

Performance-Driven Facial Animation

Lance Williams

Advanced Technology Group
Apple Computer, Inc.
20705 Valley Green Drive
Cupertino, CA 95014

ABSTRACT

As computer graphics technique rises to the challenge of rendering lifelike performers, more lifelike performance is required. The techniques used to animate robots, arthropods, and suits of armor, have been extended to flexible surfaces of fur and flesh. Physical models of muscle and skin have been devised. But more complex databases and sophisticated physical modeling do not directly address the performance problem. The gestures and expressions of a human actor are not the solution to a dynamic system. This paper describes a means of acquiring the expressions of real faces, and applying them to computer-generated faces. Such an "electronic mask" offers a means for the traditional talents of actors to be flexibly incorporated in digital animations. Efforts in a similar spirit have resulted in servo-controlled "animatrons," high-technology puppets, and CG puppetry [1]. The manner in which the skills of actors and puppeteers as well as animators are accommodated in such systems may point the way for a more general incorporation of human nuance into our emerging computer media.

The ensuing description is divided into two major subjects: the construction of a highly-resolved human head model with photographic texture mapping, and the concept demonstration of a system to animate this model by tracking and applying the expressions of a human performer.

Cr Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism--Animation. I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling -- Curve, surface, solid, and object representations. J.5 [Computer Applications]: Arts and Humanities--Arts, fine and performing.

General Terms: Algorithms, Design.

Additional Keywords and Phrases: Animation, Facial Expression, Texture Mapping, Motion Tracking, 3D Digitization.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

BACKGROUND

The seminal work of Parke [2] involved the photogrammetric digitization of human faces and expressions, and the creation of a parametric model that used local interpolations, geometric transformations, and mapping techniques to drive the features of a computer-animated face. Demonstrations of the face in action included lip-synchronized speech.

Platt and Badler [3],[4] applied physical modeling techniques to the parametric face problem. Once again, the goal was to drive the face at a higher level of control. Simulation of muscles in this paper established the use of deformation functions, rather than interpolation, to animate a face.

Brennan [5], working with hand-digitized 2D vector faces, implemented a kind of automatic caricature by exaggerating the differences of individual subjects from a prestored "norm."

Burson and Schneider [6] established a mapping correspondence between facial features in different photographs, a mapping defined by hand-digitizing key points of correspondence. Changes in shape and texture between one pair of photographs could then be used to map changes in shape and texture to a third. This process has been used to artificially "age" the photographs of missing children (that is, to estimate an image of the same child some years hence). Similar mappings have been used by Burson and Kramlich (Face Software, Inc., NY., NY.) to interpolate faces and create composites.

Concurrently, experiments at New York Institute of Technology involved mapping 2D animated features onto 3D characters (James Blinn and the author, 1976), and mapping of live-action video faces onto 3D "masks" (3D face surface models in which the eye and mouth regions have been smoothed over, so as not to interfere with the moving lips and blinking eyes of the mapped video). Live-action mapping was first essayed by Duane Palyka and the author in 1977, and was applied by Paul Heckbert in the NYIT videos, "Adventures in Success," and "3DV" (1984). At the same time, NYIT researcher Tom Brigham was doing extensive work with screen-space (2D) interpolation of texture and form. Brigham's "shape interpolation" was applied to faces, among other subjects, and must be considered an influence on this work. One conclusion of these experiments was that much of the detail and realism of the faces depicted was simple surface texture. A face model like Parke's would be much more powerful and convincing with photographic mapping, and perhaps more individual and personable, as well. The photographic mapping of [7] illustrates the power of texture alone, without surface shading of any kind.

An extended, expressive facial animation was essayed by Phillippe Bergeron, Pierre LaChapelle, and Daniel Langlois in 1985, in their short, "Tony de Peltrie"[8]. Photogrammetrics as in [2] were used to digitize both human expressions and the neutral features of a stylized model. Then, scaled differences of the various human expressions from the "neutral" human expression, were applied to the neutral stylized model. This basic cross-mapping scheme, as in [5], requires a norm to apply the mapping. Since the mapped differences are scaled, "caricature" (this time in 3D) is straightforward.

The most recent wrinkle in facial animation is Keith Waters' application of tailored local deformation functions analogous to musculature in animating a neutral face definition [9]. This technique was adopted by the Pixar animation team to control the patch-defined baby face of "Tin Toy" [10].

The formulations described in this paper are an attempt to extend the mapping of texture and expression to continuous motion mapping. Using current technologies, both human features and human performance can, in the opinion of the author, be acquired, edited, and abstracted with sufficient detail and precision to serve dramatic purposes.



Figure 1. Processed model with mosaic texture map

CONSTRUCTING THE MODEL

Dancer Annette White, a personal friend of the author, was a featured performer in Patrice Regnier's RUSH dance company which performed at SIGGRAPH in 1985. Her head was cast in plaster by Long Island sculptor Val Kupris, and the surface of the plaster head digitized at Cyberware, Inc. [11]. A number of conventional color 35mm photos of Annette were taken at this time by Nancy Burson of Face Software, Inc., with the intent of creating a complete composite photograph (in cylindrical projection) of Annette's head to match the scanned relief data of the plaster cast. Face Software shared an interest in developing electronic mask technology, and cooperated in the initial test of the model. They were able to apply their mapping techniques to warp sections of the multiple photos to the range data acquired from scanning the plaster head. To facilitate this process, the 16-bit range data was compressed to eight bits, and a Sobel filter applied. This "edge enhancement" filtering brought out the important features (edges of the lips, corners of the eyes) from

the otherwise ghostly range image. The mapping, based on a triangular mesh, is shown applied to the model in figure 1. Below the two projections of the model in the photo are the composite texture map, produced by Face Software, and an eight-bit compression of the range image after Sobel filtering, to the right. (The black regions in the filtered range image are the result of a color map artifact; they should be white.) A kerchief is worn about the head in this texture map. A shading discontinuity between photographic sections is visible in the composite map on the left.

Cyberware had been sent a plaster cast of Annette's head, and sent back a tape with the digitized surface data, scanned four times. Data were in the form of 16-bit integers defining a mesh surface in a cylindrical coordinate system, each entry a radius indexed by phi (azimuth) and Y. The mesh consisted of 512 columns of samples in phi, by 256 rows in Y. The mesh was in fine shape to serve as a map with which to register the photographic texture, but in the meantime, required some additional work to serve as a satisfactory 3D model.

The usual fashion in which Cyberware displayed digitized data, at the time this work was undertaken, was as a faceted model. Missing data -- data obscured from the sensor by occlusion of the projected laser line -- was mapped to the center of the Y axis, and was in most cases invisible. If the data were Gouraud shaded in a straightforward fashion, the missing samples would severely impact the visible ones, by clobbering the computed surface normals. In order to make smooth shading and texture mapping possible, it was necessary to refine the model: to "heal shut" the missing data, by interpolating the surface across neighboring samples. It was moreover desirable to apply a certain amount of data smoothing to suppress spurious texture, and to use larger local neighborhoods of samples to estimate surface normals. Such processing of the digitized surface, very similar to image processing, proved essential for subsequent mapping and shading. An outline of the surface processing steps follows.

Healing Shut the Missing Data

Missing data is marked, in the Cyberware format, by a reserved "Z" value. The first step in surface processing is to restore surface continuity, a prerequisite for many subsequent operations. The scanner data is copied into a floating point array with missing values set to 0.0, and a second floating point array, a matte, is set to 1.0 where valid data exists, and 0.0 elsewhere. In the vicinity of 0.0 matte values, a small blurring kernel is applied to both the matte and the surface. Where the matte value increases (due to the blur) above a threshold, the surface sample is replaced by the blurred surface divided by the blurred matte. Filtering is recursively applied until no matte values are below threshold. Thus, the missing data are replaced by smooth neighborhood estimates, that gradually grow together. An implementation of this basic general scheme was coded so that the smooth replacement regions could "grow outward" slightly, smoothing the correctly-acquired data at the boundaries. All iterations of the "healing" process were visible, and the matte threshold, outward creep, and blur kernel could be varied for different results. Practically speaking, the process worked well for the small regions of missing data on the digitized face. It could not be expected to perform well for missing regions much larger than the blur kernels used. For this reason, a more detailed discussion of the filtering algorithm would not be justified. For future work, the methods of [12] are preferred.

Hysteresis Filtering

In many types of data acquisition, noise at the scale of the sampling interval is particularly bothersome. Special



Figure 2. Processed peripheral photograph.

techniques for removing "salt and pepper" noise from imagery have been adopted for many purposes. A typical model will dampen high-amplitude sample-frequency noise in a nonlinear way. For the filtering used in the head model, a smooth estimate of the surface at a point was computed by a 3x3 blur kernel with unity gain and 0.0 as the center sample coefficient. If the estimate differed from the center sample by greater than a threshold amount, the center sample was replaced by the estimate. Such a filter is occasionally called a Tukey filter, or a "hysteresis" filter (because it modifies the data only when the threshold is exceeded).

Subjectively, this filtering process smoothed out a number of glitches and spurious details in the surface of the face. It performed equally well on small bubbles in the plaster cast and on data that, for whatever reason, seemed somewhat rougher than the plaster.

Low-Pass Filtering

By filtering and downsampling data, it is possible to trade precision for resolution directly. This tradeoff occurs in a continuous way as data are low-pass filtered. The reduced bandwidth of the surface is accompanied by an increase in precision for each of the samples. Where the data to be estimated are more bandlimited than sources of noise in the system (where the signal is pink, and the noise is white), low-pass filtering will increase the signal-to-noise ratio.

A weak low-pass filter was applied to the cylindrical R (range) coordinates of the data, before conversion of the surface from $R(\theta, Y)$ to $X(u, v)$, $Y(u, v)$, and $Z(u, v)$. The skin seemed smoother after filtering, and faint striations visible in the surface normals (assumed to be scanner artifacts) diminished.

Filtering the Normals

In fact, surface normals are extremely sensitive indicators of perturbations in the digitized surface, including all noise and artifacts. The usual methods of computing surface normals from polygon meshes are too local for imperfect data so closely spaced. It is reasonable to smooth such local normal estimates, or to use normals of a surface somewhat smoother than the one actually displayed. In this way, satisfactory shading can be achieved without smoothing features from the surface.



Figure 3. Sobel-filtered range image warped into a rage.

When filtering is applied to the range data in the cylindrical mesh, artifacts occur at the poles, where sample aliasing is greatest (because the laser line may project very steeply). Because the spacing of the samples is far from uniform in 3D space, such filtering is very anisotropic. On the other hand, filtering the components of the derived normals is much more expensive, and requires special care. Some normals may vanish (magnitudes go to zero) after filtering; all normals require re-unitization.

Some polar artifacts were tolerated in the range-filtering stage, with the intent of completing the top of the head with a "scalp mesh." This makes sense because the texture, as well as the model, would otherwise be very poorly resolved on top of the head. Despite the difficulties imposed by the "singularity," the polar map is a very convenient representation for interacting with the head model. It provided the basis for interactive texture mapping, and interactive setup of expressions.

Registering Cylindrical Texture

Most of the model processing steps described could be performed automatically. The most tedious process described so far was that performed by Face Software: taking the photographs necessary to completely record the subject's head, and then painstakingly registering the mosaic of photos to the model range data. Ideally, this step, too, would be automatic. A scanner which simultaneously captured R, G, B, and range, would simplify "mask" acquisition greatly. Such a scanner has only recently become commercially available[11].

In order to generalize the texturing process, as well as to exercise the "virtual muscles" expected to drive the face, an alternate method was devised. The first step was to capture the head texture completely in a single photo. Traditional photographic methods exist, based on slit-scan cameras. A camera which exposes its film by moving it continuously past a vertical slit, while the camera and lens rotate horizontally, is termed a "panoramic" camera. If the camera is stationary while the subject rotates on a turntable, the device is termed a "peripheral" camera. Panoramic cameras are commercially available, and adapting one for peripheral photography should be straightforward. The peripheral camera used in this work was built by Prof. Andrew Davidhazy of the Rochester Institute of Technology. Prof. Davidhazy was kind enough to

photograph Annette's head while she stood on a rotating turntable; her dance experience proved very useful in maintaining a vertical and unwavering posture during the spin. The resulting photograph, like the Cyberware scan, is a cylindrical projection.

The peripheral photograph was scanned and digitized. An interactive texture-warping program was written for an SGI Iris workstation. Using the 12-bit double-buffered display mode, the user could ping-pong back and forth between a Sobel-filtered range image and the peripheral photo. Standard digital "painting" functions were provided by the program, but its key feature was a "coordinate airbrush." This is simply a deformation function in the style of [7] and [9], described in digital painting terms. In fact, the "coordinate airbrush" implements an inverse, rather than a forward, mapping. For many purposes this is equally convenient, and much more rapid to compute. The idea is that an X, Y coordinate offset is supplied for the center of the brush (so the center pixel is replaced by the pixel value in the image at the offset X, Y), and the airbrush kernel tapers the blend of the coordinate offsets away from the center of the brush. If the offset is larger than the span of the airbrush kernel, a singularity results, like a dimple in a specular surface. In fact, these

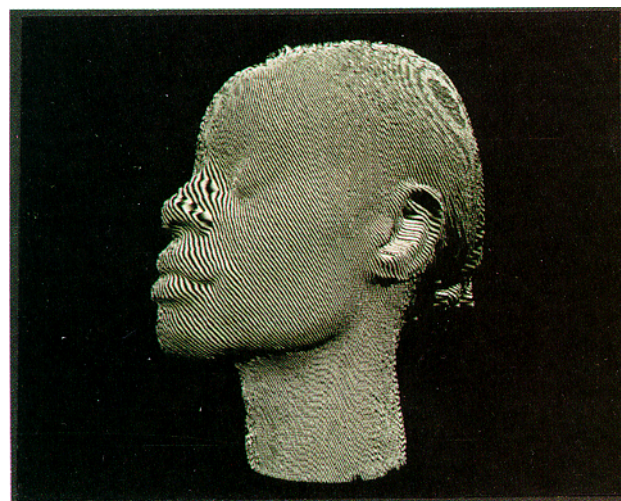


Figure 5. "Zebra" triangulation.

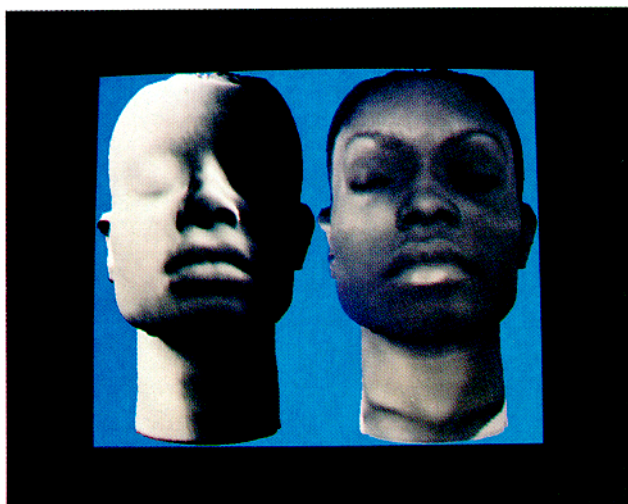


Figure 4. Final model, with and without texture.



Figure 6. "Serpentine" triangulation.

inverse mappings resemble nothing so much as reflections from curved mirrors. If the offset is smaller than the span of the brush kernel, then the mapping behaves much like the "forward" mapping, like a rubber sheet that can be stretched to fit points of correspondence together. Figure 2 shows the final texture after remapping. The general technique was to use large kernels first, and then use smaller and smaller warps to register the details. Registering the texture with the filtered range image took about three hours of interaction. Figure 3 shows a Sobel-filtered range image which has been interactively warped to change its neutral expression to one of profound annoyance (the contouring in the image is a result of the compression to 8 bits). This illustrates the primary motivation for development of this "warware": to apply similar deformation functions to animation of the 3D model.

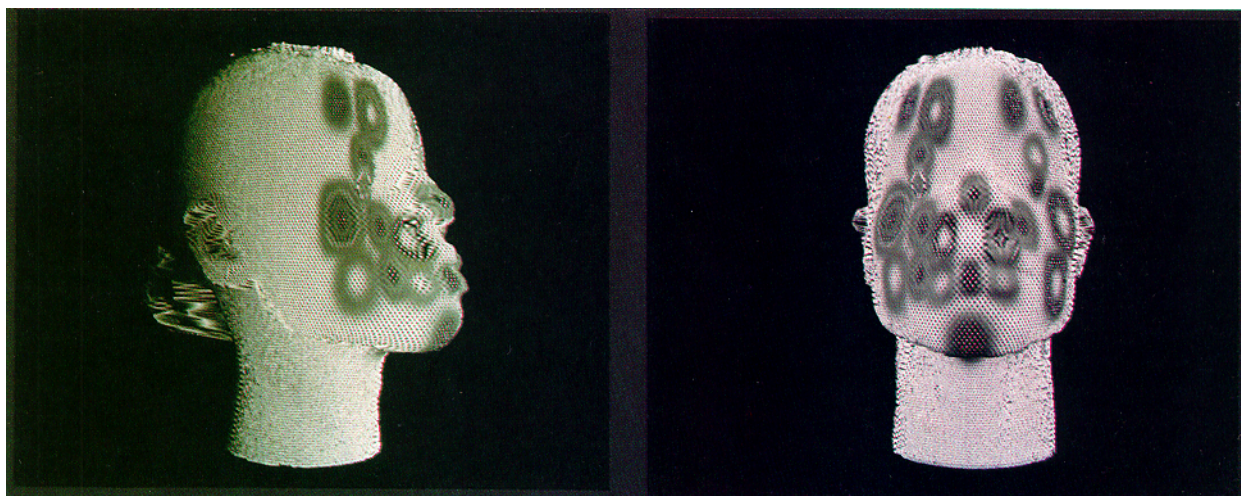
Figure 4 shows the model with and without texture. Rendering was performed using the SGI Iris graphics pipeline, and supporting rendering software by G. W. Hannaway & Associates. With a generation of graphics hardware that does not handle texture explicitly, many users effect "texture

mapping" by rendering large meshes of polygons, with texture multiplying intensities at each mesh vertex. A mesh of vertices with the same resolution as the texture map essentially reproduces the map, warped to shape, with bilinear interpolation (Gouraud shading). To compress the map, filtering is necessary. In approaching texture this way, which is certainly a practical expedient on today's Z-buffer polygon engines, mesh tessellation is an issue. Figure 5 shows the head model textured with a pixel-scale checkerboard. The striping of the texture is due to triangulation. To avoid sending twisted quadrilaterals, all mesh elements are first divided into triangles. In figure 5, the triangulation diagonals cut across the mesh in parallel, and become stripes. In figure 6, the same texture is rendered with a "herringbone" or "serpentine" triangulation. Alternate mesh quads are divided on alternate, criss-crossing diagonals. Note that the texture still does not look like a checkerboard (we could hardly expect it to at this scale), but is much more isotropic.

After the mapping was corrected, an animation of the head rotating about the Y axis was taped, and the result examined



Figures 7-9. Rotations of the mapped 3D model.



Figures 10-11. Basis functions on the model face.

for registration (which looked good!). Figures 7-9 show the model rotating, exhibiting the mapped texture in 3D.

MODIFYING EXPRESSION

The next step is applying our warware to animation of the model. First, a set of warping kernels is distributed about the face. The factors that govern this distribution include physiology (the placement of muscles in the face, and the location of the most motile areas) as well as various practical considerations relating to the planned use of spot-tracking on an actor's face to ultimately drive the model. The "basketweave" texture of figure 6 has been applied to the faces of figures 10 and 11, and the set of basis functions driving the face has been slightly offset. The resulting interpolations display the centers and relative sizes of the control kernels. The kernels are larger than they appear, but they are radially symmetric (like the basis functions of [7], rather than the more

elaborate bases used by [9]). Each is a Hanning (cosine) window, scaled to 1.0 in the center, and diminishing smoothly to 0.0 at the edge, with no negative values.

Like the enraged expression applied to the range image of figure 3, the expressions of figures 12 and 13 were created by hand-warping (with the "coordinate airbrush") the polar representation of the model. The realistic model can be stretched in a completely unrealistic way, and actually resembles a latex mask in some respects.

Tracking Expressions of Live Performers

The final link in our proof-of-concept demonstration is to increase realism by deriving the basis-function control offsets from the expressions of a live actor. Video-based tracking was the method of choice, although various mechanical schemes were considered. Video offers the most leverage



Figure 12. "Comedy"

Figure 13. "Tragedy"

Hand-animated exaggerated expressions.

because it does not restrict the performer, it is a simple, widely-available technology, and promises ultimately to permit motion-tracking without special makeup or fiducial marks.

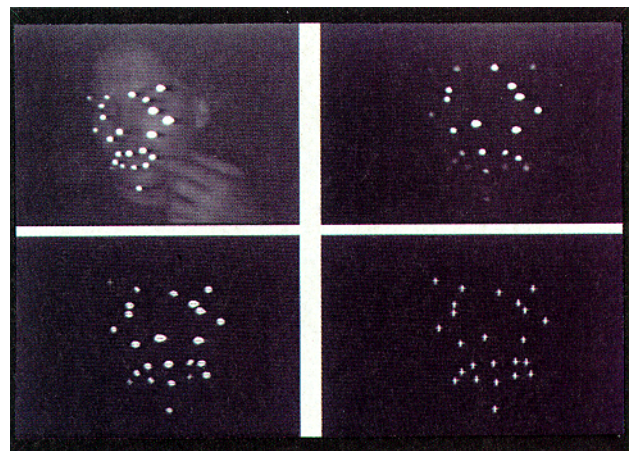
The "special makeup" in this case was Scotchlite®, a retroreflective material manufactured by 3M. Scotchlite uses a layer of tiny spheroids to behave much like a surface of tiny corner-reflectors; reflection efficiency is very high within a small angle from the incident light. A beam-splitter setup was employed to record the performer, which for budgetary reasons turned out to be the author. The beam-splitter is simply a sheet of window glass, between the camera and the performer, at 45° from the camera's optical axis. At right angles to the camera, aimed at the performer's side of the glass, was a slide projector, used as a light source. The setup is adjusted so that the light reflected from the glass illuminates the performer's face. The point of the apparatus is, that the light source and camera are coaxial. Since the light comes "from the camera's point of view," the efficiency of retroreflectors in the camera's field of view is very high. When properly set up, contrast and brightness can be adjusted so that the camera can see retroreflectors in its field of view, and little else.

A paper punch was used to make little round adhesive spots from Scotchlite tape; figure 14 shows the author applying spots to his face (the brightness of the image has been scaled so that the face is visible; originally, only the spots could be seen). A typical frame (once again, brightened) is illustrated in figure 15.

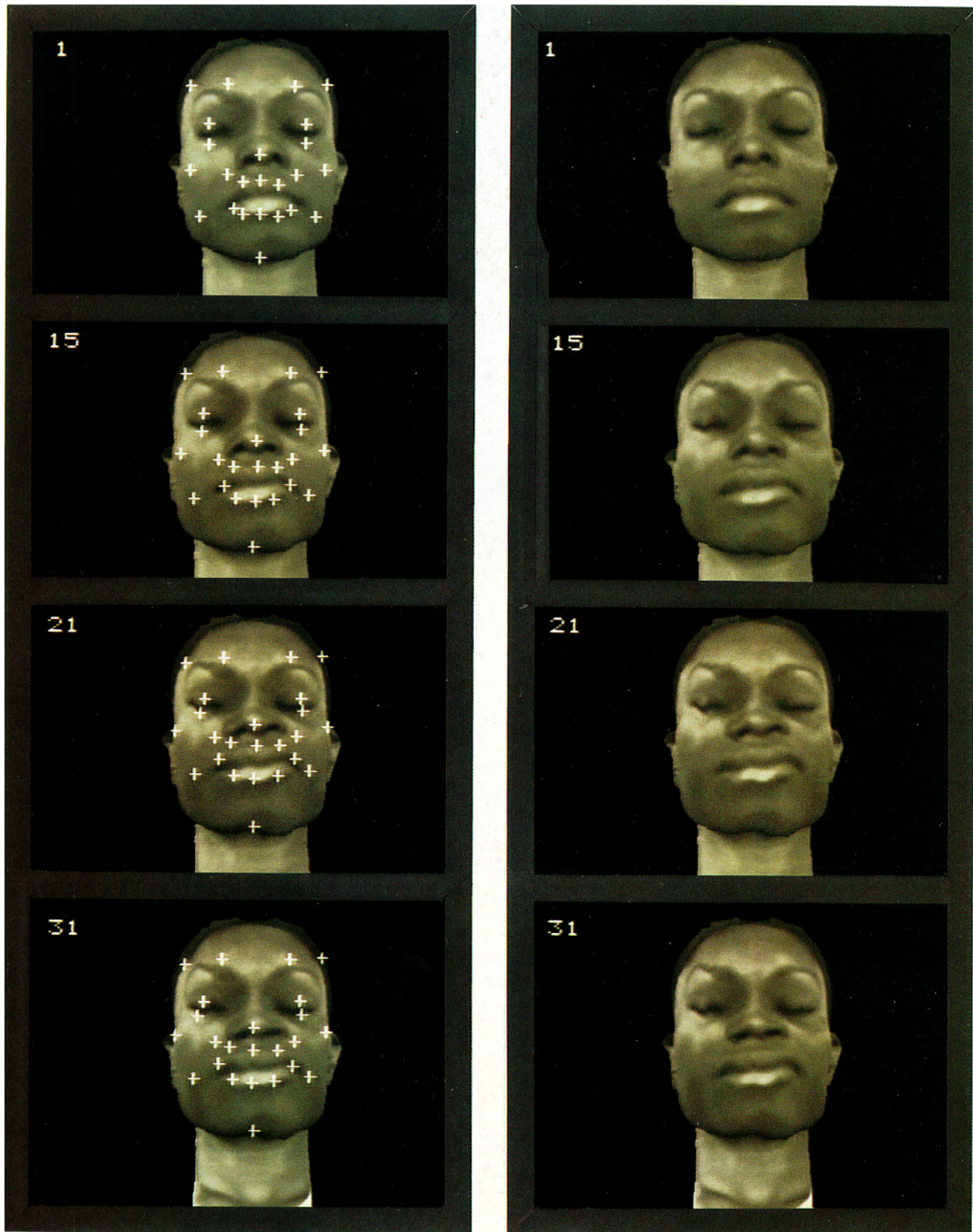
The problem of digitizing the actor's performance now becomes one of tracking a set of bright spots in a dark field. The special case of tracking spots on a face which faces the camera is particularly favorable; the spots (if placed wisely) will never actually touch one another, and can never be obscured by another part of the face. For the test animation, the head was held relatively still, and only the facial expression changed. The algorithm used was to have a human operator indicate the position of each spot in the initial frame of a digitized sequence. This takes care of the correspondence between each spot and the basis function on the face that it controls. Spots are tracked automatically in subsequent frames. The spot tracking routine outputs the X, Y

coordinates of the spot centers in each frame.

When the operator selects a spot, it is a matter of touching the spot, or the spot's near vicinity. A window about the point indicated is scanned, and the X, Y coordinates of each pixel are multiplied by the intensity of the pixel; running sums of the X and Y pixel intensity products are saved. A running sum is also kept of the pixel intensities. When the window is completely scanned, the summed X, Y's are each divided by the summed intensities. This supplies the window's "center of gravity" or first moment, a fairly robust estimate of the center of light intensity in the window, which has fractional pixel precision if the spot falls across a reasonable number of pixels. A new window is then scanned from the computed center, and the process iterated a few times. The window size should be slightly larger than the spots, so that the iterated result will be the first moment of the spot itself. In figure 16, tracking crosses have been superimposed on the spots. Note that crosses appear even where the spots they mark are very dim; the spot tracker proved robust, and the motion of the tracking crosses is quite convincing even for spots of marginal



Figures 14-17. Tracking spots on performer's face.



Figures 18-25. Tracked expressions applied to the model face.

visibility. Figure 17 shows the tracking crosses without the spots. Figures 18-25 show the model face being driven by the tracked spots, marked by the tracking crosses in one of each pair of images.

CONCLUSIONS AND FUTURE WORK

The appearance of the model head is quite realistic, and the test animation is very striking. Although short, the sequence exhibits some lifelike twitches and secondary motions which would be unlikely to arise in pure animation. The fundamental idea of mapping the motions of a live performer to a computer animated character promises to be a rich one. Previous efforts to animate faces by interpolating between various canonical expressions can now be supplemented by interpolation of canonical motion sequences. Driving a face or head of very different proportions or physiognomy should be attempted soon.

The animation of the face is very much a "proof of concept," not a completely realized system. At present, the face cannot open its eyes or mouth, and this portion of the model will demand a great deal of time. Some ripples on the eyes (visible in the last two figures) result from the fact that the performer could blink his eyes (in fact, found it hard not to) and the model could not. The test did establish that eyeblinks are quite trackable!

For reasons of efficiency, and to test the validity of the simplifications, the performer's face was tracked in 2D, and the result "projected" to the model, which was animated in the cylindrical coordinates of the range data before being converted into X, Y, Z meshes with normals. The changes in the performer's expressions were transferred to the model with appropriate scaling of the offsets, and a straight projection onto the face approximates projection onto a cylinder. A test setup with angled mirrors at the sides of the actor's head showed that 3D coordinates could be acquired with a single camera. A more fully-realized system would track expressions in 3D and apply them in Cartesian, rather than cylindrical, coordinates.

ACKNOWLEDGEMENTS

My profuse thanks go to Annette White, the model; to Nancy Burson and David Kramlich, my initial collaborators; to Ariel Shaw and Andrew Davidhazy, for extraordinary photographic assistance; to Wyndham Hannaway and Bill Bishop of G.W. Hannaway & Associates, who made available much of the computer time, utility software, and exotic paraphernalia this research required; to my manager, Mark Cutter, and to Ned Greene, Pete Litwinowicz, and Libby Patterson of Apple's Advanced Technology Animation Group, for constant support, advice, and inspiration.

REFERENCES

- [1] Walters, Graham, The Story of Waldo C. Graphic. ACM SIGGRAPH '89 Course Notes, *3D Character Animation by Computer*, August 1989.
- [2] Parke, Frederick I., A Parametric Model for Human Faces. Ph.D. dissertation, Department of Computer Science, University of Utah, 1974.
- [3] Badler, Norman, and Platt, Stephen, Animating Facial Expressions. Proceedings of SIGGRAPH '81 (Dallas, Texas, August 3-7, 1981). In *Computer Graphics 15*, 3, (August 1981), 245-252.

- [4] Platt, Stephen Michael, A Structural Model of the Human Face. Ph.D. Department of Computer and Information Science, School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, PA., 1986.
- [5] Brennan, Susan Elise, Caricature Generator. M.S. Visual Studies, Dept. of Architecture, Massachusetts Institute of Technology, Cambridge, MA. Sept. 1982.
- [6] Burson, Nancy, and Schneider, Thomas, "Method and Apparatus for Producing an Image of a Person's Face at a Different Age," U.S. Patent #4276570, June 30, 1981.
- [7] Oka, Masaaki, Tsutsui, Kyoya, Ohba, Akio, Kurauchi, Yoshitaka, Tago, Takashi, Real-Time Manipulation of Texture-Mapped Surfaces. Proceedings of SIGGRAPH '87 (Anaheim, California, July 27-31, 1987). In *Computer Graphics 21*, 4, (July 1987), 181-188.
- [8] Lachapelle, Pierre, Bergeron, Philippe, Robidoux, P., and Langlois, Daniel, *Tony de Peltrie*. [film] 1985.
- [9] Waters, Keith, A Muscle Model for Animating Three-Dimensional Facial Expression. Proceedings of SIGGRAPH '87 (Anaheim, California; July 27-31, 1987). In *Computer Graphics 21*, 4 (July 1987), 17-24.
- [10] Lasseter, John, Ostby, Eben, Reeves, William, Good, Craig, Rydstrom, Gary. *Tin Toy*. [film] Pixar, 1988.
- [11] Cyberware Laboratory, Inc.: 4020/PS 3D Scanner, 4020/RGB 3D Scanner with color digitizer. 8 Harris Court 3D, Monterey, California 93940.
- [12] Burt, P.J., Ogden, J.M., Adelson, E.H., and Bergen, J.R., Pyramid-Based Computer Graphics. *RCA Engineer*, Vol. 30, 5, Sept.-Oct. 1985.

