

MODELING AND ANIMATING THREE DIMENSIONAL DETAILED FACIAL EXPRESSIONS

Alice J. Lin and Fuhua (Frank) Cheng*
Department of Computer Science
University of Kentucky
Lexington, KY 40506, USA
ajlin0@cs.uky.edu, cheng@cs.uky.edu

ABSTRACT

Two categories of methods for generating detailed animated facial expressions are presented. The first category is to apply pattern functions to the surface of a facial model. The pattern functions allow a user to directly manipulate them and see immediate results, and allow expressive features to be applied to any existing animation or model. The location and style of an expressive feature can be specified, thus allowing the detailed facial expressions to be modeled as desired. The technique can also be applied to a variety of CG human body skins and animal faces.

The second category is to create tears, which combined with facial models can generate detailed, animated, emotional expressions. The category contains two unique methods for generating realistic, animated tears. One method is for generating teardrops, which continually change shapes while dripping down the face. The other is for generating shedding tears, which seamlessly connect with the skin as they flow along the surface of the face. The method provides real-time, vivid simulation of tears rolling down the face. These methods both broaden CG and increase the realism of facial expressions. The methods can also be applied to CG human body skins and other surfaces to simulate flowing water.

KEY WORDS

3-Dimensional Modelling, Mathematical Modelling, Facial Expressions, Tears, Animation

1. Introduction

Human beings are very accustomed to looking at real faces and can easily spot artifacts in computer-generated models. Ideally, a facial model should be indistinguishable from a real face. Generating compelling animated facial expressions is an extremely important and challenging aspect of computer graphics. The most important aspect of character animations is the character's ability to use facial expressions. Although well designed

and animated in recent 3D games, virtual reality, computer vision or animation movies, computer graphics humans are not fully satisfying since they lack important details, which include detailed surface geometry and emotional expressions: tears.

Detailed surface geometry contributes greatly to the visual realism of 3D face models. The more the detailed facial expressions are enhanced, the better the visual reality will be. However, it is often tedious and expensive to model facial details, which are authentic and real-time. We present a functional approach for adding detailed animated facial expressions to facial models.

Emotions play a critical role in creating engaging and believable characters. Facial movements and expressions are therefore the most effective and accurate ways of communicating one's emotions. Emotions and feelings can be difficult at times to put into words but our behavioral reactions to these feelings are universally accepted as being similar and easily noticeable. Although weeping and tears are a common concomitant of sad expressions, tears are not indicative of any particular emotion, as in tears of joy. A happy face with tears appears more joyous, or something in between, perhaps described as bittersweet. As far as scientists can tell, no other creature cries emotional tears the way humans do. The tear effect appears unique to our species. With tears, the face increases the richness as an instrument for communication. The way humans read other people's emotions is markedly impacted by the presence (or absence) of tears.

Provine et al. [1,2] provided the first experimental demonstration that tears are a visual signal of sadness by contrasting the perceived sadness of human facial images with tears against copies of those images that had the tears digitally removed. Tear removal produced faces rated as less sad. The research subjects suggest further that after tear removal, the faces' emotional content showed uncertainty, ambiguity, awe, concern, or puzzlement, and not just a decrease in sadness. Sometimes, after the removal of the tears, the faces do not look sad at all and instead look neutral. Recently, the increasing appearance of virtual characters in computer games, commercials, and movies makes the facial expressions even more important.

*Fuhua Cheng is currently visiting the National Tsinghua University in Taiwan.

Computer graphics researchers have been greatly interested in this subject and we have seen considerable innovation in 3D facial expressions during the last decade. However, there have been no theories and methods of making *dramatic or realistic* animated 3D tears. In the past, the artist usually used various textures to accentuate the static tears. Since tears, which consist mostly of water, has high specular reflection and transparency, those antiquated texture tears cannot achieve this degree of realism. We thus present the methods to simulate real-time shedding of tears.

2. Related Works

Facial modeling and animation research falls into two major categories, those based on geometric manipulations and those based on image manipulations. Each realm comprises several subcategories. Geometric manipulations include key-framing and geometric interpolations [3, 4], parameterizations [5-7], finite element methods [8, 9], muscle based modeling [10-14], visual simulation using pseudo muscles [15, 16], spline models [16-19] and free-form deformations [20, 15]. Image manipulations include image morphing between photographic images [21], texture manipulations [22], image blending [23], and vascular expressions [24]. At the preprocessing stage, a person-specific individual model may be constructed using anthropometry [25], scattered data interpolation [26], or by projecting target and source meshes onto spherical or cylindrical coordinates. Such individual models are often animated by feature tracking or performance driven animation [27-29]. Marker-Based Motion Capture is combining 3D facial geometry with marker-based motion capture data [29]. The commercial marker-based facial-capture systems acquire data with excellent temporal resolution, but due to their low spatial resolution, they are not capable of capturing expression wrinkles. Oat [30] presented a technique that involves compositing multiple wrinkle maps and artist animated weights to create a final wrinkled normal map that is used to render a human face. Wrinkle maps are bump maps that get added on top of a base normal map. Pighin et al. [31] presented techniques for creating 3D facial models from photographs of a human subject, and for creating transitions between different facial expressions by morphing between these different models. Bickel et al. [32] acquired a static scan of the face, and augmented a traditional marker-based facial motion-capture system by two synchronized video cameras to track expression wrinkles. The lack of a strict vertex correspondence between wrinkles and mesh over time could lead to minor drifts of the wrinkles. Zhang et al. [33] presented a geometry-driven facial expression system, which includes a framework for synthesizing facial expression textures from facial feature point motions by using an example-based approach.

Simulation of water in real-time use particle-based methods is likely the optimal choice and has become

popular in recent years. Chentanez et al. [34] combined grid-based and particle-based approaches to demonstrate a hybrid water simulation, which included a large flowing river along a valley with beaches, big rocks, steep cliffs and waterfalls. Pfaff et al. [35] presented an algorithm, which is highly suitable for massively parallel architectures, and was able to generate turbulent simulations with millions of particles. Particles are typically generated according to a controlled stochastic process. Splashing, foaming, and breaking waves are complex processes best modeled by particle systems, but the technique is inefficient in nonturbulent situations. These particles are either created or initialized with meaningful positions and velocities before the simulation starts or generated during the simulation by emitters. Particle systems simulate certain fuzzy phenomena. The parameters in the simulation are all or mostly fuzzy values. The animated tears as a part of the facial expressions need precise numeric values for its parameters. Otherwise, the tears will not look realistic at all.

3. Adding Detailed Animated Expressions

To generate detailed animated expressions, we apply pattern functions to the facial models. We defined the pattern function as a combination of primary function P and damping. Primary function $P = p_1 + p_2 + \dots$, can be only one basic pattern function p_1 or several basic pattern functions p_i , $i = 1, 2, 3, \dots$. The pattern function is layered on top of the selected area. It describes the surface of the skin change. Since the pattern function adjusts the vertices sequentially, the form and shape in the selected area changes over time. Adjusting the parameters of the pattern functions precisely controls the shape of the expression. Moving the layer of pattern functions refines the expression location. The detailed movement of the expression is a function of the time parameter t in each time interval. Damp function influences the smoothness around the selected area where we applied expression patterns. The damp function dictates expression patterns to decrease and then vanish toward the borders of the selected region.

3.1 Wrinkles

For human faces, wrinkles can roughly be classified into two different types: aging wrinkles and expressive wrinkles. Expressive wrinkles are temporary wrinkles that appear on the face during expressions at all ages. They can appear anywhere on the face during expressions, but for most people, the expressive wrinkles mainly come forth on several regions of the face such as the cheeks, the forehead, and eye corners. From these effects, we have developed a set of basic pattern functions.

Forehead wrinkles:

The pattern function of forehead wrinkles is shown below. Figure 1 shows six frames of a running cycle of this pattern. We applied this pattern function to the forehead surface (Figure 2(a)). The facial model with wrinkles in the forehead is shown in Figure 2 (b). We also applied it to Figure 4(a) and got the result in Figure 4(b).

$$p = e^{\alpha \times \left| \sin((x^2 + y^2) - t \times \sin(\arctan(\frac{y}{x})) - \sin(\beta \times \arctan(\frac{y}{x}))) \right|} - z$$

α and β are parameters. t is time.

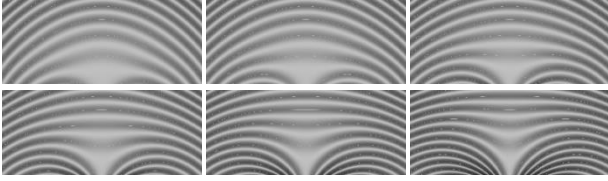


Figure 1: Six frames of a running cycle.

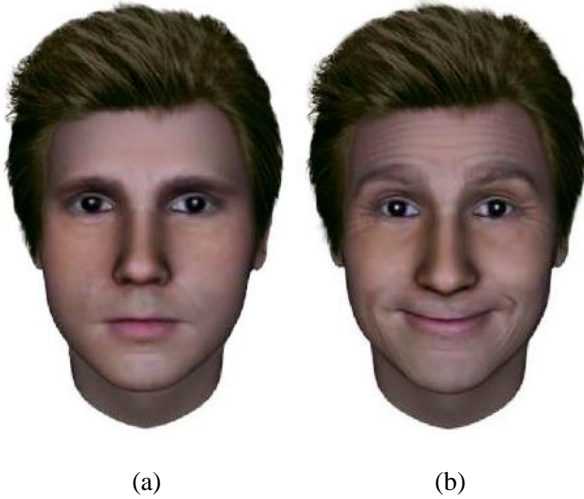


Figure 2: (a) Neutral facial expression. (b) Application of forehead wrinkles, eye corner wrinkles and cheek wrinkles.

Eye corner wrinkles:

For eye corner wrinkles, we developed the pattern function, which is shown below. Figure 3(a) is the pattern for eye corner wrinkles. We applied the pattern to both eye corners (Figure 2 (a)). The result is shown in Figure 2 (b).

$$p = \alpha \times e^{\beta \times \left| \sqrt{x^2 + y^2} - t \times \sin(\gamma \times \arctan(y/x)) \right|} - z$$

α , β and γ are parameters. t is time.

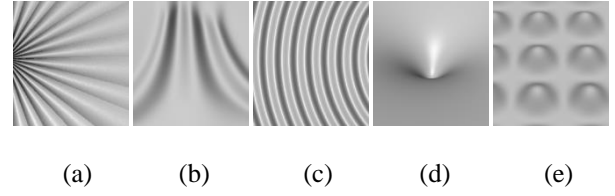


Figure 3: Created pattern functions. (a) Eye corner wrinkles (b) Frown wrinkles (c) Cheek wrinkles (d) Dimple (e) Eye pouch effects.

Cheek wrinkles:

The pattern function we developed for cheek wrinkles is shown below. Figure 3(c) is the pattern for cheek wrinkles. We add pattern functions to cheeks to improve mouth animations and make the face look alive. This achieves the realism of cheek skin. An example is shown in Figure 2 (b).

$$p = \alpha \times \sin(\beta \times \sqrt{x^2 + y^2} + t) - z$$

α , β are parameters. t is time.

Frown wrinkles:

The equation below and Figure 3(b) is the pattern function we created for the frown. An example of applying this function to Figure 4(a) is shown in Figure 4 (c).

$$p = \alpha \times \sin(x \times y / (\beta + t)) - z$$

α , β are parameters. t is time.

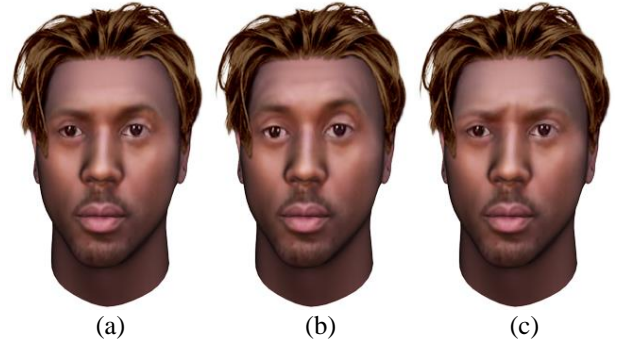


Figure 4: (a) Neutral facial expression. (b) Forehead wrinkles. (c) Frown.

3.2 Dimple and Eye Pouch Effects

Dimple:

Dimples are visible indentations of the skin. In most cases, facial dimples appear on the cheeks, and they are typically not visible until someone smiles. The pattern function we developed for a dimple is shown below. Figure 3(d) is the pattern for dimple. We applied the dimple function in Figure 5 (a), and generated the result in Figure 5 (b)

$$p = \log(x^2 + y^2 + t) - z$$

t is time.

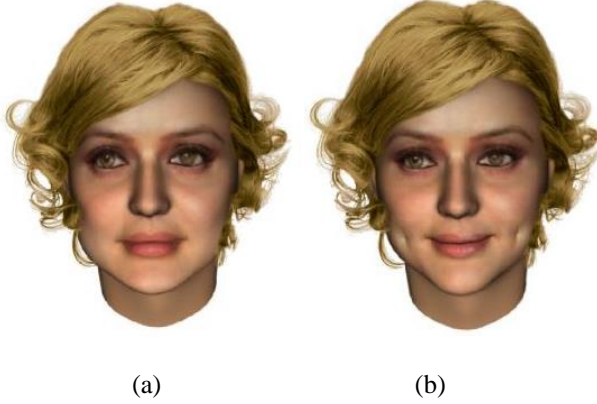


Figure 5: (a) Neutral expression. (b) Applied dimple function on cheeks.

Eye Pouch Effects:

In 3D facial expressions, people usually either exaggerate the emotion or make a straight face. Emotions, such as anger or sadness, are usually expressed as opening the mouth and eyes widely and furrowing the eyebrows or deforming the face. Subtle facial expressions also express a person's emotion. To show an angry or sad person, for instance, we can use facial skin trembling on different regions, such as the eye pouches, chin, cheeks or temple. A pattern function that we have developed for eye pouch effects is shown below:

$$p = \frac{|\cos(x + \alpha \times t) + \cos(y + \beta \times t)|^\gamma}{t} - z$$

Where α , β and γ are parameters and t is time. The degree of impact can be manipulated by adjusting parameters of the equation. The pattern is shown in Figure 3(e). Figure 6 shows a frame of a running cycle.



Figure 6: Eye pouch effects.

4. Generating Animated Tears

4.1 Dripping tears

There are two kinds of tear animations. One is the dripping tear (Figure 11), where the tear falls quickly to the ground, rather than falling slowly down the skin. The teardrop continually changes shape and falls as time increases. The formula we have developed to generate this kind of tear is:

$$\begin{cases} x = \frac{\alpha \times \sin(u) \cos(v)}{t \times \cos(u)} \\ y = \frac{\alpha \times \sin(u) \sin(v)}{t \times \cos(u)} \\ z = \beta \times t \times \cos(u) + \lambda \times t \end{cases}$$

α , β and λ are preset parameters. α and β describe the sizes of the tear, λ describes the speed of the tear dropping. t represents the time of animation. u and v are parameters. Figure 7 presents a visual representation of this phenomenon. When the teardrop is falling, time increases and the teardrop's shape continually change from left to right (Figure 7). Figure 11 is an example of dripping tears on a child's face. The example shows four frames of a running cycle.

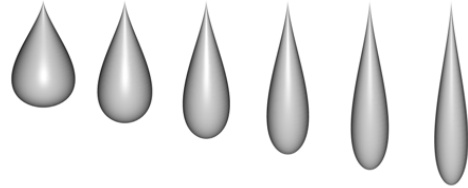


Figure 7: Six selected frames of a running cycle of a teardrop changing shape.

4.2 Shedding tears

Another type of tear is the shedding tear. This kind of tear flows along the surface of the skin on the face (Figure 12). The tear travels over the skin, but it is an individual object. The tear has to become seamlessly connected with the skin and will roll down as time passes. The procedure for generating shedding tears that we have created is the following:

Step one: We select the areas that the tear will pass through on the face, and then extract this area as an individual surface. Since the face mesh generally is not fine enough for detailed simulation and smoothness, we subdivide the extracted surface to refine meshes, and call it the base surface (Figure 8 (a)).

Step two: From the outline of the base surface, we construct the object - the tear will wrap around on the base surface. Figure 9 is a simple illustration. The red plane is the base surface. The black plane (top surface) is a surface parallel to the base surface (red). The remaining planes in Figure 9 are constructed surfaces that all connect to both parallel surfaces (red and black). Also see Figure 8 (b) and (c).

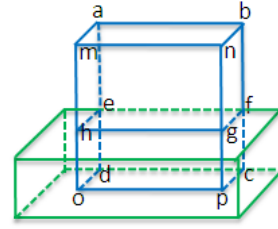


Figure 10: Making the head of the tear.

Step three: Figure 10 shows how we make the head of the tear when the tear is rolling down the face. The plane $abcd$ represents a base surface that is the same as the red plane in Figure 9. The plane $mnpq$ is the top surface and is the same as the black plane in Figure 9. Now, we extrude the planes $gfcp$, $hgpo$ and $hedo$. The result is shown by the green region. The green part of the object is the head of the tear (also see Figure 8 (d)). Then, we smooth the head of the tear (green region), giving us the result as Figure 8(e).

Step four: We randomly add noise to the tear surface to simulate the roughness of the skin. Generally, the geometric structure of the facial surface is smooth and the roughness of face skin is represented by texture. Tears have high specular reflection and transparency. With the smooth facial surface, the tear simulation will not properly achieve realism. Thus, we add random noise to the top surface (black in Figure 9) to reflect the facial skin roughness. In Figure 8(b) we see two parallel planes. It is only for demonstration. These two surfaces are seen as almost one surface in the simulation. They are very close, so we randomly choose vertices to rise from the top surface. The vertices cannot go down from the top surface; otherwise the tear may intersect with the face surface. The density of noise we add depends on the skin's roughness. The rougher skin will add more bumps. The top surface with the noise is shown in Figure 8 (f).

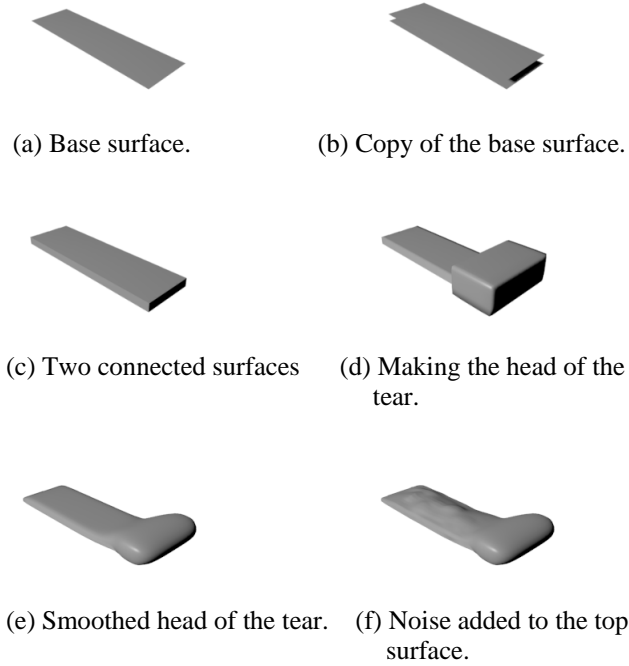


Figure 8: Generating shedding tears.

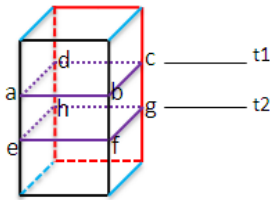


Figure 9: Constructed object (represents the tears).

When the top surface adds noises, the surface is deformed. The deformation is based on the method of moving least squares. Suppose that we have n points located at positions p_i . $P = \{p_i \in \mathbb{R}^d, d = 3\}$, $i \in \{1, 2, \dots, n\}$.

The surface function $f(x)$ is defined in arbitrary parameter domain Ω , which approximates the given scalar values f_i for a moving point $x \in \mathbb{R}^d$ in the MLS sense.

f is taken from \prod_r^d , the space of polynomials of total degree r in d spatial dimensions. $F(x)$ is a weighted least squares (WLS) formulation for an arbitrary fixed point in \mathbb{R}^d . When this point moves over the entire parameter domain, a WLS fit is evaluated for each point individually. Then, the fitting function $f(x)$ is obtained from a set of local approximation functions $F(x)$.

$$f(x) = \arg \min_{F \in \prod_r^d} \sum_{i=1}^n w_i (\|x - p_i\|) \|F(p_i) - f_i\|,$$

$$F(x) = b^T(x)c(x) = \sum_{j \in [1, m]} b_j(x)c_j(x)$$

then, $f(x)$ can be expressed as

$$f(x) = \min \sum_i w_i (\|x - p_i\|) \|b^T(p_i)c(x) - f_i\|^2$$

where, $b(x) = [b_1(x), b_2(x), \dots, b_m(x)]^T$ is the polynomial basis vector and

$c(x) = [c_1(x), c_2(x), \dots, c_m(x)]^T$ is the vector of unknown coefficients, which we want to resolve. The number m of elements in $b(x)$ and $c(x)$ is given by

$$m = \frac{(d+r)!}{d!r!}. w_i(\|x-p_i\|)$$
 is the weighting function by

distance to x . $w(\theta) = \frac{1}{\theta^2 + \varepsilon^2}$. Setting the parameter ε

to zero results in a singularity at $\theta = 0$, which forces the MLS fit function to interpolate the data.

Setting the partial derivatives of $f(x)$ to zero, we obtain

$$\frac{\partial f(x)}{\partial c_j(x)} =$$

$$2 \sum_i w_i(\|x-p_i\|) b_j(p_i) [b^T(p_i)c(x) - f_i] = 0$$

where $j = 1, \dots, m$, and we can get

$$\sum_i w_i(\|x-p_i\|) b(p_i) b^T(p_i) c(x)$$

$$= \sum_i w_i(\|x-p_i\|) b(p_i) f_i$$

If the matrix $A = \sum_i w_i(\|x-p_i\|) b(p_i) b^T(p_i)$ is not singular, i.e., its determinant is not zero, then

$$c(x) = A^{-1} \left(\sum_i w_i(\|x-p_i\|) b(p_i) f_i \right).$$

As a result,

$$f(x) = b^T(x) A^{-1} \left(\sum_i w_i(\|x-p_i\|) b(p_i) f_i \right)$$

Step five: For the animation of shedding tears, we assume Figure 9 is a path of tear flowing from the top of the object. When time starts (t_1), we select a certain portion of the vertices (see Figure 9, the vertices above the plane $abcd$) on the top side of the object. As time passes (t_2), the selected portion of the object increases in size (see Figure 9, the vertices above the plane $efgh$). For each time point, there is a corresponding set of vertices. We use the height of the tear path to indicate the covered portion of the vertices. Height is $h = s + c * t$, where s and c are constant, and t is time.

We used the above five-step procedure to make an example (Figure 12) of shedding tears that flow along the skin surface. The example shows four frames of a running cycle.

5. Conclusion

We have presented a novel approach for generating detailed, animated, realistic facial expressions by employing a set of pattern functions. Our method adapts to individual faces and allows the expressive features to be applied to any existing animation or model. The location and style can be specified, thus allowing the detailed facial expressions to be modeled as desired. The process for generating detailed facial expressions requires less human

intervention or tedious tuning. It also can be applied to a variety of CG human body skins and animal faces.

We have also presented another novel approach for generating realistic, animated tears. We have introduced a formula for generating teardrops, which continually change shape as the tear drips down the face. We also show the method and procedures for generating shedding tears. We extract the tear's path from the facial surface, and construct the three dimensional tear's object that is based on tear stains. We randomly add noise to the tear surface to simulate the roughness of the skin. We extrude and smooth the head of the moving tear to create real-time vivid rolling of the tears down the face. Our results indicate that these methods are robust and accurate. Our methods both broaden CG and increase the realism of facial expressions. It will evolve 3D facial expressions and animations significantly. Our methods can also be applied to CG human body skins and other surfaces to simulate flowing water. Our methods are not only for animation by a trained artist, but also for industry.

Acknowledgements

This work is supported by National Science Foundation of China (61020106001, 61170324), National Science Council of ROC (NSC-100-2811-E-007-021), and a joint grant of National Tsinghua University and Chang-Gung Memorial Hospital (101N2756E1).

References

- [1] R. R. Provine, K. A. Krosnowski, and N. W. Brocato, Tearing: breakthrough in human emotional signaling. *Evolutionary Psychology*, 2009, vol. 7(1), pp. 52-56.
- [2] R. R. Provine, Emotional tears and NGF: a biographical appreciation and research beginning. *Archives Italiennes de Biologie*, 2011, vol. 149(2), pp. 269-274.
- [3] A. Ennett, Digital portfolio: Tony de peltrie. *Computer Graphics World*, 1985, vol. 8(10), pp. 72-77.
- [4] F. I. Parke, Computer generated animation of faces. *Proc. ACM annual conf.*, 1972.
- [5] F. I. Parke, Parameterized models for facial animation revisited. *ACM SIGGRAPH Facial Animation Tutorial Notes*, 1989, pp. 53-56.
- [6] F. I. Parke, Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, 1982, vol. 2(9) pp. 61-68.
- [7] F. I. Parke, A parametric model for human faces, Ph.D. Thesis, University of Utah, Salt Lake City, Utah, 1974.
- [8] S. Basu, N. Oliver, and A. Pentland, 3D modeling and tracking of human lip motions. *ICCV*, 1998 pp. 337-343.
- [9] B. Guenter, A system for simulating human facial expression. *State of the Art in Computer Animation*, 1992, pp. 191-202.
- [10] N. Magnenat-Thalmann, H. Minh, M. Angelis, and D. Thalmann, Design, transformation and animation of human faces. *Visual Computer*, 1988, vol. 5 pp. 32-39.
- [11] S. Pieper, J. Rosen, and D. Zeltzer, Interactive Graphics for plastic surgery: A task level analysis and implementation. *Computer Graphics, Special Issue: ACM SIGGRAPH, Symposium on Interactive 3D Graphics*, 1992, pp. 127-134.
- [12] S. Platt, and N. Badler, Animating facial expression. *Computer Graphics*, 1981, vol. 15(3) pp. 245-252.

[13] D. Terzopoulos and K. Waters, Physically-based facial modeling, analysis, and animation. *J. of Visualization and Computer Animation*, March, 1990, vol. 1(4), pp. 73-80. [14] K. Waters. A muscle model for animating three-dimensional facial expression. *Computer Graphics (SIGGRAPH proceedings, 1987)* vol. 21 pp. 17-24.

[15] P. Kalra, A. Mangili, N. M. Thalmann, and D. Thalmann, Simulation of Facial Muscle Actions Based on Rational Free Form Deformations. *Eurographics 1992*, vol. 11(3), pp. 59-69.

[16] N. Magnenat-Thalmann, N. E. Primeau, and D. Thalmann, Abstract muscle actions procedures for human face animation. *Visual Computer*, 1988, vol. 3(5), pp. 290-297.

[17] M. Nahas, H. Hutric, M. Rioux, and J. Domey, Facial image synthesis using skin texture recording. *Visual Computer*, 1990, vol. 6(6) pp. 337-343.

[18] M. Nahas, H. Huitric, and M. Saintourens, Animation of a B-spline figure. *The Visual Computer*, 1988, vol. 3(5), pp. 272-276

[19] C. L. Y. Wang, and D. R. Forshey, Langwidere: A New Facial Animation System. *Proceedings of Computer Animation*, 1994, pp. 59-68.

[20] S. Coquillart, Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling. *Computer Graphics*, 1990, vol. 24, pp. 187-193.

[21] T. Beier, and S. Neely, Feature-based image metamorphosis. *Computer Graphics (SIGGRAPH proceedings 1992)*, vol. 26, pp. 35-42.

[22] M. Oka, K. Tsutsui, A. Ohba, Y. Jurauchi, and T. Tago, Real-time manipulation of texture-mapped surfaces. *SIGGRAPH proceedings, 1987*, pp. 181-188.

[23] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin, Synthesizing Realistic Facial Expressions from Photographs. *SIGGRAPH proceedings, 1998*, pp. 75-84.

[24] P. Kalra, and N. Magnenat-Thalmann, Modeling of Vascular Expressions in Facial Animation. *Computer Animation*, 1994, pp. 50-58.

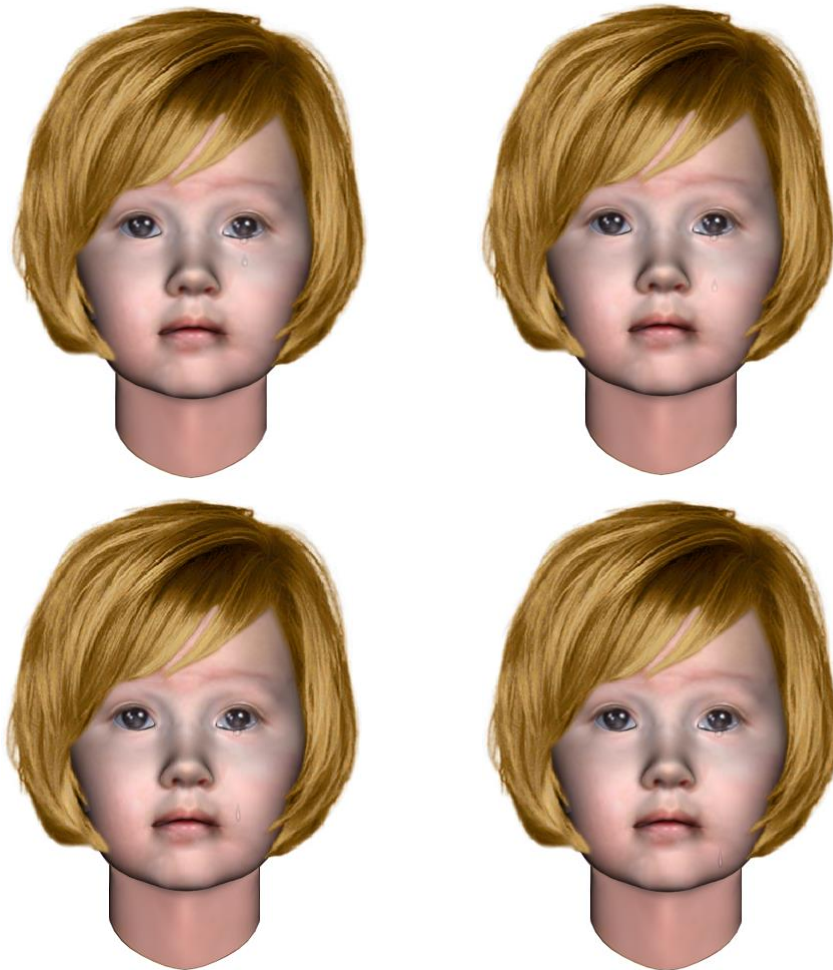


Figure 11: Four selected frames of a running cycle (dripping tears).

[25] T. Darrell, and A. Pentland, Spacetime gestures. *Computer Vision and Pattern Recognition*, 1993.

[26] D. Terzopoulos, and K. Fleisher, Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. *Computer Graphics, Proc. SIGGRAPH 1988*, vol. 22(4), pp. 269-278.

[27] I. A. Essa, and A. Pentland, Facial Expression Recognition using a dynamic model and motion energy. *Proc. of Int. Conf. on Computer Vision*, 1995, pp. 360-367.

[28] E. C. Patterson, P. C. Litwinowicz, and N. Greene, Facial Animation by Spatial Mapping. *Proc. Computer Animation 1991*, Springer-Verlag, pp. 31-44.

[29] L. Williams, Performance-driven facial animation. In *Proc. of ACM SIGGRAPH*, 1990, vol. 24, pp. 235-242.

[30] C. Oat, Animated wrinkle maps. *Advanced Real-Time Rendering in 3D Graphics and Games. SIGGRAPH Course*, 2007, pp. 33-37.

[31] F. Pighin, and J. P. Lewis, Performance-driven facial animation. *SIGGRAPH Course Notes* 2006.

[32] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister and M. Gross, Multi-Scale capture of facial geometry and motion. *ACM Transactions on Graphics*, 2007, vol. 26(3).

[33] Q. Zhang, Z. Liu, B. Guo, D. Terzopoulos, and H-Y Shum, Geometry-driven photorealistic facial expression synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 2006, vol. 12(1).

[34] N. Chentanez and M. Müller, Real-time simulation of large bodies of water with small scale details. *SCA '10 Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010, pp. 197-206.

[35] T. Pfaff, N. Thuerey, J. Cohen, S. Tariq and M. Gross, Scalable fluid simulation using anisotropic turbulence particles. *Proceedings of ACM SIGGRAPH Asia*, 2010 vol. 29(6).

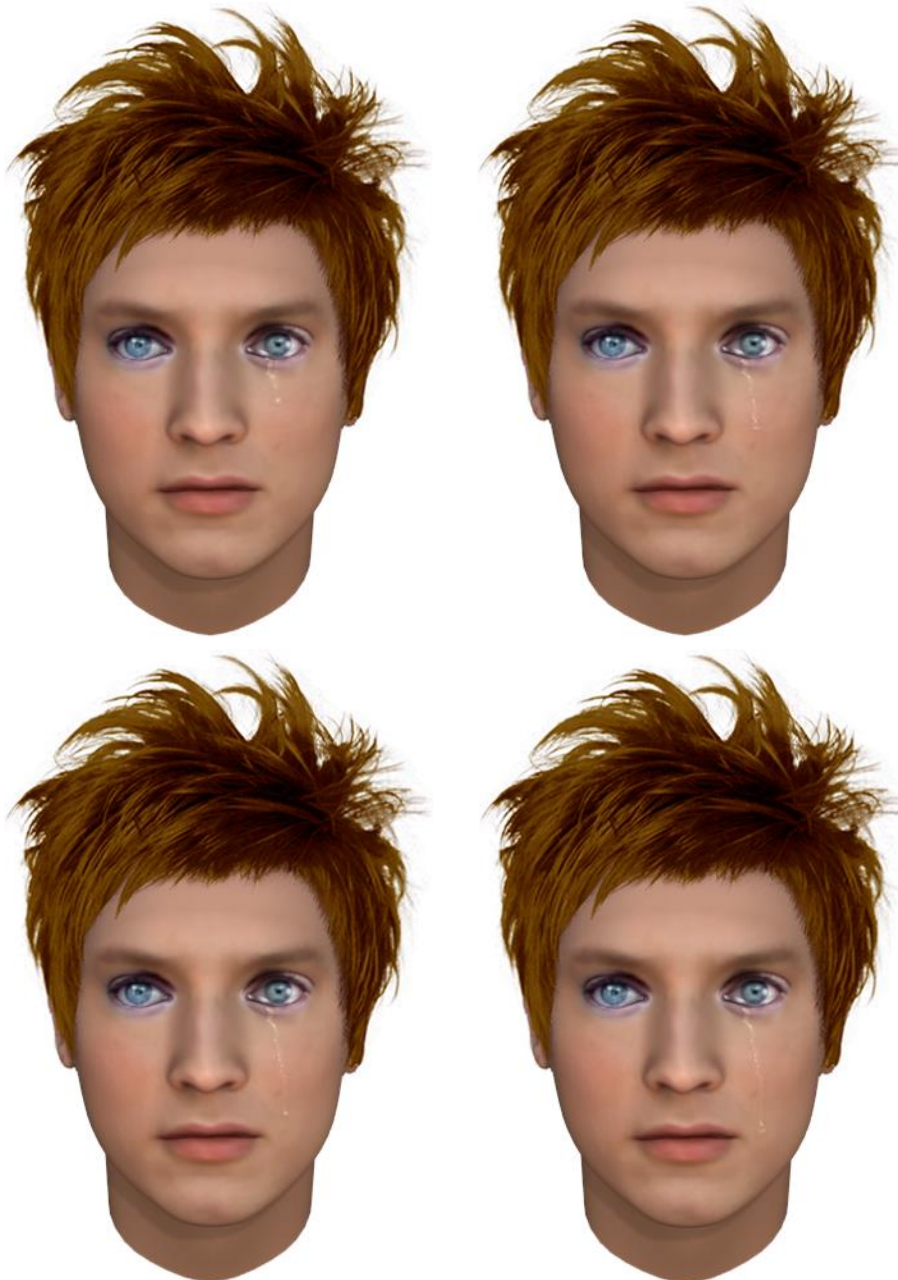


Figure 12: Four selected frames of a running cycle (shedding tears).