Bending Invariant Correspondence Matching on 3D Models with Feature Descriptor

LI, Sai Man

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Automation and Computer-Aided Engineering

The Chinese University of Hong Kong
September 2010

Thesis/Assessment Committee

Professor Charlie C.L. Wang (Supervisor)
Professor K.C. Hui (Supervisor)
Professor Ronald C.K. Chung (Chair)
Professor Y. Yam (Committee Member)
Dr. K.M. Yui (External Examiner)

Abstract

In this thesis, an automatic approach to match correspondences on 3D human bodies is proposed. The proposed algorithm maps correspondences on 3D human bodies in varied postures so that the semantic feature points can be automatically extracted. The semantic feature points are very important to establish the volumetric parameterization around human bodies for the human-centered customization of soft-products. For a given template human model with a set of predefined semantic feature points, the model is down sampled into a set of sample points (including the semantic feature points). Then the corresponding points of these samples on the human model are identified by minimizing the distortion with the help of series of transformation regardless of their differences in postures, scales and/or positions. The basic idea the proposed algorithm is to transform the template human body to the shape of the input model iteratively. To generate a bending invariant mapping, the initial corresponding/transformation is computed in a multi-dimensional scaling (MDS) embedding domain of 3D human models, where the Euclidean distance between two samples on 3D model in the MDS domain is corresponding to the geodesic distance between them in R3. As the posture change (i.e., the body bending) of a human model can be considered as being approximately isometric in the intrinsic 3D shape, the initial correspondences established in the MDS domain can greatly enhance the robustness of the proposed approach to the body bending. Once the correspondences between the surface samples on the template model and the input model are determined after iterative transformations, we have essentially found the corresponding semantic features on the input model. Lastly, the locations of semantic features on the input model are refined by a curvature map based local matching step.

此論文提請一種"含特徵描述的三維模型彎曲不變配對演算法"。此技術現廣泛應用於不規則物件的自動法設計中，例如成衣工業等。

此提請的演算法是一種融合整體循環隊列調整及局部特徵配對之技術。而局部特徵配對乃基於兩件配對模型 – 範本及輸入物件之等量簽署必須為近似為大前題。整個過程可分為三大步驟：

第一個步驟稱為**姿勢調整**。此步驟先透過兩件配對物件之等量簽署作初步配對，從而將範本變形至近似輸入物件之形狀。而物件之等量簽署可透過一種名為多維倍變之演算法獲得。

第二個步驟稱為**表面調整**。此步驟主要透過重複的優化兩件配對模型的三角表面匹配度及流暢度。優化匹配度的程序利用物件兩方的點對點配對不斷將對方的表面拉近，最後以每點及附近之點互相平衡以優化表面之流暢度。

第三個步驟稱為**特徵調整**。此步驟將第二步所得之調整結果透過一種" 特徵描述"的比對從而加以修改。而此" 特徵描述"更可不受物件之間的彎曲、微量變形或方向等等所影響，因而得出一個別於一般物件配對演算法的結果。

有關演算法的詳細介紹，分析及測試結果等可於本論文內一一找到。

Table of Content

List of Figures

Acknowledgement

This research is supported by Hong Kong RGC/CERG grant CUHK/412405 and Direct Grant CUHK/205034. Works implemented by the author (LI SAI MAN), under the supervision of Prof. Kin-Chuen Hui and Prof. Charlie C.L. Wang, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong in 2006 – 2009.

Chapter 1 Introduction

Different from the design automation functions provided in current commercial Computer-Aided Design (CAD) systems that are developed for products with regular shapes **[JH02]** and are usually driven by dimensional parameters, the design automation of human centered soft-products relies on establishing the volume parameterization of spaces around the human bodies, where a challenging step is how to extract the semantic feature points on the 3D models of human body. The semantic feature points serve as the anchor points to constrain the volume parameterization **[MN78].** In many previous researches **[LDRS05][PDJ93][WWWY03a]**, the semantic feature points (at least part of them) are specified by users or semi-automatically selected by rule-based systems. An automatic method is proposed in the thesis.

1.1 Problem definition

Given a template 3D human model M_u represented as a polygonal mesh surface $U \in \Re^3$ with a set of predefined semantic feature points, a^u, the proposed algorithm is going to find the corresponding feature points, a^v, on the surface V of an input human model M_v. Without loss of generality, V is also represented by a polygonal mesh in $\Re^3$ and both U and V have surface normal facing outwards.

The automatic extraction is challenging for two reasons. First, the semantic features are not always located at the shapes extremes, therefore the local shape matching based methods cannot robustly give satisfactory answers. Second, the robustness of local shape matching is more problematic when the postures of human bodies are varied (i.e., the 3D bodies are bended).

In this thesis, a global deformation based fitting method is proposed to automatically find the correspondences between M_u and M_v thus the locations of semantic feature points on V. Specifically, a mapping f will be found to minimize the distortion function E as

$$E(f) = \int \|V - f(U)\|^2 ds \qquad (1.1)$$

With $\|\ldots\|$ the L^2–norm in $\Re^3$. In order words, by the optimal mapping function

$$f = \arg\min E(f) \qquad (1.2)$$

The semantic feature points can be determined by

$$A^v = \{a^v \mid a^{v=} f(a^u, \forall a^u \in A^u\} \qquad (1.3)$$

To have a refined matching, some important points in a^v should have similar local shape distributions to their corresponding points in A^u. This will serve as constraints for the minimization problem defined in Eq.(1.2).

1.2. Proposed algorithm

The proposed bending invariant matching algorithm integrates both the Global Surface Alignment and the Feature Based Matching techniques, which were found as the two major techniques in shape matching studies **[EK03][CS05][CJGQ05][WWJGS06]**. Moreover, the integration of these two techniques also inherits the benefits of them in one algorithm. The matching algorithm has three steps: 1) **Posture Alignment**, 2) **Surface Fitting** and 3) **Feature Matching Refinement**.

Firstly, the posture alignment step transforms the template model towards the input model non-rigidly according to the control points mapping defined by their similar isometric signatures – the multi-dimensional scaling (MDS) embedding. The MDS embedding of a given model is defined in a k-dimensional domain according to the relative distribution of surface points on the model. The robustness of finding good initial correspondences according to the MDS embedding is guaranteed by the facts that 1) the shapes of a human body in different postures are approximately isometric to each other and 2) the isometric shapes have the same MDS embedding.

Secondly, the surface fitting step refines the surfaces of the transformed template by optimizing the fitness and the smoothness iteratively. Two main process, surface fitness optimization and surface smoothing, will be repeatedly applied until the changes on surface converge to a limited amount. The surface fitting procedure employs a bi-direction mapping concept and the orientation-aware movement, which greatly improve the fitting quality of template model.

Finally, the feature matching refinement step further refines the correspondence by adopting the feature descriptor constraints on particular surface regions. At this stage, the descriptor is encoded on a surface point with curvature distribution information on the surface around it. The concept is similar to the Curvature Maps presented by Gatzke et. Al in **[GGGZ05]** but in a constrained manner. Hence, the pre-defined semantic points on the template model can be mapped to the input model according to the feature-aligned models.

1.3. Main features

The main features of our method are outlined as follows.

- A MDS-based point matching algorithm is investigated to align the initial correspondences between the template human model and the given 3D human model. A sign-flip correction technique is developed to enhance the robustness of MDS embedding. The details of sign-flip problem can be found in chapter 3.2. Without this sign correction technique, the MDS-based method cannot be applied to find correct matching on those nearly symmetric models like human bodies **[JZ06][EK03]**.
- Starting from the initial correspondences, a global alignment technique is exploited to iteratively find a mapping function (via the point correspondences) that optimizes surface proximities and is sensitive to semantic features (see Chapter 3.2).

These main features of our method lead to a robust semantic feature extraction technique for 3D human bodies in various postures. In fact, the method proposed in this thesis can also be applied to other classes of models which are approximately isometric. Although the whole human bodies are employed as examples in this paper, there is no different to apply it to parts of human bodies (e.g. feet, hands, faces, etc) **[WCY03][WWY03b][WHT07]**.

The rest of the thesis is organized as follows. After review the related work in Chapter 2, the fundamental techniques and detail methodology of the proposed algorithm are presented in Chapter 3.1 and 3.2 respectively. The experimental results are shown and studied in section 4. Lastly, the thesis ends with the conclusion chapter.

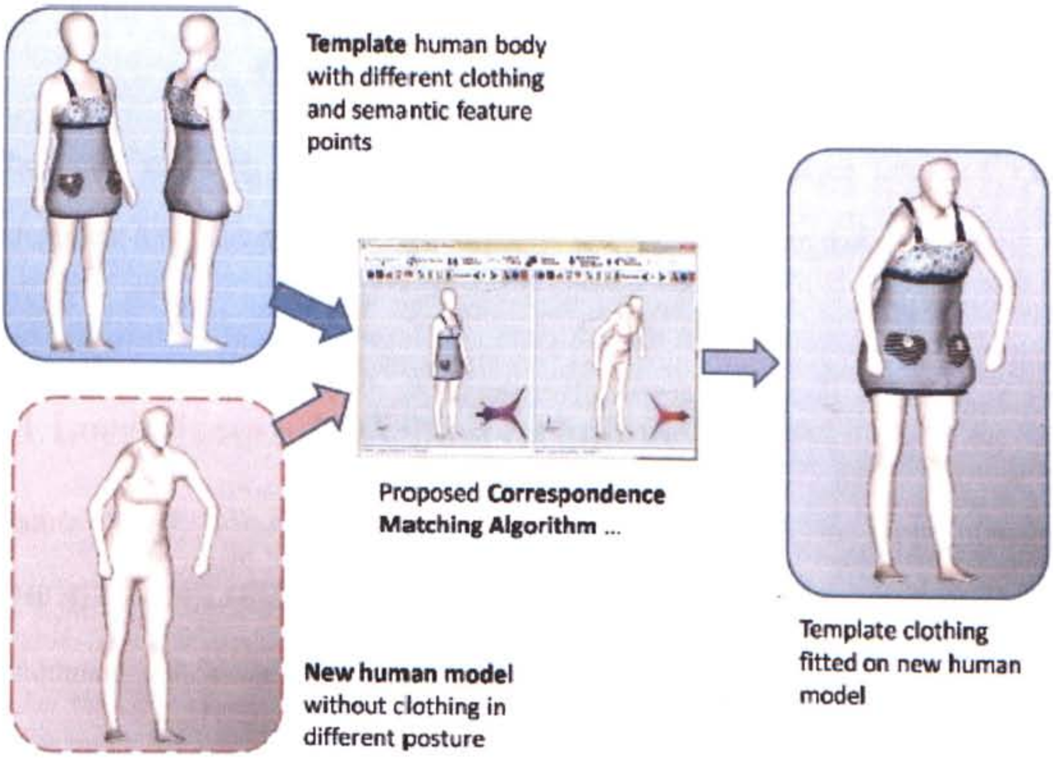

Figure 1-1: Illustration for garment design automation, where the template model has predefined features and the clothes are designed around the template body. By the correspondences between human bodies, the garments are refit to the new human model.

Chapter 2 Literature Review

The point matching algorithms in literature can be classified into two major categories: local feature matching and global iterative alignment techniques.

2.1 Local Feature Matching techniques

Feature based matching has been a common approach in shape matching **[JH02][PDJ93][LDRS05][JD03]**. It can be found in 2D applications such as photo panorama, text recognition and animation morphing. The feature are always represented by grouping regional information in point, known as descriptor. Two well known descriptors for image are shape contexts and spin images **[LD04]**, both utilizing a histogram obtained by binning the space around a point according to the Euclidean metric and collecting point counts. These methods have subsequently been generalized in a straight-forward manner to handle 3D point sets. However, neither space contexts nor spin images are invariant to shape bending. Some extra works have to be done to deal with non-rigid object matching problem in 3D scenario.

The Curvature Map introduced by Gatzke et. Al. in **[GGGZ05]** is a kind of feature descriptor which gathers the local differential geometry information at a point. A curvature map is first defined around a point v, and then accumulates curvature information from a region around v and takes one of two forms: a one-dimensional (1-D) map, which only consider the distance from v, or a two-dimensional (2-D) map that uses both the distance and the orientation information.

In most shape matching applications, geometric feature is a very important portion to be preserved during matching processes (e.g., reverse engineering of mechanical parts **[JH02]**). One significant drawbacks to incorporate curvature map in 3D object matching is the disability to handle bended objects. Although geodesic binning is invariant to bending, the histograms computed are based on curvature distributions, which are not invariant to bending.

Another bending invariant shape matching approach that incorporated a domain transformation technique has been proposed by researchers recently, that is the Multi-Dimensional Scaling (MDS) transformation.

Multi-Dimensional Scaling Transform – Apart from using the most common proximity measure – Euclidean Distance in shape matching (that implies invariance to rotation and transformation), geodesic distances can be used as a measurement between mesh vertices to handle bending invariant matching. This approach is named Multi-Dimensional Scaling Transform. Two famous approach proposed by Jain in **[JZ06]** and Kimmel in **[EK03]** illustrated the power of MDS transform in shape matching studies. Their working principal can be summarized as following.

Consider that given two triangular meshes, the objective is to find the correspondence between the mesh vertices regardless of their bending invariance. The spatial coordinates of the mesh have to be converted to an isometric domain first, such that each vertex of the mesh is represented by intrinsic structural information. Kimmel illustrated a good example to demonstrate how geodesic distance can be served as the proximity of this structure **[EK03]**.

To compute the geodesic distance between vertex pairs, the Fast Marching on Triangulated Domain introduced by Kimmel in **[KS98]** can be applied. Various similar approaches can also be found in **[SSKGH05][NK02]**. Once the geodesic distance between sampled vertices are estimated, an affinity matrix

$$A = [x_r - x_s]^T [x_r - x_s] \tag{2.1}$$

which collected the geodesic proximities between vertices *r* and *s*, will be served as the input dissimilarity matrix of the Multi-Dimensional Scaling (MDS). Further in Jain's approach **[JZ06]**, they applied a Gaussian kernel on matrix *A* such that *A* becomes

$$A_{rs}^{M} = e^{\frac{-d_{rs}^2}{2\sigma_M^2}} \tag{2.2}$$

Hence the (dis)similarity on local region can be regularized.

In Kimmel's approach **[EK03]**, they have evaluated three MDS computation methods: The Classical MDS, Least Squares MDS and the Fast MDS. According to their evaluation result, the quality of the LS MDS and Classical MDS are almost the same. Therefore, the Classical MDS and Fast MDS will be evaluated in this thesis in terms of their computation performance and quality. For detail implementation of the MDS, it will be further discussed in the fundamental technique section.

The MDS transform based matching technique is an efficient method for bending invariant shape matching problems. The main drawback of this approach is the high computation time required for geodesic distance estimation. Some approaches like **[EK03][BBK07]** proposed to combine sampling and surface recovering techniques to enhance to performance of the MDS transform. However, there is trade off between quality and performance, a good choice of sample rate must be determined according to experiments.

2.2. Global Iterative alignment techniques

In global iterative alignment matching approaches, there are two unknown variables to be determined: the correspondence and the transformation. While it is impossible to solve either variable without information regarding the other, it is possible to optimize these unknowns by determining them iteratively. Once the correspondence is given, the transformation can be guessed with reasonable knowledge. On the other hand, the correspondences can be searched if the transformation is known. Hence, it leads to a solution of the correspondence problem by alternating estimations of correspondence and transformation (e.g., **[HPM03][JSH01][RCB97]**).

The ICP algorithm is the simplest one among these methods. It utilizes the nearest-neighbor relationship to assign a binary correspondence at each step. This estimation of the correspondence is then used to refine the transformation, and vice versa. It is a very simple and fast algorithm which is guaranteed to converge to a local minimum. Chui et. Al. enhanced this algorithm in **[CR03]** by making two significant improvements: the Soft-assign idea and the Robust Point Matching – Thin Plate Spline (RPM-TPS) matching algorithm.

The basic idea of the soft-assign **[RCB97]** is to relax the binary correspondence variables to be a continuous valued matrix M in the interval [0, 1], while enforcing the row and column constraints. The continuous nature of the correspondence matrix M basically allows fuzzy, partial mapping between the point sets. Hence the correspondences are able to improve gradually and continuously during the optimization without jumping around in the space of binary permutation matrices. The row and column constraints are enforced via iterative row and column normalization of the corresponding matrix M. Chui et. Al. also proposed the RPM-TPS as the parameterization of non-rigid spatial mapping transformation. Their work is based on the RPM algorithm that involves a dual update process embedded within an annealing scheme. Obviously, the iterative alignment approach does not require any complicated algorithm or computation. Nevertheless, the initial guess mapping of correspondence must be very well enough in order to solve the bending invariant matching problem.

In the contrary, a hybrid approach of this technique on the MDS signature would be a much reliable and advanced approach for deformable shape matching. Recently, Lipman and Funkhouse **[LF09]** propose a surface correspondence matching method by repeatedly computing Mobius transformations, which needs the input models to be two-manifold – this is more restrict than the proposed approach in this thesis that is based on spatial transformation.

2.3 Other Approaches

Apart from the feature based and iterative alignment approaches, other previous work of matching approaches for shape matching such as skeletal based matching **[CS05]** and image based matching have also been studied. Some famous approaches such as the shock graph **[Kal99]**, reeb graph **[HSKK01]**, conformal geometry **[WWJGS06]**, canonical homology basis **[CJGQ05]**, etc., achieved the shape matching goal in certain field of applications. Nevertheless, the structural information of these approaches does not provided detail matching ability of mesh surfaces.

Chapter 3 Correspondence Matching

This chapter describes technical details of the proposed matching algorithm and all fundamental techniques required throughout the matching processes.

The fundamental technique section introduces the related techniques contributed by previous researchers. They include the Farthest Point Sampling (FPS) **[MD03]**, Radial Basis Function (RBF) **[WHT07]**, Geodesic Distance Approximation **[NK02]**, Multi Dimensional Scaling (MDS) **[BG97]** and the Curvature Tensor Estimation **[GG03]**. These techniques can be classified into three main categories:

Problem Simplification – The Farthest Point Sampling (FPS) and Radial Basis Function (RBF) have been adopted as a problem simplification step in order to speed up the computation. All time-consuming processes will be computed on the sample points extracted and a smooth surface is to be recovered by the RBF at the end. Therefore, a good choice of simplification parameters such as sampling rate and regularization value, is one of the key controls of the quality of the matching result. This will be discussed in detail below.

Domain Transformation – The key component of domain transformation for bending invariant shape matching is the computation of geodesic distance. In this thesis, two geodesic distance approximations will be evaluated. They are the Dijkstra's algorithm-based **[WWTY04]** approach and the wavefront propagation based approach on triangular mesh **[NK02]**. Comparison will be given in this section in terms of their performance and quality. Their influence on the final matching results will be given in experimental result section later.

The Multi-Dimensional Scaling (MDS) is most important technique incorporated in this thesis. Since the matching result highly depends on the variation of the isometric signature namely, the MDS embedding between the input models, a good quality of MDS approach is a must to guarantee a satisfactory result. According to Elad et al's introduction in **[EK03]**, two classes of the MDS approach will be evaluated in this thesis for comparison; they are Classical MDS and the Fast MDS.

Feature descriptor Encoding – One of the major objectives of the proposed matching algorithm is the geometric feature matching ability. In this thesis, an arbitrary oriented curvature descriptor is introduced to serve this purpose. Together with the posture alignment process (with the MDS embedding) and the surface fitting, the curvature tensor value is served as a strong mapping criterion in the mapping algorithm. In the following section, a quick and robust curvature estimation method will be described in detail.

The core matching algorithm section explains in detail the steps of identification of correspondence on a input model from a pre-computed template model in a 3D space. The matching algorithm is divided into three steps: they included Posture Alignment, Surface Fitting and the Feature Matching Refinement.

In the Posture Alignment step, the best sign flipping direction is to be determined first between the MDS embeddings by using a newly introduced sign flip correction method, such a problem has already been addressed in **[EK03][JZ06].** Then, the template model will be transformed towards the input models by minimizing the Euclidean distance between the sample points on the MDS embedding.

In the Surface Fitting step, all vertices on the surface of the transformed template model will be fitted on the input model steadily by a number of iterations. Unlike other vertex-to-facet surface fitting algorithm **[JZ06][BBK07],** the proposed fitting algorithm approaches the vertex to the other surface by moving the vertex along it's normal vector. This method increase flexibility of vertex movement by ensuring it to moving along single direction rather than arbitrary directions. Illustration can be found in later section. Moreover, elimination will be taken for the surface mapping with either large difference of normal angle or Euclidean distance. In addition, a bi-directional mapping between the surface will be employed to improve the mapping on high curvature regions. At the end of each iteration of the surface fitting, the changed surface will be interpolated by a smoothing. Hence the newly transformed template model becomes a surface that fit the input model and meanwhile keeps the smoothness of the template model. By adopting any nearest neighborhood searching technology, anchor points can be located by comparing this domain with the input model.

Finally, the Feature Matching Refinement step encodes the curvature distribution information in the feature descriptor of each predefined anchor points on the template model. A corresponding anchor point mapping will be located on the input model by marching through all closest points around the instant-mapping between the surface fitted template and input model. Since the feature matching refinement process is an add-on process for particular vertex with significant geometric features, only a few numbers, for example around 10, of the vertices will be selected for refinement and the final result will affect particular regions of the surface only. According to the experiment, this refinement process has significant improvement on regions with variant curvature distributions. Examples can be found in the experiment result section.

3.1 Fundamental Techniques

3.1.1 Geodesic Distance Approximation

Geodesic distance is a metric defined as the length of a shortest path connecting two vertices that resides on the surface of a 3D geometry. The application of geodesic distance has been widely used as the proximities of deformable object in many recent 3D object matching researches **[LWH07][FL95][BBK07][JZ06][EK03]**.

In the proposed matching algorithm, geodesic distance is used as the primary technique in the following fields:

- The input dissimilarities of the Multi-Dimensional Scaling (MDS)
- The bounding radius for Curvature Estimation
- The farthest distance for the Farthest Point Sampling

Various methods have been proposed to estimate the approximate geodesic distance on 3D geometries **[NK02][SSKGH05][WWTY04].** Two famous approaches are the Dijkstra's algorithm-based approach **[WWTY04]** and the Wavefront Propagation approach which was introduced by Novotni et. al in **[NK02]**. The implementation will be explained in detail below.

3.1.1.1 Dijkstra's algorithm

The Dijkstra's algorithm-based approach is the simplest way to estimate the approximate geodesic distance on a well-defined triangular mesh. The computation of the geodesic distance *g(v)* of vertex *v* from the source point *s* is described by the pseudo code below.

Algorithm 3.1: Dijkstra's based geodesic distance computation

```
for each vertex $v$ in $V$ do
    $g(v) = \infty$
end for
$g(s) = 0$
push $s$ in stack memory $h$
repeat
    pop vertex $v$ from $h$ where $v$ has minimum $g(v)$
    for each adjacent vertex $v_a$ connecting to v do
        if $g(v) + length(v, v_a) < g(v_a)$ then
            push $v_a$ to stack $h$
            $g(v_a) = g(v) + length(v, v_a)$
        end if
    end for
until h is empty
```

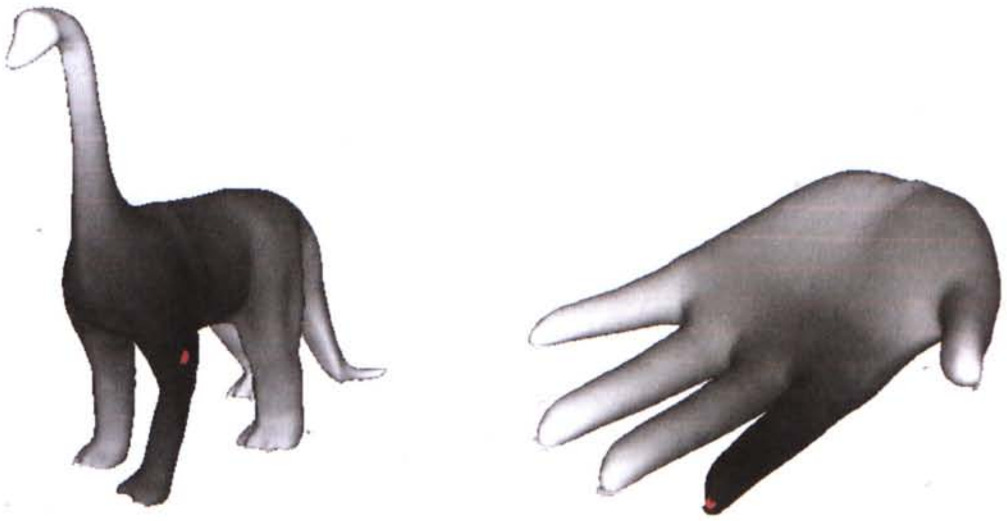

Figure 3-1: Geodesic distance from the source point (RED) by Dijkstra algorithm. Higher intensity for larger geodesic distance value.

3.1.1.2 Wavefront Propagation

The Wavefront Propagation geodesic distance approximation is presented by Novotni et. al. **[NK02]**. This method approximate the geodesic distance on vertices by iteratively propagating the geodesic distance value from the source point outwards along the surface. A modification of their work is computed as follows.

On a triangle with vertices v_1, v_2 and v_3, suppose that the geodesic distance values $T(v_1)$ and $T(v_2)$ have been assigned on v_1 and v_2, and we are looking for $T(v_3)$. Without the loss of generality, **[NK02]** assumed that the vertices v_1, v_2 and v_3 lie on the *xy*-plane. Then the pseudo-source point on the *xy*-plane can be determined by the intersections of two circles – which is centered at v_1 with radius $T(v_1)$ while the other is at v_2 with radius $T(v_2)$. For the two intersections, O_1 and O_2, the one with a larger distance from v_3 is employed as the pseudo-source. Then $T(v_3)$ is assigned as the distance from v_3 to the pseudo-source on the *xy*-plane.

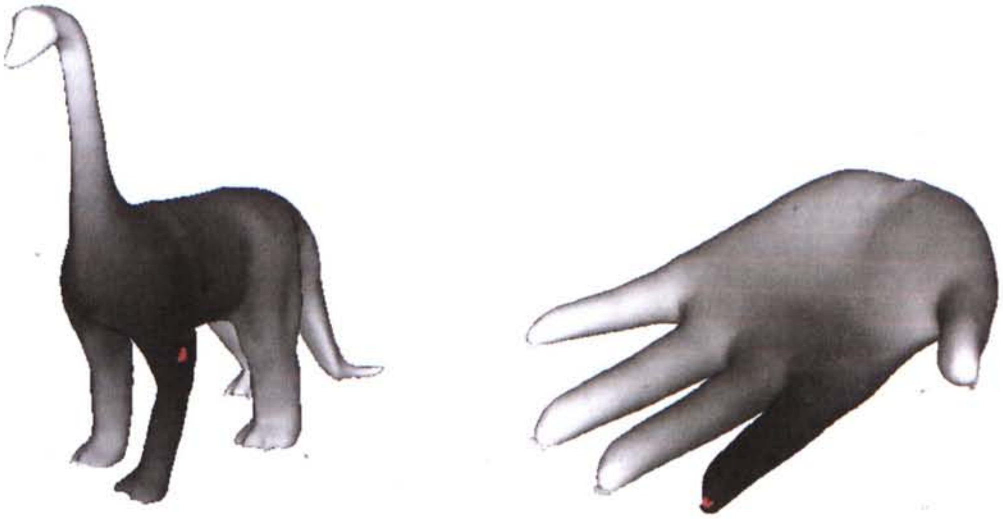

Figure 3-2: Geodesic distance from the source point (RED) by Wavefront propagation. Higher intensity for larger geodesic distance value.

According to the experiments and referring to **Figure 3-2:** Geodesic distance from the source point (RED) by Wavefront propagation. Higher intensity for larger geodesic distance value., the influence of the approximation accuracy is insignificant to the quality of the correspondence mappings and hence, it is not a major concern of geodesic distance approximation selection. On the other hand, an efficient approximation is necessary in order to improve the overall performance due to the frequent use of this technique. Therefore the Dijkstra's algorithm is served as the primary approximation method throughout the experiments.

3.1.2 Farthest Point Sampling

Sampling process is one of the possible ways to reduce the complexity of the matching problem. In this thesis, a most popular sample method – the Farthest Point Sampling (FPS) is employed.

The Farthest Point Sampling is introduced by Moenning et. al. in [MD03]. The main idea of this technique is to repeatedly placing the next sample point in the farthest point of the sampling domain. Whereas Fast Matching on Triangular Domain (FMTD) **[KS98]** represents a very efficient technique for the solution of front propagation problems and can be formulated as boundary value of partial differential equations.

Suppose the vertex set $S = \{s_1, s_2, ..., s_n\}$ is vertex set with n sample point, determined by the FPS algorithm. The implementation of will be explained in the pseudo code below:

Algorithm 3.2: Farthest Point Sampling

```
1: for each vertex $v$ in mesh $V$
```

```
        $g(v) = \infty$
end for
i = 0
$s_0 \leftarrow v_{random}$
push $s_0$ in stack memory $h$
repeat
    repeat
        pop vertex $v$ from $h$ where $v$ has min $g(v)$
        for each adjacent vertex $v_a$ connecting to v do
            if $g(v) + length(v, v_a) < g(v_a)$ then
                push $v_a$ to stack $h$
                $g(v_a) = g(v) + length(v, v_a)$
            end if
        end for
    until $h$ is empty
    $s_i \leftarrow$ vertex $v$ in $V$ where $v$ has max $g(v)$
    $i = i + 1$
until $i = n$
```

In this thesis, the sample rate will be default as 200 – 500 for the meshes with about 10,000 vertices. According to the experiment, this is the best rating range to balance the computation time and non-rigid transformations. And the average computation time of the FPS is always lower than 2 seconds.

FPS is employed as the primary sampling method based on the following criteria:

- High efficiency and easy implementation
- The samples are evenly distributed over the surface
- Suitable for freeform mesh which is constructed from triangular domain

The sample and statistical figures of the Farthest Point Sampling will be shown in the Experimental Result section later.

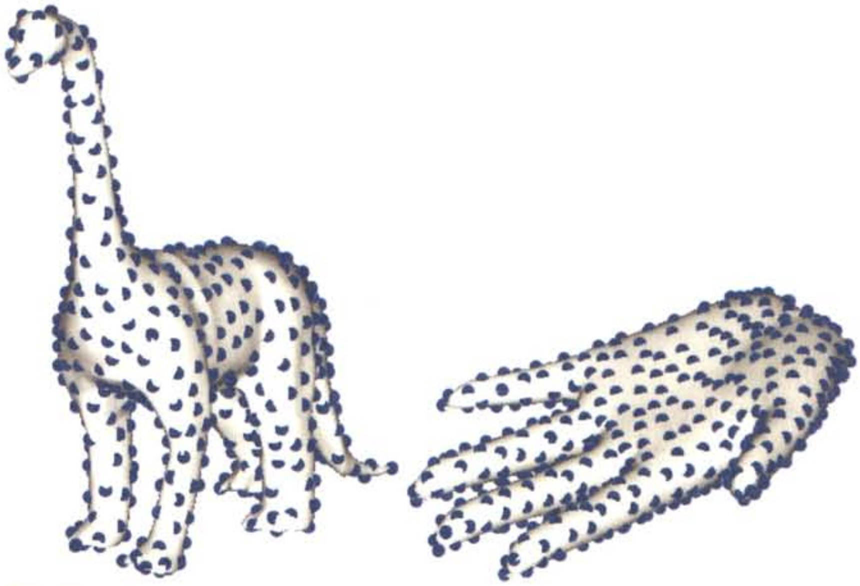

Figure 3-3: 300 Sample points (BLUE) on the triangular mesh of the dinosaur and hand models determined by the FPS algorithm.

3.1.3 Curvature Estimation

Curvature is a measurement of the local bending of a surface, which is represented by a local approximation of the tangent plane and orthogonal to the normal vector. Moreover, the two principal curvatures k_1 and k_2, with their associated orthogonal directions e_1 and e_2, are the extreme values of all the normal curvatures. Therefore, the average normal curvature K_H and the product of the normal curvature K_G of k_1 and k_2 are defined as:

$$K_H = \frac{k_1 + k_2}{2} \tag{3.1}$$

And

$$K_G = k_1 k_2 \tag{3.2}$$

These two curvatures represent an important local intrinsic property of a surface and as a result, it has been used for freeform shape matching widely, examples like **[ASDLD03][GGGZ05]**.

A recent 3D curvature tensor estimation which was proposed by Pierre et. al.

[ASDLD03] is a fast and robust method to find the principal curvature k_1 and k_2 on a triangular mesh surface region. The main idea is to estimate the curvature tensor at each vertex and interpolating these values linearly across triangles. The implementation of this approach is described below.

The curvature tensor of the vertices along the edge can be defined due to the existence of the obvious minimum (along the edge) and maximum (across the edge) curvature. Hence, for a particular surface region B, the curvature tensor matrix $T(v)$ is expressed as

$$T(v) = \frac{1}{|B|}\sum_{edges} \beta(e)|e \cap B|\overline{e}\overline{e}^t \qquad (3.3)$$

where v is an arbitrary vertex on the mesh, $|B|$ is the surface area around v over which the tensor is estimated, $\beta(e)$ is the signed angle between the normal to the two oriented triangles incidental to edge e (positive if convex, negative if concave), $|e \cap B|$ is the length of $e \cap B$ (value always between 0 and $|e|$), and e is a unit vector in the same direction as e.

In the experiment of the proposed matching algorithm, the surface region B is defined by including the vertices with the geodesic distance, sourced from vertex v, smaller than r. Where r is an arbitrary radius and is set to be $\frac{1}{20}$ of the bounding box diagonal by default.

Finally, the normal at each vertex can now be estimated by the eigenvector of $T(v)$ associated with the eigenvalue of minimum magnitude. The two remaining eigen-values k_{min} and k_{max} are estimates of the principal curvatures as v. Furthermore, as addressed in **[ASDLD03]**, the associated directions between the eigenpairs are switched. i.e. the eigenvector associated with the minimum eigenvalue is the maximum curvature direction l_{max}, and vice versa for l_{min}.

According to the experiment, this curvature estimation method is adopted in the proposed matching algorithm effectively. The application of this approach can be found in the Feature Matching Refinement section and more sample results can be found in the Experimental Result section later.

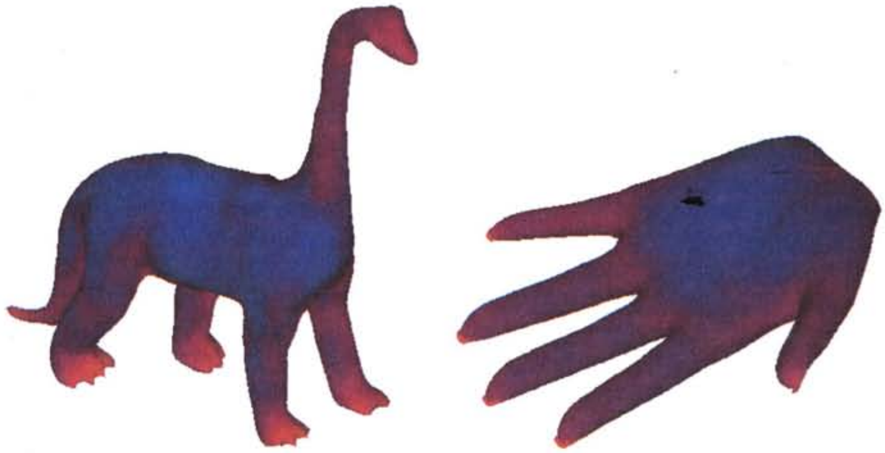

Figure 3-4: Mean curvature representation of a dinosaur and a hand. Red and blue color for higher and lower curvature respectively.

3.1.4 Radial Basis Function (RBF)

The Radial Basis Function (RBF) is a practical solution to the incompleteness of interpolating surfaces which is derived from 3D meshes. It has been widely adopted in applications like surface reconstruction **[WAAGJ05]**, medical imaging **[CFB97]** and geometry processing **[WHT07]**. However, their high computational requirements have confined a limited number of interpolation centers. These centers are the control points which are defined by the correspondence mappings in the proposed matching algorithm. The detail implementation of the Radial Basis Function will be discussed below.

Consider *f:* $R^d \rightarrow R$ *in* a real valued function of *d* variables that is to be approximated by *s:* $R^d \rightarrow R$, given the values $\{ f(x_i): i = 1, 2, \dots n\}$, where $\{x_i := 1, 2, \dots, n\}$ is a set of distinct point in R^d called the nodes of interpolation. Consider approximations of the form:

$$s(x) = p_m + \sum_{i=1}^{n} c_i \phi(\|x - x_i\|) \qquad x \in \Re^d, c_i \in \Re \qquad (3.4)$$

where p_m is a low degree polynomial, or is not present, $\|.\|$ denotes the Euclidean norm and *g* is a fixed function from R^+ *to* R. Thus, the radial basis function *s* is a linear combination of translates of the single radically symmetric function $g(\|.\|)$, plus a low degree polynomial. The space of all polynomials of degree at most *m* in *d* variables is denoted by π^d_m. Then the coefficients, c_i, of the approximation *s* are determined by requiring *s* to satisfy the interpolation conditions

$$s(x_j) = f(x_j), \qquad j = 1, 2, \dots, n \qquad (3.5)$$

Together with the side conditions

$$\sum_{j=1}^{n} c_j q(x_j) = 0, \qquad \textit{for all } q \in \pi^d_m \qquad (3.6)$$

Hence, the linear equation system can be rewritten as

$$\begin{bmatrix} G & p_m \\ p_m^T & 0 \end{bmatrix} \begin{bmatrix} c \\ 0 \end{bmatrix} = \begin{bmatrix} s \\ 0 \end{bmatrix} \tag{3.7}$$

Where G is the square matrix of the basis function $g(||x_i - x_j||)$ and $g(r) = |r|$ for any scalar r. And the unknown coefficients c_i can be found by solving the equations with LU decomposition.

In addition, the smoothness of the transformation with the RBF can be further controlled by adopting a regularization parameter λ in matrix G. Therefore, the final linear equations of (*) is presented as

$$\begin{bmatrix} K & p_m \\ p_m^T & 0 \end{bmatrix} \begin{bmatrix} c \\ 0 \end{bmatrix} = \begin{bmatrix} s \\ 0 \end{bmatrix} \tag{3.8}$$

Where $K = (G - \lambda I)$ and I is the identity matrix of an appropriate size.

In this thesis, the setting of the regularization parameter λ is always inversely proportional to the certainty of the control points mapping. That is, λ will be set to be very high at the beginning of the matching steps, and gradually decrease to zero at the final transformation of the template model.

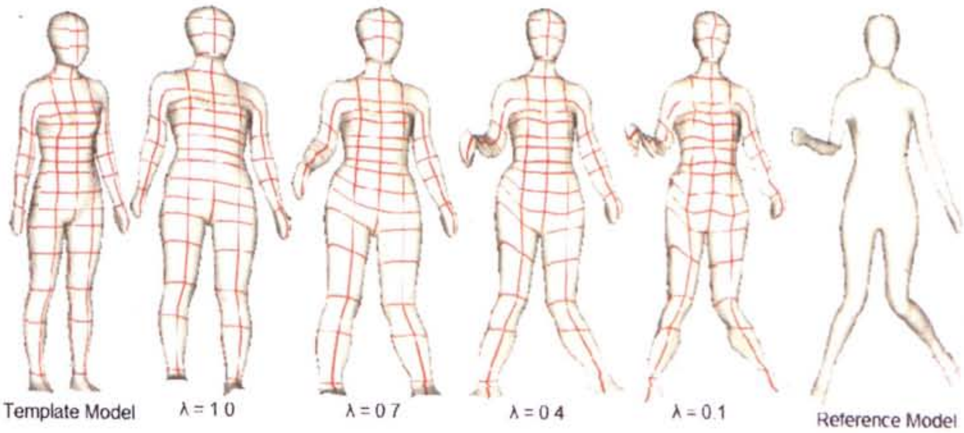

Figure 3-5: Transformation of the Template (Left) to Input model (Right) with the RBF.

Transition with different λ in 2nd column to 5th column.

3.1.5 Multi-dimensional Scaling (MDS)

Multidimensional Scaling (MDS) is used to find the spatial structure of a data set by identifying the dissimilarity information among them. This technology has been applied in many diverse applications, such as semantic structure analysis of words **[FL95],** texture analysis **[EK01]**, political science and geography analysis,…etc. In most recent 3D object matching researches, this technique has been widely used to solve the freeform object matching problem.

Several variations of the MDS algorithm were proposed by the researchers **[FL95][EK01][EK03]**. The most generic MDS approach is called the metric multidimensional scaling **[EK01]**. This approach originally assigns each item to a k-d point. Then it examines every point, computes the distances from the other $N - 1$ point and moves the point to minimize the discrepancy between the actual dissimilarities and the estimated k-d distances. However, this approach is far from perfect for an efficient demanding matching algorithm. Two other classes of MDS approach namely, the Classical MDS and the Fast MDS, have been evaluated in the research of **[EK03]**. Their implementations and advantages will be discussed in the following two parts.

3.1.5.1 Classical MDS

Let the coordinates of n points in a k dimensional Euclidean space R^k be given by x_r $(r = 1, \ldots, n)$, where $x_r = [x_{r1}, x_{r2}, \ldots, x_{rk}]^T$. The Euclidean distance between the r-th and the s-th points is given by

$$A = [x_r - x_s]^T [x_r - x_s] \qquad (3.9)$$

Let the inner product matrix be B, where the rs element given by $[B]_{rs} = b_{rs} = x_r^T x_s$. Given the squared distance matrix A, the inner product matrix is given by

$$B = -\frac{1}{2} JDJ \tag{3.10}$$

Where

$$J = I - \frac{1}{n} 1 \cdot 1^T$$

$$1_{1\times n} = [1,1,\ldots,1]^T$$

We also have that $B = XX^T$, where $X = [x_1, \ldots, x_n]^T$ is the $n \times k$ matrix of the coordinates. The inner product matrix B is symmetric, positive semi-definite, and of rank k. Therefore, B has k non-negative eigenvalues and n-k zero eigenvalues. The matrix B can be expressed in terms of its spectral decomposition,

$$B = V\Lambda V^T,$$

Where

$$\Lambda_{nxn} = diag(\lambda_1, \lambda_2, \ldots, \lambda_k, 0, \ldots, 0)$$

For convenience, the eigenvalues of B are ordered such that $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_k \geq 0$. Hence, the required coordinates are given by using the non-zeros sub-matrix Λ_{kxk} and the corresponding eigenvectors sub-matrix V_{nxk}.

$$X_{n\times k} = V_{n\times k} \Lambda_{k\times k}^{\frac{1}{2}}$$

The classical scaling is considered to be an efficient algebraic approach to solving MDS problems. It can be calculated in $O(n^2)$, where n is the number of sample points in the given model. This is because there is a need to find only the first m eigenvalues and their corresponding eigenvectors, which can be computed by variations of the power method. The figure below illustrated the MDS embedding computed by the Classical MDS

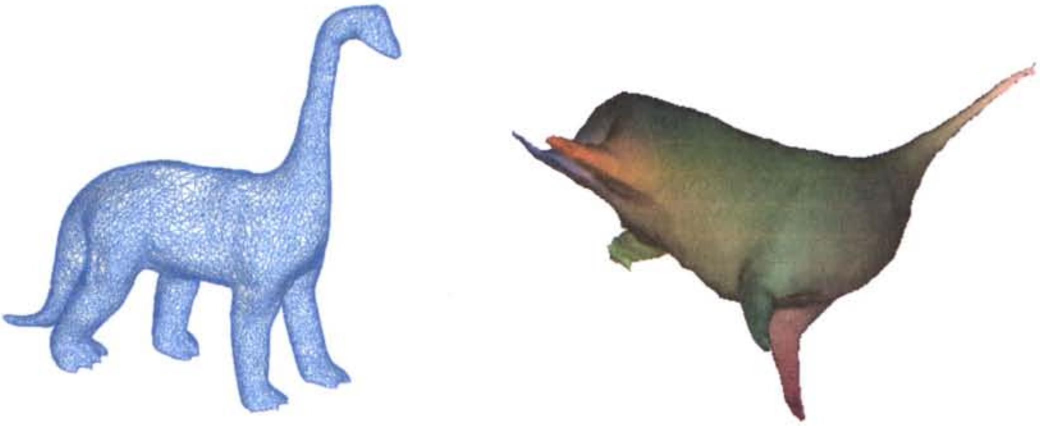

Figure 3-6: The Multi-Dimensional Scaling result (Right) of a dinosaur (Left).

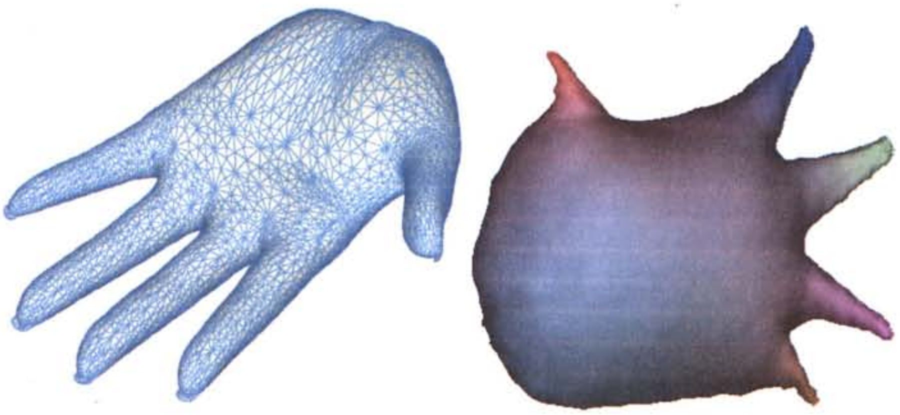

Figure 3-7: The Multi-Dimensional Scaling result (Right) of a hand mesh (Left).

3.1.5.2 Fast MDS

The Fast MDS is an efficient dimension scaling approach proposed by Faloutsos et.al in [FL95]. The efficiency of this approach is *O(nm)*, where *m* is the target dimension. This technique solves the MDS problem by recursively generating a new dimension on new the domain at each step and hence, it m-dimensional coordinates after applying the recursion m times.

The basic idea is to project the vertices on a carefully selected line *L*. Where *L* is connected by two far distanced vertices O_a and O_b. Then all other vertices are projected on *L* using cosine law

$$x_i = \frac{d_{a,i}^{\ 2} + d_{a,b}^{\ 2} - d_{b,i}^{\ 2}}{2d_{a,b}} \qquad (3.11)$$

The next step is to project all items to an *(n-1)* hyper-plane *H* that is perpendicular to the *L* and to generate a new distance matrix according to,

$$D_{ij}^{\ 2} = d_{ij}^{\ 2} - (x_i - x_j)^2$$

This step is to be repeated *m* times. At each step, the calculated x_i, $i = (1,.. , n)$, are the newly added dimension coordinates.

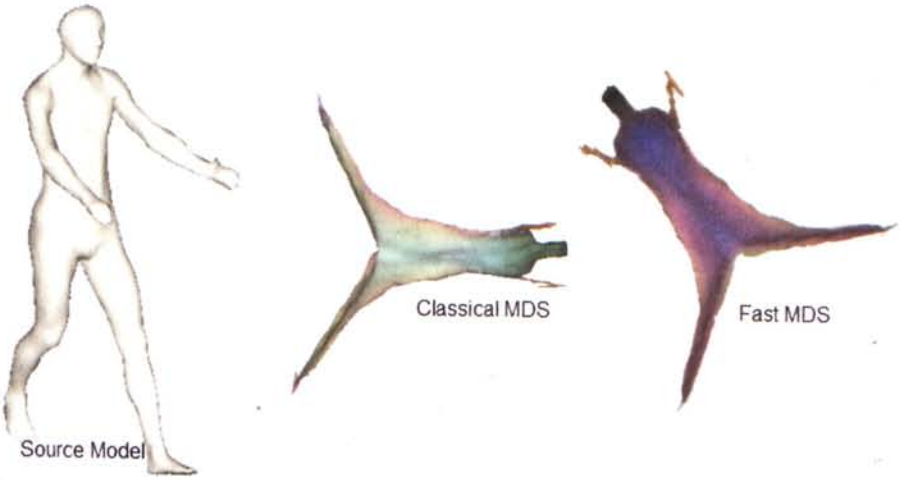

Figure 3-8: The MDS transformation result of the source model (LEFT) with Classical MDS (Middle) and Fast MDS (Right) respectively.

Although the efficiency of the Fast MDS is better than the Classical MDS, experiments indicate that the Classical MDS is more practical in use comparatively. The main reason is due to the arbitrary determination of the pivot in Fast MDS which results in an inconsistent orientation of the MDS even for two isometric similar domains. Therefore, it poses a non-solvable sign flip problem in the later posture alignment process. Examples and comparisons of these two MDS techniques will be given in the experimental result section for analysis.

3.2 Matching Processes

This section explains in detail the core matching process of the proposed correspondence matching algorithm. The include algorithm has three steps: Posture Alignment, Surface Fitting and the Feature Matching Refinement.

The matching process begins from two given clockwise-piecewise triangular meshes in $\Re^3$ space, denoted as *U* and *V* for the template model and input model in $\Re^3$ space respectively. Where the template model *U* consists of a point set $u\{u_0, u_1, \ldots, u_n\}$ and the input model *V* consists of a point set $v\{v_0, v_1, \ldots, v_m\}$. The problem definition as following:

For given *U*, *V* and predefined anchor point set a^{u} on *U*, we are going to find a mapping *f* by minimizing the energy function E such that:

$$min\ E(f) = min \sum ||V - f(U)||^2, \qquad \textbf{(3.12)}$$

and also a set of new anchor point

$$\mathrm{a}^{\mathrm{v}} = f(\mathrm{a}^{\mathrm{u}}) \qquad \textbf{(3.13)}$$

on input model *V*. For which some important points of V have similar feature descriptors to a^u for their corresponding point in a^u.

Since the matching process is split into three steps, the function *f* can be regarded as a composition of the corresponding partial functions in each step. And thus,

$$f(U) = f_{F_V}(f_{S_V}(f(_{P_V}(U))) \qquad \textit{\textbf{(3.14)}}$$

where f_{P_V}, f_{S_V} *and* f_{F_V} are three non-rigid transformation functions of the posture alignment, surface fitting and feature matching refinement processes towards model V respectively.

The remaining uncertainties are the determination of those three transformation functions f_{P_V} , f_{S_V} *and* f_{F_V} . In the following three sections, the numerical implementation of them will be described in detail.

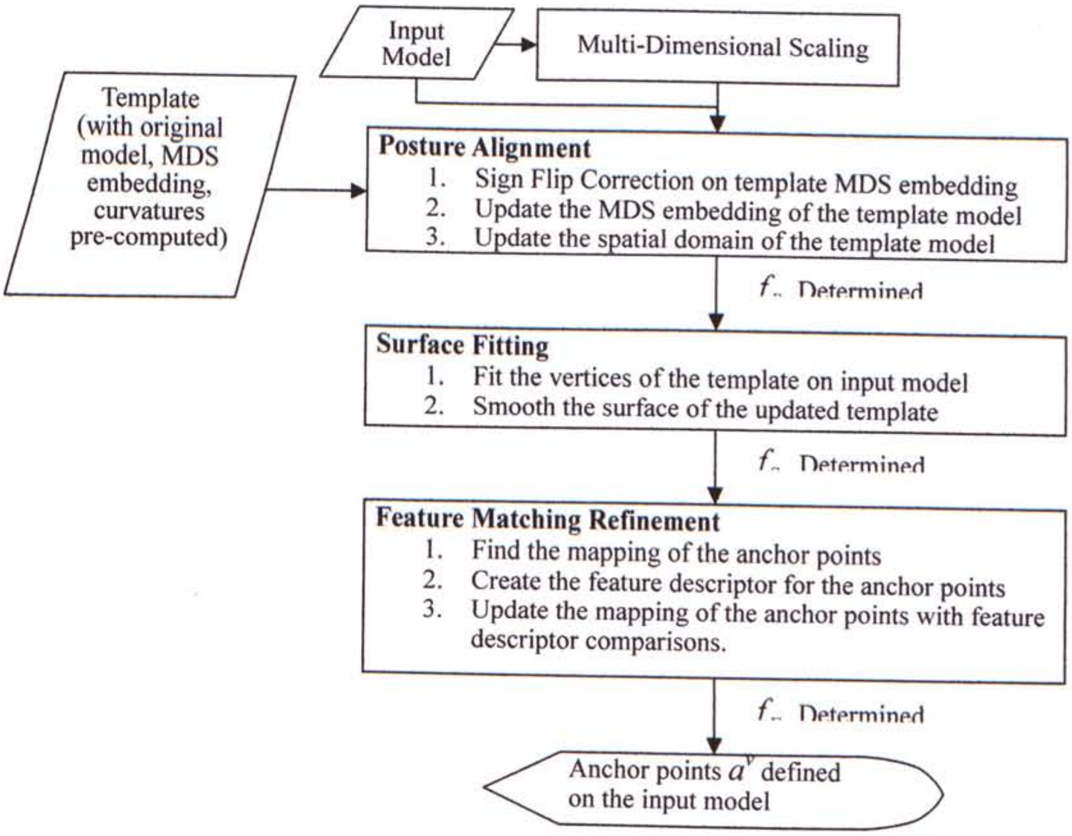
Input Model
Multi-Dimensional Scaling
Template (with original model, MDS embedding, curvatures pre-computed)
Posture Alignment
1. Sign Flip Correction on template MDS embedding
2. Update the MDS embedding of the template model
3. Update the spatial domain of the template model
f Determined
Surface Fitting
1. Fit the vertices of the template on input model
2. Smooth the surface of the updated template
f Determined
Feature Matching Refinement
1. Find the mapping of the anchor points
2. Create the feature descriptor for the anchor points
3. Update the mapping of the anchor points with feature descriptor comparisons.
f Determined
Anchor points a^V defined on the input model

Figure 3-9: Flowchart of the proposed matching algorithm

3.2.1 Posture Alignment

The posture alignment step is an alignment process for the posture of the template model to the input model. To achieve the bending invariant matching goal, the Multi-Dimensional Scaling (MDS) technique will be employed as the bending invariant signature for the meshes. Firstly, *k*-D MDS embeddings of the template model is aligned with the reference MDS embedding with a Sign Flip Correction. This process is proposed to solve the 'sign flipped' problem which induced by the arbitrary determination of the axes directions in the MDS embeddings. It is essential to overcome this problem at the very beginning of the whole matching process in order to provide a good initial condition for further computation. Once the MDS embedding is aligned correctly, the template model is then being transformed towards the input model smoothly. In order to enhance performance and to alleviate the smoothness problem, the Farthest Point Sampling (FPS) and Radial Basis Function (RBF) are incorporated in all non-rigid transformation step of the template model.

In numerical presentation, the objective of the Posture Alignment step is to find the function P_V that transform the template model U towards the input model V, such that

$$U' = P_V(U) \tag{3.15}$$

$$U'_{MDS} = P_V^{MDS}(U_{MDS}) \tag{3.16}$$

where U' and U'_{MDS} are the transformed domain of U and U'_{MDS} respectively. The function P_V here depends on the result of P_V^{MDS}, that is, the MDS embedding alignment process. Once P_V^{MDS} is determined, a binary one-to-one mapping between the vertices of the MDS embedding can be obtained. And the function P_V can be computed by using this mapping as control points in the RBF transformation. The computation of P_V and P_V^{MDS} will be discussed in the following part.

3.2.1.1 Sign Flip Correction

As the shapes of human bodies are approximately isometric, their shapes in the MDS embeddings are quite similar to each other. Ideally, the correspondence between the points $u_i \in U$ and $v_j \in V$ can be determined by the closest point search. However, such mapping between U_{MDS} and V_{MDS} is neither bijective nor robust. According to the experiment, the main challenge comes from the random selection for the sign of eigenvalues (therefore the direction of eigenvectors) in the MDS analysis. Thus the shapes of U_{MDS} and V_{MDS} can greatly differ in terms of their axis directions – called sign-flip. Therefore, a robust sign flip correction is introduced in this thesis.

Various solutions have been proposed in various shape matching researches on solving the sign flip problem that used the MDS technique. Caelli et. al in **[CK04]** proposed a dominated sign correction by ensuring that there are more positive entries in each eigenvector all the time. However, this approach is highly undesirable since according to practical experiments, the numbers of positive and negative entries are always very close.

Under the exhaustive search framework proposed by Shapiro and Brady in **[SB92]**, the alignment of the MDS embedding can be achieved by finding the combination of axes swapping which minimizes a shape different metric. For instance, there are $2^6 = 64$ sign flipping combination for 6D MDS embeddings. Therefore, 2^d different sign flipping functions can be defined for a point u in the d-dimensional MDS domain.

However, as in the presence of eigenmode switching, their determination rule will be violated in most symmetric models matching problem. Hence a detail implementation and improvement work are presented in the following part.

Suppose that two human body objects $U \in \Re^3$ and $V \in \Re^3$ are given. Their corresponding MDS embedding are shown in Figure 3-10 below.

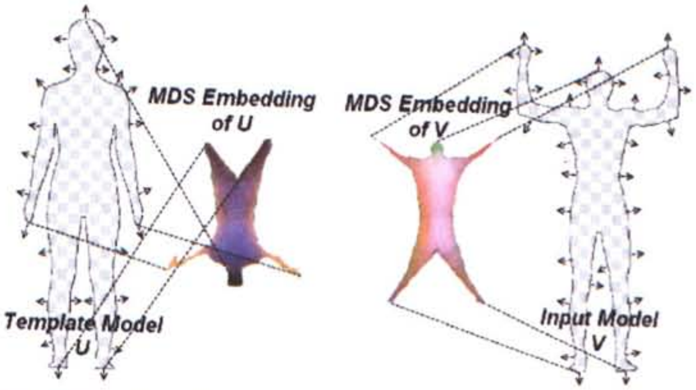

Figure 3-10: The MDS embeddings (Colored) of two human body objects (checkerboard) with surface normal (arrows) and vertex linkages (dashed line) between spatial and MDS domains.

Firstly, the framework of the proposed correction scheme was originated from the greedy approach by Shapiro and Brady. Hence, alignment of the MDS embedding can be achieved by finding the combination of axis that with minimum. For instance, there are $2^6 = 64$ sign flipping combination for a 6-D MDS embedding. Therefore, the flipped position of vertex u on U in the iteration d is presented as

$$\phi_d(u) = v \times (-1)^{d \bmod 2,4,\ldots,k} \qquad for \;\; 0 \le d \le 2^k \qquad \textbf{(3.17)}$$

Afterwards, the template model will be transformed by $f(u)$ towards the input model according to the 'best mapping' defined between U_{MDS} and V_{MDS}, such that

$$E(f^{MDS}) = \min \sum_{i=0}^{n} \left| v_{MDS} - f^{MDS}\left(\phi_d\left(u_{i_{MDS}}\right)\right) \right|^2 \qquad \textit{(3.18)}$$

is the energy function to be minimized to determine f^{MDS}. The relative transformation function *f(u)* in the spatial domain is then determined by the ICP mapping on their embeddings between v_{MDS} and $f^{MDS}\left(\phi_d\left(u_{i_{MDS}}\right)\right)$. Hence *f(u)* is determined by minimizing another energy function

$$E_\lambda(f) = \min \sum_{i=0}^{n} \left|v - f(u)\right|^2 + \lambda\left|Lf\right| \qquad (3.19)$$

where $\lambda|L f|$ is a Laplacian smoothness term that controls the surface smoothness by the regularization parameter λ. A tricky setting here, is that λ must be set to be of great value. This great value help preserving the smoothness and posture of the original surface after the transformation *f(u)*. Hence *f(u)* can be regarded as a pure orientation alignment of the template model to the input model. Figure 3-11 illustrated an example of this scenario. This setting is very important for computing the distance errors for the cost function introduced in next part. According to the experiment, $\lambda = 10^8$ would be large enough for all testing samples.

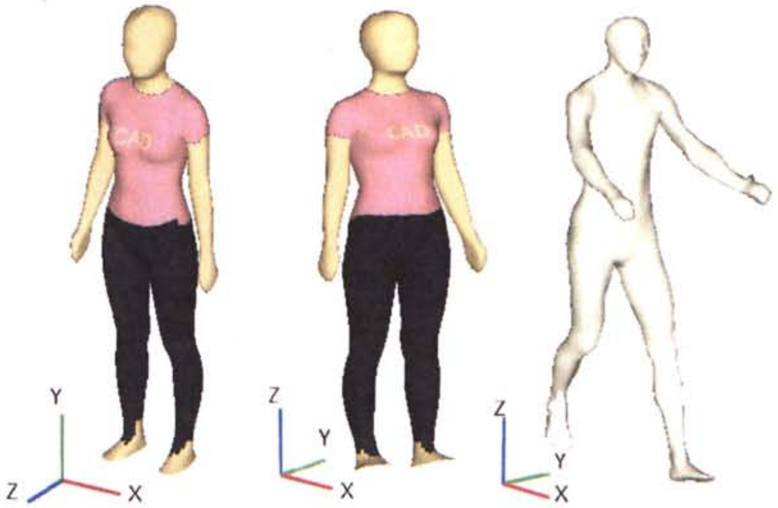

Figure 3-11: Transformation result (middle) of the template model (left) to the input model (right) with $\lambda = 10^8$. The result transformed with template's smoothness and input model's orientation.

In order to evaluate the validity of current sign flip combination for *f(u)*, a cost function *C(T)* is defined as

$$C(T) = \sum_{i=0}^{S}\sum_{j=0}^{S}\left|\left\|u_i - u_j\right|^2 - \left|T_\lambda(u_i) - T_\lambda(u_j)\right|^2\right| \qquad \textit{(3.20)}$$

for $0<i,j<S$ and S is the number of sample point.

This costing function is a generic distance changes function as adopted in Shapiro and Brady's approach. Obviously, this costing function has its own drawback. It results in two possible $T_\lambda(u)$ that holds the same minimum cost as illustrated in Figure 3-12.

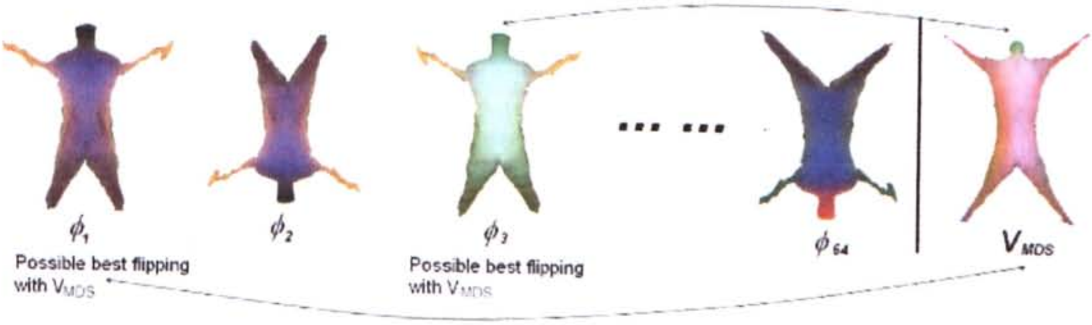
ϕ_1
ϕ_2
ϕ_3
ϕ_{64}
V_{MDS}
Possible best flipping with V_{MDS}
Possible best flipping with V_{MDS}

Figure 3-12: Illustration of several possible combination of axis flipping. It can be observed that two flipping ϕ_1 and ϕ_3 have minimum cost value.

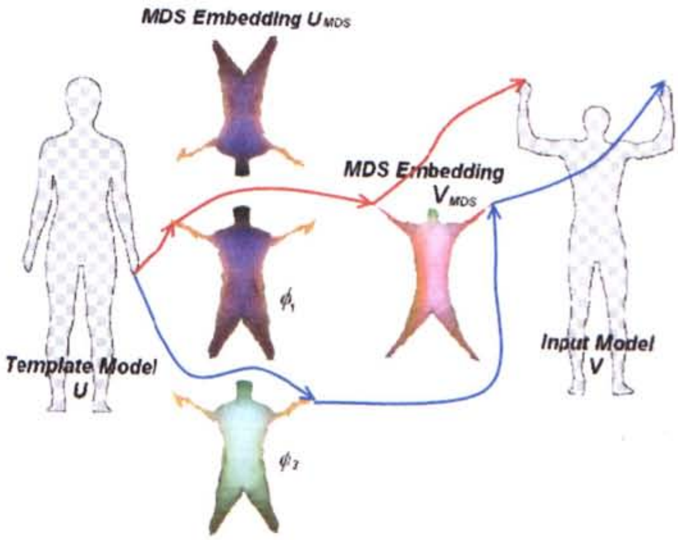

Figure 3-13: The mapping RED, BLUE line obtained by flipping U_{MDS} to ϕ_1 and ϕ_3 respectively. It shows that the mapping obtained by ϕ_1 is incorrect because it is left-right swapped.

Such drawback is caused by the inconsideration of surface normal after the transformation of $T_\lambda(u)$. Therefore, the costing function should be modified in order to guarantee a satisfactory result. It can be achieved by adopt a normal-offsetting technique in the sign flip correction.

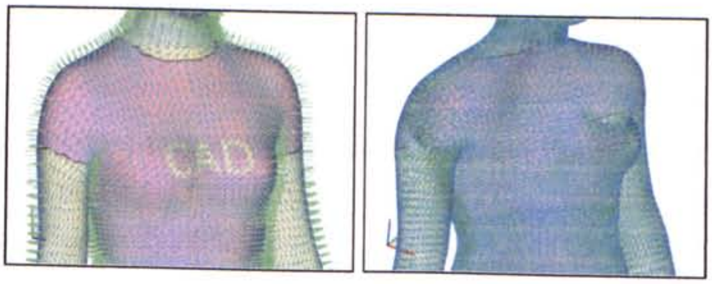

Figure 3-14: Left and right image illustrate the transformation of the spatial domain by ϕ_3 and ϕ_1 respectively. The surface normal of the spatial domain is flipped using ϕ_1.

Referring to Figure 3-14, the costing *C(T)* obtained by ϕ_1 and ϕ_3 are identical except that the surface normal of the spatial domain is inversed. The cost function *C(T)* thus, can be improved by replacing the vertex coordinate v with $(v + \omega n)$, where n is the surface normal and ω is the offset distance (ω is set to be 1.0 in general).

Hence, the costing function *C(T)* in *(3.20)* is replaced by

$$C'(T) = \sum_{i=0}\sum_{j=0} \Big| \, |(u_i + \omega n_i) - (u_j + \omega n_j)|^2 - |(T_\lambda(u_i) + \omega n_{Ti}) - (T_\lambda(u_j) + \omega n_{Tj})|^2 \, \Big|, \qquad (3.21)$$

Figure 3-15 illustrates an example with offset coordinate presented. A correct and unique minimum costs can therefore, be determined by the new costing function.

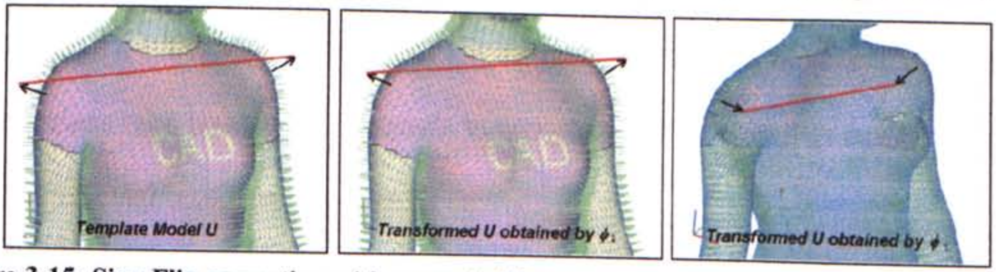
Template Model U
Transformed U obtained by ϕ_3
Transformed U obtained by ϕ_1

Figure 3-15: Sign Flip correction with normal offset component in costing function. Significant difference can be distinguished by comparing the length of the RED line which constructed by the normal offset of the vertices between the template model (Left) and the transformed models (middle: ϕ_3 and right: ϕ_1).

The pseudo code of the whole sign flip correction algorithm is presented as follow:

Algorithm 3.3: Sign Flip Correction

```
initialize $\lambda = \infty$, $C'(T_{min}) = \infty$
```

```
for $d = 0$ to $2^k$
    for each vertex $u_{MDS}$ in $U_{MDS}$
        $u'_{MDS} = \phi_d(u)$
    end for
    define the 1-to-1 mapping $f$ between $U'_{MDS}$ and $V_{MDS}$
    define the non-rigid transformation $T_{\lambda,f}$ for $U'_{MDS}$ towards $V_{MDS}$
    update $U'_{MDS} = T_{\lambda,f}(U'_{MDS})$
    compute $C'(T)$ by $-|(T_\lambda(u_i) + \omega n_{Ti}) - (T_\lambda(u_j) + \omega n_{Ti})|^2|$,        (3.21)
    if $C'(T) < C'(T_{min})$
        $T_{min} = T$
    end if
end for
update $U_{MDS} = T_{min}(U_{MDS})$
```

The proposed sign flip correction is a very reliable solution to solve the symmetric models matching problem. According to the experiment, the accuracy is achieved 90% for symmetric samples like human models, hand, animals and some primitive shape. Moreover, this correction scheme is very effective even for meshes with large vertex count. Since the operations in flipping iterations do not involve any topology linkage information, the vertex set can be reduced to a coarser level by the Farthest Point Sampling (FPS). And hence, the efficiency cost of the sign flip correction is $O(n^k)$, where k is the MDS dimension. Experimentally, the sign flip correction for k = 6 dimensions take less than 10 seconds to complete. More examples can be found in the experimental result section later.

3.2.1.2 Input model Alignment

After a proper sign flip correction of the MDS embedding, two isometric similar MDS embeddings can be used for mapping the anchor points between the template and input models. In many reviewed shape matching researches **[JZ06][CR03]**, they stop further exploring on a more accurate surface matching and hence current result is served as a coarse level of object recognition technique, rather than anchor point matching. Nevertheless, since the work in this thesis is to propose a robust vertex mapping with feature descriptor, further computation is needed to improve the mapping result on MDS embedding alone.

Some shape matching algorithm such as **[JZ06][PDJ93][WWJGS06]**, apply the rigid transformation on the input model in a single pass. This approach is based on the assumption that the initial mapping f of the control points is highly reliable. Nevertheless, this assumption is impractical for bending invariant shape matching, especially for the meshes with sharpen regions or local maxima. Therefore, Chui et. al proposed an iterative RPM-TPS transformation scheme for non-rigid alignment [CR03]. They determined the non-rigid Thin-Plate Spline (TPS) transformation function $T(u)$ and the correspondence function $f(u)$, and then increase the sensitivity between the control points steadily by reducing the regularization power. This approach is suitable for a good, but not perfect initial guess of control point mappings and hence, it is adopted in the posture alignment process.

The input model alignment process first finds the mapping f between the sample point set of two models in U_{MDS} and V_{MDS}. The number of sample point R should be increase gradually from a small number, for example $R = 50$, to avoid trapping on local maximum at the beginning of the iterations. Next, the transformation function T_λ will be found by using the control points u and $f(u)$. Initially, the value of the regularization λ is set to be the mean distance between all vertices mappings in U_{MDS} and V_{MDS}. As explained in **[JZ06]**, this scale-dependent assignment of λ is robust to the scale of the point sets. Then λ will be decreased by about 1/10 gradually after each loop. According to the experiment, the changes should be converged to ε within 10 iterations. The implementation of the Posture Alignment Step will be described in the following pseudo code.

Algorithm 3.4: Posture Alignment

```
update $u_{MDS}$ by the Sign Flip Correction
initialize $\lambda$, $R$
repeat
    define the 1-to-1 mapping $f$ between $U_{MDS}$ and $V_{MDS}$
    define the non-rigid transformation $T^{MDS}$ for $U_{MDS}$ towards $V_{MDS}$ by $f$
    update $U'_{MDS} = T^{MDS}(U_{MDS})$
    define the non-rigid transformation $T$ for $U$ towards $V$ by $f$
    update $U' = T(U)$
    set $e = u' - u$
    update $U = U'$
    update $U_{MDS} = U'_{MDS}$
    decrease $\lambda$
    increase $R$
until $e < \varepsilon$
```

As the captioned purpose, the "posture" of the template model has been aligned with the input model. Yet, the surface of the transformed template is not fitted on the input model and hence error will be accumulated in convex regions, refer to the 6th column in Figure 3-16. This problem leads to the next step, the surface fitting, to improve the mapping result by eliminating the ambiguity of anchor point mappings between the transformed template and input model. Figure 3-16 below illustrated an example of the posture alignment transition for the spatial domain and MDS embedding.

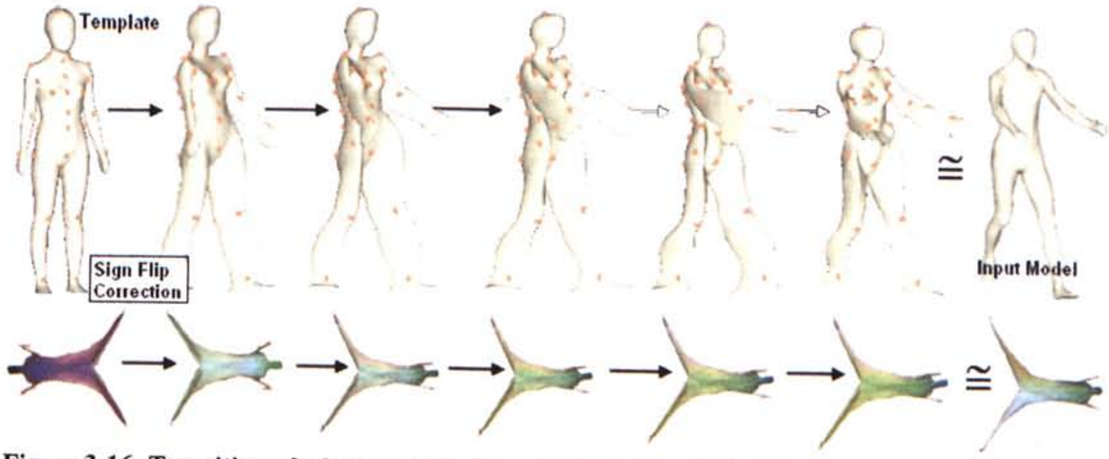

Figure 3-16: Transition of a human body from the template (left) towards input model (right). The spatial domain (top row) and their corresponding MDS embedding in the same iteration throughout the transformation.

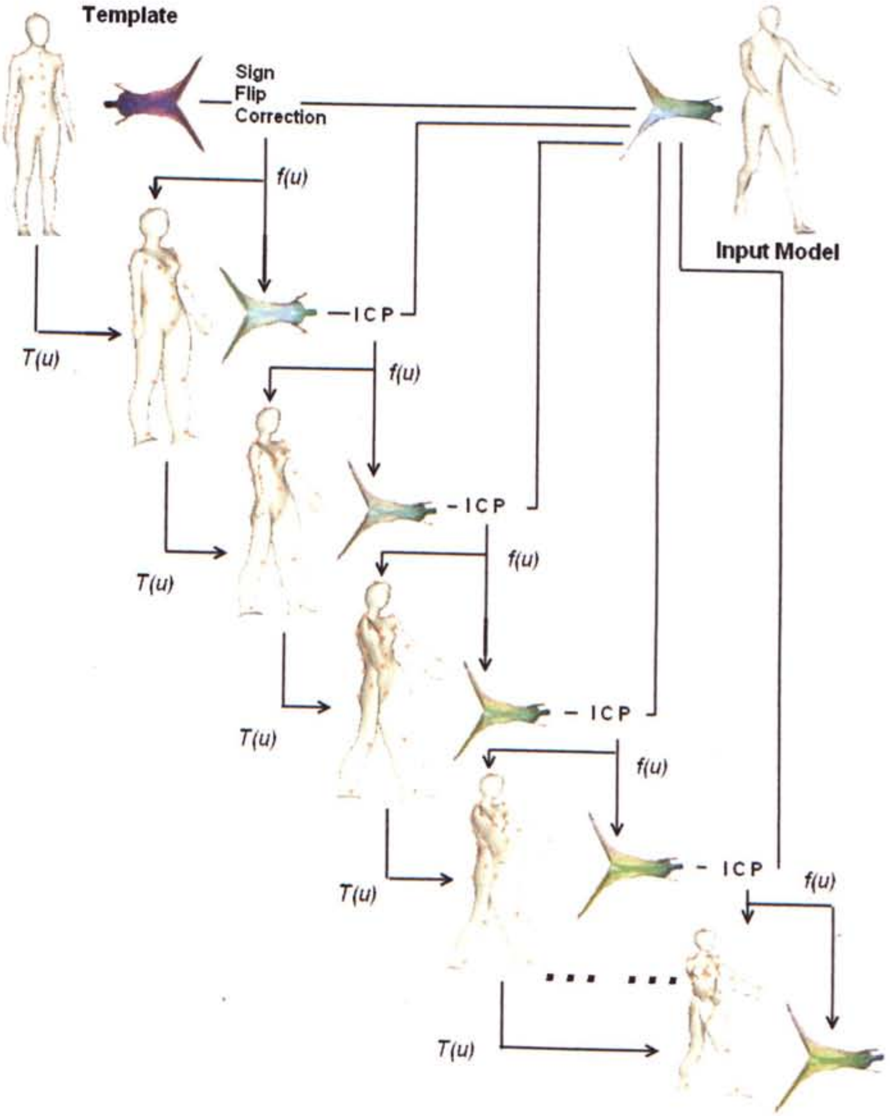

Figure 3-17: Flow of the posture alignment step and the linkage between the spatial, MDS domain of the template and input models.

3.2.2 Surface Fitting

Given a posture aligned template model *U'* and input model *V*, the fitting process in this section further increase the accuracy of the anchor point matching by optimizing the surface of *U'*. The technical details will be discussed below.

Since the posture aligned model *U'* is the transformation product which obtained by optimizing the global energy function, the vertices set *{u}* on the surface are not tightly mapped on the input model. In order to map the anchor points of the template on the input model, the whole surface of the template must be fully fitted on the input model without loss of continuity. Also, the smoothness of the surface should be kept during the fitting process. Figure 3-18 shows an example before and after the surface fitting.

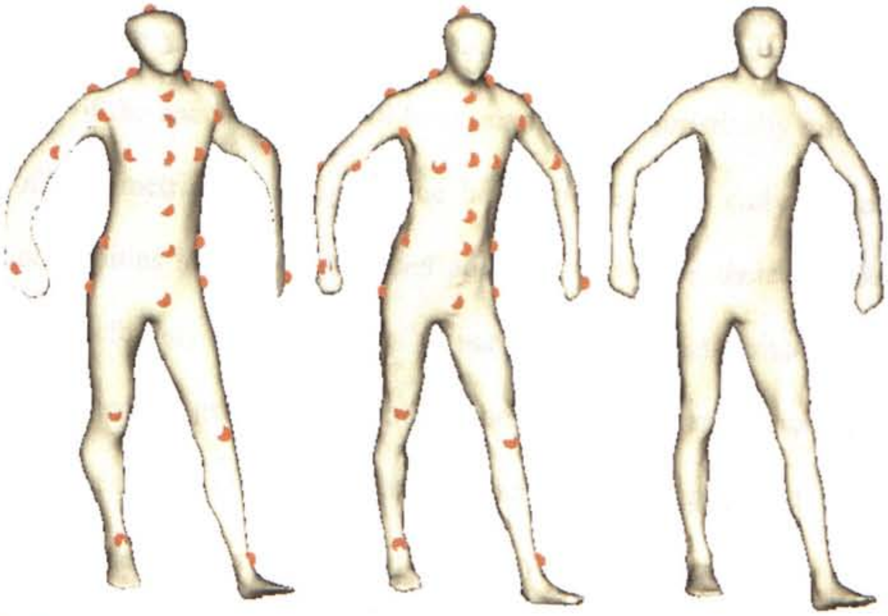

Figure 3-18: Human body mesh (left) before surface fitting and (middle) after surface fitting to the (Right) Input model.

One common approach for finite surface fitting is called multi-resolution alignment, presented in **[BBK07]**. The concept of the multi-resolution alignment is very simple. The core process is to define the transformation function and correspondence mappings in a coarse level by simplifying two meshes. Then this process will be repeated iteratively by increasing the resolution of the meshes, and the transformed result is interpolated with the one in previous iterations. The mild interpolation of transformed domain prevent the sudden jump occurred in mis-mapping anchor point pairs. And hence errors will not be accumulated in local maxima. However, the simplification step in the multi-resolution alignment is highly depends on a small distortion level of the mesh in local maxima. For example, it is infeasible to apply multi-resolution on the arm/ hand region of the human body.

Jain et. al have addressed the local maxima problem in **[JZ06]** and proposed a proximity-aided matching step right after the non-rigid transformation of the input model. This proximity-aided process first selects a number of point mappings, where these mappings are supposed to be best matched and not too closed to each other. Then the rest of the anchor points will be re-mapped by heuristically minimizing the distance of two metrics: the L_2 distance between the MDS embeddings and the geodesic proximities to the best-matched point set. However, their method is not a general solution for any bended objects scenario. Since the determination of the best mapped points and the geodesic proximities metric will be encountered required two assumptions: the quality of the best-mapped point set is very good and, the stretching effect of the whole mesh is small. Jain et. al also noted this problem and hence they restricted the number of anchor points to be fixed to three vertices only.

As the mentioned solution above poses many restrictions in usage, such as best-matched anchor point pre-requisites, small distortion level and similar geodesic proximities…etc, they are not considered to be a robust surface fitting approaches. In the following part, a robust, local matching and non-restricted fitting solution will be introduced.

Numerically, the surface fitting process is denoted as a transformation S_V, as stated in the global transformation function, where S_V is obtained by iteratively optimizing the surface smoothness and fitness between *U'* and *V*. The optimization of these two terms will be explained in the following section.

3.2.2.1 Optimizing Surface Fitness

Given a posture aligned template model *U'* and a input model *V*, each vertex u on *U'* are going to be fitted on the surface of *V*. Some generic approaches fit the surface from one to the other by the vertex-facet intersection formula **[WHT07]**. This method has two critical problems. Firstly, wrong mapping will be happen on near but opposed surfaces. Secondly, the surface smoothness will be distracted on dense regions. Figure 3-19 shows an example of surface fitting with this generic approach.

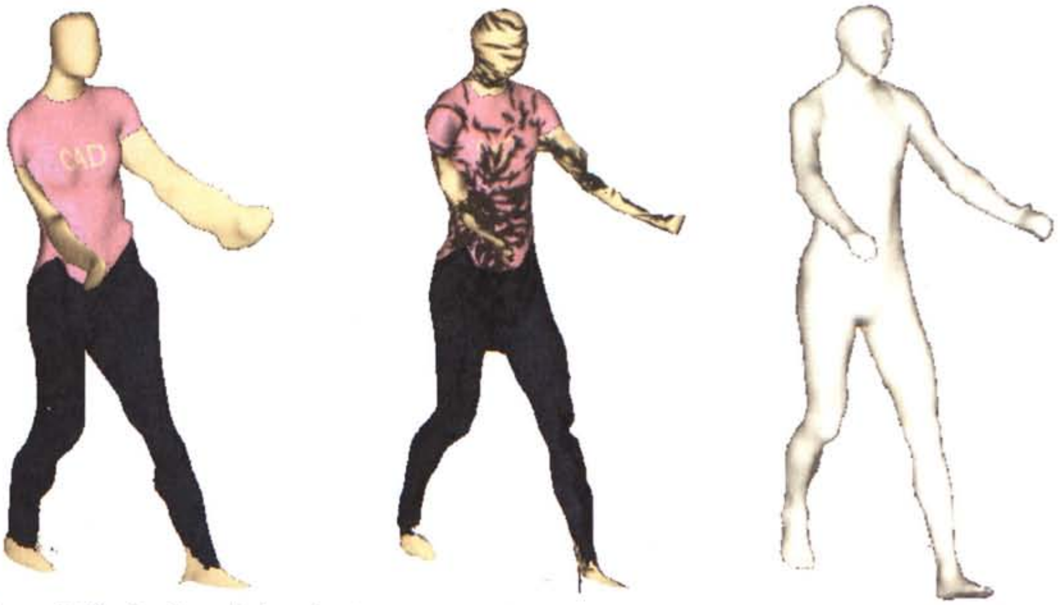

Figure 3-19: Surface fitting result (middle) of a posture aligned template (left) to input model (right) by generic approach. Smoothness distracted in black region, arm wrongly mapped to opposite side.

To solve the wrong mapping problem of the generic surface fitting approach, surface normal will be encountered in the proposed surface fitting optimization. First of all, a one-to-one mapping of each vertex of U' on V will be found by the kD-Tree technique. Each mapping of u will be further validated by checking two variable constraints. Firstly, the dot product of the angle of the normal vectors between two mapping vertices must be larger ε, where ε is set as 0.9 in the experiment. While ε will be decreased gradually in next iterations. Another constraint is the relaxation power r which controls the maximum allowing Euclidean distance between the mappings u and v. The value r is set as *0.1* initially and it will be increase by 10% in next iteration. After the validation of the mappings, vertex u can be adjusted by the distance between u and v by the following formula:

$$u = u + dn_u r \tag{3.22}$$

where n_u is the normal vector of u and d = *0.5* is a constant value to halve the changes of u along the normal direction.

Another outstanding control of the surface fitness is the introduction of the bi-directional mapping. For most matching approach, the mapping of the point is single direction, that is, just find the best mapping of u on v. In contrast, this thesis proposed a bi-directional mapping to also find the best mapping of v on u for the surface fitting optimization. The figure below illustrated the difference between single and bi-direction mapping in surface fitting.

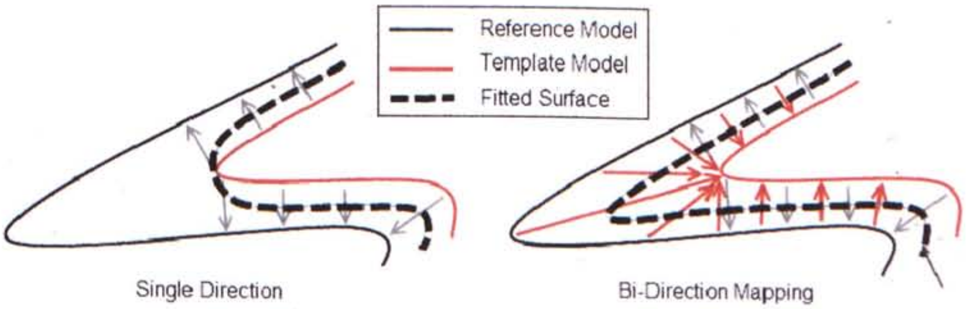

Figure 3-20: Surface fitting by single and bi-directional mapping of points. Fitted surface displayed in dashed line. Nearest mapping is displayed by the arrows from the source to target.

Obviously, the proposed bi-direction mapping has significant influence on tip points and this further improve the fitting quality of the surface. Figure 3-21 shows the difference between single and bi-direction mappings.

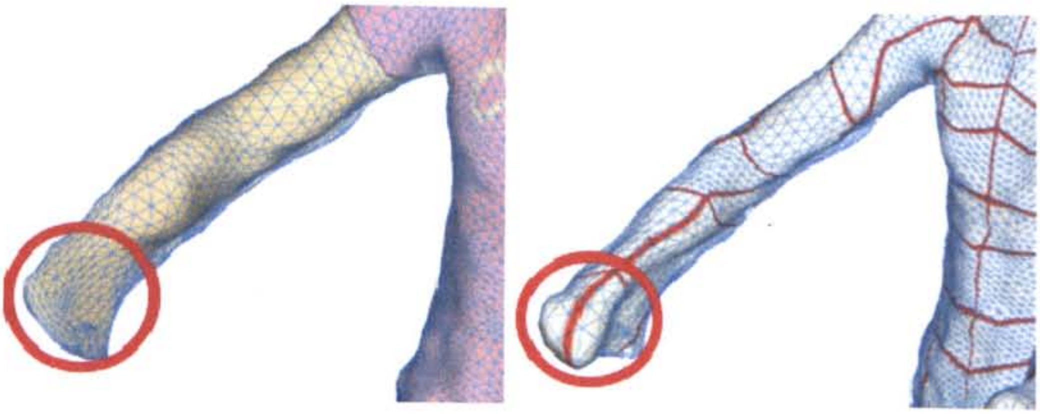

Figure 3-21: Surface fitting in single direction mapping (left) and bi-direction mapping (right)

3.2.2.2 Optimizing Surface Smoothness

Another optimization criterion of surface fitting is the smoothness. As the number of iteration increase, surface smoothness will be distracted if running fitness process alone. It is essential to interpolate the changes of the fitting process iteratively to reduce any bias on local maxima or convex region. The implement of surface smoothness is carried out by applying interpolating the vertex coordinates with the Fast Marching on Triangular Domain (FMTD) technique. The pseudo code of the smoothing process is described below:

Algorithm 3.5: Triangular Surface Smoothing

```
for each vertex $u$ in $U$
    initialize vector $z = \{0\}$
        for each vertex $u_i$ in $U$ with geodesic distance to $u$ smaller than $r$
            update $z = z + u_i G(d)$
        end for
    update $u = z$
end for
```

The radius r is chosen as two times of the mean edge length of the mesh by default. Where the Gaussian function *G(d)* is defined as $\left[\frac{1}{\sqrt{2\pi r}}\right]^{-\frac{d^2}{2r^2}}$, and d is the geodesic distance of the vertex u_i within r.

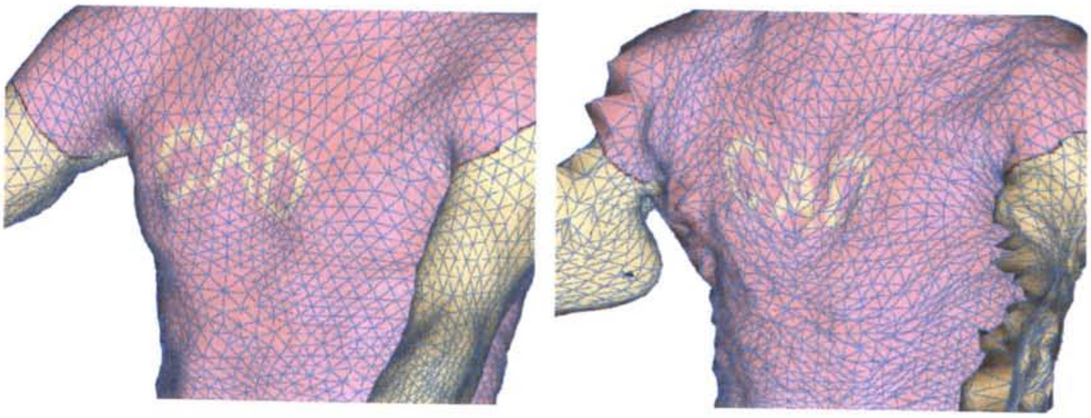

Figure 3-22: Surface fitting with (left) and without (right) smoothing optimization.

Figure 3-22 shows the significance and importance of the smoothing step in the fitting process. The implementation of the surface fitting step is described by the following pseudo code.

Algorithm 3.6: Surface Fitting Step

```
initialize $r = 0.5$, $a = 0.9$
repeat
    define the 1-to-1 mapping $f$ between $U$ and $V$
    for each vertex $u$ in $U$
        $v \leftarrow f(u)$
        if $n_u \cdot n_v > a$ then
            if $|u + dn_u r - v| < |u - dn_u r - v|$ then
                update $u = u + dn_u r$
            else
```

```
                update $u = u - dn_u r$
            end if
        end if
    end for
    decrease $a$
    increase $r$
    update $U$ by Smoothing
until $a < a_{min}$ and $r > r_{max}$
```

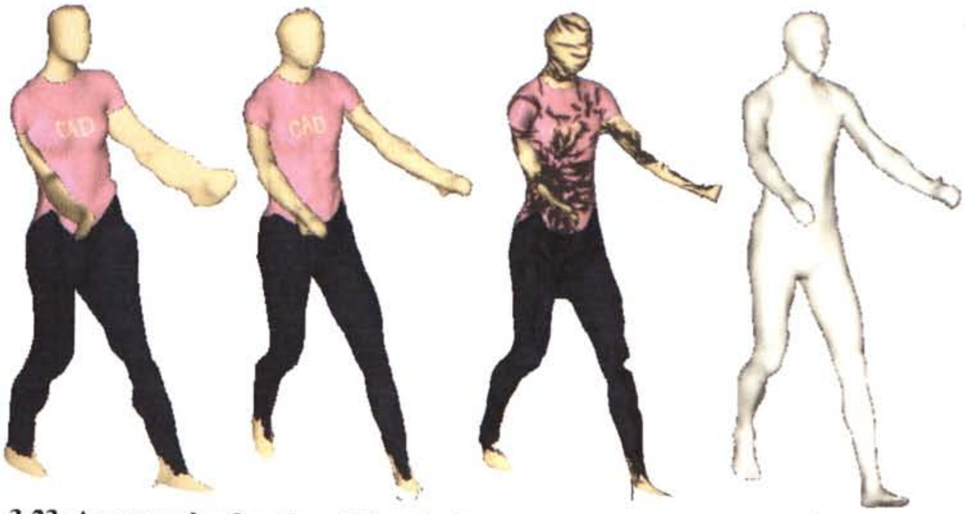

Figure 3-23: An example of surface fitting. Left: posture aligned template; 2nd column: Surface fitted template by the proposed surface fitting approach; 3rd column: Surface fitted template by simple vertex-facet fitting approach [WHT07]; Right: Input model

Efficiency, however, is one of the rooms for improvement in this refinement process. Since the surface fitting process required a one-to-one mapping for all vertices of the template and input model, and vice versa, it is not feasible to apply the proposed technology in real time computation in current stage. Some other latest technologies such as the Graphics Processing Unit (GPU) computation may improve the performance issue and further works will be done on this area in the future.

3.2.3 Feature Matching Refinement

The last step of the matching algorithm is named Feature Matching Refinement. The purpose of this section mainly improves the quality of the anchor point matching by optimizing the surface of the refined template model *U*. The technical details are discussed below.

Feature-to-feature matching has been the major objective in shape matching approaches **[JH02][PDJ93][LDRS05][JD03]**. However, the feature characteristics of an object may not be the primary consideration in different approaches due to their own nature and field of applications. For example, the iterative alignment scheme **[CR03]** and bending invariant shape in **[EK03]** optimizes the global surface smoothness instead of local feature mappings. On the other hand, some previous researches have integrated the feature matching refinement process in their proposed non-feature matching approach, such as **[JZ06][CJGQ05]**. Nevertheless, the application of these approaches has always been restricted by many limitations such as the requirement of a very good initial guess. Hence they cannot be served as a general solution for 3D shape matching applications.

In this thesis, a feature matching refinement process will be proposed which increase the sensitivity of the anchor points mapping to the features between the template and input models. The process included two parts: a feature descriptor construction and a feature descriptor matching. These parts will be described in detail in the following.

To increase the sensitivity of the mappings to local features, the feature points on template model *U* should be identified. Various advanced feature identification algorithm can be found in shape matching researches.

Jiao and Heath proposed a feature detection algorithm in **[JH02]** to identify the geometric features, such as ridges and corner, on the mesh surface. Their basic idea is to identify strong and valid curves on the surface, and then subdivided these curves at strong vertices. Hence the features of the surface can be obtained by grouping these sub-curves at invalid vertices, according to their ranking. The ranking of a vertex is defined according to their number of incident feature edges. Obviously, this feature identification algorithm required a very clear definition of edge and flat plane on the mesh and hence it is applicable for regular 3D shapes, for example the mechanical parts.

Another famous feature extract approach is Laplacian Operator. This approach has been widely adopted in 2-D image feature detection algorithm, such as the Harris Corner Detection **[LD04]**. The main idea of this algorithm is to smooth the mesh with a Gaussian operator and then subtract that smoothed surface by its original model. The feature point is determined by filtering the largest values on the Laplacian surface with a variable threshold value. In the proposed feature matching algorithm, the feature points on the template will be identified by this approach according to the several reasons. Firstly, this implementation of this approach is simple and hence the efficiency is also high. Secondly, the application of this technique regardless of the mesh natures (i.e. regular/ freeform mesh) and therefore it is suitable for the proposed algorithm. The mesh with Laplacian Operator L is defined by

$$V'' = L(V) = V - S(V)$$

where V is the input mesh and S is the Gaussian Smoothing Operator as described in the surface refinement process.

3.2.3.1 Feature descriptor

In 2D computer vision and 3D object matching, feature descriptor has always been the most popular and effective measuring unit. Based on the application of the matching algorithm, feature descriptor with different nature should be applied. A very successful example of 2D image descriptor is the SIFT descriptor **[LD04]**. The SIFT descriptor is encoded by capturing the properties like coordinates, gradient and intensity orientation…etc, on each vertex of the image. And hence the encoded information can be served as a unique signature for a robust matching approach. Practically, it is very hard to define a unique feature descriptor in 3D space due to the variations of mesh topology and deformation levels.

In this thesis, a 3D feature descriptor is proposed, which is invariant to the differences of scale, orientation and topologies between the template model U and input model V. Where the feature descriptor of a vertex v is a synthesis of the following components:

- Relative coordinates of the vertices around v within radius R to origin of F
- Normalized Gaussian curvature C_i of all vertices v_i around v within radius R

The construction of the feature descriptor and its components will be explained below.

Given a vertex v on a triangular 3D surface Π and a vertex set $\{V_R\}$ that consists of the points around v within radius R. The feature descriptor $F(v)$ of vertex v will be constructed by the following steps.

The first component is an axis frame F with three orthogonal unit vectors t_1, t_2 and t_3 will be defined at vertex v as shown in the figure below:

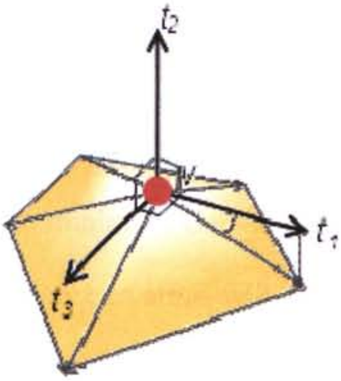

Figure 3-24: Axis frame F of the vertex v on the triangular surface

v is acted as the origin and the normal vector of v defined t_2. The first vector t_1 is defined as the projection of the edge e that connecting to the vertex v and perpendicular to t_2. And e is selected arbitrarily. The final vector t_3 is then obtained by the cross product of t_1 and t_2. Hence the function $T(v)$ that transform the vertex v from the space between $\{-R, R\}$ to $\{-1, 1\}$ in R^3 can be obtained by the RBF function with three control point pairs: $\{v+t_1R, 1\}$, $\{v+t_2R, 1\}$ and $\{v+t_3R, 1\}$.

Now the relative coordinate of each vertex v_r in $\{V_R\}$ can be embedded in the feature descriptor by applying the function $T(v_r)$. Afterwards, the curvature information of the vertices will be encoded into the descriptor by incorporating the curvature estimation technique introduced in the fundamental technique section.

The second component of the feature descriptor is the normalized Gaussian curvature. Recalling to the curvature estimation technique section, the Gaussian curvature is defined as the product of two principal curvatures k_{min} and k_{max}. These two curvatures, however is a scalar value relative to the maximum and minimum curvature of the whole mesh. i.e., a mesh with one extreme tip point will dominate a wide range of

k_{max} value of the whole mesh, as a result all other features of the mesh will be regarded as insignificant comparably. Hence, in order to provide consistency curvature information between similar surfaces, the feature descriptor embeds the normalized Gaussian curvature of the local vertex set $\{V_R\}$. Where the normalized curvature is defined by rescaling of the Gaussian curvature of v and the vertices in $\{V_R\}$, such that the vertex with minimum Gaussian curvature value will be -1 and the vertex with maximum Gaussian curvature will be equal to 1. Figure 3-25 below shows an example of the feature descriptor of a local concave region in radius R. Based on the experimental tests, selecting R as ten times of the average edge length on U will be a good trade-off between robustness and speed.

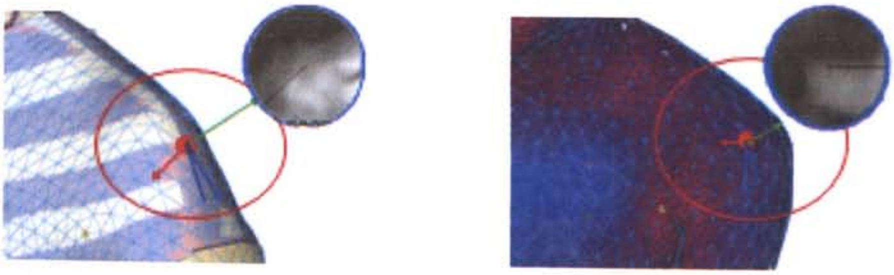

Figure 3-25: Construction of the feature descriptor: normal (green), x-axis (red), cross-product (blue). Curvature gradient map generated within red circle. Illustrated in grayscale.

3.2.3.3 Feature Descriptor matching

Once the feature descriptor scheme has been developed and the feature points on the template model has been defined, the surface of the template model U' will be refined iteratively by realigning the feature mapping between the template model U and input model V once at a time.

For simplicity, the feature mapping algorithm will focus on a single vertex a^u in U in a single iteration. The correspondence a^v of a^u must be found on the surface of the input human model V such that the feature descriptor cost

$$C_F(a^u, a^v) = \left|F(a^u) - F(a^v)\right| \qquad (3.23)$$

is minimized as

$$a^v = \arg\min C_F(a^u, a^v) \qquad (3.24)$$

.Where *F(a*^{u}*)* and *F(a*^{v}*)* are the feature descriptors of vertex a^u and a^v respectively.

The search of an optimal a^v starts from $a^v = f_{S_V}(a^u)$. A search window with radius *r* is established to include all surface points (sampled) on *V* with distance to a^v less than *r*. Then, the minimal feature descriptor cost *C* between a^u and all these surface points can be found by an exhaustive search. Note that, during the search, the local frames on the surface samples are rotated to find a best match as the axis *X* of a local frame is arbitrary on the tangent plane of the surface point..

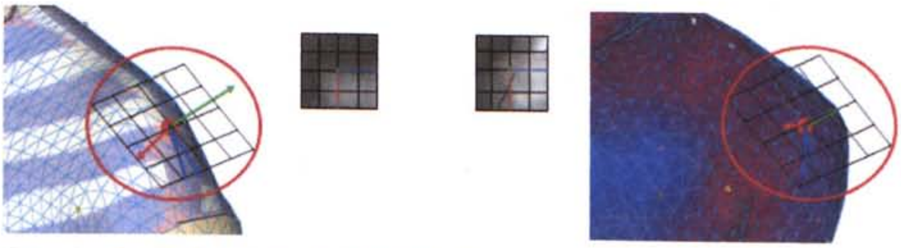

Figure 3-26: The upper row illustrated the region covered by the feature descriptor centered from an anchor point. The bottom row illustrated the feature descriptor of the anchor point. Blue region indicated the searching region and best descriptor matching of the template.

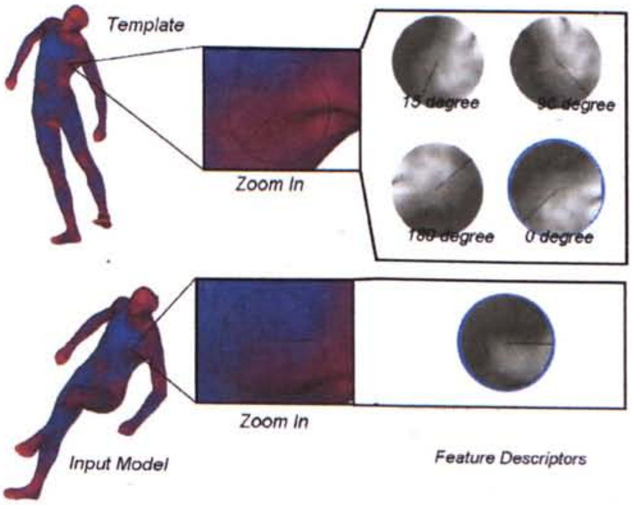

Figure 3-27: Visualization of the feature descriptors on a particular region of a vertex on two matching objects. The top row displayed the feature descriptor at different orientation.

The feature matching process is described below:

Algorithm 3.7: Feature Matching Refinement

```
Find the 1-to-1 mapping $f$ for the anchor points of $U$ on $V$
Initialize $k$, $R$, $C_{min} = \infty$
for each anchor point $a^{u}$ in $U$
    if $K(a) > M$ then
        $a^{v} = f(a^{u})$
        $a^{v}_{best} = a^{v}$
        construct the feature descriptor $F_a$ for $a^{u}$
        for each vertex $a^{v}_{i}$ with $g(a^{v}_{i}, a^{v}) < R$ in $V$
            construct the feature descriptor $F_i$ for $a^{v}_{i}$
            for each angle $d$ between $[-\varepsilon, \varepsilon]$
                update $C = | F_a d - F_i |$
                if $C < C_{min}$ then
                    $C_{min} = C$
                    $a^{v}_{best} = a^{v}_{i}$
                end if
            end for
        end for
        update $f(a^{u}) = a^{v}_{best}$
    end if
end for
```

Chapter 4 Experimental Result

In the following section, the experimental results of the proposed correspondence matching algorithm in this thesis will be illustrated. These results will be organized into four parts:

- The results of the fundamental techniques with different parameter settings
- The matching result in each matching step
- The comparison of the proposed algorithm to other similar approaches
- The statistical figures of the effectiveness, efficiency on different experiment samples

The proposed algorithm is implemented in a prototype program by Visual C++ with OpenGL library for 3D visualization of models. The experimental tests are carried on a PC with Intel Core i5 430 CPU (2.27GHz) plus 4GB main memory running 64bits MS Windows 7. Basically the computation of all examples can be completed in less than one minutes.

4.1 Result of the Fundamental Techniques

This part illustrates the results of five fundamental techniques incorporated in the proposed algorithm. They include the geodesic distance approximation, Farthest Point Sampling (FPS), Radial Basis Function (RBF), curvature estimation and Multi Dimensional Scaling (MDS).

Before evaluating the result of the proposed matching algorithm, the choosing of the parameters in the fundamental technique must be determined. Hence, this section illustrates the output of those techniques with different parameters settings, such as sample rate, warping regularity and region sizes…etc, help to analyze the influence of this settings affect the matching process.

4.1.1 Geodesic Distance Approximation

The first technique to be evaluated is the geodesic distance approximation. In this research, the geodesic distance computed from a given source point will be illustrated on the surface by grey color gradient in increasing order, i.e. the source point in white and farthest point in black. According to the experiment, the two testing method Dijkstra algorithm and Wave Front Propagation Method generated a very close result for all testing samples that with about 10,000 vertices. Hence the Dijkstra Algorithm is regarded as the primary choice because of the simplicity consideration (Implementation refer to the Fundamental Technique Section). The visual illustrations of the testing samples can be found in Figure 4-1.

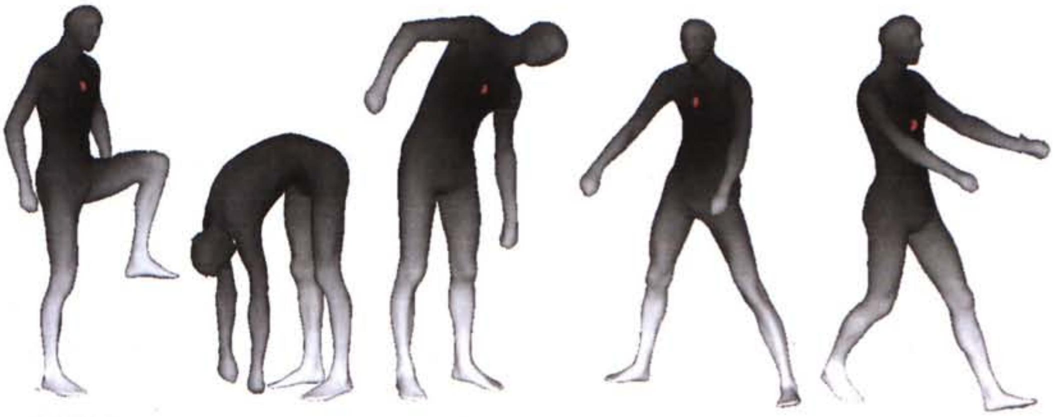

Figure 4-1: Geodesic Distance examples of human bodies in different postures. (Source point in RED, higher intensity for larger geodesic distance from the source)

4.1.2 Farthest Point Sampling (FPS)

The only parameter for the Farthest Point Sampling is the sample rate. The experiment evaluates the performance of the Farthest Point Sampling and the quality of the sample points by applying different sample rate, i.e., 10, 100, 500 and 1000, on it. The figure below illustrates the distribution of the sample points on different examples.

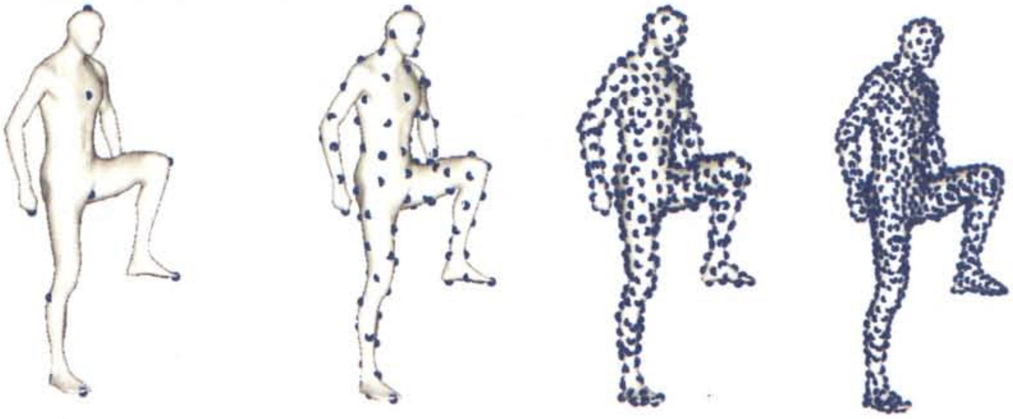

Figure 4-2: Sample points defined by the Farthest Point Sampling on example 1. Number of sample points = 10, 100, 200 and 500 from left to right.

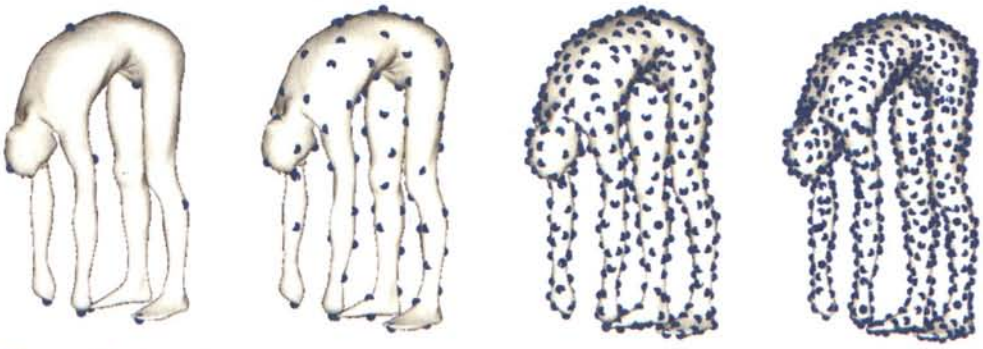

Figure 4-3: Sample points defined by the Farthest Point Sampling on example 2. Number of sample points = 10, 100, 200 and 500 from left to right.

According to the experiment, the average computation time of the Farthest Point Sampling on different sample rate is always below one second. Therefore the consideration of the sample rate is not the major concern in the Farthest Point Sampling process itself. On the other hand, the sample rate is constrained by the performance of the warping function, the Radial Basis Function as explained below.

4.1.3 Radial Basis Function (RBF)

The result of the Radial Basis Function is controlled by three main parameters: The control points, regularization parameter lambda and the basis function *g(r)*. Experimentally, the control points are generated by the Farthest Point Sampling and hence the computation time of the RBF mainly controlled by the sample rate. The second parameter lambda controls the warping smoothness of the surface, which is usually set between 0 and 1. The transition output with different lambda can be found in Figure 4-4. The basis function *g* is chosen as $g(r) = |r|$.

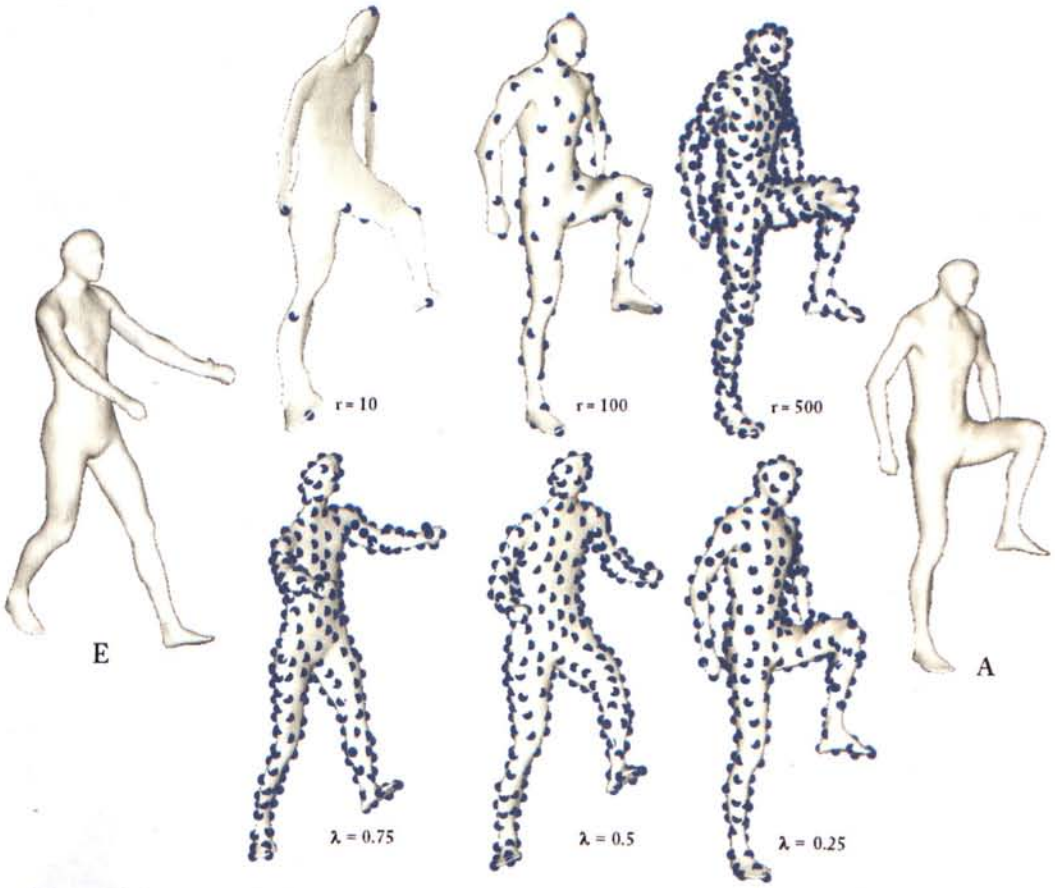

Figure 4-4: Warping result of example E (Left) to example A (Right-most) by Radial Basis Function in different parameterization. (Upper row: Fixed $\lambda = 0$ and Number of sample points = 10, 100 and 200; Bottom row: Fixed Sample Rate = 200 and $\lambda = 0.75$, 0.5 and 0.25)

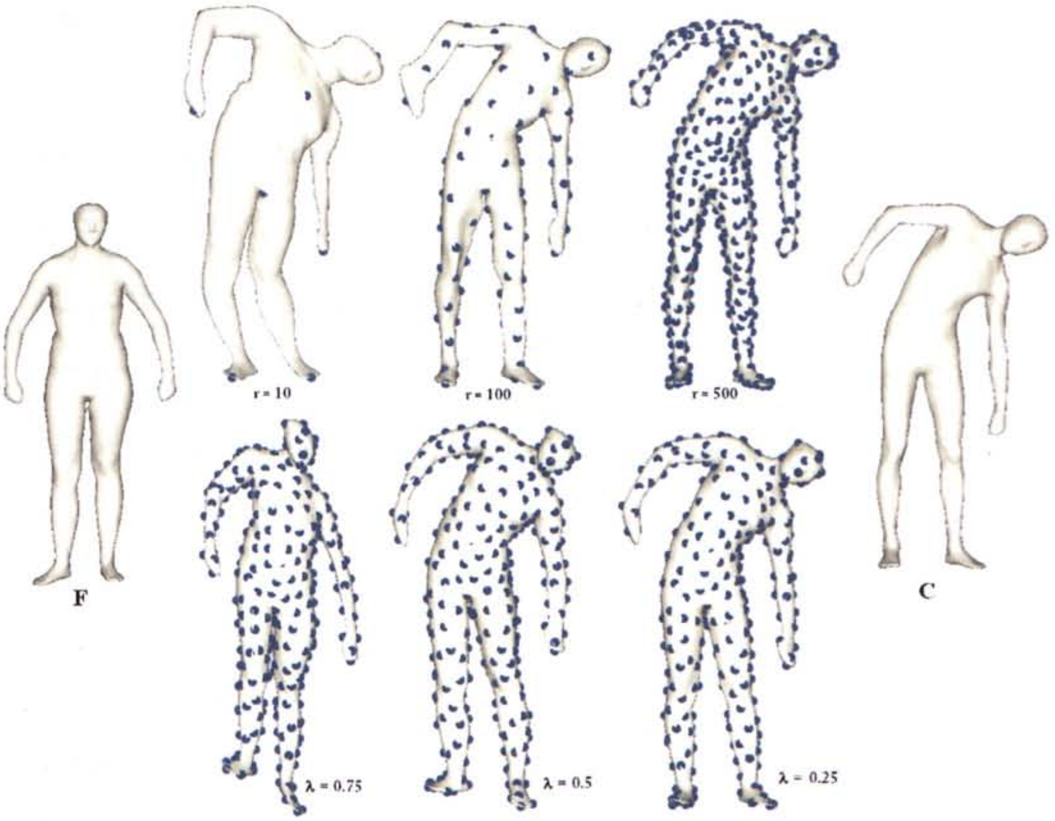

Figure 4-5: Warping result of sample F (Left) to sample C (Right-most) by Radial Basis Function in different parameterization. (Upper row: Fixed $\lambda = 0$ and number of sample points = 10, 100 and 200; Bottom row: Fixed Sample Rate = 200 and $\lambda = 0.75$, 0.5 and 0.25)

4.1.4 Curvature Estimation

In the experiment, the principle curvatures on the continuous surface of 3D models are presented by the Red (high) and Blue (low) color channels. Some examples can be found in the found in the Figure 4-6 below.

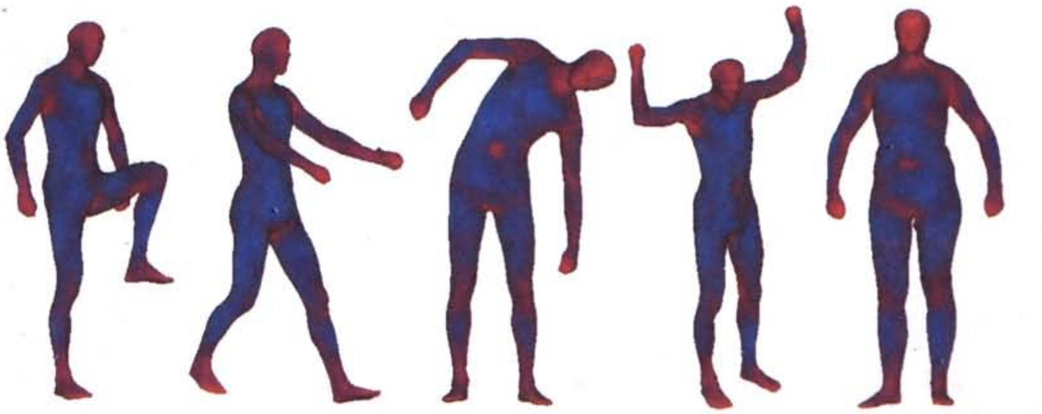

Figure 4-6: Curvature distribution of different human body examples. Blue for low Mean

Curvatures and Red for high Mean curvatures.

According to the experiments, this curvature estimation method is reliable and accuracy working on triangular domains. The major concern of this technique is the performance in practical use, where it can be controlled by adjusting the window size radius of the processing vertex. For simplicity, the bounding radius of the curvature tensor is set as the 1/100 of the bounding box size by default. And hence the computation time can be controlled within 10 seconds for all testing samples in the experiments.

4.1.5 Multi-Dimensional Scaling (MDS)

The Multi Dimensional Scaling is one of the most important techniques in the proposed matching algorithm. In this thesis, two different approaches of MDS have been evaluated; the Classical MDS and Fast MDS.

For both approaches, the dimension k of the MDS embedding result should have higher ranking than its original domain, i.e. $k > 3$, also suggested in **[JZ06]**. Therefore, the dimensional k of the MDS embedding is set to 6 in the experiment, where the 4th, 5th and 6th dimensional are presented in the RED, GREEN and BLUE channels respectively. Figure 4-7 illustrated some examples of the MDS transformation result with different parameter settings (Classical/ Fast MDS, sample rate, basis function).

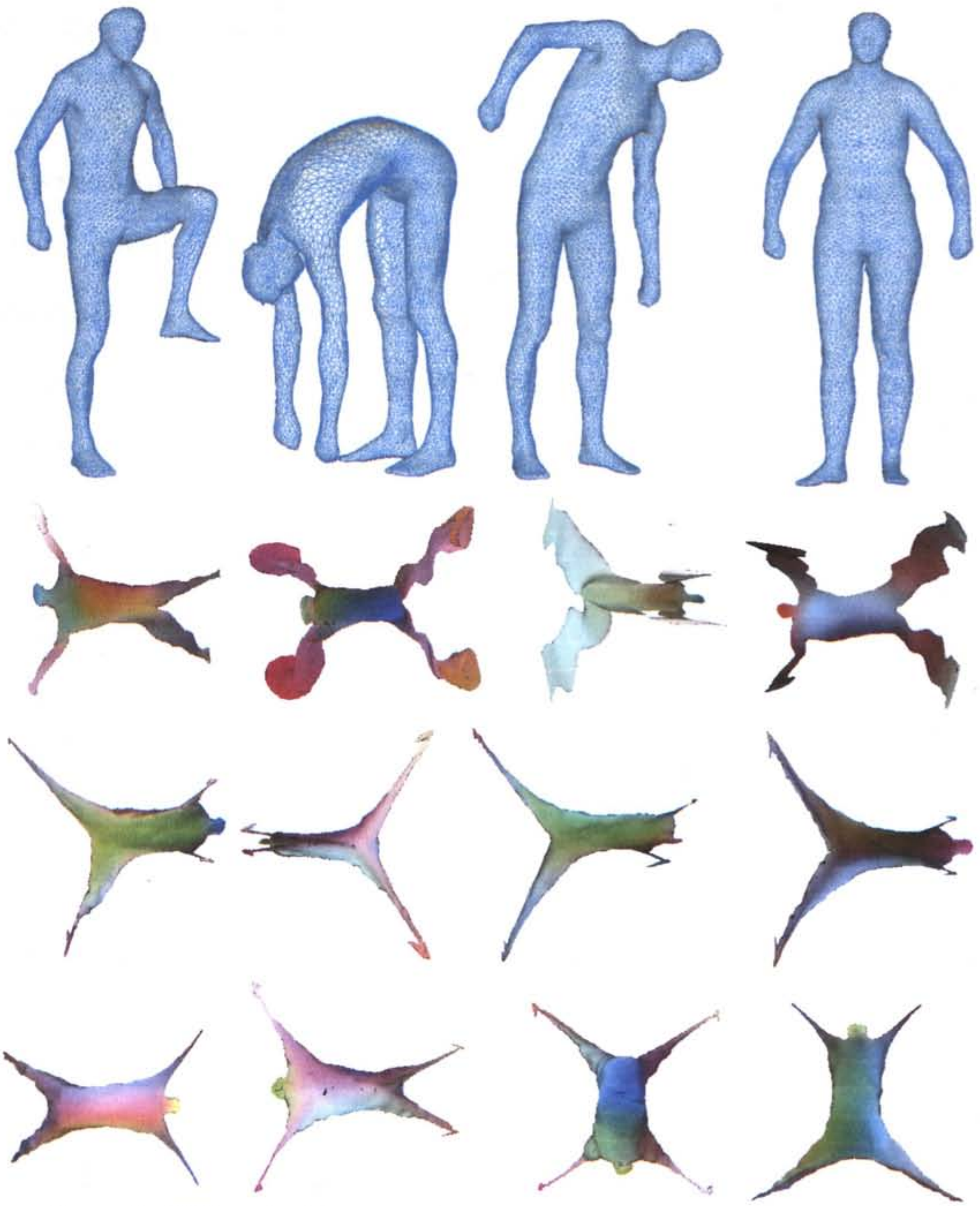

Figure 4-7: MDS result of human body with different parameterization. Top row: Original human body meshes; 2nd row: Classical MDS with 50 sample points; 3rd row: Classical MDS with 500 sample points; Bottom row: Fast MDS with 500 sample points;

4.2 Result of the Core Matching Processes

In this section, the mapping results at different core matching processes will be illustrated. The sample meshes are presented as triangular meshes with about 8,000 – 10,000 vertices, and build up in winged-edge topology. Moreover, in order to show the matching ability against deformable (or bending invariant) shapes, the chosen samples are all human body that in different postures. Figure 4-8 below illustrated the samples of human bodies

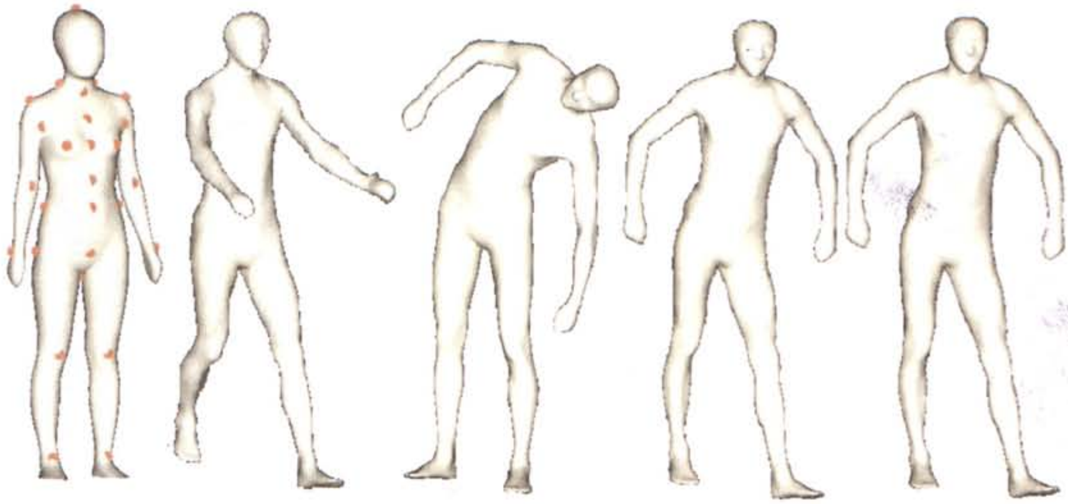

Figure 4-8: Testing examples of human bodies in differsent postures. (Left: Template model, others: Input models)

4.2.1 Posture Alignment Step

This part illustrated the first alignment step of the input model in the matching process. The quality of this step can be analyzed by evaluating three important transition result of the input model, they are the sign flip corrected MDS embedding V_{MDS}, the transformed MDS embedding towards template's MDS and the instant mapping of the correspondences on the input model by U_{MDS} and V_{MDS}.

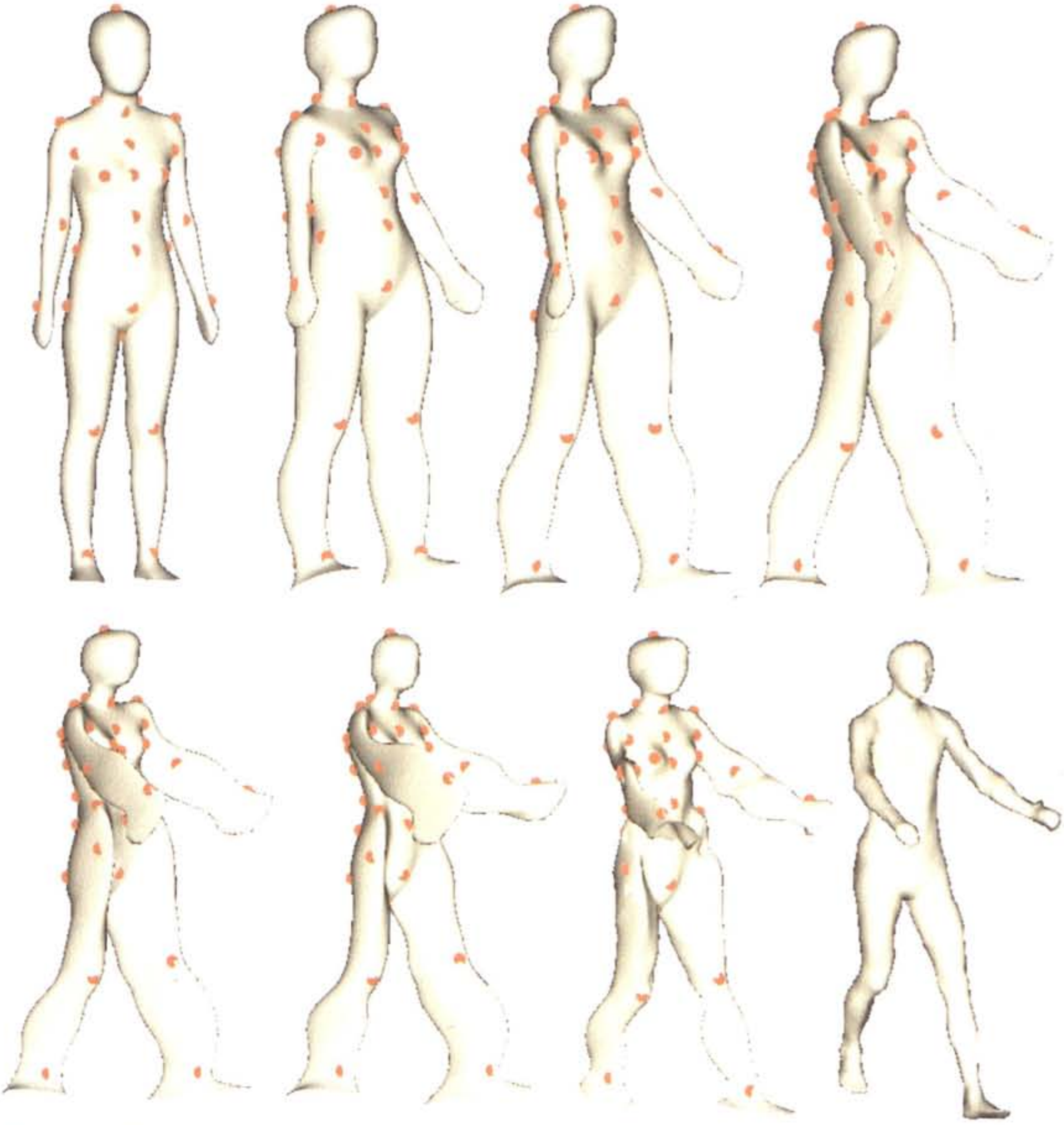

Figure 4-9: Posture Alignment example 1 – Transitions of template model (Top-Left) to input model (Bottom-Right) in posture alignment step.

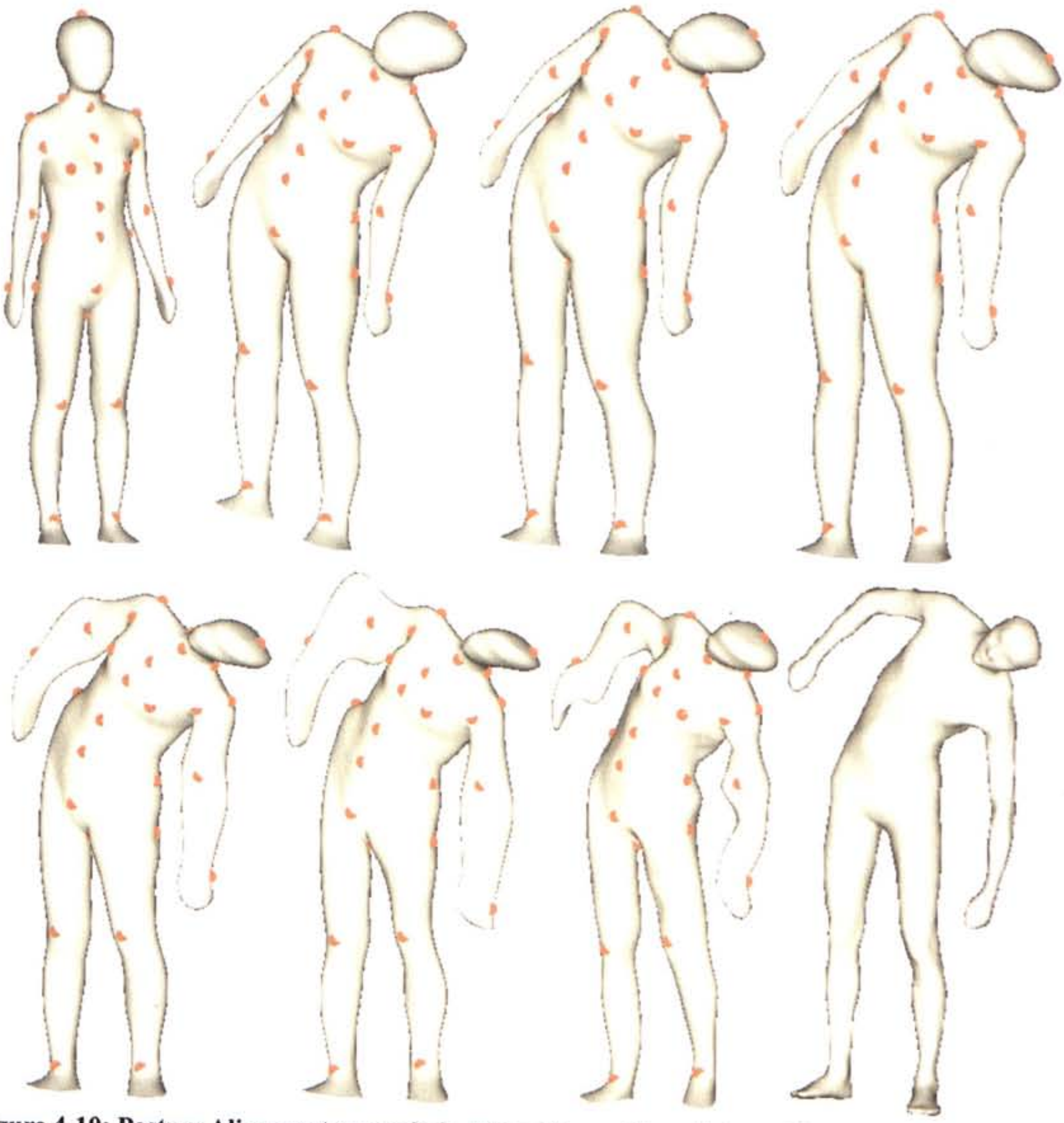

Figure 4-10: Posture Alignment example 2 – Transitions of template model (Top-Left) to input model (Bottom-Right) in posture alignment step.

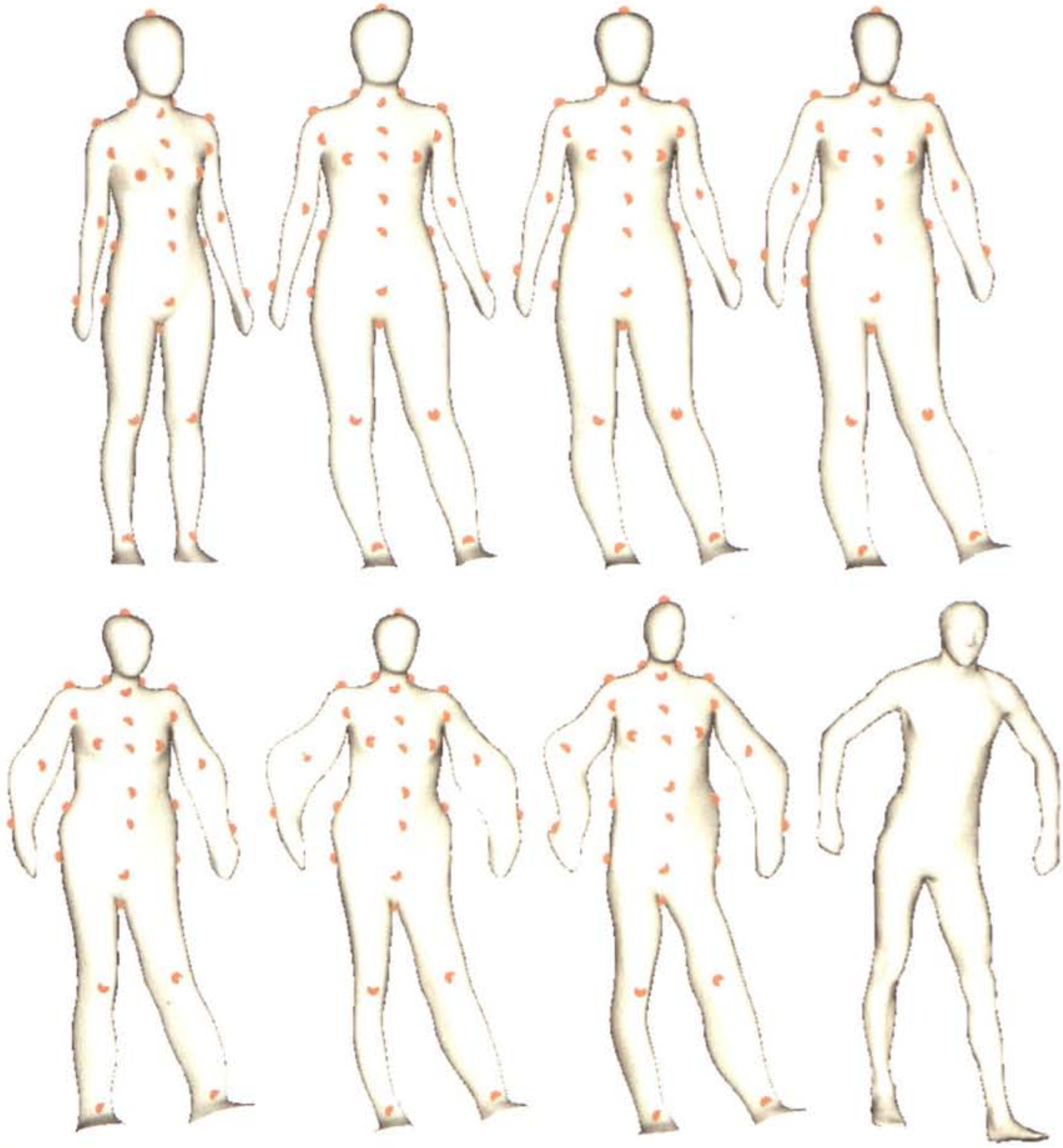

Figure 4-11: Posture Alignment example 3 – Transitions of template model (Top-Left) to input model (Bottom-Right) in posture alignment step.

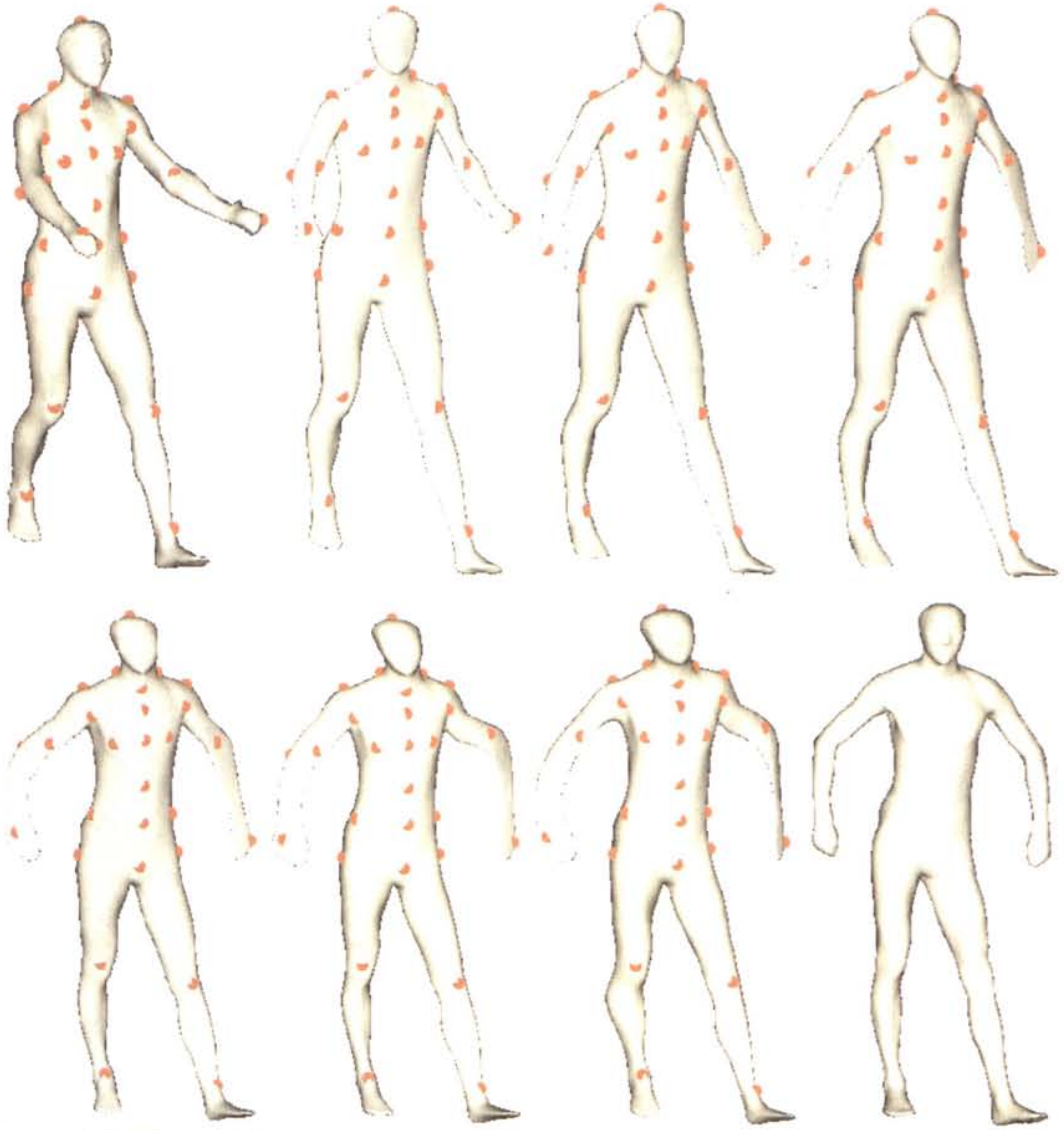

Figure 4-12: Posture Alignment example 4 – Transitions of template model (Top-Left) to input model (Bottom-Right) in posture alignment step.

4.2.2 Surface Fitting Step

The surface fitting result based on the best posture aligned template from the previous step.

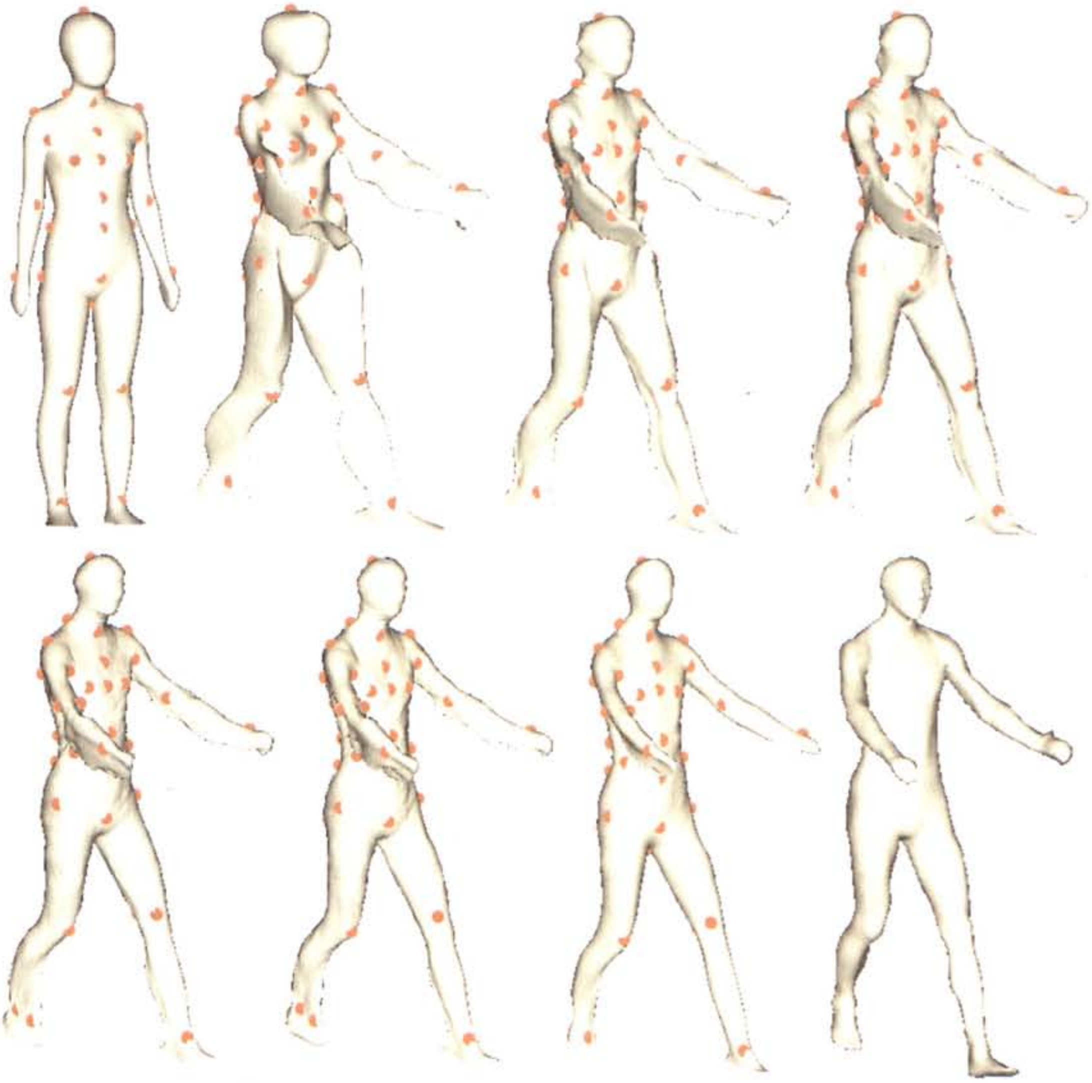

Figure 4-13: Surface Fitting example 1 – Transitions of template model (Top-Left) to input model (Bottom-Right) in surface fitting step.

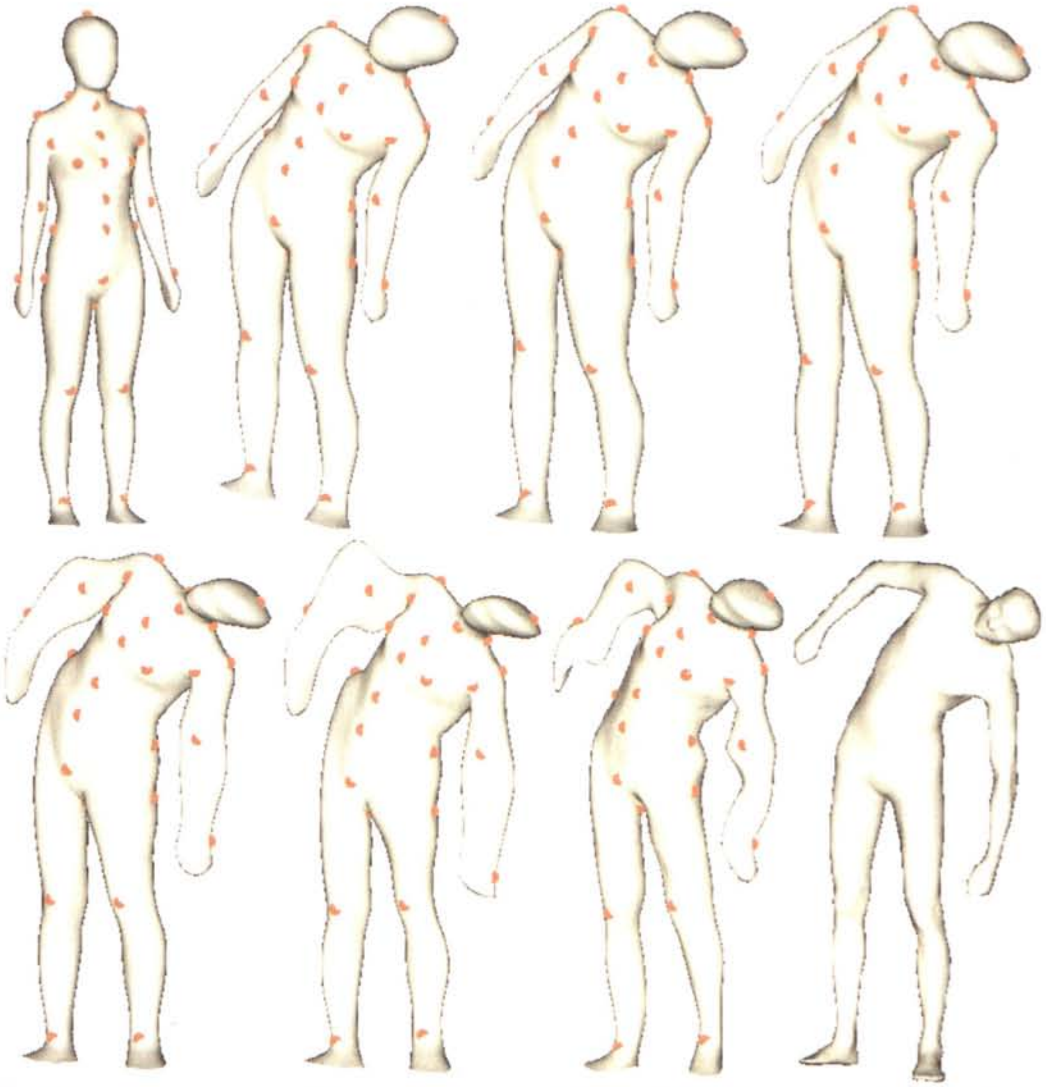

Figure 4-14: Surface Fitting example 2 – Transitions of template model (Top-Left) to input model (Bottom-Right) in surface fitting step.

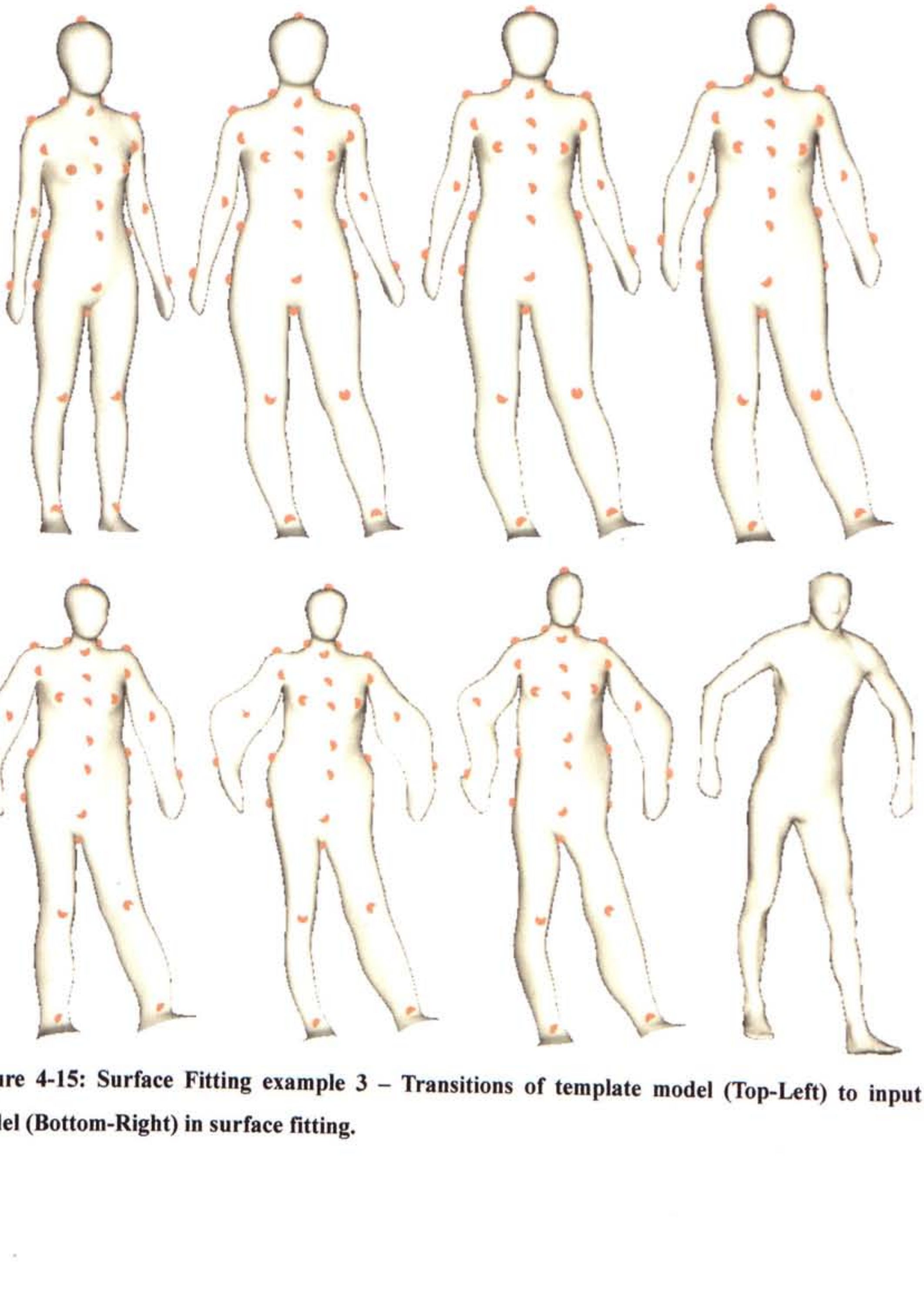

Figure 4-15: Surface Fitting example 3 – Transitions of template model (Top-Left) to input model (Bottom-Right) in surface fitting.

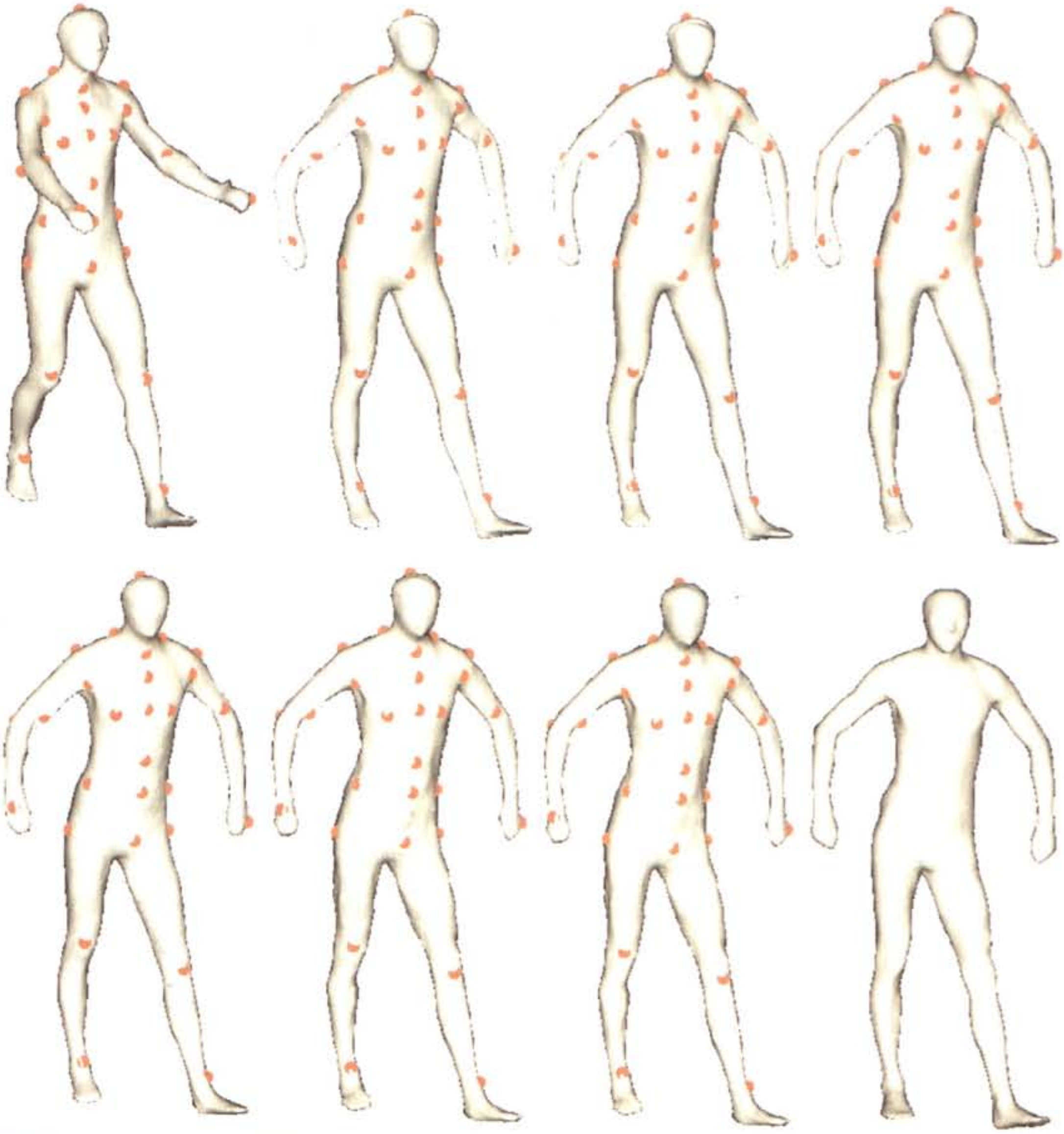

Figure 4-16: Surface Fitting example 4 – Transitions of template model (Top-Left) to input model (Bottom-Right) in surface fitting.

4.2.3 Feature Matching Refinement

The feature refinement step based on the best surface fitted template from the previous step.

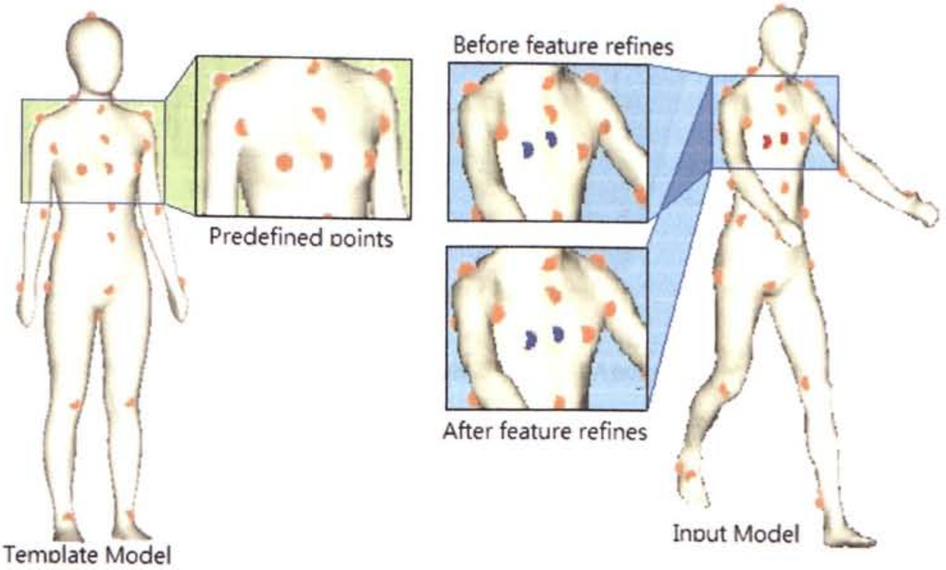

Figure 4-17: Feature Matching Refinement example 1: The template model with anchor points predefined on the left; matching result of the input model before (upper zoom window) and after (bottom zoom window) the refinement is displayed on the right. Vertices refined my the feature matching is circled in red thin line.

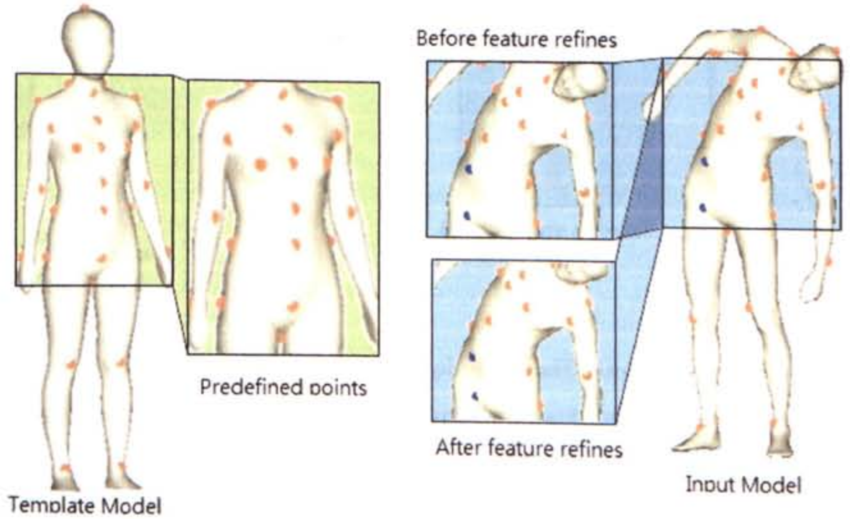

Figure 4-18: Feature Matching Refinement Sample 2 – (Left) Anchor points defined on template; (2nd column) Anchor Points on template with surface transformed after posture alignment and surface fitting; (3rd column) Anchor point mappings on input model of the 2nd column's model; (Right): Refined anchor points mapping on local surface of input model with feature descriptor.

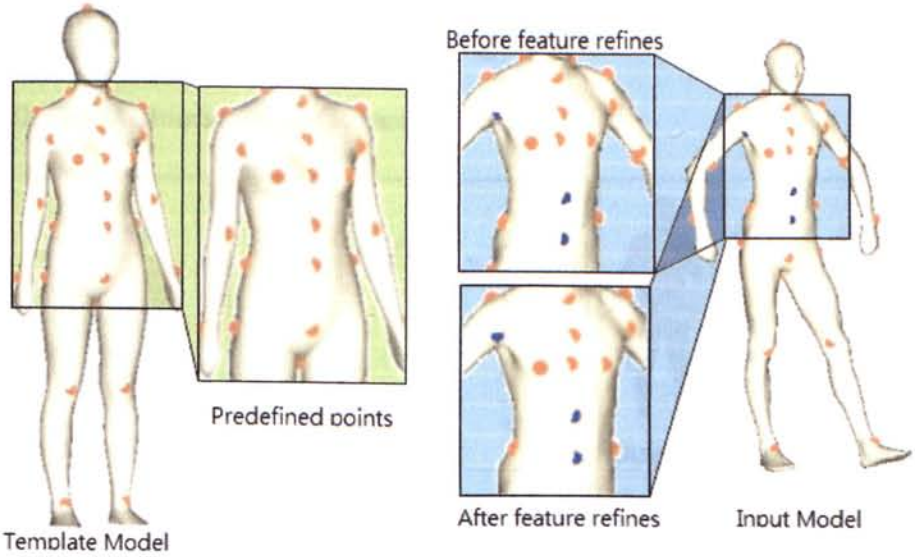

Figure 4-19: Feature Matching Refinement Sample 3 – (Left) Anchor points defined on template; (2nd column) Anchor Points on template with surface transformed after posture alignment and surface fitting; (3rd column) Anchor point mappings on input model of the 2nd column's model; (Right): Refined anchor points mapping on local surface of input model with feature descriptor.

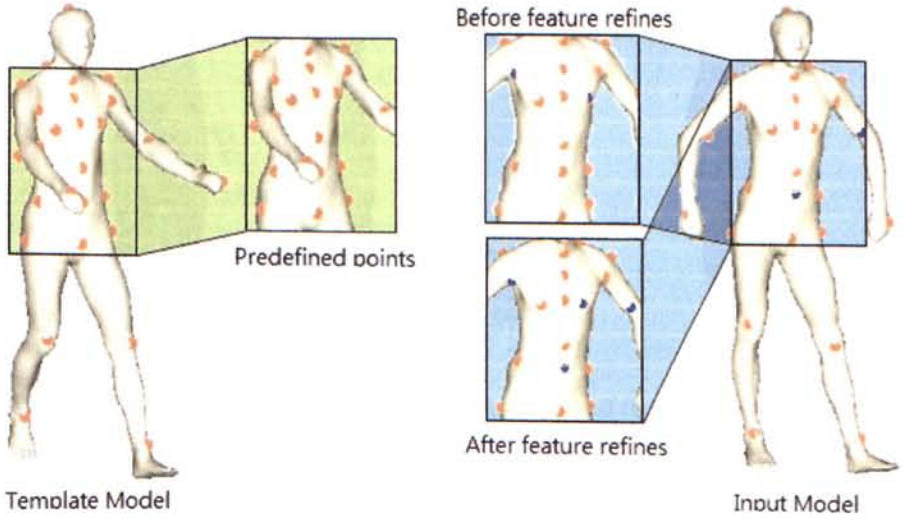

Figure 4-20: Feature Matching Refinement Sample 4 – (Left) Anchor points defined on template; (2nd column) Anchor Points on template with surface transformed after posture alignment and surface fitting; (3rd column) Anchor point mappings on input model of the 2nd column's model; (Right): Refined anchor points mapping on local surface of input model with feature descriptor.

4.2.4 Application of the proposed algorithm

4.2.4.1 Design Automation in Garment Industry

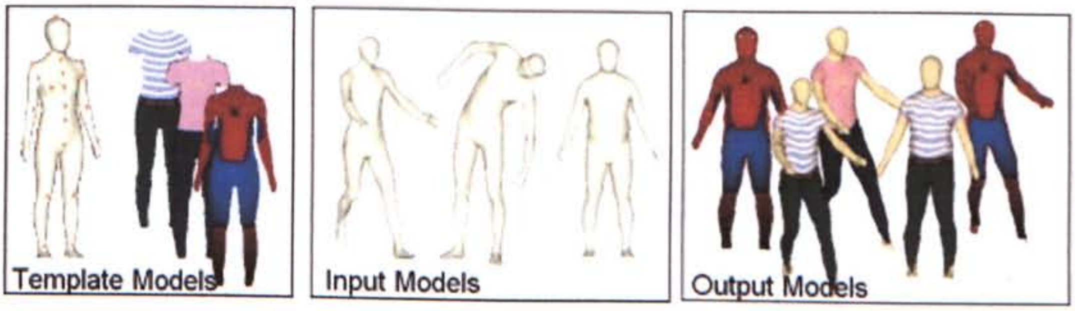

Figure 4-21: Input models with feature and garment designed (left); Input models in different postures, mesh structures or body appearances (e.g. thin or fat) (middle); The garment design fitted on the input models from the template (right).

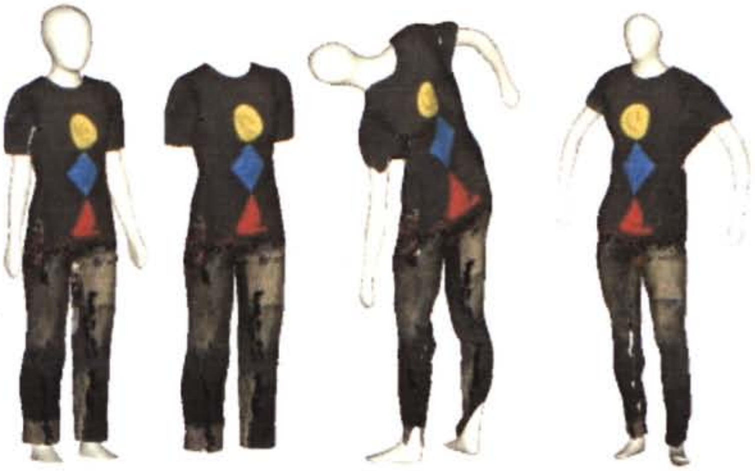

Figure 4-22: Input models with feature and garment designed (left); Input models in different postures, mesh structures or body appearances (e.g. thin or fat) (middle); The garment design fitted on the input models from the template (right).

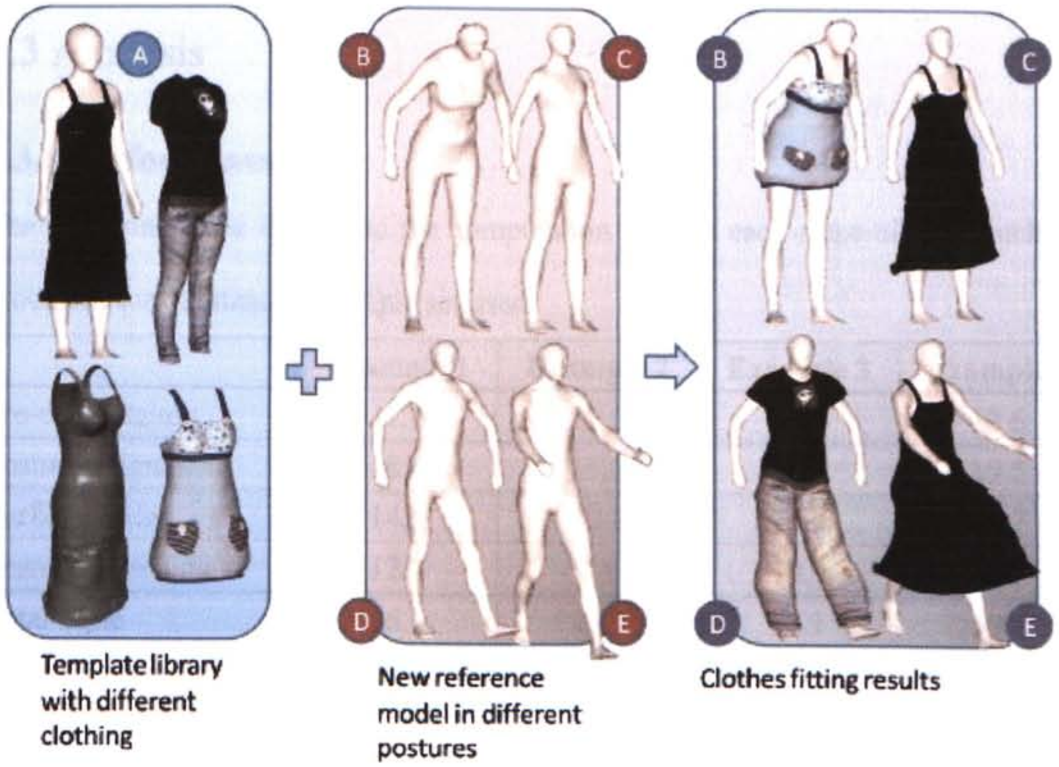
A
B
C
D
E
B
C
D
E
Template library with different clothing
New reference model in different postures
Clothes fitting results

Figure 4-23: More examples of design automation of clothes on the human models with semantic features extracted by the proposed algorithm.

The application of the proposed technology can be applied in garment industry for clothes fitting as the figure illustrated above. The technology can be further extended to more finite garment design fields such as gloves and swimsuit…etc. More other application such as computational effect (morphing, animation, 3D object panorama..etc) are under evaluation and more practical applications will be released in the future.

4.3 Analysis

4.3.1 Performance

The following table illustrated the computation time in each steps of the matching processes for the standard sample set used.

	Example 1	Example 2	Example 3	Example 4
Pre-computaiton	11.7	12.5	13.3	12.6
Posture Alignment	18.6	15.2	16.1	19.5
Surface Fitting	14.5	12.6	16.8	17.7
Feature Matching	12.1	9.6	10.9	9.9
Total Time	56.9	49.9	57.1	59.7

Table 4-1: Computation Time at different step of matching processes

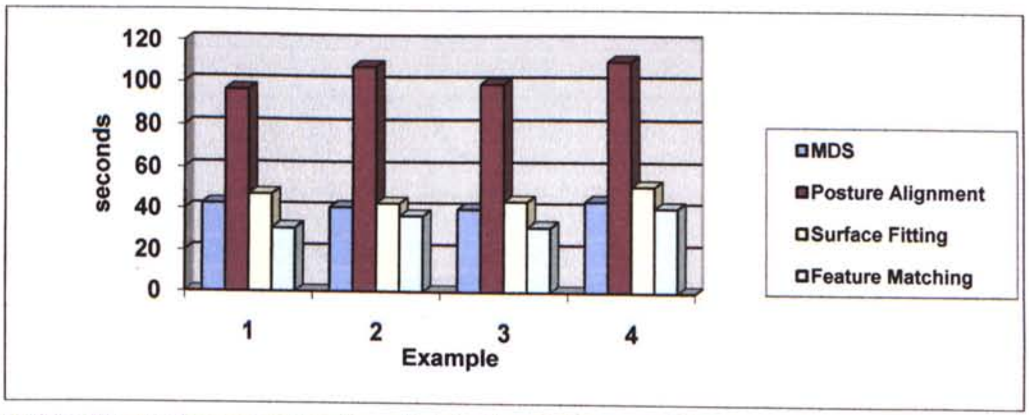

Table 4-2: Chart of computation time distribution in the matching process

The performance of the matching process highly depends on the posture alignment step. A trade off can be made by increase the reduction rate of the regularization parameter in each iteration. Hence the number of iteration in posture alignment process can be reduced.

4.3.2 Accuracy

The following table illustrated the distance error between the transformed template *U'* and the input model *V* at different matching stage.

	Example 1	Example 2	Example 3	Example 4
Initial	442.34	217.5	398.73	526.17
Sign Flip Correction	57.2	40.9	40.88	40.77*
Posture Alignment	28.74	21.25	28.77	23.52
Surface Fitting	4.11	7.56	9.54	10.17
Feature Matching	3.59	5.25	6.45	7.91

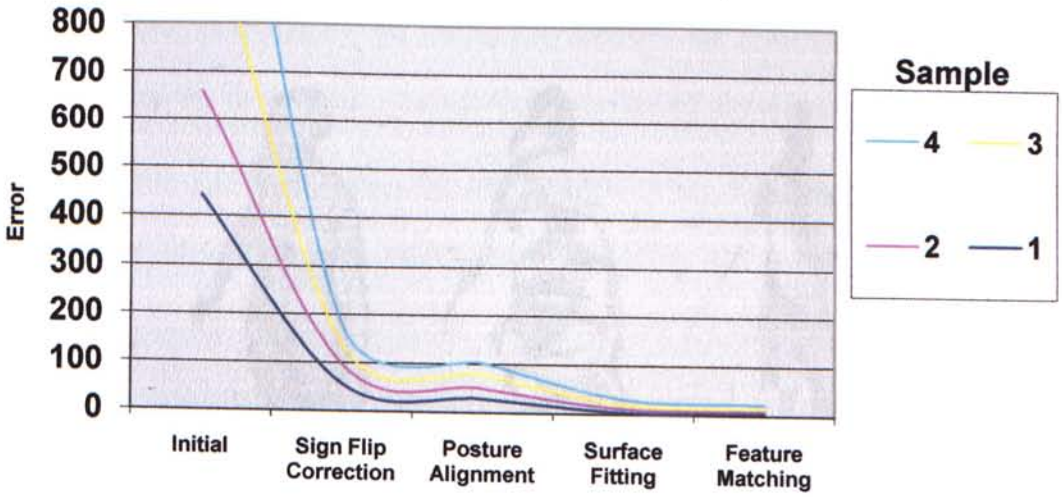

Table 4-3: Chart of distance error in different stage of the matching processes.

It is difficult to measure the accuracy of anchor points mapping numerically. Instead of measuring the anchor points only, the distance error function measure all 1-to-1 mapping of the points between template and input model. Therefore the actual accuracy of the anchor points should be different from the calculated result.

4.3.3 Approach Comparison

The table below illustrated the distance error of different matching approaches. (RED) Large error induced because of incorrect sign flip correction in MDS approaches.

	Sample 1	Sample 2	Sample 3	Sample 4
Jain's MDS approach	17.75	**421.72**	20.95	**405.1**
Elad's MDS approach	**387.5**	**426.5**	**310.91**	39.72
Chui's Iterative Alignment	76.28	56.33	44.25	71.77
Our approach	**3.59**	**5.25**	**6.45**	**7.91**

Table 4-4: Approach Comparison of the proposed approach with other previous works.

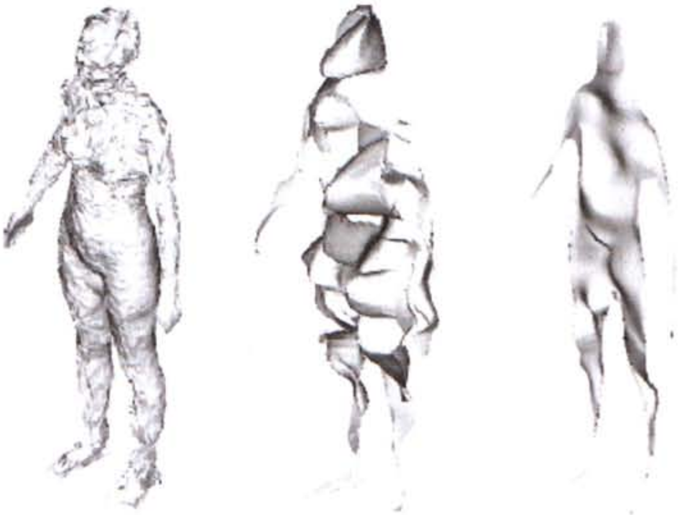

Figure 4-24: Simulation of the transformation of the template with other approaches. Jain's approach (left) transform the object by MDS mapping without surface fitting and smoothness; Elad's approach (middle) using 3-D MDS embedding for transformation; Chui's approach (right) using soft-assign correspondence determination cause local maxim in sharpen regions.

Chapter 5 Conclusion

5.1 Strength and contributions

The proposed algorithm in this thesis presents a robust correspondence identification algorithm on a 3D model by referencing the isometric similar template model. The presented approach is designed for engineering applications that required robust feature identification on the surface of 3D human bodies. The experimental testes have verified the correctness and effectiveness of the proposed approach. The research work presented in this thesis can support the geometric solution or the design automation of human-centered customization of free-form products including clothes, shoes, glasses, etc. As a preprocessing step for the volumetric parameterization for design automation [WHT07], the automatic method for extracting semantic feature points can further shorten the time of product design and fabrication cycle. The major contribution of the proposed algorithm can be outlined as follow:

- A MDS-based point matching algorithm is investigated to align the initial correspondences between the template human model and the given 3D human model.
- A sign flip correction technique is developed to enhance the robustness of MDS embedding, which is very important for solving symmetric models matching problem such as human bodies.
- Starting from the initial correspondence, a global alignment technique is exploited to iteratively find a mapping function (via the point correspondences) that optimizes surface proximities and is sensitive to semantic features.

5.2 Limitation and future works

According to the experimental tests, a satisfactory result can be obtained for those testing examples with moderate level of deformation. However, one of the limitations of the proposed approach is its restriction on the deformation effects between the models in local regions. – specifically, isometric deformation is assumed. For instance, a particular highly stretched area, a dense point distributed region or a twisted surface may vanish the validity of the algorithm. Future works can be done to overcome this problem and one possible solution is to segment the mesh surface before applying MDS transformation. This can greatly reduce the stretch error accumulated in the MDS embedding and eliminate those local dense regions. However, there is a drawback of multiple sign-flip correction problems if the divided segment is symmetrically identical, for example, sign flip correction cannot work on two arm segments alone. Therefore, the possibility of segmentation is still under evaluation at this moment.

Last but not least, more future work can be done to enhance the performance of the proposed algorithm. In current implementation, the computation time is highly depends on the sample rates, number of iterations and the RBF warping processes in each steps. In our near future work, the parallel computing power, which is nowadays available on the desktop PCs, will be consider to speed up the proposed approach.

References

[ASDLD03] Alliez P., Steiner D.C., Devillers O., Levy B, and Desbrun M., 2003, "Anisotropic polygonal remeshing", ACM Trans, on Graphics, vol.22, pp.485-493

[BDH96] Barber C.B., Dobkin P.D., and Huhdanpaa H., 1996, "The quickhull algorithm for convex hull", ACM Trans, on Mathematical Software, vol.22, no.4. pp.469-483

[Blu73] Blum H., 1973, "Biological Shape and Visual Science", J. Theor. Biol., vol.38, no.2, pp.205-287.

[BG97] Borg I. and Groenen P., "Modern Multidimensional Scaling – Theory and Applications", Springer, 1997.

[BBK07] Bronstein A.M., Bronstein M.M. and Kimmel. R., 2007, "Calculus of non-rigid surfaces for geometry and texture manipulation", IEEE Trans. Visualization and Computer Graphics

[CFB97] Carr. J.C., Fright W.R. and Beatson R.K., 1997, "Surface interpolation with radial basis functions for medical imaging", IEEE Trans. on Med. Imag, vol.16, no.1, pp.96-107

[CJGQ05] Carner C., Jin M., Gu X. and Qin H., 2005, "Topology-driven Surface Mapping with Robust Feature Alignment", IEEE Visualization, p.69

[CK04] Caelli T and Kosinov S, 2004, "An Eigenspace Projection Clustering Method for Inexact Graph Matching", IEEE Trans on PAMI, Vol. 26, No. 4, pp. 515-519.

[CR03] Chui H, and Rangarajan A., 2003, "A new point matching algorithm for non-rigid registration", Computer Vision and Image Understanding, v.89, n.2-3, p.114-141.

[CS05] Cagri. A. and Sibel. T., 2005, "An Axis-Based Representation for Recognition", Proc. of 10th IEEE ICCV, vol.2, pp.1339-1346.

[EK01] Elad A. and Kimmel R., "Bending Invariant Representations for Surfaces", Proc. Computer Vision and Pattern Recognition (CVPR), Dec. 2001.

[EK03] Elad A. and Kimmel. R., 2003, "On bending invariant signatures for surfaces", IEEE Trans, Pattern Analysis and Machine Intelligence, vol.25, no.10, pp.1285-1295

[ELPZ97] Eldar Y., Lindenbaum M., Porat M., and Zeevi Y.Y., "The Farthest Point Strategy for Progressive Image Sampling", IEEE Trans. Image Processing, vol. 6, no.9, pp. 1305-1315, 1997.

[FL95] Faloustos C. and Lin D. K.I., 1995, "FastMap: a fast algorithm for indexing, data mining and visualization of traditional and multimedia datasets", Proc, of the 1995 ACM SIGMOD international conference on Management of data, vol.24, no.2, pp.163-174

[GG03] Gatzke T.D. and Grimm C.M., 2003, "Assessing curvature metrics on triangular meshes", Washington University, St. Louis Missouri, Tech. Rep.

[GGGZ05] Gatzke T., Grimm C., Garland M., and Zelinka S., 2005, "Curvature maps for local shape comparison", Proceedings of Shape Modeling International, IEEE Computer Society Press.

[HSKK01] Hilaga M., Shinagawa Y., Kohmura T., and Kunii T.I., 2001, "Topology matching for fully automatic similarity estimation of 3D shapes", Proc. of the 28th annual conference on Computer graphics and iterative techniques SIGGRAPH'01, pp.203-212

[HPM03] Huang X., Paragios N. and Metaxas D.N., 2003, "Shape registration in implicit spaces using information theory and free form deformations", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.28, no.8, pp.1303-1318

[JZ06] Jain V. and Zhang H., 2006, "Robust 3D shape correspondence in the spectral domain", in Proc. International Conference on Shape Modeling and Applications (SMI), pp.118-129.

[JH02] Jiao X. and Heath M.T., 2002, "Feature Detection for Surface Meshes", The International Society of Grid Generation, pp.705-714

[JSHJ01] Joris V.S.M., Spanjaard S., Horvath I., and Jos J.J.O., 2001, "Fitting freeform shape patterns to scanned 3D objects", Journal of Computer and Information Science and Engineering, Vol. 1, Issue 3, pp. 218-224.

[JD03] Joshi N. and Dutta D., 2003, "Feature Simplification Techniques for Freeform Surface Models", Journal of Computing and Information Science in Engineering, vol.3. issue.3, pp.177-186

[Kal99] Kaleem S., et. al, 1999, "Shock Graphs and Shape Matching", International Journal of Computer Vision, vol.35, no.1, pp.13-32.

[KS98] Kimmel R. and Sethian J.A., "Computer Geodesic Paths on Manifolds", Proc. Nat'l Academy of Science, vol. 95, pp. 8431-8435, 1998.

[Kre99] Kreyszig E., 1999, "Advance Engineering Mathematics", John Wiley & Sons Inc, pp.925-928

[LWH07] Li S.M., Wang C.C.L. and Hui. K.C., 2007, "Correspondences Matching on 3D mesh models", Proceedings of ASME IDETC'07.

[LDRS05] Litke N., Droske M., Rumpf M., and Schroder P., 2005, "An image processing approach to surface matching", Eurographics Symposium on Geometry Processing, pp.207-216

[LD04] Lowe, D.G., 2004, "Distinctive Image Features from Scale Invariant Keypoints", International Journal of Computer Vision, 60, 2, pp.99-110.

[LF09] Lipman Y and Funkhouser T, 2009, "Mobius voting for surface correspondence", ACM SIGGRAPH, pp. 1-12.

[MN78] Marr. D. and Nishihara H.K., "Representation and recognition of spatial-organization of 3-D shapes", Proc. R. Soc. Lond. Ser. B-Biol. Sci., vol. 200, no. 1140, pp. 269-294, 1978.

[MDSB02] Meyer M., Desbrun M., Schroder P. and Barr A.H., 2002, "Discrete Differential-Geometry Operators for Triangulated 2-Manifolds", VisMath'02, pp.237-242

[MD03] Moennin C. and Dodgson N.A., 2003, "Fast marching farthest point sampling", in Proc. EUROGRAPHICS.

[NK02] Novotni M. and Klein R., 2002, "Computing geodesic distances on triangular meshes", Visualization and Computer Vision, pp.341-348

[PDJ93] Pankanti. S., Dorai C. and Jain A.K., 1993, "Robust Feature Detection for 3D Object Recognition and Matching", In Proc. SPIE Conference on Geometric Methods in Computer Vision, vol.2031, pp.366-377.

[RCB97] Rangaragan A., Chui H., and Bookstein F., 1997, "The softassign procrustes matching algorithm, in" information processing in medical imaging (IPMI)", Springer, Berlin, pp.29-42

[SB92] Shapiro L.S. and Brady J.M., 1992, "Feature based correspondence: and eigenvector approach", Image and Vision Computing, vol. 10, pp.283-288

[Set96] Sethian, "A Review of the Theory, Algorithms, and Applications of Level Set Method for Propagating Surfaces", Acta Numerica, 1996.

[SHI96] Shum H.-Y., Hebert M. and Ikeuchi K., "On 3D Shape Similarity", proc. IEEE Computer Vision and Pattern Recognition, 1996, pp. 526 – 531.

[SSKGH05] Surazhsky V., Surazhsky T., Kirsanov D, Gortler S.J. and Hoppe H, 2005, "Fast Exact and Approximate Geodesics on Meshes", proc. SIGGRAPH, pp. 553 – 560.

[Tsi95] Tsitsiklis, "Efficient Algorithms for Globally Optimal Trajectories", IEEE Trans. Automatic Control, vol. 40, pp. 1528-1538, 1995.

[WWTY04] Wang C.C.L., Wang Y., Tang K. and Yuen M.M.F., 2004, "Reduce the stretch in surface flattening by finding cutting paths to the surface boundary", Computer-Aided Design vol. 20, no.8, pp.665-677.

[Wan05] Wang C.C.L., 2005, "Parameterization and parametric design of mannequins", Computer-Aided Design, vol.37, no.1, pp.83-98

[WCY03] Wang C.C.L., Chang T.K.K. and Yuen M.M.F, 2003, "From laser-scanned data to feature human model: a system based on fuzzy logic concept", Computer-Aided Design, vol.35, no.3, pp.241-253

[WHT07] Wang C.C.L., Hui K.C.. and Tong K.M., 2007, "Volume parameterization for design automation of customized free-form products", IEEE Transactions on Automation Science and Engineering, vol. 4, Issue 1, pp.11-21.

[WWY03a] Wang.C.C.L. Wang.Y., and Yuen M.M.F., 2003, "Feature-based 3D non-manifold freeform object construction", Engineering with Computers, vol.19, no.2-3, pp.174-190.

[WWY03b] Wang C.C.L., Wang Y. and Yuen M.M.F., 2003, "Feature based 3D garment design through 2D sketches", Computer-Aided Design, vol.35, no.7, pp.659-672

[WWY05] Wang. C.C.L., Wang. Y. and Yuen M.M.F., 2005, "Design automation for customized apparel products", Computer-Aided Design, vol.37, no.7, pp.675-691

[WAAGK05] Wiley D.F., Amenta N., Alcantara D.A., Ghosh D. and Kil Y.J., 2005, "Evolutionary Morphing", in Proc. of IEEE Visualization 2005, pp.431-438.

[WWJGS06] Wang S., Wang Y., Jin M., Gu X. and Samaras D., 2006, "3D surface Matching and Recognition Using Conformal Geometry", Computer Vision and Pattern Recognition, IEEE Computer Society, vol.2, pp.2453-2460.