

A Novel Method for 3D Morphing by Deformation Matrix with Triangle Meshes

C.L. Peng, T.W. Xing, Y. Yu, Y. Zhou, S.D. Du
School of Electronic Science and Engineering
Nanjing University
Nanjing, China

Abstract—We introduce a 3D morphing method which generates a merged model given a series of triangle meshes. Our morphing, based on a set of parameters between the source and target shapes, can show the process of the transformation from the source to the target smoothly. We choose a model as our reference mesh, and obtain corresponding unified models from other models which may have different number of vertices or facets. Given these unified models, parameters between any two meshes can be computed integrally or separately for each rigid part. Different forms of combination of the parameters can generate different merged models. To address the collapsed situation happened occasionally, shape and pose morphing are separated for some parts in our work. By merging different parts of different models, we can get a merged shape, e.g. an animal with the horse head and the cat body. As an application of our 3D morphing method, quantifying the difference between any two models can be done efficiently, represented by the distance between any two sets of low-dimensional parameters reduced from the initial parameters using Principal Component Analysis (PCA). Character replacement and model driven are another two applications. Characters in 2D images are used to guide our morphing work and depth image sequence is used to drive our merged model to show the same pose as the character in the sequence respectively.

Keywords—3D morphing; difference quantifying; character replacement; model driven

I. INTRODUCTION

A variety of related work has been done in the morphing of 3D models. One popular approach in [1] is sphere embedding. Athanasiadis *et al.*[2] made some improvements using an optimization technique, a mapping geometrically similar to the original object while preserving connectivity and topology can be obtained. Kraevoy *et al.*[3] propose another method by constructing a common base domain based on his another work, Matchmaker [4]. Given the common base domains, after initial cross-parameterization and compatible remeshing, a final new mesh can be generated by smoothing and refinement. Bronstein *et al.*[5] explore the problem of similarity criteria between nonrigid shapes. Assuming the models have point-to-point correspondences, Allenet *et al.*[6] do the morphing between two models by taking linear combination of the vertices.

Different human models have similar scale, while different animals may have a huge difference in shape and size. We take an inspiration in the work of deformation transfer by Sumner and Popović [7]. They use a 3×3 matrix to represent the

deformation of each source triangle to the corresponding target. We use this transformation matrix to address the task of 3D morphing. Getting the models with the point-to-point correspondences is just like to warp the template mesh onto the targets. Kaicket *et al.*[8] review a series of methods to compute correspondences between geometric shapes. With a model as the deformable template and the other models as the targets, the classical method is an extending Iterative Closest Points(ICP) algorithm for modeling non-rigid objects, such as [9, 10]. However, this method is not stable and good results are yielded only when the inter-frame deformations are small.

In this paper, an efficient 3D morphing method is introduced. With these unified models, we can compute the transformation matrices from the source model to the target, using the method similar to SCAPE[11]. Given the transformation and a source model, 3D morphing can be done smoothly. Merging different parts which belong to different models is included in our work, e.g. a shape which consists of a human body and a cat head. The other two applications of our method are character replacement in 2D image and driving a merged model by a motion sequence.

Our 3D morphing method has two notable contributions:

- The new model generated by using our morphing method maintains the point-to-point correspondences with the template mesh, without increasing the number of mesh vertices and facets.
- Character replacement in an image and model driven by a depth sequence can be done with the help of our 3D morphing method.

II. METHOD

One of the main purposes of our approach is to deform the source shape to the target smoothly and another is to generate a merged model. By adding sparse markers on any two models, deformation transfer can morph one triangulated mesh to another gradually in the process of an iterated closet point algorithm with regularization. Assuming the meshes have point-to-point correspondences, affine transformation matrices between the paired triangles can be computed like SCAPE. Taken horse and cat as an example, as the the proportion of cat's transformation parameters is increasing, the model generated is more like a cat.

Our database mainly comes from a CG software called Poser. SCAPE datasets are also included. Every model in

SCAPE datasets have 12500 vertices and 25000 triangular facets, with point-to-point correspondences.

A. Definitions

The terms used in our method are defined in this way that our datasets consist of a source model S and a set of targets $T = \{T^1, \dots, T^N\}$. The source mesh $S = \{V_S, F_S\}$ has a set of vertices $V_S = \{v_1, \dots, v_j\}$ and a set of triangular facets $F_S = \{f_1, \dots, f_k\}$.

B. Transformation Parameters

Assuming our models have correspondence between each pair of triangular facets, transformation parameters can be computed in the following method. Let the paired triangular facets f_k and \widehat{f}_k have the vertices $v_{k,1}, v_{k,2}, v_{k,3}$ and $\widehat{v}_{k,1}, \widehat{v}_{k,2}, \widehat{v}_{k,3}$. Since we just want to use an affine transformation defined by a 3×3 matrix Q , not using a displacement vector d , deformations are applied in the local coordinate system. We represent the deformation as $\widehat{V} = Q \times V$ where $V = [v_2 - v_1, v_3 - v_1]$, $\widehat{V} = [\widehat{v}_2 - \widehat{v}_1, \widehat{v}_3 - \widehat{v}_1]$. Since it is not enough to determine the affine transformation matrices, we add a regularization to constrain the problem following [6, 7]. With a smoothness constraint which indicates the transforms should be similar in adjacent triangular facets, this problem can be described as the following form:

$$\min_{Q_1, \dots, Q_N} \sum_{k=1}^N \sum_{i=2,3} \left\| Q_k V_{k,i} - \widehat{V}_{k,i} \right\|^2 + \omega_s \sum_{k=1}^N \sum_{j \in \text{adj}(k)} \|Q_k - Q_j\|^2 \quad (1)$$

where $V_{k,i} = v_{k,i} - v_{k,1}, i = 2, 3$, and N is the number of the triangular facets. This equation can be solved integrally for the entire shape separately for each rigid part by using the least squares method.

III. SHAPE MORPHING

Given the source model and a set of transformation parameters from the source to the target, 3D morphing can be achieved smoothly. Without loss of generality, each triangular facet f_k can be represented as two vectors $v_{2,1}, v_{3,1}$. The generated triangle can be represented as the following formulation:

$$\widehat{V}_{k,i} = (\omega Q_k^m + (1 - \omega) Q_k^n) V_{k,i}, i = 2, 3 \quad (2)$$

where Q^m and Q^n belong to any two different models and the weight ω is used to guide the morphing between the two shapes. With the decrement of the proportion one model occupied, the model generated tends to be more like the other. Model size and orientation should be unified firstly. For convenience, we adjust the model so that their boundary is a unit cube and the orientation is along the z-axis. And then we can define a function whose dependent variable is the parameter ω and independent variable is the distance from the current facet to a defined plane, denoted by z-axis coordinate in this case.

As stated above, what we have now are a series of triangular facets. Each one is denoted by two vectors, not three vertices. Generally speaking, the number of the facets in a model is about twice as the number of vertices. So as to resolve the vertices, a

start vertex should be defined firstly. And then this problem can be described as the following form:

$$\min_{v_1, \dots, v_{N_v}} \sum_{k=1}^{N_f} \|(v_{k,i} - v_{k,1}) - V_{k,i}\|^2, i = 2, 3 \quad (3)$$

where N_v and N_f denote the number of the vertices and facets respectively. This overconstraint problem can also be solved by using the least squares method. During the process of the morphing, interpenetration between different parts sometimes occurs. Figure 1(a) (left) shows a situation where the ear goes into the head when a dog model transforms to a cat. Since dog ear is different from cat both in shape and orientation, merging these two parts may cause distortion if the proportion distribution is not reasonable. This problem can be solved by a refinement method introduced in DRAPE [12]. A penalty function is defined in the following form:

$$p(E) = \sum_{v_i \in E} \|\epsilon + n_{v_j} (v_i - v_j)\|^2 \quad (4)$$

where E is the set of vertices belonging to ear, v_j is the nearest vertex in the head to v_i and n_{v_j} is the normal of the vertex v_j . ϵ is a variable which ensures the ear vertices lie outside the head. To regularize the solution, two additional terms are added which named smooth warping and damping:

$$s(E) = \sum_{v_i \in E} \left\| (v_i - \widehat{v}_i) - \frac{1}{|N_i|} \sum_{j \in N_i} (v_i - v_j) \right\|^2 \quad (5)$$

$$d(E) = \sum_{v_i \in E} \|v_i - \widehat{v}_i\|^2 \quad (6)$$

where \widehat{v}_i is the vertices of the ear before transformation and N_i is the set of vertices adjacent to v_i . Then our object function can be defined as:

$$\min_{v_i \in E} p(E) + \omega_s s(E) + \omega_d d(E) \quad (7)$$

This equation can be solved using the least squares method iteratively. Figure 1(a) shows the process where the ear goes out of the head gradually.

The essence of this problem is that ears of dog and horse are not in the same posture. If we treat the ear as a rigid part, a rotation matrix R can be computed. By using this matrix, the dog ear can be rotated to the same orientation as the horse ear. And then shape deformation matrices Q_k can be computed. By this way, pose and shape morphing can be separated. The generated triangle can be represented as the following formulation:

$$\widehat{V}_{k,i} = \text{Rotate} \left((\omega Q_k^m + (1 - \omega) Q_k^n) V_{k,i}, q(\omega) \right), i = 2, 3 \quad (8)$$

where $q(\omega)$ is a quaternion used to represent the rotation and Rotate is a function defined to rotate the vector by the quaternion. The rotation matrix R can be converted to a quaternion qi , and $q(\omega)$ is a function of ω , interpolated between qi and $[1, 0, 0, 0]$. Since quaternion is suitable for rotation interpolation, $q(\omega)$ can be computed easily by linear

interpolation or spherical linear interpolation. As is illustrated in Figure 1 (b), pose and shape of the ear are merged separately.

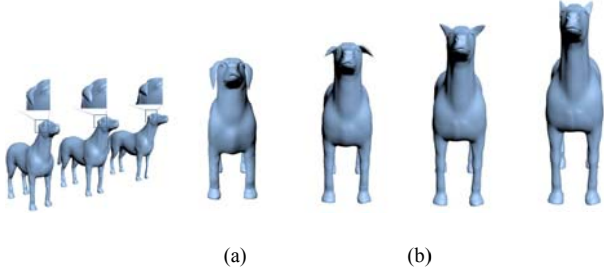


FIGURE 1. (A): MORPHING PROCESS OF DOG EAR; (B): TRANSFORMING A DOG TO A HORSE.

A. Shape Merging

Our method can merge any number of models, but for convenience we take the merging of two models for example. Figure 2 (a) shows a merged shape whose head is from a horse and other parts are from a cat. To get such a model, we need to determine firstly which model every part should belong to. Then conjunction between different parts should be smoothed. As the new model is generated by using the least squares method, this problem has been eased to a certain extent, but what we want is a smooth model. To address this defect, the mutation at the conjunction can not be allowed. So we apply a method to guide this smooth transition. As is shown in Figure 2(c), we firstly get the user-defined conjunction by labeling an area on the surface mesh manually. Then boundary of this area can be computed easily since these triangular facets are shared by two different parts. After that, the Dijkstra algorithm is applied to compute the geodesic distance from every vertex in the labeled area to the boundary. Figure 2(d) shows the result by labeling gradient colors on the mesh to represent the distances. A linear function is used to determine the proportions the two models account for respectively based on the distances from the current facets to the boundary.

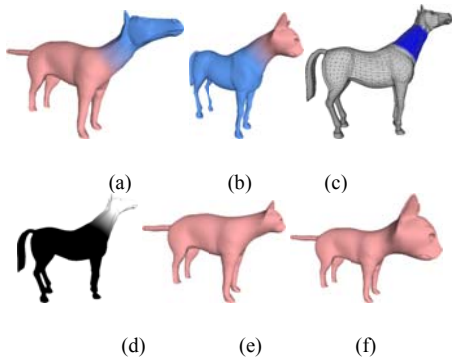


FIGURE 2. (A, B): PARTS MERGING; (c, d): CONJUNCTION SMOOTH METHOD; (E, F): SIZE DEFORMATION.

Size deformation is also included in our work. In Figure 2 (e, f), the cat with a small head or a big head is shown.

B. Consensus Correspondence

Our work above is based on the unified models which have point-to-point correspondences. But the models we have in hand do not have this characteristic. So a correspondence

system introduced by [6, 7] has been taken. It is just like to wrap a source surface onto the target. We choose the horse model as the source mesh which has N vertices, $|T|$ facets and the other models as targets. To compute the deformed vertices $\widehat{v}_1, \dots, \widehat{v}_N$, a set of affine transformations T_i , $i \in [1 \dots |T|]$ are defined to minimize the object function:

$$E = \omega_S E_S + \omega_I E_I + \omega_C E_C + \omega_M E_M \quad (9)$$

where $E_S = \sum_{i=1}^{|T|} \sum_{j \in \text{adj}(i)} \|T_i - T_j\|^2$, $E_I = \sum_{i=1}^{|T|} \|T_i - I\|_F^2$, $E_C = \sum_{i=1}^N \|v_i - c_i\|^2$, $E_M = \sum_{k=1}^m \|\widehat{v}_{s_k} - m_k\|^2$. E_S is defined to achieve a smooth deformation. This regularization can ensure the transformations for the adjacent facets to be similar when the source mesh transforms to the target. E_I , deformation identity, is defined to make the transformations similar to a identity matrix. This term can prevent the drastic change in the mesh caused by the deformation smoothness term. E_C is defined to represent the distance between each vertex in the source and the closest vertex in the target. To avoid the local minima, a marker error E_M is introduced to guide the deformation. Since corresponding vertices should be found correctly in E_C , we need to make the two models in a similar shape and position firstly. To achieve this purpose, sparse markers need to be added onto the models manually. s_k is the index for marker k on the source mesh and m_k is the corresponding marker on the target mesh.

An iterative approach is used to minimize this energy function. In the first iteration, we ignore the distance error between closest vertices by setting the weights $\omega_S = 1$, $\omega_C = 0$, $\omega_M = 10$. The marker error is the dominant constraint to guide the deformation in this phase. After this phase, the markers in the source mesh are transformed to the position of the corresponding markers in the target mesh. And the other vertices move to the target mesh restricted by the smoothness error E_S . Then a set of closest vertices can be computed using this model and the target. In the second phase, the optimization problem is solved by increasing the ω_C from 1 to 50 in four steps preserving $\omega_S = 1$, $\omega_M = 1$. After each step, a new model which is more like the target will be generated. And we update the closest points for the deformed source mesh from the target mesh. If the normals of the two corresponding vertices are less than 90° in orientation, this pair is valid.

IV. APPLICATION

In this section, we will introduce two applications based on our work above. One is to quantifying the differences between different animals, the other is to replace a character in a 2D image.

A. Differences Quantifying

While people can distinguish different animals easily, it is always difficult for a computer to tell the differences between different animals. Inspired by the work of SCAPE, each model can be represented by a column vector β_i . Since we have obtained the transformation matrices Q_k^i for each instance i and triangular facets k , a simple linear subspace which is used to generate Q^i can be estimated by using PCA:

$$Q^i = \varphi_{U,\mu}(\beta^i) = \overline{U\beta^i + \mu} \quad (10)$$

where μ is the mean value of the matrix composed of Q_i and U are the first n eigenvectors computed by using PCA.

Unlike SCAPE, what we want to do is not to learn the shape deformation model, but to get a set of low-dimensional parameters to represent our instances. Given these low-dimensional parameters, differences between any two animals can be quantifying. By computing the Euclidean distance between any two sets of the N dimensional parameters, the difference can be denoted as:

$$D_{i,j} = \sqrt{\sum_{k=1}^N (\beta_k^i - \beta_k^j)^2} \quad (11)$$

where N is the dimension of β . Figure 3 (a) shows the distances from every instance to the horse model assuming each β have 5 degrees of freedom while (b) is a graph which draw the β of seven models in the 3D space. And these animals can be divided into four groups intuitively: Horse and camel, Cat, lion and wolf, Dog and Bear.

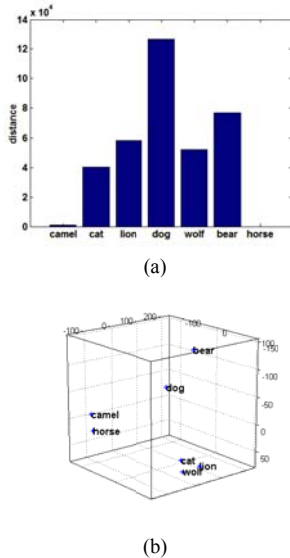


FIGURE III. DIFFERENCES QUANTIFYING.

B. Character Replacement

Assuming there is a human in a 2D image and we want to replace it with another character which is composed of a human body and a cat head, what we need to do is to project such a model onto the image. Firstly, we need to get a model whose pose and shape are the same as the character in the image. Pose estimation can be done accurately by locating the projections of joints of the model on the image [13, 14]. We use the method similar to [13] by labeling a series of markers on the image, the positions of which are the joints of the corresponding 3D model, the pose can be obtained by using the scaled orthographic projection method assuming the size of every rigid part is known.

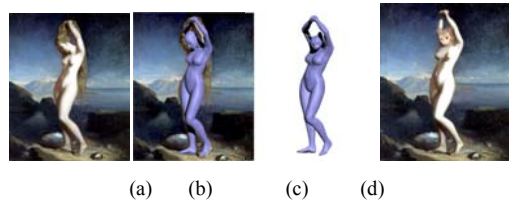


FIGURE IV. CHARACTER REPLACEMENT.



FIGURE V. MORPHING FROM A HUMAN HEAD TO A CAT HEAD.

Then our 3D morphing method can be used to get a merged shape. Figure 4 shows the result: a shape composed of a human body and a cat head. The human model comes from Poser. The trunk and limbs are completely from the human model, while the head is completely from the cat. The process of morphing is shown in Figure 5.

V. SUMMARY

In this paper, we have introduced a method to do the 3D morphing for any number of unified models by using the transformation parameters between them. Rules for this morphing can be defined by users regardless you want to merge the models integrally or separately. Differences quantifying and character replacement are two major applications of our work.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Foundation of China under Grant No. 61271231 and Jiangsu Natural Science Foundation under Grant No. BK2011337.

REFERENCES

- [1] M. Alexa, Recent advances in mesh morphing. *COMPUTER GRAPHICS Forum* 200, 3, 1–23. (2002)
- [2] T. Athanasiadis, I. Fudos, C. Nikou, and V. Stamati, Feature-based 3D morphing based on geometrically constrained sphere mapping optimization. *Proceedings of the 2010 ACM Symposium on Applied Computing*, ACM, 1258-1265.
- [3] V. Kraevoy, A. Sheffer, Cross-parameterization and compatible remeshing of 3D models. *ACM Transactions on Graphics* 23, 3, 861-869. (2004)
- [4] V. Kraevoy, A. Sheffer and C. Gotsman, Matchmaker: constructing constrained texture maps. *ACM Transactions on Graphics* 22, 3, 326-333. (2003)
- [5] A.M. Bronstein, M. M. Bronstein and R. Kimmel, Topology-invariant similarity of nonrigid shapes. *International journal of computer vision* 81, 3, 281-301. (2009)
- [6] B. Allen and B. Curless and Z. Popović, The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Transactions on Graphics* 21, 3, 587–594. (2003)
- [7] R. W. Sumner and J. Popović, Deformation Transfer for Triangle Meshes. *Proceedings of ACM SIGGRAPH 2004* 23, 3, 399-305.
- [8] O. Van Kaick, H. Zhang, G. Hamarneh and D. Cohenor, A survey on shape correspondence. In *Computer Graphics Forum*, vol. 30, Wiley Online Library, 1681–1707. (2011)
- [9] D. Hahnel, S. Thrun and W. Burgard, An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In *Proc. IJCAI, Acapulco, Mexico* (2003)

- [10] J. Sümuth, M. Winter and G. Greiner, Reconstructing Animated Meshes from Time-Varying Point Clouds. In Computer Graphics Forum, vol. 27, Wiley Online Library, 1469–1476.
- [11] D. Anguilov, P. Srinivasan, D. Koller, S. Thrun and J. Rodgers, SCAPE: Shape completion and animation of people. ACM Transactions on Graphics 24, 4, 408-416. (2005)
- [12] P. Guan, L. Reiss, D. A. Hirshberg, A. Weiss and M. J. Black, DRAPE: DRessing Any Person. ACM Transations on Graphics 31, 4 (2012)
- [13] C. J. Taylor, Reconstruction of articulated objects from point correspondences in a single uncalibrated image. CVIU 80, 10, 349–363. (2000)
- [14] A. Hornung, E. Dekkers and L. Kobbelt, Character animation from 2D pictures and 3D motion data. ACM Transactions on Graphics (TOG) 26, 1, 1 (2007)