10.1007/s12541-009-0009-0

Three-dimensional Morphing of Similar Shapes Using a Template Mesh

Dong-Jin Yoo[1,#]

1 Department of Computer Aided Mechanical Design Engineering , Daejin University, Pocheon-Si, Kyeonggi-Do, South Korea, 487-711
Corresponding Author / E-mail: djyoo@daejin.ac.kr, TEL: +82-31-539-2031, FAX: +82-31-539-1970

KEYWORDS: 3D shape morphing, Template mesh, Implicit function, Mesh smoothing, Modified Laplacian coordinate

Shape morphing is the process of transforming a source shape into a target shape, through a series of intermediate shapes. There are two important problems to be considered in three-dimensional shape morphing: conforming mesh generation and path interpolation. In this paper, a novel approach in which a template mesh is mapped directly to the target mesh is proposed for the efficient treatment of the conforming mesh generation problem. Our mapping technique is based on a shape deformation method using an implicit function and the well-known mesh smoothing scheme, so the implementation of the method is very simple and robust. After mapping the source mesh to the target mesh, i.e., after obtaining a consistent mesh parameterization of the two shapes, the intermediate shapes are obtained by linear interpolation of the modified Laplacian coordinates of the source and target meshes. We demonstrate many examples of morphing between various shapes, including a model of the human head, a head sculpture model, and models of the human body in different poses to show the validity and effectiveness of the proposed method.

Manuscript received: June 16, 2008 / Accepted: September 11, 2008

1. Introduction

Shape morphing is the process of transforming a source shape into a target shape through intermediate shapes. Morphing has been widely used in many areas, such as computer-aided design, computer graphics, movies, and animation. In the early stages of research, most work focused on two-dimensional (2D) shape-morphing applications[1-3] such as image morphing and 2D shape blending. More recently, more attention has concentrated on three-dimensional (3D) shape morphing. Two main problems in 3D shape morphing are conforming mesh generation and path interpolation.

Many methods for the generation of the conforming mesh have been proposed, including harmonic mapping,[4] MAPS,[5] and consistent mesh parameterization.[6-8] However, these techniques require tedious and time-consuming manual operations by the user.

A number of approaches to path interpolation have been presented. Cohen et al.[9] proposed a mixed method of warp transformation and distance-field interpolation to model the path interpolation problem. Alexa et al.[10] also suggested a model for the solution of the path interpolation problem using body elements to obtain more natural intermediate shapes. Breen et al.[11] described a technique that uses the level-set method to morph between shapes that have different topologies, such as the MRI scanning model and the scanned model in STL format. A method based on linear interpolation of Laplacian coordinates was proposed for generating more natural and plausible intermediate shapes by Alexa.[12] Recently, Yan et al.[13] proposed an approach based on the finite element method to describe the behavior of intermediate shapes. However, they applied the method to the morphing of relatively simple geometry using a small number of bar and rectangular elements. In our previous work,[14] we proposed a 3D morphing method in which the mean value coordinate and Laplacian coordinate are used simultaneously for morphing between two models that have different topologies and dissimilar shapes. After deforming the source mesh towards the target mesh as closely as possible using the shape deformation method based on a control mesh, the intermediate shapes were generated by linear interpolation of the distance fields of the two models. This method was successfully applied to the morphing of a large complex model composed of triangle meshes, as shown in Fig. 1. However, when the two models have significantly different poses, as shown in Fig. 2, the morphing result will be unsatisfactory if we use the previous morphing methods.

Therefore, we require a novel method that can take into account pose differences of the two models. Zhang et al.[15] suggested an approach based on a least-squares mesh method for achieving reasonable morphing results between two models with significantly different poses. This involves a template-based mapping technique for establishing a point-to-point correspondence between the two models. However, the method requires a specific equation solver suitable for large sparse matrices. Brett et al.[16] developed a method for fitting

high-resolution template meshes to detailed human body range scans with sparse 3D markers. After formulating the optimization problem in which the objective function is a weighted combination of three measures (data error, smoothness error, and marker error), they solved for the unknowns, i.e., the degrees of freedom of an affine transformation at each template vertex, using a non-linear optimizing solver. However, they assumed that the pose of the template mesh should be similar to the target mesh. If the poses are quite different, then the optimized template mesh can become stuck in local minima.

In this paper, we propose a novel method that uses shape deformation based on the radial basis function; conventional mesh smoothing is used for mapping a high-resolution template mesh to the detailed target mesh with a set of markers on the models to be morphed. Our mapping technique is based on a consistent parameterization framework, similar to the framework of Zhang et al.[15] and Brett et al.[16] However, our technique differs significantly from those methods in some respects. Our method requires neither a specific equation solver for the sparse matrix nor a non-linear optimizing solver for minimizing complex object functions. This direct mapping method based on radial basis deformation and mesh smoothing is the key new concept of this paper.

The main contribution of this paper is a template-based mapping technique for establishing point-to-point correspondence and consistent mesh parameterization among a set of similar models in different poses. In addition, we introduce a linear interpolation method based on modified Laplacian coordinates to generate more natural and plausible intermediate shapes.[14] The proposed method also requires manual placement of markers; however the source mesh is mapped directly to the target mesh without any intermediate domain, such as the cylinder, sphere, or polygonal mesh used in earlier methods. The markers are required only for natural and plausible morphing between the two models. For example, if a marker is located at the tip of the nose on one face, then it should be located at the tip of the nose on the other face. Therefore a relatively small number of markers is sufficient to obtain proper vertex correspondence between the two models. This is an improvement over earlier methods in which marking is tedious and time consuming, since both the positions and numbers of markers are very important for the quality of the mapped mesh on the intermediate domain and the morphing results. The proposed method does have a limitation in that it is suitable only for models that have similar gross shapes, much like other earlier methods based on template mesh mapping.[15,16]

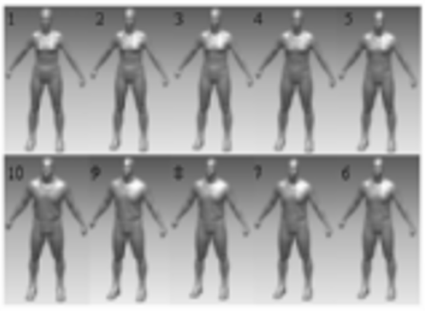


Fig. 1 Results of morphing a woman to a man using distance-field interpolation[14]

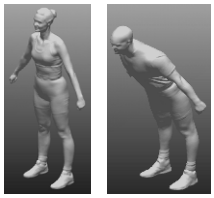


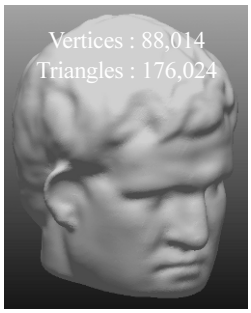
Fig. 2 Models with different shapes and poses

The remainder of this paper is organized as following. In Section 2, we propose a template-based mapping method in which the source mesh is mapped directly to the target mesh without any intermediate domain. In Section 3, we introduce a path interpolation method in which a modified Laplacian coordinate is used to speed up the convergence of a Gauss-Seidel iterative solver. In Section 4, we demonstrate the utility of our approach by presenting a variety of morphing applications for various types of digital characters. We conclude the paper with some discussion and ideas for future work in Section 5.

2. Mapping from a template mesh to a target mesh

We now describe our technique for mapping a template mesh to a target mesh. This means creating a new triangle mesh that approximates the shape of the target mesh using the same topology as the template mesh. Here, the *topology* means the number of vertices and triangles of the triangle mesh and connectivity information. Figure 3 provides the basic concept of this mapping procedure. After performing this mapping procedure, the new triangle mesh in Fig. 3(c) can be generated that approximates the shape of the target mesh in Fig. 3(b) and has the same topology as the template mesh in Fig. 3(a). That is, the original template mesh is exactly equal to the source mesh in both shape and topology. Therefore we use the terminology original template mesh instead of source mesh in this paper.

(a) Igea model (template mesh) (b) Agrippa model (target mesh)



(c) A new mesh that has the same geometry as Agrippa and the same topology as Igea (final mapped template mesh)

Fig. 3 Mapping the connectivity of a template mesh onto a target mesh

In this paper, we propose a shape deformation method based on an implicit surface interpolation scheme[21-25] to establish a point-to-point correspondence between a template mesh and a target mesh. Traditionally, an implicit surface is used to interpolate the coordinates of an unorganized point cloud. However in this method, the discrete displacements identified at some selected markers in the model,

instead of the coordinates of markers, are interpolated to create a smooth scalar field for the displacements. Since the scalar field is faithful to the input displacements and smooth over the whole surface of the model, the displacements at arbitrary points located on the model can be calculated in a continuous manner. Using this interpolation for the discrete displacements at a set of marked points, the original template mesh (source mesh) can be deformed smoothly toward the target mesh while preserving the correspondence between selected pairs of markers, as shown in Fig. 4.

The equation of an implicit surface can be defined as [25]

$$f(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \phi(\mathbf{x}-\mathbf{c}_j) + P(\mathbf{x}) , \tag{1}$$

where $\mathbf{c}_j$ represents the position vectors of the user-selected markers on the original template mesh, λ_j indicates the unknown weights for the linear combination of radial basis functions that should be calculated, ϕ is the basis function, and $P(\mathbf{x})$ is a first-degree polynomial. The basis function for interpolation, ϕ, is the well-known thin-plate radial basis function defined by

$$\phi(\mathbf{x}) = |\mathbf{x}|^2 \log(|\mathbf{x}|) . \tag{2}$$

The implicit surface interpolation problem for the discrete displacements at the user-selected markers can be written in the following form by substituting the required displacement value at each position vector of the user selected markers into Eq. (1).

$$dist_i = \sum_{j=1}^{N} \lambda_j \phi(\mathbf{c}_i - \mathbf{c}_j) + P(\mathbf{c}_i) , \tag{3}$$

where $dist_i = f(\mathbf{c}_i)$ for $(1 \le i \le N)$. Here, $dist_i$ is the displacement between the original template mesh and the target mesh in the x-, y-, and z-directions at the ith marker, and N is the number of user-selected markers for the deformation. The weights λ_j must satisfy the following orthogonal condition:[22,25]

$$\sum_{j=1}^{N} \lambda_j = \sum_{j=1}^{N} \lambda_j C_j^x = \sum_{j=1}^{N} \lambda_j C_j^y = \sum_{j=1}^{N} \lambda_j C_j^z = 0 . \tag{4}$$

Substituting the constraint Eqs. (3) and (4) into Eq. (1), we obtain a linear system of equations in matrix form as follows:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \cdots & \phi_{1N} & 1 & C_1^x & C_1^y & C_1^z \\ \phi_{21} & \phi_{22} & \phi_{23} & \cdots & \phi_{2N} & 1 & C_2^x & C_2^y & C_2^z \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{N1} & \phi_{N2} & \phi_{N3} & \cdots & \phi_{NN} & 1 & C_N^x & C_N^y & C_N^z \\ 1 & 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\ C_1^x & C_2^x & C_3^x & \cdots & C_N^x & 0 & 0 & 0 & 0 \\ C_1^y & C_2^y & C_3^y & \cdots & C_N^y & 0 & 0 & 0 & 0 \\ C_1^z & C_2^z & C_3^z & \cdots & C_N^z & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \\ P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} dist_1 \\ dist_2 \\ \vdots \\ dist_N \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

where $\phi_{ij} = \phi(\mathbf{c}_i - \mathbf{c}_j)$. The unknowns λ_j and the coefficients of $P(\mathbf{x})$ can be obtained by solving the equations.

After determining λ_j and the coefficients of $P(\mathbf{x})$, we can obtain the displacements at all vertices of the original template mesh using Eq. (1). This means that we can easily obtain the coordinates of the vertices of the deformed template mesh by simply adding the displacement to the original template mesh coordinate at each vertex. The deformed template mesh is thus exactly the same as the original template mesh in topology. The locations of marker points should be selected carefully to establish a plausible and natural morphing result.

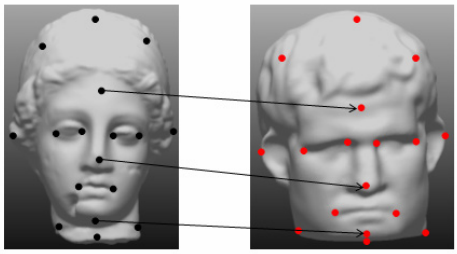


Fig. 4 A point-to-point correspondence for the generation of a smooth implicit surface for the displacements

If a vertex is located at the tip of the nose on one face, then it should be located at the tip of the nose on the other face. The locations and required number of user-selected markers must be chosen appropriately according to the geometrical complexity of the template mesh and the target mesh. Based on numerical experience, we recommend using 30-50 markers on each model to ensure stable and smooth implicit surface generation considering the complexity of the models. After deforming the original template mesh toward the target mesh, we must project the deformed template mesh to the target mesh to obtain a final mapping result, because the radial basis deformation process itself cannot obtain a finely mapped mesh. Since the deformed template mesh is obtained from the rough displacement field interpolated at the user-selected markers, it is similar to the target mesh only in gross shape. To obtain a robust mapping result, we use the well-known mesh smoothing technique to remove the sharp features existing in the template mesh because vertices on sharp features can be projected in an unwanted manner. In the worst case, the projected template mesh may have self-intersecting triangles, as shown in Fig. 5.

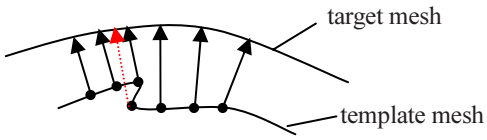


Fig. 5 Example of the self-intersecting triangles

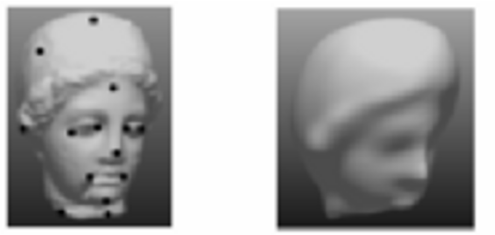
To overcome this problem, we introduce a mesh-smoothing scheme in which each vertex is iteratively relocated to the averaged center of gravity of the neighboring triangles. For mesh smoothing, a vertex is relocated to the averaged center of gravity of the neighboring triangles to improve the quality of the current triangle mesh as follows:[25]

$$\mathbf{P} = \frac{\sum_{i=1}^{n_c} A_i \mathbf{C}_i}{\sum_{i=1}^{n_c} A_i}, \tag{6}$$

where $\mathbf{P}$ is the position vector of the new location, A_i is the area of the ith triangle, $\boldsymbol{C}_i$ is the center of gravity of the ith triangle, and n_c is the number of triangles connected to the current vertex. After smoothing the deformed template mesh, each vertex in the template mesh is projected perpendicularly on the target mesh by calculating the closest point that has the minimum distance between a vertex and the target mesh. This projection and smoothing process is repeated until the distance between the projected template mesh and the target mesh is within a defined threshold. To this end, we define an error function based on the sum of the squared distances between each vertex in the template mesh and the target mesh as follows:

$$E = \sum_{i=1}^{n} dist^2(\mathbf{P}_{template,i}, \mathbf{P}_{target,i}), \quad (7)$$

where n is the number of vertices in the template mesh, $\boldsymbol{P}_{\text{template,i}}$ is the ith vertex of the template mesh, $\boldsymbol{P}_{\text{target,i}}$ is the point projected perpendicularly onto the target mesh of the $\boldsymbol{P}_{\text{template,i}}$, and the *dist*() function computes the distance to the closest compatible point on the target mesh.



(a) original template mesh (b) deformed and smoothed template mesh

(c) final mapped template mesh (approximated target mesh)

Fig. 6 Generating consistent mesh by mapping the connectivity of the template mesh onto the target mesh

We use a threshold of 10^{-6} mm^2 in our calculations. To accelerate the minimum-distance calculation, we pre-compute a cell structure for the target mesh, so that the candidate closest triangles are allocated to each cell in advance. Of course, the points marked previously should be fixed in the course of this iterative projection and smoothing process. As shown in Fig. 6, the template mesh of Fig. 6(a) is deformed and smoothed to the intermediate mesh of Fig. 6(b), and this intermediate mesh is transformed to the final mapped mesh of Fig. 6(c) through the iterative projection and smoothing process. Based on numerical results, we have determined that we can obtain a complete mapping result within 10 iterations on average.

3. Morphing between two shapes using the dual linear interpolation of a modified Laplacian coordinate

After building a one-to-one vertex correspondence between two shapes, many methods simply use linear interpolation of the coordinates of the two shapes to generate the intermediate shapes. However, shrinkage or kinks may occur in the morphing sequences using this method because large rotations cannot be represented correctly by linear interpolation.[14]

To address this problem, several researchers have considered non-linear approaches to interpolating the shapes. Sun et al.[17] and Sheffer et al.[18] used dihedral angles and edge lengths to interpolate two shapes composed of triangle meshes. Alexa et al.[12] used a Laplacian coordinate for interpolating the vertices. Surazhsky et al.[19] considered the interior information of given shapes and proposed methods for controlling local volume distortions. Recently, Hu et al.[20] used dual Laplacian coordinates to interpolate the vertices. In that method, the intermediate shapes are recovered from the interpolated mean curvature flow in the dual mesh domain. This can generate visually pleasing and physically plausible morphing sequences and avoid the shrinkage that appears in the traditional linear interpolation method. That method also uses an iterative scheme to calculate the Laplacian coordinates of intermediate shapes and a sparse solver to calculate the vertex positions of these intermediate shapes. Therefore implementation of this method is expensive in computing time and memory if the models to be morphed have a large number of vertices.

In this paper, we use one of our earlier approaches[14] for interpolating two shapes when the vertex correspondences between them have been well established. Our method is based on a modified Laplacian coordinate, similar to the framework of Alexa et al.[12] and Hu et al.[20] However, our method is significantly different from those methods in that we use an iterative solver by slightly modifying the form of the traditional Laplacian coordinate instead of using a sparse matrix equation solver. Our approach has two main advantages over existing interpolation approaches. This modification of the Laplacian coordinate enables the treatment of a model with a large number of vertices and provides rapid convergence to the solution.

Let $\boldsymbol{P}(\boldsymbol{P}_1, \boldsymbol{P}_2, \cdots, \boldsymbol{P}_n)$ be the mesh vertex positions and N_i be the index set of vertices adjacent to $\boldsymbol{P}_i$. The Laplacian coordinate of a vertex $\boldsymbol{P}_i$ is defined as follows:

$$\boldsymbol{\delta}(\mathbf{p}_i) = \mathbf{p}_i - \frac{1}{N_i}\sum_{j \in N_i} \mathbf{P}_j . \quad (8)$$

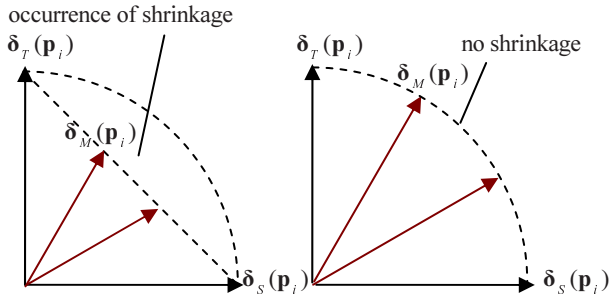
Since the Laplacian coordinate $\boldsymbol{\delta}(\mathbf{p}_i)$ is the difference vector from a vertex $\boldsymbol{P}_i$ to the center of its adjacent vertices, it describes the local geometry at the vertex $\boldsymbol{P}_i$. Let the Laplacian coordinate of a vertex $\boldsymbol{P}_i$ in the source mesh be $\boldsymbol{\delta}_S(\mathbf{p}_i)$, and the Laplacian coordinate of the vertex $\boldsymbol{P}_i$ in the target mesh be $\boldsymbol{\delta}_T(\mathbf{p}_i)$. Then the Laplacian coordinate $\boldsymbol{\delta}_M(\mathbf{p}_i)$ of the vertex $\boldsymbol{P}_i$ in an intermediate shape can be described as follows:

$$\boldsymbol{\delta}_M^U(\mathbf{p}_i) = (1-u)\frac{\boldsymbol{\delta}_S(\mathbf{p}_i)}{|\boldsymbol{\delta}_S(\mathbf{p}_i)|} + u\frac{\boldsymbol{\delta}_T(\mathbf{p}_i)}{|\boldsymbol{\delta}_T(\mathbf{p}_i)|} (0 \le u \le 1) \quad (9a)$$

$$|\boldsymbol{\delta}_M(\mathbf{p}_i)| = (1-u)|\boldsymbol{\delta}_S(\mathbf{p}_i)| + u|\boldsymbol{\delta}_T(\mathbf{p}_i)| (0 \le u \le 1) \quad (9b)$$

$$\boldsymbol{\delta}_M(\mathbf{p}_i) = |\boldsymbol{\delta}_M(\mathbf{p}_i)|\boldsymbol{\delta}_M^U(\mathbf{p}_i) \quad (9c)$$

In the dual linear interpolation method proposed in this work, the direction and magnitude of Laplacian coordinates are interpolated as shown in Eq. (9). We can prevent the shrinkage problem using this dual linear interpolation of the direction and magnitude of the Laplacian coordinates as shown in Fig. 7.



(a) traditional linear interpolation (b) dual linear interpolation

Fig. 7 Simplified diagram showing the difference between the Laplacian coordinate interpolation schemes

Since $\boldsymbol{\delta}_M(\mathbf{p}_i)$ means a relative position vector of $\boldsymbol{P}_i$ with respect to its

adjacent vertices, we must solve a large sparse linear system to obtain its real 3D coordinate.

$$[A]\,[V]_M = [D]_M\,, \tag{10}$$

where $[D]_M$ is a matrix of Laplacian coordinates, $[A]$ is the mesh adjacency matrix, and $[V]_M$ is a matrix of 3D coordinates. Since the size of the adjacency matrix is proportional to the number of vertices of the models, solving the linear system may require a lot of computing time and memory when using an un-optimized direct solver based on the LU decomposition or Gaussian elimination methods. To overcome this, we introduced a Gauss-Seidel iterative method based on a modified Laplacian coordinate suggested in our previous work.[14] Using that method, we can obtain the solution without constructing the adjacency matrix $[A]$ by calculating column by column to reduce memory use and computation time significantly. However, one problem still remains. If we use the conventional Laplacian coordinate defined by Eq. (8), the coefficient matrix $[A]$ in Eq. (10) cannot satisfy the sufficient condition required for convergence given by Eq. (11).

$$\left|a_{ii}\right| \geq \sum_{j \neq i} \left|a_{ij}\right|. \tag{11}$$

To accelerate the convergence rate, we use a modified Laplacian coordinate as follows:[14]

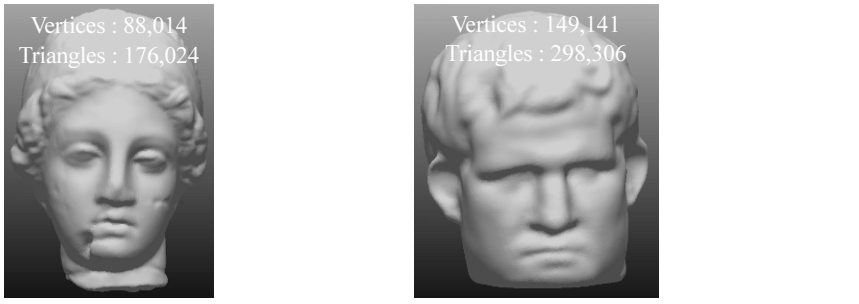
$$\boldsymbol{\delta}(\mathbf{p}_i) = C_1\mathbf{p}_i - C_2 \frac{1}{N_i} \sum_{j \in N_i} \mathbf{P}_j\,. \tag{12}$$

We can greatly reduce the number of iterations required for convergence by using a value larger than 1.0 for C_1 and a value less than 1.0 for C_2.

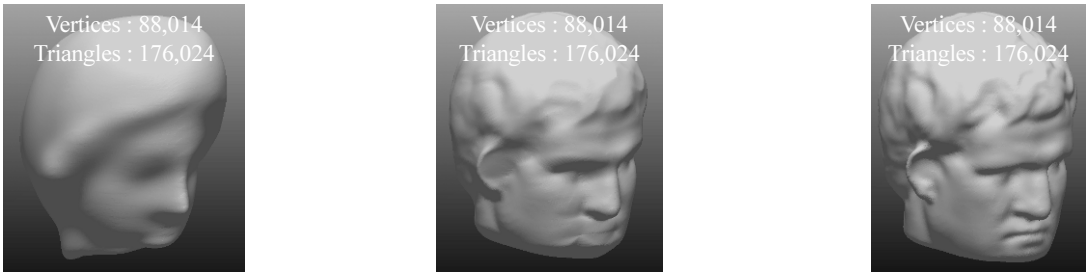
4. Results and Discussion

Various morphing operations were performed for large and complex triangle meshes with arbitrary shapes and poses to verify the effectiveness and validity of the proposed morphing algorithm. The proposed method was implemented using C language on a 3-GHz Pentium IV computer with 512 MB of memory. The most time-consuming part of our algorithm is mapping the template mesh to the target mesh. In all our experiments, it usually took 5-10 iterations for the mapping procedure to converge. The number of iterations depended on the geometrical complexity of the models. Table 1 shows the computation time for the examples illustrated in this paper. We used our mapping algorithm to create a mutually consistent mesh parameterization for all models. With a consistent parameterization, we could morph any two shapes by taking dual linear combinations of the modified Laplacian coordinates. Several examples with simple geometry are demonstrated in Figs. 8-12. These generally required 10-15 iterations for the iterative solver to converge. The results indicate that the proposed direct mapping and morphing method can be successfully applied to various types of model.

The robustness and flexibility of the proposed method were verified using more complex whole-body models. It is more difficult to find consistent parameterization for whole-body models with different shapes and poses. However, the proposed mapping algorithm was successfully applied to these complex models without any difficulty, as shown in Fig. 13. Figure 14 illustrates the transition from a standing woman to a bowing man. It is clear that the proposed dual interpolation of Laplacian coordinates of the two models based on Eq. (9) eliminates the shrinkage or kinks that may be caused by the conventional simple linear interpolation of Laplacian coordinates shown in Fig. 15(a). We obtained visibly pleasing and physically plausible morphing sequences between fairly different shapes and poses. The body rotated and deformed naturally without any defects, as shown in Fig. 15(b). The experimental results show that our approach is efficient and robust in computation, since our approach for obtaining the Laplacian coordinates of intermediate shapes is non-iterative, while preventing the shrinkage. In addition, our iterative solver based on modified Laplacian coordinates is more efficient in time and memory, since we can obtain vertex positions from the interpolated Laplacian coordinates without constructing an adjacency matrix $[A]$ by calculating the equations column by column, as



(a) Igea model (original template mesh, source mesh) (b) Agrippa model (target mesh)



(c) deformed and smoothed template mesh (d) intermediate projected template mesh (e) final mapped template mesh

Fig. 8 Generating a consistent mesh by mapping the connectivity of the Igea model onto the Agrippa model

Fig. 9 Morphing results of the Agrippa and Igea models

(a) Igea model (original template mesh) (b) deformed and smoothed template mesh (c) final mapped template mesh

Fig. 10 Generating a consistent mesh by mapping the connectivity of the Igea model onto a girl model

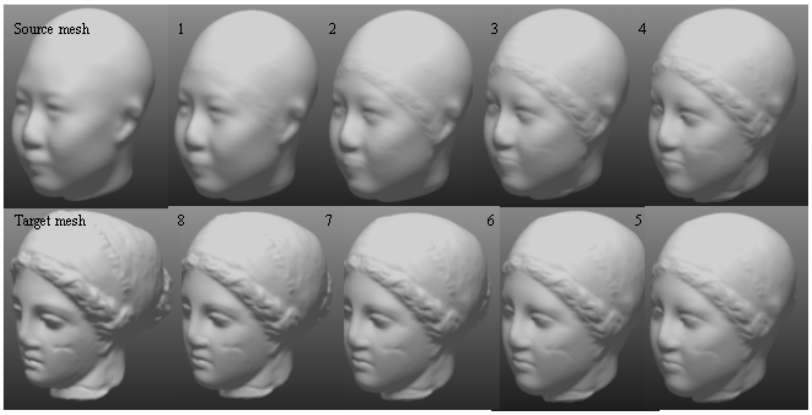


Fig. 11 Morphing results of girl and Igea models

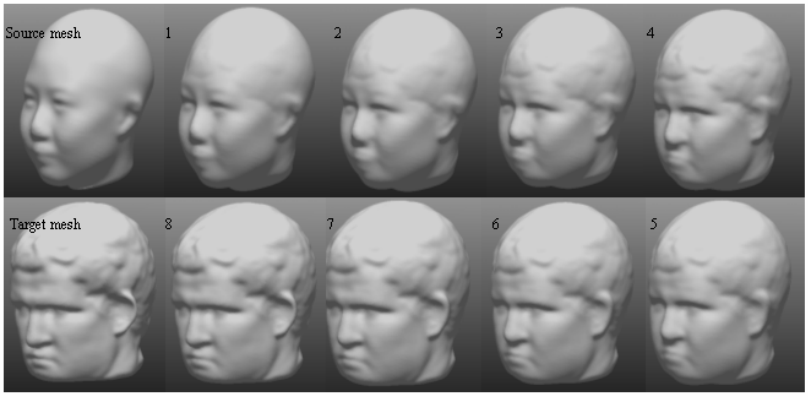


Fig. 12 Morphing results of girl and Agrippa models

(a) a bowing man (original template mesh) (b) deformed and smoothed template mesh (c) final mapped template mesh

Fig. 13 Generating a consistent mesh by mapping the connectivity of a bowing man onto a standing woman

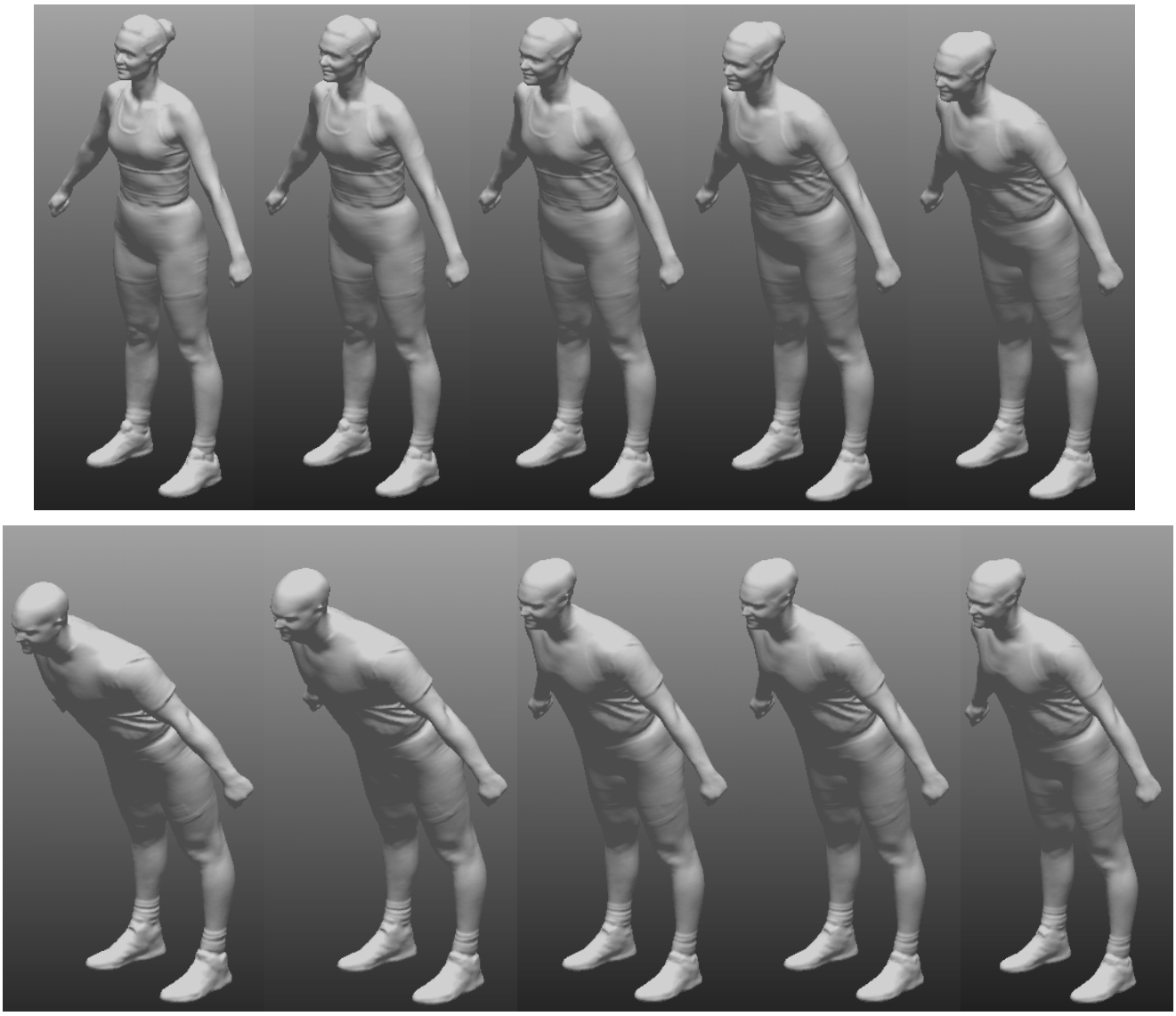
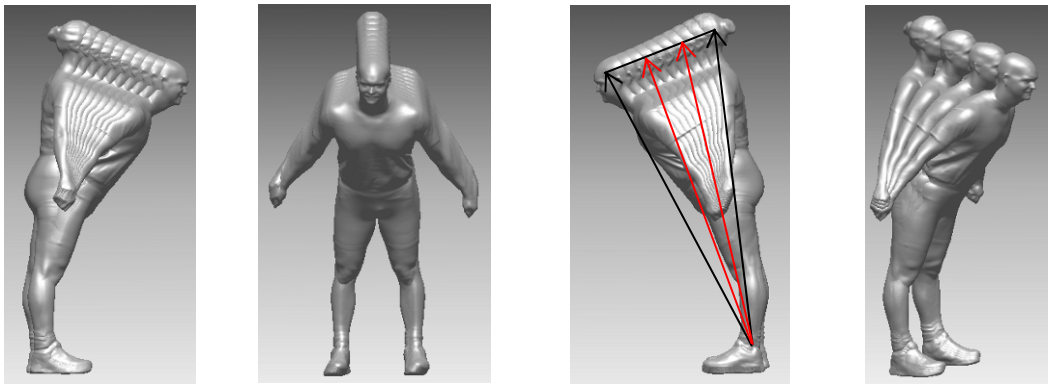
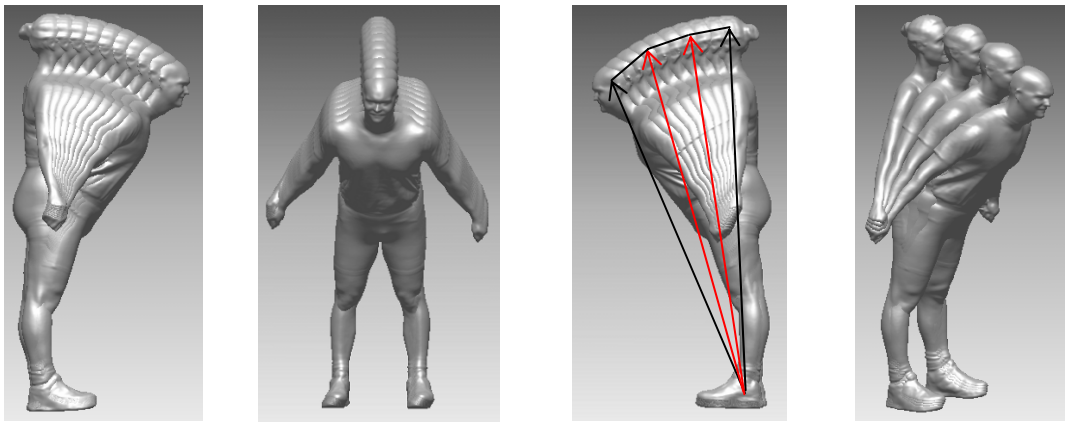


Fig. 14 Results of morphing a standing woman to a bowing man



(a) simple linear interpolation of Laplacian coordinates[12]



(b) dual linear interpolation of Laplacian coordinates based on Eq. (9)

Fig. 15 Comparison of Laplacian coordinate interpolation schemes

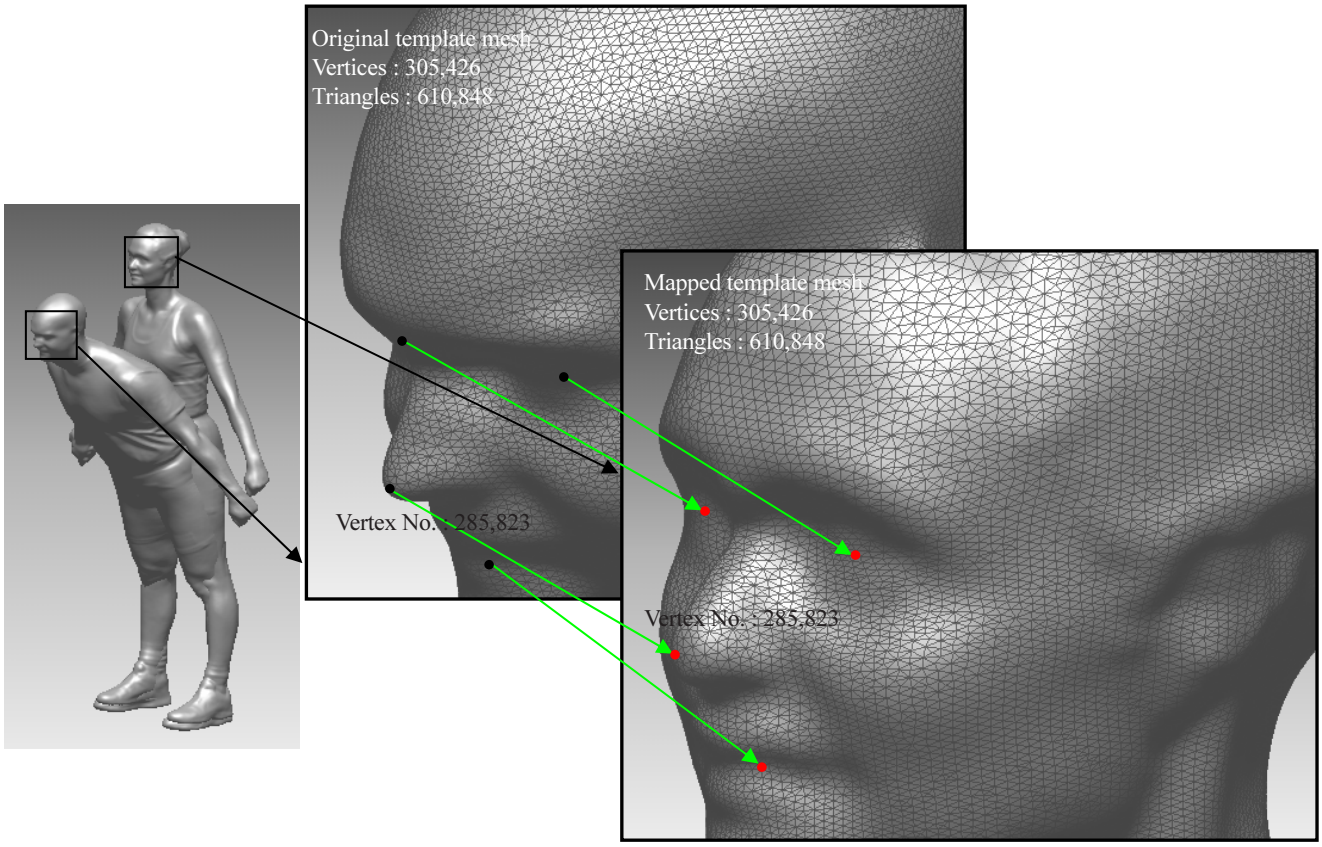
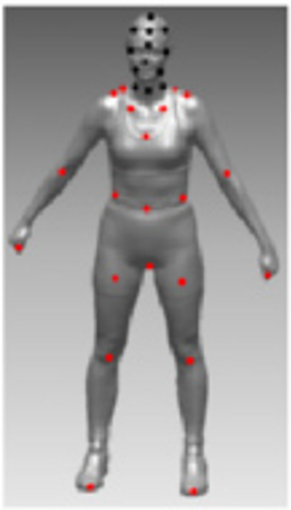


Fig. 16 Detailed mesh views of the original template mesh and the mapped template mesh



(a) markers for head rotation

(b) source mesh

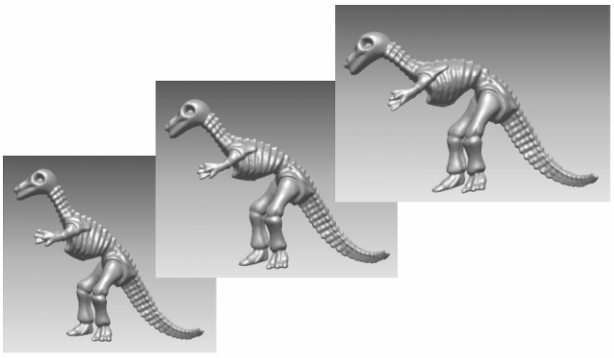
(c) target mesh generated from rotational deformation

Visibly pleasing and physically plausible morphing sequences

(d) head rotation of a standing woman

Fig. 17 Morphing results of a standing woman with head rotation

(a) rotational deformation of Indian statue model



(b) rotational deformation of Dinosaur model

Fig. 18 Examples of large rotational deformation using radial basis function

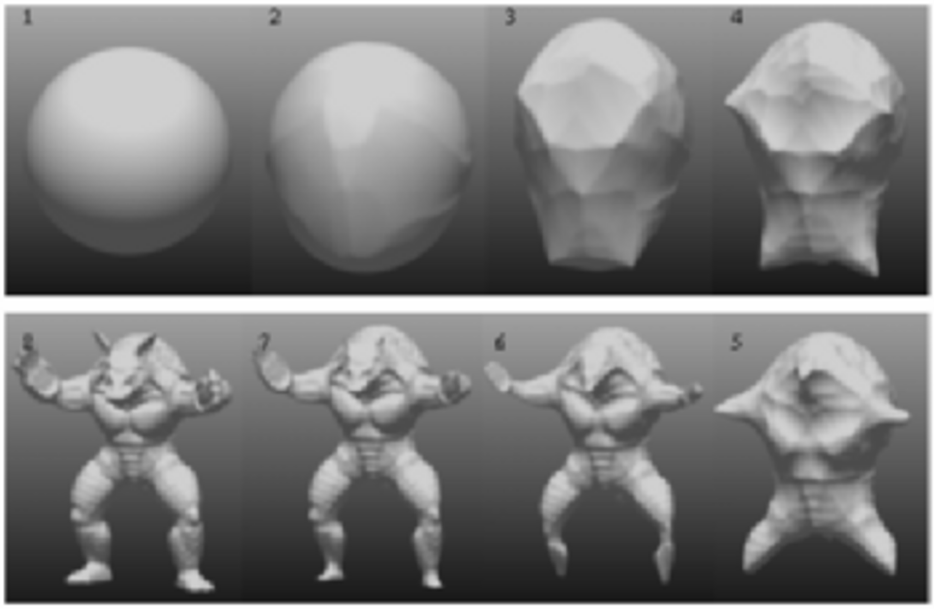


Fig. 19 Results of morphing a sphere to the Armadillo model[14]

Table 1 Computational results

Process	Number of vertices (triangles) of model	Proposed method Time (s)		Previous method[14] Time (s)	
		Mapping	Laplacian interpolation	Deformation	Distance-field calculation and interpolation
Morphing Igea to Agrippa	Template : 88,014 (176,024) Target : 149,141 (298,306)	124 number of user-selected markers : 37	7	9	473
Morphing girl to Igea	Template : 88,014 (176,024) Target : 73,152 (145,809)	76 number of user-selected markers : 32	7	9	322
Morphing girl to Agrippa	Template : 88,014 (176,024) Target : 88,014 (176,024)	0 (mapping is not required)	7	9	342
Morphing standing woman to bowing man	Template : 305,426 (610,848) Target : 195,486 (390,968)	420 number of user-selected markers : 48	21	impossible	
Morphing standing woman with head rotation	Template : 195,486 (390,968) Target : 195,486 (390,968)	12 number of user-selected markers : 52	17	impossible	
Morphing sphere to Armadillo model	Source : 125,476 (250,457) Target : 305,622 (613,600)	very difficult or may be impossible		0 (deformation is not required)	975

(Hardware: Pentium IV, 3-GHz CPU, 512 MB RAM)

described in Section 3. Our approach is several tens of times faster than the computational results of Hu et al.[20] in obtaining an intermediate shape from two conforming meshes. Table 1 of Hu et al.[20] gives the per-iteration running time for the examples used in that paper. Furthermore, that method can handle only a small number of vertices, as shown in Table 1,[20] since it requires the construction of a large sparse adjacency matrix. Figure 16 shows a detailed mesh view of the original template mesh and the mapped template mesh. By using a somewhat simplistic deformation method based on a radial basis function and robust mesh smoothing scheme, we obtained consistent mesh parameterizations between relatively complex shapes without the aid of a specific equation solver for the large sparse matrix required by Zhang's method,[15] or a non-linear optimizing solver required by Brett's method,[16] as shown in Fig. 16.

Figure 17 shows the morphing results of a standing woman with head rotation. If the source and target meshes are almost exactly the same shape, we do not produce a new mesh with a different shape. We perform only the radial basis deformation, and there is no need to carry out further projection processing for final mapping, as shown in Fig. 17. If we require rotation of the model, we simply substitute the constraints in Eq. (3) with the appropriate rotational displacements calculated from the simple rotation algorithm. Therefore, the constraint in Eq. (3) can take into account both linear and rotational displacements. Of course, if we require a rotation angle greater than about 30°, it would be more reasonable to perform the rotating process incrementally by dividing the rotation angle. Therefore Eq. (3) can take into account both small and large rotations in a consistent framework. By transforming a large rotation angle into a set of small rotation angles, we can obtain a natural and plausible rotated model even in the case of large rotation angles, as shown in Fig. 18.

Our morphing method has a limitation as previously mentioned. To the best our knowledge, one morphing method cannot be always superior to all other methods for all applications; all methods have both advantages and disadvantages. The proposed method is not suitable for morphing two models that are quite dissimilar in shape, i.e., a sphere and Armadillo model of Fig. 19, since it is very difficult or even impossible to obtain a final mapped template mesh. However, the proposed method can be used effectively to morph two models with similar shapes and different poses. For example, our previous method[14] is suitable for morphing two models that are quite dissimilar in shape, as shown in Fig. 19. However, when two models are much different in shape and poses as shown in Fig. 14, using the previous method will produce unsatisfactory results. Therefore, the appropriate algorithm should be selected according to the geometrical characteristics of the models to be morphed. Of course, some method may be devised to combine the advantages of these two methods. This is an open area for further study.

5. Conclusions and future work

We have shown that our template-based mapping method works fairly well in practice. We were able to map all of our examples to a reasonable degree, even for models in significantly different poses. Our method is based on shape-deformation using an implicit function and a simple mesh smoothing scheme. Therefore, the implementation of related algorithms is very simple and robust. In addition, more plausible and natural intermediate shapes can be obtained without the shrinkage or kinks that may occur in conventional linear interpolation of coordinates. Moreover, the convergence of the iterative solver is greatly improved by using modified Laplacian coordinates.

Our morphing framework has several limitations. First, the input template mesh must be accurate enough to represent the target mesh. If the input template mesh cannot describe the target mesh appropriately, a proper refinement technique must be introduced, and this means a change in the topology of the template mesh. Consequently, the point-to-point correspondence must be rebuilt in the iterative process and the implementation will be more complex and difficult. Second, the graphical user interface for the mapping must be more convenient and provide a simple and unified interface. The selection and editing of markers should be performed easily. Third, all of the models illustrated in this paper are composed of closed meshes with genus zero. For practical applications, the mapping algorithm should be extended to handle more complex models. This is an area that is open for future research. In addition, the local self-intersection problem during morphing of complex models needs to be investigated further.

ACKNOWLEDGEMENT

This work was supported by Daejin University Research grants in 2008.

REFERENCES

1. Beier, T. and Neely, S., "Feature-based image meta-morphosis," In Proc. SIGGRAPH '92, pp. 35-42, 1992.
2. Sederberg, T. W. and Greenwood, E., "A physically based approach to 2-D shape blending," In Proc. SIGGRAPH '92, pp. 25-34, 1992.
3. Sederberg, T. W., Gao, P., Wang, G. and Mu, H., "2D shape blending: an intrinsic solution to the vertex path problem," In Proc. SIGGRAPH '93, pp. 15-18, 1993.
4. Kanai, T., Suzuki, H. and Kimura, F., "Metamorphosis of arbitrary triangular meshes," IEEE Computer Graphics and Applications, Vol. 20, No. 2, pp. 62-75, 2000.
5. Lee, A. W. F., Dobkin, D., Sweldens, W. and Schroder, P., "Multiresolution mesh morphing," In Proc. SIGGRAPH '99, pp. 343-350, 1999.
6. Praun, E., Sweldens, W. and Schroder, P., "Consistent mesh parameterization," In Proc. SIGGRAPH '01, pp. 179-184, 2001.
7. Schreiner, J., Asirvatham, A., Praun, E. and Hoppe, H., "Inter-surface mapping," ACM Transactions on Graphics, Vol. 23, No. 3, pp. 870-877, 2004.
8. Kraevoy, V. and Sheffer, A., "Cross-parameterization and compatible remeshing of 3d models," ACM Transactions on Graphics, Vol. 23, No. 3, pp. 861-867, 2004.
9. Cohen-or, D., Levin, D. and Solomovici, A., "Three dimensional distance field metamorphosis," ACM Transactions on Graphics, Vol. 17, No. 2, pp. 116-141, 1998.
10. Alexa, M., Cohen-or, D. and Levin, D., "As rigid as possible shape interpolation," In Proc. SIGGRAPH '00, pp. 157-164, 2000.
11. Breen, D. E. and Whitaker, R. T., "A level-set approach for the metamorphosis of solid models," IEEE Transactions on Visualization and Computer Graphics, Vol. 7, No. 2, pp. 173-192, 2001.
12. Alexa, M., "Differential coordinates for local mesh morphing and deformation," The Visual Computer, Vol. 19, No. 2, pp. 105-114, 2003.

13. Yan, H. B., Hu, S. M. and Martin, R., "Morphing based on strain field interpolation," Computer Animation and Virtual Worlds, Vol. 15, Issues 3-4, pp. 443-452, 2004.

14. Yoo, D. J., "Three Dimensional Shape Morphing of Triangular Net," Journal of the Korean Society for Precision Engineering, Vol. 25, No. 1, pp. 160-170, 2008.

15. Zhang, L., Liu, L., Ji, Z. and Wang, G., "Manifold parameterization," In Proceedings of 24th Computer Graphics International '06, pp. 160-171, 2006.

16. Brett, A., Brian, C. and Zoran, P., "The space of human body shapes: reconstruction and parameterization from range scans," ACM SIGGRAPH '03, pp. 27-31, 2003.

17. Sun, Y., Wang, W. and Chin, F., "Interpolating polyhedral models using intrinsic shape parameters," Journal of Visualization and Computer Animation, Vol. 8, No. 2, pp. 81-96, 1997.

18. Sheffer, A. and Kraevoy, V., "Pyramid coordinates for morphing and deformation," In Proceedings of the International Symposium on 3D Data Processing, Visulalization and Transmission, pp. 68-75, 2004.

19. Surazhsky, V. and Gotsman, C., "Intrinsic morphing of compatible triangulations," International Journal on Shape Modelling, Vol. 9, No. 2, pp. 191-201, 2003.

20. Hu, J., Liu, L. and Wang, G., "Dual Laplacian morphing for triangular meshes," The Journal of Visualization and Computer Animation, Vol. 18, Issue 4-5, pp. 271-277, 2007.

21. Turk, G. and O'Brien, J. F., "Modelling with implicit surfaces that interpolate," ACM Transactions on Graphics, Vol. 21, No. 4, pp. 855-873, 2002.

22. Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C. and Evans, T. R., "Reconstruction and representation of 3D objects with radial basis functions," In Proceedings of SIGGRAPH '01, pp. 67-76, 2001.

23. Kojekine, N., Hagiwara, I. and Savchenko, V., "Software tools using CSRBFs for processing scattered data," Computer & Graphics, Vol. 27, No. 2, pp. 311-319, 2003.

24. Ohtake, Y., Belyaev, A., Alexa, M., Turk, G. and Seidel, H. P., "Multi-level partition of unity implicits," ACM Transactions on Graphics, Vol. 22, No. 3, pp. 463-470, 2003.

25. Yoo, D. J., "Filling Holes in Large Polygon Models Using an Implicit Surface Scheme and the Domain Decomposition Method," International Journal of Precision Engineering and Manufacturing, Vol. 8, No. 1, pp. 3-10, 2007.