Real-Time Dynamic Wrinkles of Face for Animated Skinned Mesh

Ludovic Dutreve, Alexandre Meyer, Saida Bouakaz

Real-Time Dynamic Wrinkles of Face for Animated Skinned Mesh

L. Dutreve and A. Meyer and S. Bouakaz

Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France
PlayAll

Abstract. This paper presents a method to add fine details, such as wrinkles and bulges, on a virtual face animated by common skinning techniques. Our system is based on a small set of reference poses (combinations of skeleton poses and wrinkle maps). At runtime, the current pose is compared with the reference skeleton poses, wrinkle maps are blended and applied where similarities exist. The poses evaluation is done with skeleton's bones transformations. Skinning weights are used to associate rendered fragments and reference poses. This technique was designed to be easily inserted into a conventional real-time pipeline based on skinning animation and bump mapping rendering.

1 Introduction

Recent rendering and animation advances improve the realism of complex scenes. However, a realistic facial animation for real-time applications is still hard to obtain. Some reasons could explain this difficulty. The first one remains from the lack of computational and memory resources supplied by an interactive system, and another one remains from the difficulty to simulate small deformations of the skin like wrinkles and bulges when muscles deform its surface. As small as considered details are, they greatly contribute to the recognition of facial expressions, and thus to the realism of virtual faces.

Since many works have been proposed for large-scale animations and deformations, only a few of them deal with real-time animation of small-scale details. We denote small-scale details in an *animation* context, *i.e.* wrinkles and bulges appearing while muscles contractions, instead of the micro-structures of the skin independent to facial expressions. Recent progress in motion capture or wrinkles generation techniques allow the production of these details in a form of high-resolution meshes that are not fully usable for real-time animation applications such as video games. Nevertheless, converting these detailed meshes into few wrinkle maps may be a good input for our real-time and low-memory technique.

Oat [1] proposed a technique to blend wrinkle maps to render animated details on a human face. A mask map defines different facial areas, coefficients are manually tuned for each area to blend the wrinkle maps. This technique provides good results at an interactive time, but the task of defining wrinkle

maps coefficients at each frame or key-frame of the animation is a long and tedious work. Furthermore, a new animation will require new coefficients and previous work could not be reused. It is why we intend to improve and adapt it to a face animated by skinning techniques.

Our dynamic wrinkling system is based on a small set of reference poses. A reference pose is a pair of a large-scale deformation (a skeleton pose) and its associated small-scale details (stored in the form of a wrinkle map). During the animation, the current pose is compared with reference poses and coefficients are automatically computed. Notice that comparison is done at a "bone level" resulting in local blends and independence between areas of the face.

The main contribution of our approach is that we propose a technique to easily add animation details on a face animated by skinning technique. Wrinkle maps coefficients are computed automatically by using a small set of reference poses. We do not use a mask map to separate areas of the face, we propose to use skinning weights as a correlation between bones movements and reference wrinkle maps influences. Moreover, it was designed to be easily inserted into a conventional runtime pipeline based on skinning and bump mapping such as new generation video game engines or any other interactive applications. Indeed, by blending wrinkle map on the GPU, our approach does not modify the per-pixel lighting, no additional rendering pass is required. And by only taking the bones positions as input, our approach does not need to change the animation aspect by skinning. Thus, the implementation requires a few efforts. The supplementary material needed compared to a skinning-based animation system is only a few set of reference poses, two are sufficient for the face. The runtime computation cost is not much more important than for classical animation and rendering pipelines. Finally as results, our approach greatly improves the realism by increasing the expressiveness of the face.

2 Related Work

Although many articles have been published on the large scale facial animation, adding details to an animation in real-time remains a difficult task. Some methods proposed physical models for skin aging and wrinkling simulation [2–5]. Some research focused on details acquisition from real faces, based on intensity ratio images which are then convert to normal maps [6], shape-from-shading [7] or other similar self-shadowing extraction techniques [8, 9], or with the help of structured light projectors [10, 11]. Some works proposed details transfer technique to use existing details to new faces [6, 11, 9]. We focus this state of the art on wrinkling systems for animated mesh.

Volino *et al.* [12] presented a technique to add wrinkles on an existed animated triangulate surface. Based on the edges length variations, the amplitudes of applied height-maps are modified. For each edge and texture map, a shape coefficient is calculated to know the influence of an edge with the given map. More the edge is perpendicular to wrinkles, and more its compression or elongation will disturb the height map. The rendering is done by a ray-tracer with

bump mapping shading and an adaptive mesh refinement. Bando *et al.* [13] proposed a method to generate fine and large scale wrinkles on human body parts. Fine scale wrinkles are rendered using bump mapping and are obtained by using direction field defined by user. Large scale wrinkles are rendered by displacing vertices of a high-resolution mesh and obtained by using Bezier curves manually drawn by user. Wrinkles amplitudes are modulated along mesh animation by computing triangle shrinkage. Na *et al.* [11] extended the Expression Cloning technique [14] to allow a hierarchical retargeting. Transfer could be apply to different animation level of details, from a low-resolution up to a high-resolution mesh where dynamic wrinkles appear.

While some techniques were proposed to generate wrinkles, few approaches focused on real-time applications [15, 1]. Larboulette *et al.* [15] proposed a technique to simulate dynamic wrinkles. The user defines wrinkling areas by drawing a perpendicular segment of wrinkles (wrinkling area), following by the choice of a 2D discrete control curve (wrinkle template). The control curve conserve its length while the mesh deformation, generating amplitude variations. Wrinkles are obtained by mesh subdivision and displacement along the wrinkle curve.

Many methods need a high resolution mesh or a costly *on-the-fly* mesh subdivision scheme to generate or animate wrinkles [9, 7, 11, 4]. Due to real-time constraints, these techniques are difficult to use in an efficient way. Recent advances in GPU computing allow to render efficiently fine details by using bump maps for interactive rendering.

Oat presented in [1] a GPU technique to easily blend wrinkle maps applied to a mesh. Maps are subdivided in regions, for each region, coefficients allow to blend between the wrinkle maps. This technique requires few computational and storage costs, three normal maps are used for a neutral, a stretched and a compressed expressions. Furthermore it is easily implemented and added to an existing animation framework. As we explain in the introduction, the main drawback of this method is that it requires manual tuning of the wrinkle maps coefficients for each region. Our method aims to propose a real-time dynamic wrinkling system for skinned face generating automatically wrinkle maps coefficients and without the requirement of a mask map.

3 Overview

Figure 1 shows the complete framework of our wrinkling animation technique. The first step is to create o reference poses. Each of them consists on a skeleton pose (large scale deformation) and a wrinkle map (fine scale deformation). While the runtime animation, the current skeleton pose is compared with reference poses and coefficients are calculated for each bones of each reference poses. Skinning and reference poses/bones weights are used to blend wrinkle maps with the default normal map and render the face with surface lighting variations. This last step is done on the GPU while the per-pixel lighting process.

The remainder of this paper is organized as follows. In Section 4, we briefly present the existing large-scale deformation technique and the input reference

poses. The section 5 presents the main part of the paper, we show how we use the reference poses in real-time to obtain dynamic wrinkling and fine details animation. In Section 6, we present some results and conclude this paper.

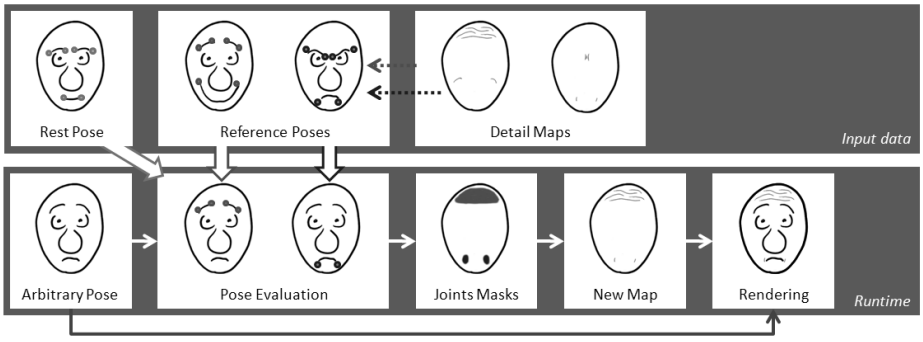

Fig. 1. Required input data are a classic skinned mesh in a rest pose and some reference poses (reference pose = skeleton pose + wrinkle map). In runtime, each pose of the animation is compared with the reference poses, bone by bone, then skinning influence is used as masks to apply the bones poses evaluation. Wrinkle maps are blended on the GPU and a per-pixel lighting allows to render the current frame with dynamic wrinkles and details.

4 Large-Scale Deformations and Reference Poses

As mentioned above, our goal is to adapt the wrinkle maps method to the family of skinning techniques which perform the large scale deformations of the face. They offer advantages in terms of memory cost and simplicity of posing. Many algorithms have been published [16–19] about skinning and its possible improvements. Our dynamic wrinkling technique could be used with any of the skinning methods cited. We only need that mesh vertices are attached to one or more bones with a convex combination of weights.

Reference poses are manually created by CG artist. He deforms the facial "skeleton" to obtain the expressions where he want to wrinkles appear. These expressions should be strongly pronounced to provide better results, *i.e.* , if artist wants that wrinkles appear while eyebrows rise up, he should pose the face with the maximum eyebrows rising up, when wrinkles furrows are deepest.

Since influences work with bones or group of bones, it is possible to define a pose where wrinkles appear in various areas of the face. Having details in different areas will not cause that all of these details appear at the same time at an arbitrary frame. For example, details around the mouth would appear independently with forehead details, even if they are described in a same reference pose.

5 Animated Wrinkles

Our goal is to use preprocessed wrinkles data to generate visual dynamic fine deformations of the face surface, instead of computing costly functions and algorithms in order to generate wrinkles on the fly. We explain in this section how reference poses are used in real-time and at arbitrary poses. The runtime algorithm consists on three main steps:

- Computing influences of each reference pose on each bone.
- Associating reference poses and vertices by using skinning weights and influences computed at the first step.
- Blending wrinkle maps and the default normal map while the per-pixel lighting process.

5.1 Pose Evaluation

The first step is to compute the influence of each reference pose on each bone. This consists to find how the bone position at an arbitrary frame differs from its position at the reference poses[1]. Computing these influences at the bone level instead of a full pose level allows to determine regions of interest. This offers the possibility to apply different reference poses at same time. Resulting in the need of less reference poses (only 2 are sufficient for face: a stretched and a compressed expression).

We define the influence of the pose $\mathcal{P}_k$ for the bone $\mathcal{J}_i$ at an arbitrary frame f by this equation:

$$\mathcal{I}_{\mathcal{P}_k}(\mathcal{J}_{if}) = \begin{cases} 0 & \text{if} \quad \mathcal{J}_{i\mathcal{P}_0} = \mathcal{J}_{i\mathcal{P}_k} \\ \alpha_{ikf} & \text{if} \quad 0 \leq \alpha_{ikf} \leq 1 \\ 0 & \text{if} \quad \alpha_{ifk} < 0 \\ 1 & \text{if} \quad \alpha_{ifk} > 1 \end{cases}$$

with

$$\alpha_{ifk} = \frac{(\mathcal{J}_{if} - \mathcal{J}_{i\mathcal{P}_0}) \odot (\mathcal{J}_{i\mathcal{P}_k} - \mathcal{J}_{i\mathcal{P}_0})}{||\,(\mathcal{J}_{i\mathcal{P}_k} - \mathcal{J}_{i\mathcal{P}_0})\,||}$$

where $\odot$ denotes the dot product and $||.||$ denotes the euclidean distance. This computation consists on projecting orthogonally $\mathcal{J}_{if}$ onto the segment $(\mathcal{J}_{i\mathcal{P}_0}, \mathcal{J}_{i\mathcal{P}_k})$.

So, at each frame, reference poses influences for bone $\mathcal{J}_i$ could be written as a vector of dimension $o+1$ (o reference poses plus the rest pose) $\boldsymbol{\alpha_{if}} =< \alpha_{if0}, \alpha_{if1}, ..., \alpha_{ifk}, ...\alpha_{ifo} >$. α_{if0} is the influence of the rest pose.

$$\alpha_{if0} = max\left(0, 1 - \sum_{k=1}^{o} \alpha_{ifk}\right)$$

[1] Notice that we deal with bones positions in the head root bone coordinates system, so we could assume that head rigid transformation will not cause problems while pose evaluation.

5.2 Bones Masks

Once we know the reference poses influences for each bone, we could use them for the per-pixel lighting. The main idea is to use skinning weights to compute the influence of reference poses for each vertex, and by the interpolation done during the rasterization phase, for each fragment (Fig. 2). Since wrinkles and expressive details are greatly related to the face deformations, we can deduce that these details are related with bones transformations too. So we associate bones influence with reference poses influences. We assume that skinning weights are convex, resulting in a simple equation (influence of the pose $\mathcal{P}_k$ for vertex $\boldsymbol{v_{jf}}$ at frame f):

$$Inf\left(\boldsymbol{v_{jf}}, \mathcal{P}_k\right) = \sum_{i=1}^{n} \left(w_{ji} \times \mathcal{I}_{\mathcal{P}_k}\left(\mathcal{J}_{if}\right)\right)$$

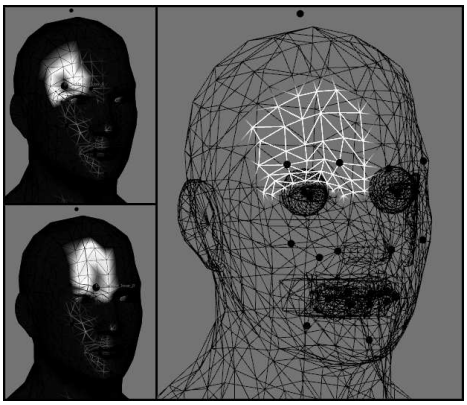

Fig. 2. The two left images show the skinning influences of the two bones of the right eyebrow. A reference pose has been set with this two bones rising up. The last image shows the influence of this reference pose for each vertex attached to these bones.

Skeletons may greatly move along applications, complex animations require a lot of bones per face. This could generate some redundancy between bones transformations. If CG artist defines a pose which require a similar displacement of 3 adjacent bones, a wrinkle would appear along these 3 bones areas. At runtime, it may appears artifacts or wrinkle breaks if the 3 bones don't move similarly. To avoid this, we simply give the possibility to group some bones together. Their transformations still leave independent, but their poses weights are linked, *i.e.* the average of their coefficients is use instead of their initial values.

5.3 Details Blending

The final step of our method is to apply wrinkle maps to our mesh by using coefficients $Inf(\boldsymbol{v_{jf}}, \mathcal{P}_k)$ computed at the next step. Two methods can be used depending on how wrinkle maps have been generated:

- Wrinkle maps are neutral map improvements (*i.e.* details of the neutral map such as pores, scares and other static fine details are present in the wrinkle map).
- Wrinkle maps only contain deformations associated with the expression.

In the first case, a simple blending is used. Since same static details are present in both neutral and wrinkle maps, a blending will not produce a loss of static deformations while in the second case, a simple averaging will cause it. For example, a fragment influenced by 100 percents of a wrinkle map will be drawn without using the neutral map, resulting in the fact that details of the neutral map will not appear. A finest blending is required. [1] proposed one for normal map in tangent space, let $\overrightarrow{WN}$ the final normal, $\boldsymbol{W}$ the normal provided by the blending of the wrinkle maps and $\boldsymbol{N}$ the normal provided by the default normal map:

$$\overrightarrow{WN} = normalize\left(\boldsymbol{W}.x + \boldsymbol{N}.x, \boldsymbol{W}.y + \boldsymbol{N}.y, \boldsymbol{W}.z \times \boldsymbol{N}.z\right)$$

The addition of the two first coordinates makes a simple averaging between the direction of the two normals, given the desired direction. The z components are multiplied, this leads to increase details obtained from the two normals. Smaller the z value is, and more bumpy is the surface, multiplication allows to add the neutral surface variation to the wrinkled surface.

6 Results and Conclusion

Our test model *Bob* contains 9040 triangles for the whole body which is a current size for video-games. The face is rigged with 21 bones. Animation runs at more than 100 fps on a laptop equipped with a Dual Core 2.20GHz CPU, 2Go RAM and a NVidia 8600MGT GPU. Rendering is done using OpenGL and NVidia Cg for GPU programming. We use tangent space normal maps as wrinkle maps in our experiments. We focus our tests on the forehead wrinkles because they are the most visible expressive wrinkles and generate the higher visual deformations. Furthermore, this area is subjected to different muscles actuations, compressed and stretched expressions generate different small surface variations. Similar to [20], we use a feature-points based animation transfer to animate our characters. This facial animation retargeting algorithm is based on feature-points displacements and Radial Basis Function regression (Fig 3).

Figure 4 shows 4 facial expressions with and without dynamic wrinkling. By analyzing the top row, you may notice that expression recognition is not easy. The neutral and the angry expressions of the eyebrows are not very distinct. However, the bottom row shows that wrinkles improve the expression recognition. Figure 5 shows the independence between facial areas without additional mask maps.

Fig. 3. Example frames of an animation providing by a 2D tracker and transfer in real-time to our skinned face improved with our wrinkling system.

Pose evaluation is done on the CPU, resulting coefficients are send to the GPU. The *bones Masks* step is perform into the vertex shader, *i.e.* skinning weights are multiplied by poses coefficients, resulting in o values (one value for each reference pose). The rasterization step is done and we obtain the interpolated values for each fragment in the pixel shader where we use these coefficients to blend normal maps and compute the lighting. Large scale deformation as well as the rendering are not modified, the addition of our method to an existing implementation is easy. No additional rendering pass is required, only few functions should be add to the different steps cited above. Data send to GPU equals $o \times n$ floating values with o the number of reference poses and n the number of bones.

Our technique is greatly artist-dependent. Three steps are important to obtain good results. First, a good rigging is primary since we directly use skinning weights as bones masks, and so, it defines how each vertex will be influenced by the different reference poses. Second, the reference poses should consist on an orthogonal set of skeleton poses as much as possible, to avoid an over fitting. Notice that blending reference poses in a same area is possible (last column of Fig. 4), but it becomes a problem if similar bones displacements lead to different fine-details. Finally, detail maps quality greatly influences the visual results.

Our choice to use skinning weights as bones masks offers many advantages. They allow us to relate large scale and small scale deformations, and so, we do not need additional mask textures. They ensure that vertices influenced by reference poses are vertices which are displaced accordingly with bones too. However, weights become smaller when they are far away from the bones location, and so wrinkles depth become smaller too, even if they should be as visible as those near bones.

We have presented a technique to use pre-generated reference poses to generate in real-time wrinkles and fine-details appearing while an arbitrary skinned face animation. In addition to providing interesting visual results, the requirements that we considered necessary and/or important have been met. Our dynamic animation wrinkles runs in real-time, the use of per-pixel lighting allows us to dispense with high-resolution meshes or costly subdivision techniques. Furthermore, it is based on widely-used techniques such as skinning and bump

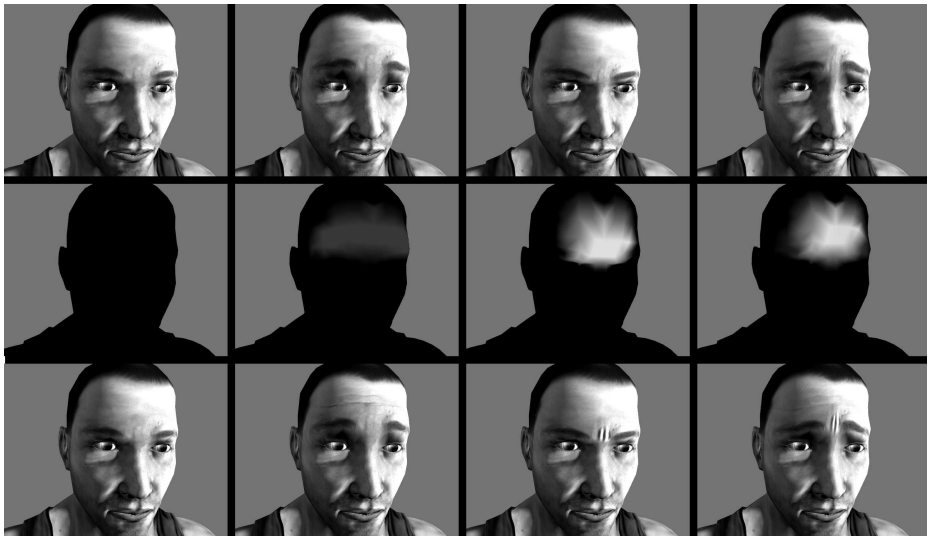

Fig. 4. The first row shows our character without dynamic wrinkles. The second rows shows reference poses influences. The last row shows our character with details. Notice that the second and the third column define the two reference poses.

mapping. Its implementation does not present technical difficulties and does not modify usual animation and rendering pipeline. However, results depend greatly of the quality of the input data provided by the CG artist. We plan to investigate this issue by developing specific tools to help him/her in the reference poses creation.

References

1. Oat, C.: Animated wrinkle maps. In: ACM SIGGRAPH 2007 courses. (2007) 33–37
2. Wu, Y., Kalra, P., Magnenat-Thalmann, N.: Simulation of static and dynamic wrinkles. In: Proc. Computer Animation 96. (1996) 90–97
3. Boissieux, L., Kiss, G., Magnenat-Thalmann, N., Kalra, P.: Simulation of skin aging and wrinkles with cosmetics insight. In: Proc. of Eurographics Workshop on Animation and Simulation. (2000)
4. Kono, H., Genda, E.: Wrinkle generation model for 3d facial expression. In: ACM SIGGRAPH 2003 Sketches & Applications. (2003) 1–1
5. Venkataramana, K., Lodhaa, S., Raghava, R.: A kinematic-variational model for animating skin with wrinkles. Computers & Graphics **29** (2005) 756–770
6. Tu, P.H., Lin, I.C., Yeh, J.S., Liang, R.H., Ouhyoung, M.: Surface detail capturing for realistic facial animation. J. Comput. Sci. Technol. **19** (2004) 618–625
7. Lo, Y.S., Lin, I.C., Zhang, W.X., Tai, W.C., Chiou, S.J.: Capturing facial details by space-time shape-from-shading. In: Proc. of the Computer Graphics International. (2008) 118–125

Fig. 5. This figure demonstrates that reference poses influences are independent between areas of the face. Only the right area of the forehead is influenced by the stretch wrinkle map on the middle image while the whole forehead is influenced on the right image.

8. Bickel, B., Botsch, M., Angst, R., Matusik, W., Otaduy, M., Pfister, H., Gross, M.: Multi-scale capture of facial geometry and motion. In: ACM Trans. Graph. Volume 26. (2007)
9. Bickel, B., Lang, M., Botsch, M., Otaduy, M., Gross, M.: Pose-space animation and transfer of facial details. In: Proc. of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. (2008)
10. Zhang, L., Snavely, N., Curless, B., Seitz, S.M.: Spacetime faces: High-resolution capture for modeling and animation. In: ACM Annual Conference on Computer Graphics. (2004) 548–558
11. Na, K., Jung, M.: Hierarchical retargetting of fine facial motions. In: Computer Graphics Forum. Volume 23. (2004) 687–695
12. Volino, P., Magnenat-Thalmann, N.: Fast geometrical wrinkles on animated surfaces. In: Proc. of the 7-th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media. (1999)
13. Bando, Y., Kuratate, T., Nishita, T.: A simple method for modeling wrinkles on human skin. In: Proc. of the 10th Pacific Conference on Computer Graphics and Applications. (2002) 166–175
14. Noh, J.Y., Neumann, U.: Expression cloning. In: ACM Trans. Graph. (2001) 277–288
15. Larboulette, C., Cani, M.P.: Real-time dynamic wrinkles. In: Computer Graphics International. (2004)
16. Lewis, J.P., Cordner, M., Fong, N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In: ACM Trans. Graph. (2000) 165–172
17. Kry, P.G., James, D., Pai, D.: Eigenskin: Real time large deformation character skinning in hardware. In: Proc. of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation. (2002) 153 – 160
18. Kurihara, T., Miyata, N.: Modeling deformable human hands from medical images. In: Proc. of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation. (2004) 357– 365
19. Rhee, T., Lewis, J.P., Neumann, U.: Real-time weighted pose-space deformation on the gpu. In: Computer Graphics Forum. Volume 25. (2006) 439–448
20. Dutreve, L., Meyer, A., Bouakaz, S.: Feature points based facial animation retargeting. In: Proc. of the 15th ACM Symposium on Virtual Reality Software and Technology. (2008) 197–200