EUROGRAPHICS 2019/ O. Bimber and A. Fusiello *Poster*

Towards Point-based Facial Movement Simulation

T. Uhlmann and G. Brunnett

[1]Chemnitz University of Technology, Faculty of Computer Science, Germany

Abstract

Lifelike animated characters are hard to create and involve extensive manual effort during generation of the geometry, adding a rig to the model, and creating the actual movements of limbs and tissue. Blendshapes is the leading technique regarding facial animations. This very simple mathematical model can be computed efficiently, thus it is suited for real time visualization. On the other hand it is hard to control, requires a skilled animator, and is still very error prone. Point-based simulation techniques became increasingly popular in recent years for animation of elastic surfaces, but have barely been used for facial animation. We argue, that this kind of technique is much more suited for simulation and animation of the non-rigid deformations of facial tissue. We build our proposal on a meshless finite elements technique, which has already been used successfully for the elastic simulation of surface models in real time and describe extensions to the approach so it can be applied for facial animation.

1. Introduction

Animation of human characters and facial animation is in the focus of computer scientists for more than three decades and is still an open research topic. While movie and gaming companies show impressive results when in comes to realistic facial expressions and deformations, these are the result of extensive manual effort, expensive, specialized equipment, and offline rendering techniques. The Δ-blendshapes technique is almost ubiquitous in facial animation. It is a very simple technique which adds a weighted sum of vector deformation fields to a base mesh in order to reproduce an elastic deformation. This is computational inexpensive and renders the technique applicable for real time rendering. Although each blendshape affects the geometry of the surface, there is no connection between them in a computational manner. This makes it hard to detect errors algorithmically such as typical self-intersection issues as depicted in figure 1. Creating a high quality rig for a face is therefore very difficult, since even slightest inaccuracies can be detected by a human observer, but can occur easily when combining blendshapes. A point-based simulation does no suffer from these shortcomings, since only a single set of control nodes defines the deformation.

Several approaches to facial animation have been proposed over the years. To the best of our knowledge the only point-based simulation technique was proposed by Bickel et al. [BLB*08]. The authors used a thin shell simulation method to simulate large and fine scale deformations of the face. We like to refer the reader to Ersotelos and Dong [ED08] for a comprehensive overview on facial animation techniques. Bender et al. [BMO*14] published an extensive survey on point-based simulation methods employed by the computer graphics community.

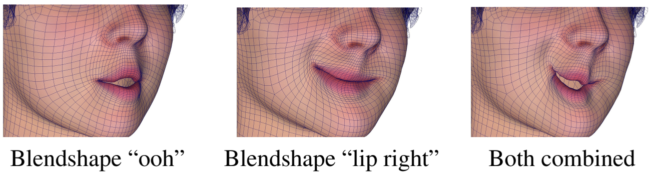
Blendshape "ooh" Blendshape "lip right" Both combined

Figure 1: *A typical artifact created by the blendhspaes technique on a professional rig. Combining two blendshapes can already result in self-intersections of the mesh and unrealistic deformations.*

2. A Meshless Finite Elements Approach for Facial Movement Simulation

Our approach is similar in spirit to Adams et al. [AWO*10], who successfully created an elastic deformation simulation of triangle surface meshes. The basic idea is to use a sparse set of control nodes directly placed on the surface. Their influence on the mesh's vertices is defined by weights. Then a deformation field is simulated to transfer the forces which act on the nodes onto the vertices. As result the vertices are dragged along when the nodes are displaced from their initial position, hence creating an elastic and smooth deformation of the surface. A shape function is computed for each joint-vertex pair during the creation of the model, which defines how a vertex is displaced by each control node. During runtime, a simple sum of weighted vectors needs to be computed and applied to the vertex's initial position. Hence, it is computational as efficient as the blendshape technique and can be implemented as a GPU shader program.

The resulting research questions are therefore how the control

DOI: 10.2312/egp.20191049

nodes have to placed, how appropriate weights can be found, and how actual animations can be created and controlled.

The MPEG-4 international standard already contains extensive definitions of facial feature points and facial animation parameters. Hence, using the Face Definition Parameters as control points will be the first choice. Valid research questions are, whether a subset of these feature points may suffice for a given set of facial movements and if they are dense enough for meshes of arbitrary resolution.

Determining appropriate weights automatically will require a sophisticated optimization algorithm. A simple estimation of weights using the geodesic distance as proposed by Adams et al. [AWO*10] can be an initial guess, but will not suffice to represent the surface tissue movement caused by the underlying 43 muscles of a real face. Hence, the weights have to be optimized, so that all possible deformations can be executed correctly. This is similar to finding linear blend skinning weights, which is a difficult and tedious task if done manually. Assuming we have a triangle mesh of a neutral facial expression and deformations of the base mesh showing particular facial expressions. The control nodes are positioned and weights are initialized based on geodesic distance. Deformation fields have to be computed, which can reconstruct all given facial expressions. This problem definition renders finding the weighs manually impossible, since we have two sets of unknown parameters. The displacement vectors of the nodes on the one hand and the weights on the other hand. We propose to use an alternating optimization approach as it is known from non-rigid template fitting [JQL*17]. Two nested loops can be employed. Positions of the control nodes are optimized in the outer one while the weights are fixed. The inner loop optimizes weights while positions are fixed. An analysis-by-synthesis approach can be used to control the optimization. The deformed mesh is created by the current set of parameters and compared to the target mesh by their geometric distance. The resulting error value will drive the optimization. It can be assumed, that the function to be optimized will exhibit a considerable amount of local minima. Therefore, simple gradient descent will not suffice and methods such as simulated annealing or genetic algorithms have to be used to find the global minimum. It is also desirable to develop a template model which can be individualized later to new facial meshes, since this optimization process is computationally demanding.

For practical applications it would be beneficial if the point-based model can be converted to a blendshape model, and vice versa. The approach described earlier is exactly the conversion from the blendshape model to the point-based one. Conversion in the other direction is an easy task. The point-based method also uses a base mesh. Deformed meshes can be created by applying the respective animation data at full strength to create blend targets, which will always be in vertex correspondence with the base mesh. Creating the blendshape model is a simple matter of building the displacement fields.

In contrast to blendshapes, the individual facial movements will not affect the vertices of the mesh directly, but only the control nodes, hence omitting artifacts such as self-intersection of the mesh. Limiting the control nodes' total range of motion will prohibit implausible vertex displacement typically observed when using blendshapes. The sparse set of control nodes also creates an easier to control rig for animations, due to the reduced degrees of freedom.

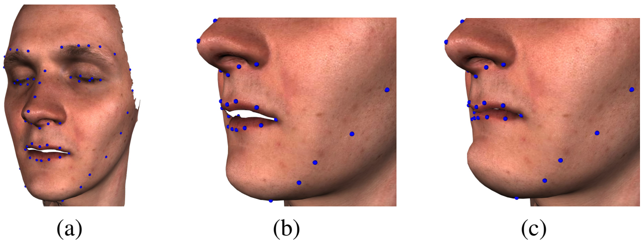
(a) (b) (c)

Figure 2: *(a) We use a sparse set of automatically detected facial feature points on a 3D-scanned human face. (b) Mouth in neutral position. (c) Closing the mouth by displacing the feature points surrounding the lips.*

3. Proof of Concept and Future Work

To proof our concept we implemented the meshless finite elements approach of Adams et al. [AWO*10] and used a 3D-scan of a human face. We automatically detect a sparse set of feature points employing state of the art facial landmark detection [ALS16]. These feature points become the control nodes driving the deformation. Figure 2 shows the feature point set on our test model and how facial deformation can be created. The resulting deformations are smooth, hence realistically reproducing the deformation of facial tissue. We currently use weights which are purely based on the geodesic distance from the control node. This does not offer sufficient control of the facial geometry. Therefore we are planning to elaborate on the described analysis-by-synthesis approach to create fine-tuned weights which offer a better level of control.

References

[ALS16] Amos B., Ludwiczuk B., Satyanarayanan M.: *OpenFace: A general-purpose face recognition library with mobile applications.* Tech. rep., CMU-CS-16-118, CMU School of Computer Science, 2016. 2

[AWO*10] Adams B., Wicke M., Ovsjanikov M., Wand M., Seidel H.-P., Guibas L. J.: Meshless shape and motion design for multiple deformable objects. *Computer Graphics Forum 29*, 1 (2010), 43–59. doi:10.1111/j.1467-8659.2009.01536.x. 1, 2

[BLB*08] Bickel B., Lang M., Botsch M., Otaduy M. A., Gross M.: Pose-Space Animation and Transfer of Facial Details. In *Eurographics/SIGGRAPH Symposium on Computer Animation* (2008), Gross M., James D., (Eds.), The Eurographics Association. doi:10.2312/SCA/SCA08/057-066. 1

[BMO*14] Bender J., Müller M., Otaduy M. A., Teschner M., Macklin M.: A survey on position-based simulation methods in computer graphics. *Computer Graphics Forum 33*, 6 (2014), 228–251. doi:10.1111/cgf.12346. 1

[ED08] Ersotelos N., Dong F.: Building highly realistic facial modeling and animation: a survey. *The Visual Computer 24*, 1 (Jan 2008), 13–30. doi:10.1007/s00371-007-0175-y. 1

[JQL*17] Jiang T., Qian K., Liu S., Wang J., Yang X., Zhang J.: Consistent as-similar-as-possible non-isometric surface registration. *The Visual Computer 33*, 6 (Jun 2017), 891–901. doi:10.1007/s00371-017-1390-9. 2