Structure-Aware Transfer of Facial Blendshapes

Christos Mousas * Christos-Nikolaos Anagnostopoulos †

Intelligent Multimedia and Virtual Reality Lab
Dept. of Cultural Technology and Communication
University of the Aegean, Greece

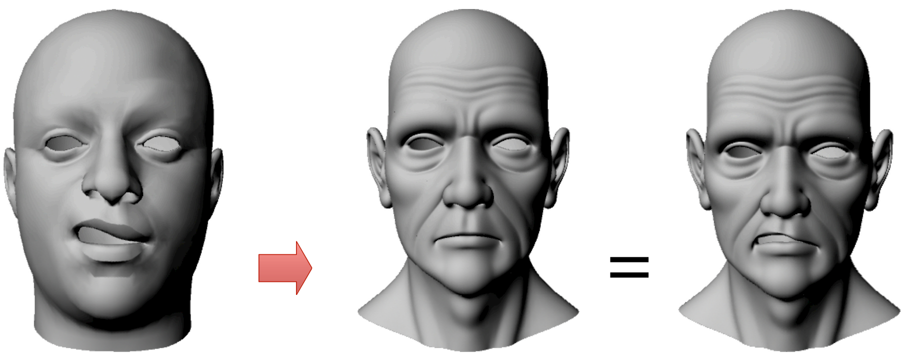


Figure 1: Using a reference face model that contains a blendshape (left) and a source face model (middle), the facial deformation of the reference model is transferred to the source model (right).

Abstract

This paper presents a novel mesh deformation method for transferring reference facial blendshapes to a source face model. The presented method uses information provided by a source and a reference face model to analyze the mesh similarities and, consequently, to transfer the deformation. This is achieved by considering the distribution of vertices between the two face models in conjunction with the gradients of the meshes. Both of these components are assigned to an optimization function that is solved in two steps: (i) distribution transfer and (ii) gradient maintenance optimization. Finally, in order to analyze the efficiency of such a method, the presented method was evaluated against previous solutions in terms of deformation transfer error, and computational time.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric Algorithms

Keywords: facial animation, blendshapes, deformation transfer, distribution, gradient, optimization

1 Introduction

In character animation pipeline, there are a number of experts in different fields who must be involved in order to produce the final animation of a virtual character. In addition to other experts, a modeler, a rigger and an animator are always required to model, rig and animate the character. However, a variety of methodologies to automate the content creation and animation pipeline have been proposed during recent years. Specifically, instead of modeling the shape of the character by hand, one can simply capture and reconstruct the shape [Li et al. 2009] using a 3D scanner or RGBD sensors like Microsoft Kinect. In addition, rather than rigging the character by hand, one can rig the character automatically based on example-based rigging techniques [Baran and Popović 2007]. Finally, instead of synthesizing the desired motion of the virtual characters using time-consuming keyframe techniques, it is possible to record the required motion directly using a motion capture system [Deutscher and Reid 2005]. Then, the motion sequences can be easily retargeted automatically [Monzani et al. 2000] to any character, placed in a 3D environment and rendered within the desired virtual scene. The aforementioned techniques that are used to automate the content creation and animation pipeline also can be applied to the facial animation process.

In data-driven facial animation, blendshape-based techniques are used extensively to synthesize a facial expression of a character. Generally, blendshape-based facial animation can be characterized as an effective method of defining facial deformations with intuitive user control. As mentioned in [Saito 2013], one of its disadvantages is the difficulty as well as the time required to produce all of the required facial blendshapes by hand, since each character requires an average of 100 or more blendshapes. Therefore, the development of the required number of blendshapes is a time consuming process as a great number of facial expressions must be generated for each different character that takes part in the film or game.

To automate the development of the required blendshapes, methodologies that are known as example-based deformation transfer [Saito 2013][Noh and Neumann 2001][Sumner and Popović 2004] have been proposed in recent years. With deformation transfer methods, modelers can focus only on modeling the required blendshape of a reference face model and then transfer automati-

*e-mail: chr.mousas@aegean.gr
†e-mail: canag@ct.aegean.gr

SCCG '15, April 22 – 24, 2015, Smolenice, Slovakia.
© 2015 ACM. ISBN 978-1-4503-3693-2/15/04…$15.00
DOI: http://dx.doi.org/10.1145/2788539.2788546

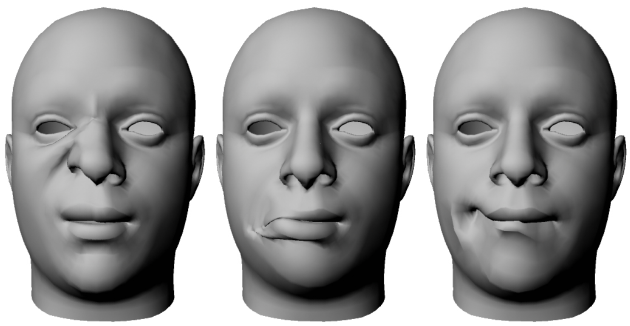


Figure 2: Examples of bad deformations transferred to a face model.

cally the designed blendshapes to different source face models (see Figure 1). The new facial expressions that the deformation techniques produce are usually sufficiently acceptable and an artist can immediately begin the clean-up required to improve the representation of the facial model. Hence, artists can benefit from these techniques since the time required for the facial rigging pipeline is reduced dramatically. Generally, in addition to the advantages that the blendshape transfer process can provide, there are limitations due to face deformations. As mentioned in [Sumner 2005], such techniques generally penetrate the eyelids or prevent them from closing completely. Moreover, the deformations are not always smooth. Therefore, the crumpling is particularly concentrated on the models of the face. Some simple examples of such poorly transferred deformations are shown in Figure 2.

Based on the aforementioned limitation that may appear during the blendshape transfer process, this paper presents a novel method to transfer efficiently the required deformations, while also solving the previously described undesirable aspects. Specifically, gradient maintenance is considered to be necessary for keeping a reference face model in its initial structure. Moreover, by using the distribution of vertices, it is possible to assess vertices positions. Therefore, by combining distributions and gradients in a facial blendshape transfer process, a basis for structure-aware blendshape transfer is provided. The method presented, which acts on the structure of face models, provides a smooth transfer of a reference blendshape to a source model, resulting in minimizing or removing completely crumpling artefacts (see Figures 3 and 4).

Besides the blendshape transfer process that is presented in this paper, two evaluation studies were conducted. In the first study, the presented method is evaluated against previous solutions by computing the deformation error, in both distribution and gradient, based on a simple metric. The computational time that is required to transfer a facial deformation to a source model was also evaluated.

To summarize, the main contributions of this paper are the following. Firstly, it presents a distribution-aware and gradient maintenance (instead of gradient transfer) method that transfers blendshapes efficiently from a reference to a source face model. Secondly, it is also presents a metric to compute the similarity error in vertices distribution and gradient that occurred after the blendshape transfer process.

The remainder of this paper is organized in the following manner. Section 2 provides related work on facial animation and blendshape transfer methodologies. Section 3 describes the proposed structure-aware method for transferring facial blendshapes. Section 4 presents the evaluations that were conducted in conjunction with the associate results. Finally, conclusions are drawn and potential future work is discussed in Section 5.

2 Related Work

During recent years a significant body of work on the development of facial animation techniques has been published. Among other methodologies, the first one is proposed is by Parke [Parke 1972]. However, for more recent state-of-the-art techniques, it is suggested that one reads [Osipa 2010], [Deng and Noh 2007] and [Lewis et al. 2014]. In the following two subsections, facial animation and blendshape transfer methods are presented.

2.1 Facial Animation

Capturing facial animation of real humans can be characterized as a time consuming process. This is especially true when dealing with marker-based techniques. In marker-based motion capture, a variety of markers are placed into the face model of a performer. The markers can be tracked by using a number of cameras or sensors. Then, the positions of the markers are assigned to a face model for generating the required animation. Generally, marker-based techniques can be characterized by their robustness. This is the main reason that they are widely used for real-time facial animation [Guenter et al. 1998][Lin and Ouhyoung 2005]. However, a variety of approaches for capturing the face shape and which do not involve markers are quite useful for retrieving fine-scale dynamics such as wrinkles and folds. However, all of the previously described approaches require highly specialized sensors (e.g., by using 3D scanners). Moreover, the capturing process should take place in a controlled studio environment [Zhang and Huang 2004][Bradley et al. 2010]. Based on the aforementioned methods, facial motion is recovered by using non-rigid registrations, and a variety of tracking algorithms across sequences of source geometry, texture or combinations of geometry and texture capture [Li et al. 2009][Bradley et al. 2010][Furukawa and Ponce 2009][Weise et al. 2009].

With the development of low-cost sensors (e.g., Microsoft Kinect) for capturing both the geometry and the texture of human shapes, a variety of interesting methodologies that simplify the procedure have been developed [Weise et al. 2011][Bouaziz et al. 2013][Li et al. 2013]. Specifically, in [Weise et al. 2011], a user-specified dynamic expression model is constructed in a pre-processing stage. This model is then registered with the observed depth data during the applications' runtime. This method achieves real-time performance and results that are more robust and accurate than previous video-based methods. Finally, a dynamic expression model with online tracking, which is proposed in [Bouaziz et al. 2013] and [Li et al. 2013], demonstrates impressive tracking and animation results for an arbitrary number of users without any training or calibration.

2.2 Deformation Transfer

As already mentioned, facial blendshapes are always required in an animation production pipeline. However, since generating blendshapes by hand is a time-consuming process, a variety of methodologies have been proposed to automatically transfer reference blendshapes to target face models. Generally, linear blendshape-based facial animation [Bergeron and Lachapelle 1985] provides good realism and control. However, as mentioned previously hundreds of blendshapes are usually required to animate face models realistically.

Among the first methodologies to solve this problem of mass production of facial blendshapes is the expressions cloning method that is proposed in [Noh and Neumann 2001]. This method sets benchmarks on the face model and spatially warps its geometry using a radial basis function to transfer the facial expressions. The basic disadvantage of this method is that it introduces interpolation artifacts. Moreover, the choice of optimal benchmarks is adhoc. Other methodologies, such as those presented in [Sumner 2005][Huang et al. 2006][Yu et al. 2004][Huang et al. 2014] apply deformation of an example mesh to a target mesh in the gradient space. The method that is presented differs from the previously mentioned solution in terms of gradient maintenance instead of gradient transfer. Moreover, by incorporating the distribution of vertices, the presented method can be characterized as a structure-aware deformation transfer. It should be noted that both the distribution and gradient have been incorporated into an optimization function that solves the blendshape transfer process.

Due to the deformation applied to a 3D mesh, penetrations and separations are generally the main problems that appear. In order to satisfy the requirements with these problems, spatial relationship-based approaches, such as that presented in [Zhou et al. 2010], have also been introduced. In these methods, maintaining the position of each body part relative to other body parts is achieved during the retargeting process. Generally, the encoding of the spatial relations between body parts is the main idea behind these approaches. It should be noted that this spatial relationship can be achieved using minimal spanning trees or Delaunay tetrahedralization. The basic advantage of such techniques is their ability to maintain coherent information, such as contacts between body parts, without the need to specify them. In [Li et al. 2010] a facial blendshapes transfer technique that extends a face blendshape transfer framework is proposed. This method computes transfers while respecting character-related features (e.g. wrinkles) that are provided by artists. Finally, one of the most recent works on facial blendshape transfer is proposed in [Saito 2013]. This method incorporates content and smoothness awareness during the blendshape transfer process. This has been solved by proposing the use of virtual triangles in conjunction with a Laplacian energy term that is implemented in the linear problem.

In this paper a novel method is presented that automatically transfers the facial blendshapes of a reference to a source face model. Our solution takes the advantages of shape deformation by using distribution matching and gradient maintenance and combines them in an optimization function. The advantage of this method in comparison to the approaches presented previously is its ability to maintain the structure of the face model while transferring the required deformation. Moreover, aspects such as crumpling and eyelid penetration are eliminated since the presented method acts simultaneously on the structures and the vertices of the face models, instead of acting directly only on the vertices.

3 method

The blendshape transfer process has two main goals. The first it is that the output model must retain the representation of the source model. The second is that the reference deformation should appear in the output model. The method that was developed to deal with these two goals is presented below. In the remainder of this paper, the subscript s represents the source face model, r represents the face model that contains a reference blendshape, and o, the output, represents the source face model, s, deformed based on r.

To achieve the required deformation, the presented approach maintains the gradient parameter from the source mesh, and applies the deformation from the mesh that contains the required blendshape (reference face model). The gradient map maintenance of a source face model and a reference face model's distribution of vertices are used as our target. For simplicity, a mesh of a face model F is represented as $F = \{\Phi v, \nabla m\}$, where Φ represents the distribution of vertices v, and ∇ represents the gradient of a mesh m.

In the presented method, the face deformation transfer problem is formulated as an optimization problem that minimizes the above equation in least square sense:

$$\min\left(\sum \|\Phi v_o - \Phi v_r\|^2 + \lambda \sum \|\nabla m_s - \nabla m_o\|^2\right) \tag{1}$$

where λ is a user-defined weight factor that changes interactively the importance of gradient maintenance and vertices positions. However, since both distribution and gradient should count equally, $\lambda = 1$. Generally, Equation 1 is difficult to solve fast because the first term of the distribution, Φv, is a statistical term that acts on the face model, whereas the gradient term is applied to the structure of a three-dimensional model. For this reason, the optimization problem is split into two steps as presented below.

In a first step, a distribution transfer method was used to transform the source face model, s, into an intermediate model f that has a similar form as the reference model r. To achieve this transformation, the source and the reference face models are represented as two sets of sample vertices, where the dimension of each sample is $N = 3$ (x, y, and z axes). To reshape the source face model to an intermediate model f it is now necessary to transfer the corresponding distribution of vertices of the source to the reference face model. To simplify this transformation process, the vertices are adopted from a continuous position probability density function (pdf): p from the intermediate (f) face model's vertices, and g for the reference (r) face model's vertices. In this case, the problem of the transformation is to find a differentiable mapping T that transforms the original position pdf p to a new position pdf that fits the target pdf g. The aforementioned problem can be expressed as the mass transportation problem [Rachev and Rüschendorf 1998].

A one-dimensional solution was used for the transformation process. This means that each of the axes of the vertices is treated separately. Specifically, the differentiable mapping returns the following constraint that corresponds to a change of vertices positions:

$$p(f)df = g(r)dr \tag{2}$$

Now, by integrating both sides of the equality the following is provided:

$$\int^{f} p(f)df = \int^{T(f)} g(r)dr \tag{3}$$

By using cumulative probability density function notations P and G for p and g respectively, the representation for the mapping T becomes:

$$\forall f \in \mathbf{R}, \quad T(f) = G^{-1}(P(f)) \tag{4}$$

where $G^{-1}(x) = \inf\{p|G(f) \geq x\}$. Finally, the use of discrete look-up tables can easily solve this mapping.

Having retrieved the intermediate face model f, the optimization problem of Equation 1 can be rewritten as:

$$\min\left(\sum \|\Phi v_o - \Phi v_f\|^2 + \lambda \sum \left[\left(\frac{\partial o}{\partial x} - \frac{\partial s}{\partial x}\right) + \left(\frac{\partial o}{\partial y} - \frac{\partial s}{\partial y}\right) + \left(\frac{\partial o}{\partial z} - \frac{\partial s}{\partial z}\right)\right]\right) \tag{5}$$

This means that the output face model, o, is likely to maintain the deformation of the newly synthesized face model, f, and the gradients of s, which is the source face model. It should be noted that these sums are taken over the vertices in the relative face model.

By using matrix notation in the second step, it is possible to rewrite Equation 5. Hence, instead of solving a minimization problem, we can define a solution that is described in a linear system of equations. This linear system is expressed as:

$$[I + \lambda \times K] \times o = \Phi v_f + \lambda \times K \times s \tag{6}$$

where K represents:

$$K = D_x^T D_x + D_y^T D_y + D_z^T D_z \quad (7)$$

In Equation 6, I denotes the identity matrix and the matrices, that are represented by K, of the gradient term, along with the x, y and z axis, are computed by D_x, D_y and D_z respectively. It should be noted that, in the presented method, D_x, D_y and D_z are forward Sobel difference operators, and D_x^T, D_y^T and D_z^T are backward Sobel difference operators. Figure 3 and 4 illustrate examples of facial deformations that have been transferred to different models generated with the aforementioned method.

Generally, Equation 6 could be described as a system of multiple linear equations. Solving such a linear system a lot of memory and CPU time is required. However, to solve such a problem, a variety of solutions have been proposed in recent years. Specifically, a quadtree-related [Agarwala 2007] and a multi-grid-related [Bolz et al. 2003] technique could be applied in the presented method, because the first technique can reduce the scale of solving such a problem and the second technique can raise the effectiveness of computation.

4 Evaluations and Results

In this section, the method presented is evaluated based firstly on a metric that measures the performance of the face deformation transfer and secondly on the computational time that is required to transfer a reference blendshape to a source face model. In order to illustrate the efficiency of the method presented, the aforementioned transfer process and the computational time are compared to methodologies proposed in [Saito 2013][Noh and Neumann 2001][Sumner and Popović 2004] and [Pyun et al. 2003]. The experimental evaluation presented was performed with an Intel i7 that has a 2.2GHz CPU and 8GB of memory. Both evaluations and the associated results are presented in the following two subsections.

Finally, to illustrate the efficiency of transferring facial blendshapes from a reference to a source face model, we captured and retargeted to our models a number of motion sequences. For the capturing process, we used a 120Hz Vicon system that employed six cameras. For the retargeting process, we used the radial basis function (RBF) method, as proposed in [Deng et al. 2006]. Example of blendshape-based animation that was retargeted to our characters is shown in the accompanying video.

4.1 Blendshape Transfer

In this evaluation process, the characteristics of the source face model and the deformation of the reference face model are considered to be the key components in the facial blendshape transfer process presented. For that reason, an evaluation metric that considers both the distribution and the gradient to compute the incorrect estimates of the facial blendshape transfer method is presented below. The presented metric e is defined as:

$$\begin{aligned} e &= e_{distribution} + \lambda \times e_{gradient} \\ &= \sum \|\Phi v_o - \Phi v_r\|^2 + \lambda \times \sum \|\nabla m_o - \nabla m_s\|^2 \end{aligned} \quad (8)$$

where M is the total number of vertices in a face model, Φ and ∇ denote the distribution and the gradient map of the relevant face models respectively, and λ is a weighted factor that is set to $\lambda = 1$ in order to represent the equal importance between vertices and gradient

For each of the different methodologies, the same source face model and the 48 blendshapes of the face model provided by [Osipa 2010] was used. For each method, the average of e for the 48 different blendshapes and the standard deviation were computed. It should be noted that the lower the e is, the lower the error of the deformations that are generated is. Examples of transferred facial blendshapes that are based on both the presented method and the four methods used for the evaluation process are shown in Figure 5. The results obtained from this evaluation appear in Table 1. Specifically, the experimental results show that the method presented manages to transfer quite efficiently a reference deformation to a source face model. This means that the method provides the best and most stable performance in terms of distribution and gradient.

Methods	e	σ
[Saito 2013]	0.103	0.010
[Noh and Neumann 2001]	0.436	0.187
[Sumner and Popović 2004]	0.271	0.142
[Pyun et al. 2003]	0.388	0.093
our method	0.097	0.013

Table 1: The results obtained while evaluating the presented method against [Saito 2013][Noh and Neumann 2001][Sumner and Popović 2004] and [Pyun et al. 2003] in distribution and gradient as presented in Equation 8. e indicates the error and σ is the standard deviation.

4.2 Computational Time

In this paper, we also determined the computation time in transferring a reference blendshape to a source model. The times that were determined are shown in Table 2. Again, for this evaluation process, the 48 blendshapes and the face model provided by [Osipa 2010] were used. For all of the methodologies, the average time (in seconds) and the standard deviation were determined. A comparison between the presented method and [Saito 2013][Noh and Neumann 2001][Sumner and Popović 2004] and [Pyun et al. 2003] indicates that the time the presented method requires is a disadvantage. However, the aforementioned methodologies do not provide deformations that are as accurate as with the presented method.

Methods	t	σ
[Saito 2013]	0.41	0.15
[Noh and Neumann 2001]	0.17	0.11
[Sumner and Popović 2004]	0.23	0.09
[Pyun et al. 2003]	0.59	0.32
Our method	0.86	0.47

Table 2: The computational times that are required to transfer a reference blendshape to a source model when using the proposed method and the methodologies presented in [Saito 2013][Noh and Neumann 2001][Sumner and Popović 2004] and [Pyun et al. 2003]. t indicates the time (in seconds) and σ the standard deviation.

In addition, the following also could be stated about the required computational time. The presented method uses a high rate of CPU time since it is necessary to solve a vast number of linear equations. However, there are various techniques to solve huge-scale linear equations that achieve real-time performance. These include the quadtree-based scale reduction method [Agarwala 2007], and the generalized GPU multigrid-related method [Bolz et al. 2003][Goodnight et al. 2003]. Based on the aforementioned techniques, we believe strongly that our method could benefit from these solutions, since the computational time of our method can be improved significantly.

However, facial blendshape transfer is generally required in order to provide artists with a face model that contains all required

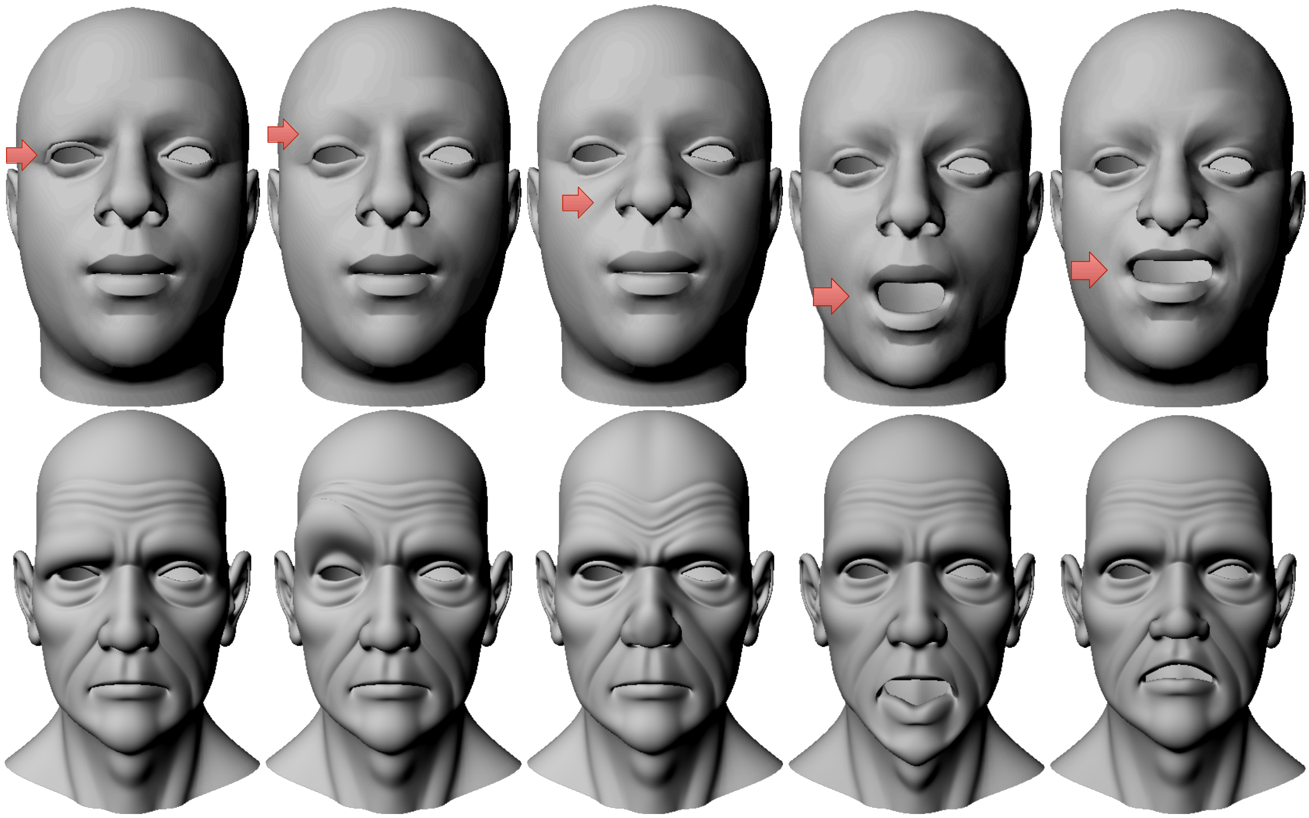


Figure 3: Example of transferred facial blendshapes based on the method presented. The upper row illustrates the reference blendshapes (the red arrow points to the reference deformation), and the lower row represents blendshapes that have been transferred to a different face model produced with the presented method.

deformations. Based on the blendshapes that are provided, artists begin the clean-up to improve the representation of the facial model. It could be mentioned that the computational time may not be a factor that influences an artist's decision to choose the presented method. This is especially true, if the artist's time-consuming process to scalp or model the required blendshapes by hand is taken into consideration. Hence, the few seconds that each of the methodologies requires to efficiently transfer a number of reference blendshapes to a source face model could be interpreted as a speed-up of the hand-made process.

5 Conclusions and Future Work

In this paper, a method was presented to automatically transfer facial blendshapes from a reference to a source face model. The novelty of this method is due mainly to its ability to affect facial blendshape transfer by proposing an optimization fuction that counts the distribution and the gradient of the relevant face models. Both the distribution and the gradient can represent effectively the structure of the face models, while also providing the necessary parameters to efficiently transfer the required blendshapes.

To analyze the effectiveness of the presented method, two different evaluations were conducted. Firstly, a metric for measuring the deformation transfer process was introduced. Secondly, the computation time that is required by the system to automatically transfer a reference blendshape to a source face model was measured. Both of these evaluations were conducted for the presented method and also the four methodologies that were proposed previously.

Based on the evaluations, the following should be mentioned. The presented method achieves a lower error rate than the previous methodologies. However, the computational time that is required is higher than those required with the previous solutions. This is because the presented method uses a great deal of memory to solve the iterative process of distribution matching and the system of linear equations. We believe that, by using the generalized GPU multigrid-related methods [Bolz et al. 2003][Goodnight et al. 2003] and the quadtree-based technique [Agarwala 2007], the computational time of the presented method could be decreased dramatically.

In our future work, we wish to develop further the facial blendshape transfer process. Firstly, we believe that, by implementing parameters such as elasticity [Mpiperis et al. 2008], mesh curvature descriptors [Jaiman et al. 2006] and primitive surface features [Wang et al. 2006], into an optimization function, it will be possible to minimize even more the errors detected in the evaluation process. Moreover, we believe that the more efficient methods to solve the linear system of equations that resulted from the gradient domain could be quite beneficial if implemented. Possible examples include vectorization techniques [Wenyin and Dori 1998].

References

AGARWALA, A. 2007. Efficient gradient-domain compositing using quadtrees. *ACM Transactions on Graphics 26*, 3 (August), Article No. 94.

BARAN, I., AND POPOVIĆ, J. 2007. Automatic rigging and animation of 3d characters. *ACM Transactions on Graphics 26*, 3 (August), Article No. 72.

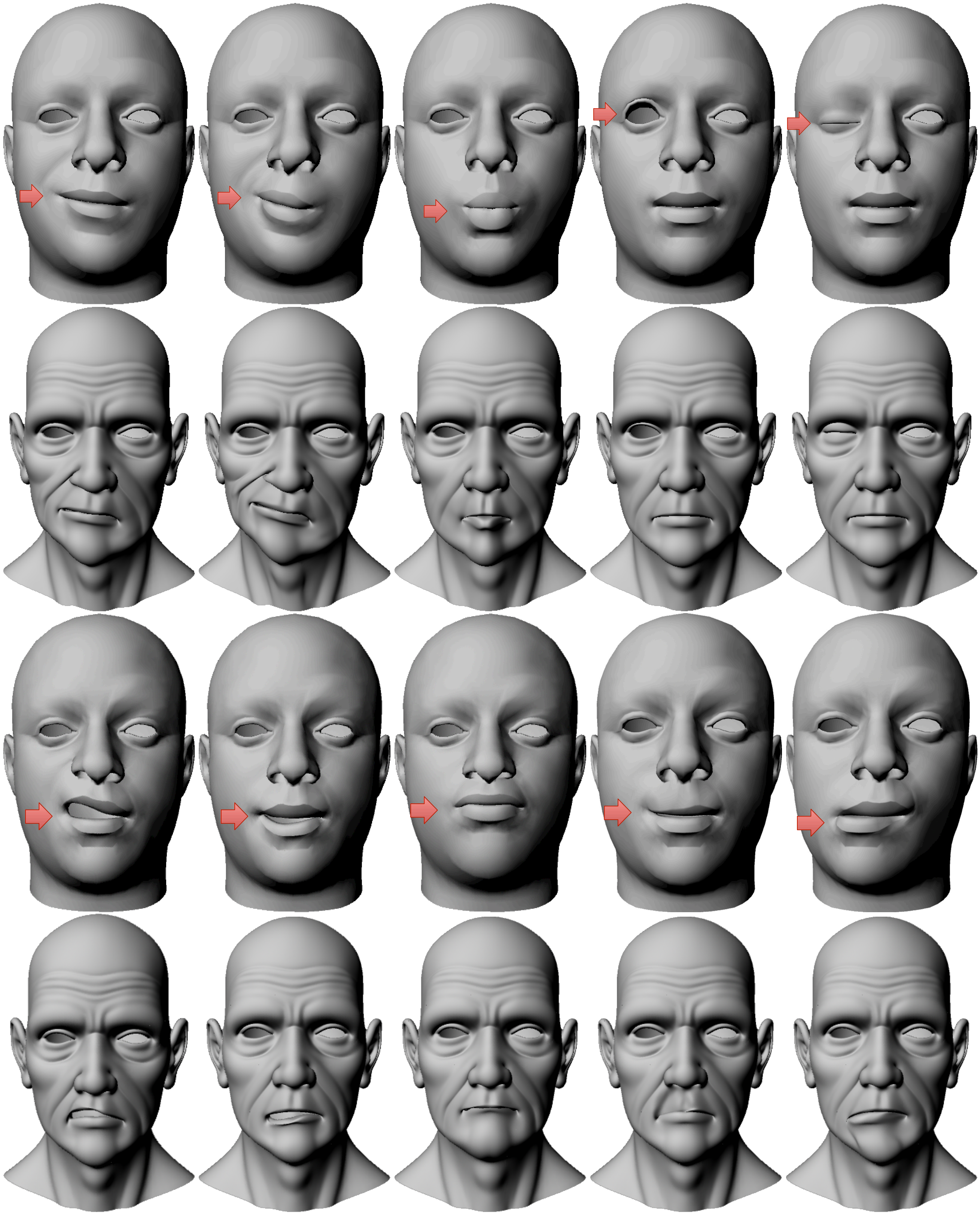
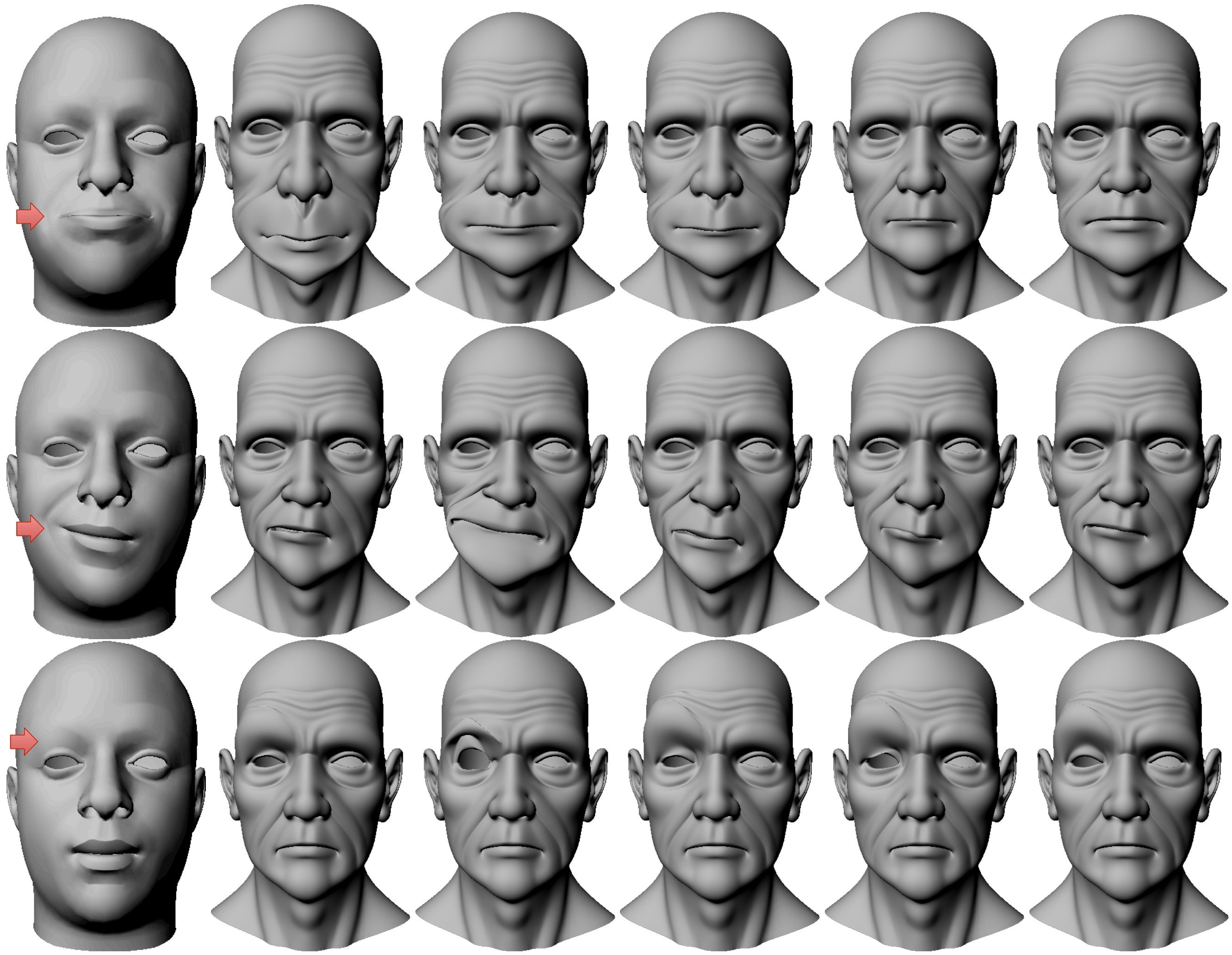


Figure 4: Additional examples of transferred facial blendshapes based on the method presented. The first and the third row illustrate the reference blendshapes (the red arrow points to the reference deformation). The second and the fourth row represent blendshapes that have been transferred to a different face model produced with the presented method.



(a) reference (b) Saito (c) Noh & Neumann (d) Sumner & Popović (e) Pyun et al. (f) our method

Figure 5: Results obtained with the presented facial blendshape transfer method (right column) and previous solutions (Saito [Saito 2013], Noh and Neuman [Noh and Neumann 2001], Summer and Popović [Sumner and Popović 2004], and Pyun et al. [Pyun et al. 2003]). The left column illustrates the reference blendshapes (the red arrow points to the reference deformation).

BERGERON, P., AND LACHAPELLE, P. 1985. Controlling facial expressions and body movements in the computer generated animated short "tony de peltrie". In *SIGGRAPH Tutorial Notes, Advanced Computer Animation Course*.

BOLZ, J., FARMER, I., GRINSPUN, E., AND SCHRÖODER, P. 2003. Sparse matrix solvers on the gpu: conjugate gradients and multigrid. *ACM Transactions on Graphics 22*, 3 (July), 917–924.

BOUAZIZ, S., WANG, Y., AND PAULY, M. 2013. Online modeling for realtime facial animation. *ACM Transactions on Graphics 32*, 3, Article No. 40.

BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. 2010. High resolution passive facial performance capture. *ACM Transactions on Graphics 29*, 4, Article No. 41.

DENG, Z., AND NOH, J. 2007. Computer facial animation: A survey. In *Data-Driven 3D Facial Animation*, Springer London, 1–28.

DENG, Z., CHIANG, P. Y., FOX, P., AND NEUMANN, U. 2006. Animating blendshape faces by cross-mapping motion capture data. In *Symposium on Interactive 3D Graphics and Games*, 43–48.

DEUTSCHER, J., AND REID, I. 2005. Articulated body motion capture by stochastic search. *International Journal of Computer Vision 61*, 2, 185–205.

FURUKAWA, Y., AND PONCE, J. 2009. Dense 3d motion capture for human faces. In *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, 1674–1681.

GOODNIGHT, N., WOOLLEY, C., LEWIN, G., LUEBKE, D., AND HUMPHREYS, G. 2003. A multigrid solver for boundary value problems using programmable graphics hardware. In *ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, Eurographics Association, 102–111.

GUENTER, B., GRIMM, C., WOOD, D., MALVAR, H., AND PIGHIN, F. 1998. Making faces. In *Annual Conference on Computer Graphics and Interactive Techniques*, ACM, 55–66.

HUANG, J., SHI, X., LIU, X., ZHOU, K., WEI, L. Y., TENG, S. H., BAO, H., GUO, B., AND SHUM, H. Y. 2006. Subspace gradient domain mesh deformation. *ACM Transactions on Graphics 25*, 3, 1126–1134.

HUANG, Z., YAO, J., ZHONG, Z., LIU, Y., AND GUO, X. 2014. Sparse localized decomposition of deformation gradients. *Computer Graphics Forum 33*, 7 (October), 239–248.

JAIMAN, R. K., JIAO, X., GEUBELLE, P. H., AND LOTH, E. 2006. Conservative load transfer along curved fluid–solid interface with non-matching meshes. *Journal of Computational Physics 218*, 1, 372–397.

LEWIS, J. P., ANJYO, K., RHEE, T., ZHANG, M., PIGHIN, F., AND DENG, Z. 2014. Practice and theory of blendshape facial models. In *Eurographics - State of the Art Reports*, Eurographics Association, 199–218.

LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics 28*, 5 (December), Article No. 175.

LI, H., WEISE, T., AND PAULY, M. 2010. Example-based facial rigging. *ACM Transactions on Graphics 29*, 4, Article No. 32.

LI, H., YU, J., YE, Y., AND BREGLER, C. 2013. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics 32*, 4, Article No. 42.

LIN, I. C., AND OUHYOUNG, M. 2005. Mirror mocap: Automatic and efficient capture of dense 3d facial motion parameters from video. *The Visual Computer 21*, 6, 355–372.

MONZANI, J. S., BAERLOCHER, P., BOULIC, R., AND THALMANN, D. 2000. Using an intermediate skeleton and inverse kinematics for motion retargeting. *Computer Graphics Forum 19*, 3 (September), 11–19.

MPIPERIS, I., MALASSIOTIS, S., AND STRINTZIS, M. G. 2008. Bilinear elastically deformable models with application to 3d face and facial expression recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE Press, 1–8.

NOH, J. Y., AND NEUMANN, U. 2001. Expression cloning. In *Annual Conference on Computer Graphics and Interactive Techniques*, 277–288.

OSIPA, J. 2010. *Stop staring: facial modeling and animation done right (3rd edition)*. John Wiley and Sons.

PARKE, F. I. 1972. *Computer generated animation of faces*. Master's thesis, University of Utah.

PYUN, H., KIM, Y., CHAE, W., KANG, H. W., AND SHIN, S. Y. 2003. An example-based approach for facial expression cloning. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 167–176.

RACHEV, S. T., AND RÜSCHENDORF, L. 1998. *Mass Transportation Problems: Volume I: Theory*. Springer.

SAITO, J. 2013. Smooth contact-aware facial blendshapes transfer. In *Symposium on Digital Production*, ACM Press, 7–12.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Transactions on Graphics 23*, 3 (August), 399–405.

SUMNER, R. W. 2005. *Mesh modification using deformation gradients*. Ph.d. dissertation, Massachusetts Institute of Technology.

WANG, J., YIN, L., WEI, X., AND SUN, Y. 2006. 3d facial expression recognition based on primitive surface feature distribution. In *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, 1399–1406.

WEISE, T., LI, H., VAN GOOL, L., AND PAULY, M. 2009. Face/off: Live facial puppetry. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 7–16.

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Realtime performance-based facial animation. *ACM Transactions on Graphics 30*, 4, Article No. 77.

WENYIN, L., AND DORI, D. 1998. A survey of non-thinning based vectorization methods. In *Advances in Pattern Recognition*, Springer Berlin Heidelberg, 230–241.

YU, Y., ZHOU, K., XU, D., SHI, X., BAO, H., GUO, B., AND SHUM, H. Y. 2004. Mesh editing with poisson-based gradient field manipulation. *ACM Transactions on Graphics 23*, 3 (August), 644–651.

ZHANG, S., AND HUANG, P. 2004. High-resolution, real-time 3d shape acquisition. In *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE.

ZHOU, K., XU, W., TONG, Y., AND DESBRUN, M. 2010. Deformation transfer to multicomponent objects. *Computer Graphics Forum 29*, 2 (May), 319–325.

Acknowledgements

The authors would like to thank Jason Osipa for providing the face model and the blendshapes that were used in our study. In addition, we would like to thank the anonymous reviewers who helped us to improve this paper. The remainder face model that presented in this paper was downloaded from TurboSquid (http://www.turbosquid.com/)