# 3D Morphing for Triangle Meshes using Deformation Matrix

Tianwei Xing, Yipin Yang, Yao Yu, Yu Zhou, Xianglei Xing, Sidan Du
NanJing University, School of Electronic Science and Engineering
JiangSu Province, China
tianweix163@gmail.com

## 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 two-dimensional 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, correspondence, difference quantifying, character replacement, model driven

## 1 INTRODUCTION

3D morphing plays an important role in many applications of computer graphics such as 3D animation and games. Smooth morphing from one shape to another is always one of the most popular special effects in movies, e.g. sphinx, an animal with a lion body and a human face. Existing methods for 3D morphing can be summarized into two categories: volume-based and mesh-based. Since volume-based method is computationally expensive compared to mesh-based, much of work has been done for the latter. First stage of any morphing algorithm is the establishment of an accurate mapping between the source and the target shape. Then suitable forms for the morphing are used to get a merged result. Spherical embedding is a popular method in this area, but it has an inevitable disadvantage: significant increase of the number of mesh vertices and facets.

In this paper, an efficient 3D morphing method is introduced. In order to get models with point-to-point correspondences, we use a semi-automatic method which is a variant of iterated closet point algorithm. With these unified models, we can compute the transformation matrices from the source model to the target, using the method similar to SCAPE [Ang05a]. Given the transformation and a source model, 3D morphing can be done smoothly. As the target model accounts for an increasingly proportion, the model generated gradually becomes the target from the source. Merging different parts which belong to different models is also included in our work, e.g. a shape which consists of a human body and a cat head. Geodesic distance is used to determine the proportions the two models occupy respectively in order to smooth the conjunction.

Taking the inspiration of SCAPE, we try to analyze the transformation parameters to get a mathematical model and quantify the similarity between different shapes. Euclidean distance of N-Dimensional parameters reduced by Principal Component Analysis (PCA) between any two models represents the shape difference. The other two applications of our method are character replacement in two-dimensional image (Figure 10) and driving a merged model by a motion sequence (Figure 13), (demonstrated in our accompanying video).

## 2 RELATED WORK

A variety of related work has been done in this area about the morphing of 3D models. Alexa [Ale02a] summarize the work about mesh morphing before 2003.

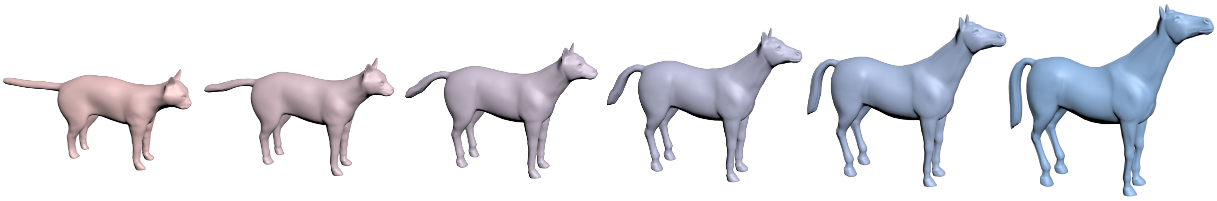


Figure 1: 3D morphing from a cat to a horse.

One popular approach in his summary is sphere embedding. By embedding the polyhedron models on a unit sphere and aligning the features, users can reconstruct a merged shape. Mocanu *et al.* [Moc12a] also use the similar method. Unlike traditional morphing method, they introduced a technique named pseudo metamesh construction to avoid significantly increasing the number of mesh triangles. Kraevoy *et al.* [Kra04a] propose another method by constructing a common base domain based on his another work, Matchmaker [Kra03a]. Given the common base domains, after initial cross-parameterization and compatible remeshing, a final new mesh can be generated by smoothing and refinement. Assuming the models have point-to-point correspondences, Allen *et al.* [All03a] do the morphing between two models by taking linear combination of the vertices. Although our models after processing have this correspondence, this method does not suit us. Different human models have similar scale, while different animals may have a huge difference in shape and size. Point combining directly ignores the difference in scale. In order to solve this problem, we take an inspiration in the work of deformation transfer by Sumner and Popović [Sum04a]. They use a $3 \times 3$ 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.

Our initial triangulated models do not have the same vertices and faces, not to mention the point-to-point correspondences. Getting the models with the point-to-point correspondences is just like to warp the template mesh onto the targets. Kaick *et al.* [Van11a] 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 [Hah03a]. However, this method is not stable and good results are yielded only when the inter-frame deformations are small. Coherent Point Drift (CPD) [Myr10a] is another approach to solve the point set registration. During the process of the point set registration, the template points drift to the target as a whole with certain transform form, maintaining the consistency of topology. Although CPD algorithm has some advantages, its inherent limitations can not be avoided, such as local optimum and slow convergence speed. Allen *et al.* [All03a] do this work with the optimization of a penalty function. By adding some markers on both the source and target meshes manually, optimizing the combining error of data, smoothness and marker can get a satisfying result. Based on his work, Sumner [Sum04a] has made some improvements by adding an identity error to the objective function to avoid drastic change in the shape of the model caused by the smoothness error.

Principal Component Analysis (PCA) has been used to analyze the features of human shape in Allen's work [All03a]. But it is performed over the coordinates of the template vertices. In the work of SCAPE, Anguelov [Ang05a] tried to perform PCA over the transformation matrices between the template shape and the other models. We use PCA like SCAPE, but what we want to do is not to generate a new model using the reduced dimensional parameters. Quantifying the difference between different models is our purpose.

## 3 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. Our model of 3d morphing adopts another method. Assuming the meshes have point-to-point correspondences, affine transformation matrices between the paired triangles can be computed like SCAPE. Different combinations of the transformation parameters can build different models. 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. However, animal models do not have this feature. Different shapes have different vertices and facets. So as to get the consensus meshes, we take the approach similar to deformation transfer.

## 3.1 Definitions

The terms used in our method are defined in the following way:

Our datasets consist of a source model S and a set of targets T = $\{T^1,\ldots,T^N\}$. The source mesh S = $\{V_S, F_S\}$ has a set of vertices $V_S = \{v_1,\ldots,v_j\}$ and a set of triangular facets $F_S = \{f_1,\ldots,f_k\}$.

## 3.2 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 $\tilde{f_k}$ have the vertices $v_{k,1}, v_{k,2}, v_{k,3}$ and $v_{\tilde{k},1}, v_{\tilde{k},2}, v_{\tilde{k},3}$. Since we just want to use a affine transformation defined by a $3 \times 3$ matrix $\mathbf{Q}$, not using a displacement vector $\mathbf{d}$, deformations are applied in the local coordinate system. We represent the deformation as $\tilde{\mathbf{V}} = \mathbf{Q} \times \mathbf{V}$ where

$$\mathbf{V} = [v_2 - v_1, v_3 - v_1], \tilde{\mathbf{V}} = [\tilde{v_2} - \tilde{v_1}, \tilde{v_3} - \tilde{v_1}] \qquad (1)$$

Since this is not enough to determine the affine transformation matrices, we add a regularization to constrain the problem following Allen *et al.* [All03a] and Sumner *et al.* [Sum04a]. With a smoothness constraint which indicates the transforms should be similar in adjacent triangular facets, this problem can be described as the following form

$$\min_{Q_1,\ldots,Q_N} \sum_{k=1}^{N} \sum_{i=2,3} \|Q_k V_{k,i} - \tilde{V_{k,i}}\|^2 + w_s \sum_{k=1}^{N} \sum_{j \in adj(k)} \|Q_k - Q_j\|^2 \qquad (2)$$

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 or separately for each rigid part by using the least squares method.

## 3.3 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

$$\tilde{V_{k,i}} = (\omega Q_k^m + (1-\omega) Q_k^n) V_{k,i}, i = 2,3 \qquad (3)$$

where $Q^m$ and $Q^n$ belong to any two different models and the weight $\omega$ 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.

The parameter $\omega$ above is a constant, but it can also be a variable. Rules can be defined by users to guide our shape merging. 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 $\omega$ and independent variable is the distance from the current facet to a defined plane, denoted by z-axis coordinate in this case. Several forms have been used to determine the weight, such as linear, piecewise, exponential and so on. Some of the results are enumerated in the Figure 2. The color represent the proportions each model accounts for.

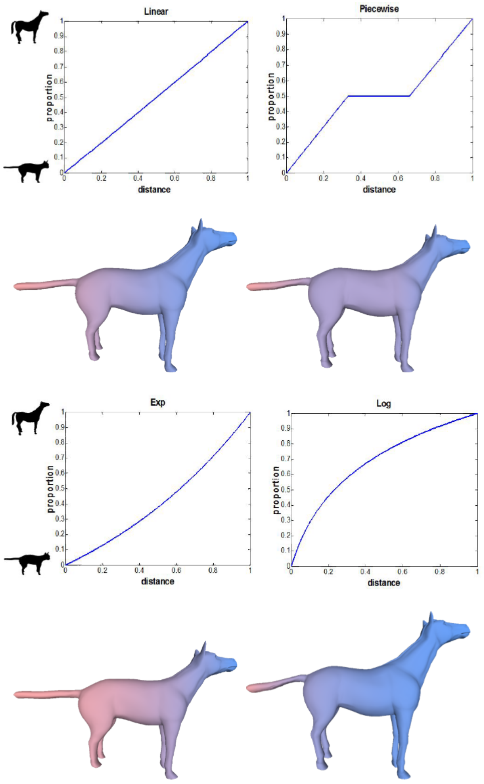


Figure 2: Several forms have been used to determine the weight: linear, piecewise, exponential, logarithmic. Corresponding models generated are below the four functions.

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 resolved the vertices, a start vertex should be defined firstly. And then this problem can be described as the following form

$$\min_{v_1,\ldots,v_{N_v}} \sum_{k=1}^{N_f} \|(v_{k,i} - v_{k,1}) - V_{k,i}\|^2, i = 2,3 \qquad (4)$$

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 3(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 [Gua12a]. A penalty function is defined in the following form:

$$p(E) = \sum_{v_i \in E} \|\varepsilon + n_{v_j}(v_i - v_j)\|^2 \tag{5}$$

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 $v_j$. $\varepsilon$ 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:

$$s(E) = \sum_{v_i \in E} \|(v_i - \tilde{v_i}) - \frac{1}{|N_i|} \sum_{j \in N_i} (v_j - \tilde{v_j})\|^2 \tag{6}$$

$$d(E) = \sum_{v_i \in E} \|v_i - \tilde{v_i}\|^2 \tag{7}$$

where $\tilde{v_i}$ 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

$$\min_{v_i \in E} p(E) + \omega_s s(E) + \omega_d d(E) \tag{8}$$

This equation can be solved using the least squares method iteratively. Figure 3 shows the process where the ear goes out of the head gradually.

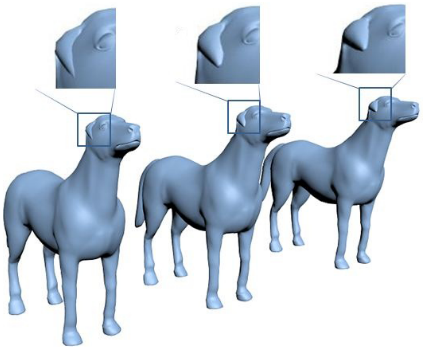


Figure 3: From left to right, this figure shows the initial morphing, the result after the first iteration of optimization, and the final result respectively.

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:

$$\tilde{V_{k,i}} = Rotate((\omega Q_k^m + (1-\omega) Q_k^n) V_{k,i}, q(\omega)), i = 2,3 \tag{9}$$

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 $\omega$, 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 4, pose and shape of the ear are merged separately.

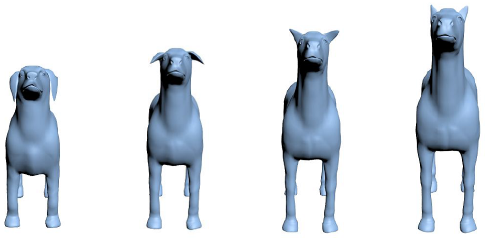
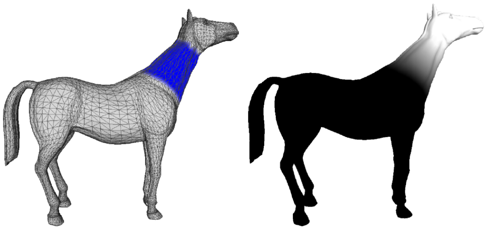


Figure 4: Transforming a dog to a horse while pose and shape of the ear are merged separately.

## 3.4 Shape merging

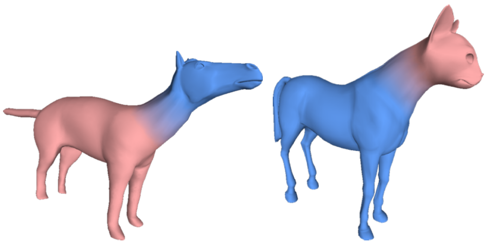
Assuming the unified models have been divided into several parts, merging different parts which belong to different models can be done smoothly, e.g. a shape which consists of a cat head, a horse trunk and four lion limbs. Using the vertices directly is not suitable for shape merging since it may cause huge distortion in the conjunction between different parts while using facets can solve this problem reasonably. Our method can merge any number of models, but for convenience we take the merging of two models for example. Figure 6 (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 5 (a), 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 5 (b) 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.

Size deformation is also included in our work. In Figure 7, the cat with a small head or a big head is shown.



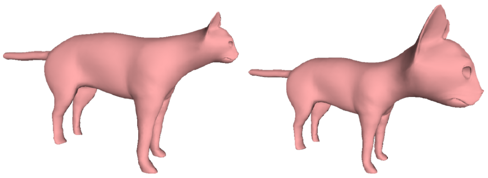
(a) (b)

Figure 5: Conjunction smooth method: (a) a horse mesh and the blue area is selected; (b) black and white represent body and head respectively, while the gradient colors represent the geodesic distance from every vertex in the labeled area to the boundary.



(a) (b)

Figure 6: Parts merging: (a) a model consists of a cat body and a horse head; (b) a model consists of a horse body and a cat head.



(a) (b)

Figure 7: Size deformation: the cat head is 0.5 times (a) and 2 times (b) of the original size respectively.

### 3.5 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 Allen *et al.* [All03a] and perfected by Sumner *et al.* [Sum04a] 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 $\tilde{v_1}, \ldots, \tilde{v_N}$, a set of affine transformations $T_i, i \in [1 \ldots |T|]$ are defined to minimize the object function:

$$E = w_S E_S + w_I E_I + w_C E_C + w_M E_M \qquad (10)$$

where

$$E_S = \sum_{i=1}^{|T|} \sum_{j \in adj(i)} \|T_i - T_j\|^2 \qquad (11)$$

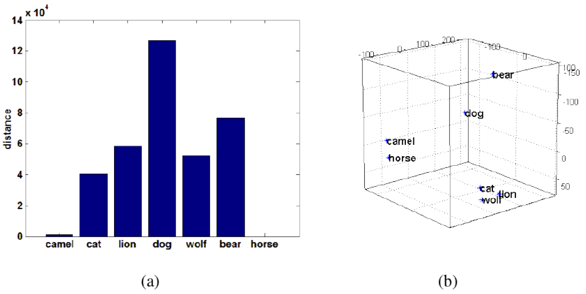
$$E_I = \sum_{i=1}^{|T|} \|T_i - I\|_F^2 \qquad (12)$$

$$E_C = \sum_{i=1}^{N} \|v_i - c_i\|^2 \qquad (13)$$

$$E_M = \sum_{i=1}^{m} \|\tilde{v_{s_k}} - m_k\|^2 \qquad (14)$$

$E_S$ 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 $w_S = 1, w_C = 0, w_M = 10$. 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 $w_C$ from 1 to 50 in four steps preserving $w_S = 1, w_M = 1$. 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.



(a) (b)

Figure 8: Differences Quantifying: (a) the distances from every instance to the horse model assuming each $\beta$ have 5 degrees of freedom; (b) the $\beta$ of seven models in the three-dimensional space.

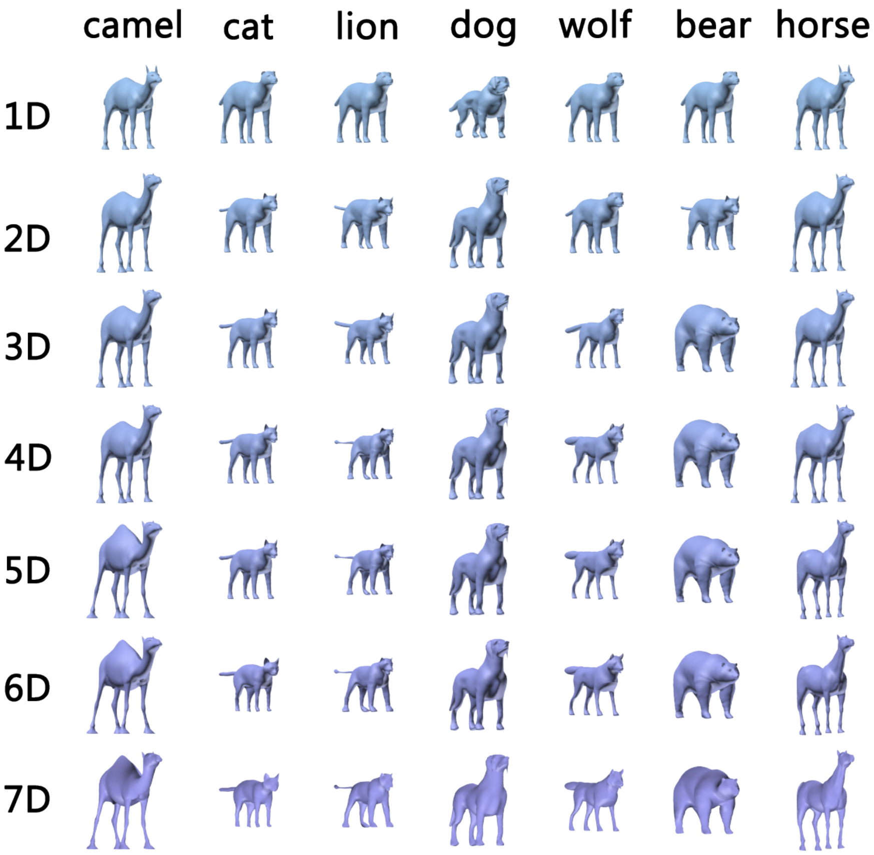


Figure 9: Model generated by using reduced N-Dimensional $\beta$.

# 4 APPLICATION

In this section, we will introduce three applications based on our work above. One is to quantifying the differences between different animals, and the second is to replace a character in a two-dimensional image with another deformed model. The last one is using a video sequence to drive a merged model.

## 4.1 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 $\beta_i$. 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:

$$Q^i = \varphi_{U,\mu}(\beta^i) = \overline{U\beta^i + \mu} \tag{15}$$

where $\mu$ 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:

$$D_{i,j} = \sqrt{\sum_{k=1}^{N} (\beta_k^i - \beta_k^j)^2} \tag{16}$$

where $N$ is the dimension of $\beta$. Figure 8 (a) shows the distances from every instance to the horse model assuming each $\beta$ have 5 degrees of freedom while (b) is a graph which draw the $\beta$ of seven models in the three-dimensional space. And these animals can be divided into four groups intuitively:

1. horse and camel
2. cat, lion and wolf
3. dog
4. bear

Figure 9 enumerates 7 × 7 models and each row is reconstructed by using the reduced N-Dimensional $\beta$. With the dimension of $\beta$ continues to reduce, the details of the models generated are lessened gradually. It is clearly that bear undergoes dramatic changes when the dimension of $\beta$ is reduced to two.

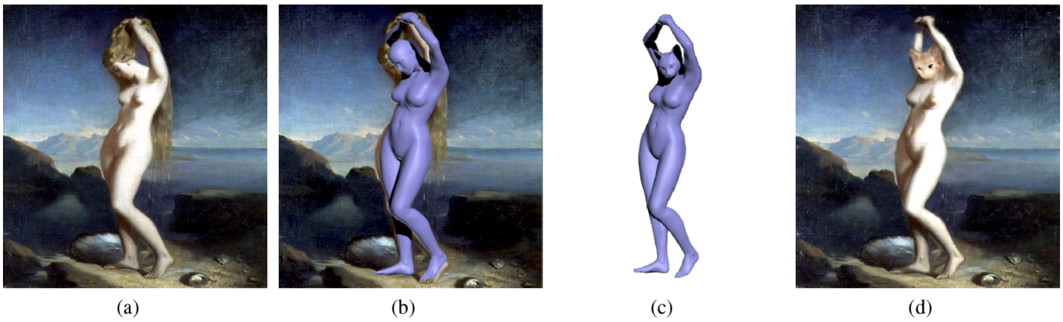

(a) (b) (c) (d)

Figure 10: Character replacement: (a) an image of venus; (b) a corresponding model; (c) a merged model by replacing the human head with a cat head; (d) transfer texture of the image to the merged model and then project the model to the image.

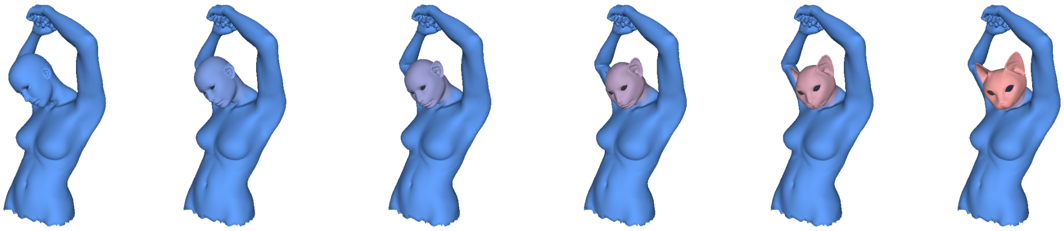


Figure 11: Morphing from a human head to a cat head.

## 4.2 Character Replacement

Character replacement is another application of our work. Assuming there is a human in a two-dimensional 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 [Tay00a] [Hor07a] [Gua09a]. We use the method similar to Taylar: 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. Shape estimation is relatively difficult and since this is not our focus, we do not pay much attention to this phase. The shape of the model is adjusted manually.

Then our 3D morphing method can be used to get a merged shape. Figure 10 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 11.

## 4.3 Model Driven

In this section, we try to drive our merged model to move according to a video sequence. The sequence is composed of a series of depth images obtained from Kinect. Since SCAPE is our basic work, a approximate model can be obtained from the work of shape completion [Ang05a]. The only difference is that we use CPD algorithm to find correspondences between incomplete meshes and SCAPE models while shape completion has a set of markers between the two meshes. Since the incomplete meshes collected from Kinect are not very accurate, sometimes the results from shape completion converge to a local optimum. So after shape completion, we examine if the result is correct or not

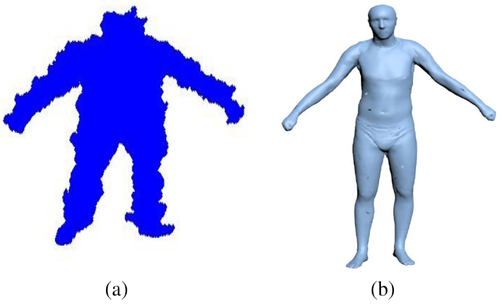

(a) (b)

Figure 12: Point cloud (a) from a depth image and the corresponding complete mesh (b) from SCAPE.

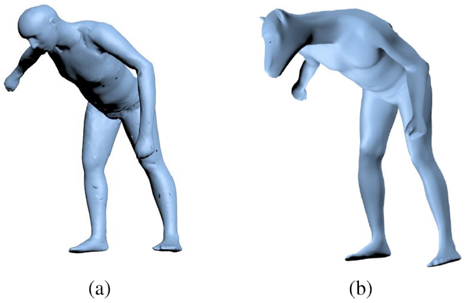

(a) (b)

Figure 13: Transfer deformation of a SCAPE model (a) to our merged model (b).

by computing the distance between the two meshes. If the distance is over a threshold, we think this method collapse. And then we use an approach introduced by Wei *et al.* [Wei12a]. By combining pose tracking and pose detection, an more accurate model can be obtained. Figure 12 shows a set of point cloud (a) from depth image and a generated model (b) corresponding to the point cloud.

Since all our models have point-to-point correspondences, it is quite easy to do the work of deformation transfer without any modification. Figure 13 shows the result of our work.

## 5 CONCLUSION

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. Compared to the work of Kraevoy *et al.* [Kra04a], one significant superiority of ours is that we can use a video sequence to drive the merged model to be in the same pose. Since our method is based on the unified models, deformation transfer can be done without any modification.

Unified models should be obtained firstly before the 3D morphing. This is a difficult task in our work. Since different models may differ from each other greatly in some parts, it is hard to warp the source mesh onto the target perfectly. To address this problem, we have tried some methods. One is to add the number of vertices and facets by interpolation and this indeed plays a role to a certain extent. Another is to label more markers on the surface mesh but this is not a good choice. Fortunately, when we do the morphing only for several parts, not integrally, we do not need to warp the whole source onto the target, but only the parts we want to merge. For example, if the model generated is composed of a human body and a cat head, we need only to warp the human head onto the cat head and not to care the body.

During the process of the morphing, interpenetration occasionally occurs between different parts. Although this problem can be addressed by the method introduced in DRAPE, it still needs users to find the interpenetrating area manually. To address this problem fundamentally, we introduce a method to separate the pose and shape morphing. Since the models we have in hand are nearly in the same pose, this situation only occurs in special parts of the body, e.g. the dog and horse ears. Another limitation of our work is that the models for the morphing should have the similar orientations, and their size can not differ too much from each other. Orientation is defined as the direction from the tail to the head and size is normalized. The models should be adjusted according to this rule before the morphing.

## 6 REFERENCES

[Ale02a] Alexa M. Recent Advances in Mesh Morphing. COMPUTER GRAPHICS Forum, pp.1-23, 2002.

[All03a] Allen B., Curless B. and Popović Z. The space of human body shapes: Reconstruction and parameterization from range scans. ACM TOG, pp.587-594, 2003.

[Ang05a] Anguelov D., Srinivasan P., Koller D., Thrun S. and Rodgers J. SCAPE: Shape Completion and Animation of People. ACM TOG, pp.408-416, 2005.

[Gua09a] Guan P., Alexander W., Alexandru O. B. and Michael J. B. Estimating human shape and pose from a single image. ICCV, pp.1381-1388, 2009.

[Gua12a] Guan P., Reiss L., Hirshberg D. A., Weiss A. and Black M. J. DRAPE: DRessing Any PErson. ACM TOG, 2012.

[Hah03a] Hahnel D., Thrun S., and Burgard W. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. IJCAI, 2003.

[Hor07a] Hornung A., Dekkers E. and Kobbelt L. Character animation from 2D pictures and 3D motion data. ACM TOG, pp.1, 2007.

[Kra03a] Kraevoy V., Sheffer A. and Gotsman C. Matchmaker: constructing constrained texture maps. ACM TOG, pp.326-333, 2003.

[Kra04a] Kraevoy V., Sheffer A. Cross-parameterization and compatible remeshing of 3D models. ACM TOG, pp.861-869, 2004.

[Moc12a] Mocanu B. and Zaharia T. A complete framework for 3D mesh morphing. SIGGRAPH, ACM, pp.161-170, 2012.

[Myr10a] Myronenko A., Song X.B. Point Set Registration: Coherent Point Drift. PAMI, pp.2262-2275, 2010.

[Sum04a] Sumner R. W., Popović J. Deformation Transfer for Triangle Meshes. SIGGRAPH, pp.399-405, 2004.

[Tay00a] Taylor C. J. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. CVIU, pp.349-363, 2000.

[Van11a] Van Kaick O., Zhang H., Hamarneh G. and Cohen-Or D. A survey on shape correspondence. Computer Graphics Forum, Wiley Online Library, pp.1681-1707, 2011.

[Wei12a] Wei X., Zhang P. and Chai J. Accurate Realtime Full-body Motion Capture Using a Single Depth Camera. ACM TOG, 2012.

[Zho10a] Zhou S., Fu H., Liu L., Cohen-Or D., and Han X. Parametric reshaping of human bodies in images. ACM TOG, pp.126, 2010.