square distance metric, which means that it guarantees the correct registration given that the two shapes are somehow close in terms of position and orientation, we used different combinations of position and orientation of bone fragment relative to the template bone. For each combination, we applied ICP to try to match and register each fragment features with the four template bone's features. The best match score, in terms of minimum root-mean-square error, is used along with the position/orientation and transformation to register the bone fragment with the corresponding template model.

We refer to template model here as an average mold that captures the primary shape information of a skeletal element. Three dimensional statistical bone atlases are constructed from large datasets of bones [11], [12], [13], [14]. The atlas guarantees point-to-point and surface correspondence across the entire sample, which allows for an accurate calculation of an average template. This template bone captures the global shape characteristics for each bone across an entire population which can then be used to guide the placement of the fragmentary pieces into anatomically correct space and to estimate missing fragments.

The presented work is embedded in a 3D virtual environment where the user load surface models, start the procedure and view the final results. The user can accept or reject the matching and registration results after running the procedure, as well as she/he can manipulate each model separately either by hand or through GUI controls and start the procedure again.

Figure 1-3 highlights the proposed framework where fragmentary pieces are first CT scanned and segmented to generate surface models. The resultant surface models are then used to extract fragmentary bones features, followed by matching and registration. The visual feedback block presents user interactions through developed 3D virtual environment.

The document is organized as follows. Chapter 2 covers surface properties and methods used for feature extraction in the literature, while chapter 3 covers matching and alignment methods. Chapter 4 introduces the proposed framework, while chapter 5 presents the results and discussion, and finally chapter 6 states the conclusion and future work.

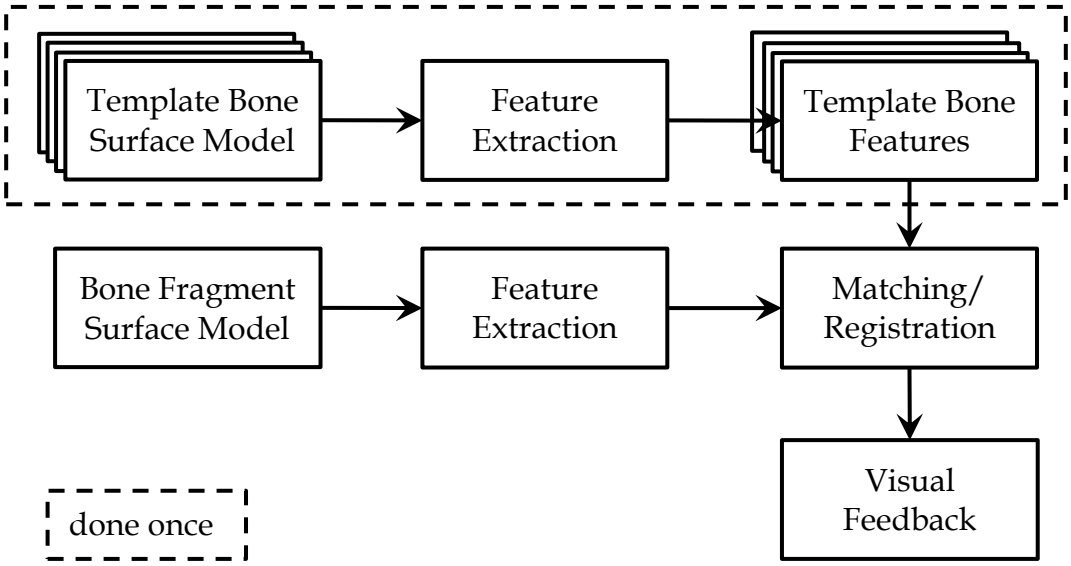

Figure 1-3: Proposed framework

2. SURFACE PROPERTIES

2.1 Surface Curvature

A spherical surface is different from plane in which it is curved while the plane is flat. In Riemannian geometry there is a way of making this general concept into a mathematical concept, which is called curvature [15]. A sphere and a plane both have a constant curvature value, i.e. the curvature at every point is the same. The difference is that the sphere has a non-zero curvature value at all points while the plane has a zero curvature value.

2.1.1 Fundamental Forms

A parametric surface is a surface in the Euclidean space R^3 defined by a parametric equation. Surface curvature, area, differential properties such as the first and second fundamental forms, Gaussian, mean, and principal curvatures can be computed from a given parameterization [16]. A surface mesh can be either represented by or approximated to parametric surface, which is a general representation of surfaces.

Given a surface defined in parametric representation, Eq. 2-1,

$$\vec{r} = \vec{r}(u, v) \qquad \text{Eq. 2-1}$$

For example,

$$\vec{r} = (au^2 + bv^2,\ cu + dv,\ e) \qquad \text{Eq. 2-2}$$

where a, b, c, d and e are constants, and u, v are surface parameters, then the surface first derivatives can be defined, Eq. 2-3,

$$\overrightarrow{r_u} = \frac{\partial \vec{r}}{\partial u}, \qquad \overrightarrow{r_v} = \frac{\partial \vec{r}}{\partial v} \qquad \text{Eq. 2-3}$$

where the surface first order properties (tangent plane, and normal vector) can be defined, Eq. 2-4,

$$(\overrightarrow{r_u}\,,\ \overrightarrow{r_v}), \qquad \overrightarrow{n} = \frac{\overrightarrow{r_u} \times \overrightarrow{r_v}}{|\overrightarrow{r_u} \times \overrightarrow{r_v}|} \qquad \text{Eq. 2-4}$$

and the surface second derivatives can be defined, Eq. 2-5,

$$\overrightarrow{r_{uu}} = \frac{\partial}{\partial u}\left(\frac{\partial \vec{r}}{\partial u}\right), \qquad \overrightarrow{r_{uv}} = \frac{\partial}{\partial v}\left(\frac{\partial \vec{r}}{\partial u}\right), \qquad \overrightarrow{r_{vv}} = \frac{\partial}{\partial v}\left(\frac{\partial \vec{r}}{\partial v}\right) \qquad \text{Eq. 2-5}$$

The first and the second surface fundamental forms can be defined, Eq. 2-6, Eq. 2-7.

$$I = Edu^2 + 2Fdudv + Gdv^2 \quad \text{Eq. 2-6}$$

$$II = Ldu^2 + 2Mdudv + Ndv^2 \quad \text{Eq. 2-7}$$

where

$$E = \overrightarrow{r_u}.\overrightarrow{r_u}, \qquad F = \overrightarrow{r_u}.\overrightarrow{r_v}, \qquad G = \overrightarrow{r_v}.\overrightarrow{r_v} \quad \text{Eq. 2-8}$$

$$L = \overrightarrow{r_{uu}}.\vec{n}, \qquad M = \overrightarrow{r_{uv}}.\vec{n}, \qquad N = \overrightarrow{r_{vv}}.\vec{n} \quad \text{Eq. 2-9}$$

2.1.2 Principal Curvatures

Surface curvature at point, which describes the local shape behavior, can be expressed in terms of the two principal curvatures, k_1, k_2. The principal curvatures are the eigenvalues of matrix constructed from the coefficients of the first and the second fundamental forms of the surface, and they are the solution of Eq. 2-10, Eq. 2-11 [17]. The two principal curvatures characterize the rate of the maximum and the minimum bending of the surface and the tangent direction in which they occur. The two principal curvatures at any point on a surface can, according to Euler's formula [17], determine the rate of surface bending along any tangent direction at the same point.

$$det(II - \lambda I) = det \begin{vmatrix} L - \lambda E & M - \lambda F \\ M - \lambda F & N - \lambda G \end{vmatrix} = 0 \qquad \text{Eq. 2-10}$$

$$k_1 = \lambda_1, \qquad k_2 = \lambda_2 \qquad \text{Eq. 2-11}$$

Figure 2-1 shows example of surfaces with normal planes in the direction of the principal curvatures, [18].

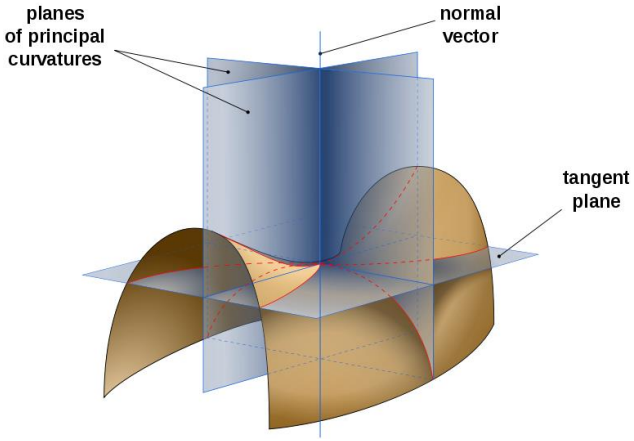

Figure 2-1: Saddle surface with normal planes in directions of principal curvatures, [18]

2.1.3 Gaussian and Mean Curvatures

Gaussian and mean curvatures, Eq. 2-12, Eq. 2-13, are another two measures of the surface curvature. They have greater geometrical significance than the principal curvatures, and can also be used to describe the local behavior of the surface.

$$K = k_1 k_2 \quad \text{Eq. 2-12}$$

$$H = \frac{k_1 + k_2}{2} \quad \text{Eq. 2-13}$$

2.1.4 Curvedness

A different formula for representing surface curvature is proposed in [19] based on the principal curvatures. Curvedness, Eq. 2-14, measures the intensity of the surface curvature and describes how strongly curved the surface.

$$C = \sqrt{\frac{{k_1}^2 + {k_2}^2}{2}} \quad \text{Eq. 2-14}$$

2.1.5 Shape Index

A quantitative measure of surface shape is proposed also in [19], which describes different surfaces shape by assigning different values except for planar points, which have indeterminate value. Based on shape index, Eq. 2-15, local surface shapes can be classified into cup, cap ($|S_I(p)| = 1$), rut, ridge ($0 < |S_I(p)| < 1$) and saddle ($S_I(p) = 0$). Figure 2-2 shows different local shape variations based on the definition of shape index.

$$S_I(p) = \frac{2}{\pi} \arctan\left(\frac{k_1 + k_2}{k_2 - k_1}\right) \quad \text{Eq. 2-15}$$

Figure 2-2: Local shape variations based on shape index, Eq. 2-15, [20]

2.2 Surface Roughness

Surface roughness is one of the surface properties and it is generally used to describe the level of noise presented on the surface. A new roughness measure is proposed in [21] to be used for 3D visual masking, based on the idea that rough region can hide some geometric distortions with quite similar frequencies. Surface roughness is used in a geometric matching scheme for reassembling fractured objects [4] to classify the faces of the fractured objects into original, which come from the boundary surface of the unbroken object, and fracture faces, which were created when the object broke.

Different methods have been proposed to measure surface roughness. In [22], surface roughness is defined as the root mean square of the difference between the mean curvature and the average mean curvature of L-ring neighborhood where L is the neighborhood level at which the surface roughness is being computed, Eq. 2-16. Mean curvature at point is calculated by averaging all directional curvatures at object's point, Eq. 2-17. A normalization step is done, by resizing the model so that the average

distance between the centroid and all vertices is a unit sphere, before calculating the mean curvature in order to compare two different models on the same ground and removing the scale factor.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(k(x_i) - k^L(x_i))^2} \quad \text{Eq. 2-16}$$

$$k^L(x_i) = \frac{1}{|N(x_i)|}\sum_{x_j \in N(x_i)} k(x_j) \quad \text{Eq. 2-17}$$

A different roughness measure is proposed [21] for visual masking applications, and is calculated by adaptively smoothing the 3D object, and calculating the curvature of each point of the smoothed version and the original version of the object. An average curvature value is processed for each point which corresponds to the mean of the curvature of all points from its local window. Roughness map is then constructed by processing an asymmetric difference between each point's average curvature values on the original and the smoothed version of the object, Eq. 2-18. Figure 2-3 shows an overview of the presented algorithm in [21]. Figure 2-4 shows an example of surface roughness generated in [21].

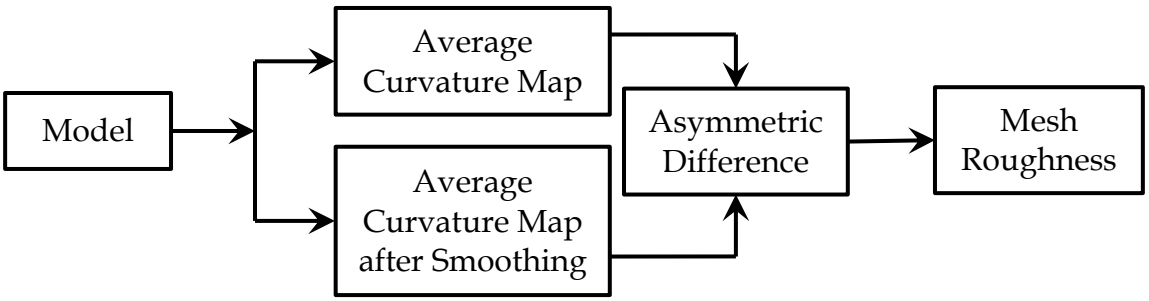

Figure 2-3: Overview of mesh roughness algorithm, [21]

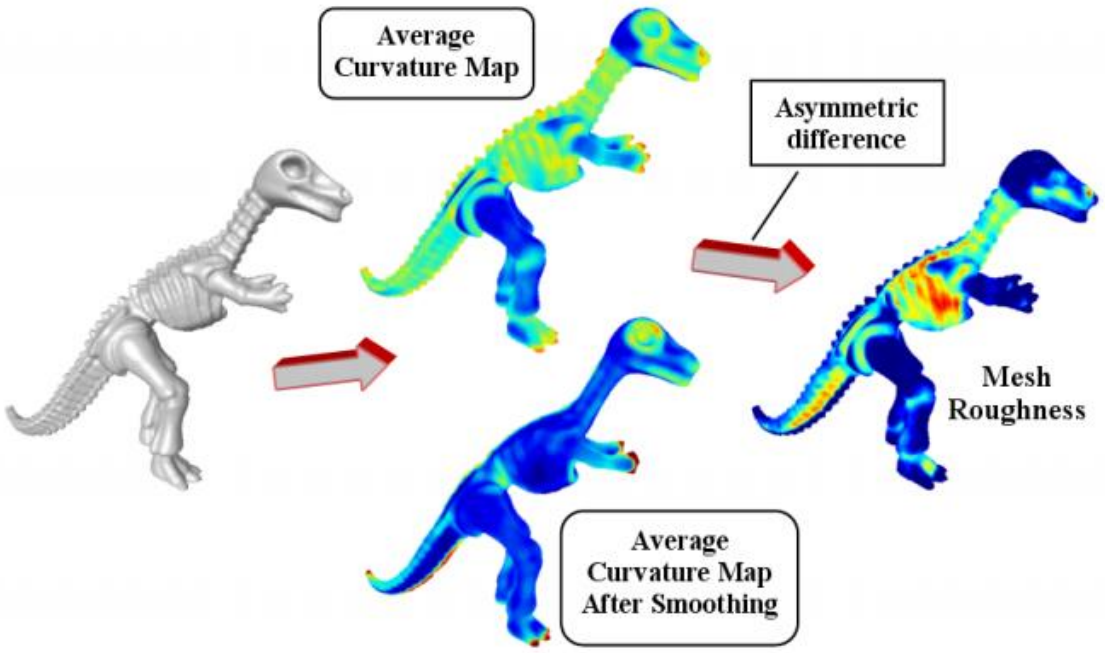

Figure 2-4: Mesh roughness example, [21]

$$R(v_i) = \begin{cases} AC(v_i) - AC(v_i^s) & if\ AC(v_i) > AC(v_i^s) \\ 0 & else \end{cases} \quad \text{Eq. 2-18}$$

Surface roughness is incorporated in generating surface curvature information instead of integral invariants in [4] because the latter is more expensive in the sense of computations and unstable at the small scales. Roughness is calculated as the integration of bending energy $e_k(p)$ over the local neighborhood, Eq. 2-19, where the bending energy is defined in terms of squared differences of normal vectors at the local neighborhood, Eq. 2-20.

$$e_k(p) = \frac{1}{k} \sum_{i=1}^{k} \frac{\left|\left|n_p - n_{q_i}\right|\right|^2}{||p - q_i||^2} \qquad \text{Eq. 2-19}$$

$$\bar{e}_{k,r}(p) = \frac{1}{|N_r(p)|} \sum_{q \in N_r(p)} e_k(q) \qquad \text{Eq. 2-20}$$

Concepts of local and global roughness are introduced in [22], where the local roughness is described as the average behavior of all vertices in their local region, generally one neighborhood, whereas the global roughness is described as the average behavior of all vertices with respect to larger region, generally a complete surface. In this sense, a roughness profile or local-to-global roughness plot can be generated which captures the information about the roughness of the object from local region around vertices to global region which is the entire object. Figure 2-5 shows an example of the roughness profile for a sphere with 14% random noise added in the radial direction, and for bunny using roughness measure proposed in [22].

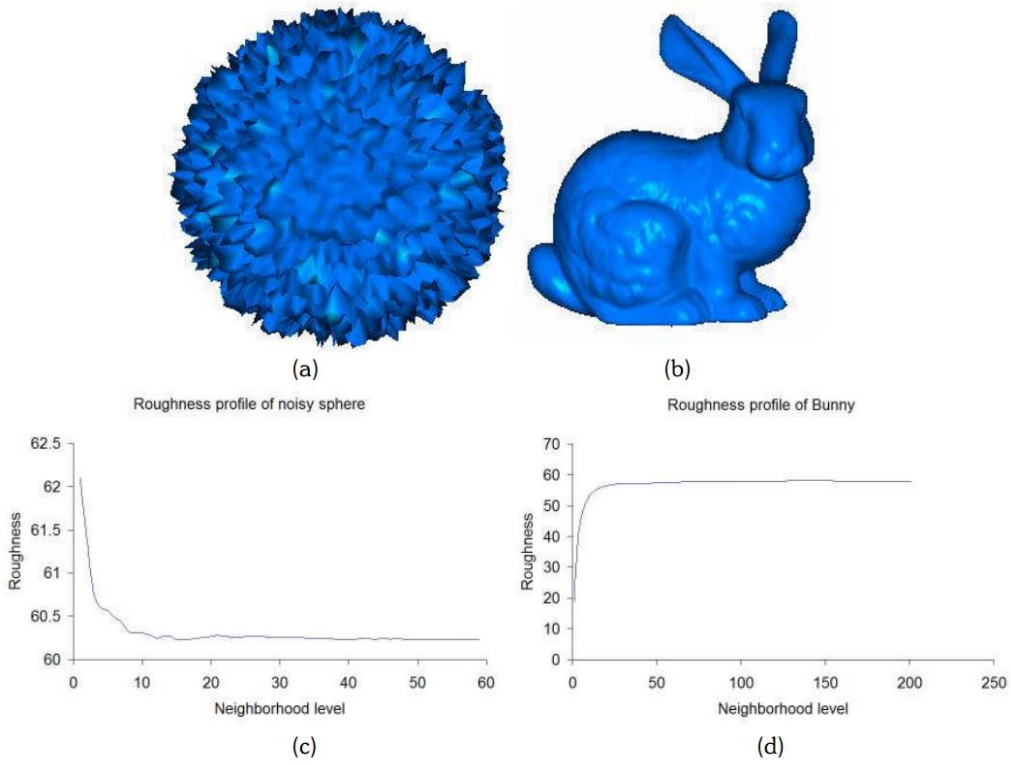

Figure 2-5: (a) Noisy sphere, (b) bunny and (c) & (d) their roughness profiles, [22]

2.3 Surface Features

Features is a term often used to refer to prominent elements that capture most of the relevant shape information such as corners or sharp edges. Feature-based methods have become a broadly used paradigm in shape representation, retrieval, matching and other applications due to their excellent performance in practice [23]. In matching 3D shapes, correspondence problem is one of the important challenges where the goal is to find matches between point in two 3D shapes. The advantage of using features-based

methods for correspondence problems is the ability to identify similar points on two shapes, therfore reducing the set of potential correspondence candidates. Similarity problem is another challenge in which a quantification critera has to be established to deremine the degree of similarity between two shapes. Feature-based methods allow to represent the shape as a collection of primitive elements that simplify the solution of the problem.

Feature-based methods can be divided into two main stages. The first stage is feature detection, where the goal is to find the location of stable point that capture most of the relevant shape information (features). The second stage is feature description, where the goal is to find a way to represent (describe) the shape properties at these feature points [23]. In order to adopt for differences in position, direction and scaling, feature detectors and descriptors should be affine-invariant, to insure stability under transformation that an object can undergo.

Thus, one of the main challenges of a general feature-based 3D shape analysis method is to find a set of features that are invariant to shape transformations and carry sufficient information to allow using these features for finding correspondence and similarity.

2.3.1 Feature Detection

One of the basic requirements of any feature detector is robustness. In other words, the feature detector should be able to detect the same set of corresponding points in two instances of a shape. Although there is no single way to define a feature, there is a common sense in selecting a feature; detected feature points should contain sufficient information to distinguish a shape from the others.

Usually feature detection is achieved via local curvature analysis [24]. Different shape encoding methods have been used to detect feature points. A valuable survey of surface curvature calculation methods is presented in [25] along with a suite of test cases that has been developed to evaluate the accuracy of these methods. The survey categorizes the developed mesh curvature calculation methods into two groups. The first group, surface fitting methods, involves finding a function that fits the mesh locally, either by interpolation or by approximation, and calculating the curvature of the fitting function. If the function is interpolating, then the function goes through each vertex, and in this case a specific relationship between the number of vertices and the number of coefficients is required in order to solve the problem for the fitting function coefficients. If the function is approximating, then the function minimizes some measure of distance from the vertices, and a minimum number of vertices is required to compute the function coefficients as a solution to a least square minimization. The second group, discrete methods, involves developing discrete approximations based on the definition

of curvature. These methods do not use an intermediate analytical fit of the surface. Discrete methods often approximate an integral equation around a vertex by a summation of contributions attributed to each face or edge adjacent to the vertex.

2.3.1.1 Differential-based Methods

Differential surface properties are used heavily in the previous work to describe the local characteristics of the surface at point, either in the form of first order differential properties representing face normal or tangent plane, or in the form of second order differential properties representing principal curvatures, or in the form of third order differential properties representing directional derivatives of the principal curvatures [11].

In [26], a generalized feature extraction approach is proposed for 2D and 3D objects in mesh representation whenever the feature to be calculated can be written as a signed sum of features of the elementary shape (triangle in the 2D case and tetrahedron in the 3D case), and the feature of the elementary shape can be derived in an explicit form. As a proof of concept, mesh area (for 2D objects) and volume (for 3D objects) are first introduced and then the work extended to calculate shape moments and Fourier transform.

A 3D feature detector is proposed in [27] to extract features from uniformly triangulated meshes by extending the difference of Gaussian, DoG, operator to non-planar surfaces instead of dealing with volumetric grids. Feature detection is done in three steps. First, the extrema of the function's Laplacian (DoG) are found across scales using a one-ring neighborhood. The scale space representation of scalar function f defined on a mesh can be built progressively as Eq. 2-21.

$$f_0 = f, \quad f_1 = f_0 * g_\sigma, \quad f_2 = f_1 * g_\sigma, \dots \qquad \text{Eq. 2-21}$$

where g_σ is the Gaussian kernel defined as Eq. 2-22.

$$g_\sigma(x) = \frac{e^{-(-x^2/2\sigma^2)}}{\sigma\sqrt{2\pi}} \qquad \text{Eq. 2-22}$$

and the discrete convolution operator $*$ is defined as Eq. 2-23, Eq. 2-24.

$$(f * k)(v_i) = \frac{1}{K} \sum_{v_j \in N(v_i)} k(|\left|\overrightarrow{v_i v_j}\right||) f(v_j) \qquad \text{Eq. 2-23}$$

$$K = \sum_{v_j \in N(v_i)} k(|\left|\overrightarrow{v_i v_j}\right||) \qquad \text{Eq. 2-24}$$

Convolved functions are subtracted in order to obtain the difference of Gaussian operator, Eq. 2-25.

$$DoG_1 = f_1 - f_0, \quad DoG_2 = f_2 - f_1, \ldots \qquad \text{Eq. 2-25}$$

Second, the extrema are thresholded, by selecting the top 5% of the maximum number of vertices, sorted by magnitude. Third, the unstable extrema are eliminated, thus retaining those mesh locations exhibiting some degree of cornerness using Hessian operator.

Mean curvature is used in a different framework [28] that concerned about describing the 3D surface in 2D image, named shape image. This way, the problems of detecting features, building descriptors and matching surfaces are reduced from 3D space to 2D space and hence become simpler. Shape image, the proposed 2D representation, is developed in [28] using conformal mapping which create a conformal surface representing image, I_C, calculated by minimizing the harmonic energy over the surface. A one-to-one correspondence is supposed to exist between the vertices in the surface patch, M, and the vertices in its conformal image, I_C. Shape attributes at each vertex of the surface patch, M, are used to interpolate and compute the corresponding attributes values at each pixel of the conformal image.

Conformal representation surface $S(u,v)$ is represented by conformal factor function, $\lambda(u,v)$, and mean curvature function, $H(u,v)$. Figure 2-6 shows the human neocortical surface (a) and its corresponding mean curvature (b) and conformal factor (c). The composite shape image is shown in (d), [28]. In order to extract feature points from the shape image, a diffusion-based algorithm is used to extract distinctive features which are then used for matching purpose.

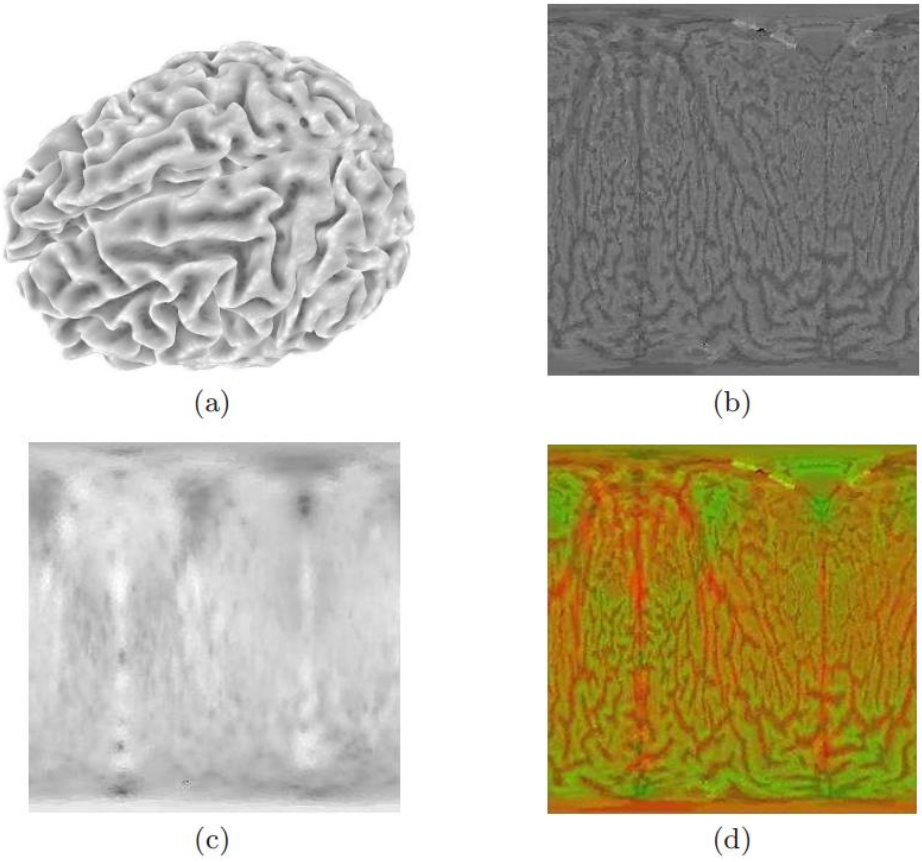

Figure 2-6: Shape image example, [28]

In order to avoid the blurring effect of Gaussian smoothing, inhomogeneous linear diffusion filtering, Eq. 2-28, is used instead in [28] to generate a set of diffusion images at different scales.

$$\frac{\partial I_i}{\partial t} = div\left(g\left(\left|\nabla f_{I_i}\right|^2\right)\nabla I_i\right) \quad \text{Eq. 2-26}$$

where div is the divergence operator, ∇ is the gradient operator, f_{I_i} is the original image, I_i is the resulting image at a specific scale and g is a common diffusivity which combines information from the conformal factor function, $\lambda(u, v)$, and the mean curvature function, $H(u, v)$. Peronal-Malik's model is used as a numerical solution for Eq. 2-28. To extract extrema, difference of diffusion is used instead of difference of Gaussians, Eq. 2-29.

$$DoD^{t_i} = I^{t_{i+1}} - I^{t_i}, \quad i = 1,2,\dots,n-1 \quad \text{Eq. 2-27}$$

Feature points are identified as local minima/maxima of the DoD images across scales, which are done by comparing each pixel in the DoD images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected

as a candidate key-point. This algorithm is carried out through all the channels of the vector image: DoD_i, $(i = 1, \ldots, m)$. The maximum and minimum which are found in every channel are considered as the interest points, which are used then for matching purpose.

Local shape variations are used in a multi-scale feature extraction algorithm to sample a representative set of feature points to be used for registration of 3D surfaces [29]. Shape index, one of the representations for local shape variations, is used to estimate local curvature at a point on the surface, Eq. 2-15. Local shape variation at a point is estimated by calculating the standard deviation of the shape index values of the points' neighbors, Eq. 2-28, Eq. 2-29, based on the assumption that the distribution is Gaussian, and points on the surface with high local shape variations are selected as feature points. The size of a point's neighborhood is defined as its scale, which is practically chosen to be the number of rings surrounding the point.

$$\sigma_{N_p} = \sqrt{\frac{\sum_{p_j \epsilon N_p} (S_I(p_j) - \mu_{N_p})^2}{n}} \qquad \text{Eq. 2-28}$$

$$\mu_{N_p} = \frac{\sum_{p_j \epsilon N_p} S_I(p_j)}{n} \qquad \text{Eq. 2-29}$$

Local shape variation is also used for detecting and extracting line-type features on point-sampled surfaces at multiple scales [30]. Surface variation parameter is calculated at each point as a ratio of Eigen-values, Eq. 2-31, of the covariance matrix, C, constructed from the k-nearest neighbors of that point, Eq. 2-30, and as long as the parameter k changes, they could obtain the parameter value at different scale. In these equations $\bar{p}$ is the centroid of the local neighborhood, and $\lambda_0 \leq \lambda_1 \leq \lambda_2$ are the Eigen-values of the covariance matrix. An automatic scale selection method is used to determine feature weights by selecting the strongest local maximum in the surface variation at all points across the scale axis. The feature weight is then calculated as the summation of number of times that the surface variation exceeds a certain threshold across all the scales in order to account for the persistence of features over all scales.

$$C = \frac{1}{k}\begin{bmatrix} p_{i_1} - \bar{p} \\ \vdots \\ p_{i_k} - \bar{p} \end{bmatrix}^T . \begin{bmatrix} p_{i_1} - \bar{p} \\ \vdots \\ p_{i_k} - \bar{p} \end{bmatrix} , i_j \in N_p \qquad \text{Eq. 2-30}$$

$$\sigma_n(p) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \qquad \text{Eq. 2-31}$$

A similar method is presented in [31] to extract feature lines from a surface point cloud. The work classifies the surface behavior at point to one of four patterns, named crease, border, corner or junction, and typical surface pattern. The crease pattern consists of crease lines that either terminates in junctions or singleton ends or they close to form a

loop. The border pattern consists only of border loops. Input points that lay on a crease are called crease points, points on the border loops are border points. At a junction the corresponding data point is called a corner or junction point and at singleton ends the corresponding data point is called end points. In order to classify the points, the algorithm uses the centroid and the correlation matrix of a set of neighbors around the point to define a correlation ellipsoid that adopts the general form of the neighbor points. Based on the eigenvalues of the correlation matrix, they estimate the curvature, k, and defined four penalty functions for each point; the curvature penalty function that encapsulate the curvature information, Eq. 2-32, the crease and the border vector valued penalty functions which encapsulate the information of how well the eigenvalues fit to the crease case or the border case, Eq. 2-33, Eq. 2-34, and a corner penalty function, Eq. 2-35, where $\lambda_0 \leq \lambda_1 \leq \lambda_2$, and $\{e_0, e_1, e_2\}$ are the covariance matrix Eigenvalues and Eigenvectors respectively. Given these penalty functions, a minimum spanning pattern is computed on a subset of the neighbor graph and the short branches of the minimum spanning pattern are removed in a second step to construct the surface feature lines.

$$\omega_k(p) = 1 - \frac{k(p)}{k_{max}} \qquad \text{Eq. 2-32}$$

$$\vec{\omega}_{cr}(p) = \frac{\max\{\lambda_1(p) - \lambda_0(p), |\lambda_2(p) - (\lambda_0(p) + \lambda_1(p))|\}}{\lambda_2(p)} . e_2(p) \qquad \text{Eq. 2-33}$$

$$\vec{\omega}_b(p) = \frac{|\lambda_2(p) - 2\lambda_1(p)|}{\lambda_2(p)} . e_2(p) \quad \text{Eq. 2-34}$$

$$\omega_{co}(p) = \frac{\lambda_2(p) - \lambda_0(p)}{\lambda_2(p)} \quad \text{Eq. 2-35}$$

Automatic detection of intrinsic geometric features is also addressed in [32] using faces normal vectors. A novel notion of relative edge strength is introduced which indicates whether two edges are likely to compose a feature curve. The concept is extended to define strong curves and strong vertices that are used then in the detection algorithm.

Geometric snakes have been used in [33] to extract feature lines based on normal variation of adjacent triangles. This system requires user interaction to specify an initial feature curve, which is then evolved under internal and external forces and re-projected onto the surface using a local parameterization.

2.3.1.2 Integral-based Methods

Despite differential surface properties play a central role in geometry processing, they are sensitive to noise and cannot be directly computed on multiple scales. On the other hand, integration of geometric functions over the neighborhood diminish or eliminate these obstacles. Recently, integral invariants have been proved to be more stable, less sensitive to noise and can be directly computed and used on muliple scales [34].

Integral invariants were first introduced in [35], [36] along with integral distance and area invariants as examples, to show the superior performance of integral invariants on noisy data, especially for the reliable retrieval of shapes. This open the door to define different integral invariants like integral volume invariants which were proposed in [4], [5], [34], [37] for local analysis of surfaces in 3D space.

Integral invariants are defined by integrating spatial functions over moving domains centered at surface points [4]. As we are dealing with surfaces in 3D space, we are interested in presenting proposed integral invariants descriptors in the literature that is related to surfaces in 3D space.

Volume integral invariant, $V_r(p)$, of a point p on boundary surface of a domain D is proposed in [5], [6], [34] as:

$$V_r(p) = \int_{B_r(p)} 1_D(x)\, dx \qquad \text{Eq. 2-36}$$

assuming that a surface in R^3 is the boundary of a domain D, $1_D(x) = 1$ if the point x is contained in D, and 0 otherwise. $B_r(p)$ denotes a ball with radius r and center p. In other words, $V_r(p)$ is the volume of the intersection of ball with radius r centered at p

with the domain, $B_r(p) \cap D$, Figure 2-7. If the patch of surface contained in $B_r(p)$ is planar, then the integration result will be the volume of half ball, or $V_r(p) = \frac{2}{3}\pi r^3$.

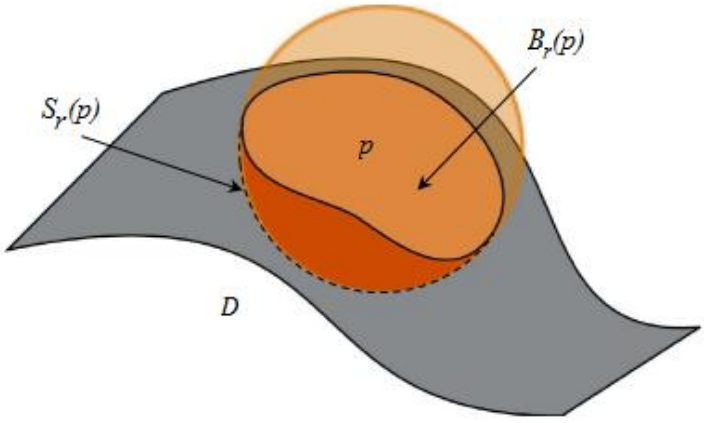

Figure 2-7: Volume and surface area integral invariant, [38]

In [4], volume integral invariants are defined as a scaled version of the previous definition, Eq. 2-36, in order to normalize the integration results, so that the planar patch around the point p will have a value of 0.5.

$$V_r(p) = \frac{3}{4\pi r^3}\int_{B_r(p)} 1_D(x)\, dx \qquad \text{Eq. 2-37}$$

Volume distance integral invariant, $VD_r(p)$, defined in [4] as the weighted integral of the squared distance function over the entire ball $B_r(p)$.

$$VD_r(p) = \frac{15}{4\pi r^5}\int_{B_r(p)} d^2(x,\varphi)\,dx \qquad \text{Eq. 2-38}$$

Surface area integral invariant, $SA_r(p)$, of a point p on boundary surface of a domain D is proposed in [34] as Eq. 2-39.

$$SA_r(p) = \int_{S_r(p)} 1_D(x)\,dx = \frac{dV_r(p)}{dr} \qquad \text{Eq. 2-39}$$

In other words, $SA_r(p)$ is the surface boundary of the intersection of sphere with radius r centered at p with the domain, $SA_r(p) \cap D$, Figure 2-7. Here, if the patch of surface contained in $SA_r(p)$ is planar, then the integration result will be the surface area of half sphere, or $SA_r(p) = 2\pi r^2$. It is worth to note, as known, that the surface area integral is the derivative of the volume integral.

Volume integral, $V_r(p)$, based on the definition in Eq. 2-36, and surface area integral, $SA_r(p)$, are related to mean curvature H at p as $r \to 0$ as in Eq. 2-40, Eq. 2-41 [34].

$$V_r(p) = \frac{2\pi}{3}r^3 - \frac{\pi}{4}r^4\,H + O(r^5) \qquad \text{Eq. 2-40}$$

$$SA_r(p) = 2\pi r^2 - \pi r^3\,H + O(r^4) \qquad \text{Eq. 2-41}$$

And volume distance integral, $VD_r(p)$, is related to the difference of the principal curvatures k_1, k_2 at p as $r \to 0$ as in Eq. 2-42 [4].

$$VD_r(p) = 1 - (k_1 - k_2)^2 \frac{r^2}{28} + O(r^3) \quad \text{Eq. 2-42}$$

Integral invariants are particularly suited for multi-scale representation since the scale is given by the radius of the kernel, r and because features smaller than radius r hardly influence the result of computation, therfore integral invariants, for relatively larger scales, are robust to noise.

To select feature points, the term point bumpiness, Eq. 2-43, is defined in [5] based on the volume integral invariant.

$$S(p) = \left[\frac{1}{N}\sum_{i=1}^{N}(V_{r_i} - \frac{2}{3}\pi {r_i}^3)^2\right]^{\frac{1}{2}} \quad \text{Eq. 2-43}$$

$$r_i = r_{min} + i.\frac{(r_{max} - r_{\min})}{N-1} \quad \text{Eq. 2-44}$$

N is selected to be 6, r_{max} is 0.1 times the average size of all fragments, $r_{\min} = r_{max}/2$. The feature points are selected as the points with bumpiness above a pre-defined hard threshold, Eq. 2-45.

$$\varepsilon_{bump} \approx \frac{1}{12N}\sum_{i=1}^{N}\frac{4}{3}\pi {r_i}^3 \qquad \text{Eq. 2-45}$$

In [4], volume and volume distance integral invariants are used with different scales in the feature points selection process. For each invariant at some scale, its range of values, say $[0, b]$, is divided into 32 levels. For each pair of levels, $l_i < l_j$, an arbitrary point whos value is in between is used in depth first search on K-nearest neighbors to cluster the points and extract connected components. A dense set of overlapping feature clusters is produced to be used in the matching process.

2.3.2 Feature Description

Given a set of feature points, a shape descriptor, representation or description, at these feature points can then be computed. Construction of shape descriptor depends very much on the representation in which the shape is given and the kind of information available, and the desired invariance properties [23].

2.3.2.1 Shape Context

The concept of shape context descriptor is introduced in [39] to describe the structure of the shape as relations between a reference point and the rest of the shape points. Given

the coordinates of a point x on the shape, the shape context descriptor at point x is constructed as a log-polar histogram of the directional vectors from x to the rest of the points, $y - x$. The descriptor is generic and applicable to any shape representation in which the point coordinates are explicitly given including mesh representation, point cloud representation or volume representation.

2.3.2.2 Mesh HOG

Mesh HOG, Mesh Histogram of Gradients, is introduced in [27] as a shape descriptor assuming the shape is defined in mesh representation and there is some given function f defined at the mesh vertices, maybe a geometric quantity such as curvature. The descriptor at point x is computed by creating a local histogram of gradients of f in the neighborhood of x.

The gradient ∇f is defined as a vector in the 3D space projected onto the 3 orthonormal planes, describing the local coordinate system at x, which is created based on the normal vector and the tangent plane. For each of the three planes, the plane is divided in 4 polar slices, starting with an origin and continuing in the direction dictated by the right hand rule with respect to the other orthonormal axis vector. When projected onto the plane, each vertex v_i falls within one of the spatial slices, Figure 2-8, (b). For each spatial slice, orientation histograms are computed with 8 bins for each of the projected

gradient vectors $\nabla f(v_i)$ of the vertices v_i that projected onto that spatial slice, Figure 2-8, (c). The final descriptor is obtained by concatenating histogram values for each of the three planes.

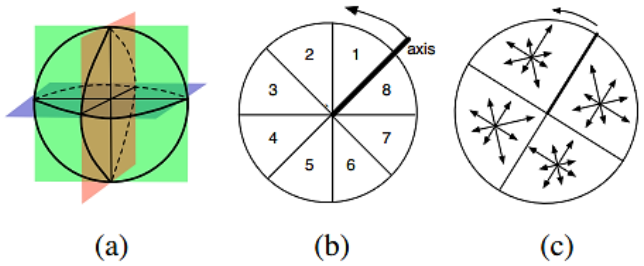

Figure 2-8: (a) Orthogonal planes, (b) 8-bin projected 3D histogram orientation slices, (c) 4-spatial polar slices with 8 orientation slices each, [27]

Like Mesh HOG, local gradient orientation histograms are used in [28] as the key entries of the descriptor. The feature descriptor is computed as a set of orientation histograms on 4×4 pixel neighborhoods, which are relative to the key-point orientation. Histograms contain 8 bins each, and each descriptor contains an array of 4×4 histograms around the key-point. This leads it to be a feature vector with 4×4×8 = 128 elements.

2.3.2.3 Heat Kernel Signatures

The heat kernel signature, HKS, is proposed in [40] based on heat diffusion equations and defined as the diagonal of the heat kernel, which reflect the stability property of point under the diffusion process. Given some fixed time values $t_1, \dots, t_n$, for each point x on the shape, the HKS is an n-dimensional descriptor vector,

$$p(x) = (K_{t_1}(x,x), \dots, K_{t_n}(x,x)) \quad \text{Eq. 2-46}$$

In order to adapt for the dependence of HKS on the global scale of the shape, a scale-invariant HKS, based on local normalization is proposed in [41].

2.3.2.4 Spin Images

The idea of spin images is introduced in [42], [43] and defined as a rotation invariant representation of surface variation that is obtained by projecting part of the surface, S, at a vertex p into the local coordinate system defined by the surface normal $\vec{n}(p)$ at p. By varying the projection area, it is possible to control the representation continuously from local to global [44]. The projection of a point q to the Spin image of p uses a reminiscent of cylindrical coordinates. That is, the x-coordinate of the Spin image corresponds to the distance from q to the line in direction $\vec{n}(p)$ passing through p and

the y-coordinate of the spin image corresponds to the signed distance from q to the tangent plane of p, Figure 2-9 [45], [46].

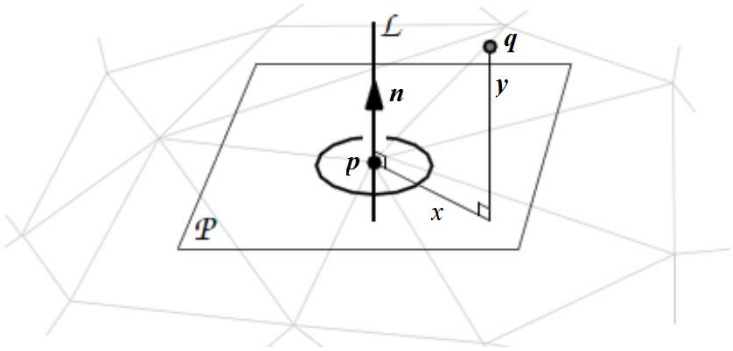

Figure 2-9: Spin image illustration, [45]

The part of S to be projected to the spin image of p depends on three parameters:

- The width w of the spin image,
- The bin size b of the spin image,
- The support angle

A vertex q of S is projected to the spin image of p if and only if:

$$||q - p|| < wbr \quad \text{and} \quad \angle(\vec{n}(p), \vec{q} - \vec{p}) < \alpha \qquad \text{Eq. 2-47}$$

where $||q - p||$ denotes the Euclidean distance between p and q, r denotes the resolution of the mesh, and $\angle(\vec{n}(p), \vec{q} - \vec{p})$ denotes the angle between the two vectors $\vec{n}(p)$ and $\vec{q} - \vec{p}$. Spin images have a variety of desirable properties, they are invariant with respect to rotation, translation, and scaling and robust with respect to noise and clutter. By adjusting w or b, Spin images can go from a local representation at p to a global representation of S. Figure 2-10 shows eight frames of an animation that motivate the name spin-image, [45]. The spin-image generation process can be visualized as a sheet spinning around the oriented point basis, accumulating points as it sweeps space.

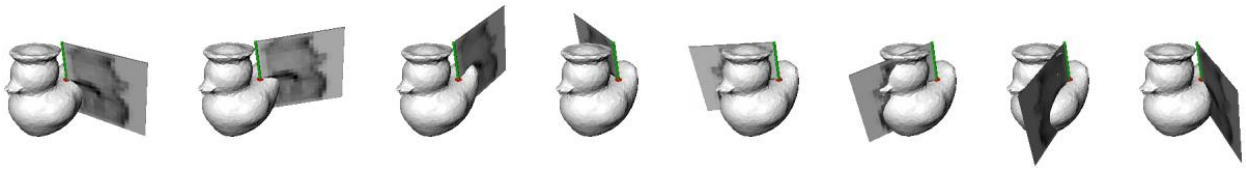

Figure 2-10: Spin image generation process, [45]

Spin images are used as local descriptors at the identified feature points for 3D registration in [29]. A modified version of spin images is proposed in [44], folded spin images, where the spin images of a canonical form of the surface are used as surface descriptors instead of using spin images of the oriented surface. This modification is done to obtain a surface descriptor that is invariant to isometric deformations. The dependence on orientation is eliminated by setting the support angle α to 180^{o} and by

folding the spin image along the median y-coordinate, such that the y-axis corresponds to the unsigned distance.

3. SURFACE MATCHING AND ALIGNMENT

Surface matching and alignment are two fundamental issues in computer vision and of both theoretical interest and practical importance as they are closely related to many applications such as shape recognition, classification, and registration. Given two surfaces in 3D space, the goals of surface registration are, to:

1. Determine whether the two surfaces are similar in terms of shape through using some similarity measure or shape distance metric, similarity problem
2. Establish correspondences between the two surfaces, correspondence problem
3. Find the rigid transformation between the two surfaces, alignment problem

3.1 Surface Matching

Surface matching is the process of determining whether two surfaces are equivalent in terms of shape, and if so, how to establish correspondences between these two surfaces. One of the important requirements of any surface matching algorithm, in addition to the accuracy and efficiency, is the robustness to noise and surface resolution. Many 3D surface matching methods have been proposed in the area of shape recognition, classification and registration including appearance-based methods, object silhouettes-based methods, exhaustive search methods, 3D correlation methods and feature-based methods. Despite the variance of the matching methods presented in the literature, feature-based methods become the most effective, and hence the most used, methods for matching partial 3D surfaces, which is the focus of the present work.

Matching of partial surfaces is a much harder problem than global matching, as it requires searching and defining the corresponding sub-parts before measuring the similarities [47]. Moreover, the similarity measures of the presented matching methods are based mainly on the global properties of the 3D models [47] such as moment invariants, Fourier descriptors, histograms and shape distributions, and harmonic based representations. These methods can easily cause similar local features of sub-parts to be misaligned and, consequently, result in an improper global similarity [47], [48].

3.1.1 Image Based Methods

Instead of matching two surfaces directly in 3D space, which may be unnecessary, computationally expensive or lack of accuracy, different image based matching methods are proposed in the literature to match partial 3D surfaces based on 2D image processing methods. To do so, appropriate geometric surface attributes, e.g. mean curvature or texture, are mapped into the parameter plane, or parametric map, through global optimization and the resulting map is a diffeomorphism, i.e., one-to-one. The resulting 2D parametric maps preserve both the shape and the continuity of the underlying surfaces, and they are stable, insensitive to resolution changes and robust to occlusion and noise [49]. Harmonic Mapping [50], conformal mapping [28], [51], [52],

[53] and least-squares conformal mapping [49] are examples of the image-based methods in the literature. Although this approach reduces all computations to the 2D settings, it is limited to 3D surface with disc-like topology [28], [49], [50], [51], [54] which makes the problems of face recognition and brain mapping be the best applications for such methods.

Matching of 3D partial surfaces can be achieved through matching of spin images [45]. The magnitude of linear correlation coefficient, R, is used to express the similarity C between two images P and Q, Eq. 3-1.

$$C(P,Q) = \left(\text{atanh}\left(R(P,Q)\right)\right)^2 - \lambda\left(\frac{1}{N-3}\right) \qquad \text{Eq. 3-1}$$

where N is the number of the overlapping pixels used in the computation, λ is a weighting factor.

3.1.2 Feature-Based Methods

Given two surfaces, their shape distance, or distance metric, is a scalar that quantifies the similarity of these two surfaces. Although there is no single definition for similarity measure exists, usually similarity measures are in some sense the inverse of distance metrics that generates large values for similar objects and either small, zero or negative

values for dissimilar objects [55]. In a feature-based scheme, typically a similarity measure between two surfaces can be achieved by combining, in a way or another, the distance metric of the surfaces feature points, e.g. the inverse of the Euclidean distance between either the positions of the feature points, Eq. 3-2, or the descriptor values of the feature points, Eq. 3-3. Point pairs whose distance above certain threshold can be considered similar. A comprehensive survey on different similarity measures presented in the literature can be found in [56]. Also a survey on shape correspondence can be found in [57].

$$dist(i,j) = \left|\left|X_i,\ X_j\right|\right| \quad \text{Eq. 3-2}$$

$$dist(i,j) = \left|\left|F_i,\ F_j\right|\right| \quad \text{Eq. 3-3}$$

Statistical approach is used in to match 3D partial surfaces using differential geometry [58], [59], [60]. Curve distance, DC, Eq. 3-4, is introduced to measure the similarity between two points from different surfaces $p_i \in S_1,\ p_j \in S_2$, and used to construct a point-pair set, PS, Eq. 3-5.

$$DC(p_i, p_j) = \frac{1}{2}\left(\frac{\left|K(p_i) - K(p_j)\right|}{\left|K(p_i) + K(p_j)\right|} + \frac{\left|H(p_i) - H(p_j)\right|}{\left|H(p_i) + H(p_j)\right|}\right) \quad \text{Eq. 3-4}$$

$$\begin{cases} PS = \{ (p_i, p_j) \mid p_i \in S_1,\ p_j \in S_2, DC(p_i, p_j) < \varepsilon c, |K(p_i)| > \sigma \} \\ \varepsilon c = \frac{1}{2}(\max(DC(p_i, p_j)) + \min(DC(p_i, p_j))) , \sigma = \frac{1}{N}\sum_{i=1}^{n} K(p_i) \end{cases} \quad \text{Eq. 3-5}$$

where $K(p_i)$ and $H(p_i)$ are the Gaussian and mean curvatures of the surface S_1 at point p_i. Plane points, whose Gaussian curvature and mean curvature are zero, are discarded before matching as they will reduce the matching efficiency.

They also introduced the similarity distance between 2 triangles $\Delta {p_1}^1{p_2}^1{p_3}^1$, $\Delta {p_1}^2{p_2}^2{p_3}^2$, Eq. 3-6.

$$DT = \frac{1}{3}\sum_{1 \le i < j \le 3} \frac{\left| |{p_i}^1{p_j}^1| - |{p_i}^2{p_j}^2| \right|}{|{p_i}^1{p_j}^1| + |{p_i}^2{p_j}^2|} \quad \text{Eq. 3-6}$$

and the absolute distance between $\Delta {p_1}^1{p_2}^1{p_3}^1$ and $\Delta {p_1}^2{p_2}^2{p_3}^2$, Eq. 3-7.

$$DA = 2\frac{\sum_{i=1}^{3}|{p_i}^1{p_i}^2|}{\sum_{1 \le i < j \le 3}^{3}(|{p_i}^1{p_j}^1| + |{p_i}^2{p_j}^2|)} \quad \text{Eq. 3-7}$$

A triangle pair set, TS, is also defined, Eq. 3-8.

$$TS = \begin{cases} (\Delta\, {p_1}^1 {p_2}^1 {p_3}^1, \Delta\, {p_1}^2 {p_2}^2 {p_3}^2) \mid ({p_i}^1 {p_i}^2) \in PS, DT < \varepsilon t \\ \varepsilon t = (\max(DT) + \min(DT))\ /2 \end{cases} \quad \text{Eq. 3-8}$$

For a specified transformation T_0 in PS, they defined the scoring function for similar triangle pair, Eq. 3-9.

$$Score(T_0) = \sum S(\Delta\, {p_1}^1 {p_2}^1 {p_3}^1, \Delta\, {p_1}^2 {p_2}^2 {p_3}^2, T_0) \quad \text{Eq. 3-9}$$

$$\begin{cases} S(\Delta\, {p_1}^1 {p_2}^1 {p_3}^1, \Delta\, {p_1}^2 {p_2}^2 {p_3}^2, T_0) = \begin{cases} 1 & DA(T_0(\Delta\, {p_1}^1 {p_2}^1 {p_3}^1, \Delta\, {p_1}^2 {p_2}^2 {p_3}^2) < \varepsilon a \\ 0 & \text{others} \end{cases} \\ \varepsilon a = \dfrac{1}{2}(\max(DA) + \min(DA)) \end{cases} \quad \text{Eq. 3-10}$$

The optimal transformation is the transformation with the maximum score, Eq. 3-11.

$$Score(T_P) = \max\big(Score(T)\big) \quad \text{Eq. 3-11}$$

In [5], fractured surface matching is accomplished in two stages. An initial matching is done by calculating the difference between the values of volume integral invariants over different scales, Eq. 2-36, calculated at selected feature points, Eq. 2-43, of the two surfaces and a difference less than a specified threshold is selected for the next step. In Eq. 3-12, $\left({p_1}^{j_1}, {p_2}^{j_2}\right)$ is an initial matching point pairs if they satisfy the condition.

$$\begin{cases} \frac{1}{N}\left|\sum_{i=1}^{N} V_{r_i}\left({p_1}^{j_1}\right) - \sum_{i=1}^{N} V_{r_i}\left({p_2}^{j_2}\right)\right| < \varepsilon v \\ \varepsilon v = \frac{1}{25\ N}\sum_{i=1}^{N}\frac{4}{3}\pi {r_i}^3 \end{cases} \quad \text{Eq. 3-12}$$

To reduce huge number of initial matchings and discard initial matching outliers, local surface similarity or similarity of surface patches is used and expressed as a collection of geometric compatibility constraints, distance constraints, Eq. 3-13, normal vector constraints, Eq. 3-14, and uniqueness constraints, Eq. 3-15. The surface patch of a point p is defined as its neighbor surface, experimentally selected as 0.2 times the average size of all fragments. Assuming that the surface patches of an initial point pair have M matching points pairs, each point $\left({p_1}^{j_{1m}}, {p_2}^{j_{2m}}\right)$ should satisfy the compatibility constraints to set the initial point pair $\left({p_1}^{j_1}, {p_2}^{j_2}\right)$ as true point pairs.

$$\begin{cases} |d_1 - d_2| < \varepsilon_{dist} \\ d_1 = ||{p_1}^{j_{1m}} - {p_1}^{j_1}|| \\ d_2 = ||{p_2}^{j_{2m}} - {p_2}^{j_2}|| \\ \varepsilon_{dist} = 2.5 f_{res} \end{cases} \quad \text{Eq. 3-13}$$

where f_{res} is grid precision.

$$\begin{cases} |\alpha_1 - \alpha_1| < \varepsilon_n \\ \alpha_1 = {n_1}^{j_{1m}} \cdot {n_1}^{j_1} \\ \alpha_2 = {n_2}^{j_{2m}} \cdot {n_2}^{j_2} \\ 0 \le |\alpha_1 - \alpha_1| \le 2 \\ \varepsilon_n = 1 \end{cases} \quad \text{Eq. 3-14}$$

$$\begin{cases} \nexists \left({p_1}^{j_{1m}} = {p_1}^{j_1}, {p_2}^{j_{2m}} \neq {p_2}^{j_2}\right) \\ \nexists \left({p_2}^{j_{2m}} = {p_2}^{j_2}, {p_1}^{j_{1m}} \neq {p_1}^{j_1}\right) \end{cases} \quad \text{Eq. 3-15}$$

The optimal matching is achieved in the second stage using greedy algorithm [6], which is based on growing and combining the best correspondences, by minimizing the distance metric, Eq. 3-16, from point pair level up to build a hierarchy of 16 points correspondence, a partial correspondence. This partial correspondence is used to compute the transform (R, t) that minimizes the matching error, Eq. 3-17, which is applied then to the entire feature point set P_1.

$$dRMS^2(P_1, P_2) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\left| \left| p_1^i - p_1^j \right| \right| - \left| \left| p_2^i - p_2^j \right| \right| \right)^2 \quad \text{Eq. 3-16}$$

$$cRMS^2(P_1, P_2) = \min_{R,t} \frac{1}{n} \sum_{i=1}^{n} \left| \left| R\left(p_1^i\right) + t - p_2^i \right| \right|^2 \quad \text{Eq. 3-17}$$

A similar greedy algorithm is used in [27] in order to select the set of best matches. For each descriptor vector, F_i^1, at feature point i from surface S_1, the best matching descriptor vector, F_j^2, at feature point j from surface S_2 can be found in terms of the

Euclidean distance between these two vectors, Eq. 3-3. A cross validation is performed, by checking that best match for F_i^1 from S_2, and best match for F_j^2 from S_1, and the first candidate match is accepted.

Forward search algorithm is used in [4] after passing several consistency tests on pair of correspondences. Correspondence pairs are constructed by finding for each cluster, instead of point, in S_1 all the clusters in S_2 having the same descriptors values. Several filtering stages are applied. First, the clusters bounding curves signatures are compared and only the consistent ones are moved to the next stage. Second, feature clusters topology are used then to discard redundant and false correspondences by considering feature correspondence pairs generated by different descriptors, and confirming the correspondence pairs of feature clusters belonging to different descriptors in a bottom-to-top scheme. Overlapped clusters are considered corresponding to the same descriptor. Third, geometric and registration consistencies are added to select pair of feature correspondences. Geometric consistency is achieved by comparing distance and deviation of normal vectors between correspondences pairs. All geometrically consistent correspondence pairs are checked using local registration by minimizing the sum of squared distances between corresponding points.

3.2 Surface Alignment

Surface alignment or registration is the problem of finding the rigid transformation between two surfaces. While different methods have been proposed for finding alignment of 3D data [57], the Iterative Closest Point (ICP) algorithm [61] becomes the most well-known method and the most widely used method for fine geometric alignment of 3D models [62], [63]. ICP is generally applicable to any two objects represented by point clouds and it has been proven to converge to a local minimum with respect to the mean-square distance metric. The basic idea of ICP is that, given two shapes represented as two point clouds, the algorithm iteratively:

1. Establishes point correspondences given the current alignment of the data
2. Computes a rigid transformation based on the established correspondence, [62]

The main limitation of ICP is that it is guaranteed to find a locally, not globally, optimal alignment as it is a local optimization method and therefore is only effective when the initial position of the input shapes is close to the correct alignment [6].

Various ICP variants have been proposed in the literature and they are classified in [63] based on:

1. Selecting source points
2. Matching
3. Weighting the correspondences

4. Removing outliers
5. Assigning error metric
6. Minimizing the error metric

The proposed ICP variants differ based on selection of some points from the input point clouds, and they range from using:

1. All the input points, [61]
2. Uniformly subsampled points, [64]
3. Randomly sampled points, [65]
4. Normal-space sampled points, [63] (to Ensure that samples have normal distributed as uniformly as possible)

In terms of differences in matching, different techniques are proposed.

1. Closest point, [61] (stable, slow)
2. Normal shooting, [66] (good for smooth meshes)
3. Closest compatible point, [67], [68], [69], [70] (based on features, e.g. normal, color,…)
4. Projection, [71], [72]

In terms of differences in weighting the correspondences, assigned weights are based on:

1. Inter-point distance, [67]
2. Compatibility of normal [63], colors [67]
3. Noise characteristics, [63]
4. Covariance matrices, [62]

In terms of differences in removing outliers, they range from removing:

1. None, [61]
2. The worst percent pairs, [70], [73]
3. Pairs with distance above threshold, [63]
4. Pairs on mesh boundaries, [64]
5. Inconsistent pairs with neighboring pairs, [71]

In terms of assignment and minimization of error metric, most of the variants minimize the sum of squared distances between corresponding point using closed form solutions to determine the rigid transformation (R, t) that minimizes the error [62].

There exist other ICP variant methods that have been proposed in the literature for accelerating the algorithm using kd-trees, closest point caching, graphics processing unit (GPU) or parallelization as the algorithm is of quadratic complexity [62].

One of the most interesting variants of ICP is the one proposed in [62], Anisotropic ICP (A-ICP). This is because all the priori proposed variants of ICP and the original ICP assume the isotropic zero-mean Gaussian noise. A-ICP proposed in [62] generalize the algorithm such that it can cope with anisotropic localization errors by assuming that the localization error in each point is normally distributed with zero-mean and a covariance matrix, Eq. 3-18.

$$\Sigma_{\vec{p}} = V_{\vec{p}} S_{\vec{p}}^2 V_{\vec{p}}' \quad \text{Eq. 3-18}$$

$$S_{\vec{p}}^2 = diag\left(\sigma_{\vec{p},1}, \sigma_{\vec{p},2}, \sigma_{\vec{p},3}\right) \quad \text{Eq. 3-19}$$

where the columns of $V_{\vec{p}}'$ are the principal axes of the localization error and $S_{\vec{p}}^2$ is a diagonal matrix with $\sigma_{\vec{p},i}$ representing the standard deviation along the principal axis i, [62].

Assuming independence of localization errors of points, the covariance matrices $\Sigma_{\vec{x}}, \Sigma_{\vec{y}}$ of any two points $\vec{x}, \vec{y}$ can be incorporated, Eq. 3-20, to determine the anisotropic weighted distance between the two points, Eq. 3-21, and hence to find the closest point, Eq. 3-22, Eq. 3-23.

$$\Sigma_{\vec{x}\vec{y}} = \Sigma_{\vec{x}} + \Sigma_{\vec{y}} \quad \text{Eq. 3-20}$$

$$W_{\vec{x}\vec{y}} = w\,\Sigma_{\vec{x}\vec{y}}{}^{-\frac{1}{2}} \quad \text{Eq. 3-21}$$

$$d_{new}(\vec{x},\vec{y}) = \left|\left|W_{\vec{x}\vec{y}}(\vec{x}-\vec{y})\right|\right|_2 \quad \text{Eq. 3-22}$$

$$C_{new}(\vec{x},Y) = \arg\min_{\overrightarrow{y_i}\in Y} d_{new}(\vec{x},\overrightarrow{y_i}) \quad \text{Eq. 3-23}$$

where w is normalization constant.

And assuming that the principal components of the localization error in each point are independent and normally distributed with zero means for both shapes, the A-ICP algorithm iteratively minimizes the anisotropically weighted fiducial registration error, Eq. 3-24, Eq. 3-25.

$$FRE^2_{weighted} = \sum_{i=1}^{N} \left|\left|W_i\left(R\overrightarrow{x_i} + \vec{t} - \overrightarrow{z_i}\right)\right|\right|_2^2 \quad \text{Eq. 3-24}$$

$$W_i = w\left(R\Sigma_{\overrightarrow{x_i}}R' + \Sigma_{\overrightarrow{z_i}}\right)^{-\frac{1}{2}} \quad \text{Eq. 3-25}$$

where $\overrightarrow{z_i}$ represents a point in Y.

4. PROPOSED METHOD

In this chapter, we put our emphasis on the details of the proposed framework, highlighted in Figure 1-3.

4.1 Template Bone Models

A statistical shape atlas, SSA, is an average mold, or a template mesh, that captures the primary shape variation of a bone [11], along with its statistical modes of variations. A proposed method for creating a SSA and generating dense correspondence across populations for complex anatomical structures was developed in [11] based on the idea of active shape model. SSA is constructed by applying principal component analysis (PCA) on of a set of similar shape instances, or training shapes. This method utilized scale space and multi-resolution registration to ensure accurate correspondence between anatomical features. Figure 4-1 shows different training pelvis bones used as input to the SSA creation procedure, and the resulting mean or template bone along with one mode of variations. In the present study, we used femur [14], humerus, pelvis, and skull template bone surface models [12], [13].

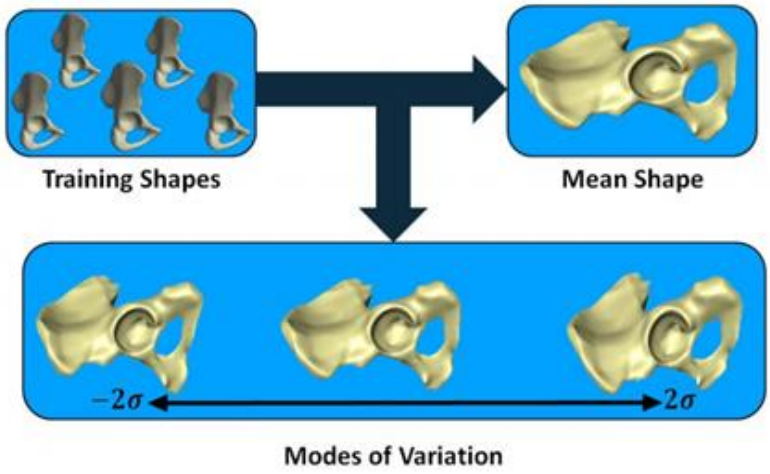

Figure 4-1: Pelvis statistical shape atlas, [11]

4.2 Surface Models Generation

Figure 4-2 shows the generation process of bones surface models. The bone fragments were first CT scanned using $0.625 \times 0.625 \times 0.625\, mm^3$ voxel resolution. The DICOM images acquired from the CT scans were then manually segmented, and surface models were generated though 3D reconstruction. Figure 4-3 shows example of surface models generation process, where the CT image is segmented to create a binary region of inside /outside the bone, which is used for generating the 3D surface model.

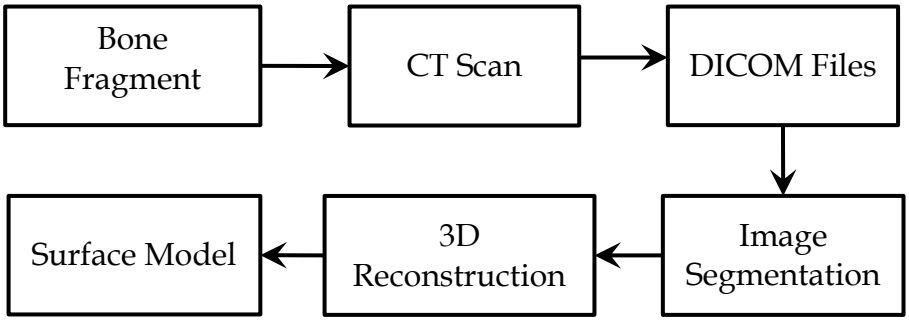

Figure 4-2: Surface models generation process

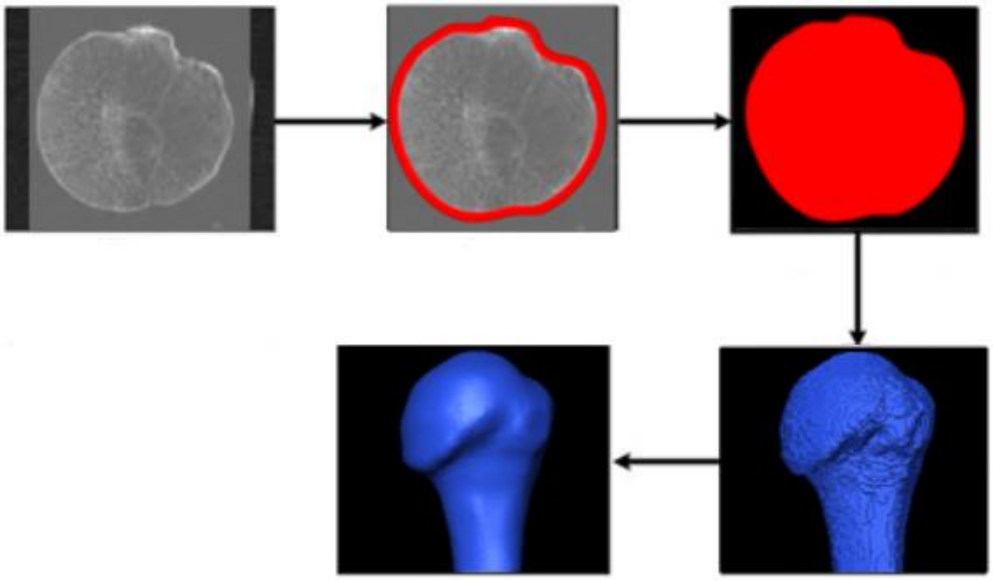

Figure 4-3: Example of surface models generation process, [11]

4.3 Feature Extraction

After generation of fragments surface models, the process of feature extraction start by calculating mesh differential properties, upon which the surface roughness is measured. Then Gaussian Mixture Model (GMM) is used to extract adaptive threshold value from

the histogram of roughness values. By applying the extracted threshold value on the surface roughness, we could extract surface features. Figure 4-4 highlights the process of feature extraction.

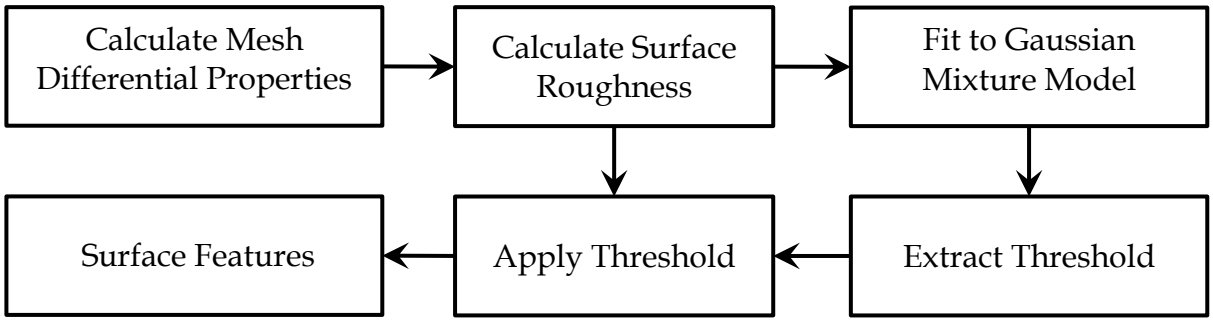

Figure 4-4: Feature extraction process

4.3.1 Mesh Differential Properties

Different methods were proposed for approximation of the surface differential properties. A valuable evaluation of different curvature estimation can be found in [74]. The evaluation showed that the Gauss-Bonett scheme gives the best results for estimation of overall curvature. Consequently, Gauss-Bonett scheme was chosen as the method for estimation of surface differential properties. Below are the details of the calculation of differentials properties using a discrete Gauss-Bonett scheme. Figure 4-5 shows the neighborhood notation used in Gauss-Bonett scheme.

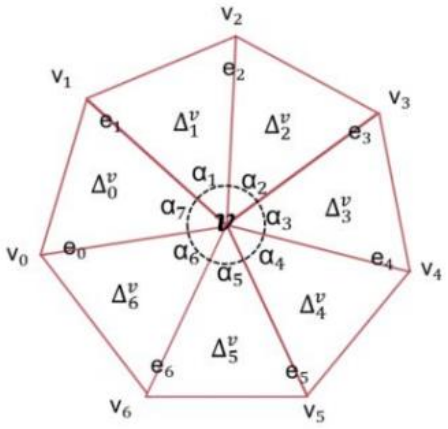

Figure 4-5: Neighborhood notation of Gauss-Bonett scheme, [11]

The following steps are used for calculating mesh differential properties using Gauss-Bonett scheme.

1. Face normal, Eq. 4-1.
2. Weighted vertex normal, Eq. 4-2.
3. Gaussian and mean curvature, Eq. 4-3, Eq. 4-4.
4. Maximum and minimum curvature, Eq. 4-5, Eq. 4-6.

$$n_f = e_i \times e_{i+1} \qquad \text{Eq. 4-1}$$

$$n_v = \sum_{f=0}^{n} n_f \,.\, \alpha_f \qquad n_v = \frac{n_v}{|n_v|} \qquad \text{Eq. 4-2}$$

$$K = \frac{2\pi \; - \; \sum_{i=0}^{n-1} \alpha_i}{A/3}, \qquad A = \sum_{i=0}^{n-1} a_i \,, \qquad a_i = \frac{\|e_i \times e_{i+1}\|}{2} \qquad \text{Eq. 4-3}$$

$$H = \frac{0.25 \sum_{i=0}^{n-1} \|e_i\| \beta_i}{A/3} \quad \text{Eq. 4-4}$$

$$k_1 = H + \sqrt{\max(0, H^2 - K)} \quad \text{Eq. 4-5}$$

$$k_1 = H - \sqrt{\max(0, H^2 - K)} \quad \text{Eq. 4-6}$$

where β_i is the deviation of normal i.

4.3.2 Surface Roughness

For the present application, curvedness showed the best results for presenting local shape variations. We have proposed a different multi-scale surface roughness measure based on weighting the variance of the point's neighbors curvature. The newly proposed measure, Eq. 4-7, is based on the curvedness, Eq. 2-14, and it forms the weights based on the point-to-neighbors edge lengths.

$$R_i = \frac{1}{|N_i|} \sum_{j \in N_i} \left| C_j - \overline{C_\iota} \right| * \frac{E_{ij}}{E_n} \quad \text{Eq. 4-7}$$

$$\overline{C_\iota} = \frac{1}{|N_i|} \sum_{j \in N_i} C_j \,, \qquad E_n = \sum_{j \in N_i} E_{ij} \quad \text{Eq. 4-8}$$

where R_i is the roughness at point i, $\overline{C_\iota}$ is the average curvedness of the neighbors, Eq. 4-8, C_j is the curvedness at neighbor $j \in N_i$, E_{ij} is the length of the edge between i, j,

E_n is the total edge length which is the summation of the edge lengths of the point i's neighbors, Eq. 4-8. The proposed measure can be computed at multiple scales by incorporating k-level neighborhood points in the equation, $k = 1,2,\ldots$ Figure 4-6 - Figure 4-9 show roughness maps of femur, humerus, pelvis and skull template bones computed at 1st, 2nd, 3rd, 4th and 5th neighborhood levels.

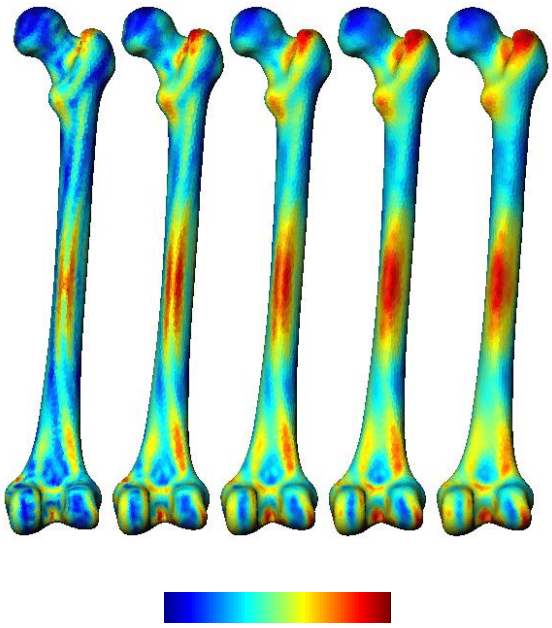

Figure 4-6: Roughness maps of femur template bone at 1st , 2nd , 3rd , 4th and 5th neighborhood levels

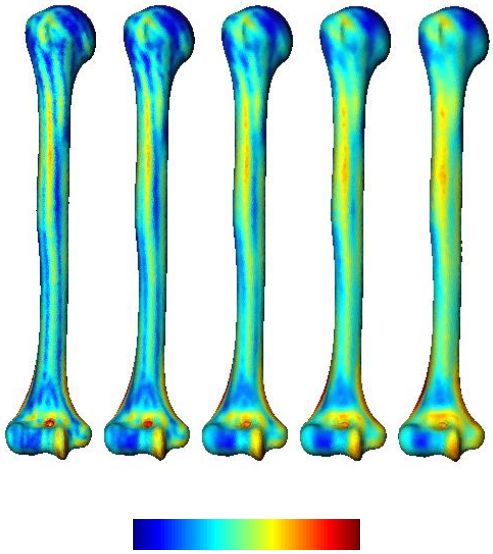

Figure 4-7: Roughness maps of humerus template bone at 1st , 2nd , 3rd , 4th and 5th neighborhood levels

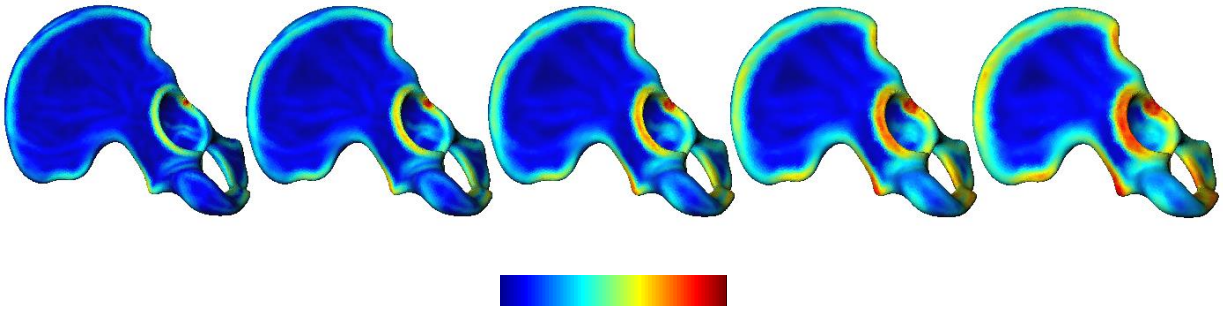

Figure 4-8: Roughness maps of pelvis template bone at 1st , 2nd , 3rd , 4th and 5th neighborhood levels

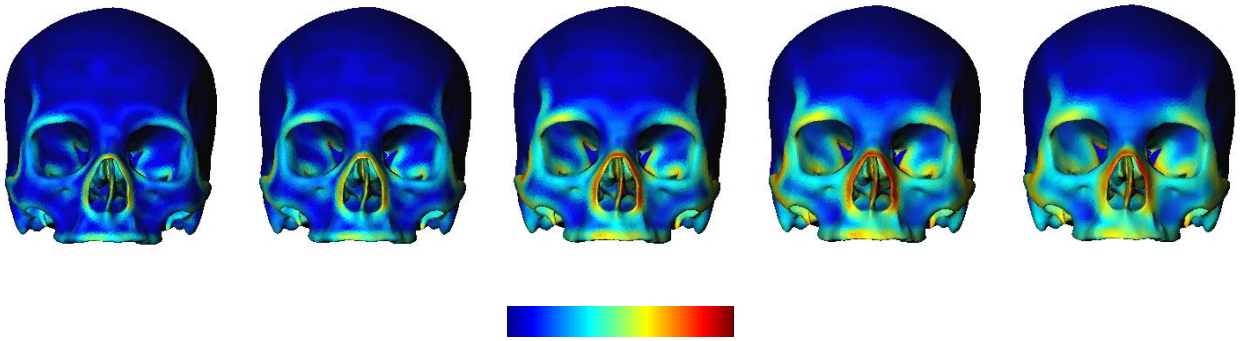

Figure 4-9: Roughness maps of skull template bone at 1st , 2nd , 3rd , 4th and 5th neighborhood levels

4.3.3 Gaussian Mixture Model

Gaussian mixture model is a parametric model used for clustering data in which analytic Gaussian density functions are fitted to the data assuming the data were drawn from number of Gaussian distributions. In our problem, the clustering problem can be formalized as, assuming that there are a finite number of roughness probability density functions in the surface model, and each roughness distribution can be modeled by one Gaussian, then the surface model roughness can be modeled by a mixture of g component Gaussian distributions in some unknown proportions, $\pi_i,\ i = 1, 2, \dots, k$. The probability density function (PDF) of a data point x can be defined, Eq. 4-9, Eq. 4-10.

$$f(x|\psi) = \sum_{i=1}^{k} \pi_i \, f(x, \mu_i, \Sigma_i) \qquad \text{Eq. 4-9}$$

$$0 \leq \pi_i \leq 1, \qquad \sum_{i=1}^{k} \pi_i = 1 \qquad \text{Eq. 4-10}$$

where ψ is a vector containing parameters $\pi_i, \mu_i, \Sigma_i,\ \ i = 1, \dots,\ k$, and μ_i, Σ_i are mean and covariance of the i^{th} Gaussian distribution, which can be described as, Eq. 4-11.

$$f(x, \mu_i, \Sigma_i) = \frac{1}{2\pi} |\Sigma_i|^{-\frac{1}{2}} e^{\frac{1}{2}(X-\mu_i)' \Sigma_i^{-1} (X-\mu_i)} \qquad \text{Eq. 4-11}$$

and π_i is the i^{th} mixing weight. Expectation Maximization (EM) algorithm can be used to find estimations for the unknown parameters ψ [75]. The EM algorithm is an iterative computation of maximum likelihood estimators when the available data can be viewed as incomplete [76]. The basic idea of the EM algorithm is to associate a complete data model to the incomplete observed data to simplify the computations of maximum likelihood estimates [76]. As the name implies, the EM algorithm consists of two steps, that are repeated alternatively up to achieve an arbitrarily small change, expectation of parameters given the current parameters estimates, and estimation of the parameters by maximizing of the likelihood of the expectation step. In [76], more details about Gaussian mixture model and EM algorithm were presented.

For the problem in hand, we're assuming the calculated surface roughness of each template model is a mixture of two Gaussian distributions, that represents distribution of surface points with low roughness values (smooth surface points) and high roughness values (feature points). Figure 4-10 and Figure 4-11 show the histogram and the two distributions of Gaussian components of femur and skull template bone models surface roughness calculated using the proposed roughness measure, Eq. 4-7, Eq. 4-8.

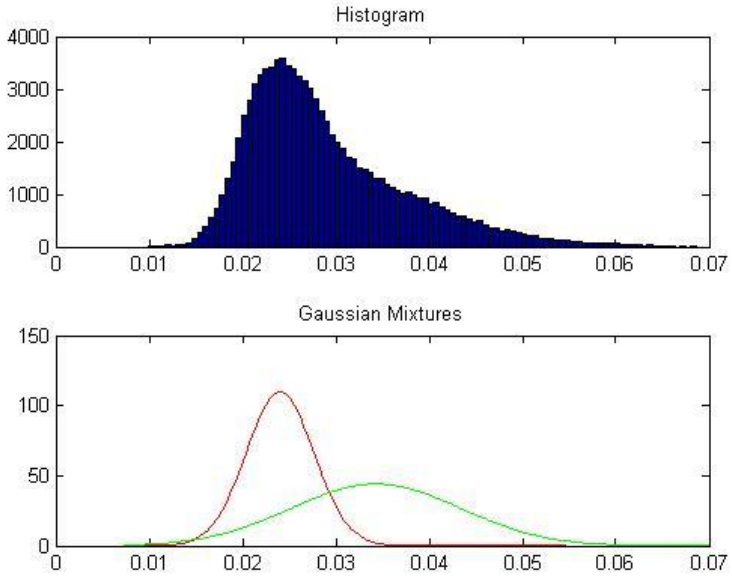

Figure 4-10: Surface roughness of femur template bone model, histogram and distributions of Gaussian components

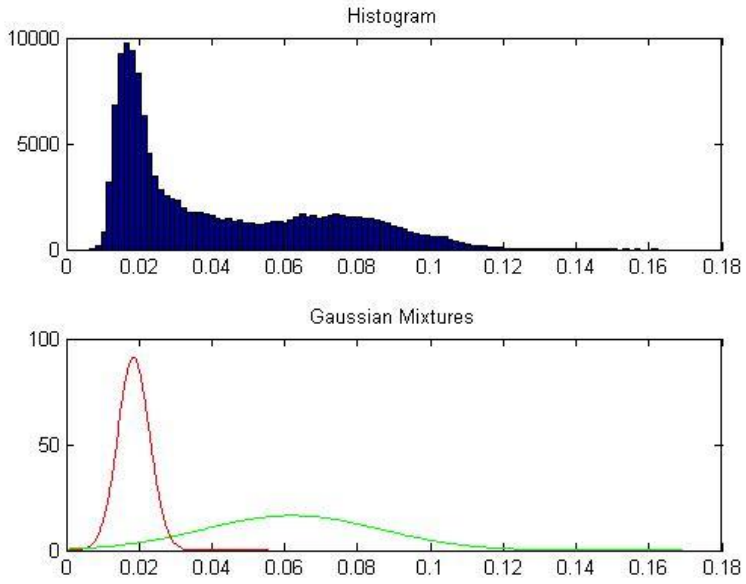

Figure 4-11: Surface roughness of skull template bone model, histogram and distributions of Gaussian components

Based on the idea that fractured surfaces introduce higher values of roughness, compared with native surfaces, we assumed that the calculated surface roughness of each bone fragment is a mixture of three Gaussian distributions, representing native surface smooth points, native surface features and fracture surfaces points. Actually, not all fracture surfaces points have high roughness values, but fracture surfaces have some points with high roughness values. Figure 4-12 shows the surface roughness of femur fragment bone model and Figure 4-13 shows histogram and distributions of Gaussian components of its surface roughness values. It is worth to note here that the features of

the original bone still can be detected by the proposed surface roughness measure, in the presence of high curvature, or high rough, points at the edge of the fracture surface.

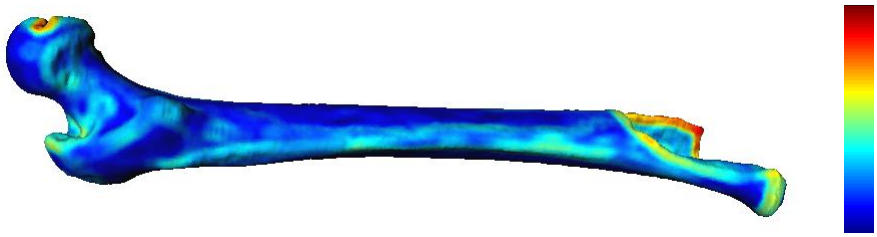

Figure 4-12: Surface roughness of femur fragment bone model

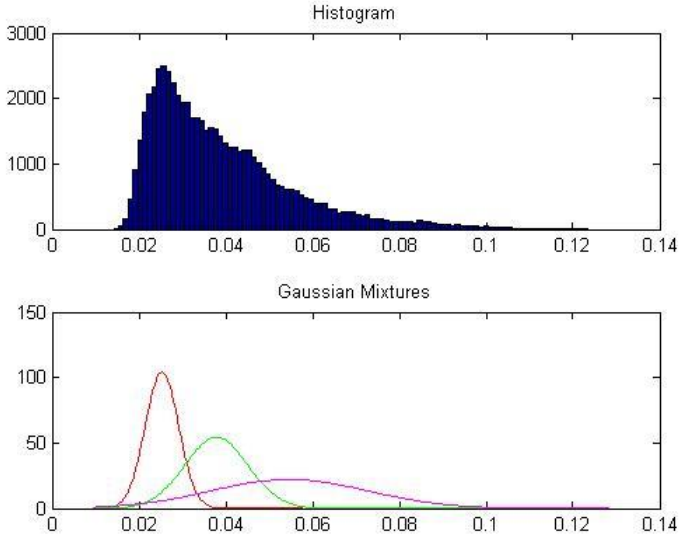

Figure 4-13: Surface roughness of femur fragment bone model in Figure 4-12, histogram and distributions of Gaussian components

By comparing Figure 4-10 and Figure 4-13, we can notice the difference between the two histograms. The histogram of the fragment bone surface roughness is spreading up to roughness value of 0.12 while that of the template bone is spreading up to 0.07 only. This means that the fragment bones introduce higher values of surface roughness to the histogram, i.e. a new range added that can be modeled by another Gaussian distribution. By taking a closer look to the second graph of the same figures, we can notice that the first two Gaussian distributions are very close. Table 4-1 shows statistics and GMM results of roughness of the template and the fragment bones surface models, which highlights our point of view.

Table 4-1: Femur template and fragment bones roughness statistics and GMM results (cm)

Model	Min	Max	μ_1	σ_1	μ_2	σ_2	μ_3	σ_3
Template	*0.010213*	*0.068350*	*0.024019*	*0.003630*	*0.034277*	*0.009047*	-	-
Fragment	*0.014785*	*0.122825*	*0.025304*	*0.003833*	*0.037813*	*0.007334*	*0.054616*	*0.018046*

We've selected the mean of the second Gaussian distribution, μ_2, resulting from GMM analysis as the threshold that filter out the features from template bone surface models. To extract similar points from the fragment bone, we've selected the range between the second, μ_2, and the third, μ_3, means as the range of roughness that include the features of the original bone, Figure 4-14 and Figure 4-15 show the extracted features from template and fragment femur bone models using GMM extracted thresholds.

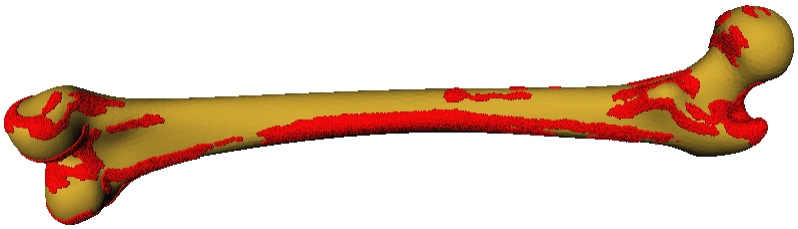

Figure 4-14: Femur template bone model features

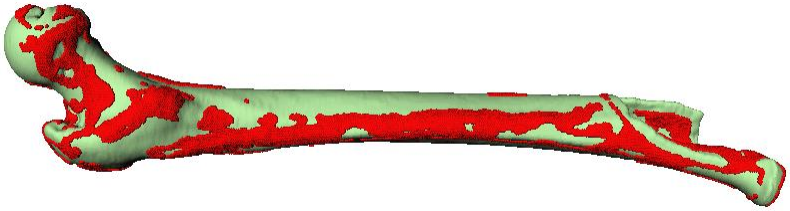

Figure 4-15: Femur fragment bone model features

4.4 Matching and Registration

The process of matching and registration is highlighted in Figure 4-16 and Figure 4-17. Basically the process is divided into two stages. The first stage involves matching of the extracted fragment features with the features of the four template models (B_i). Fragment features are positioned at different locations (t_j) with respect to the bounding box of the template model, and at each position, they are rotated around different coordinate system axes, (R_k). For each position and orientation, fragment features are matched with template features using iterative closest point (ICP).

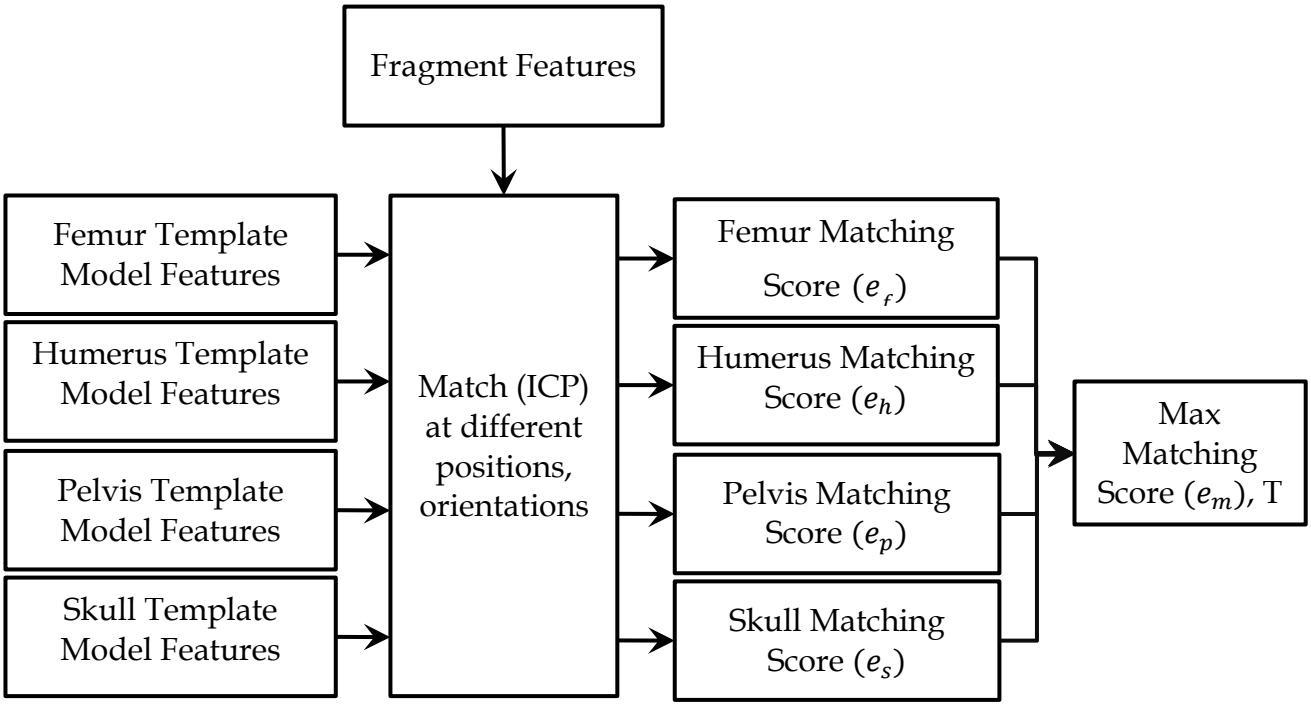

Figure 4-16: Matching and registration process – A

From all matchings, we select the best matching score (B_m, t_n, R_l) based on the minimum root-mean-square error, RMS_{ijk} and extract its rigid transformation, T. The second stage involves registration of the fragment surface model with the template surface model based on the best matching score from the first stage. The fragment model is first positioned and oriented using (t_n, R_l), transformed by T and then registered with the template model B_m using ICP. If the output error from the registration process (e_r) is greater than the output error from the feature matching process (e_m), then we are facing a false positive matching. To solve this issue, we

discard the template model with the max feature matching from the matching process and used the second best feature matching score template model to register with the fragment. If the registration error is smaller than the feature matching error, then we are done with the matching, registration and identification too.

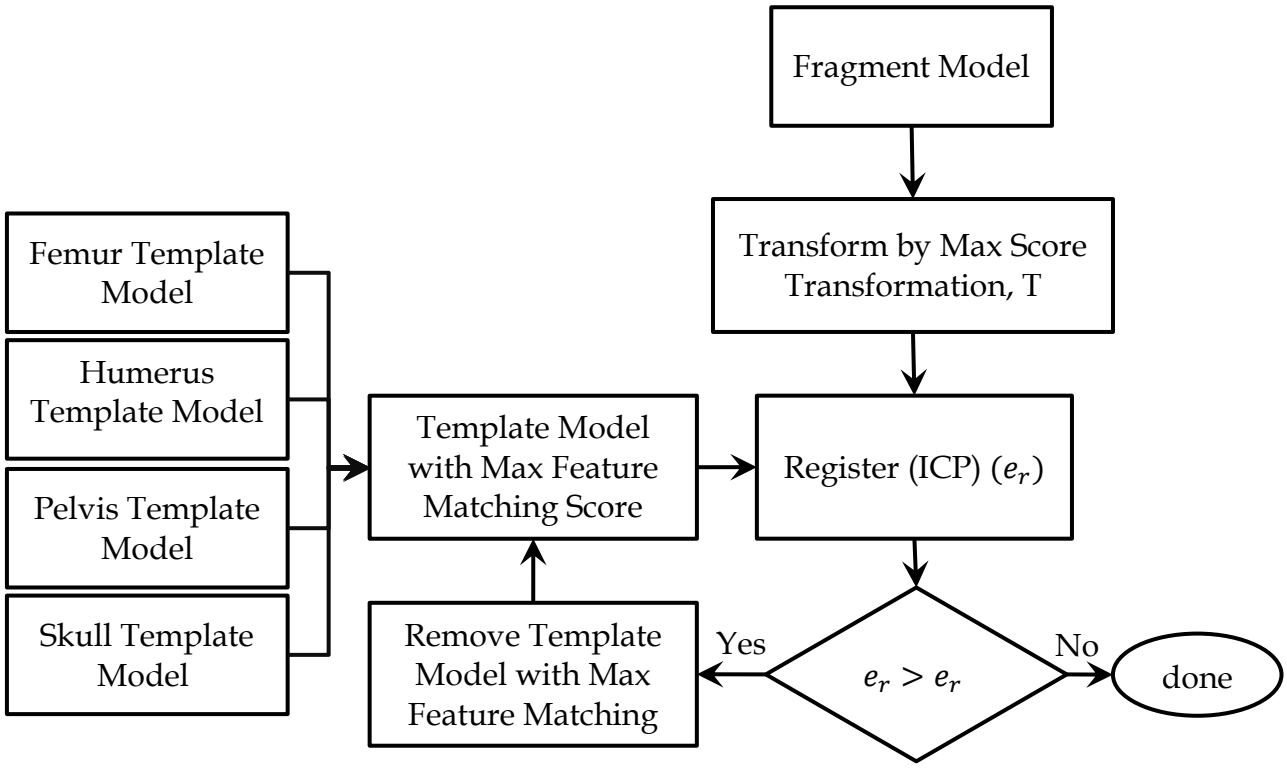

Figure 4-17: Matching and registration process – B

4.5 Virtual Environment

As part of the work, we've developed a virtual 3D environment that include surface roughness calculations, alignment using ICP, and our matching/registration procedure. The developed environment enables the user, through graphical user interface (GUI), to

load scanned models in either surface model file format (*.surf) or open inventor file format (*.iv). The GUI have the capability of loading one or multiple files at once each containing one or more models. Once the models loaded to the scene viewer, the user has the ability to:

1. Do rotation, panning and zooming through interactive viewer widget as well as wheel controls
2. Return to the home view, set the home view, view all the loaded models
3. Switch between parallel and perspective projection modes.
4. Change the lighting direction
5. Change background color
6. Select specific model for further operation

The loaded models are assigned random keys as well as random colors which can be updated through material editboxes. The user has the ability to apply transformation (translation, rotation and scaling) through user controls as well as interactive viewer. The transformations can be applied, reset or confirmed for the selected model(s). The interface enables the user to save the modified models separately. Figure 4-18 shows models loaded into the virtual environment.

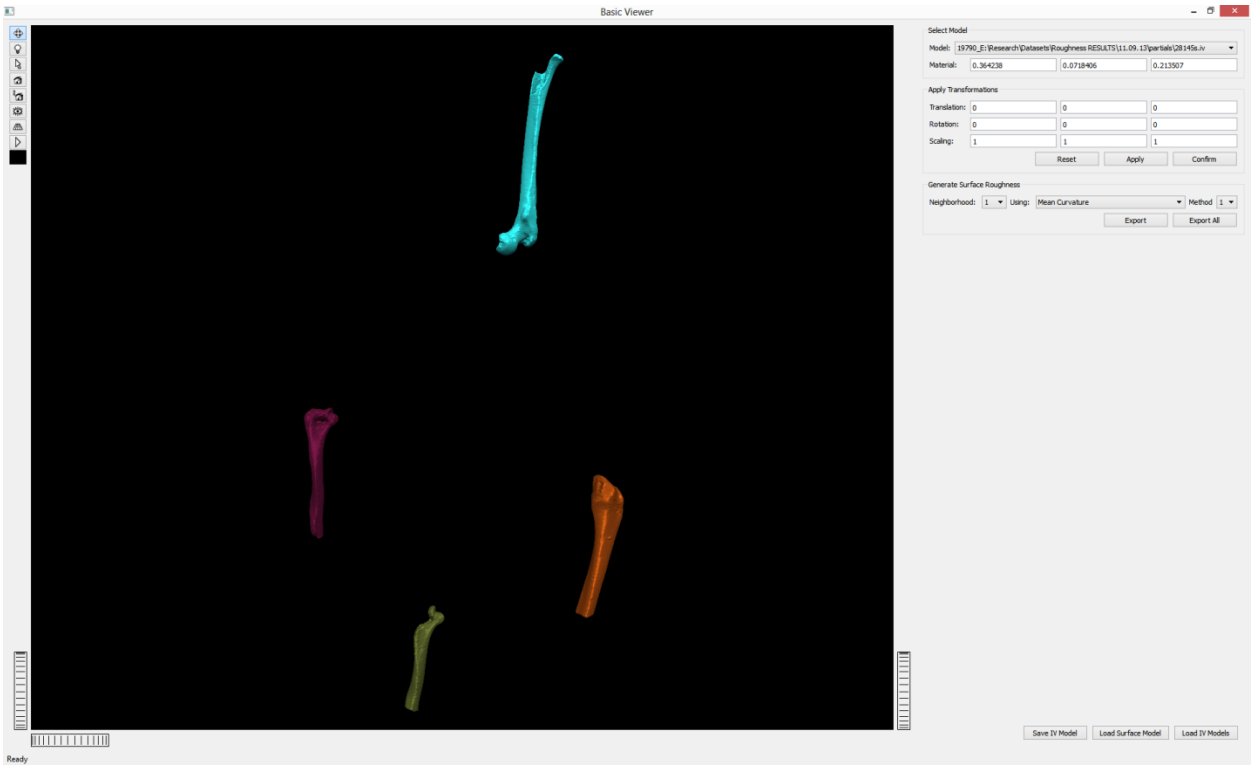

Figure 4-18: Fragment models loaded in the developed virtual environment

The GUI also provide the user the control to generate and export surface roughness values for the selected model(s) as well as all using variable neighborhood size, different local curvature measures (mean curvature, shape index, curvedness and signed curvedness) and different scale-space calculation scheme (difference-based, variance-based, edge weighted-based).

For our application, after loading the fragment models in the scene to the viewer, the user can select different fragments through select tool, right click and select “align selected models with template”. This will pop-up a small dialog to choose which

template model to align with, or the user can check "All" to test alignemnt with all template bones. Once accept, a separate alignment dialog will appear showing the selected fragments and the template model(s) loaded.

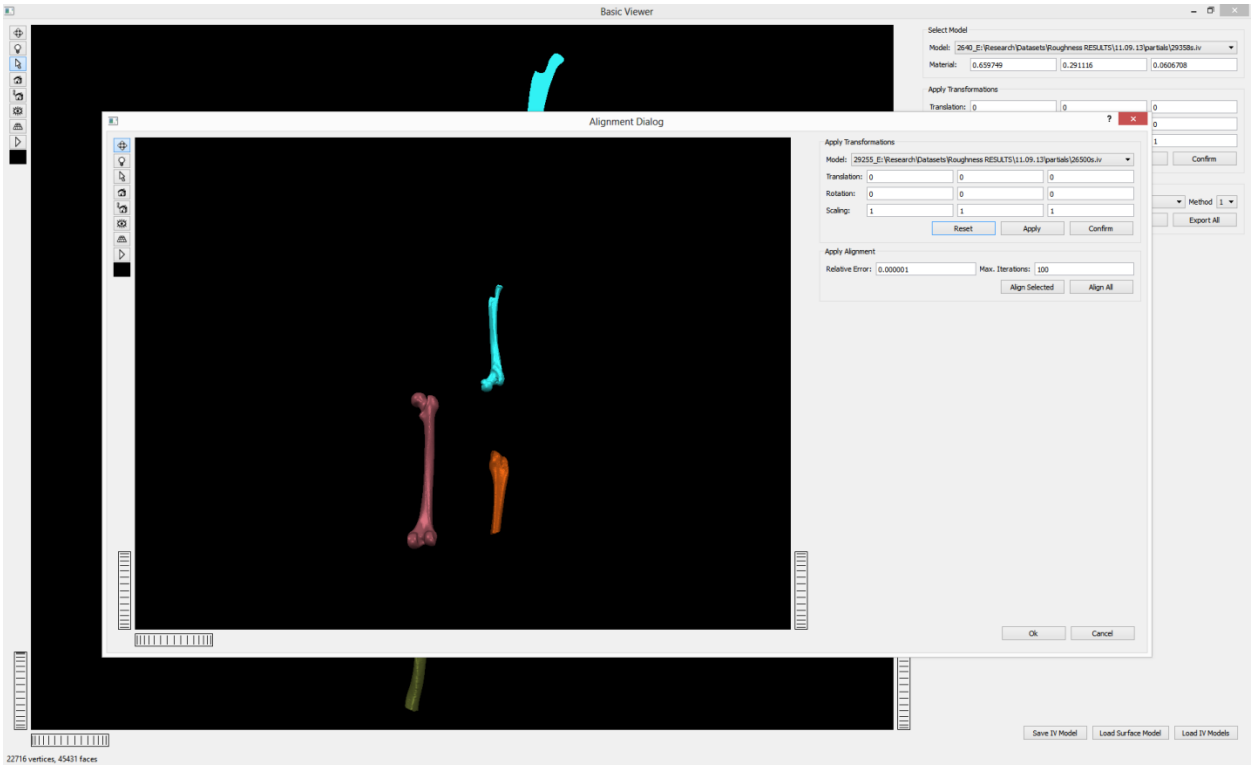

Figure 4-19: Fragment models and template models loaded in the alignment dialog

In the alignment dialog, the user has the ability to apply any transformation to the loaded fragment models as well as to the template model before starting the algorithm. Default values for maximum iterations and minimum reltive error of ICP algorithm are loaded and can be changed by the user. The dialog enables the user to either start the matching/registration procedure for the selected model(s) or for all. Once done, the

GUI will display the fragment models registered with the proper template bone. The user can accept/reject matching(s)/alignemnt(s) by right clicking at the fragment and select accept/reject selected/all. If the alignment been accpted, it will be reflected to the fragment model in the main viewer, and it will be neglected if rejected. Aain, the user can apply transformations to the fragment(s) of rejected matching(s)/alignemnt(s) and start the procedure again.

5. RESULTS AND DISCUSSION

5.1 Results

This study is part of NIJ grant (NIJ-2011-2805, SL#000944) entitled "Computerized Reconstruction of Fragmentary Skeletal Remains for Stature, Sex and Ancestry Estimation" [2]. In order to test our algorithm, sample of bone fragments from the Morton Shell Mound osteological collection served as the test sample. The Morton sample represents over 25,000 human bone fragments from approximately 125 individuals. The collection is on temporary loan to the Department of Anthropology and Middle Eastern Studies at Mississippi State University from the Museum of Natural Science at Louisiana State University [2]. Bone fragmentation in the Morton sample ranges from slight (with greater than 33% of an element represented) to high (small fragments less than 2 cm) [2]. The samples provided to this study were restricted femur and humerus fragments and classified as being slight fragmentations. We also incorporated simulated partial fragments of the template bone. This study incorporated statistical bone atlases for four skeletal elements (femur, humerus, pelvis and skull) developed in [11], and used their average models as templates.

To verify the proposed algorithm, for each fragment, matching scores against all the template bones at different positions and orientations, were listed, based on the root mean square error, where the root mean square error is calculated in centimeters as the average of distances between corresponding points. The template bone with the

minimum non-false-positive root mean square error was considered the best match, and the fragment was identified as being part of it.

Figure 5-1 shows template models, their roughness maps and detected features. Figure 5-2 - Figure 5-4 show fragment models, their roughness maps and detected features. Figure 5-5 shows simulated partial fragments. Figure 5-6 - Figure 5-15 show snapshots from the GUI after the matching and registration processes were completed.

Table 5-1 - Table 5-12 show the feature matching and registration root mean square errors (RMSE) with each template model at different orientations and positions. The rotations used in this experiment are:

1. Identity (model original orientation) *(0,0,0)*
2. *180º* around x-axis, *(180,0,0)*
3. *180º* around y-axis, *(0,180,0)*
4. *180º* around z-axis, *(0,0,180)*

and the used positions are:

1. Template model center, *C*
2. Template model minimum (min x, y, z of the bounding box), *Min*
3. Template model maximum (max x, y, z of the bounding box), *Max*

In Table 5-1 - Table 5-12, bold borders highlight the minimum root mean square error of matching fragment features with the corresponding template bone model. Red colors highlight false positive matchings which are detected by comparing feature matching errors and registration errors. Green colors highlight correct true positive matchings which are verified by comparing feature matching errors and registration errors.

It is worth to note that RMSE values listed in Table 5-1 - Table 5-12, which represent the average of distances between corresponding points, are relative measures and only used to select the best template match for the tested fragment. RMSE values depend on many factors including the number of feature points in both the fragment and the template models. If we have a correct matching and enough iterations of ICP are used in registration, RMSE value will be zero if and only if the fragment model completely and exactly coincides with template model, which is impractical.

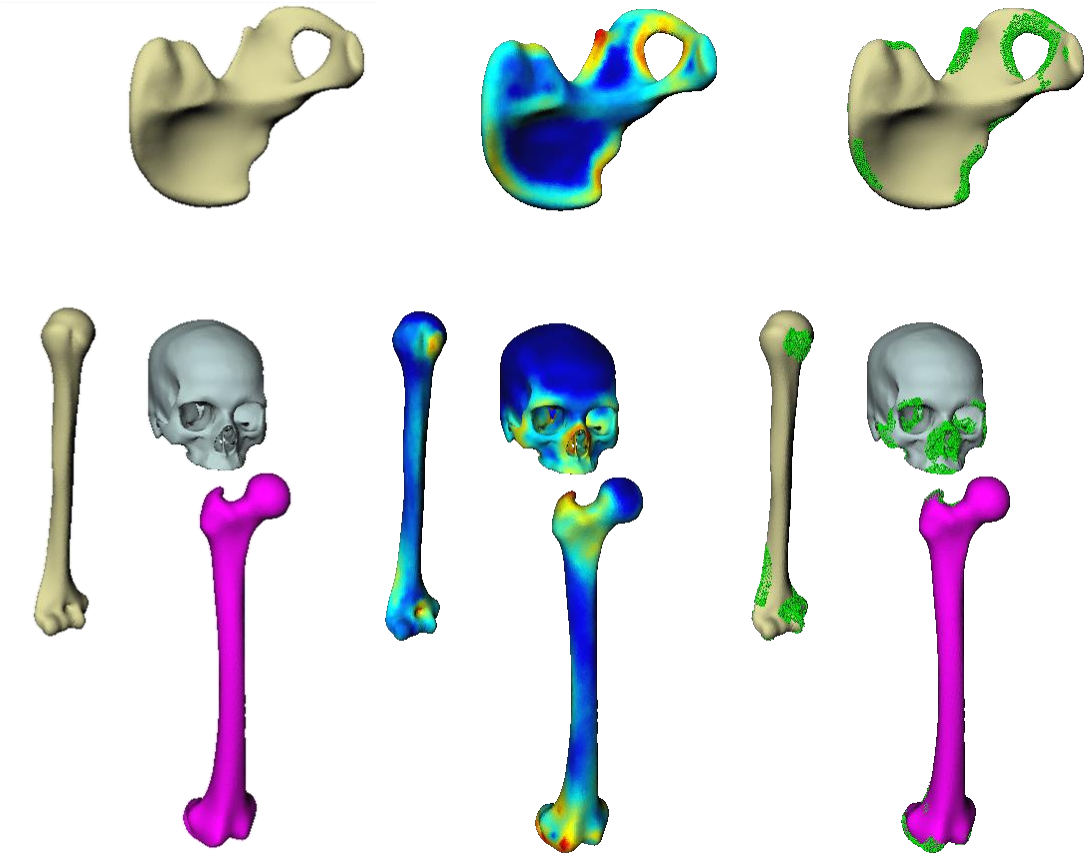

Figure 5-1: Template models, their roughness maps and detected features

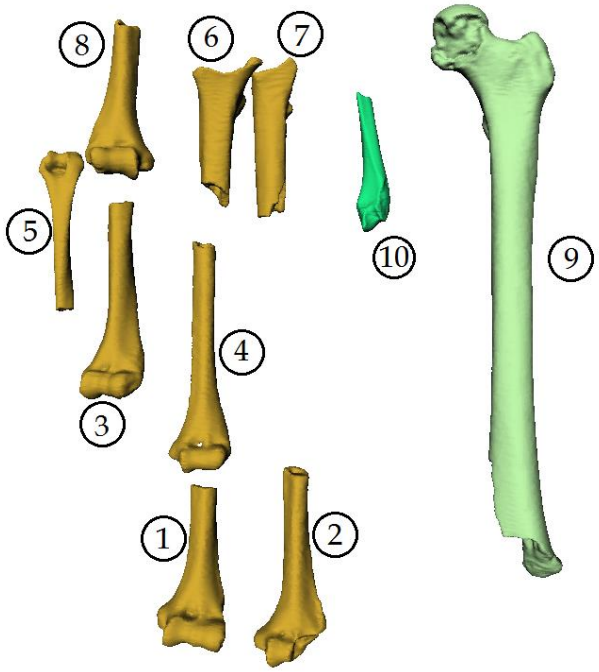

Figure 5-2: Fragment models

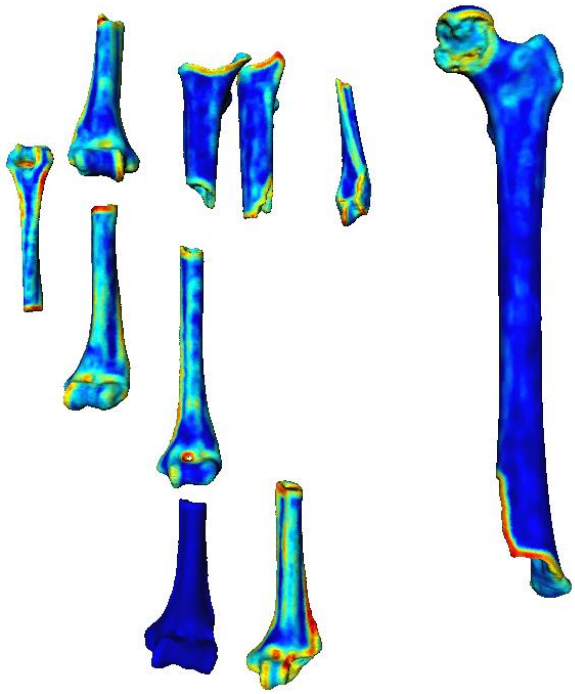

Figure 5-3: Fragment models roughness maps

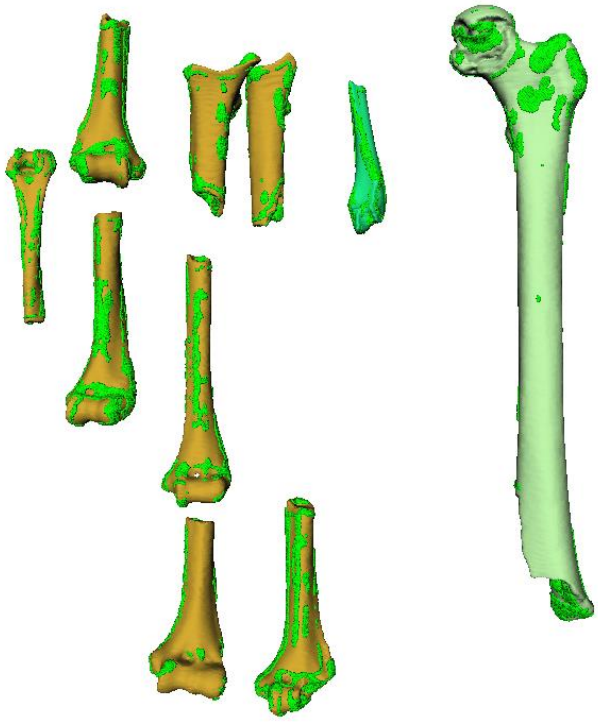

Figure 5-4: Fragment models detected features

Figure 5-5: Simulated partial fragments

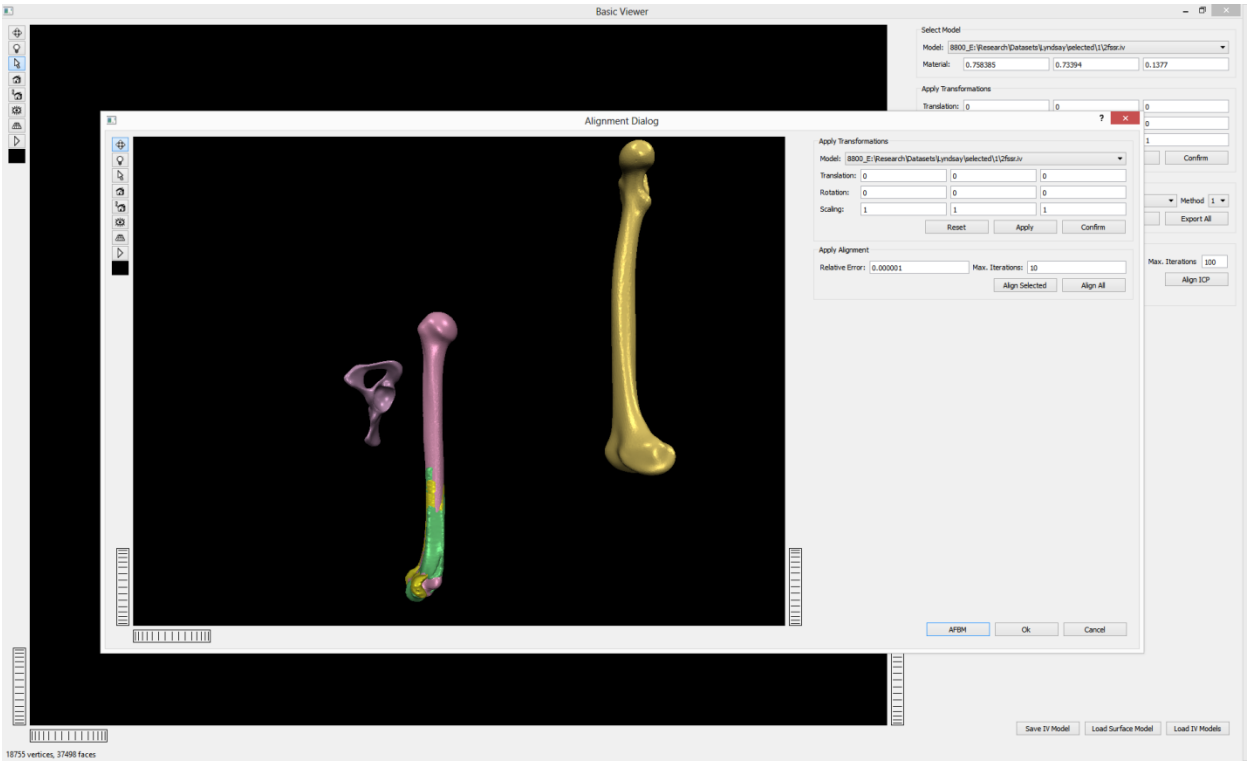

Figure 5-6: Fragments 1, 2 matched and registered, (Table 5-1, Table 5-2)

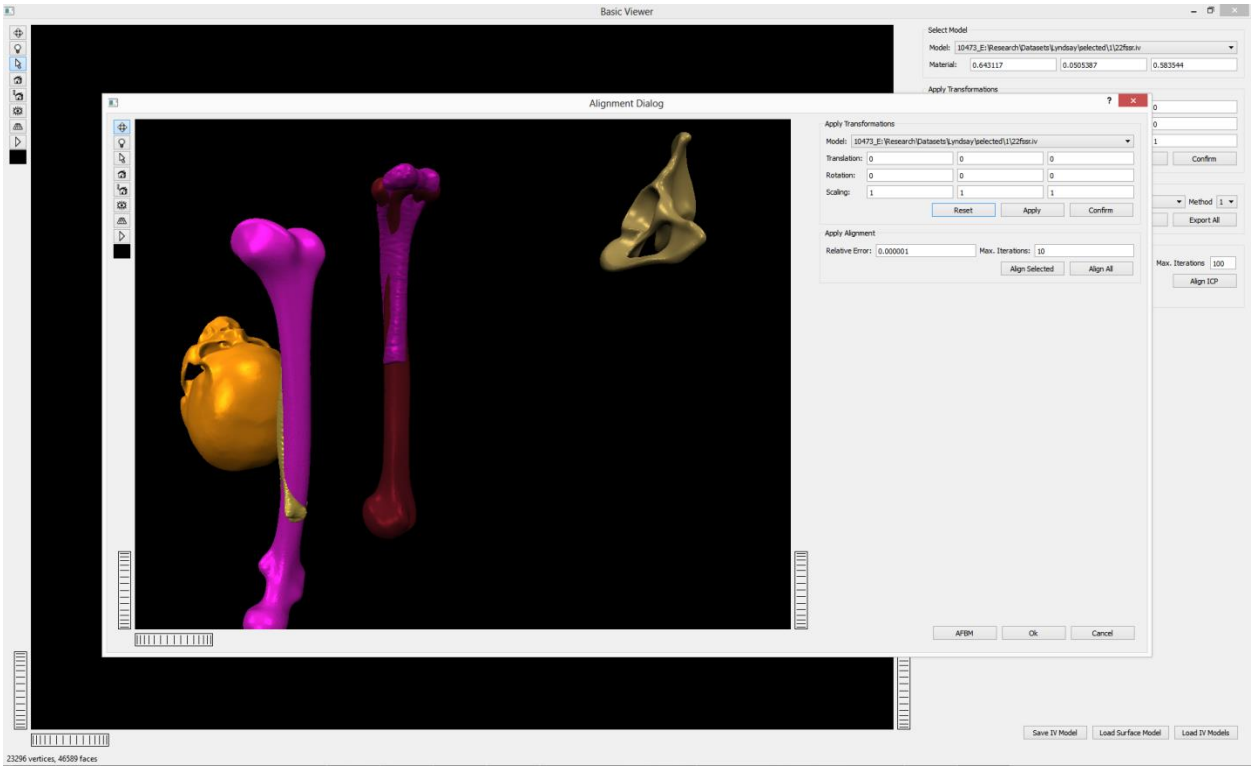

Figure 5-7: Fragments 3, 4 matched and registered, (Table 5-3, Table 5-4)

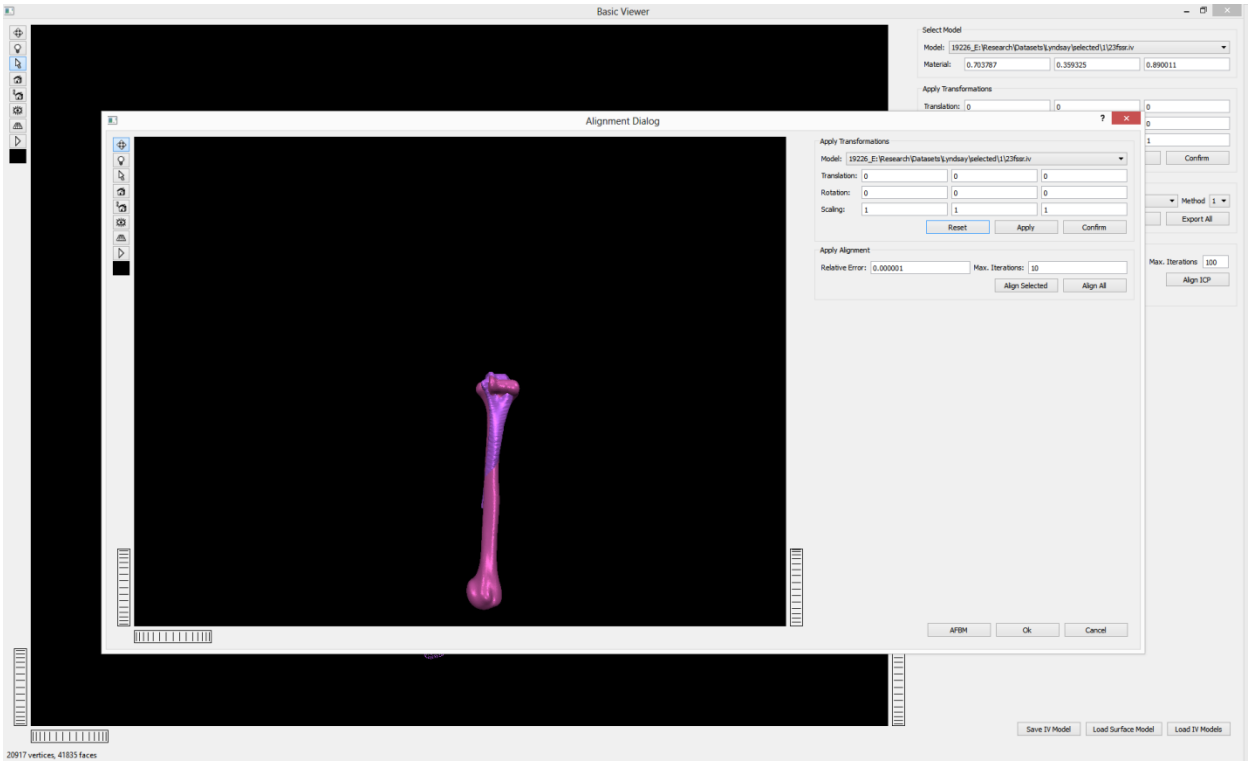

Figure 5-8: Fragment 4 matched and registered, (Table 5-4)

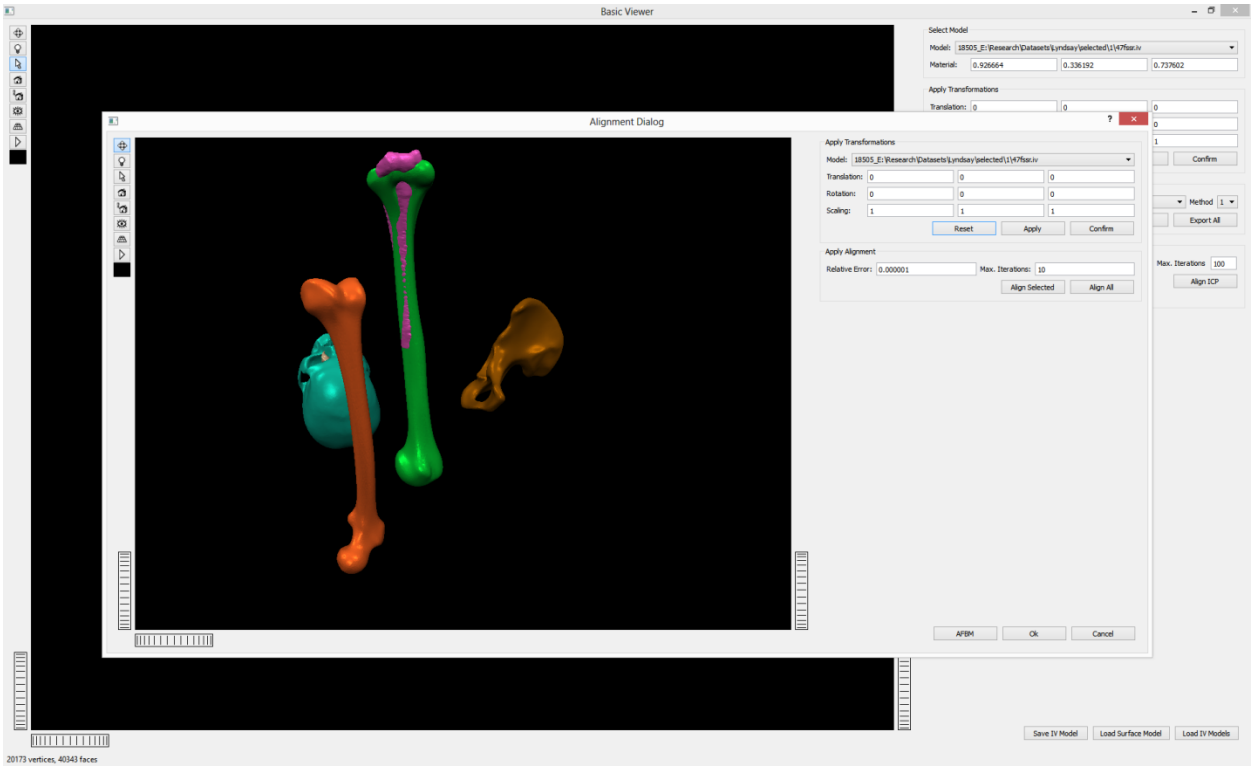

Figure 5-9: Fragments 5, 6 matched and registered, (Table 5-5, Table 5-6)

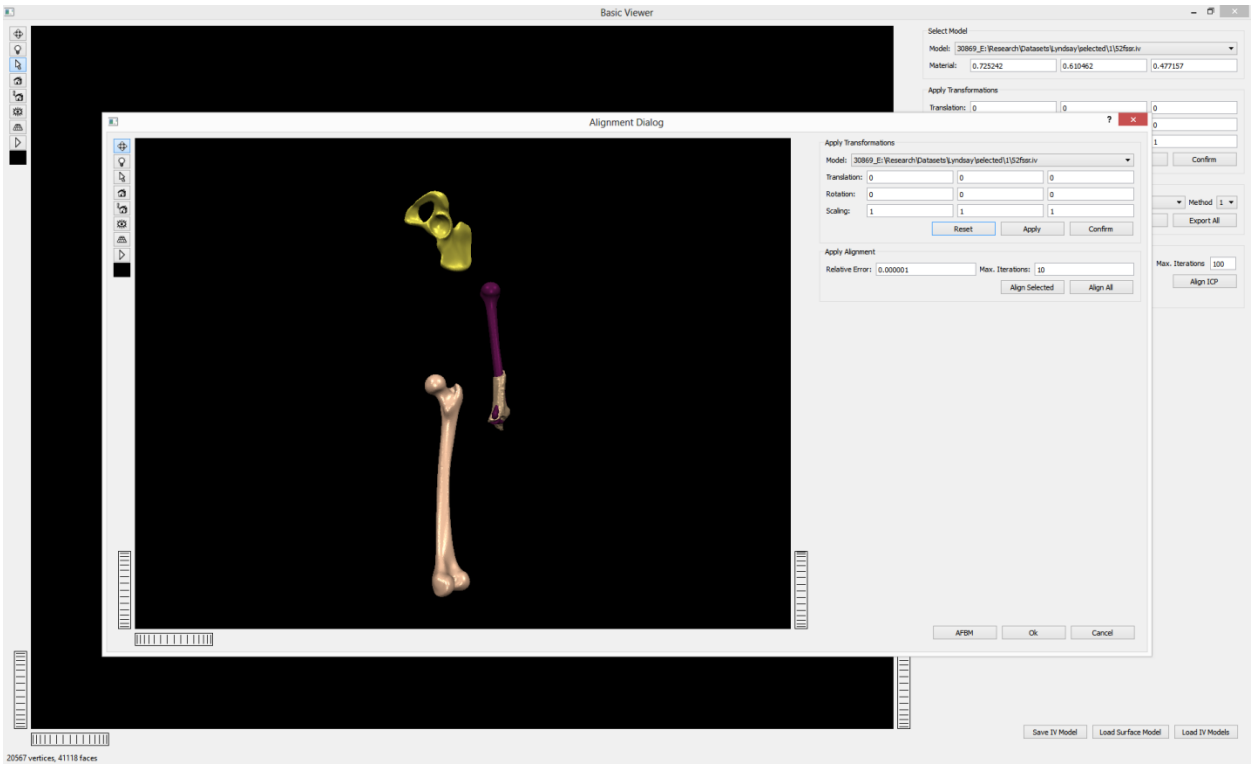

Figure 5-10: Fragment 6 matched and registered, (Table 5-6)

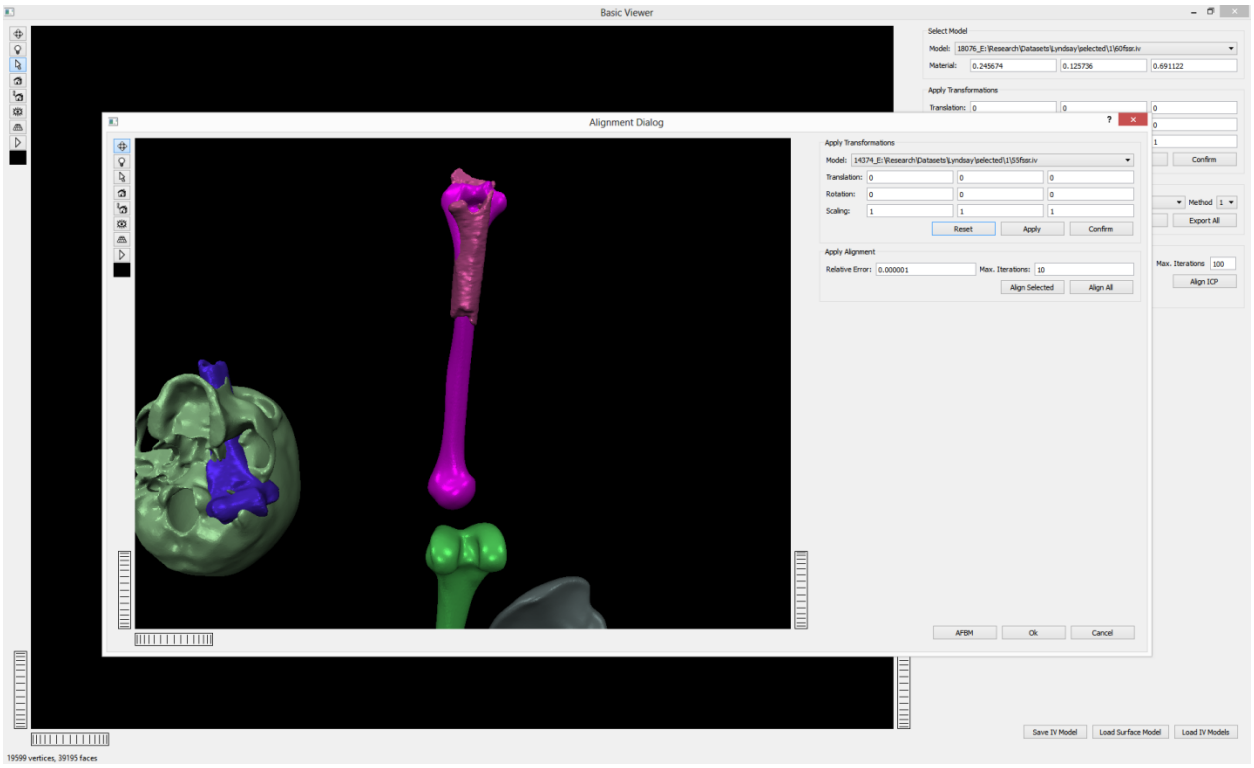

Figure 5-11: Fragments 7, 8 matched and registered, (Table 5-7, Table 5-8)

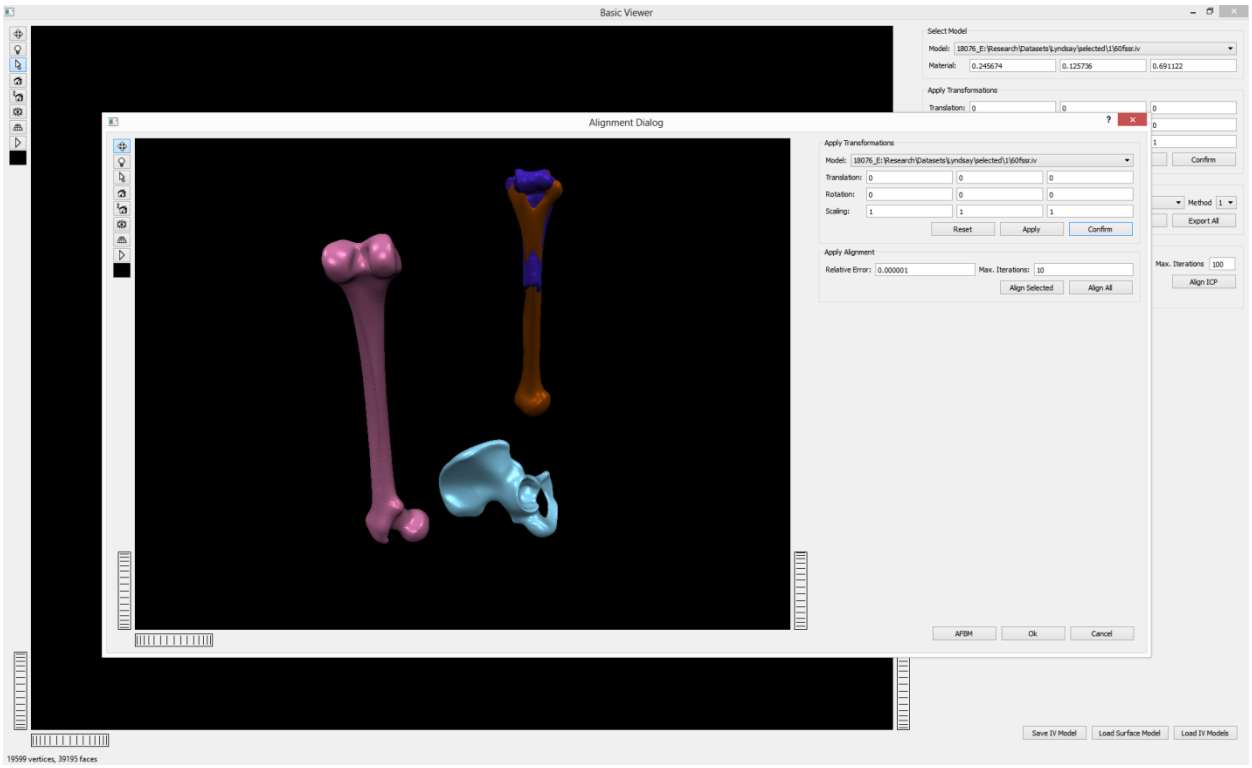

Figure 5-12: Fragment 8 matched and registered, (Table 5-8)

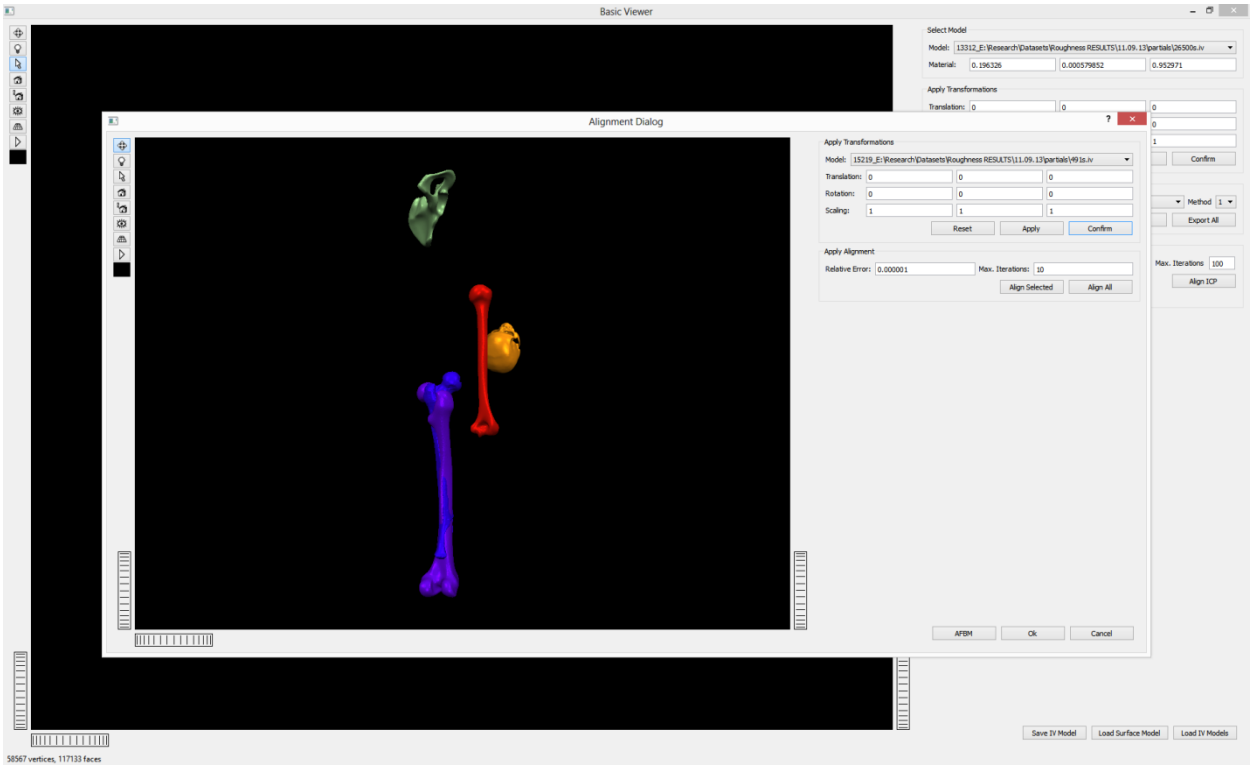

Figure 5-13: Fragment 9 matched and registered, (Table 5-9)

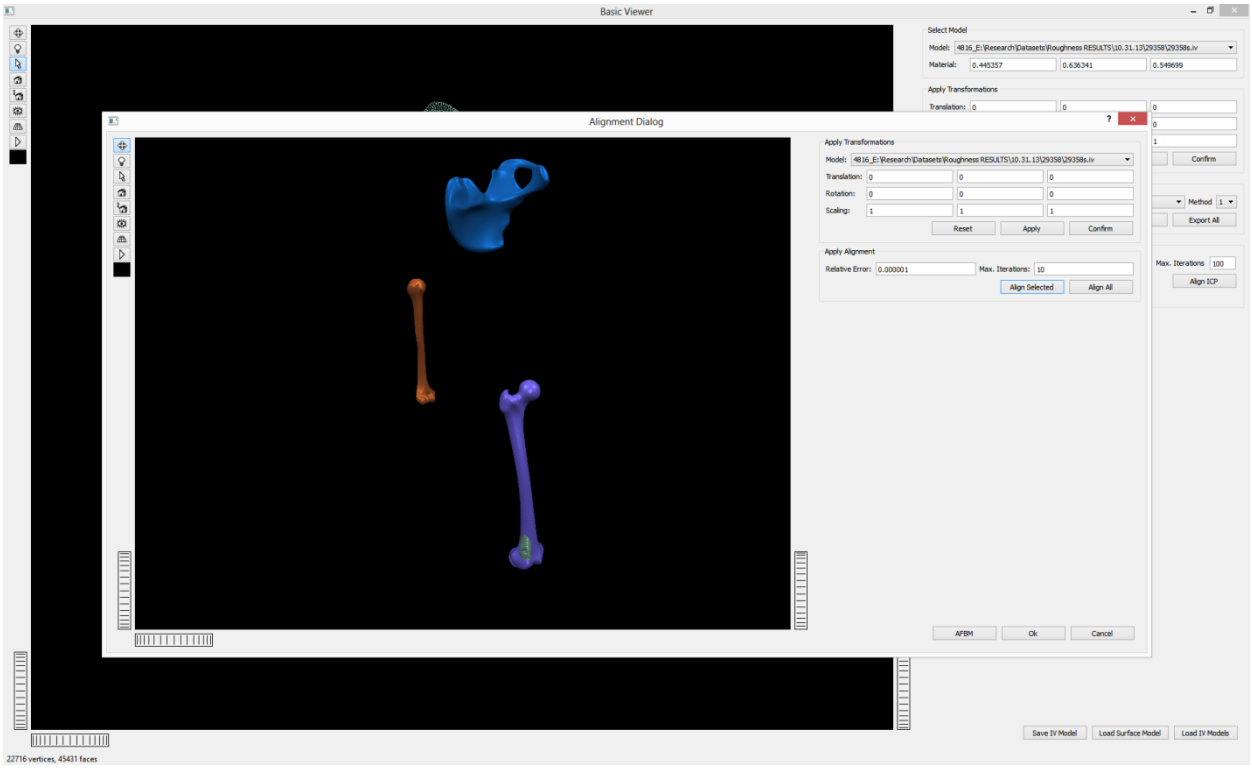

Figure 5-14: Fragment 10 matched and registered, (Table 5-10)

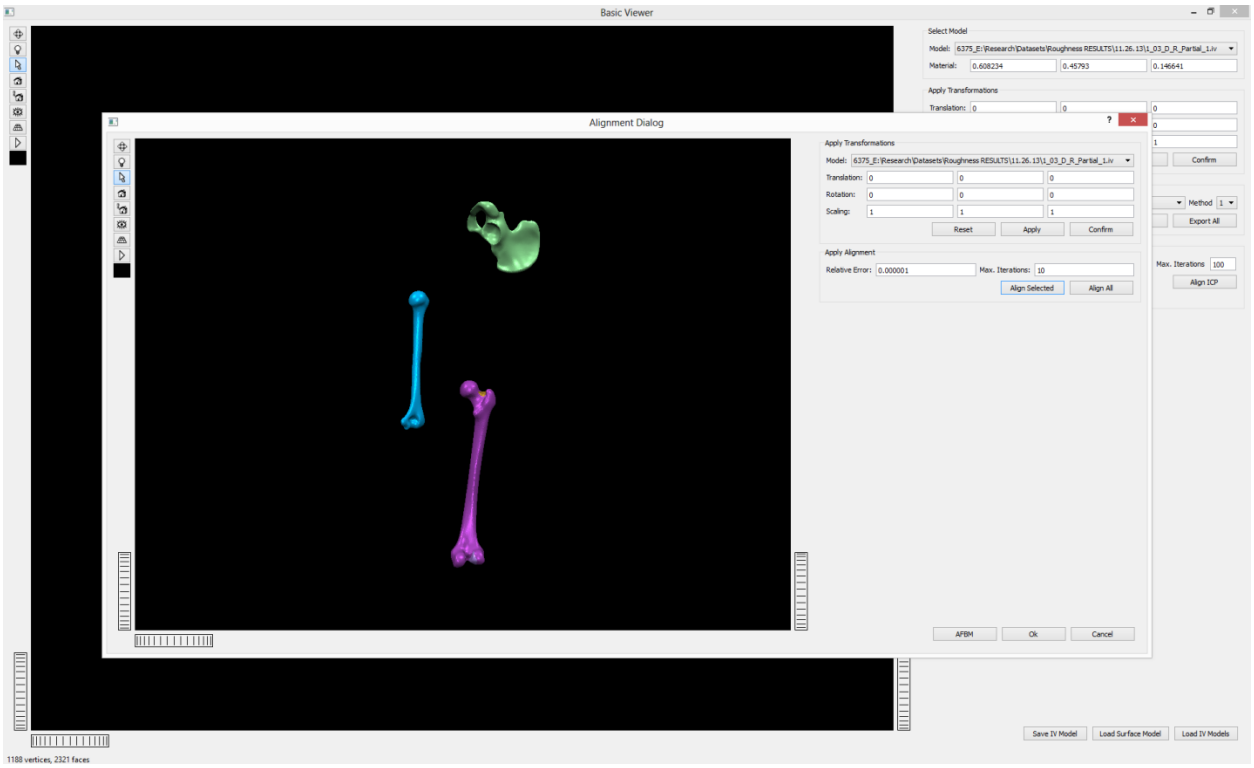

Figure 5-15: Fragments 11, 12 matched and registered, (Table 5-11, Table 5-12)

Table 5-1: Fragment 1 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*0.802575*	*0.964580*	*0.912572*	***0.385311***
	(0,0,0)	*Min*	*0.792741*	*0.560098*	*1.159475*	*0.715297*
	(0,0,0)	*Max*	*0.929298*	*1.558419*	*0.813505*	*0.877184*
	(180,0,0)	*C*	*0.733280*	*1.061273*	*0.699231*	*0.520537*
	(180,0,0)	*Min*	*0.803595*	*0.967835*	*1.086065*	*0.674283*
	(180,0,0)	*Max*	*0.792741*	*0.584711*	*1.159475*	*0.663049*
	(0,180,0)	*C*	*1.251750*	***0.514663***	*0.941809*	*0.615125*
	(0,180,0)	*Min*	*0.798748*	*0.958978*	*0.739336*	*0.535935*
	(0,180,0)	*Max*	*1.031602*	*0.963872*	*1.050825*	*0.624497*
	(0,0,180)	*C*	*1.318214*	*0.560098*	*1.159475*	*0.742322*
	(0,0,180)	*Min*	*1.134816*	*0.616592*	*0.813505*	*0.621891*
	(0,0,180)	*Max*	*0.998479*	*0.987909*	*0.757167*	*0.656431*
Registration	**-**	**-**	**-**	***0.191653***	**-**	***0.457082***

Table 5-2: Fragment 2 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*0.717382*	*0.820956*	*0.784872*	***0.455202***
	(0,0,0)	*Min*	*0.816057*	*0.639038*	*1.296141*	*0.722036*
	(0,0,0)	*Max*	*1.166123*	***0.557928***	*0.783834*	*0.698709*
	(180,0,0)	*C*	*0.667225*	*0.764418*	*0.911756*	*0.829937*
	(180,0,0)	*Min*	*0.785084*	*0.846746*	*0.987817*	*0.876161*
	(180,0,0)	*Max*	*1.184897*	*0.633950*	*1.296141*	*0.652107*
	(0,180,0)	*C*	*0.712840*	***0.557928***	*0.783834*	*0.685014*
	(0,180,0)	*Min*	*0.772428*	*0.783334*	*0.847573*	*0.766267*
	(0,180,0)	*Max*	*1.120830*	*0.837383*	*1.003036*	*0.650761*
	(0,0,180)	*C*	*1.373790*	*0.639038*	*1.296141*	*0.676959*
	(0,0,180)	*Min*	*0.710961*	*0.899850*	*0.783834*	*0.694614*
	(0,0,180)	*Max*	*1.284121*	*0.945493*	*0.932003*	*0.718372*
Registration	**-**	**-**	**-**	***0.276091***	**-**	***0.498471***

Table 5-3: Fragment 3 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*0.851570*	*0.970754*	*1.291666*	*0.698427*
	(0,0,0)	*Min*	*1.003247*	*0.772353*	*1.297800*	*0.950202*
	(0,0,0)	*Max*	*1.582501*	***0.640472***	*1.155850*	*0.755699*
	(180,0,0)	*C*	*0.873902*	*0.881022*	*1.212483*	*0.888306*
	(180,0,0)	*Min*	*0.902806*	*0.992278*	*1.164985*	*0.821211*
	(180,0,0)	*Max*	*1.003247*	*0.792260*	*1.292888*	*0.890012*
	(0,180,0)	*C*	*1.522238*	*2.051451*	*1.002017*	*0.933749*
	(0,180,0)	*Min*	*0.850877*	*0.915115*	*1.163749*	*0.971257*
	(0,180,0)	*Max*	*1.227372*	*1.064651*	*1.265988*	*0.836307*
	(0,0,180)	*C*	*1.450552*	*0.789034*	*1.381006*	*0.778849*
	(0,0,180)	*Min*	*1.569201*	***0.640472***	*0.898699*	*0.755699*
	(0,0,180)	*Max*	*1.280068*	*0.972668*	*1.068066*	*0.834491*
Registration	**-**	**-**	**-**	***0.216632***	**-**	**-**

Table 5-4: Fragment 4 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	***0.634426***	*0.758959*	*1.083199*	*0.753343*
	(0,0,0)	*Min*	*0.832352*	*0.809756*	*1.280035*	*1.151337*
	(0,0,0)	*Max*	*0.697350*	*2.206949*	*0.755857*	*1.162084*
	(180,0,0)	*C*	*0.689120*	*0.779390*	*0.719143*	*0.931962*
	(180,0,0)	*Min*	*0.844802*	*0.826226*	*1.539378*	*1.023874*
	(180,0,0)	*Max*	*0.832352*	***0.664907***	*1.374949*	*1.036417*
	(0,180,0)	*C*	*1.633687*	*0.785814*	*0.717724*	*0.892938*
	(0,180,0)	*Min*	*0.676583*	*0.856163*	*0.699361*	*1.019877*
	(0,180,0)	*Max*	*1.118812*	*0.938659*	*1.177654*	*0.933198*
	(0,0,180)	*C*	*1.678348*	*0.769760*	*1.319568*	*0.874453*
	(0,0,180)	*Min*	*1.639266*	*0.676516*	*0.755857*	*0.889838*
	(0,0,180)	*Max*	*1.299202*	*1.025759*	*1.181482*	*0.985922*
Registration	**-**	**-**	***0.759642***	***0.400716***	**-**	**-**

Table 5-5: Fragment 5 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*0.653042*	*0.688023*	*1.288499*	*0.887601*
	(0,0,0)	*Min*	*1.848610*	*1.091300*	*1.331419*	*1.131558*
	(0,0,0)	*Max*	*0.815242*	*2.477213*	*0.969793*	*1.119564*
	(180,0,0)	*C*	*0.747442*	*0.653291*	*1.154197*	*0.917926*
	(180,0,0)	*Min*	*0.973369*	*0.869785*	*1.133173*	*0.937049*
	(180,0,0)	*Max*	*1.639758*	*0.788989*	*1.306433*	*1.043943*
	(0,180,0)	*C*	*1.627108*	***0.563865***	*0.873753*	*1.012574*
	(0,180,0)	*Min*	*0.807728*	*0.823401*	*0.793645*	*0.963396*
	(0,180,0)	*Max*	*1.019317*	*0.944453*	*1.232534*	*1.051472*
	(0,0,180)	*C*	*1.929275*	*0.623213*	*1.331419*	*0.840516*
	(0,0,180)	*Min*	*1.682084*	*2.394447*	*0.781884*	*1.217733*
	(0,0,180)	*Max*	*1.193122*	*0.993110*	*0.842187*	*0.944739*
Registration	*-*	*-*	*-*	***0.212250***	*-*	*-*

Table 5-6: Fragment 6 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*1.067816*	*1.017375*	*1.137926*	***0.519864***
	(0,0,0)	*Min*	*0.769991*	*1.571829*	*0.991290*	*0.653663*
	(0,0,0)	*Max*	*0.733875*	*1.617423*	*1.168724*	*0.765838*
	(180,0,0)	*C*	*1.095761*	*0.999484*	*0.817612*	*0.604732*
	(180,0,0)	*Min*	*0.922370*	*1.199356*	*0.897323*	*0.644030*
	(180,0,0)	*Max*	*1.140861*	*1.084735*	*1.328621*	*0.717264*
	(0,180,0)	*C*	*1.105867*	*1.193531*	*1.136828*	*0.572719*
	(0,180,0)	*Min*	*0.809357*	*1.177787*	*0.776236*	*0.838214*
	(0,180,0)	*Max*	*0.886105*	***0.619259***	*1.118246*	*0.689018*
	(0,0,180)	*C*	*0.839027*	*1.353967*	*1.248824*	*0.688358*
	(0,0,180)	*Min*	*0.935819*	*1.575506*	*1.116743*	*0.694483*
	(0,0,180)	*Max*	*0.797450*	*0.741255*	*0.903432*	*0.784214*
Registration	*-*	*-*	*-*	***0.447495***	*-*	***0.650724***

Table 5-7: Fragment 7 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*0.866351*	*0.895070*	*0.858046*	*0.625949*
	(0,0,0)	*Min*	*1.216897*	*1.981192*	*0.950091*	*0.664104*
	(0,0,0)	*Max*	*1.357860*	*1.939043*	*0.852324*	*0.815576*
	(180,0,0)	*C*	*0.907307*	*0.744502*	*0.886752*	*0.794831*
	(180,0,0)	*Min*	*1.117382*	*1.058749*	*0.823335*	*0.711124*
	(180,0,0)	*Max*	*1.188870*	*1.725844*	*1.188366*	*1.028885*
	(0,180,0)	*C*	*1.168265*	*1.955062*	*0.852324*	*0.946078*
	(0,180,0)	*Min*	*0.898104*	*0.969566*	*1.031420*	*0.779191*
	(0,180,0)	*Max*	*0.798713*	***0.608945***	*0.943754*	*0.714335*
	(0,0,180)	*C*	*1.225997*	*0.740568*	*0.898191*	*0.706295*
	(0,0,180)	*Min*	*1.165880*	*0.701325*	*0.852324*	*0.648455*
	(0,0,180)	*Max*	*0.798630*	*0.707131*	*0.941355*	*1.254969*
Registration	**-**	**-**	**-**	***0.401983***	**-**	**-**

Table 5-8: Fragment 8 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*0.825544*	*0.992510*	*1.000341*	*0.582249*
	(0,0,0)	*Min*	*1.036173*	*0.698729*	*1.309680*	*0.724365*
	(0,0,0)	*Max*	*1.148025*	*0.643173*	*0.931440*	*0.656876*
	(180,0,0)	*C*	*0.795756*	*0.948780*	*1.058205*	*0.729856*
	(180,0,0)	*Min*	*0.913503*	*1.002345*	*0.904657*	*0.639946*
	(180,0,0)	*Max*	*1.326482*	*0.666594*	*1.023206*	*0.917776*
	(0,180,0)	*C*	*1.009784*	*0.575545*	*0.931440*	*0.708202*
	(0,180,0)	*Min*	*0.716187*	*1.014025*	*0.952813*	*0.726100*
	(0,180,0)	*Max*	*0.929911*	*0.839107*	*1.061480*	*0.627843*
	(0,0,180)	*C*	*1.210366*	***0.573982***	*1.309680*	*0.713132*
	(0,0,180)	*Min*	*1.159956*	*0.598969*	*0.931440*	*0.663150*
	(0,0,180)	*Max*	*0.973291*	*0.961756*	*0.829414*	***0.549071***
Registration	**-**	**-**	**-**	***0.389257***	**-**	***0.659601***

Table 5-9: Fragment 9 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*0.969511*	*1.310079*	*4.157721*	*3.582253*
	(0,0,0)	*Min*	*0.851369*	*1.837184*	*5.674205*	*4.436572*
	(0,0,0)	*Max*	*0.966758*	*1.682619*	*4.140410*	*3.687656*
	(180,0,0)	*C*	*0.854517*	*1.441528*	*4.148481*	*3.687656*
	(180,0,0)	*Min*	*1.977563*	*1.296698*	*4.189647*	*3.639694*
	(180,0,0)	*Max*	*1.361051*	*1.902031*	*5.674205*	*3.971776*
	(0,180,0)	*C*	*1.398325*	*1.701948*	*4.346008*	*5.700282*
	(0,180,0)	*Min*	*1.937506*	*1.302198*	*4.058093*	*3.710260*
	(0,180,0)	*Max*	***0.845232***	*1.317639*	*5.989043*	*3.924378*
	(0,0,180)	*C*	*1.003068*	*1.813807*	*5.958167*	*3.137132*
	(0,0,180)	*Min*	*1.118151*	*1.679994*	*4.135905*	*3.665310*
	(0,0,180)	*Max*	*1.105052*	*1.433642*	*4.148181*	***2.949372***
Registration	*-*	*-*	***0.845232***	*-*	*-*	*-*

Table 5-10: Fragment 10 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*0.96444*	*1.024249*	*1.376579*	***0.88377***
	(0,0,0)	*Min*	*1.21091*	*1.125478*	*1.360447*	*1.11123*
	(0,0,0)	*Max*	*1.38474*	*1.181153*	*1.355861*	*1.30193*
	(180,0,0)	*C*	***0.93237***	*1.062953*	*1.333417*	*1.02804*
	(180,0,0)	*Min*	*1.05104*	*1.124818*	*1.629149*	*1.09485*
	(180,0,0)	*Max*	*1.21091*	*1.119254*	*1.430147*	*1.21935*
	(0,180,0)	*C*	*1.38475*	*1.181153*	*1.344785*	*0.97362*
	(0,180,0)	*Min*	*1.19566*	*1.127881*	*1.480803*	*1.42496*
	(0,180,0)	*Max*	*1.42789*	*1.034258*	*1.307638*	*1.17133*
	(0,0,180)	*C*	*1.21091*	*1.130641*	*1.250394*	*1.17021*
	(0,0,180)	*Min*	*1.38475*	*1.181153*	*1.322909*	*1.18062*
	(0,0,180)	*Max*	*1.20968*	*1.277763*	*1.515595*	*1.33316*
Registration	*-*	*-*	***0.37425***	*-*	*-*	***0.98825***

Table 5-11: Fragment 11 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*0.753485*	*0.905731*	*0.698198*	***0.472178***
	(0,0,0)	*Min*	*0.524365*	*0.620936*	*1.037405*	*0.684808*
	(0,0,0)	*Max*	*0.525871*	*0.696011*	*1.171846*	*0.601883*
	(180,0,0)	*C*	*0.700987*	*0.753764*	*0.713947*	*0.537053*
	(180,0,0)	*Min*	*0.589620*	*1.024134*	*0.604302*	*0.532287*
	(180,0,0)	*Max*	*0.612898*	*0.620936*	*1.037405*	*0.650855*
	(0,180,0)	*C*	***0.486868***	*0.681591*	*1.171846*	*0.775401*
	(0,180,0)	*Min*	*0.663404*	*1.079034*	*0.602452*	*0.511565*
	(0,180,0)	*Max*	*0.661963*	*0.605899*	*0.880934*	*0.580683*
	(0,0,180)	*C*	*0.809234*	*0.499618*	*0.958234*	*0.661455*
	(0,0,180)	*Min*	*0.848543*	*0.729571*	*1.171846*	*0.693175*
	(0,0,180)	*Max*	*0.562951*	*0.607215*	*0.718381*	*0.597892*
Registration	**-**	**-**	***0.249710***	**-**	**-**	***0.544000***

Table 5-12: Fragment 12 matching and registration RMSE (cm)

RMSE Type	*Rotations*	*Position*	*Femur*	*Humerus*	*Pelvis*	*Skull*
Matching	*(0,0,0)*	*C*	*1.366484*	*1.430760*	*1.186663*	***0.414900***
	(0,0,0)	*Min*	*1.132600*	*1.039763*	*1.336400*	*0.959977*
	(0,0,0)	*Max*	*1.148431*	*0.983605*	*1.448458*	*0.460329*
	(180,0,0)	*C*	*0.952497*	*1.342848*	*0.988702*	*0.532524*
	(180,0,0)	*Min*	*1.114323*	*1.713051*	*0.968376*	*0.581360*
	(180,0,0)	*Max*	*1.246817*	*0.784027*	*1.254724*	*0.777154*
	(0,180,0)	*C*	*1.345439*	*0.955813*	*1.448458*	*0.479076*
	(0,180,0)	*Min*	*1.608315*	*1.675711*	*0.954680*	*0.795663*
	(0,180,0)	*Max*	***0.739734***	*0.947338*	*1.495229*	*0.507942*
	(0,0,180)	*C*	*1.131481*	*1.527743*	*1.252292*	*0.848205*
	(0,0,180)	*Min*	*1.147760*	*1.050073*	*1.132909*	*0.671835*
	(0,0,180)	*Max*	*0.836843*	*0.755775*	*1.004744*	*0.571146*
Registration	**-**	**-**	***0.278584***	**-**	**-**	***0.449865***

Table 5-13 summarizes matching minimum RMSE. In Table 5-13, green color highlights a correct matching, while red color highlights an incorrect matching. Table 5-14 list the test sample, total number of correct and incorrect matchings.

Table 5-13: Matching minimum RMSE (cm)

Fragment	*Femur*	*Humerus*	*Pelvis*	*Skull*	*Bone*
1	*0.73328*	***0.514663***	*0.699231*	***0.385311***	*Humerus*
2	*0.667225*	***0.557928***	*0.783834*	***0.455202***	*Humerus*
3	*0.850877*	***0.640472***	*0.898699*	*0.698427*	*Humerus*
4	***0.634426***	***0.664907***	*0.699361*	*0.753343*	*Humerus*
5	*0.653042*	***0.563865***	*0.781884*	*0.840516*	*Humerus*
6	*0.733875*	*0.619259*	*0.776236*	***0.519864***	*Femur*
7	*0.79863*	***0.608945***	*0.823335*	*0.625949*	*Femur*
8	*0.716187*	***0.573982***	*0.829414*	***0.549071***	*Humerus*
9	***0.845232***	*1.296698*	*4.058093*	*2.949372*	*Femur*
10	***0.932372***	*1.024249*	*1.250394*	***0.883769***	*Femur*
11	***0.486868***	*0.499618*	*0.602452*	***0.472178***	*Femur*
12	***0.739734***	*0.755775*	*0.954680*	***0.414900***	*Femur*

Table 5-14: Overall evaluation

Test Samples	*Correct Matchings*	*Incorrect Matchings*
12	*10*	*2*

5.2 Discussion

From Table 5-13, and by looking to the fragment models, Figure 5-2, Figure 5-5, Figure 5-6 -Figure 5-15, we can see that the proposed method did work well with fragments having original bone features, as it identifies the correct template bone for fragments 1, 2, 3, 4, 5, 8 (fragments of humerus), 9, 10, 11, 12 (fragments of femur) although the fragments are from one side of the body and template from the other side. The method fails to identify the correct template for the fragments 6, 7 and detect them as being fragments of skull and humerus template model while they are actually fragments of femur. This is due to the assumption mentioned above, the fragment bones should have some of the original bone features in order for the proposed method to generate correct results.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

We've introduced an automated technique for matching of fragmentary human skeletal remains for improving forensic anthropology practice and policy. The proposed technique involved creation of surfaces models for the fragmentary elements using computerized tomographic scans followed by segmentation. A feature extraction technique is proposed where the surface roughness map of each model is measured using local shape analysis measures. Adaptive thresholding, based on Gaussian mixture model, is then used to extract model features. A multi-stage technique is then used to identify, match and register bone fragments to their corresponding template bone model. First, extracted features are used for matching with different template bone models using iterative closest point algorithm with different positions and orientations. The best match score, in terms of minimum root-mean-square error, is used along with the position and orientation and the resulting transformation to register the fragment bone model with the corresponding template bone model using iterative closest point algorithm. The proposed method showed good results assuming that fragment bone have features from the original bone.

6.2 Future Work

The presented work can be significantly improved in different aspects, including and not limited to:

- Extracting feature based on integral invariants instead of surface differential properties as it is more robust, can be computed at multiple scales by default. Although, integral invariants need to be optimized somehow as it is very expensive in terms of computations.
- Matching based on feature clusters instead of feature points as it is more robust.
- Registration based on AICP (Anisotropic ICP) instead of ICP as it incorporates point's localization errors in calculating closest point as well as in calculating the rigid transformation, which is closer to the real case.
- Estimating MNE, MNI.

REFERENCES

[1] **Adams, Bradley J. and Byrd, John E.** *Recovery, Analysis, and Identification of Commingled Human Remains.* s.l. : Springer, 2008.

[2] **Mahfouz, Mohamed R., Shirley, Natalie R. and Herrmann, Nicholas P.** *Computerized Reconstruction of Fragmentary Skeletal Remains.* Mechanical, Aerospace and Biomedical Engineering, University of Tennessee. Knoxville : s.n., 2011.

[3] **Haddow, Scott D.** *Scotthaddow.* [Online] Jul 2012. scotthaddow.wordpress.com/tag/osteology-2.

[4] *Reassembling Fractured Objects by Geometric Matching.* **Huang, Qi-Xing, et al.** 3, NY : s.n., Jul 2006, ACM Transactions on Graphics, Vol. 25, pp. 569-578.

[5] *Fractured Surfaces Matching for Reassembling Broken Solids.* **Li, Qunhui, Zhou, Mingquan and Geng, Guohua.** 2012. Fifth International Symposium on Computational Intelligence and Design . Vol. 2, pp. 511-514.

[6] *Robust global registration.* **Gelfand, Natasha, et al.** Aire-la-Ville, Switzerland : s.n., 2005. The Third Eurographics Symposium on Geometry Processing. pp. 197–206.

[7] *A Computational/Experimental Platform for Investigating ThreeDimensional Puzzle Solving of Comminuted Articular Fractures.* **Thomas, Thaddeus P., et al.** 3, s.l. : Taylor and Francis, 2011, Computer Methods in Biomechanics and Biomedical Engineering, Vol. 14, pp. 263-270.

[8] *Virtual 3D Bone Fracture Reconstruction via Inter-fragmentary Surface.* **Zhou, Beibei, et al.** Kyoto : s.n., 2009. IEEE 12th International Conference on Computer Vision Workshops . pp. 1809-1816.

[9] **Herrmann, Nicholas P. and Devlin, Joanne Bennett.** Assessment of Commingled Human Remains Using a GIS-Based Approach. [book auth.] Bradley J. Adams and John E. Byrd. *Recovery, Analysis, and Identification of Commingled Human Remains.* Totowa, NJ : Humana Press, 2008, 13.

[10] *Restoration of Full Anatomy from Partial Bones.* **Abdel Fatah, Emam E. and Mahfouz, Mohamed R.** Salt Lake City, Utah : s.n., 2013. 11th International Symposium of Computer Methods in Biomechanics and Biomedical Engineering.

[11] **Abdel Fatah, Emam E.** *Three Dimensional Nonlinear Statistical Modeling Framework for Morphological Analysis.* Mechanical, Aerospace and Biomedical Engineering, The University of Tennessee. Knoxville, TN : s.n., 2012. PhD Dissertation.

[12] *Improving Sex Estimation from Crania Using a Novel Three-dimensional Quantitative Method (in print).* **Abdel Fatah, Emam E., et al.** Journal of Forensic Science.

[13] *A Three-dimensional Analysis of Bilateral Directional Asymmetry in the Human Clavicle.* **Abdel Fatah, Emam E., et al.** 4, 2012, American Journal of Physical Anthropology, Vol. 149, pp. 547-559.

[14] *Automatic Methods for Characterizing of Sexual Dimorphism of Adult Femora: Distal Femur.* **Mahfouz, Mohamed R., et al.** 6, 2007, Computer Methods in Biomechanics and Biomedical Engineering, Vol. 10, pp. 447-456.

[15] **Wikipedia.** Curvature. *Wikipedia.* [Online] Nov 2013. en.wikipedia.org/wiki/Curvature.

[16] **Wikipedia.** Parametric Surface. *Wikipedia.* [Online] Sept 2013. en.wikipedia.org/wiki/Parametric_surface.

[17] **Pressley, Andrew.** *Elementary Differential Geometry.* London : Springer-Verlag, 2001.

[18] **Gaba, Eric.** Surface Curvature Planes. *Wikipedia.* [Online] Wikimedia Inc., Jun 2006. en.wikipedia.org/wiki/File:Minimal_surface_curvature_planes-en.svg.

[19] *Surface Shape and Curvature Scales.* **Koenderink, Jan J. and Doorn, Andrea J. van.** 8, Newton, MA : s.n., Oct 1992, Image and Vision Computing, Vol. 10, pp. 557-565.

[20] *Computer-Aided Diagnosis for CT Colonography.* **Hiroyuki, Yoshida and H., Dachman Abraham.** 5, Oct 2004, Seminars in Ultrasound, CT and MRI, Vol. 25, pp. 419 - 431.

[21] *A Roughness Measure for 3D Mesh Visual Masking.* **Lavoue, Guillaume.** Lyon : s.n., 2007. 4th Symposium on Applied Perception in Graphics and Visualization. pp. 57-60.

[22] *Roughness as a Shape Measure.* **Kushunapally, Rakesh, Razdan, Anshuman and Bridges, Nathan.** 1-4, 2007, Computer-Aided Design and Applications, Vol. 4, pp. 295-310.

[23] **Bronstein, Alexander M., Bronstein, Michael M. and Ovsjanikov, Maks.** Feature-based Methods in 3D Shape Analysis. [book auth.] Nick Pears, Yonghuai Liu and Peter Bunting. *3D Imaging, Analysis and Applications.* s.l. : Springer, 2012, 5, pp. 185-220.

[24] *Robust Feature Detection and Local Classification for Surfaces based on Moment Analysis.* **Clarenz, Ulrich, Rumpf, Martin and Telea, Alexandru.** 5, Oct 2004, IEEE Transactions on Visualization and Computer Graphics, Vol. 10, pp. 516-524.

[25] *Estimating Curvature on Triangular Meshes.* **Gatzke, Timothy D. and Grimm, Cindy M.** 11, s.l. : World Scientific Publishing Company, Apr 2005, International Journal of Shape Modeling, Vol. 12.

[26] *Efficient Feature Extraction For 2D/3D Objects in Mesh Representation.* **Zhang, Cha and Chen, Tsuhan.** Pittsburgh, PA : s.n., 2001. International Conference on Image Processing. Vol. 3, pp. 935-938.

[27] *Surface Feature Detection and Description with Applications to Mesh Matching.* **Zaharescu, Andrei, et al.** 2009. International Conference on Computer Vision and Pattern Recognition. pp. 373-380.

[28] **Lai, Zhaoqiang and Hua, Jing.** 3D Surface Matching and Registration through Shape Images. [book auth.] Dimitris Metaxas, et al. *Medical Image Computing and Computer Assisted Intervention - MICCAI 2008.* New York : Springer Berlin Heidelberg, 2008, 6, pp. 44-51.

[29] *Multi-scale Feature Extraction for 3D Surface Registration using Local Shape Variation.* **Ho, Huy Tho and Gibbins, Danny.** New Zealand : s.n., 2008. 23rd International Conference in Image and Vision Computing. pp. 1-6.

[30] *Multi-scale Feature Extraction on Point-Sampled Surfaces.* **Pauly, Mark, Keiser, Richard and Gross, Markus.** 3, Nov 2003, Computer Graphics Forum, Vol. 22, pp. 281-289.

[31] *Feature Extraction from Point Clouds.* **Gumhold, Stefan, Wang, Xinlong and MacLeod, Rob.** Salt Lake City, Utah : s.n., 2001. 10th International Meshing Roundtable. pp. 293-305.

[32] *Feature Detection for Surface Meshes.* **Jiao, Xiangmin and Heath, Michael T.** 2002. 8th International Conference on Numerical Grid Generation in Computational Field Simulations.

[33] *Geometric Snakes for Triangular Meshes.* **Lee, Yunjin and Lee, Seungyong.** s.l. : Eurographics, 2002. Computer Graphics Forum. Vol. 21, pp. 229-238.

[34] *Integral Invariants for Robust Geometry Processing.* **Pottmann, Helmut, et al.** 1, s.l. : Elsevier, Jan 2009, Computer Aided Geometric Design, Vol. 26, pp. 37-60.

[35] **Manay, Siddharth, et al.** Integral Invariant Signatures. *Computer Vision - ECCV 2004.* s.l. : Springer Berlin Heidelberg, 2004, pp. 87-99.

[36] *Integral Invariants for Shape Matching.* **Manay, Siddharth, et al.** 10, Oct 2006, IEEE Transaction on Pattern Analysis and Machine Intellegence, Vol. 28, pp. 1602-1618.

[37] **Mara, Hubert.** *Multi-Scale Integral Invariants for Robust Character Extraction from Irregular Polygon Mesh Data.* Natural Sciences and Mathematics, Ruprecht-Karls-Universitat Heidelberg. 2012. PhD Dissertation.

[38] **Levallois, Jeremy.** Integral Invariant Curvature Estimator 2D/3D. *DGtal - Digital Geometry Tools and Algorithms Library.* [Online] Feb 2013. libdgtal.org/doc/nightly/moduleIntegralInvariant.html.

[39] *Shape Matching and Object Recognition Using Shape Context.* **Belongie, Serge, Malik, Jitendra and Puzicha, Jan.** 4, Apr 2002, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 24, pp. 509-522.

[40] *A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion.* **Sun, Jian, Ovsjanikov, Maks and Guibas, Leonidas.** 5, s.l. : The Eurographics Association and Blackwell Publishing Ltd., July 2009, Computer Graphics Forum, Vol. 28, pp. 1383-1392.

[41] *Scale-Invariant Heat Kernel Signatures for Non-Rigid Shape Recognition.* **Bronstein, Michael M. and Kokkinos, Iasonas.** San Francisco, CA : s.n., 2010. IEEE conference on Computer Vision and Pattern Recognition. pp. 1704-1711.

[42] **Johnson, Andrew E.** *Spin-Images: A Representation for 3-D Surface Matching.* The Robotics Institute, Carnegie Mellon University. 1997. PhD. Dissertation.

[43] *Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes.* **Johnson, Andrew E. and Hebert, Martial.** 5, May 1999, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, pp. 433-449.

[44] *Posture Invariant Surface Description and Feature Extraction.* **Wuhrer, Stefanie, Azouz, Zouhour Ben and Shu, Chang.** CA : s.n., 2010. IEEE Conference on Computer Vision and Pattern Recognition. pp. 374-381.

[45] *Surface Matching for Object Recognition in Complex 3-D Scenes.* **Johnson, Andrew Edie and Hebert, Martial.** 1998. Image and Vision Computing.

[46] **Papadimitriou, Georgios.** Spin Images. *The University of Edinburgh, School of Informatics.* [Online] Sep 2010. homepages.inf.ed.ac.uk/rbf/cvonline/local_copies/av0910/spinimages.pdf.

[47] *Salient Geometric Features for Partial Shape Matching and Similarity.* **Gal, Ran and Cohen-Or, Daniel.** 1, New York, NY : s.n., Jan 2006, ACM Transactions on Graphics, Vol. 25, pp. 130-150.

[48] *Surface Partial Matching and Application to Archaeology.* **Itskovich, Arik and Tal, Ayellet.** 2, Technion, Israel : Elsevier, Jan 2011, Computers and Graphics, Vol. 35, pp. 334-341.

[49] *Conformal Geometry and Its Applications on 3D Shape Matching, Recognition, and Stitching.* **Wang, Sen, et al.** 7, July 2007, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, pp. 1209-1220.

[50] *Harmonic Maps and Their Applications in Surface Matching.* **Zhang, Dongmei and Hebert, Martial.** 1999. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 2.

[51] **Gu, Xianfeng and C.Vemuri, Baba.** Matching 3D Shapes Using 2D Conformal Representations. [book auth.] Christian Barillot, David R. Haynor and David R. Haynor. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004.* s.l. : Springer Berlin Heidelberg, 2004, pp. 771-780.

[52] *Mutual Information-Based 3D Surface Matching with Applications to Face Recognition and Brain Mapping.* **Wang, Yalin, Chiang, Ming-Chang and Thompson, Paul M.** Washington, DC : s.n., 2005. IEEE International Conference on Computer Vision. Vol. 1, pp. 527-534.

[53] *Dense Non-rigid Surface Registration Using High-Order Graph Matching.* **Zeng, Yun, Wang, Chaohui and Wang, Yang.** San Francisco, CA : s.n., 2010. IEEE Conference on Computer Vision and Pattern Recognition. pp. 382 - 389.

[54] *An Image Processing Approach to Surface Matching.* **Litke, Nathan, et al.** Switzerland : s.n., 2005. Eurographics Symposium on Geometry Processing.

[55] **Wikipedia.** Similarity Measure. *Wikipedia.* [Online] Wikimedia Foundation, Inc., Oct 2013. en.wikipedia.org/wiki/Similarity_measure.

[56] *Comprehensive Survey on Distance Similarity Measures between Probability Density Functions.* **Cha, Sung-Hyuk.** 4, 2007, International Journal of Mathematical Models and Methods in Applied Sciences, Vol. 1, pp. 300-307.

[57] *A Survey on Shape Correspondence.* **Kaick, Oliver van, et al.** 6, s.l. : Blackwell Publishing, Sept 2011, Computer Graphics Forum, Vol. 30, pp. 1681-1707.

[58] **Guo, Kehua and Duan, Guihua.** 3D Partial Surface Matching Using Differential Geometry and Statistical Approaches. [book auth.] Jian Yang, Fang Fang and Changyin Sun. *Intelligent Science and Intelligent Data Engineering.* s.l. : Springer Berlin Heidelberg, 2013, pp. 172-180.

[59] *Differential and Statistical Approach to Partial Model Matching.* **Guo, Kehua, Liu, Yongling and Duan, Guihua.** Changsha : Hindawi Publishing Corporation, 2013, Mathematical Problems in Engineering, Vol. 2013.

[60] *A Novel Matching Algorithm to 3D Incomplete Object.* **GUO, Kehua.** 1, 2011, Journal of Computational Information Systems, Vol. 7, pp. 73-79.

[61] *A Method for Registration of 3D Shapes.* **Besl, Paul J. and McKay, Neil D.** 2, Feb 1992, IEEE Transaction on Pattern Analysis and Machine Interlligence, Vol. 14, pp. 239-256.

[62] *Convergent Iterative Closest-Point Algorithm to Accommodate Anisotropic and Inhomogeneous Localization Error.* **Maier-Hein, Lena, et al.** 8, Aug 2012, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, pp. 1520-1532.

[63] *Efficient Variants of the ICP Algorithm.* **Rusinkiewicz, Szymon and Levoy, Marc.** 2001. Third International Conference on 3D Digital Imaging and Modeling. pp. 145-152.

[64] *Zippered Polygon Meshes from Range Images.* **Turk, Greg and Levoy, Marc.** New York, NY : s.n., 1994. 21st Annual Conference on Computer Graphics and Interactive Techniques. pp. 311-318.

[65] *Registration and integration of multiple range images for 3D model construction.* **Masuda, Takeshi, Sakaue, Katsuhiko and Yokoya, Naokazn.** Washington, DC : s.n., 1996. International Conference on Pattern Recognition. pp. 879-883.

[66] *Object Modeling By Registration of Multiple Range Images.* **Chen, Yang and Medioni, Gerard.** CA : s.n., 1991. IEEE International Conference on Robotics and Automation. Vol. 3, pp. 2724-2729.

[67] *Three-Dimensional Registration Using Range and Intensity Information.* **Godin, Guy, Rioux, Marc and Baribeau, Réjean.** 1994. SPIE. Vol. 2350, pp. 279-290.

[68] *Rigid, Affine and Locally Affine Registration of Free-Form Surfaces.* **Feldmar, Jacques and Ayache, Nicholas.** 2, Hingham, MA : s.n., May 1996, International Journal of Computer Vision, Vol. 18, pp. 99-119.

[69] *ICP Registration using Invariant Features.* **Sharp, Gregory C., Lee, Sang W. and Wehe, David K.** 1, Jan 2002, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, pp. 90 - 102.

[70] *Multiview Registration for Large Data Sets.* **Pulli, Kari.** CA : s.n., 1999. Second International Conference on 3D Digital Imaging and Modeling. pp. 160-168.

[71] *Registration and Integration of Multiple Object Views for 3D Model Construction.* **Dorai, Chitra, et al.** 1, Jan 1998, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, pp. 83-89.

[72] *A Generalization of the Metric-Based Iterative Closest Point Technique for 3D Scan Matching.* **Armesto, Leopoldo, Minguez, Javier and Montesano, Luis.** Anchorage, AK : s.n., 2010. IEEE International Conference on Robotics and Automation. pp. 1367 - 1372.

[73] *The Trimmed Iterative Closest Point Algorithm.* **Chetverikov, Dmitry, et al.** Washington, DC : s.n., 2002. International Conference on Pattern Recognition. Vol. 3, pp. 545-548.

[74] *A comparison of Gaussian and mean curvatures estimation methods on triangular meshes.* **Surazhsky, Tatiana, et al.** 2003. IEEE International Conference on Robotics and Automation. Vol. 1, pp. 1021-1026.

[75] *A New Image Thresholding Method Based on Gaussian Mixture Model.* **Huang, Zhi-Kai and Chau, Kwok-Wing.** 2, s.l. : Elsevier, Nov. 2008, Applied Mathematics and Computation, Vol. 205, pp. 889-907.

[76] **Picard, Franck.** *An Introduction to Mixture Models.* Laboratoire Statistique et Genome, University d'Evry. France : s.n., 2007. pp. 1-16.

VITA

Ali S. Mustafa was born in Cairo, Egypt, in 1985. He received the bachelor degree in systems and biomedical engineering from Cairo University in 2007. He worked as a software engineer at Diagnosoft where he was involved in developing medical software tool to help physicians better understands and assesses the health of cardiac muscle in order to improve diagnosis, prognosis, and treatment. In 2009, he was chosen to be a teaching and research assistant at Cairo University, and in parallel he worked as Research and Development Engineer at IBE Tech where he was involved in building a library for real-time speckle reduction in ultrasound images. He worked as a senior software engineer at Diagnosoft from 2010 to 2012 before joining the research team at the Center for Musculoskeletal Research at the University of Tennessee. He received the Master of Science degree in biomedical engineering from the University of Tennessee in 2013. His current research interests include medical imaging, image processing and computer vision.