Animating Images with Drawings

Peter Litwinowicz†
Lance Williams†
Apple Computer, Inc.

ABSTRACT

The work described here extends the power of 2D animation with a form of texture mapping conveniently controlled by line drawings. By tracing points, line segments, spline curves, or filled regions on an image, the animator defines features which can be used to animate the image. Animations of the control features deform the image smoothly. This development is in the tradition of "skeleton"-based animation, and "feature"-based image metamorphosis. By employing numerics developed in the computer vision community for rapid visual surface estimation, several important advantages are realized. Skeletons are generalized to include curved "bones," the interpolating surface is better behaved, the expense of computing the animation is decoupled from the number of features in the drawing, and arbitrary holes or cuts in the interpolated surface can be accommodated. The same general scattered data interpolation technique is applied to the problem of mapping animation from one image and set of features to another, generalizing the prescriptive power of animated sequences and encouraging reuse of animated motion.

Keywords: Image warping, animation, scattered data interpolation.

Background

Rich detail and texture are usually reserved for the background paintings in an animation. Production economics do not permit the foreground figures to be dressed in plaid, for example, and such effects are difficult to achieve by traditional means. Three-dimensional computer animation offers shading and texture, but the stylization of form possible in traditional cel animation has proved more elusive. Much of the motivation for the work described here comes from the technique of traditional animation, where all action is portrayed by drawings -- points, lines, and curves -- defined at arbitrary instants of time. Interpolation defines the full sequence from the sparse keys. We envision a similar interpolation in space, embedding the objects and characters the artist has drawn in a surface controlled by the lines of the drawings. The interpolated motion of corresponding keyframe drawings is used to define spatial deformations which may be applied to other images. "Feature-based" deformations, controlled by the motion of arbitrarily-placed points, curves, and regions, offer a direct and natural method of animating complex images and forms. An earlier attempt at animating drawings by their features [Litw91] required the user to define a mesh of bilinear Coons patches [Forr72]. The curved boundaries of the patches could be aligned with features of interest to the animator, and subsequently animated to control the image. Although the Coons patches are inexpensive to evaluate, the manual division of the image into a mesh, and the necessity of animating all of the patch boundaries to control the motion, require substantial time and effort. Specifying and animating only the features of interest is both vastly more general and a great deal easier for the animator.

Deformations based on tensor-product splines [Sed86][Farin90] are actually a more recent development than "feature based" deformations defined by line segments, which were introduced by Burtnyk and Wein in [Burtn76]. The goal of that work was to permit an animated "skeleton" of linked line segments to drive the animation of a drawing, in this case by polygonal tessellation of regions around the "bones" of the skeleton. An alternate parametrization, based on a skeleton derived from the shape of the matte or support of the image region to be animated, has been described by Wolberg [Wolb89]. In this case, the skeleton is the result of successive thinning operations applied to the original shape. The image warping algorithm is specialized for morphing, that is, for transforming between two image/shapes. Driving an image warp by modifying the skeleton alone would require a slightly different algorithm. Automatic "medial axis" skeletons of this type might be useful for some purposes, but there is no guarantee that the "bones" will align with features the animator is interested in controlling directly.

More recent skeleton animation work appeals to smoother interpolation functions. Van Overveld [vanO90] describes a physical simulation which is calculated for a simple skeleton, then applied to a more complex model by a distance-weighted "force field." The field is defined by a dense set of points on the limbs of the skeleton. The formula used is equivalent to Shepard's interpolation, a simple scattered data interpolant originally developed for terrain surfaces [Shep68]. In [Beier92], Beier and Neely developed an algorithm for image morphing based on Shepard's interpolation, with a significant novelty: the control primitives were extended to include line segments as well as points. Since the line segments can be aligned with important edges in the image, the metamorphosis was termed, "feature based." At edge-like features of the image, a single line segment does the work of dozens of points, and offers a natural and intuitive means of interpolating local orientations.

Thin-plate spline surfaces were introduced to computer-aided geometric design by Harder and Desmarais [Hard72]. Application of finite-element methods to computing smooth surfaces over

†c/o Apple Computer, Inc.
1 Infinite Loop, MS 301-3J
Cupertino, CA 95014 USA
email: Pete Litwinowicz: litwinow@apple.com
Lance Williams: lance.w@apple.com

scattered data for CAGD purposes was first essayed by Pilcher [Pilch74]. Smooth scattered data interpolants, introduced as analogues of physical surfaces, have more recently been applied in vision and image reconstruction [Grim81]. The demands of rapid processing for practical vision systems has motivated attempts to compute some of these surfaces using fast numerical methods [Terz88], and our animation system utilizes these techniques. We have implemented a system which performs the scattered data interpolation for animated deformations or morphing by using multigrid finite-difference evaluation of a thin-plate spline surface. This approach extends the "feature" primitives to curves and solid regions, realized as densely sampled points. In addition to generalizing the control primitives, the underlying surface which defines the deformation is better behaved than a Shepard's interpolant. This is particularly important when deformations are used for animation, without the texture interpolation invoked in a "morph."

Description of the problem and our solution

Given starting and ending shapes for a set of primitives in the plane, such as curves, lines and points, we would like to calculate a warp that transforms regions between the primitives in a well-behaved and intuitive way. By aligning curves, lines and points with features in an image, intuitive controls for image warping are easily constructed. Deformation of an image can then be accomplished by applying the warp defined by the original drawing and any other drawing of the same features. It is then possible to animate an image simply by animating the drawing, and applying the corresponding image warp at each frame.

From a control primitive's original and final shape we can derive a set of displacements. For a point the displacement is simply a $(\Delta x, \Delta y)$ pair. For a line or polyline, continuous displacements all along the length are defined by the initial and final shape. Not only the "skeleton animation" of Burtnyk and Wein [Burtn76], but a number of subsequent facial animation systems and morphing programs are based on triangulation of displaced points [Gosh86]. For interpolating a set of scattered points, Delaunay triangulation is frequently used. A triangulation is defined for the original feature set, then the vertices are interpolated toward the final shape, and the triangles texture-mapped from the original image. While rapid to compute, the warp is generally not as smooth as desired. The triangulation can be seen in the resulting animation, as the texture map shears along the edges of the triangles.

Beier and Neely [Beier92] advanced a modified Shepard's interpolant which added line segments as control primitives. This method interpolates displacements using a distance-weighted technique and produces smoother interpolations than triangulation. The usual difficulty with these distance-weighted interpolants is trading off "cusps" against "flats" at the data points. In addition, the interpolation may become very expensive as the number of primitives increases, since each contributes at every point on the surface. To give the animator local control, Beier and Neely associate a finite region of influence -- a threshold distance from a point or line segment -- with each primitive. The process of specifying the region size for each primitive can potentially be tedious, and for many warps, no combination of region extents and inverse-distance weighting exponents yields the desired result [Rupr92].

Instead of explicit control over the size of the basis functions used in the interpolation process, our goal was to provide a technique which automatically extended regions of influence to the next user-defined primitive. Another goal was to have a nice "smooth" interpolant, but at the same time provide a mechanism for intuitively introducing discontinuities in the interpolant where appropriate. Finally, we wished to provide curves as deformation primitives.

The thin-plate spline provided a nice compromise for our goals. The region of influence for a particular primitive is global, but the region most affected is the area between a primitive and its nearest neighbors. The thin-plate spline is C^1 continuous, certainly smoother than a piecewise planar triangulated surface, and not so potentially cuspy as a Shepard's interpolant.

The thin-plate spline is one solution to a class of scattered data interpolation problems that have the following problem statement (from [Franke79]): "Construct a smooth bivariate function, F(x,y), which takes on certain prescribed values, $F(x_k,y_k) = f_k$, k = 1, . . . , N. The points (x_k,y_k) are not assumed to satisfy any particular conditions as to spacing or density, hence the term 'scattered.'"

How does our problem map onto the scattered data problem? For a number of known (x_k,y_k) positions in the image plane, we have known displacements $(\Delta x_k,\Delta y_k)$ as defined by our original and destination drawings. Substituting Δx_k for f_k above, we calculate a smooth interpolating function for the x-displacements for an entire image, and similarly for the y-displacements. The thin plate also has the added constraint that the surface everywhere should minimize the following smoothness functional:

$$\iint_{\Omega} \frac{\partial^2 F}{\partial^2 u} + 2\frac{\partial^2 F}{\partial u \partial u} + \frac{\partial^2 F}{\partial^2 v}\, dudv$$

where Ω is the domain of the surface, and F is the surface itself. Encoded in Ω are the cuts and holes in the surface.

The thin plate spline can be solved by using a $d^2 \log d$ basis function at each point (where d is the distance from the point), and solving the linear system. This becomes extremely expensive as the number of known points increases. By solving the problem on a discrete grid, the solution time is dependent on the strain energy in the plate and not on the number of data points (beyond a small initialization cost). Another advantage to discretizing the problem is that discontinuities in the interpolant are easy to handle. In the continuous problem, it is not obvious how to change the basis function $d^2 \log d$ to handle irregular discontinuities. Our grid sizes are on the order of the image size, in pixels; we make sure that we have at least one grid element per pixel in the image. Finally, we use a coarse-to-fine multiresolution method to calculate our interpolants [Terz88].

We present the animator with curves, polylines, and points as deformation primitives. When solving the problem on a discrete grid we must scan convert the primitives' displacements onto the grid. In practice, we discretize the primitives into equidistant samples.

The animator specifies discontinuities in the surface by supplying an extra black-and-white matte; an image pixel is "connected" to neighboring pixels labeled "nonzero" in this matte. For most purposes, the animator uses the ordinary alpha matte in the role of discontinuity matte as well, but they may be specified separately. For instance, the eyes and mouth of the characters in Figure (1) are on a separate cel level, with holes specified in the top layer. These holes also specify discontinuities in the interpolant, so when the top eyelid closes, it does not affect the lower eyelid.

In applying the displacements we use a forward mapping technique, as opposed to the inverse mapping technique implemented by Beier and Neely (the former is a "many to one" mapping, the latter, "one to many"). All the warped images in the color plates illustrating this paper were generated using a forward mapping, including the pictures demonstrating Beier-Neely interpolation. The Beier interpolation picture actually uses 2 points for the interior points and four lines along the edges; the displacements for the points are weighted with Shepard's formula and the edges with the Beier-Neely modifications for lines. The checkerboards warped with thin-plate interpolation have displaced interior feature points and four stationary lines along the edges.

The interpolated displacements for the entire surface are applied to the image at each pixel. The image is rendered as a polygon mesh; each original pixel becomes a polygon vertex in the mesh, except where the discontinuity matte breaks the connections. Subpixel positioning of the displaced quadrilateral endpoints is important for good results.

Observations

Our experience suggests that the imposition of a structure to animate the image, such as a grid or mesh of polygons, can impose a heavy burden on the animator. It is far more intuitive to specify, and animate, a simple drawing which parametrizes and controls the image.

Polygonal texturing may not be smooth enough for the extreme deformations used in animation. Distance-weighted interpolants may not be smooth enough, either, and may limit the number of control primitives for practical purposes.

A very valuable feature of the thin-plate spline surface is its *idempotency*. New features can be added at any time without modifying the current mapping, and subsequently serve as handles for further animation. With a polygonal mapping, this can be ensured by subdividing only triangles in which new control points are introduced. With a distance weighted mapping, this property is impossible to achieve, and one must settle for gradually, over time, blending in the contribution of newly introduced control features.

The iterative relaxation used to compute the multigrid spline surface can profit from frame-to-frame coherence in animation. By using the last surface computed as an estimate for the next frame, the expense for the sequence is greatly reduced. The first frame of the example animated cat sequence took 5 min., 34 sec.; subsequent frames, on average, 3 min., 30 sec. (surface computed as a 513x513 grid on a MIPS 36Mhz R3000).

There are several ways to trade off computation and quality in the surface. One is to evaluate the surface on a coarse grid, and use tensor-product interpolation to upsample it. Another is to increase the error permitted when iteration is ceased, or to perform a fixed number of iterations. In this case, a modified form of Southwell iteration [Gera94] offers improved results for the same number of cycles. We implemented this option at the suggestion of Eric Chen, who was inspired by the "shooting" method of computing radiosity [Gortler93].

Acknowledgments

The authors would like to extend heartfelt thanks to Laurence Arcadias for the cartoon face animated in our illustrations, and to Subhana Ansari for layout.

References

[Beier92] Beier, T. and S. Neely. "Feature-Based Image Metamorphosis," Computer Graphics, Volume 26, Number 2, July 1992, pp. 35-42.

[Burtn76] Burtnyk, N. and M Wein. "Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation," CACM, Vol. 19, Number 10, October 1976.

[Farin90] Farin, G. Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide, Second Edition. Academic Press, NY, 1990.

[Forr72] Forrest, A. "On Coons and Other Methods for the Representation of Curved Surfaces," Computer Graphics and Image Processing, 1, 1972, pp. 341-369.

[Franke79] Franke, R. "A Critical Comparison of Some Methods for Interpolation of Scattered Data," Report NPS-53-79-03 of the Naval Postgraduate School, Monterey, CA. Obtained from U.S. Department of Commerce, National Technical Information Service.

[Gera94] Gerald, Curtis and P. Wheatley, Applied Numerical Analysis, Fifth Edition, Addison-Wesley Publishing Company, pp. 159-164.

[Gortler93] Gortler, Steven, Michael Cohen and Philipp Slusallek, "Radiosity and Relaxation Methods: Progressive Refinement is Southwell Relaxation," Princeton University Research Report CS-TR-408-93, February 1993.

[Gosh86] Goshtasby, A., "Piecewise Linear Mapping Functions for Image Registration," Pattern Recognition 19:6, 1986, pp. 459-466.

[Grim81] Grimson, W., From Images to Surfaces, MIT Press, 1981.

[Hard72] Harder, R., and R. Desmarais, "Interpolation Using Surface Splines," J. Aircraft, Vol. 9, February 1972, pp. 189-191.

[Litw91] Litwinowicz, P. "Inkwell: A $2^1/_2$-D Animation System," Computer Graphics, Volume 25, Number 4 , 1991, pp. 113-121.

[Patterson91] Patterson, E., P. Litwinowicz and N. Greene, "Facial Animation by Spatial Mapping," Computer Animation 1991, Springer-Verlag, NY 1991, pp. 31-44.

[Pilch74] Pilcher, David, "Smooth Parametric Surfaces," in Computer Aided Geometric Design, by Barnhill and Riesenfeld, Academic Press, NY, 1974, pp. 237-253.

[Rupr92] Ruprecht, D., and H. Müller, "Image Warping with Scattered Data Interpolation Methods," Research Report 443, Dortmund University, November 6, 1992.

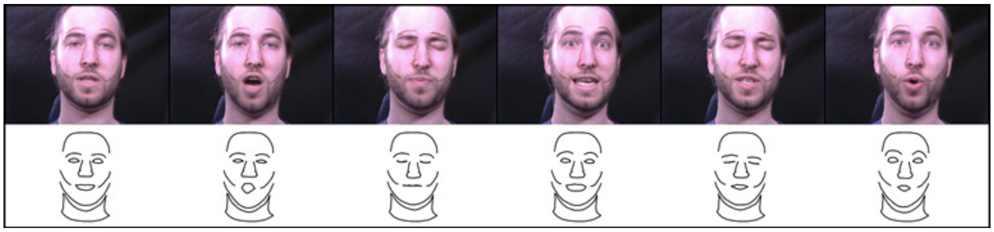
[Shep68] Shepard, D., "A Two-Dimensional Interpolation Function for Irregularly Spaced Data," Proceedings of the 23rd Nat. Conf. ACM, 1968, pp. 517-523.

[Sed86] Sederberg, T. and S. Parry. "Free-Form Deformation of Solid Geometric Models," *Computer Graphics,* Volume 20, Number 4, August 1986, pp. 151-160.

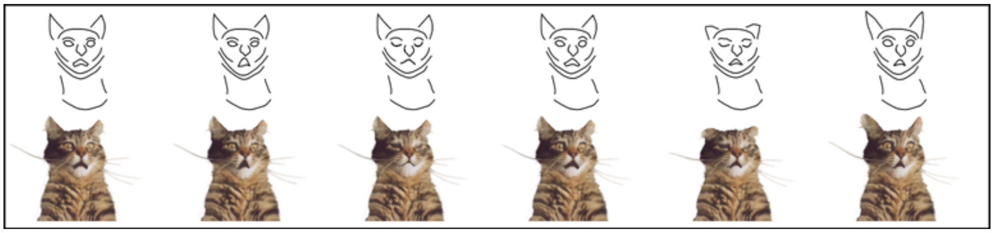
[Terz88] Terzopoulos, D. "The Computation of Visible-Surface Representations," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No.4, July 1988.

[Wolb89] Wolberg, George. "Skeleton Based Image Warping," The Visual Computer, Vol. 5, Number 1/2, March 1989, pp. 95-108.

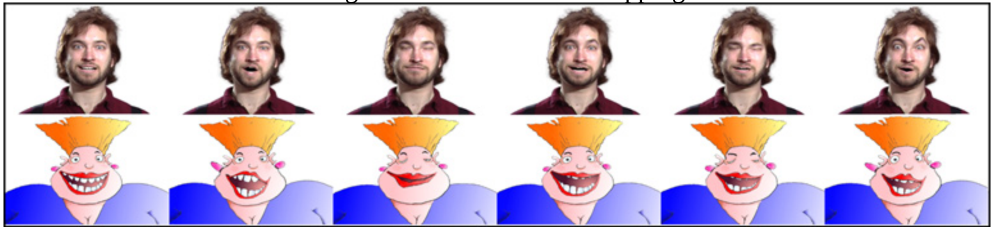
[vanO90] van Overveld, C. W. A. M., "A technique for motion specification in computer animation," The Visual Computer, Number 6, 1990, pp. 106-116.



Key features of the subject have been traced by hand as line drawings.



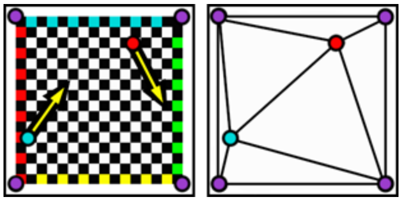
The leftmost drawings define an automatic mapping to the cat's face.



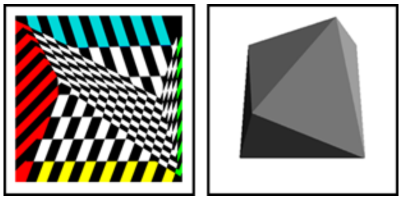
Thin-plate spline surfaces animate images from drawings.

Animation

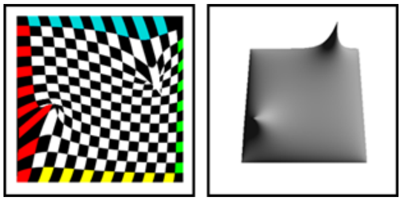
The top row of pictures shows sample frames from a video sequence. Key features of the subject have been rotoscoped to extract animated line drawings. Using cross-synthesis procedures described in [Patterson91], and the line drawings of the actor and cat in the leftmost column, the animation for the other line drawings has been automatically generated for the cat. Using these animated line drawings and the original cat photograph (shown in the leftmost column), thin-plate spline surfaces are used to compute each frame of the animation. The final two rows of frames, at left, show two more characters animated in this way, with the original faces shown in the leftmost column. The eyes and teeth are animated on a separate layer.



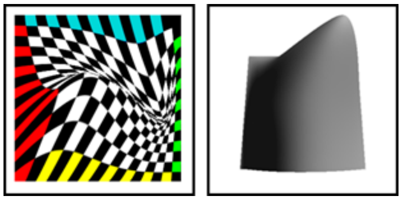
2 displaced points; triangulation.



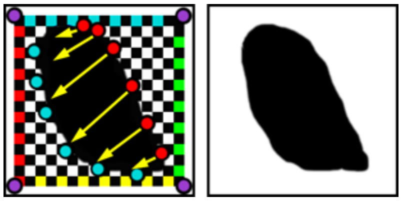
Polygonal mapping; ΔX surface.



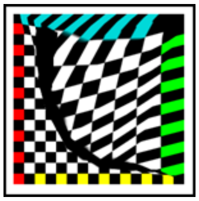
Beier interpolation; ΔX surface.



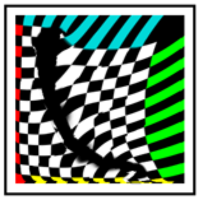
Thin-plate spline; ΔX surface.



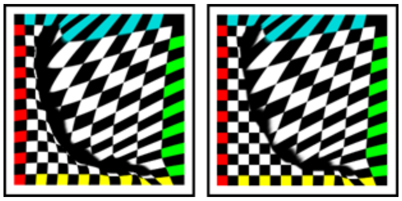
6 displaced points; discontinuity map.



Polygonal interpolation.



Beier interpolation.



without "hole" with "hole"

Thin-plate spline.

Holes and Cuts

The two columns of pictures on the far left show the relative smoothness of image warps based on various scattered data interpolation methods. At near left, we show the process of introducing deliberate "holes" or "cuts" in interpolating surfaces:

Top left test image (closing gap in checkerboard). The six red control points are moved to the tips of the yellow arrows. The blue points, as well as each edge of the square, are held in place. Top right, the map which controls the continuity of the surface.

Polygonal interpolation. Triangulation results in sharp bends within the checkerboard as it stretches. Interpolation is local, however, with no influence across the gap.

Beier interpolation. Folds and creases appear in the checkerboard, and the left edge of the hole folds under as the right edge approaches. The "hole" in this and the next example is simply a matte (transparency map) which makes part of the surface transparent.

Thin-plate spline, with no holes in the surface. As the right edge is stretched, the left edge folds under.

Thin-plate spline, with an actual "hole" in the interpolated displacement surface. As the right edge is stretched, the left edge is relatively unaffected. This type of control is necessary for the animation examples at the top of the page. When closing an eyelid, the animator doesn't want to affect the region below the eye. Continuity and translucency are specified with independent maps.