Animating Facial Images with Drawings

Gamze D. Tunalı, Bülent Özgüç
Department of Computer Engineering and Information Science
Bilkent University
tunali@cs.bilkent.edu.tr, ozguc@bilkent.edu.tr

Abstract

The work presented here describes the power of 2D animation with texture mapping controlled by line drawings. Animation is specifically intended for facial animation and not restricted by human face.

We have initially a sequence of facial images which are taken from a video sequence of the same face and an image of another face to be animated. The aim is to animate the face image with the same expressions as those of the given sequence.

This work encourages the reuse of animated motion by gathering facial motion sequences into a database. Furthermore, by using motion sequences of a human face, non-human characters can be animated realistically or complex characters can be animated by the help of motion sequences of simpler characters.

Key words: facial animation, facial expression, snakes, active contour models, multigrid relaxation, multilevel B-Spline interpolation.

1 INTRODUCTION

In this study, 2D facial animation is performed which is controlled by line drawings. By aligning curves, lines and points with features in an image, intuitive controls for image warping are constructed. Deformation of an image can be accomplished by applying the warp defined by the original drawing and any other drawing of the same features. Animation is done simply by animating drawings and applying the image warp at each frame. The ability of mapping animation from one image and a set of features to another gives a power to animated sequences and enables the reuse of animated motion for any single face image.

Initially, we have consecutive frames of a video sequence of the same face and an image of another face that is to be animated. The goal of our work is to animate a given face image with the same expressions as those of the given video frames. As the first step, we outline some features which are of interest to us by hand on the first frame of the sequence, and then carry them to their real places by the help of snakes. There are some other methods to track the facial features [ISP94, Saj92]. Features are mouth, eyes, nose, eyebrows, etc. For each feature on the first frame, we specify a corresponding feature on the image to be animated. Features can be considered as a collection of sequenced points on the image plane. For each feature correspondence, we have a set of point pairs. Using the cross-synthesis method explained in [EPG91], we get an interpolation function which gives a pixel position on the image to be animated for each pixel on the first frame. Litwinowicz used multilevel surface reconstruction method to find the interpolation function, but it is slow. Furthermore, there will be no discontinuity in the function, therefore using this method is not always advantegous. Instead, multilevel B-spline interpolation is used, which is faster and simpler than multilevel surface reconstruction method in this respect.

After the features are specified on the first frame, places of these features on the other frames in the sequence should be tracked. This process is performed automatically using two motion estimation techniques which are block matching and optical flow.

Since we have acquired the interpolation function which gives pixel position correspondences of two initial images and tracked each features on the video frames, similar animated drawing sequence can be produced for the image to be animated. To produce the full set of animated images we need a warp function for each animated drawing frame. For a number of known (x_k, y_k) positions in the image plane which are control points of the features, we have a set of known displacements $(\Delta x_k, \Delta y_k)$ as defined by the original and produced feature drawings. To warp the image according to the drawings, we need two interpolation functions $F_1(x_k, y_k) = \Delta x_k$ and $F_2(x_k, y_k) = \Delta y_k$. First function is a smooth interpolating function for the x-displacements for an entire image, and the second is similarly for the y-displacements. In this case, some discontinuities can be pointed out by the user for some features, therefore multilevel surface reconstruction method is more convenient. For each produced animated drawing frame of the sequence, a warp function is computed by using the original drawings and then applying the warp functions F_1 and F_2, original image is animated.

2 BACKGROUND

Animating drawings by their features is first studied by Litwinowicz et al. [Lit91] by using a mesh of bilinear Coons patches [For72]. Coons patches are inexpensive to evaluate, but they require manual division of the image into a mesh and all of the patch boundaries should be animated to control the motion. Therefore, it is time consuming and requires a substantial manual effort. Specifying and animating only the features of interest is more general and easier.

The most recent related work is done by Litwinowicz et al. [EPG91]. In this case, an actor's facial expressions are captured from video by the help of fluorescent spots on the actor's face. By these spots *motion control points* are tracked. The acquired motion control points qre spatially mapped to the a synthetic face, giving new control points which are used to animate synthetic face.

Litwinowicz et al. enhanced the idea from motion control points to the drawings in [LW94] and proposed thin plate splines for visual surface reconstruction. In this work, feature specification and automatic tracking phase are leaved untouched.

3 MAJOR STEPS OF OUR WORK

3.1 Feature Specification

After deciding the features which are important for our animation purposes, they should be localized on the image accurately. Initially they are outlined by hand and then *snakes* are applied to carry them to their real places. A snake is an energy minimizing-spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Snakes are active contour models. They lock onto nearby edges, localizing them accurately.

Kass, Witkin and Terzopoulos [KWT88] has developed the snakes (Active Contour Models). The problems of the method of Kass, Witkin and Terzopoulos are numerical instability and a tendency for points to bunch up on strong portions of an edge contour. Amini et al. [ATW88] has pointed out these problems and propose a new algorithm using dynamic programming. This method is more stable and allows the inclusion of hard constraints but it is slow and having the complexity $O(nm^3)$ where n is the number of control points and m is the size of the neighborhood in which a point can move during a single iteration. Another method, a greedy algorithm for active contours was proposed by Williams and Shah [WS92]. This new method retains the improvements of older methods and also brings a new improvement, lower complexity. The complexity of the algorithm is $O(nm)$. The control points are more evenly spaced, so the estimation of curvature is more accurate.

The greedy algorithm is stable, flexible and allows hard constraints and runs much faster than the dynamic programming method. Since it is superior to the other methods, greedy algorithm is employed in the feature specification.

3.2 Automatic Feature Tracking

The features which are important for the animation purposes are outlined in the first frame by hand. Then, by using *active contours method* [KWT88] mentioned in section 3.1, they are carried to their exact place on the image. For the other frames of the sequence, automatic edge finding process is applied to track the edges specified on the first frame.

During the edge finding process for each frame, the endpoints of snakes generally tend to move away from the corresponding features in the first frame. According to the motion of the features, they can slide back and forth along an edge. So, snakes that have a length preserving constraint are of little use for our work. Furthermore, if a feature moves far enough from one frame to another, a snake may switch edges. Because of these problems, intensive user interaction may be necessary to extract motion from video sequences.

To track and position the endpoints of a snake, Litwinowitcz et al. [PL94] introduced the use of block matching technique for the first time. After block matching technique, the endpoints of a snake are held in place and non end-points are moved by optical flow method and then energy minimization process takes place. This technique avoids the sliding of a snake back and forth between frames. The details of the block matching technique can be found in [Tun96].

As to the second problem, a snake can find an incorrect edge due to the large motion between frames. Litwinowitcz et al. [PL94] proposed the optical flow technique for the first time. Optical flow techniques generally do not produce perfect results for the motion of edges. However, after optical flow method [HS81, Tun96] is applied, energy minimization method can find the correct place as a last step. Thus, optical estimation is used to push a snake near to its desired edge.

3.3 Feature Correspondences

After the feature correspondence between the two faces is set by the animator, a scattered data interpolation should be applied to find the correspondence between all the pixels of the two images. Litwinowitcz [LW94] proposed multilevel surface reconstruction method [Ter83a], [Ter83b] to find the interpolation function, but it is slow. Furthermore, there will be no discontinuity in the function, therefore using this method is not adventegous.

Uniform cubic B-spline surfaces are a good choice because they offer nice properties such as continuity and local control. B-spline method is much simpler and faster than the energy minimization method [LCSW95].

After getting an interpolation function by using B-Spline method, for any given set of feature drawings, we can find the corresponding drawings of the face that is to be animated.

3.4 Multigrid Visual Surface Reconstruction

A control primitive's original and final shape defines a set of displacements. Namely, for a number of known (x_k, y_k) positions on the image plane, there are known displacements $(\Delta x_k, \Delta y_k)$ as defined by the original and final drawings. We should construct interpolating functions $F_1(x_k, y_k) = \Delta x_k$ and $F_2(x_k, y_k) = \Delta y_k$ to apply the image warp at each frame. Since the points (x_k, y_k) are arbitrarily spaced on the image domain, the term scattered [LW94] is used.

The thin-plate spline is one solution to our goals. Effect of a particular primitive is global but the area most affected is between primitive and its nearest neighbors. The thin-plate spline is C^1 continuous, certainly smoother than a piecewise planar triangulated surface, and not so cuspy as a Shepard's interpolant [LW94].

The solution of the thin-plate spline requires computation on each point and solving a linear system. It is extremely expensive when the number of points increases. Discretizing the problem, the solution time is dependent on the strain energy in the plate and not on the number of the data points (beyond a small initialization cost) [LW94]. The grid sizes are on the order of the image size in pixel. To get the function value at each pixel, we make sure that at least one grid element corresponds to each pixel. So the size of the grid is large and we will use coarse to fine multiresolution method to calculate our intepolants efficiently.

3.4.1 The Discrete Surface Reconstruction Problem

A closed form solution to the variational principle for visible surface reconstruction is infeasible due to the irregular occurrence of constraints and discontinuities [Ter88]. So, by using finite element model, local approximations can be performed and the problem can be discretized.

To discretize the problem, the domain of the problem and then the functionals defined for continuous problem should be discretized. Terzopoulos et al. formulize the problem by considering both depth and orientation constraints. Since only depth constraints exist in our work (Δx, Δy values), discretized formulas simplified by only considering the depth constraint [Tun96].

4 SAMPLE ANIMATED FRAMES

In facial animation, finding features and tracking them correctly is the most important part. In the study of Litwinowitcz et al. [LW94], actors in the video sequences have a make up on their faces to highlight the important details [PL94]. In this study, we do not have a video sequence recorded in a similar way. Therefore, snakes find and track edges only according to the intensities and lighting highly effects the intensities. In this work, generally synthetic facial image sequences are used as the given sequence and their features are not definite adequately. But the implemented software in finding the corresponding drawings and warping the image according to these drawings work well. This is demonstrated in all of the example figures.

In figure 1, features of the face are not very definite, also some are visualized as incomplete. Because of the light source located on the right, right half of the lower lip is indefinite, so our snake algorithm passes this part reaching through a nearest edge. Although feature drawings of the sequence are not very correct, animated drawings are highly consistent with them. Similarly, produced (animated) images show the effects of deflections of animated drawings from the original drawings. Since only the outer border of the mouth is selected, open mouth is not realized by the animation.

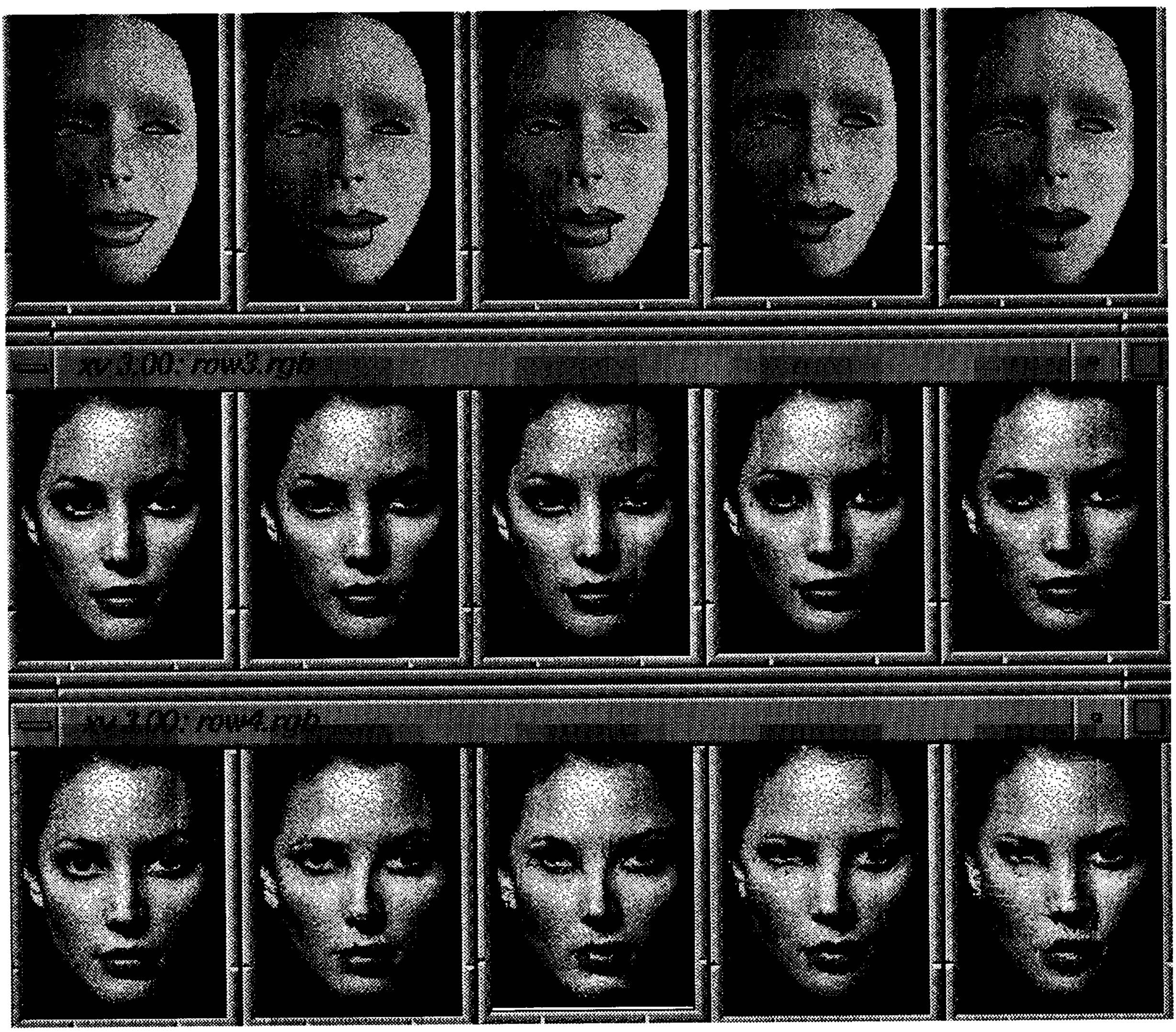

Figure 1: Tracked features, generated drawings on the original image and the animated image sequence

To realize an animation most realistically, deciding the features to be specified is highly important. In figure 2, we only want to map the change of mouth, and the corresponding mouths are selected on both images. Animated image shows a deformation mostly on the lower half of it. This part shrinks since the mouth does the same. If we could outline the borders of the faces, it would give a better result.

The same problem arises also in figure 3. The eyes, the mouth and the left and right sides of the head are specified on the first frame and the corresponding features are specified on the face of the woman. The first row of figure 3 shows the frames of the given sequence. Left eyelid closes very slowly from first frame to third and mouth shrinks as kissing. Since motion should be smooth between frames for optical flow motion estimation technique, big changes between frames are not allowed.

While the features correspond to left eye and the mouth, they effect the nearby regions on the face. There is a shadow on the left eye visualized as an eyebrow but it is only a shadow on the face. While eyelid slightly closes, this shadow remains on the same place. But, while corresponding eyelid is closing, the eyebrow on it comes down by the effect of this eyelid. The same effect is shown around the mouth, especially on the nose.

To overcome these unwanted effects, two curves, one for the left eyebrow and the other for the right side of the nose is added to the existing features in figure 4. Because of the reasons mentioned in the previous section, these newly added features are not tracked properly on the second frame. But the improvements on the eyebrow and nose are visualized clearly. In spite of the closing eyelid, eyebrow remained in its place. Nose is more close to its original appearance. If the left side of the nose could be specified properly, then more improvements could be achieved. Unfortunately, this part on the given sequence is too bright to be caught by the snakes.

5 CONCLUSION

In this work, 2D facial animation controlled by drawings is presented. By aligning curves, lines and points with features (salient objects on a face), intuitive controls for image warping are constructed. Motion is obtained basically by animating drawings and applying image warp at each frame of the sequence.

This work encourages the reuse of animated motion by gathering facial motion sequences into a database. For any single image, a sequence can be selected and animation can be realized. New features can be added at any time to both images (first frame and given image) without modifying the current mapping. By using motion sequence of a human face, non-human or synthetic faces can be realistically animated in cartoons and films. Similarly, by the motion sequence of simple characters, more complex characters can be animated. As a new consideration, by using these motion sequences, some objects other than a face can be animated by aligning some of their parts with features of the faces.

Some future works can be proposed related to this work. Multilevel surface reconstruction algorithm takes too much time, so it is the bottleneck of our animation system. Parallel implementation of this part greatly reduces the total animation time.

Extra information besides the given image to be animated helps to produce more realistic results. For example, if a neutral face image is given to be animated mouth may be opened at any animated frame. In that case, to produce realistic images, teeth should be visualized. But with our inputs, this is not possible. Small additional image information other than the neutral face could be very useful to produce the more realistic animations.

References

[ATW88] A. A. Amini, S. Tehrani, and T. E. Weymouth. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. In *Second International Conference on Computer Vision*, pages 95–99, 1988.

[For72] A. Forrest. On coons and other methods for the representation of curved surfaces. *Computer Graphics and Image Processing*, 1:341–369, 1972.

[HS81] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.

[LCSW95] S. Y. Lee, K.Y. Chwa, S. Y. Shin, and G. Wolberg. Image metamorphosis using snakes and free-form deformations. In *SIGGRAPH 95 Conference Proceedings*, pages 11–12, 1995.

[LH94] P. Litwinowizc and M. Hoch. Enhanced snakes for edge tracking. Technical Report 57, Advanced Technology Group Apple Computer, Inc., February 1994.

[Lit91] P. Litwinowizc. Inkwell: A 2 1/2-D animation system. In *Computer Graphic*, volume 25, pages 113–121, 1991.

[LW94] P. Litwinowicz and L. Williams. Animating images with drawings. In *SIGGRAPH 94 Conference Proceedings*, pages 409–412, 1994.

[PLG91] E. Patterson, P. Litwionwitcz, and N. Greene. Facial animation by spatial mapping. In *Computer Animation'91*, pages 31–44. Springer-Verlag, 1991.

[PTT94] I. S. Pandzic, N. M. Thalmann, and D. Thalmann. Real-time facial interaction. *Displays*, 15(3):157–163, 1994.

[Saj92] H. Saji. Extraction of 3D shapes from the moving human face using lighting switch photometry. In *Computer Animation'92*, pages 69–86, 1992.

[Ter83a] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. In *Computer Vision, Graphics, And Image Processing*, pages 52–96, 1983.

[Ter83b] D. Terzopoulos. Multilevel reconstruction of visual surfaces: Variational principles and finite-element representations. In *Multiresolution Image Processing and Analysis*, pages 237–310. Springer-Verlag, 1983.

[Ter88] D. Terzopoulos. The computation of visible-surface representation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 10, pages 417–438, July 1988.

[Tun96] G. D. Tunali. Animating facial images with drawings. Master's thesis, Bilkent University, Computer Engineering and Information Science Dept., July 1996.

[WS92] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 55(1):14–26, January 1992.

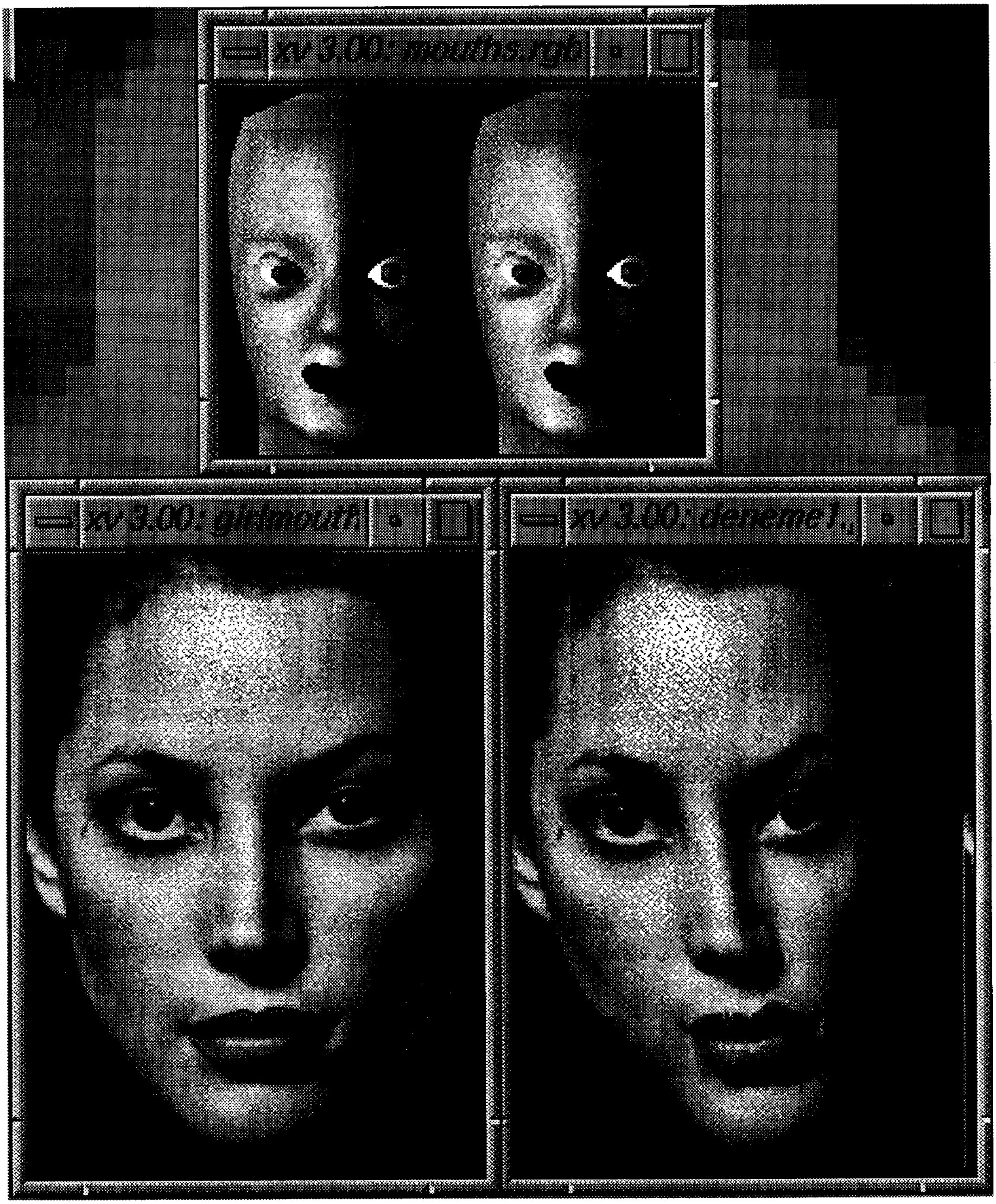

Figure 2: Generating a new frame by only specifying the mouths

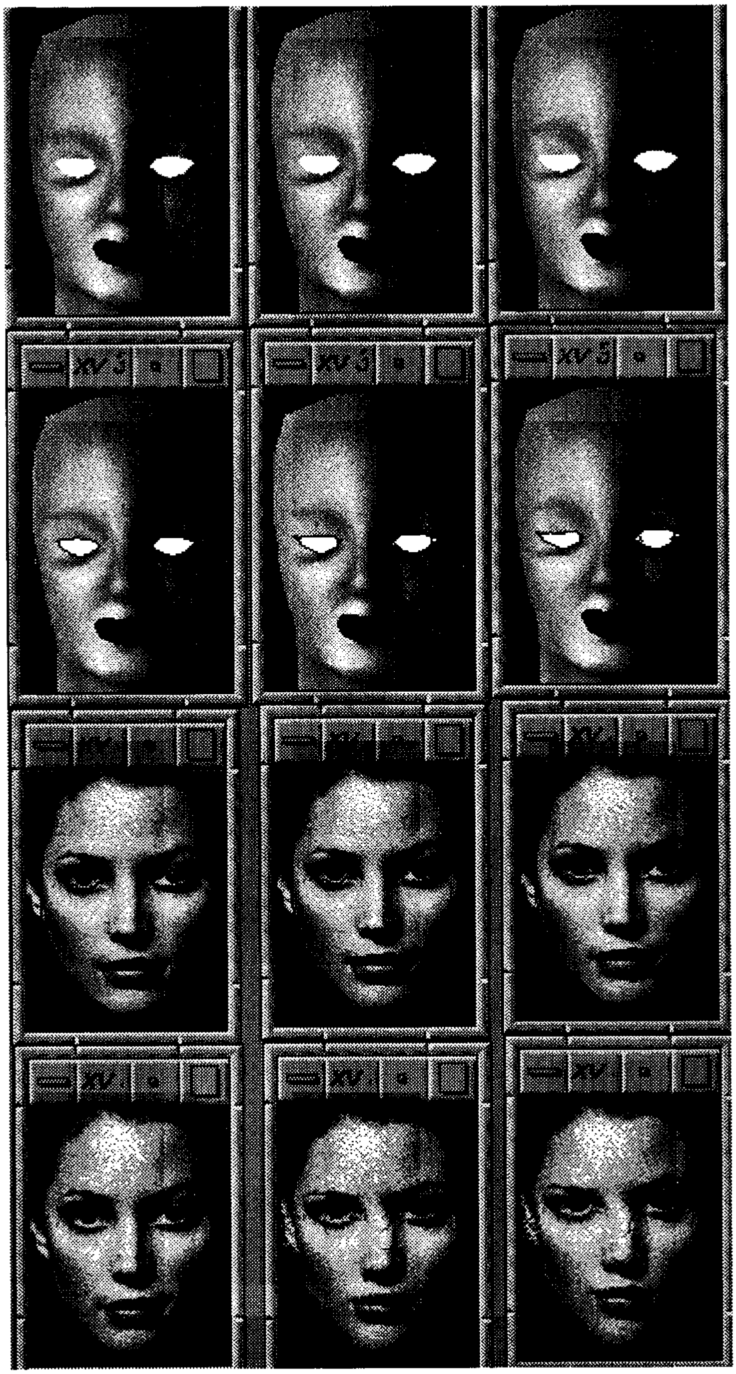

Figure 3: Change of the left eye and the mouth

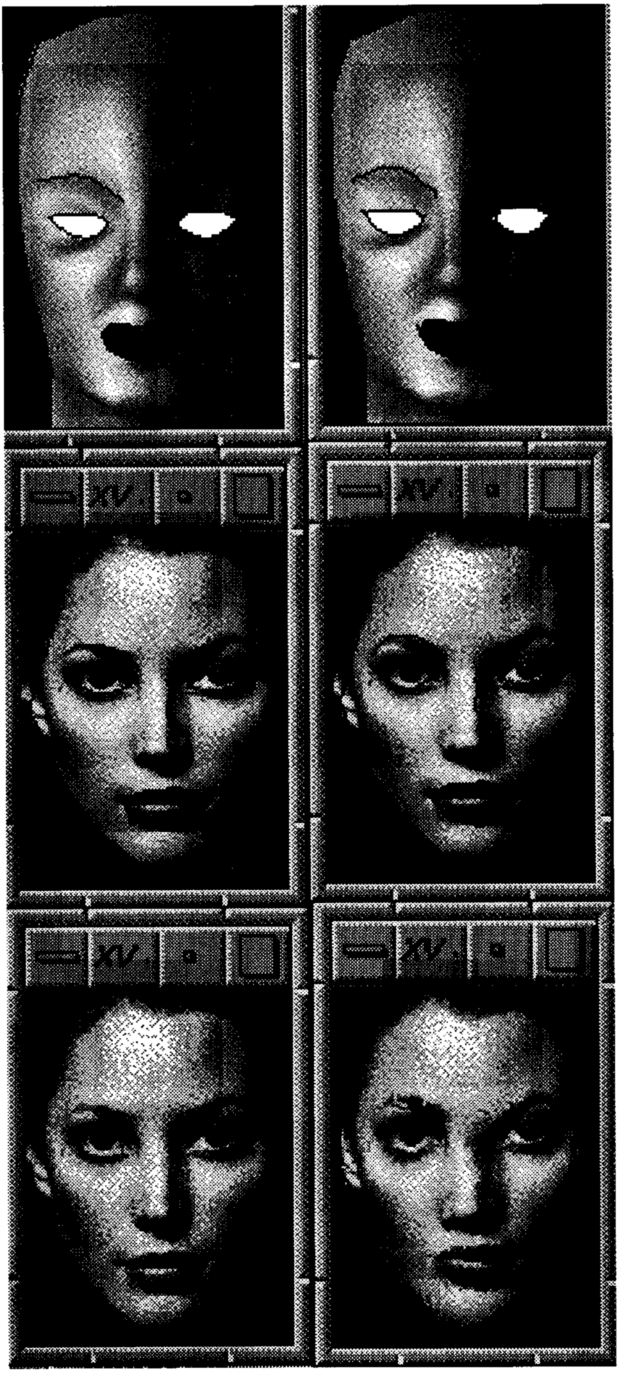

Figure 4: Adding two new features to the features of previous figure