

Character Transfer: Example-based individuality retargeting for facial animations

Takuya Kato
Waseda University
3-4-1 Okubo Shinjuku-ku
Japan, Tokyo
takuya_lbj@ruri.waseda.jp

Shunsuke Saito
Waseda University
3-4-1 Okubo Shinjuku-ku
Japan, Tokyo
shun-1616@moegi.waseda.jp

Masahide Kawai
Waseda University
3-4-1 Okubo Shinjuku-ku
Japan, Tokyo
doara-waseda@toki.waseda.jp

Tomoyori Iwao
Waseda University
3-4-1 Okubo Shinjuku-ku
Japan, Tokyo
sazabi@akane.waseda.jp

Akinobu Maejima
Waseda University
3-4-1 Okubo Shinjuku-ku
Japan, Tokyo
a.maejima@aoni.waseda.jp

Shigeo Morishima
Waseda Research Institute
for Science and Engineering
3-4-1 Okubo Shinjuku-ku
Japan, Tokyo
shigeo@.waseda.jp

ABSTRACT

A key disadvantage of blendshape animation is the labor-intensive task of sculpting blendshapes with individual expressions for each character. In this paper, we propose a novel system "Character Transfer", that automatically sculpts blendshapes with individual expressions by extracting them from training examples; this extraction creates a mapping that drives the sculpting process. Comparing our approach with the naïve method of transferring facial expressions from other characters, Character Transfer effectively sculpted blendshapes without the need to create such unnecessary blendshapes for other characters. Character Transfer is applicable even the training examples are limited to only a few number by using region segmentations of the face and the blending of the mappings.

Keywords

Facial animation, blendshape animation, individual expressions, blendshape modification, facial model segmentation, mapping blending

1. INTRODUCTION

Facial expressions of CG characters are playing greater roles in films and computer games. *Blendshape animation* is able to create arbitrary expressions for CG characters by linearly combining basis expressions called *blendshapes*. Therefore, this technique is applied to a wide variety of CG content to create realistic facial expressions. However, it is challenging for artists to parameterize the blending coefficients and sculpt blendshapes to realize blendshape animation.

Recent research has introduced sophisticated methods for automatically estimating the blending coefficients. Although these methods provided benefit to artists in creating rough blendshape animations efficiently, it often proved difficult to

create ideal expressions by only controlling the blending coefficients. As a result, artists have been required to sculpt an enormous number of high-quality blendshapes for many different scenes to achieve high-quality facial animations.

To define the problem more specifically, sculpting blendshapes with *individual expressions* remains a labor-intensive process since such expressions greatly differ depending on the target characters. Individual expressions are diverse facial expressions that characters have in addition to the semantics of the expressions. For example, the facial expressions of a laughing monkey and a laughing human slightly differ in geometry because each character moves its parts, such as mouth, eyes and others, differently. This is not the only difference between characters' expressions; expressions of characters of the same type (e.g. monkeys) also differ, because each individual often has his/her own distinct facial expressions. Hence it is time-consuming for artists to sculpt an enormous number of blendshapes with individual expressions when it comes to create blendshapes for many different individual characters.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

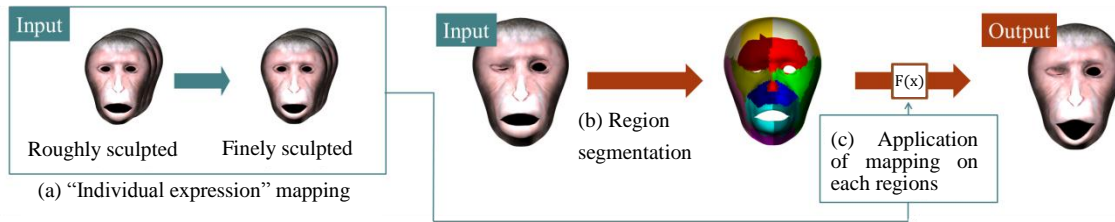


Figure 1: Overview of “Character Transfer” that (a) creates mappings using training examples. (b) Automatically segments the region, and (c) applies blended mappings onto each segment region.

In this paper, we introduce a method called “Character Transfer” that modifies the roughly created input blendshapes of arbitrary expressions to sculpt more specific individual expressions. Character Transfer sculpts the individualities by applying individual expressions extracted from a small number of training examples on segmented regions of a face.

There are three contributions in this method. One is that we introduced the method to define the individual expressions as mapping. This allows us to extract individuality quantitatively so that it can be applicable to the other facial expressions. Secondly we introduced a novel segmentation method that considers the geometry of the facial model and the facial expressions of the training examples. This allows the system to be effectively applicable even with the limited number of training examples. The third contribution is that we proposed a novel blending method of the individualities that avoids unnatural deformations caused by a naïve linear blending when applied on the input blendshape. By using this blending method it is possible to generate the individual expressions for arbitrary expressions. We show that Character Transfer can effectively modify the roughly created facial model with arbitrary expression with fully automatic algorithm other than sculpting training examples.

2. RELATED WORKS

Highly realistic facial animation using blendshape animation is a well-established approach and has been an active research discipline [ALE09]. Above all, facial retargeting using blendshape animation is efficiently applied for many facial animations of CG characters [BER85, CHU04]. Accordingly, many methods focus on capturing the facial expressions and estimating blending coefficients finely. Some methods are primarily focused on tracking facial expressions from facial feature points on two-dimensional video frames of a web camera [CAO13]. By combining data from a depth sensor and facial feature point on two-dimensional video frames, high quality facial tracking is achieved [LI13, BOU13].

These methods’ approaches realize to construct high-quality facial mesh models of the actor by fitting the

generic facial model onto the tracked data; however, since the user’s facial expressions do not geometrically equal the CG character’s facial expressions, individual expressions is not considered. To rig individual expressions into the facial animations, sculpting high-quality blendshapes is necessarily in preprocessing that matches its semantics to the actor’s expressions.

Other related works studies focused on sculpting blendshape and modifying blendshapes. Some methods sculpted facial models from photographs [PIG98] or 3D scan data [ZHA04, WEI09] where human’s data of these input is required. Other methods creating linear PCA model is proposed although it required many training example for PCA to sculpt facial model of basis expression [BLA99, BLA03, VLA05]. Above all, *Deformation Transfer* [SUM04] is a method that transfers the facial expressions from other characters; this approach has been a fundamental method for sculpting blendshapes. Although this method is applicable semi-automatically, this method does not consider the geometry or the facial expression of the target character. As a result, individual expressions are not rigged on the target character using this approach. Recent studies specific to the facial expressions proposed an approach to improve artifact yielded when transferring facial expressions of other characters via Deformation Transfer [SAI13]. Although this method is able to remove artifacts by adding a few virtual triangles between the eyelid and lips, it is not applicable to rig individual expressions because it only controls deformation of virtual triangles between eyelids in which the topology of the virtual triangles differs.

Some method for modifying existing blendshapes by using training examples of target characters to rig individual expressions has also been proposed. Method proposed in [CHO05, LIU08] is only applicable for generic predefined model using sparse motion capture data. One method that is effective for modification of blendshapes is proposed in [LI10]. In this method, the modification transfers respective features of individual expressions by solving reproductions of the training examples. Nonetheless, while the modification is effective for blendshapes that were particularly similar to the training examples,

the method is not effective when training examples were dissimilar. Accordingly, this method required a large numbers of training examples in order to appropriately modify arbitrary blendshapes.

Based on the above problems, we have three primary goals, summarize as follows:

- 1) To be able to successfully apply our method even though the number of training examples is limited.
- 2) To be able to apply our method to blendshapes that are roughly sculpted by an arbitrary method.
- 3) To be able to apply our method to blendshapes that are dissimilar to the given training examples.

Our first goal is straightforward and has certainly been the goal of much of the related research. Our second goal provides versatility and has precedent, for example, with the improved method noted above that is only applicable when using Deformation Transfer. We set our third goal since related research in modifying blendshapes is not applicable for expressions dissimilar to the training example. By achieving these goals, Character Transfer can be applicable for many CG character facial animations.

3. PROPOSED METHOD

3.1 Overview

As outlined in Figure 1, Character Transfer rigs individual expressions from training examples. For training examples, we provided the set of both poorly and finely expressive facial models. Using the idea of a gradient tensor, our system creates the mappings between deformations of every expression in the training example set, which include facial models with rest pose. Furthermore, our system defines these mappings as individual expressions. Finally, applying region segmentation to the facial model, Character Transfer generates appropriate individual expressions for each region by blending the mapping appropriately and applying them onto each region of the roughly created input blendshapes.

Apart from creating the sculpted training examples, all steps introduced in this section are fully automatic and applicable to characters of arbitrary geometries.

3.2 “Individual expression” mapping

For individual expression mapping, we first created mappings from training examples that define the individual expressions. From the roughly sculpted blendshapes, for example, generated via Deformation Transfer, the individual expressions can be defined by comparing expressions with the finely sculpted blendshapes. Therefore, we defined individual expressions as the differences between the poorly expressive blendshapes and the finely expressive blendshapes.

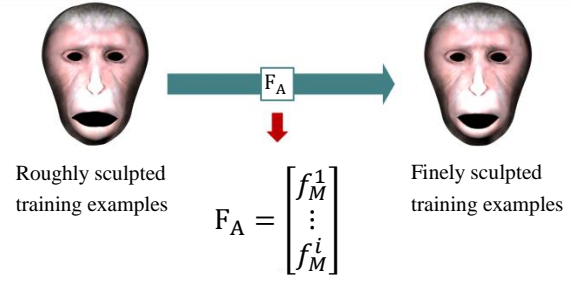


Figure 2. Creation of Individual expression mapping.

To define these differences, we adopted a deformation gradient to distinguish the deformation of the triangles. The deformation gradient for a single triangle is a 3×3 matrix describing the rotation and scaling required to go from a non-deformed state to a deformed state. The deformation gradient is computed by adding auxiliary vertex v_{j4} , which is computed by solving

$$v_{j4} = v_{j1} + \frac{(v_{j2} - v_{j1}) \times (v_{j3} - v_{j1})}{\sqrt{(v_{j2} - v_{j1}) \times (v_{j3} - v_{j1})}} \quad (1)$$

Here, $v_{jk} \{k = 1, 2, 3\}$ represent the vertices of the j^{th} triangle of the mesh model. Deformation gradient J_j is then computed by solving the linear system

$$J_j = V'_j V_j^{-1} \quad (2)$$

Here, V_j and V'_j are 3×3 matrices that contain non-deformed and deformed edge vectors of the j^{th} triangle, respectively; That is,

$$V_j = [v_{j2} - v_{j1} \quad v_{j3} - v_{j1} \quad v_{j4} - v_{j1}]$$

$$V'_j = [v'_{j2} - v'_{j1} \quad v'_{j3} - v'_{j1} \quad v'_{j4} - v'_{j1}]$$

In our system, we compute deformation gradients for finely and roughly sculpted blendshapes from facial models with rest pose. More specifically, let m be the facial expression of the training example, and s_m and t_m be the roughly and the finely sculpted training examples of expression m , respectively. For the i^{th} triangle, we create deformation gradients for facial models with rest pose from s_m and t_m to form $s_m^i \in \mathbb{R}^{3 \times 3}$, and $t_m^i \in \mathbb{R}^{3 \times 3}$, respectively, as

$$s_m^i = V'_{s_m^i} V_{s_m^i}^{-1} \quad (3)$$

$$t_m^i = V'_{t_m^i} V_{t_m^i}^{-1}$$

$$V_{s_m^i} = [v_{s_m^i 2} - v_{s_m^i 1} \quad v_{s_m^i 3} - v_{s_m^i 1} \quad v_{s_m^i 4} - v_{s_m^i 1}]$$

$$V_{t_m^i} = [v_{t_m^i 2} - v_{t_m^i 1} \quad v_{t_m^i 3} - v_{t_m^i 1} \quad v_{t_m^i 4} - v_{t_m^i 1}]$$

$$V'_{s_m^i} = [v_{s_m^i 2} - v_{s_m^i 1} \quad v_{s_m^i 3} - v_{s_m^i 1} \quad v_{s_m^i 4} - v_{s_m^i 1}]$$

Here, $v_{s_m^i k} \{k = 1, 2, 3, 4\}$ are the vertices of the roughly sculpted blendshape of the i^{th} triangle, and $v_{t_m^i k} \{k = 1, 2, 3, 4\}$ are the vertices of the finely sculpted blendshape of the i^{th} triangle. We next create a mapping between s_m^i and t_m^i . For each

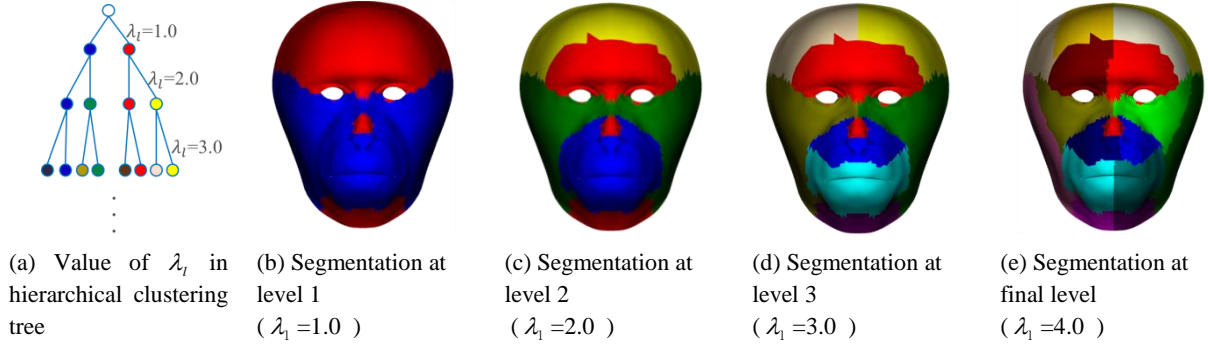


Figure 3. Hierarchical region segmentation.

i^{th} triangle of \mathbf{s}_m^i and \mathbf{t}_m^i , we incorporate mapping $\mathbf{f}_m^i \in \mathbb{R}^{3 \times 3}$, which combines \mathbf{s}_m^i and \mathbf{t}_m^i as follows:

$$\mathbf{f}_m^i = \mathbf{t}_m^i \mathbf{s}_m^{i-1} \quad (4)$$

We define this mapping for each triangle of every training example set. Therefore, each of the training examples sets have a mapping that is able to modify a given blendshape in a manner similar to that of the training example. Note that it is not required to create the roughly created input blendshapes of arbitrary expressions using specific technique: we only need to apply the selected technique to generate all training examples.

3.3 Hierarchical region segmentation

The mappings extracted from the training examples are applicable only to the expressions of the given training examples. Since each facial parts moves independently, the mapping that fits to the input facial expression is different depending on each facial part.

There are several methods to segment the facial model into regions, especially regarding facial retargeting. Some automatic segmentation methods have not considered the geometry of the facial model, for example, the method proposed by [JOS03].

Another proposed method is to use three-dimensional motion capture marker data on the human face to segment it into regions [TEN11]. This method is not preferable for our system because Character Transfer required no facial expressions data of human and did not consider the geometry of the facial model. Considering these problems, we propose a novel automatic region segmentation method which is effectively embedded for Character Transfer.

We first segment the facial model into triangular unit regions by considering the geometry of the facial model with rest pose and the expressions of the training examples via a hierarchical algorithm. Multidimensional vector $\mathbf{p}_i \in \mathbb{R}^{D \times 1}$ ($D = 3 \times 3 \times (1 + M)$) is defined for each i^{th} triangle in the target shape as follows.

$$\mathbf{p}_i = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{d}_{11}, \mathbf{d}_{12}, \mathbf{d}_{13}, \dots, \mathbf{d}_{m1}, \mathbf{d}_{m2}, \mathbf{d}_{m3})^T$$

Here, M is the number of training examples, $\mathbf{v}_i \in \mathbb{R}^{3 \times 1}$ is a position containing the $(x; y; z)$ spatial coordinates of the f^{th} ($f = 1, 2, 3$) vertex the i^{th} triangle of the facial model with rest pose, and $\mathbf{d}_{mf} \in \mathbb{R}^{3 \times 1}$ is the displacement vector from \mathbf{v}_f to the position of the f^{th} vertex in the i^{th} triangle in the m^{th} ($m = 1, 2, \dots, M$) training example. To segment the face of the target shape effectively, we recursively split polygons into two clusters. Multiplying \mathbf{p}_i by a weight vector \mathbf{e}_t , a center vector of cluster \mathbf{c}_t ($t = 1, 2$) can be found by minimizing the following:

$$\varphi_i = \min_{\mathbf{c}_t \in C} \sum_{\mathbf{p}_i \in P} \|\mathbf{e}_t \mathbf{p}_i - \mathbf{c}_t\| \quad (5)$$

$$P = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$$

$$C = \{\mathbf{c}_1, \mathbf{c}_2\}$$

$$\mathbf{e}_l = (\mathbf{a}_1^l, \mathbf{a}_2^l, \mathbf{a}_3^l, \mathbf{b}_1^l, \mathbf{b}_2^l, \mathbf{b}_3^l, \dots, \mathbf{b}_{M1}^l, \mathbf{b}_{M2}^l, \mathbf{b}_{M3}^l)$$

$$\begin{cases} \mathbf{a}_f^l = (1 - 1/\lambda_l^2) \mathbf{s} \\ \mathbf{b}_{fm}^l = 1/\lambda_l^2 \mathbf{s} \end{cases} \quad \mathbf{s} = (1, 1, 1)^T$$

Here, l is the number of recursive levels, and λ_l is a constant weight parameter independently controlled by the number of recursive cluster. As shown in Figure 3 (a), the value of λ_l corresponds to a level of hierarchical clustering tree. Changing λ_l for each level of the tree, the influence of the movements and geometry is modified as clustering progresses by formula (5). Figures 3 (b), (c) and (d) show the segmented regions in each level of this hierarchical clustering. In higher levels of the tree, the influence of geometry becomes strong and facial parts are segmented vertically and horizontally. Our technique realizes effective segmentation for asymmetrical controls in blendshapes such as the regions of the right, left, upper and lower eyelids. Our technique is effectively applicable to arbitrary models. In this paper, we have segmented the face model into 16 regions by our method as shown in Figure 3 (e).

3.4 Blending the mappings

A single mapping extracted from a training example is only able to modify the expression in a single way. To apply Character Transfer to arbitrary expressions from the limited number of training examples, it is necessary to create new mapping by blending the

individualities which each of the mapping have. We incorporate a method to generate new mappings for each segmented region of the blendshape by blending the elements of the mappings. By blending the mappings in which the training examples are geometrically similar to the input facial expression with estimated coefficients, Character Transfer is able to generate a mapping effective for the input facial expression.

To measure the similarity, we estimate the blending coefficients by first estimating the naïve blendshape coefficients for each region.

For the r^{th} region of each blendshape, blending coefficients $\mathbf{w}_r \in \mathbb{R}^{M \times 1}$ are computed by solving the following linear system

$$\mathbf{v}_r = \mathbf{B}_r \mathbf{w}_r \quad (6)$$

Here, $\mathbf{v}_r \in \mathbb{R}^{3n \times 1}$ is the vector that contains coordinate values of n vertices on the input blendshape and $\mathbf{B}_r \in \mathbb{R}^{3n \times M}$ is the matrix that contains coordinate values of n vertices on the M training examples. We compute the above equation is solving the following minimization problem below:

$$\mathbf{E}(\mathbf{w}_r) = \|\mathbf{v}_r - \mathbf{B}_r \mathbf{w}_r\| \quad (7)$$

More specifically, we independently compute the coefficients for each region by solving this equation for each region.

When blending the mappings, the most straightforward approach is to apply a linear blending of each element: however, linear blending induces the possibility of a triangle to collapsing or flipping since it controls both rotational and scaling essence at the same time as shown in Figure 4 (a). In Character Transfer we applied novel blending method to naturally blend the mapping with using interpolation method of two deformation gradient proposed by [KAJ12]. The interpolation method proposed is able to interpolate two deformation gradients by spherical linear interpolation of quaternion and exponential map of matrix. Let \mathbf{f}_m^i be the m^{th} mapping, our system applies polar decomposition [SHO92] to the mapping to decompose into rotation matrix $\mathbf{Rotf}_M^i \in \mathbb{R}^{3 \times 3}$ and positive definite symmetric matrix $\mathbf{Symf}_M^i \in \mathbb{R}^{3 \times 3}$. Next, we apply a different blending method to compute the matrix according to blending coefficients effectively. For the i^{th} triangle, we compute the rotation matrix according to blending coefficients by spherical linear interpolation of the quaternion. The interpolated rotation matrix can be computed by the degree of rotation according to the blending coefficient by solving the equation below:

$$\mathbf{Rot}_i = \prod_{m=0}^M \text{slerp}(q_1, q_{\mathbf{Rotf}_M^i}, w_{jm}) \quad (8)$$

Here, *slerp* is an operator which means that performs spherical linear interpolation of quaternions,

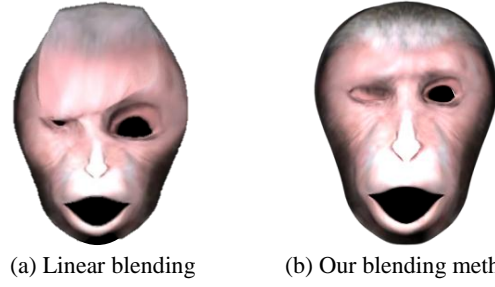


Figure 4. Illustrative comparison of linear blending and our blending method, our blending method successfully avoided unnatural deformations

q_1 and $q_{\mathbf{Rotf}_M^i}$ is the quaternion of identity matrix and \mathbf{Rotf}_M^i , and w_{jm} is the blending coefficient for region r with which the i^{th} triangle is affiliated. For a positive definite symmetrical matrix, our system uses a logarithm and an exponential map of the matrix. Using this approach, we interpolates positive definite symmetrical matrix according to the blending coefficient by solving the equation below:

$$\mathbf{Sym}_i = \exp(\sum_{m=0}^M w_{jm} \log(\mathbf{Symf}_M^i)) \quad (9)$$

By using a blended mapping for both rotation matrix and positive definite symmetric matrix, our system solves the equation below to generate the mapping matrix:

$$\mathbf{Blendf}^i = \mathbf{Rot}_i \cdot \mathbf{Sym}_i \quad (10)$$

This blending method is able to blend mapping with arbitrary blending coefficients while preserving property $\det(\mathbf{Blendf}^i) > 0$, i.e., the blended mapping does not unnaturally collapse or flip.

The definition of the mapping is that it modifies deformation gradient of roughly created input blendshapes by multiplying it with the ones of finely created blendshapes. Accordingly, we apply this blended mapping to the i^{th} deformation gradient defined by the facial model with rest pose and input facial expression $\mathbf{J}_i \in \mathbb{R}^{3 \times 3}$ as follows:

$$\mathbf{B} = [\mathbf{Blendf}^1 \times \mathbf{J}_1, \dots, \mathbf{Blendf}^i \times \mathbf{J}_i, \dots, \mathbf{Blendf}^N \times \mathbf{J}_N]^T$$

Finally, we solve equation (11) for x , y , and z coordinates for the vertices of the output blendshape sculpted by Character Transfer, $\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_N]^T \in \mathbb{R}^{N \times 3}$ as follows:

$$\min \|\mathbf{B} - \mathbf{A}\mathbf{X}'\| \quad (11)$$

Here, $\mathbf{A} \in \mathbb{R}^{3M \times N}$ is a large sparse matrix in which $\mathbf{A}\mathbf{X}' \in \mathbb{R}^{3M \times 3}$ is the deformation gradient defined between the facial model with rest pose and the blendshape sculpted by Deformation Transfer, which is proposed by [SUM05]. Equation (11) can be computed by solving the linear system below:

$$\mathbf{X}' = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (12)$$

By solving (12) to directly compute the coordinate value of the vertices without separate computations in regions, the computed positions of the vertices form the continuities across the segmented regions. Since our method solves minimization problem (11) in which the semantics of the minimization are almost the same equation solved in Deformation Transfer, Character Transfer can be incorporated into the framework of Deformation Transfer. This property allows other methods to be implementable naturally to the given equation, including the modification method proposed in [SAI13]. Furthermore, since Character Transfer is applicable via only a few training examples and roughly created input blendshapes, Character Transfer support the modification of the blendshape prior to other modification methods.

4. RESULTS

Figure 5 shows the blendshapes sculpted by Character Transfer in comparison with blendshapes sculpted by an artist and by Deformation Transfer. We sculpted the blendshapes of a monkey with only the front of the face, which consisted of 5K vertices. We created four training examples to modify the blendshape shown in Figure 6. For roughly created input blendshape of arbitrary expressions, we applied Deformation Transfer to generate such blendshapes by transferring expressions defined in blendshapes of humans. The correspondence of triangles between monkey and human models were defined by adopting the semi-automatic correspondence technique introduced in [SUM04]. The time required for all the steps to create one blendshape took approximately 13 s by using Intel Core™ i7-2600 CPU without parallelization. Character Transfer required no manual parameter settings. However, the blending coefficient can be manually set if the user wishes to have such control. We also created facial animations as shown in supplemental video using the blendshapes sculpted via Character Transfer and compared them to those sculpted via Deformation Transfer. The facial animation videos consist of random facial movements created from 14 blendshapes for 270 frames. Comparing our blendshapes with ones sculpted by an artist with those sculpted by Deformation Transfer, our blendshapes are considered more similar to those sculpted by the artist.

One of major causes of the artifacts yielded by naïve Deformation Transfer is the differences of how each characters move their facial parts. Some of these artifacts can be seen around lips in Figure 5 because of the size difference of monkey's lips and those of human. As a result, the unnatural lip movements were observed; however, since our training examples

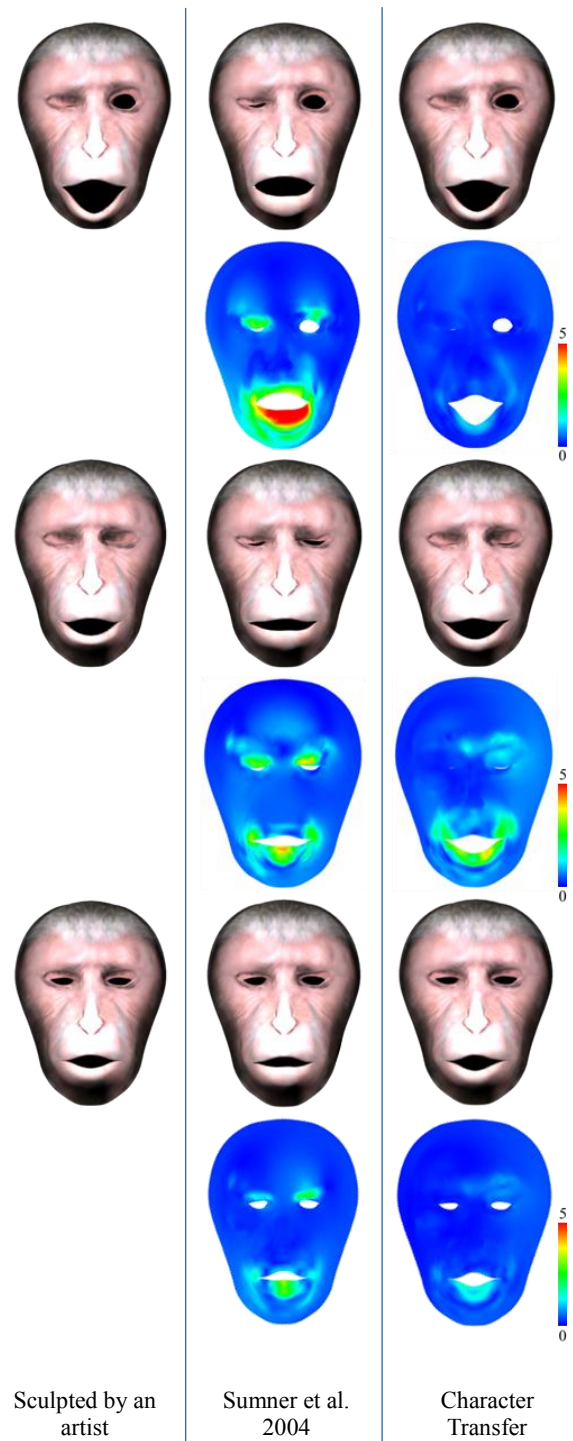


Figure 5. Comparison of blendshapes sculpted by an artist. Versus our approach and that of Sumner et al.; note that the color maps show the vertex errors as compared with the blendshapes sculpted by an artist.

have information on how monkeys move their lips, the artifacts can be removed. As shown in figure 5, the artifacts around mouth and eyelid are mostly modified compared to the blendshape sculpted using

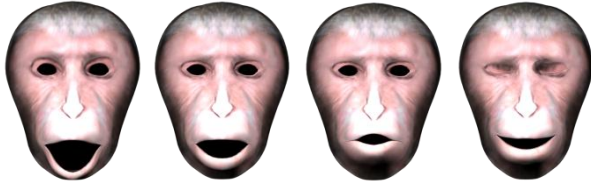


Figure .6 Training examples used for modifications.

naïve Deformation Transfer. Although these artifacts can be modified using modification method proposed in [SAI13], such work had the limitation that the effective way of refining arbitrary movement of opening and closing eyelids could not be defined when the target model had a different topology than that of the source model. In Character Transfer, it is possible to sculpt movements of an arbitrary amount of opening and closing of the eyelids by only sculpting training examples with eyelid movements.

From the results shown above, the goals we set out in Section 2 above were achieved indicating that Character Transfer is versatile and applicable to many applications that require facial animations.

5. EVALUATION

The fundamental goal of our approach was to sculpt blendshapes that are geometrically similar to those sculpted by an artist. Therefore, we evaluated the geometrical similarity between the two by computing the distance between vertices of the blendshape sculpted by Character Transfer and those sculpted by Deformation Transfer. We show our results using an error map in which error is shown by the color of the vertices in Figure 5. The maximum error shown in red is 5cm, whereas the minimum error shown in blue is 0cm. Note that the distance between the inner corners of the eyes is 50cm (figure (6)). Errors involving the eyes and mouth were identified as strikingly different for the blendshapes sculpted by Deformation Transfer.

We computed average root means square error of all vertices of three blendshapes sculpted by Character Transfer and those sculpted by Deformation Transfer. From the results are summarized in Table 1, Character Transfer is proven to be effective in that the blendshapes sculpted by Character Transfer has less error on average. We also computed root means square error of all vertices of the blendshapes using our hierarchal region segmentation approach more specifically; we subjectively defined regions for three blendshapes, which are shown in Figure 7. Results are summarized in Table 1 reveal that our region segmentation algorithm was able to effectively segment the region for the input blendshape.

We also evaluated the root means square error on all vertices of the facial animations created by using

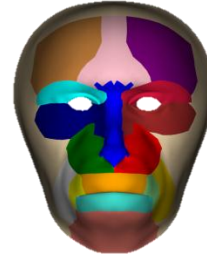


Figure 7. Subjectively defined Regions

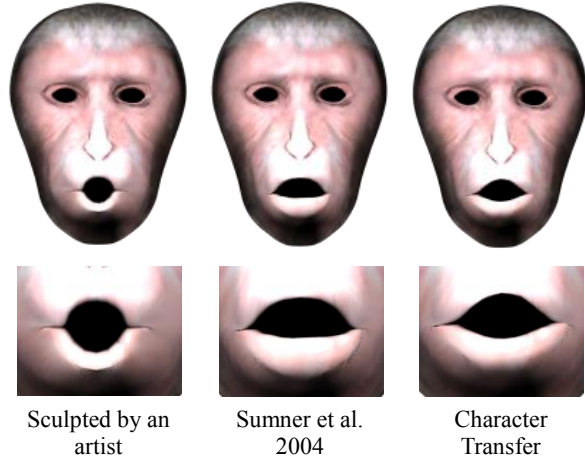


Figure 8. Comparison of blendshapes with extreme facial expression rendered by an artist, note that the geometry of the lips is slightly modified.

	Sumner et al. 2004	Region defined subjectively	Character Transfer
	3.629 cm	0.764 cm	0.611 cm

Table 2. Comparison with Deformation Transfer by the result of root means square error on all vertices.

sculpted by Deformation Transfer. Results are summarized in Table 2 and show that the facial animations created by our blendshapes have less error as compared with those that used blendshapes sculpted by Deformation Transfer. One of the goals we set out to achieve was to make Character Transfer applicable to blendshapes with arbitrary facial expressions. We applied Character Transfer to the facial expression which is dissimilar from the training example shown in Figure 8. Although results showed that Character Transfer is not able to create an exactly similar geometry, some of the features were correctly modified such as the way in which the lips moved. Modification for extreme expressions is difficult to apply using modification of [LI10] because it is only applicable when the training

examples are similar to the input facial expression. As for limitation of method, we observed that Character Transfer is not able to modify facial expressions in which it is impossible to estimate the blending coefficients given the set of training

	Sumner et al. 2004	Character Transfer
	1.032 cm	0.726 cm

Table 2. Comparison by the result of root means square error on all vertices when creating facial animation.

examples. Therefore, facial expressions on training examples are required to be as extreme as possible to create similar expressions to the training examples. This disadvantage also affects the hierarchical region segmentation algorithm of Character Transfer. Since our hierarchical region segmentation algorithm considers the way in which training examples move their facial parts in lower parts of tree, the segmentation results will not always be effective; however, it is an improvement over the modification method by [LI10] because blended facial expressions of training examples are required to be similar, not exact.

6. CONCLUSION AND FUTURE WORK

In this paper, we have presented a system called Character Transfer that modifies roughly created input blendshapes of arbitrary expressions to sculpt individual expressions. This method of example based modification of individuality is applicable to arbitrary expressions by blending the individualities extracted from training examples. By our novel blending method, we avoid the visual artifacts often introduced by the blending mapping. Further, the number of training examples can be reduced by blending several mappings to generate a new mapping. Character Transfer can automatically sculpt the blendshapes by only a few numbers of training examples. We also introduced a novel method to effectively segment the regions for the geometry and expressions of examples.

The key contribution in this paper is that we have formulated a novel method of modifying blendshapes that can be applied even when the number of training examples is limited. To the best of our knowledge, this property was never achieved in previous research; furthermore, we introduced novel segmentation and mapping blending approach.

For our future work, we aim to create a system that selects effective training examples. The training examples are currently selected subjectively; if we have a system that is able to systematically select training examples, we would dramatically increase

the usefulness of Character Transfer because it would become fully automatic. We are also interested in applying an improved approach to estimating more effective blending coefficients for Character Transfer. To date, we have only investigated the naïve estimation of blending coefficients for each region, but Character Transfer sometimes had problems since the coefficients could not be solved in the effective range. Better results would be generated if we were able to introduce a novel estimation method that better fits to Character Transfer.

7. REFERENCES

- [ALE10] Alexander, O., Rogers, M., Lambeth, W., Chiang, M., and Debevec, P: The digital emily project: photoreal facial modeling and animation. In *SIGGRAPH'09 Course*, 2009.
- [BER85] Bergeron, P., and Lachapelle, P.: Controlling facial expressions and body movements in the computer generate animated short 'Tony de Peltrie'. In *SIGGRAPH'85 Tutorial Notes Advance Computer Animation Course*, 1985.
- [BLA99] Blanz, V., and Vetter, T.: A morphable model for the synthesis of the 3D face, *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999.
- [BLA03] Blanz, V., Basso, C., Poggio, T., and Vetter, T.: Reanimating transfer for detailed-preserving surface editing, In *Vision, Modeling, Visualization 2006*, 357-364, 2003.
- [BOU13] Bouaziz, S., Wang, Y., and Pauly M.: Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG) SIGGRAPH 2013 Conference Proceedings*, Volume 32 Issue 4, July, Article No.40, 2013.
- [CAO13] Cao, C., Weng Y., Lin S., Zhou K.: 3D shape regression for real-time facial animation, *ACM Transactions on Graphics (TOG) SIGGRAPH 2013 Conference Proceedings*, Volume 32, Issue 4, July, Article No. 41, 2013.
- [CHO05] Choe, B., and Ko, H-S, Analysis and synthesis of facial expressions with hand-generated muscle actuation basis, *SIGGRAPH 2005*.
- [CHU04] Chuang, E.: Analysis, Synthesis, and Retargeting of Facial Expressions. PhD thesis, Stanford University, 2004.

- [JOS03] Joshi, P., Tien, W. C., Desbrum, M., and Pighin, F. 2003. Learning controls for blend shape based realistic facial animation. SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 187–192, 2003.
- [KAJ12] Kaji, S., Sampei, H., Sakata S., Yoshihiro M., and Anjyo K.: Mathematical analysis on affine maps for 2D shape interpolation, Proceeding SCA '12, Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation Pages 71-76, 2012.
- [LI10] Li, H., Weise T., Pauly M.: Example-based facial rigging. ACM Trans. Graph. 29, 4, 1–6, 2010.
- [LI13] Li, H., Yu J., Ye Y., Bregler C.: Realtime facial animation with On-the-fly correctives, ACM Transactions on Graphics (TOG) SIGGRAPH 2013 Conference Proceedings, Volume 32, Issue 4, July, Article No.42, 2013.
- [LIU08] Liu, X., Mao, T., Xia, S., Yu, Y., and Wang, Z. Facial animation by optimization blendshape from motion capture data, CASA2008 Special Issue volume 19, Issue 3-4, 2008.
- [PIG98] Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., and Salesin, D. H. Synthesizing realistic facial expressions from photograph, In Proc. SIGGRAPH '98, 1998.
- [SAI13] Saito, J.: Smooth contact-aware facial Blendshapes transfer. Digipro '13 Proceedings of the Symposium on Digital Production Pages 7-12, 2013.
- [SHO92] Shoemake K. and Duff T.: Matrix animation and polar decomposition. In Proceedings of the conference on Graphics, interface '92, Morgan Kaufmann Publishers Inc, pp. 258–264., 1992.
- [SUM04] Sumner R., Popovic J.: Deformation transfer for triangle meshes. ACM SIGGRAPH 2004 Papers. ACM, Los Angeles, California. 1015736 399-405, 2004.
- [SUM05] Sumner R.: Mesh modification using deformation transfer. PhD thesis, Massachusetts Institute of Technology, 2005.
- [TEN11] Tena J. R., Torre F. and Mathews I.: Interactive Region-Based Linear 3D Face Models. ACM SIGGRAPH2011, Vol.30, Article 76 , 2011.
- [VLA05] Vlasic, D., Brand, M., Pfister, H., and Popovic, J.: Face transfer with multilinear models. ACM Trans. Graph. 24, 2005.
- [WEI09] Weise, T., Leibe, B., and Gool, L. V. Fast: Fast 3D scanning with automatic motion compensation, in Proc. CVPR' 07, 2007.
- [ZHA04] Zhang, L., Snavely, N., Curless, B., and Seitz, S. M.: Spacetime face: High-resolution capture for modeling and animation, ACM Annual Conf. on Comp. Graphics, 2004.

