

Conformal Image Warping

Carl Frederick and Eric L. Schwartz

Courant Institute of Mathematical Sciences and NYU Medical Center

This report describes numerical and computer-graphic methods for conformal image mapping between two simply connected regions. The immediate motivation for this application is that the visual field is represented in the brain by mappings which are, at least approximately, conformal.¹⁻⁴ Thus, to simulate the imaging properties of the human visual system (and perhaps other sensory systems), conformal image mapping is a necessary technique.

There are two distinct aspects to this problem. First, one must implement a numerical or analytic method which allows for the computation of a given conformal mapping, constrained by the shape of the two simply connected regions (hereafter known simply as regions) to be mapped, and by a single point and orientation correspondence between them. Second, it is necessary to apply a space-variant texture-mapping algorithm to warp the image, once the mapping itself has been specified.

For generating the conformal map, we show a method for analytic mappings, and also an implementation of the Symm algorithm for numerical conformal mapping.⁵ The first method evaluates the inverse mapping function at each pixel of the range, with antialiasing via multiresolu-

tion texture prefiltering and bilinear interpolation.⁶ The second method is based on constructing a piecewise affine approximation of the mapping in the form of a joint triangulation, or triangulation map, in which only the nodes of the triangulation are conformally mapped. The texture is then mapped by a local affine transformation on each pixel of the range triangulation with the same antialiasing used in the first method.

We illustrate these algorithms with examples of conformal mappings constructed analytically from elementary mappings, such as the linear fractional map, the complex algorithm, etc. We also show applications of numerically generated maps between highly irregular regions, and also an example of the visual field mapping that motivates this work.

In addition to providing a necessary tool for simulation of cortical architectures, these illustrations may be of pedagogical use for students attempting to visualize the geometric properties of elementary conformal mappings; they may also find applications in such areas as fluid mechanics and electrostatics, where conformal mapping is a natural and basic tool.

Conformal mappings can be defined in a number of ways, each emphasizing different aspects of their geometric or analytic properties:⁷

- Complex analytic functions $f(z)$, for $\frac{df}{dz} \neq 0$ represent conformal mappings.
- A conformal mapping is locally isotropic (an infinitesimal area element is magnified equally in all directions).
- Infinitesimal angles are preserved by conformal mapping.

- The real and imaginary parts of the map function are harmonic conjugate functions; they satisfy the Laplace equation, $\nabla^2 f(x,y) = 0$ and intersect orthogonally. This property provides important practical applications to areas of potential theory (electrostatics, fluid mechanics, etc.) where the Laplace equation occurs.

The Riemann mapping theorem guarantees the existence and uniqueness of conformal mappings between regions. In a given region, there is a conformal

Riemann Mapping Theorem

A typical statement of the Riemann mapping theorem is in Ahlfors's *Complex Analysis*.⁷

Given any simply connected region Ω which is not the whole plane, and a point $z_0 \in \Omega$, there exists a unique analytic function $f(z)$ in Ω , normalized by the condition $f(z_0) = 0$, $f'(z_0) > 0$, such that $f(z)$ defines a one-to-one mapping of Ω onto the disk $|w| < 1$.

Our statement of this theorem is that uniqueness is specified by a point correspondence and an orientation. This is equivalent to the above statement: The point correspondence specifies the mapping of the point that maps into the origin of the unit circle $f(z_0) = 0$. The statement that $f'(z_0) > 0$ is equivalent to fixing the orientation of the unit disk, since it specifies that only a (positive) scaling and no rotation occur at this point. Other orientations of the mapping can be generated by multiplication of $f(z)$ by $e^{i\theta}$.

mapping of that region onto the unit disk⁷ (circle of radius one). This mapping is made unique by fixing a single point in the region onto the center of the unit disk, and fixing the orientation of the unit disk. (See the sidebar for the Riemann mapping theorem.)

The problem of conformal mapping of textures (images) has two distinct parts: The mapping function itself must be provided, and since the scaling induced by the mapping can change continuously, texture mapping must be space variant. We will describe two different methods for implementing the conformal mapping (one analytic and one numeric) and two different methods of texture mapping (one pixel-based and one polygon-based).

Elementary analytic maps

This method performs texture mapping by evaluating a given elementary analytic map on a pixel-by-pixel basis.

When the inverse or a desired branch of inverse mapping can be described analytically, a direct method is possible. A naive approach is to evaluate $f(z)$ at each point, copying the texture value at z to its image position in the w -plane. This method leads to significant aliasing in the w -plane. Instead we use the following method: We consider a small region of the w -plane; the inverse mapping $f^{-1}(w)$ is determined and $f^{-1}(w)$ is evaluated at each point within this region. The texture value at $z = f^{-1}(w)$ is retrieved and used as the value at w . Effectively, we find all the points in the z -plane that map to our piece of the w -plane by $f(z)$. We still get aliasing with this method, but it can be handled easily by one of the standard

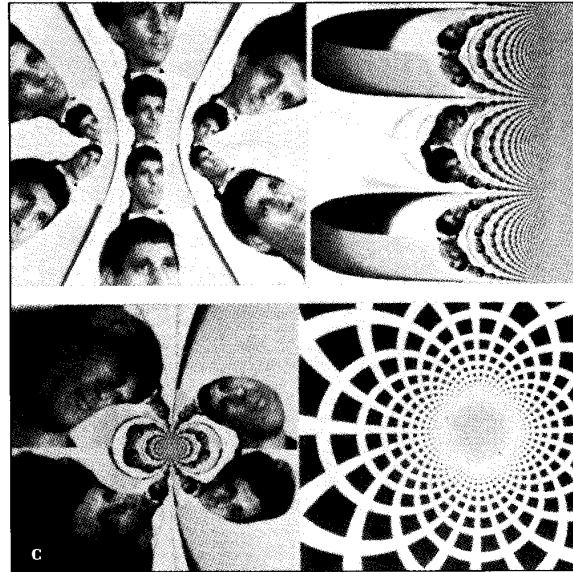


Figure 1. (a) Complex cosine. The original texture (which has been repeated over the plane) is depicted in Figure 2. (b) Complex logarithm. Horizontal lines are images of lines radiating from the origin in the domain region. This image shows one complete “period strip” plus pieces of two others. Horizontal lines start to curve on the left due to a floating-point precision error. (c) A fractional linear map

$$w = \frac{z - 1}{z + 1}$$

illustrated by the face image and also by a repeating rectangular lattice texture. The effect of this map on the plane can be imagined as poking a hole in the plane at the point $(-1, 0)$, turning it inside out and flipping it over.

methods for prefiltering.^{6,8} See Figure 1 for examples of familiar elementary functions in complex variables. Figure 2 shows an example of mapping generated by composition from fractional linear maps and elementary functions.

Numerical conformal mapping

Next we describe a method for constructing numeric conformal mappings from one finite region to another. This is applicable when only the shapes of the two regions to be mapped (and a point and orientation correspondence) are known.

In this approach one of the regions of interest is triangulated (e.g., using a Voronoi method⁹). The second region can then be triangulated by mapping the nodes of the initial triangulation to the second region and then using the initial connectivity matrix to trian-

Symm Algorithm for Conformal Mapping

Symm has described an integral equation method for computing the conformal mapping of a given, simply connected domain onto the interior of the unit circle. This method performs well for a domain described by a large number of (polygonal) vertices. It is based on the observation that a solution for conformally mapping a given, simply connected domain D with boundary L , in the z plane, onto the unit disk $|w| \leq 1$, in the w plane, in such a way that a point $z_0 \in D$ goes into the center $w = 0$, is provided by (up to arbitrary rotation)

$$w(z) = \exp[\log(z - z_0) + \gamma(z, z_0)]$$

where $\gamma = g + ih$, and g satisfies the boundary value problem (h is conjugate to g)

$$V^2 g = 0 \text{ for } z \in D,$$

$$g = -\log|z - z_0| \text{ for } z \in L$$

This identity can be understood by observing that

- $w(z)_0 = 0$

- On the boundary ($z \in L$), $w(z) = e^{i[(\arg(z - z_0)) + h]}$, i.e., $|w(z)| = 1$, so the boundary L is mapped to the boundary of the unit circle.

- For interior points $z \in D$, we have $w(z) \leq 1$, since $|w(z)| = 1$ on the boundary, and $w(z)$ must take its maximum in this region on the boundary, by the maximum principle.⁷

Thus, Symm replaces the problem of finding $w(z)$ with the problem of finding an analytic function $g(z) = \log(w) - \log(z - z_0)$. Essentially, the problem is transformed from the original domain to a complex logarithmic representation of it. Symm then goes on to outline methods of finding γ by creating a set of Fredholm integral equations, which are numerically solved by standard methods. One difficulty in implementing this work, not emphasized in it, is that line integrals of the argument of an analytic function are evaluated in this algorithm. Direct (i.e., naive) numerical implementation of the equations found in this paper is not correct. Careful attention must be paid to ensuring that a continuous branch of the argument is used when performing a line integral of the argument of a complex function.

gulate the second region. (The connectivity matrix can be defined as a binary valued Cartesian product on the set of points in the region. If two points are connected by an edge, the matrix entry is 1, otherwise, 0. In this way the topology of the triangulation map is specified.)

It is important to note that this joint triangulation is potentially problematic. There is no guarantee that

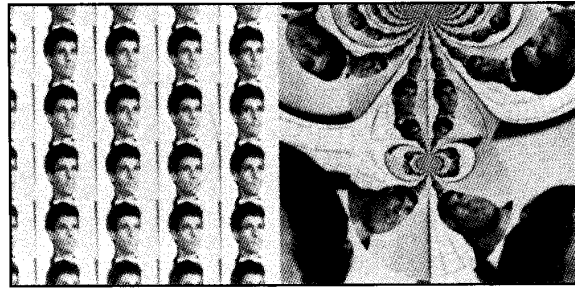


Figure 2. This pair illustrates the conformal map that takes the unit disk to the upper half disk. The superimposed circle and half circle isolate the regions in the planes that make this a useful transformation. Its equation is

$$w = - \frac{\left(\frac{1+z}{1-z} i \right)^{\frac{1}{2}} - 1}{\left(\frac{1+z}{1-z} i \right)^{\frac{1}{2}} + 1}$$

the connectivity matrix of a triangulation still describes a triangulation when its nodes have been mapped to a new region. Some of its edges might then intersect, violating the definition of triangulation. For a given mesh, a joint triangulation can always be generated by adding pairs of points to the original point mapping, as proved by Saalfeld.¹⁰ Saalfeld proves that the joint triangulation can be generated, but this method requires exponential refinements (subdivisions) of its triangles. In our experience, no special means of generating valid triangle maps has been necessary, because any reasonably well-chosen mesh size leads to a valid triangle map.

We begin with the mapping of an arbitrary region to the unit disk. Next we discuss mapping between two arbitrarily shaped regions.

Mapping an arbitrary region to the unit disk

Henrici has recently surveyed existing numeric conformal mapping algorithms.¹¹ For arbitrarily shaped regions (which are approximated by polygonal boundaries with large numbers of vertices), Symm's algorithm is preferable. (See the sidebar on Symm's algorithm.) The Symm algorithm is initialized with a description of the boundary of the region to be mapped and the point in this region that is the pre-image of the center of the unit disk. It returns the

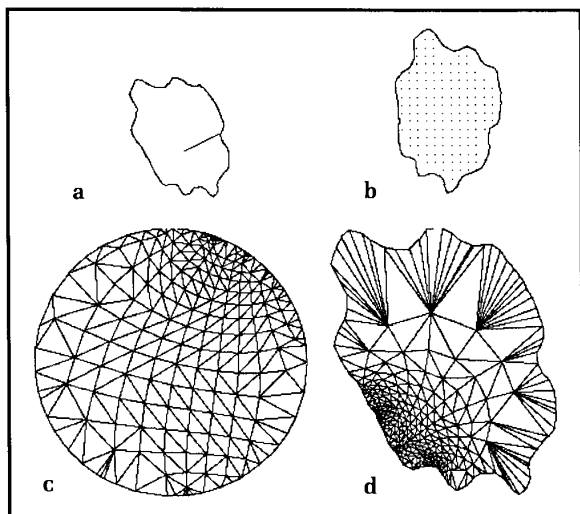


Figure 3. (a) A region is represented by a simple polygon. The vector pointing inward from the boundary indicates the unique conditions imposed on the conformal mapping. Its interior end denotes the point that will be mapped to the origin of the unit disk. Its other end determines the rotation of the unit disk by specifying which point on the boundary will be mapped to the coordinates (1,0). (b) The interior is partially filled with points on a rectangular grid using a floating-point scan-conversion algorithm. (c) The boundary and interior points are mapped to the unit disk using Symm's algorithm. This set of points is then triangulated. (d) The same triangulation connectivity information is used on the domain points to generate a domain triangulation.

mapping of the boundary and interior region to the unit disk. One drawback of this method is that it provides the mapping of the desired region into the unit disk, but it does not provide the inverse mapping.

To implement our algorithm, first, a point approximation of the mapping is constructed by applying Symm's interior mapping to a sampling of the region at rectangular grid locations. The density of this grid is kept low for economy of computation, and in many cases this undersampling gives satisfactory results. The sampling is generated by a floating-point scan-conversion algorithm operating as a polygon fill. The boundary points defining the region are mapped using Symm's boundary correspondence function.

Second, this set of image points (interior and boundary) in the unit disk is triangulated using the Delaunay triangulation,¹² and the same topology is applied to generate a triangulation of the points in the original region (see Figure 3). Finally, triangle-by-triangle texture mapping¹³ is used to deform the texture in the

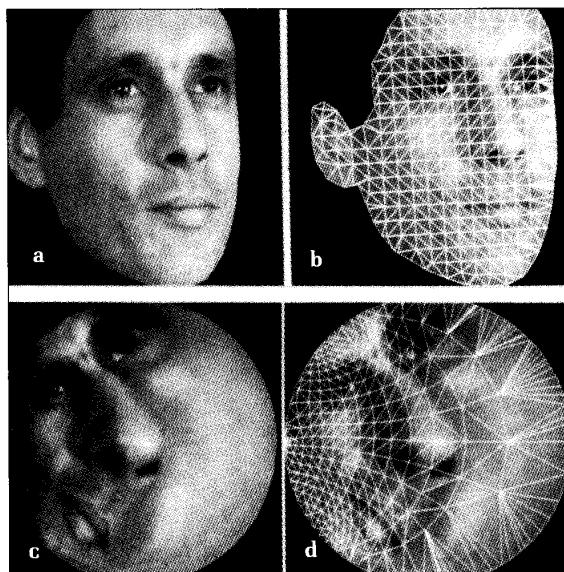


Figure 4. Mapping an arbitrary region onto the unit disk using the same method as in Figure 3. In this case the orientation vector is a horizontal line segment beginning at the highlight on the nose and ending on the region boundary to the right. (a) The original texture. (b) The original texture overlaid with a triangulation graph. (c) The face texture mapped onto the unit disk. (d) The unit disk overlaid with a triangulation graph.

region to its unit-disk counterpart. This is effectively a piecewise affine approximation of the conformal map (see Figure 4).

Mapping one region to another region

To create a mapping from one region to another, we start by generating the mapping from each region onto the unit disk, either analytically or using the Symm algorithm. We then triangulate the range set from the second-region mapping. Optimized point location¹² is then used for each point in the range set of the first-region mapping to determine which triangle from the range set of the second mapping contains that point. The affine mapping determined by this triangle and its counterpart in the domain of the second region mapping is used to map the included point from the first-region range set to the second region. A cleaner solution would make use of a general method capable of mapping the unit disk to an arbitrary region, thus avoiding the numerical inversion of the

Algorithm for Region-to-Region Mapping via the Unit Disk

P_x : Region boundary-point set and pre-image of unit disk origin
 S_x : Point set; either the sampled region or its disk mapping
 T_x : Triangulation of S_x
 A_x : Set of affine maps
 $(S1_{region1}, S1_{disk}) \leftarrow \text{Symm}(P1)$
 $(S2_{region2}, S2_{disk}) \leftarrow \text{Symm}(P2)$
 $T2_{disk} \leftarrow \text{Delaunay_triangulate}(S2_{disk})$
 $T2_{region2} \leftarrow \text{connect } S2_{region2}$
 with same topology as $T2_{disk}$
 $A2 \leftarrow \text{generate_affine_maps_from}(T2_{disk}, T2_{region2})$
 For each point $S1_{disk,i}$ in $S1_{disk}$
 $j \leftarrow \text{locate } S1_{disk,i}$ among $T2_{disk}$
 $S1_{region2,i} \leftarrow A2_j(S1_{disk,i})$
 $T1_{disk} \leftarrow \text{Delaunay_triangulate}(S1_{disk})$
 $T1_{region1} \leftarrow \text{connect } S1_{region1}$ with same topology as $T1_{disk}$
 $T1_{region2} \leftarrow \text{connect } S1_{region2}$ with same topology as $T1_{disk}$
 The joint triangulation is $(T1_{region1}, T1_{region2})$

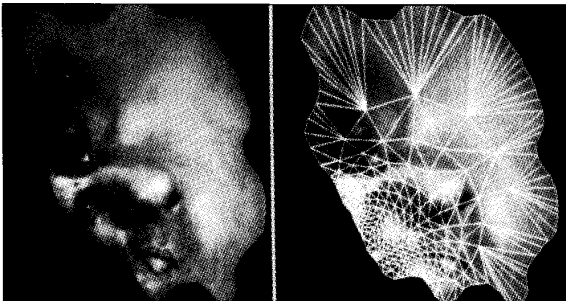


Figure 5. This is the texture mapping of the region (specified in Figure 3) via the unit disk, as explained in the text. Note that the triangulation is the result of the piecewise affine approximation to the inverse of the conformal mapping that transformed the region in Figure 3 to the unit disk. The highlight on the nose was chosen as the point to satisfy the first uniqueness condition of the Riemann mapping theorem. The orientation of the nose can be seen to follow the orientation vector as it was transformed from the horizontal segment of Figure 4 to its image in Figure 3a.

Algorithm for Region-to-Region Mapping via the Unit Disk with Boundaries

$(S1_{region1}, S1_{disk}) \leftarrow \text{Symm}(P1)$
 $(S2_{region2}, S2_{disk}) \leftarrow \text{Symm}(P2)$
 $T2_{disk} \leftarrow \text{Delaunay_triangulate}(S2_{disk})$
 $T2_{region2} \leftarrow \text{connect } S2_{region2}$
 with same topology as $T2_{disk}$
 $A2 \leftarrow \text{generate_affine_maps_from}(T2_{disk}, T2_{region2})$
 $A3 \leftarrow \text{generate_boundary_affine_maps_from}(S2_{disk}, S2_{region2})$
 $A4 \leftarrow \text{generate_boundary_affine_maps_from}(S1_{disk}, S1_{region1})$
 For each point $S1_{disk,i}$ in the interior of $S1_{disk}$
 $j \leftarrow \text{locate } S1_{disk,i}$ among $T2_{region2}$
 $S1_{region2,i} \leftarrow A2_j(S1_{disk,i})$
 For each point $S1_{disk,i}$ on the boundary of $S2_{disk}$
 $j \leftarrow \text{locate } S1_{disk,i}$ among boundary points of $S2_{disk}$
 $S1_{region2,i} \leftarrow A3_j(S1_{disk,i})$
 For each point $S2_{disk,i}$ on the boundary of $S2_{disk}$
 $j \leftarrow \text{locate } S2_{disk,i}$ among boundary points of $S1_{disk}$
 $S2_{boundary1,i} \leftarrow A4_j(S2_{disk,i})$
 $S1_{region1} \leftarrow \{S1_{region1}, S2_{boundary1}\}$
 $S1_{disk} \leftarrow \{S1_{disk}, \text{boundary points of } S2_{disk}\}$
 $S1_{region2} \leftarrow \{S1_{region2}, \text{boundary points of } S2_{region2}\}$
 $T1_{disk} \leftarrow \text{Delaunay_triangulate}(S1_{disk})$
 $T1_{region1} \leftarrow \text{connect } S1_{region1}$ with same topology as $T1_{disk}$
 $T1_{region2} \leftarrow \text{connect } S1_{region2}$ with same topology as $T1_{disk}$
 The joint triangulation is $(T1_{region1}, T1_{region2})$

Symm mapping. Some ideas on how this might be done are outlined by Henrici.¹¹

The algorithm for region-to-region mapping via the unit disk (see the sidebars on this) does not explicitly mention boundary points to simplify its presentation. To make the boundaries appear properly, we do additional processing. Since this is a discrete approximation, points on the boundary of the first region are not likely to map to the points that define the boundary of

the second region (unless the first boundary is very dense with points).

The effect of mapping the boundary points from the first to the second region then appears to change the shape of the second region, since the boundary points are mapped nonuniformly around the second region's boundary. To maintain the shape of the second boundary, its points are added to the points from the first region mapping. Thus, each boundary point from the disk-mapping of the second region is located between two of the boundary points from the disk-mapping of the first region.

Using the affine transformation, which maps the boundary edge in the first-region disk-mapping back to the first-region boundary edge, the point from the second region is mapped back to the boundary of the first region. All second-region boundary points mapped to the first region in this way are then added to the original set from the first region. We call this augmented set of boundary points the refinement of the boundary. It is used as the new boundary of the mapping. To make the first region's boundary points map to the second region's boundary, the same method is used. This assures that the boundaries of both regions appear correctly (see Figures 4 and 5).

Application to visual cortex

In the case of visual cortex, there is considerable experimental evidence that the mapping of the retina to the surface of primary visual cortex is approximately isotropic. Thus, to model the representation of a visual image on the surface of the cortex, we need to construct a conformal approximation to this map, and to perform a conformal texture map of given visual field images. To illustrate this process, we show a numerical flattening of the surface of primary visual cortex of the monkey, which we have performed recently.¹⁴ In this work we were able to identify a single point (the representation of the blind spot, or optic disk) in the eye, and an orientation (the orientation of the horizontal meridian). These observations together with the flattened representation (and its boundary) were sufficient to generate the cortical map function.⁴ The agreement of this method of determining the cortical map—and direct microelectrode measurements of the cortical map function—is excellent. Figure 6 shows a natural scene, mapped via this conformal approximation. The details of its construction are outlined in the sidebar on conformal mapping of the retinal image to a section of the brain.

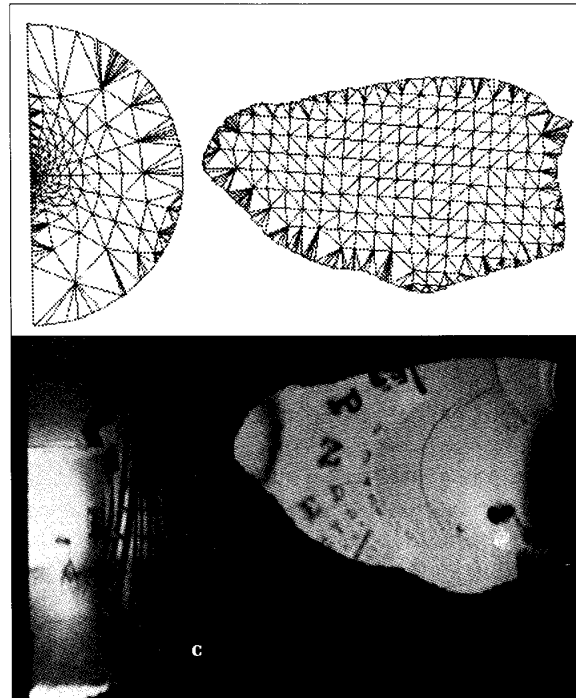


Figure 6. (a) The half-retina, flattened to a half-disk. (b) 6a is mapped to the flattened visual cortex. The fovea is at the midpoint of the vertical edge in the boundary of the retinal image. (c) In cortex space the area around the fovea is extremely enlarged due to the space-variant nature of the mapping. We used a special technique to achieve high-resolution texture near the fovea, which cannot be represented at full resolution in a small picture. Thus, the area near the fovea in the retinal picture lacks detail.

Comparison with earlier work

An algorithm for the Schwarz-Christoffel method of generating conformal maps of polygonal regions has been developed by Trefethen.¹⁵ Using Trefethen's algorithm as a basis, Fiume et al. discuss interpolation of mesh approximations to such maps and the critical issue of filtering.¹⁶ However, this approach is not general, since the conformal mapping algorithm it uses is adequate only for polygonal approximations to regions that have no more than perhaps 10 sides. For many actual applications of conformal mapping in computer-graphic contexts, it is necessary to handle arbitrary domains, approximated by polygonal boundaries with large numbers of nodes. The method

we present here is adequate for the general case, as suggested by some of the difficult domain shapes illustrated.

Of the four methods described by Fiume for interpolation of a point-sampling of the mapping, the best is bilinear interpolation. Our approach is to use 2D linear interpolation over triangles as a first-order approximation to the actual mapping. The geometric interpretation of linear interpolation can be imagined as finding the interpolated points as points on a triangulated polyhedral surface for each coordinate.

This method is also known as barycentric coordinate interpolation. It is a fundamental building block of simple, stable interpolation within a computer-graphics polygon-rendering context, and so of texture-mapping polygons as well.

Bilinear interpolation, geometrically interpreted, amounts to finding the interpolated points as points on a ruled surface where each surface patch of the ruled surface is defined by four mesh points. This ruled surface is smoother than the polyhedral surface, but the difference between them is insignificant. For our purposes, bilinear or other higher order interpolation methods appear to be unnecessary; and they thwart the advantages of rendering in the comfortable world of triangles.

Once the conformal mapping has been approximated by a piecewise affine map, we use a standard pyramidal antialiasing technique⁶ to filter the textured triangles. More sophisticated filtering techniques could be used, as in recent work¹⁷ involving space-variant kernels. ■

Acknowledgments

We thank Nick Trefethen for helpful discussions on this subject and Adi Perry and Daphna Weinshall for contributing to early stages of the work. The research described here was supported by grant AFOSR 88-0275, the System Development Foundation, and the Nathan S. Kline Psychiatric Research Center. Use of Dr. Yehezkel Yeshurun's face is gratefully acknowledged.

References

1. D.M. Daniel and D. Whitteridge, "The Representation of the Visual Field on the Cerebral Cortex," *J. Physiology*, Vol. 159, 1961, pp. 203-221.
2. E.L. Schwartz, "On the Mathematical Structure of the Retinotopic Mapping of Primate Striate Cortex," *Science*, Vol. 227, 1985, p. 1066.

Conformal Mapping of the Retinal Image to a Section of the Brain

The problem of conformally mapping the retina to V1, e.g., can be regarded as a problem of mapping the half-unit sphere (an approximation to the retina) conformally to a flattened model of the surface of visual cortex. For flattening of the cortex, we use an algorithm—developed in our lab—which achieves an average error of roughly 5 percent in mapping a 3D representation of visual cortex into 2D. The mapping problem is solved in our case by decomposing it into three steps:

1. Mapping the quarter unit sphere (one half of a retina) conformally to the half unit disk.
2. Mapping the half-unit disk conformally to the unit disk, where the point in the half disk mapped to the origin of the target disk is a parameter of this mapping, to be chosen dependent on the specific data.
3. Mapping the unit disk conformally to an arbitrary 2D domain, the flattened brain data.

Step 1:

We use the following stereographic projection (which provides a conformal mapping of the quarter-unit sphere to the half-unit disk):

$$(\theta, \phi) \rightarrow \left(\frac{\cos\phi \sin\theta}{1 - \cos\phi \cos\theta}, \frac{\sin\phi}{1 - \cos\phi \cos\theta} \right)$$

where θ and ϕ are the corresponding eccentricity and azimuth of the unit sphere, and the output represents 2D Cartesian coordinates.

Step 2:

The following transformation T is a conformal complex transformation which maps the half-unit disk to the unit disk, where a point z_0 internal to the half-unit disk is mapped to the origin of the target unit disk, with a rotation of α radians: $T = H \cdot G \cdot F$, where

$$F = \frac{z - i}{z + i}, \quad G = z^2, \quad \text{and} \quad H = e^{i\alpha} \frac{z - z_0}{z - \bar{z}_0}$$

Step 3:

We use Symm's algorithm to map a given, simply connected domain conformally onto the interior of the unit circle. The results of this approximation have been checked against direct microelectrode measurement of the map function of visual cortex,⁴ with excellent agreement.

3. D.C. Van Essen, W.T. Newsome, and J.H.R. Maunsell, "The Visual Representation in Striate Cortex of the Macaque Monkey: Assymetries, Anisotropies, and Individual Variability," *Vision Research*, Vol. 24, 1984, pp. 429-448.
4. D. Weinshall and E.L. Schwartz, "A New Method for Measuring the Visuotopic Map Function of Striate Cortex: Validation with Macaque Data and Possible Extension to Measurement of the Human Map," *Neuroscience Abstracts*, Wash., D.C., 1987, Vol. 13, p. 1291.

5. G.T. Symm, "An Integral Equation Method in Conformal Mapping," *Numerische Mathematik*, Vol. 9, 1966, pp. 250-258.
6. L. Williams, "Pyramidal Parametrics," *Computer Graphics* (Proc. SIGGRAPH), Vol. 17, No. 3, July 1983, pp. 1-30.
7. L. Ahlfors, *Complex Analysis*, McGraw-Hill, Highstown, N.J., 1966.
8. F. Crow, "Summed-Area Tables for Texture Mapping," *Computer Graphics* (Proc. SIGGRAPH), Vol. 18, No. 3, July 1984, pp. 207-212.
9. S. Fortune, "A Sweepline Algorithm for Voronoi Diagrams," *ACM Symp. Computational Geometry*, ACM, New York, 1987, pp. 1-16.
10. A. Saalfeld, "Joint Triangulations and Triangulation Maps," *ACM Symp. Computational Geometry*, ACM, New York, 1987, pp. 195-204.
11. P. Henrici, *Applied and Computational Complex Analysis*, Vol. 3, John Wiley, New York, 1986, pp. 402-413.
12. F.P. Preparata and M.I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.
13. J.F. Blinn, and M.E. Newell, "Texture and Reflection in Computer Generated Images," *CACM*, Vol. 19, No. 10, Oct. 1976.
14. E.L. Schwartz et al., "Computational Neuroscience: Applications of Computer Graphics and Image Processing to 2D and 3D Modeling of the Functional Architecture of Visual Cortex," *CG&A*, Vol. 8, No. 4, July 1988, pp. 13-23.
15. L. Trefethen, "Numerical Computation of the Schwarz-Christoffel Transformation," *SIAM J. Scientific and Statistical Computing*, Vol. 1, 1980, pp. 82-102.
16. E. Fiume, A. Fournier, and V. Canale, "Conformal Texture Mapping," *Proc. Eurographics*, North-Holland, Amsterdam, 1987, pp. 53-64.
17. A. Fournier and E. Fiume, "Constant-Time Filtering with Space-Variant Kernels," *Computer Graphics* (Proc. SIGGRAPH), Vol. 22, No. 4, Aug. 1988, pp. 229-238.



Carl Frederick works with NYU Medical Center's Computational Neuroscience Labs, where he has developed algorithms to model the primate brain digitally. He has also implemented mathematical and computational models used in simulating primate vision structures. He has worked within the computer graphics industry in the analysis, design, and implementation of systems at Digital Effects and Omnibus.

Frederick has a BS in mathematics and philosophy from the University of Wisconsin, Superior, and an MS in mathematics from the Courant Institute of Mathematical Sciences, NYU.



Eric L. Schwartz is associate professor at the Computational Neuroscience Laboratories of New York University Medical Center. He is also a research scientist at Nathan Kline Research Institute and an adjunct associate professor of computer science at the Courant Institute of Mathematical Sciences of NYU. His current research interests are anatomy and physiology of primate visual systems, computational vision and robotics, and computer and mathematical simulation of the nervous system. A major theme of this research is the application of mathematical and computer models to describe the various architectures that have been experimentally observed in primate visual systems, and to infer the possible computational significance of these architectures for both machine and biological visual computation.

Schwartz received his AB in physics and chemistry in 1967 and his MA, MPhil, and PhD in high-energy physics from Columbia University in 1973.

Frederick and Schwartz can be contacted at the Computational Neuroscience Laboratory, New York University School of Medicine, Department of Psychiatry, 550 Fifth Avenue, New York, NY 10016.