Animated Deformations with Radial Basis Functions

Jun-yong Noh
Computer Science Department
IMSC
University of Southern California
noh@graphics.usc.edu

Douglas Fidaleo
Computer Science Department
IMSC
University of Southern California
dfidaleo@graphics.usc.edu

Ulrich Neumann
Computer Science Department
IMSC
University of Southern California
uneumann@graphics.usc.edu

ABSTRACT

We present a novel approach to creating deformations of polygonal models using Radial Basis Functions (RBFs) to produce localized real-time deformations. Radial Basis Functions assume surface smoothness as a minimal constraint and animations produce smooth displacements of affected vertices in a model. Animations are produced by controlling an arbitrary sparse set of control points defined on or near the surface of the model. The ability to directly manipulate a facial surface with a small number of point motions facilitates an intuitive method for creating facial expressions for virtual environment applications such as an immersive teleconferencing system or entertainment. Smooth deformations of the human face or other models are possible and illustrated with examples of a variety of expressions and mouth shapes.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Geometric algorithms, languages, and systems; I.3.7 [**Computer Graphics**]: Animation;

General Terms

Algorithms, Performance, Design

Keywords

Geometry Deformation, Radial Basis Functions, Facial Animation, MPEG-4

1. INTRODUCTION

Tele-immersion is a metaphor that provides distant people with realistic experiences of face-to-face meeting. In a shared virtual space, eye contact and gaze gestures between conference participants are possible. Virtual objects can be manipulated and inspected with a haptic feedback device. 3D representations of people or objects are sensed, transmitted, and reconstructed to provide the viewing freedom necessary to give participants the sense that they occupy a shared space.

Model-based representations of real and virtual scenes often contain deformable surfaces such as faces, bodies, cloth, paper, etc. The transmission, storage, and dynamic deformation of these surfaces are efficiently performed by sensing (encoding) and synthesizing (decoding) the deformed surfaces. For example, a 3D model of a person can be transformed and deformed to match the behavior of the conference participant. In addition, the viewpoint at the decoding side can be changed to an arbitrary position allowing the inspection of the person from different angles.

High-level behavior encoding of the scene also leads to high compression ratios. For human faces, typically, the 3D model is transmitted once initially and animation parameters are sent at subsequent time frames. In an MPEG-4 implementation, when a model is not provided by the encoder, the decoder uses the model that already resides in the decoder side. Preparing person specific models and animating them require two sets of parameters. The first is the definition parameter set, primarily responsible for creating a person specific model, which includes the specification of shape, size and texture information. The second set contains the animation parameters that allow a variety of facial expressions and body postures.

Due to the complexity of the structure and behavior of human faces, modeling and animation of the face is often considered separately from whole body modeling and animation. Indeed, deformation mechanisms for facial animation are different from body animation controllers that manipulate joint angles of a human figure. In this paper, we will concentrate on deformations required for facial animation. We develop methods for deforming a given 3D model to create facial expressions and mouth shapes. The creation of person specific models is a separate sub-problem and example approaches can be found in [5, 6, 15, 20, 26]. We briefly survey common facial animation techniques in the following subsection.

1.1 Background

Pighin et al. [20] combine 2D morphing with 3D transformations of a geometric model to produce facial animation. The success of their animation depends on how realistically a collection of facial models can be created with various expressions. It requires the selection of a number of feature points and careful preparation of texture maps. This approach is a good modeling technique but animations are limited to interpolations between pre-made models. Facial animation using Finite Element Methods (FEM) [2, 7, 8, 11, 19] faithfully reconstructs facial geometry. FEM implicitly defines interpolation functions between nodes based on a description of the physical properties of the material, typically a stress-strain relationship. When external forces are applied, the displacements of the nodes are computed to minimize local stresses and strains imposed onto the nodes.

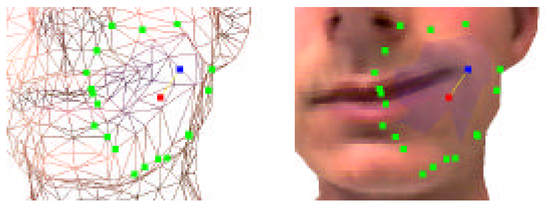
Layered skin models based on mass-spring systems mimic the anatomical structure and dynamics of the human face [15]. The mesh consists of three-layers corresponding to skin, fatty tissue, and muscles tied to bones. Elastic spring elements connect each mesh node and each layer. Muscle forces propagate through the mesh to create deformations. The computational cost for such spring systems can be very high. Kalra et al. [13] perform facial animations using free-form deformation (FFD). FFD deforms volumetric objects by manipulating control points arranged in three-dimensional cubic lattices. Conceptually, the face mesh is embedded in an imaginary, clear, and flexible control box containing a 3D grid of control points. As the control box is squashed, bent, or twisted into arbitrary shapes, the embedded mesh deforms accordingly. The basis for the control points is a tri-variate tensor product Bernstein polynomial.

Vector-based muscle models [27, 28] are adapted widely for their compact representation. A delineated deformation field models the action of muscles upon skin. The muscle definition includes the vector field direction, an origin, and an insertion point. The cone shaped field extent is defined by cosine functions and fall off factors. Facial animation is achieved by changing the contraction parameters of the embedded muscles under the face mesh. This approach assumes that muscles are placed under the face mesh in correct locations. Placing muscles in 3D space, however, is not intuitive or consistent from model to model. Deformation methods also include spring muscles [21], spline models [4], and parametric models [18]. Performance driven facial animation (PDFA) [9, 10, 14] has used video streams as an input, generating animations that mimic the tracked subject using the methods described above.

We use Radial Basis Functions (RBF) to create various expressions and their animations. In our previous work [9], we showed deformations using RBF in performance driven facial animation (PDFA). Briefly, a small number of tracked feature points determine new positions of vertices on the mesh for every frame. The use of RBF's guarantees smooth geometric deformation. Our method is similar to Guenter et al.'s [10] system in that 3D feature points are used as a driving force, however RBF's allow a more direct and compact representation of animation parameters than an ad hoc smoothing method. Our former work shares similar limitation with others, however, in that the mesh needs to be equipped with predefined feature points to ensure correct deformations. We eliminate the necessity for pre-defined feature points in our current approach. In addition, instead of having a single global RBF deformation engine, we exploit a number of small RBF deformation elements to localize the deformations.

Our approach provides unique advantages over existing methods for creating facial animation. First, most existing approaches require animation mechanisms (e.g., muscles, or FFD) explicitly embedded in the facial mesh. For example, muscles must be placed under the mesh [27, 28], or spring and stiffness constants must be determined in advance [2] and tuned each time a new face mesh is created. In contrast, our RBF method works on any mesh without modification. Second, the creation of facial animations can be easily automated with our method. When video data of an actor are available, feature points tracked on the subject are directly applied to deform the face model to mimic his/her expressions. The number and locations of tracked feature points does not need to be fixed. As more points are tracked, deformation control and fidelity are increased. This flexibility is due to the direct deformation capability of the RBF system. Third, the behavior of our deformation method is very predictable. External forces are applied directly to the facial surface by each tracked point, and nearby nodes are displaced smoothly. Since the surface is directly manipulated, the process is more intuitive than methods that indirectly affect the facial surface [13].

The geometry deformation mechanism using RBF is discussed in section 2. Automated animation generation is described in section 3. Results are shown in section 4 followed by discussions in section 5 and conclusions in section 6.



A red point represents initial control point position while a blue point represents new position. Green points are anchor points and the area inside the green points is the region of influence. The shape of the region of influence is affected by the mesh regularity.

Figure 1 Geometry Deformation Element defined on the facial surface

2. GEOMETRY DEFORMATION

Face mesh geometry is locally deformed by a geometry deformation element (GDE). A GDE is the smallest deformation unit defined on the surface of the face. A GDE consists of a control point, the region of influence around the control point, anchor points that lie in the boundary of the influence region, and an underlying RBF system (figure 1, 4). The movable control point and the stationary anchor points determine the displacement of the vertices in the influence region. Specifying any point on the face creates a GDE. A control point may be derived from a 2D image by projecting it to the 3D mesh surface. The region of influence is bound by a distance metric that determines the stationary anchor points. The number of mesh vertices in the influence region can be large or small. An influence region of one vertex reduces deformation to vertex manipulation.

2.1 Algorithm Summary

Creating a deformation element:

1. Specify a control point on the mesh and an influence extent to control the deformation around the point. The selected point does not have to coincide with any of the vertices in the mesh.
2. For points selected in 2D images, convert them to 3D points on the model surface by ray casting.
3. Find the nearest vertex on the mesh to the selected 3D point. This vertex becomes the root for the search tree of mesh edges.
4. Search down the tree of mesh edges with a Breadth First Search, determining all vertices within a specified distance metric.
5. Leaf nodes of the search tree become the anchor points and, together with the specified control point, initialize the RBF system associated with the GDE (figure 4).

Actuating a deformation element:

1. Specify a new position of the control point either by mouse dragging or tracking a facial feature in a video sequence.
2. Convert the new 2D control point to 3D by ray casting.
3. The RBF system computes the new locations of all vertices in the influence region based on the new control point position and the stationary anchor points.

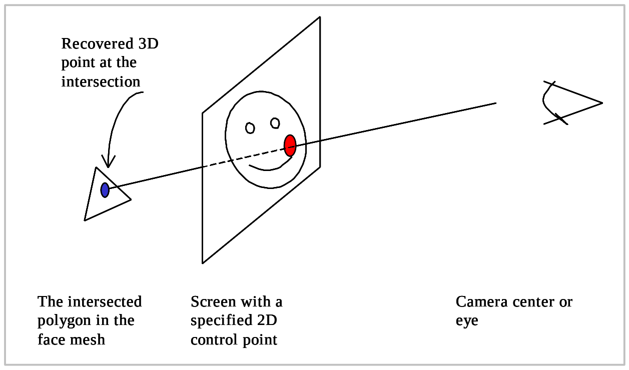
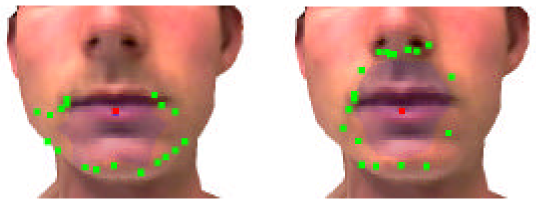

Recovered 3D point at the intersection
The intersected polygon in the face mesh
Screen with a specified 2D control point
Camera center or eye

Figure 2 Computing 3D coordinates

2.2 Ray Casting

Note that the first step in creating and actuating deformation elements can be accomplished by manual input or automation. In either case a 2D point is identified (by mouse click or feature detection and tracking). Unless the selected point coincides exactly with a mesh vertex, its 3D location is unknown. However, we have a 3D face model so we can approximate the 3D point position by ray casting. The intersection point of the model with the ray emitted from the camera center through the specified 2D image point gives the 3D location on the face mesh (figure 2). Direct specification of 3D control point positions are also possible to handle cases where the control points move out of the mesh or along silhouettes.



Edge based method selects lower mouth region while distance based method selects whole mouth region starting from the same location.

Figure 3 Comparison between edge based search method and distance based search method

2.3 Search Methods and Distance Metrics

Once a 3D control point is specified, the region of influence and anchor points can be determined. We consider the edges of the face mesh to be an arbitrary tree with a root located at the nearest vertex to the specified control point. The GDE influence region and anchor points are determined by searching down the tree using a Breadth First Search [16]. During traversal, vertices are tested against a distance metric to see if they fall in or out of the influence region. We experimented with two distance metrics. One is based on edge depths and the other is based on Euclidean distance. The edge depth metric marks all vertices within some integer number N of mesh edges as in the influence region. The distance metric computes the Euclidian distance between traversed vertices and the control point, and when that distance is below threshold, the vertex is marked as within the influence region. The threshold is a real number scaled to the mesh coordinate units.

The two metrics serve different purposes. For example, when opening the mouth, the influence regions should be separate in the upper lip area and the lower lip area. In this case, an edge based metric finds the lower part of the mouth mesh for any control point on the lower lip without affecting the upper mouth region (Figure 3). In cases where mesh density is very irregular, for example in the eye regions, an edge metric produces very irregular shaped influence regions. The distance metric produces regular shaped influence regions regardless of mesh density variations. In many cases, we find that both metrics produce similar deformations. For large influence regions or very dense meshes, the number of boundary points can become large. In these cases we limit the number of perimeter points to 20 sampled evenly along the boundary vertex set.

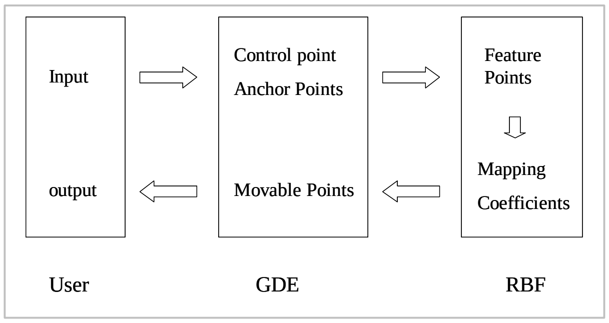


Figure 4 Relationships between GDE and RBF

2.4 Radial Basis Functions

Inspired by the recent success of RBF approaches in 3D model generation [5, 19, 26] and its demonstrated capability in image warping [24] we exploit RBF volume morphing to directly drive 3D geometry deformation of face models.

2.4.1 Interpolation/Approximation with RBF

Classical approximation theory solves the problem of approximating or interpolating a continuous multivariate function $f(\vec{x})$ by an approximation function $F(\vec{x},\vec{c})$ with an appropriate choice of parameter set $\vec{c}$ where $\vec{x}$ and $\vec{c}$ are real vectors ($\vec{x}$ = x$_1$, x$_2$, ….., x$_n$ and $\vec{c}$ = c$_1$, c$_2$, ……,c$_n$). Finding a parameter set $\vec{c}$ is often referred to as learning or training in the neural network sense. In the training stage, a goal is to figure out $\vec{c}$ given an approximation function F and a set of training examples, which will provide the best approximation of f. Radial Basis functions are often chosen as an approximation function F for its power to deal with irregular sets of data in multi-dimensional space in approximating high dimensional smooth surfaces [22]. Examples of the RBFs are Gaussian functions $h(r)=e^{-(\frac{r}{c})^2}$, multi-quadrics $h(r)=\sqrt{r^2+c^2}$, and thin plate splines $h(r^2)=r^2\log r$ with a linear term added. Radial basis functions are named because of their radially symmetric distances parameters. They can be implemented using simple neural network with one hidden layer.

2.4.2 Face Deformations with RBF

In this section, we refer to a specified GDE control point and its anchor points as "feature" points since their distinctions are meaningless to the RBF system. As depicted in figure 4, each GDE has one RBF system for displacements computations. A RBF system deforms the facial mesh based upon the motions of all feature points. The mappings between initial positions and new positions of the feature points are described in terms of the vector coefficients. We compute this mapping at each frame. (This is known as training in the neural network terminology – but this is implicit and no explicit training process is required). The rest of the nodes in the influence region are transformed based upon the coefficients computed. The radial basis function approximation equation is

$$F(\vec{x})=\sum_{i=1}^{N}c_i h(\|\vec{x}-\vec{x}_i\|) \qquad (1)$$

where $h(r)=\sqrt{r^2+s^2}$ for Hardy multi-quadrics. s is called a stiffness constant that regulates the local or global effects of the feature points and r is the Euclidian distance between the feature point and the input point. When computing mapping coefficients, input points are the feature points themselves and when evaluating the new positions, input points are the points in the influence region. Plugging the Hardy basis function into equation (1) results in

$$\vec{x}^{target}{}_j=F(\vec{x}^{source}{}_j)=\sum_{i=1}^{N}c_i\sqrt{\|\vec{x}^{source}{}_j-\vec{x}_i\|^2+s_i^2} \qquad (2)$$

where $1\le j\le$ Number of feature points, in our case, N. The dimension of $\vec{x}$ is three (i.e. x,y,z coordinates of each feature point). $\vec{x}^{source}$ denotes the initial positions of the feature points and $\vec{x}^{target}$ denotes new positions of the feature points. Stiffness coefficient s_i is determined as suggested by Eck [3] for softer deformation where feature points are widely scattered and stronger where closely located.

$$s_i=\min_{j\neq i}\|\vec{x}^{source}{}_j-\vec{x}_i\| \qquad (3)$$

Substitution of N feature points into the equation (2) results in a linear system of N equation whose solution is of the form

$$\vec{c}=H^{-1}\vec{x}^{target} \qquad (4)$$

The solution given by equation (4) assumes that there is no spurious data. In general, however, data is not noise-free. In the presence of noise, using the matrix H constructed with the assumption of perfect data may not produce a useful result. In this case, Thikhonov and Arsenin [25] provide a very simple solution. We simply replace the matrix H by (H + λI) for equation (4). Then equation (4) becomes

$$\vec{c}=(H+\lambda I)^{-1}y \qquad (5)$$

where λ is a "small" parameter. The magnitude of λ is proportional to the noise magnitude. Notice that equation (5) becomes identical to (4) by setting λ to 0 where noise free perfect data are assumed. We simply set λ to be 0.01, determined experimentally.

The linear system of equation (5) is easily solved by LU decomposition [23] to obtain the coefficient set $\vec{c}$. The LU decomposition of the matrix H happens only once at the initialization of the RBF system for each GDE. Only a back-substitution is computed for deformation frames with new positions of the feature points $\vec{x}^{target}$. Thus the deformation computation is fast. Once the system is solved, the deformed positions for vertices in the mesh influence region are obtained from the computed coefficients.

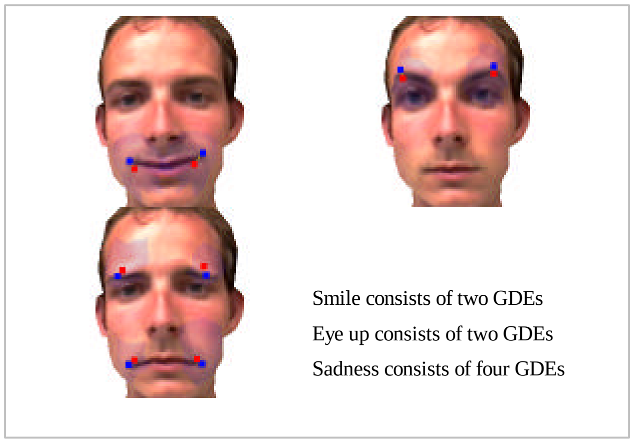

Smile consists of two GDEs
Eye up consists of two GDEs
Sadness consists of four GDEs

Figure 5 Expressions as a collection of Geometry Deformation Elements (GDEs)

2.5 Generating Expressions

A geometry deformation element is the smallest unit for surface deformation. One or more deformation elements constitute an expression. For example, two deformation elements make one smile expression (figure 5). In this way, a variety of expressions are possible by using various combinations of deformation elements. We can control a set of deformation elements with a single parameter *d* where *d*=0 corresponds to a neutral expression, and *d*=1 corresponds to the maximum displacements of all the control points of the member deformation elements. The expression can then be animated simply by changing the control parameter over the range of [0, 1]. Mouth shapes used for speech synthesis are created and controlled similarly. Mixtures of multiple control parameters are also possible.

3. AUTOMATION

In the previous section, we described how to define deformation elements on the surface of the face and manually create and control various expressions. Once a gallery of expressions is constructed, animation across existing expressions can be achieved by key frame interpolations of expression parameter values. It is desirable to automate or at least semi-automate the construction of the initial expression database (figure6).

We take the approach of performance driven facial animation (PDFA) to construct expressions. In PDFA, a human actor is tracked with a camera while generating facial expressions and mouth shapes. This recorded or on-line video stream is analyzed to extract the motion of salient facial features. These motions then drive the deformation of the face model to produce similar expression animations. PDFA was first introduced in 1990 [29] and used in various contexts. PDFA has been used to drive 2D animation [1, 7, 14] and 3D animation [9, 10]. A major difficulty of using PDFA to automatically generate 3D facial animations lies in the ambiguous relationship between the tracked features and the animation mechanism. Given a sparse set of tracked displaced points on the face, estimating the animation parameters that invoke the displacements is an inverse problem not easily solved with many existing approaches [13, 28].

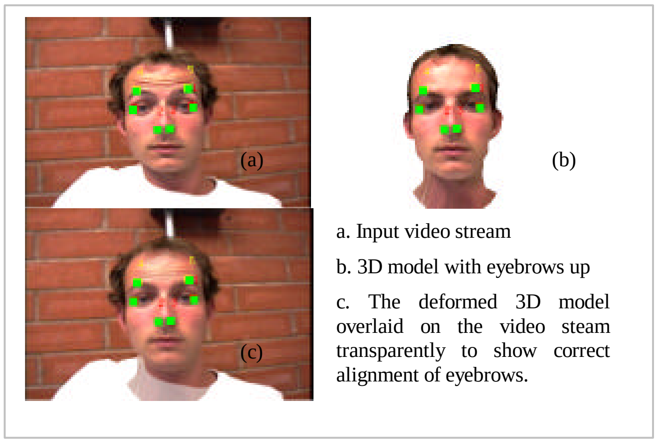

(a)
(b)
(c)
a. Input video stream
b. 3D model with eyebrows up
c. The deformed 3D model overlaid on the video steam transparently to show correct alignment of eyebrows.

Figure 6 Deformations driven by feature points in the video stream

In contrast, our GDEs can be controlled directly by the feature motion vectors measured in the images. This direct relationship between the tracked feature points and the GDE control points is a major advantage in simplifying a PDFA system. The steps to automate the generation of facial expressions can be summarized as follows:

1. Video streams are captured containing the subject making various expressions and mouth shapes.
2. Salient control points are identified (manually or automatically) on the subject face(s) and tracked over the expressive sequences.
3. The 2D tracked points from the video streams are treated as GDE control points and converted into 3D points using ray casting.
4. The 3D control point motions are input to the GDEs where deformations are produced as depicted in section 2.1.

For the tracking of the feature points and pose estimation of the head in the video streams, we use existing work [30] adapted to suit our purposes. Feature tracking and pose estimation methods are likely to produce erroneous results due to analysis errors and non-ideal imaging conditions. Our animation application provides an interactive editing interface that allows the animator to manually correct or override the tracking and pose results to achieve the desired animations.

4. RESULTS

We create a variety of expressions and mouth shapes by choosing different tracking/control points and different influence regions and directly manipulating these points on the face. Figure 9 shows sample expressions. Small modification of the lip region conveys a feeling of dissatisfaction (a) or decisiveness (b). Moving one eyebrow upward shows cleverness (c). Pulling lips and eyebrows corners down create sadness (d). For a sly look, only a small number of vertices are displaced around the lip corner and eyebrow using small influence zone (e, f). The same impression may be created by totally different expressions. For instance, facial expressions for anger may vary from person to person, or even within the same person (g, h).

Sample animation sequences from a neutral state to full expression (figure 7, 8) and between two expressions are shown (figure 10). With the 3D model, inspection from an arbitrary viewpoint is possible (figure 7, 8). Figure 11 shows models generated with the automated process. Without any prior preparation, the same GDE techniques are applied to “Skip’s” 3D model. Three red points on the eye sockets and the tip of the nose are used for pose estimation while the motions of yellow points are used for deformations of the face. Additional MPEG movies are also available on http://scf.usc.edu/~noh. We use a facial mesh with 1954 polygons. Real time (30Hz) animations are achieved with a moderately configured 500MHz PC.

5. DISCUSSION

Radial basis functions guarantee a smooth surface deformation from a sparse set of control points. However, if a control point is moved too far from its original position, say outside the influence region, large discontinuities occur around the anchor points. Because anchors points are stationary at the boundary of the influence region, no influence of the control point will propagate through the anchor points. Such large control point motions do not occur in practice and most deformation engines would produce unnatural effects under similar conditions. By limiting the deformation regions we eliminate the need to prepare or alter the mesh as required by other methods, including our previous approach [9]. Currently, we assume that a specified control point can only be moved within the region of influence. As a possible adaptation to allow the control point to move beyond the influence region, one can consider a dynamic tree search to regenerate a larger region of influence with new anchor points.

Our method estimates the required 3D control points using ray casting. However, erroneous results can occur. We allow manual editing of the 3D control points as needed to compensate for such errors. An ultimate solution for this problem may be to impose constraints on the facial surface such as bone structures. However, this introduces limitations to the use of simple meshes. Any points on the facial surface can be used as control points to deform the model. We can restrict this flexibility to use a subset or all of the feature points MPEG-4 defines. Our deformation mechanism can be completely MPEG-4 compliant and extendible if additional tracking data is available. In the mouth regions, for example, additional control data may come from a speech stream analysis. Animations using techniques such as arithmetic coding or DCT coding as described in [17] can achieve data rate as low as 300 – 2000 bits/s.

6. CONCLUSION

Creating lifelike facial animation is a crucial factor in developing an immersive virtual environment. We presented a method for animating deformations of 3D face models. In scenes containing faces, the analysis of facial images can efficiently encode the data needed to control the deformations. Our GDE method is a novel approach to deforming face models by directly manipulating feature points. The RBF based computation produces localized smooth deformations on the face. This approach is applicable to most meshes without special initialization. The process requires minimal and intuitive human intervention and is easily automated by the use of a feature tracking system with video streams.

7. ACKNOWLEDGEMENT

This work was supported by DARPA and the Annenberg Center at USC. Additional Support and research facilities are provided by the NSF through its ERC funding of the Integrated Media Systems Center. We recognize the contributions to this work from all our colleagues in the CGIT laboratory. Special thanks go to Tae-Yong Kim, Bolan Jiang, Suya You, and Reyes Enciso.

8. REFERENCES

[1] A. Abayejani, T. Starner, B. Horowitz, A. Pentland, Visually Controlled Graphics, IEEE PAMI 15(6), June 1993

[2] S. Basu, N. Oliver, A. Pentland, 3D Modeling and Tracking of Human Lip Motions, ICCV, 1998, 337-343

[3] M. Eck, Interpolation Methods for Reconstruction of 3D Surfaces from Sequences of Planar Slices, CAD und Computergraphik, Vol. 13, No. 5, Feb. 1991, 109 – 120

[4] P. Eisert and B. Girod, Analyzing Facial Expressions for Virtual Conferencing, IEEE, Computer Graphics and Applications, 1998, vol. 18, no. 5, 70-78

[5] R. Enciso, J. Li, D. Fidaleo, T-Y. Kim, J-Y.Noh, U. Neumann, Synthesis of 3D Faces, International Workshop on Digital and Computational Video, 2000

[6] M. Escher, I. Pandzic, N. Thalmann, Facial Deformations for MPEG-4, IEEE Computer Animation, 1998, 56 - 62

[7] I. A. Essa, S. Basu, T. Darrell, A. Pentland, Modeling, Tracking and Interactive Animation of Faces and Heads using Input from Video, Proceedings of Computer Animation June 1996 Conference, Geneva, Switzerland, IEEE Computer Society Press

[8] I. A. Essa, T. Darrell, A. Pentland, Tracking Facial Motion, Proceedings of the IEEE Workshop on Non-rigid and Articulate Motion, Austin, Texas, November, 1994

[9] D. Fidaleo, J-Y. Noh, T. Kim, R. Enciso, U.Neumann, Classification and Volume Morphing for Performance-Driven Facial Animation, International Workshop on Digital and Computational Video, 2000

[10] B. Guenter, C. Grimm, D. Wood, H. Malvar, F. Pighin, Making Faces, Siggraph proceedings, 1998, 55 - 66

[11] B. Guenter, A system for simulating human facial expression. In State of the Art in Computer Animation, 1992, 191–202

[12] I. R. Jackson, Convergence properties of radial basis functions. Constructive approximation, 1988, 4:243-264

[13] P. Kalra, A. Mangili, N. M. Thalmann, D. Thalmann, Simulation of Facial Muscle Actions Based on Rational Free From Deformations, Eurographics 1992, vol. 11(3) 59–69

[14] I. Koufakis, B. F. Buxton, Very low bit rate face video compression using linear combination of 2D face views and principal components analysis, Image and Vision computing, 1999, 17, 1031 – 1051

[15] Y. C. Lee, D. Terzopoulos, K. Waters. Realistic face modeling for animation. Siggraph proceedings, 1995, 55-62

[16] Edward F. Moore. The shortest path through a maze. In Proceedings of the International Symposium on the Theory of Switching, Harvard University Press, 1959, 285 – 292

[17] J. Ostermann, Animation of Synthetic Faces in MPEG-4, IEEE Computer Animation, 1998, 49 - 55

[18] F. I. Parke, Parameterized models for facial animation. IEEE Computer Graphics and Applications, 1982, vol. 2(9) 61 – 68

[19] S. Pieper, J. Rosen, and D. Zeltzer, Interactive Graphics for plastic surgery: A task level analysis and implementation. Computer Graphics, Special Issue: ACM Siggraph, 1992 Symposium on Interactive 3D Graphics, 127–134

[20] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D. H. Salesin, Synthesizing Realistic Facial Expressions from Photographs, Siggraph proceedings, 1998, 75-84

[21] S. Platt, N. Badler, Animating facial expression. Computer Graphics, 1981, vol. 15(3), 245-252

[22] T. Poggio, F. Giros, A theory of networks for approximation and learning. Technical Report A.I. Memo No. 1140, Artificial Intelligence Lab, MIT, Cambridge, MA, July 1989

[23] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, Cambridge University Press, ISBN 0-521-43108-5

[24] D. Ruprecht, H. Muller, Image Warping with Scattered Data Interpolation, IEEE Computer Graphics and Applications, 1995, 37 – 43

[25] A.N. Tikhonov and V.Y. Arsenin, Solution of Ill-Posed Problems and the regularization method. Soviet Math. Dokl., 1963, 4:1035-1038

[26] F. Ulgen, A step Toward universal facial animation via volume morphing, 6th IEEE International Workshop on Robot and Human communication, 1997, 358-363

[27] K. Waters, J. Frisbie, A Coordinated Muscle Model for Speech Animation, Graphics Interface, 1995, 163 – 170

[28] K. Waters. A muscle model for animating three-dimensional facial expression. In Maureen C. Stone, editor, Computer Graphics (Siggraph proceedings, 1987) vol. 21, 17-24

[29] L. Williams, Performance Driven Facial Animation, Siggraph proceedings, 1990, 235 - 242

[30] P. Zhenyun, Suya You and Guangyou Xu, "A Fast Method for Detection Facial Features Under Varied Poses", China Journal of Image and Graphics, 2 (4), 1997.

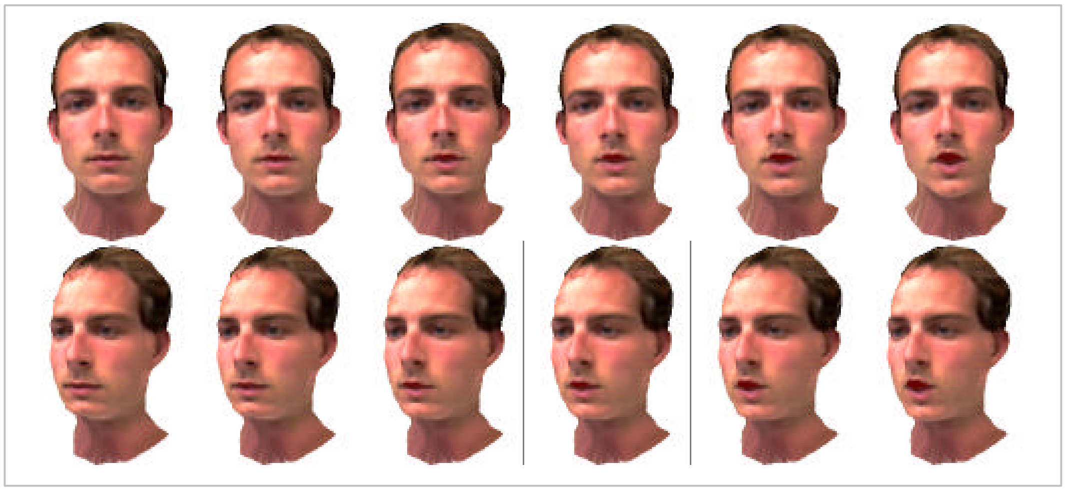


Figure 7 Sequence of making 'A' sound

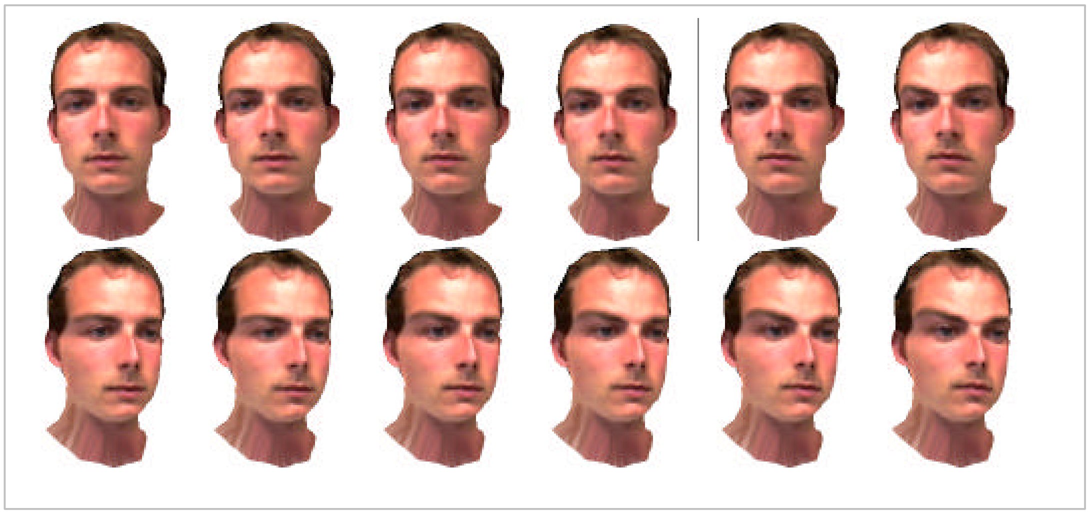


Figure 8 Sequence of making angry face

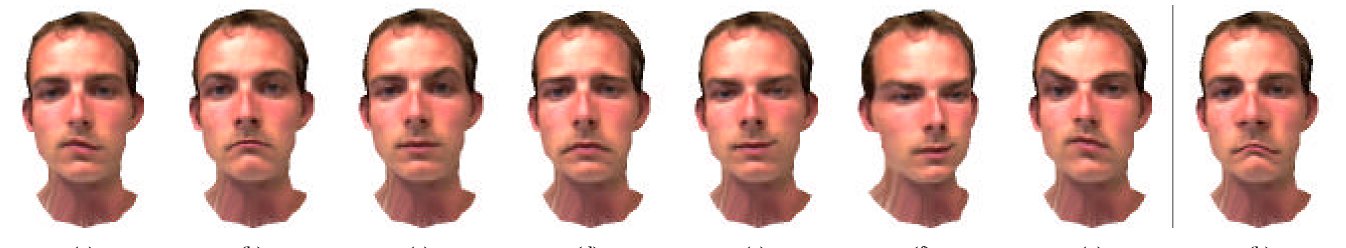


Figure 9 Sample expressions created by one or more GDEs

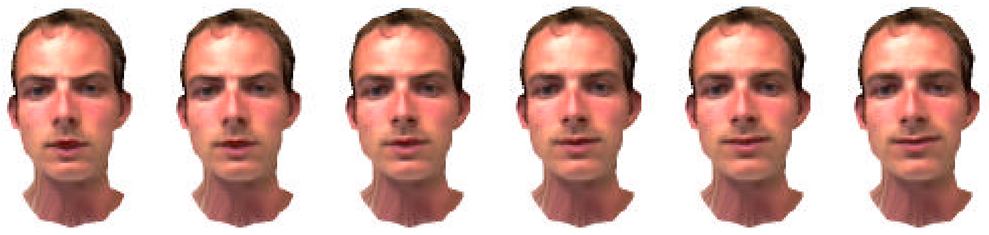


Figure 10 Transitions between two expressions

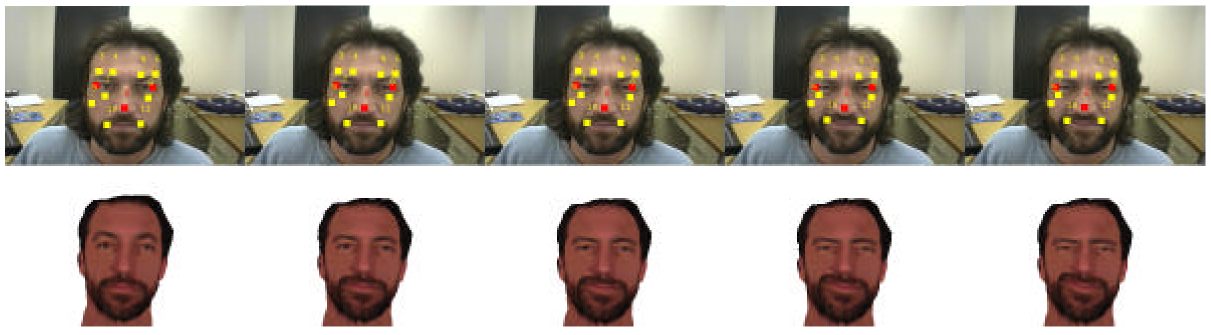


Figure 11 Automated animations with input video stream