MODELING FACIAL EXPRESSIONS BY INTERACTIVE EVOLUTION

H.-F. HEIMBRECHT, G. TROGEMANN
Academy of Media Arts
Department of Computer Science / AV-Media
Peter-Welter-Platz 2
D-50676 Cologne, Germany

E-mail: fh@khm.de, trogemann@khm.de

Artists rarely have a specific goal in mind when they begin modeling 3D shapes and motions. Especially in facial animation for 3D-characters the process of modeling is actually the search for a goal rather than the attainment of one. This type of conceptualizing process is hardly supported by conventional 3D modeling techniques. We present a new way of creating facial expressions for 3D-character animation based on the concept of interactive evolution. Instead of deforming the facial topology directly, the creation process is controlled by mutating and mating computer generated facial motions. The genetic representation is based on Active Lines corresponding to special facial structures. Basic Motions are calculated from the Active Line with the help of a fast non-physical material-function. Facial expressions are animated in realtime by merging Basic Motions.

1 Introduction

In recent years facial animation has become relevant in many different areas, for example in telecommunications, man-machine-interaction and games. But film production and advertising are still the main application areas of facial animation. In these fields two different types of figures have to be distinguished: Virtual Humans and 3D-Characters. The objective in the first case is the realistic modeling and animation of a human actor [12] whereas 3D-characters try to continue the tradition of cartoon animation [11]. In this paper we focus on the problem of facial animation for 3D-characters.

Although character animation does not require realistic behaviour it is nevertheless a difficult task to model convincing expressions for character faces. The reason can be found in special qualities of cartoon faces:

- The faces lack a uniform anatomy. Size, position and shape of typical facial structures vary heavily and often only a small subset of human facial structures is used. In contrast unusual parts of the face (e.g. ears) are often active parts of facial expressions.
- In many cases the face is very flexible and performs unusual motions compared to human faces (e.g. squash and stretch, anticipation, follow

through and exaggeration [11]). Some figures are even morphing between different facial appearances.

- The facial expressions can not be developed isolated from each other, they are closely related forming the personality of the character. Often even the shape of the face is subsequently changed if some expressions did not suit the imagined personality.

Characters do not look and behave like humans but nevertheless show convincing and believable facial expressions. They come to life through animation not through anatomic details and human-like motion. This needs an experienced animator who can control the strong interdependencies of character personality, expressions and facial shape.

Several different facial animation techniques have been developed since the first computer-generated facial model was introduced by Parke in the early 1970s [14]. As stated by Reeves [15] often simple facial models work better for character animation than complex physics-based models (e.g. [13]). Most of the anatomical complexity (bones, muscles, skin, etc.) supported by complex models is not needed in character animation. On the other hand basic cartoon animation rules are often not provided by these techniques because they are counterproductive in animating realistic faces. An example for an sophisticated animation technique supporting dynamic deformation methods for character animation can be found in [4].

Facial motions in simple facial models are often realized as composition of several isolated mesh deformations. These techniques provide total control over every aspect of motion but have a main drawback: The animator has to model the single deformations in full detail. This article deals with a new modeling paradigm trying to overcome this problem. The resulting technique allows a more creative and experimental type of modeling facial expressions. The corresponding prototype complements mesh deformation tools by a more creative sketching-like process.

2 Animation Technique

We use a simple kind of parametrization technique for character animation. In the first step several isolated motions called *Basic Motions* of the face (e.g. raising the left eyebrow) are modelled individually by mesh deformation. Basic Motions can be interpreted as generalized FACS Action Units [7] only describing the isolated deformation without making presuppositions about the anatomy and physiology of the face. Then the facial animation is constructed by defining the extent of each Basic Motion at any point of time and merging all resulting motions. Because our research is concentrated on interactive figures (e.g. co-presenters in TV-shows, Fig. 1) we developed numerical optimiziations of this method which allow a performance-based realtime animation [8] of the character.

The main advantages of this technique are the precise control over any detail of the motion, the integrative description of mimic and gesture and the realtime capabilities. The main drawback is that specifying many isolated facial deformations in full detail is a tedious and time-comsuming task and therefore hinders the creative process. The frog-character in Fig. 1 needs about 50 Basic Motions to perform facial expressions and lip-synchronization in the quality required for TV-shows. Keeping track of all the Basic Expressions interdependently forming the personality of the character therefore turned out to be another problem of this method.

Figure 1: Virtual Character "Frog" co-presenting the weather report in german TV. The character is animated in realtime using the described animation technique (ca. 80 Basic Motions) on a SGI Onyx. The motions are controlled by two actors using tracking devices for the body and data gloves for the face.

3 Modeling revisited

The basic process of our animation method is modeling Basic Motions by creating complex deformations of the facial mesh. Several advanced mesh deformation techniques like free form deformation [17], hierarchical b-splines [9], and constrained optimization modeling [22] have been developed. But in spite of these techniques modeling facial motions turned out to be a very difficult task mainly because of 4 reasons:

- The number of degrees of freedom in interacting with a complex 3D shape moving in 3D space is overwhelming. Especially the 2D projection of the shape and the commonly used devices mouse and keyboard turned out to be crude interfaces for 3D modeling.

- Apart from this problem most modeling tools require special training and knowledge about the used deformation techniques. Often the animator is not concentrated on the actual task of modeling facial expressions but on controlling the modeling technique.

- The most serious problem concerning the modeling process is basically a problem of imagination: The animator has an abstract but ambigous idea of the expression he would like to create. The difficulties in putting this idea into practice are very similar to trying to draw the face of a well-known person. It is easy to imagine the face, it is easy to recognize it, but it is quite difficult to transform the imagination to a real picture by using pencil and paper. It is not even possible to imagine details of the face exactely.

- The problem of developing isolated Basic Motions remains. As mentioned before the interplay of Basic Motions is essential for the personality of the character. A more sketching-like and experimental process is needed in which Basic Motions can be created and refined in parallel.

These problems can not be solved by a single mesh deformation technique but require a new type of modeling process. An interesting approach in this direction is the SKETCH-System [23], which allows to sketch 3D-Scenes in 2D by using a set of predefined gestures. This approach creates an environment for the needed conceptualizing modeling process but is much to imprecise for creating subtle facial motions.

4 Modeling by Evolution

Our aim was the development of a process which allows a more creative and experimental type of modeling facial expressions. It is interesting that artists achieve good results by using a trial and error process. First a rough sketch is generated which is compared to the imagination and then enhanced in several passes of correcting or restoring. The same trial and error process is used by animators when defining Basic Motions. But they have to deal with complex 3D-surfaces moving and deforming in 3D-space. Therefore the modeling interaction is complicated and hinders the creative process.

Artistic work is a highly interactive evolutionary process moving forward and backwards in the attempt to match an ambigous idea. The basic idea of our approach is to make explicit use of the evolutionary aspects of the process. The possibility of generating complex 2D- and 3D-forms by interactive artificial evolution has been demonstrated in several approaches (e.g. [5, 19, 21, 1]). The question is wether it is possible to use an evolutionary process to match a preexisting ambigous idea and wether the achievable quality is sufficient for modeling subtle facial motions.

The evolutionary modeling process performed by the user could be organized as follows:

1. The system generates a couple of random generated facial motions of a given figure and presents them to the user.

2. The user evaluates the generated expressions in 3D by rotating, zooming in details and sliding through time.

3. The user selects the best motions from his point of view and stores them into a population.

4. The user generates new random generated motion, or he decides to let the system mutate a member of the population or to mate two or more members of the population. Afterwards he continues with step 2.

The difference between the manual and the evolutionary modeling process is visualized in Fig. 2. Under the assumption that the process ensures the needed quality and very short response times it should provide the following qualities:

- No knowledge about the underlying mesh or the mesh deformation technique is needed to perform the modeling process. Only evaluation knowledge is required which can easily be provided because humans are specialists in judging facial motions.

- The process can be performed on the level of conceptualizing and ambigous ideas because a wide variety of variations can be generated fast and several ways to realize the idea can be followed simultaneously.

- Because of random factors in the operators there is even a possibility of surprising creative results not imagined by the user before.

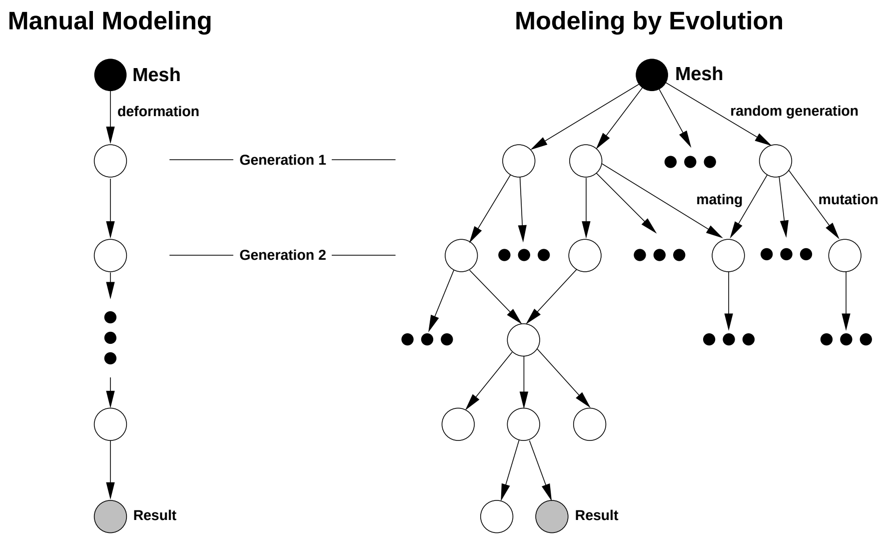


Figure 2: Comparison between the two modeling concepts.

- The system provides a simple interface. Only selecting expressions and choosing between the operators random, mutation and mating is required.

The main technical problem of this approach is the fact that the system has to generate "useful" motions. If the operators mainly produce motions which disrupt facial structures it will be almost impossible to find something good enough to proceed with.

5 Active Lines: Constructing the Search Space

The described paradigm can be interpreted as a type of search process: the user searches for a realization of his idea in the space of program-generated facial motions. The search space and the possibilities of searching are controlled by the definition of the genetic code and the evolutionary operators. In the first step it is therefore essential to identify a suitable search space of the problem. In the following we assume a facial representation of the form

$$G = (V, E), \;\; V \subset \mathbb{R}^3, \;\; E \subset V \times V \tag{1}$$

where G is a connected triangular mesh with a set of surface-points V and a set of edges E. V and E are the results of editing the raw scanning data. Any point of this mesh can move on a *path* defined by a spacecurve of the form

$$p : [0, 1] \to \mathbb{R}^3 \;\text{ with }\; p(0) = 0. \tag{2}$$

Therefore a path describes the relative movement of the corresponding point. Let P be the set of all possible paths. A *deformation* of the face is a function assigning a path to every point of the facial representation:

$$d : V \to P \tag{3}$$

Let D_G be the set of all deformations of a given facial representation G. This set also contains a very small subset BAM_G of all reasonable Basic Motions. D_G is therefore a search space of the problem. But D_G is much too great to perform an interactive search and the elements of BAM_G are wide-spread and not analytically extractable. The aim is to find a production system which constructs a subset of D_G as small as possible and containing BAM_G.

To find a suitable production system some knowledge about facial motions is needed. Because of the very convincing results of 2D-character animation we first analysed the mechanisms which are used there. It is obvious that it is possible to sketch convincing facial expressions with only a few special lines. Typical lines are eyebrows, eyelids, lips and cheeks [2]. Other hints for the importance of facial lines can be found in psychology. For example line drawings are often used by psychologists to describe facial expressions (e.g. [6]). An examination of manually constructed Basic Motions shows that animators are using the principle of moving main lines even when modeling exaggerated character-specific 3D-facial motions.

This concept can be formalized by defining the *Active Line* of a facial representation $G = (V, E)$ as the vector $al = (\{v_1, \ldots, v_n\}, \delta)$, where:

[i] $v_i \in V$, $i = 1, \ldots, n$

[ii] $(v_i, v_{i+1}) \in E$, $i = 1, \ldots, n-1$

[iii] $\delta : \{v_1, \ldots, v_n\} \to P$ is the *path distribution function* of the Active Line.

If $v_1 = v_n$ the Active Line is called *closed* otherwise it is called *open*. Some examples of Active Lines are shown in Fig. 3.

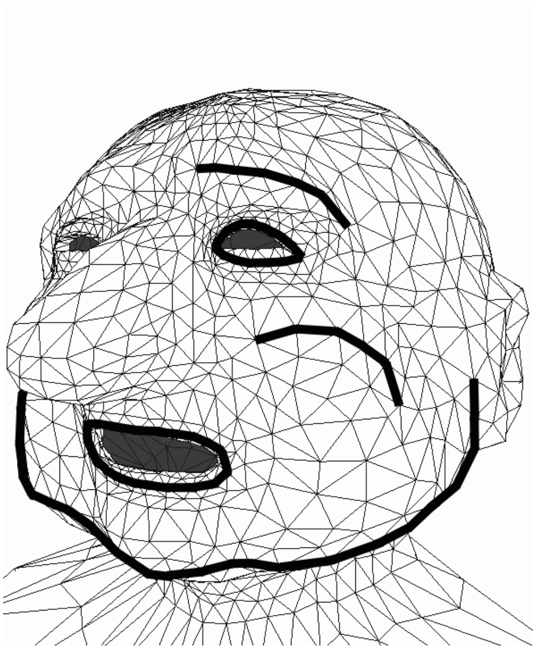


Figure 3: Examples for Active Lines.

6 Mimesys

To demonstrate the usability of the described evolutionary process and the concept of the Active Line we implemented a proof-of-concept prototype called *Mimesys* (Fig. 4) which provides easy user-interaction and a fast visualisation of the created motions.

To realize the prototype three problems had to be solved:

1. The concept of the Active Line has to be coded into a genomic structure.

2. The evolutionary operators of random generation, mutation and mating have to be defined on basis of the genome.

3. A material function has to be defined which is capable of transforming the genotype to the phenotype, i.e. is able to calculate the mesh deformation from the genomic structure.

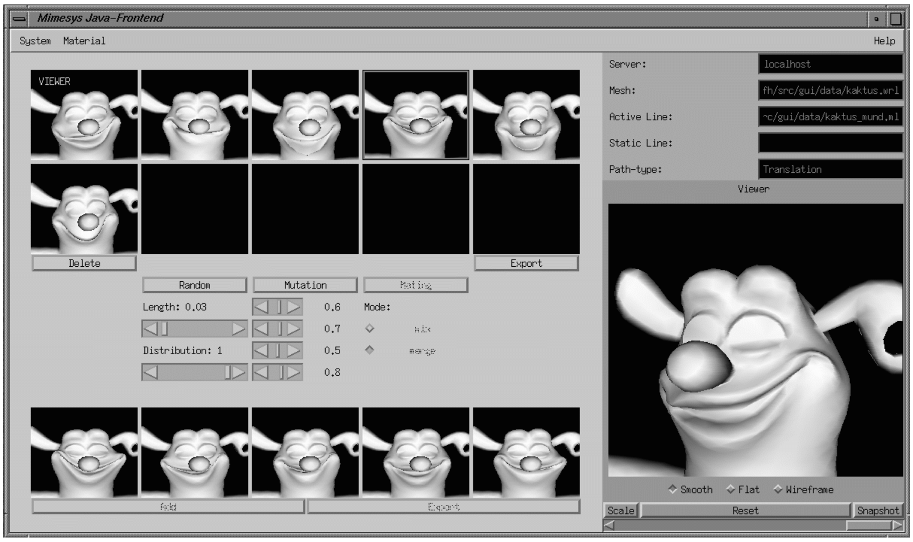


Figure 4: Mimesys-Interface. On the right there is a full 3D-representation of the character to evaluate the quality of a Basic Motion. On the top the current population is visualized, on the bottom descendants after the last operation. The modeling process is performed only by selecting individuals and using the evolutionary operators in the middle.

6.1 Genome

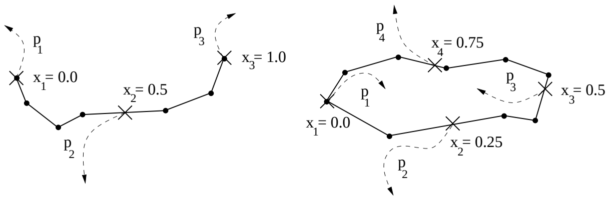
There are several possibilities of coding an Active Line into a genome. A naive approach would take the paths in all points of the Active Line. Because there is no relation between adjacent points this genome would spread out a search space mostly consisting of motions which destruct the facial structure. Therefore it is necessary to define constraints for the motion of adjacent points.

A more detailed examination of Active Lines shows the following properties:

- The line is unfractured and mostly waveless.
- The end parts and the middle part of the line determine the motion.

An ad hoc genome can therefore be constructed by only using the endpoints and the middle point of the line. These points are used afterwards to interpolate the motion of the other points of the line. In the case of open lines the genome therefore consists of the pathes of 3 main points, in the case of closed lines the pathes of 4 main points are used (Fig. 5).

The described structure of the genome has to be filled with suitable representations of the paths. In general any type of spacecurve (e.g. splines) can be used. It turned out that for Basic Motions path-types based on a combination of translations and rotations are sufficient because the motion complexity is achieved by merging several Basic Motions. These types are simple to code in vectors and besides can be efficiently used by the rendering pipeline. A genome for translation paths can for example simply be coded by using the 3 or 4 translation vectors of the main points of the Active Line.



x_1	x_2	x_3
p_1	p_2	p_3

x_1	x_2	x_3	x_4
p_1	p_2	p_3	p_4

Figure 5: Genome for open and closed Active Lines.

6.2 Operators

Three important evolutionary operators exist: random generation, mutation and mating.

Random generation: Frequently random generation is implemented by random choice of the genomic values. To get good starting points for the evolutionary process we defined an random operator which produces specific shapes with defined probabilies (e.g. 20% of generated motions are horizontally symmetric - the motion of one end point of the line is mirrored). This shape information is not part of the genome.

Mutation: The mutation operator interprets the user-specified quality as measure for the extent of variation. If the quality is low there has to be a great change in the motion paths. To gain better control over the process different qualities can be specified for the main paths, so that mutation can be constrained to parts of the line.

In the case of translation paths for example the quality affects the mean value of a normal distribution. The value calculated from this normal distribution is used as radius for a sphere around the endpoint of the motion. One point on this sphere is chosen to be the new endpoint of the motion.

Mating: The recombination operator is simply implemented as randomly weighted interpolation between the main paths of the participating Active Lines.

6.3 Material

The Basic Motion of the face has to be calculated from the motion of the Active Line. The physically-based simulation of elasticity leads to differential equations [20] and is therefore too expensive in this case. Because the modeling process is driven by evaluation of motions the system has to produce about 10 Basic Motions per second to maintain an adequate interactivity. As an alternative to simulating elasticity the deformation can be interpreted as an interpolation problem. In the two-dimensional case of image-warping this approach showed good results [16]. In this case the points of the Active Line can be interpreted as isolated data points with values representing the vector of coefficients of the space curve.

$$M = \{(v_i, c_i) \mid v_i \in V,\, c_i \in \mathbb{R}^d,\, i = 1, \ldots, n\} \tag{4}$$

Scattered Data Interpolation Methods can now be used to interpolate each component of the vector of coefficients separately:

$$f_j : \mathbb{R}^3 \to \mathbb{R},\, \forall (v_i, c_i) \in M : f_j(v_i) = c_{i,j}\,,\; j = 1, \ldots, d \tag{5}$$

This is a well-known problem and several techniques have been developed in this area. Because a fast reaction on users input is needed, the Shepard approach seems to be suitable (Shepard 1968). It calculates the values as weighted average of the values at the data points and therefore does not require the solution of equations.

$$f_j(x) = \sum_{i=1}^{n} w_i(x)\lambda_i(x)c_{i,j} \tag{6}$$

The weight function $w_i : \mathbb{R}^3 \to \mathbb{R}$ uses the distances $d_i(x)$ from the given data points v_i:

$$w_i(x) = \frac{\sigma_i(x)}{\sum\limits_{k=1}^{n} \sigma_k(x)} \quad \text{with} \quad \sigma_i(x) = \frac{1}{d_i(x)^\mu} \tag{7}$$

The function $\lambda_i : \mathbb{R}^3 \rightarrow \mathbb{R}$ is an extension of the Shepard interpolation called *mollifying function*. These functions are used to attenuate the influence of the data points. To get controllable material-like properties we developed an extension of the Franke-Little weights (Franke 1982):

$$\lambda_i(x) = \begin{cases} 1 & \text{if } d_i(x) \leq a_i \\ \left(1 - \frac{d_i(x) - a_i}{R - a_i}\right)^{\mu} & \text{if } a_i < d_i(x) \leq R \\ 0 & \text{if } d_i(x) > R \end{cases} \quad (8)$$

R directly controls elasticity. With low R the generated visual effect is plastic-like, with higher R it becomes more and more elastic. The elasticity effect of a given R depends directly on the size of the actual figure. To provide a size-independent elasticity parameter e to the user, R is calculated relative to the length d_{max} of the figures bounding box diagonal as $R = e \cdot d_{max}, e \in [0, 1]$. The a_i define the area around the data points in which no attenuation takes place. This avoids peak-like deformations if an isolated point moves very far away from its original position. Therefore the a_i are calculated relative to the distance between the endpoint of the curve and the point v_i.

The distances $d_i(x)$ turned out to be more problematic. Using the surface-distance for $d_i(x)$ results in drawbacks if convex forms are calculated, because nearby points in the volume can be far away on the surface. In contrast space-distance turned out to be problematic with concave forms, because the material effect jumps over the cavity. A very good approximation of the effect of elasticity can be achieved by using the surface-distance in the weight-function and the space-distance in the mollifying function.

Because of the interpolation of coefficent vectors the selection of the representation of the path-type is essential for the visual quality of the material simulation. It turned out that it is sufficient to approximate the paths in the points of the Active Line by simple polylines. The resulting polylines after deformation are used to construct the pathes by interpolation. For more complex path-types other representations might be useful (e.g. using the control points describing a Bézier curve).

7 Experimental Results

Figure 6 shows an example for modeling a Basic Motion with Mimesys. First a set of starting points for the process is generated with the help of the random operator. It often creates directly useable motions. Interesting motions are saved in the population and varied by mutation and mating. Better offsprings are adopted to the population and used as starting points for following generations. Often the process converges and leads to a suitable result after a

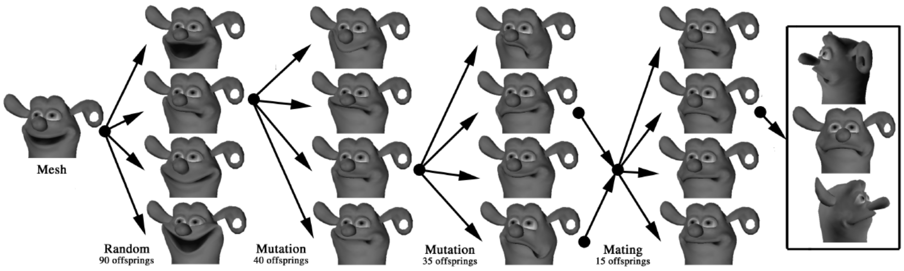


Figure 6: Modeling a Basic Motion in Mimesys. Only chosen examples of each generation are shown.

couple of generations. An extensive application is presented in figure 7. In this example we used an human-like actor to evaluate if our technique is able to produce all important human facial expressions. Fourteen Basic Motions of the eyes, eyebrows, the mouth and the chin have been generated and subsequently combined to facial expressions matching the Hjortsjo scheme[6]. Altogether approximately 500 individuals had to be created to find these Basic Motions by using the described operators.

The process itself shows the expected behaviour. It is possible to produce a wide variety of motions in a fast and easy way and to focus the process in a direction when the imagined expression substantiates. In several cases the system even produced surprising unexpected motions which can be stored and improved in parallel. The effectivity of the process depends strongly on the number of generations needed to get an acceptable result. Because the animator chooses the parent and the operator and controls the variation by constraints his experience is still an important factor in the process. But in contrast to conventional modeling the necessary experience is located at the level of evaluating, combining and varying whole facial motions.

The quality of the evolved Basic Motions is at the level of intermediate results of the modeling process which have to be enhanced manually to satisfy the quality demands of an experienced animator. These small enhancements can also be done by evolution, but the process is ineffective in this case, because a very high number of offsprings has to be generated. At this point of the process the animator knows his aim exactly but it becomes more and more difficult for him to evaluate if a small variation is an improvement. Additionally choosing parameters for constraining the variations becomes harder. These effects result in a loss of convergence and therefore in an increasing amount of time needed for small enhancements.

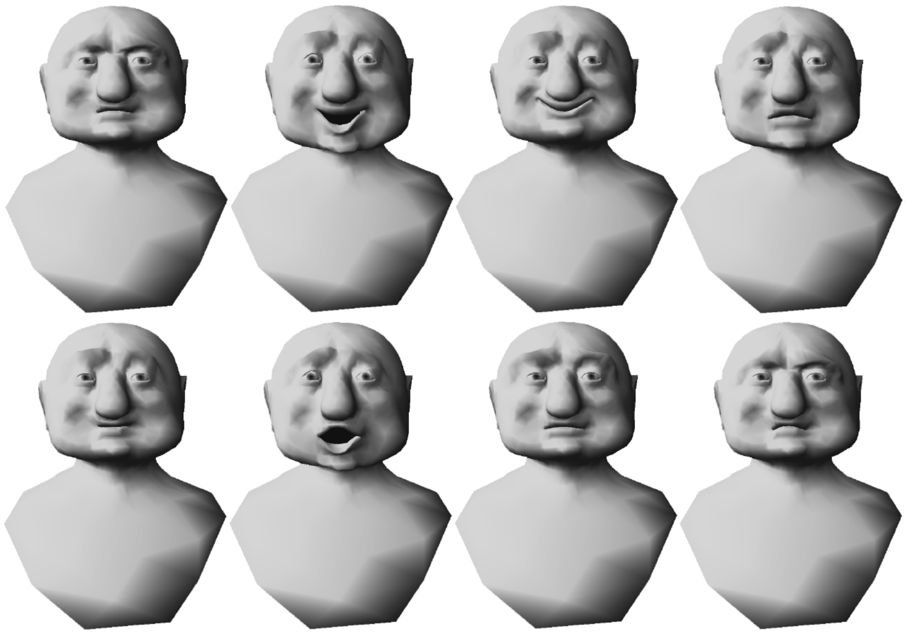


Figure 7: Evolved Hjortsjo scheme (angry, happy, self-satisfied, sad, sceptic, frightened, condescending, disgusted).

8 Conclusions

It is possible to enhance the process of modeling Basic Motions by using interactive evolution based on the concept of Active Lines. The approach leads to the required conzeptualizing modeling process. But the process is ineffective in doing fine adaptions or if the user exactly knows how defined points should move. The quality of the evolved motions is not fully satisfying at the moment because it is constrained by the ad hoc defined genomic structure. Several more sophisticated possibilities of coding an Active Line into the genome will be examined and evaluated. The described approach is capable of complementing mesh deformation tools by a creative sketching-like process. The integration of the process in a powerful modeling environment (e.g. as a plugin for Alias/Wavefront Maya) in which manual adaptions and conceptualizing can be performed in parallel will be the next step in development.

References

1. E. Baker and M. Seltzer, "Evolving Line Drawings", Graphics Interface '94, pp. 91-100, 1994.
2. P.Blair, "Cartoon Animation", Walter Foster Publishing, Laguna Hills, 1994.
3. O. Bunsen and G. Fleischmann, "Mesh Optimization for Animation Purposes", In: Deussen O, Lorenz P (eds) Simulation and Animation '97, Society for Computer Simulation International, 1997.
4. J. Chadwick, D. Haumann, R. Parent, "Layered Construction for Deformable Animated Characters", Proc. Siggraph'89 Conference, Boston, pp 243–252, 1989.
5. R. Dawkins, "The Blind Watchmaker", W.W. Norton, New York/London, 1987.
6. I. Eible-Eibesfeldt, "Similarities and Differences between Cultures in expressive Movements", In: Nonverbal Communication, Cambridge University Press, 1972.
7. P. Ekman and W. Friesen, "Facial Action Coding System", Consulting Psychologists Press, Palo Alto, 1978.
8. G. Fleischmann, R. Opalla R, A. Mähler, "Real Time Animation of 3D Characters", Proceedings of the 96 European SMPTE conference on Imaging Media, Cologne, 1996.
9. D. R. Forsey and R. H. Bartels, "Hierarchical B-Spline refinement", Computer Graphics, 22(4):205-296, 1988.
10. R. Franke, "Scattered data interpolation: Test of some methods", Mathematics of Computation, 38:181-200, 1982.
11. J. Lasseter, "Principles of Traditional Animation Applied to 3D Computer Animation", Computer Graphics, 21(4):35–44, 1987.
12. W. Lee, N. Magnenat-Thalmann, "From Real Faces To Virtual Faces: Problems and Solutions", Proc. 3IA'98, Limoges, 1998.
13. Y. Lee, D. Terzopoulos, K. Waters, "Realistic modeling for facial animation", Proc. Siggraph'95 Conference, Los Angeles, pp 55–62, 1995.
14. F. Parke, "Computer generated animation of faces", Masters thesis, University of Utah, Salt Lake City, UTEC-Csc-72-120, 1972.
15. W. T. Reeves, "Simple and complex facial animation: Case studies", SIGGRAPH 1990, Course Notes 26, 1990.
16. D. Ruprecht, H. Müller, "Free Form Deformation with Scattered Data Interpolation Methods", In: Geometric Modeling, Springer, Berlin Heidelberg New York, pp 267–281, 1993.

17. T. W. Sederberg and S. R. Parry, "Free-Form Deformation of Solid Geometric Models", Proc. SIGGRAPH 86, 20(4):151-160, 1986.
18. D. Shepard, "A two dimensional interpolation function for irregular spaced data", Proceedings 23rd. ACM National conference, pp 517–524, 1968.
19. K. Sims, "Artificial Evolution for Computer Graphics", Computer Graphics (Siggraph '91 proceedings), 25(4):319-328, 1991.
20. D. Terzopoulos, A. Platt, J. Barr, K. Fleischer, "Elastically deformable models", Computer Graphics, 21(4):205-214, 1987.
21. S. Todd and W. Latham, "Evolutionary Art and Computers", Academic Press, London, 1992.
22. W. Welch and A. Witkin, "Variational Surface Modeling", Proc. SIGGRAPH 92, 26(2):157-166, 1992.
23. R. C. Zelesnik, K. P. Herndon, J. F. Hughes, "SKETCH: An Interface for Sketching 3D Scenes", Proc. SIGGRAPH 96, pp. 163-170, 1996.