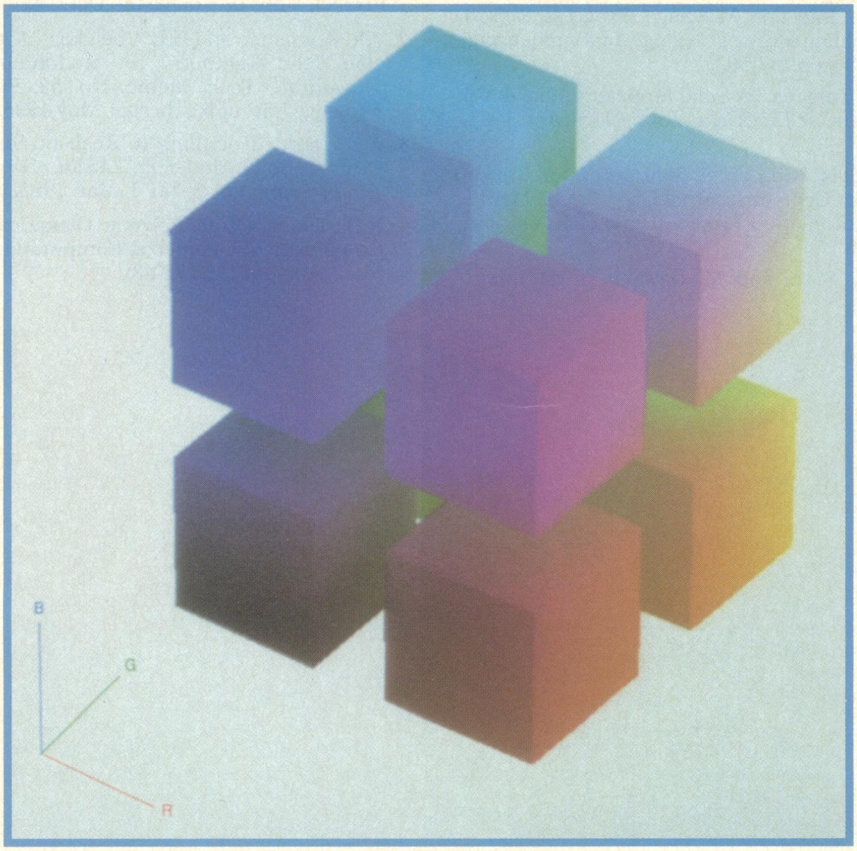
Two-Part Texture Mappings

Eric A. Bier
University of California, Berkeley

Kenneth R. Sloan, Jr.
University of Washington



Most published techniques for mapping two-dimensional texture patterns onto three-dimensional curved surfaces assume that either the texture pattern has been predistorted to compensate for the distortion of the mapping or the curved surfaces are represented parametrically. We address the problem of mapping undistorted planar textures onto arbitrarily represented surfaces. Our mapping technique is done in two parts. First the texture pattern is embedded in 3-space on an intermediate surface. Then the pattern is projected onto the target surface in a way that depends only on the geometry of the target object (not on its parameterization). Both steps have relatively low distortion, so the original texture need not be predistorted. We also discuss interactive techniques that make two-part mapping practical.

Systems designed to perform realistic image synthesis use synthetic texturing to reproduce a surface with varied coloring (e.g., a painted vase) or to approximate the appearance of surface micro-structure (e.g., the pits in the surface of an orange). In the first case the reflection coefficients are varied according to a texture pattern.[1,2,3] In the second case the surface normals are varied according to a

0272-1716/86/0900-0040$01.00 © 1986 IEEE

texture pattern.[4] A very general view of this process is that the system is computing a color function $\vec{C}(x,y,z)$, for $[x,y,z]$ on the surface of the object, or a normal perturbation function $\vec{N}(x,y,z)$ for the same $[x,y,z]$. We address the problem of computing these functions for a general class of surfaces. For brevity we will consider only $\vec{C}(x,y,z)$ henceforth. The results extend naturally to $\vec{N}(x,y,z)$.

One possible choice for $C(x,y,z)$ is a simple function of the variables x, y, and z. For instance, if we are trying to draw a picture of the red-green-blue color cube, we might use a very simple function defined on a 3D volume: $\vec{C}(x,y,z) =$ [red, green, blue] $(x,y,z) = [x/s, y/s, z/s]$ where s is the side of the cube, as shown in Figure 1 (left). Recent articles by Peachey[5] and Perlin[6] use "solid texturing" to achieve wonderful effects when the source of texturing really is three-dimensional (e.g., wood grain). However, the source of the texture pattern is often not a three-dimensional function but rather a two-dimensional image, such as a photograph or sketch, defined on a plane $[x_a, y_a]$ (where a stands for artwork). In this case we have the problem of finding a mapping from the artwork plane to the object surface. In other words, we must find a mapping M: $[x_a, y_a] \rightarrow [x,y,z]$, for those $[x,y,z]$ on the object surface. The nature of this mapping will depend on the desired effect.

Here we discuss mappings where the original artwork does not need to be predistorted to compensate for the mapping, and where the effect of the mapping is independent of the representation of the object surface. Such mappings must have low distortion for a useful class of target object, and must depend only on the geometry of these objects. The generality we seek is not without cost. As we will see, the low distortion of the mapping must often be paid for with the effort required to steer around its discontinuities.

We propose that a mapping M from the texture plane $[x_a, y_a]$ to the object surface $[x,y,z]$ be done in two parts. The first part, S: $[x_a, y_a] \rightarrow [x_s, y_s, z_s]$, maps each texture point $[x_a, y_a]$ to a point on a simple intermediate surface in 3-space. The second part, O: $[x_s, y_s, z_s] \rightarrow [x,y,z]$ maps each point of the intermediate surface to a point on the object surface. The mapping name S is a mnemonic for "mapping to intermediate *S*urface"; O is a mnemonic for "mapping to *O*bject."

Why should the mapping be done in two parts? Placing a two-dimensional texture on a three-dimensional surface involves both embedding the texture plane in 3-space and fitting the embedded surface to a particular object. By analogy, we can think of ourselves as being in the one-size-fits-all clothing business. Our first task is to make the garment from planar fabric. Our second task is to stretch it onto a particular customer.

Unlike the garment manufacturer, we have no idea what shape our customers will take. Hence, we must make very loose-fitting general-purpose garments indeed. These "garments" are the intermediate surfaces of our mapping M.

We share another problem with the manufacturer of garments: We must decide where to cut the fabric to include our favorite parts of the fabric pattern in the garment (and to assure that the pattern will meet itself nicely at the seams). Fortunately, we can use interactive graphics to help us cut up texture patterns in a pleasing manner. Ideally, we would like to be able to see how changes made in the initial cutting of the fabric will show up when the garment is made, or even when the garment is on a particular customer. In other words, as we change the mapping M, we would like to see the results of our changes on the intermediate surface and on the target object.

This article describes our technique, which has been implemented as part of the Solidviews 3D illustration system at Xerox PARC.[7] (Solidviews should not be confused with Lexidata's SOLIDVIEW.) We will begin by reviewing previous work in texture mapping of predistorted and undistorted textures. Then we will look at low-distortion mappings that embed the planar artwork in 3-space on a convex intermediate surface (a plane, cylinder, box, or sphere). Next we will describe some general techniques for mapping a texture from a convex surface to an arbitrary surface, and analyze some properties of the techniques. On the basis of this analysis, we will choose five promising mappings and give simplified versions of the equations used to implement them. The distortion characteristics of these mappings will then be analyzed. We will discover a trade-off between distortion and discontinuities, and suggest that interactive techniques be used to make discontinuities less noticeable, making our low-distortion mappings practical. Finally, we will discuss these interactive techniques.

Previous work

Early work on texture mapping involved predistorting textures or mapping them onto special types of surfaces. More recent work involves mapping undistorted textures onto arbitrary surfaces.

Mapping predistorted textures

Blinn and Newell[1] use a texture mapping technique to give the effect of environmental reflections in curved surfaces (e.g., the reflection of a paned window in a teapot). The key features of the environment are mapped onto a sphere of directions before rendering. At each surface point during rendering, the direction vector of reflected light is used as an index into the sphere of directions to determine the reflected color at that point. This environmental mapping technique is a quick way to give the appearance of environmental reflections without requiring that a real environment be modeled.

One difficulty with environmental mapping is that the artist must map environmental features onto a sphere. Since most paint programs draw only on the plane, a mapping from plane to sphere must be used. The most popular mapping (the spherical coordinate mapping $[x_a,y_a] \rightarrow [\theta,\phi]$; $0<\phi<2\pi$, $0<\phi<\pi$) greatly distorts the planar image. Hence, it is necessary for the artist to predistort the environmental features to get the desired effect.

We address the problem of mapping undistorted images onto object surfaces. The environmental mapping of Blinn and Newell is not a solution to this problem. However, the environmental mapping does use the idea of an intermediate surface, in this case a sphere, which is used to divide the mapping M into two simple parts. We will develop this idea later.

We are interested in undistorted images because they are easy to create or obtain. It is easy to scan in a photograph (e.g., of tree bark), scan in a pen-and-ink drawing, use a paint program to make a sampled image, or use a draw program to make a synthetic image. Images that are predistorted to look good after a mapping such as the spherical coordinate mapping are rare.

Mapping undistorted textures onto special surfaces

Techniques already exist to map undistorted images onto parametric surfaces and polygons. Catmull[3] and Blinn[2] have used a simple scheme to map images onto parametric surfaces. The texture definition space $[x_a,y_a]$ is mapped with scaling to the parameter space $[u,v]$ which is used to generate the parametric surface $\vec{Q}(u,v)=[X(u,v), Y(u,v), Z(u,v)]$. The distortion of this mapping can be high if the parameterization isn't chosen carefully. Feibush[8] maps textures onto polygons in a simple manner, which takes advantage of the planarity of polygons; the texture plane is the same as the plane of the polygon, and the texture can be rotated, scaled, and translated as desired in that plane to suit the artist.

These techniques rely on special properties of the object surfaces (e.g., planarity) or of their representation (e.g., parametric), yet they may still have high distortion. We describe methods that will work on arbitrary surfaces with arbitrary representations, usually with low distortion.

Mapping undistorted textures onto arbitrary surfaces

Barr[9] has already done some work in this area. Although his superquadric shapes can be represented parametrically, he has discovered that his deformation operations twist the constant-parameter lines of his shapes extremely, and that he does not always want his texture patterns to be deformed in the same fashion. Barr uses a two-part mapping, which he calls "decals." The texture pattern is mapped first onto a quadric surface in 3-space. The color of a particular surface point is determined from the position and normal direction of the surface at that point, by finding where a line through that surface point and parallel to the surface normal hits the textured quadric in 3-space. We will discuss Barr's mapping further in a later section.

Peachey mentions some two-part mappings in his paper on solid texturing.[5] His orthogonal and cylindrical projections correspond to our slide projector and shrinkwrap mappings, respectively (these will be defined below).

Mapping from 2D to 3D

Let us first consider the mapping S from the texture plane to the intermediate surface. What sort of surfaces can we use for the intermediate surface so that S will have little distortion? Certainly any shapes that can be made by cutting, folding, and curling paper (i.e., without stretching it) will have no distortion at all. These shapes include planes, cylinders, and blocks. Later we describe the S mapping to each of these classes of surface. We contrast these mappings with several sphere mappings, which have high distortion.

Plane *S* mapping

The texture pattern is already on a plane, so this mapping may seem like the identity mapping. Remember, however, that we are embedding a two-dimensional object in 3-space. This requires information about which plane in 3-space will be used, where on that plane the texture pattern is to be placed, and how the pattern will be scaled. All in all, we must specify eight parameters (three translational components, x, y, and z; three rotational components, roll, pitch, and yaw; and two scaling components, c and d). Since all of the S mappings involve specifying a translation and rotation (a coordinate frame), we will ignore the six translation and rotation parameters when we express S mappings formally. The plane mapping, then, is simply

$$S : [x_a, y_a] \rightarrow [cx_a, dy_a],$$
$$S^{-1} : [x_s, y_s] \rightarrow [\frac{1}{c}x_s, \frac{1}{d}y_s],$$

where c and d are scaling factors. No clipping of the artwork occurs.

Planes are not very useful as intermediate surfaces because most objects we want to map patterns onto are topologically equivalent to a sphere. If we use planes, then we will have to complicate our O mapping to bridge this gap in topology as well as the gap in surface shape. Plane-shaped garments don't fit most of our clients.

Cylinder *S* mapping

Figure 2a illustrates a scanned-in pencil sketch of a pair of eyes. The region inside the rectangle is about to be mapped onto a cylindrical tube in the same way that a wine label is placed on a bottle. That is, the image inside the rectangle is curled around until its left edge touches its right edge. Figure 2b shows the eye's artwork mapped onto an ellipsoid.

We can represent the cylinder parametrically as a set of points $[\theta, h]$. The cylinder S mapping has six parameters in addition to the coordinate frame. We must choose the radius r of the cylinder, scale the artwork by $[c,d]$ about its center, and choose a position $[\theta_0, h_0]$ on the cylinder for the center of the artwork. This is sufficient to define a many-to-one mapping from vertical strips of the artwork plane (each of length 2π) to the cylinder. To make S one-to-one, we can choose a particular vertical strip. This leaves five parameters, r, c, d, θ_0, and h_0.

$$S : [x_a, y_a] \rightarrow [\frac{c}{r}x_a + \theta_0, d\, y_a + h_0], \quad \text{if } -\pi r < x_a \leq \pi r.$$
$$S^{-1} : [\theta, h] \rightarrow [\frac{r}{c}(\theta - \theta_0), \frac{1}{d}(h - h_0)], \quad \text{for } -\pi < \theta < \pi.$$

Cylinders are more useful than planes as intermediate surfaces. When the target surface is a solid of revolution, cylinders are often just what is needed. Inherent in the cylinder mapping is a single vertical edge of discontinuity.

Box *S* mapping

Figure 3 illustrates an S mapping from a plane to a box that is reminiscent of wrapping a package.

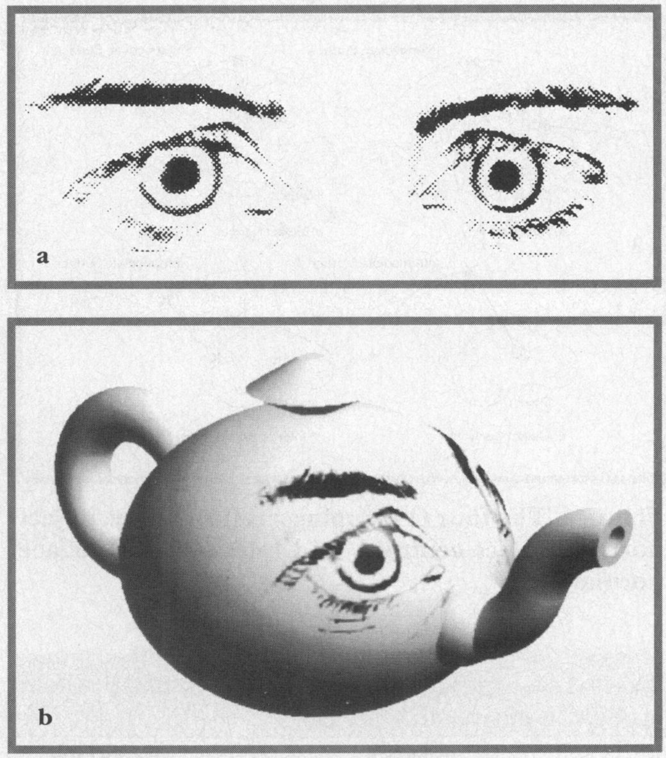


Figure 2. A scanned-in pencil sketch is (a) framed to show how much of it will be used when it is wrapped around a cylinder intermediate surface and (b) mapped onto the body of a teapot using the shrinkwrap mapping.

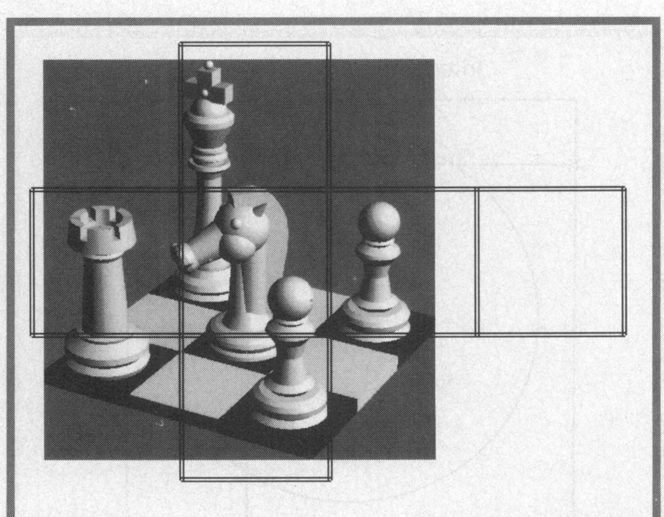


Figure 3. An unfolded box is placed over the chess texture pattern. Only those parts of the pattern inside the box will end up on the box intermediate surface.

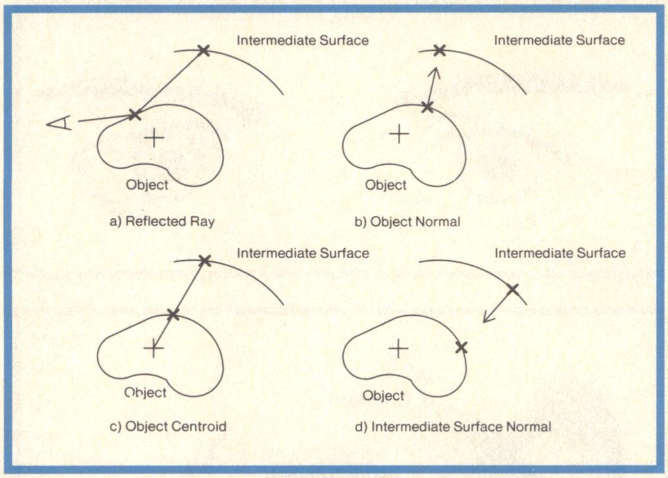


Figure 4. The four *O* mappings: reflected ray, object normal, object centroid, and intermediate surface normal.

Table 1. Properties of the three *O* (intermediate surface to object) mappings.

	SINGLE-VALUED	INVERTIBLE	CONTINUOUS
OBJECT NORMAL	Yes	Hopeless	If Surface Normal Is
OBJECT CENTROID	Yes	Yes	Yes
INTERMEDIATE SURFACE NORMAL	With Arbitration	Yes	Rarely

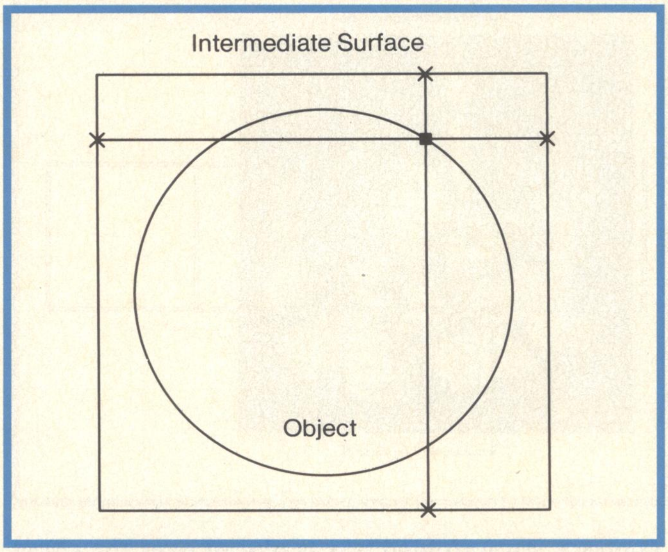


Figure 5. A two-dimensional analog of the ambiguous mapping problem. Four points on the box map to the same point on the circle.

There are seven parameters in this mapping. We must choose the side lengths $[a,b,c]$ of the box, scale the artwork by $[s_x,s_y]$, position the artwork on the unfolded box at position $[x_{a0},y_{a0}]$. In most cases, this mapping will rather drastically clip the original artwork. The mapping is not difficult, but it is tedious to describe, so we will omit a formal description.

Boxes are topologically equivalent to spheres. Hence they are the intermediate shape of choice for general-purpose use. Unfortunately, we have paid for this generality with two unpleasant properties of box *S* mapping. First, we are forced to clip a nonrectangular chunk from the original artwork. Second, artwork pieces that were not adjacent in the original pattern become adjacent at seven of the 12 edges of the box.

In three cases these discontinuities are less of a problem. First, if the artwork we are placing is just a decal (it will take up a small fraction of the target object's surface), then we may be able to fit the artwork within the four horizontally adjacent rectangles of the unfolded box. If this is true, however, we would do better to use the cylinder surface to reduce distortion. Second, if the pattern contains a few foreground figures on a constant-color background, it may be possible to place the artwork on the unfolded box so that most of the edges are on the background. Third, if the pattern is relatively homogeneous and invariant to 90-degree rotations (like checkerboards), it may be possible to position the unfolded box to hide the seams.

Sphere *S* mapping

There are no zero-distortion mappings from the plane to the sphere. In fact, many such well-known mappings as the spherical coordinates mapping and the Mercator projection have unbounded distortion at the two poles. However, there are some relatively low distortion mappings from the plane to a hemisphere, such as the stereographic projection. Using such a mapping, we can map two circular patterns onto two hemispheres. A discontinuity will occur only at the equator. Unfortunately, this line of discontinuity will show up for all but the most isotropic of patterns.

The stereographic projection is defined as

$$x_a = 2p/(1 + \sqrt{1 + p^2 + q^2}),$$
$$y_a = 2q/(1 + \sqrt{1 + p^2 + q^2}),$$

where p and q result from the gnomonic projection

$$[p, q] = [\tan\phi\cos\theta, \tan\phi\sin\theta].$$

where ϕ and θ are the usual angles of spherical coordinates.

Despite the stretching involved in sphere S mapping and its one large discontinuity, it may provide an alternative when the erratic discontinuities of box mapping are unacceptable.

Dealing with finite patterns

Many S mappings are easier to implement if the artwork is assumed to fill an infinite plane. Since any given sampled image will be finite, we give each artwork a background color, which is used when $[x_a, y_a]$ is exterior to the sampled image. This method works well to wrap a wine label on a wine bottle. For this example, we would use cylinder S mapping, and the artwork would be made up of both a sampled image that looks like a wine label and a glass green background color appropriate for the bottle itself.

Mapping from intermediate surface to target object

Now we wish to map the texture from our intermediate surface to an arbitrary target object. Our mapping should involve only the geometric properties of the target object (not its representation). Where possible, the mapping should also take advantage of the properties of the intermediate surface to achieve efficiency and low distortion. We have encountered these four techniques for performing the O mapping (or its inverse):

1. *Reflected ray.* Fire a ray from the eyepoint to the object. Compute the reflected ray and follow it until it reaches the intermediate surface. This maps in the O^{-1} direction (from the object to the intermediate surface).
2. *Object normal.* Follow a line in the surface normal direction until it reaches the intermediate surface (O^{-1} direction). Barr has dubbed this the "decal" mapping.[9]
3. *Object centroid.* Follow a line from the object's centroid through the point of interest on the object surface until it reaches the intermediate surface (O^{-1} direction). A slightly different mapping uses the centroid of the intermediate surface instead. Because the analysis is very similar, we consider only the object centroid case.
4. *Intermediate surface normal.* Follow a line from an intermediate surface point in the direction of the intermediate surface normal, until the line reaches the object surface (O direction). Both of Peachey's projection mappings fall into this category.[5] We'll sometimes refer to this mapping by its initials, ISN.

Figure 4 diagrams these four O mappings. The reflected ray method is useful for environmental reflections, but not for mapping artwork onto surfaces, since it is viewpoint dependent. The other three methods are appropriate for artwork mapping. How shall we choose one rather than another? Some of the issues are as follows:

1. Is the mapping single-valued?
2. Is the mapping invertible?
3. Is the mapping continuous?
4. Does the mapping have low distortion for most objects?

In this section we will try to answer the first three questions. The answers are summarized in Table 1. Later we will analyze distortion characteristics.

Single-valued mapping

For normal rendering we use O^{-1} (later we will discuss a type of interactive rendering that uses O). Since rendering is our primary concern, we would like O^{-1} to be well defined for each of the three mapping styles, even if O isn't well defined. Otherwise, the final image will be ambiguous. The object normal and object centroid mappings are usually single-valued, and the ISN mapping can be made so.

Object normal

The inverse of the object normal mapping (from object to intermediate surface) is single-valued if the object surface is entirely contained inside the intermediate surface (or if the intermediate surface is a plane). This follows from the fact that all our intermediate surfaces are convex. The term *single-valued* is used loosely here, since the object normal could miss the intermediate surface altogether, as with the plane and cylinder. We decide that the mapping takes on a background color in such cases, which gives the mapping a single value everywhere.

Object centroid

This case is like the object normal case, but we use the direction from the object's centroid instead of the surface normal. The mapping is single-valued if the object's centroid is within the intermediate surface.

Intermediate surface normal

This case is not single-valued in general. Figure 5 shows that many different points on a box inter-

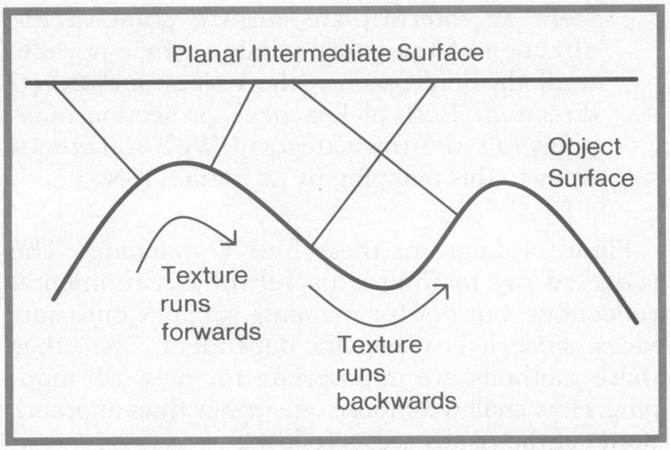


Figure 6. Surface concavities can cause the texture pattern to reverse if the object normal mapping is used.

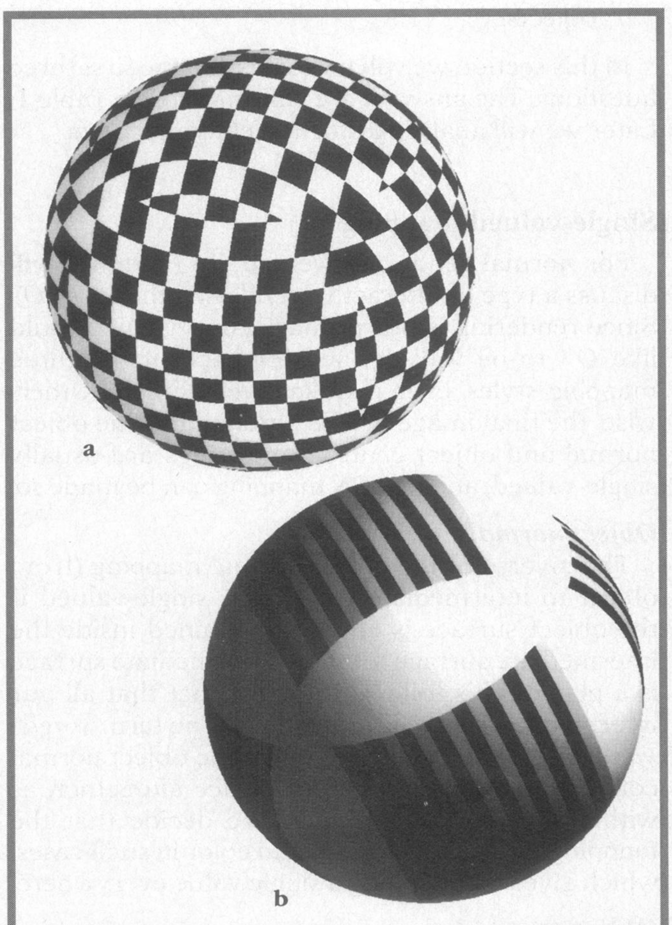


Figure 7. ISN/box mapping. In (a) the side of the box equals the diameter of the sphere. There is a line of discontinuity caused by each edge of the cube. In (b) the side of the cube equals one half the diameter of the sphere. Note that some sphere points receive no artwork color. A different texture pattern has been used to make the edges of the texture distinct.

mediate surface can point at the same object surface point (in fact, six box points will paint each object point). In Solidviews we make the mapping single-valued by choosing the intermediate surface point whose normal is closest in direction to the normal at the object surface point. Since the intermediate surfaces are convex, this is always adequate. Note that with our rules for this, the object need not be contained inside the intermediate surface (we can let the intermediate surface cast rays in both the inward and outward normal directions). The optimal ISN mapping from a cube to a sphere makes use of this.

Table 2. The appropriateness of the 12 intermediate surface/*O* mapping combinations. Very inappropriate combinations are X'ed out. Somewhat inappropriate combinations are \'ed out.

	CYLINDER	BOX	SPHERE	PLANE
OBJECT NORMAL	Topologically Unpromising	OK	OK	Redundant (See below)
OBJECT CENTROID	Topologically Unpromising	Centroid/ Box	Centroid/ Sphere	Redundant (See below)
INTERMEDIATE SURFACE NORMAL	Shrinkwrap	ISN/Box	Redundant (See above)	Slide Projector

Invertible mappings

Is *O* single-valued? No, not for any of the mapping styles. If the object is not convex, then multiple object points can be painted by the same intermediate surface point. This is not a terrible problem, however. To perform *O* correctly, we must find all object points painted by a given intermediate surface point. How difficult this is depends on the mapping.

With ISN we can just find the intersection of the intermediate surface's normal line with the object and then test each surface point as though we were performing O^{-1} to find the subset that is actually painted (some may be ruled out by the step that makes the mapping single-valued).

Similarly, with the object centroid style, we need find only those points intersected by the line segment spanning the centroid and the intermediate surface points of interest.

The object normal style is hopeless. We would need to find all surface points whose surface normal aims at the intermediate surface point. We know of no efficient algorithms to do this.

Continuity of mappings

Let us define a mapping to be *continuous* if every pair of adjacent object surface points is painted from adjacent points on the intermediate surface.

The object centroid method is continuous under this definition.

The object normal method is continuous everywhere the surface normal is continuous, which is often adequate. It is worth noting, however, that if the object is not convex, then the pattern appears inverted in the concavities. Traveling along a line on a surface, one might see the texture appear forward, backwards, and forward again, as shown in Figure 6. Barr noted (in a private communication) that the decal should be kept close to the object surface, so that this inversion does not occur.

The intermediate surface normal method can have gross discontinuities at object surface points where the arbitration scheme, used to make the mapping single-valued, switches from favoring one intermediate surface region to favoring another. For instance, when the intermediate surface is a cube and the object is a sphere, as shown in Figure 7a, there is a line of discontinuity for each of the 12 edges of the cube. Figure 7b shows the lapses in texture that occur when the cube is much smaller than the sphere.

The 12 *O* mappings

The three mapping types and the four intermediate surfaces can be combined to form 12 types of O mapping (see Table 2). Of these 12, only five will be considered further. The rest will be eliminated because of poor invertibility or poor continuity in comparison with other mappings of similar function.

We eliminate all object normal mappings (Barr's decals) because of their poor invertibility and because they reverse texture in concavities. It is strange that the object normal mappings are the only mappings that take object curvature into account, yet are the least desirable for our purposes. The other mappings are considered on a case by case basis.

Cylinders

Both the object normal/cylinder mapping and the object centroid/cylinder mapping are eliminated because they inherit the topological problem of the cylinder: It is open at both ends. Both of these mapping styles map a large portion of the object surface to very distant points of the cylinder, which are either uninteresting (for small decals) or highly distorted (for plane-filling patterns). Fortunately, the ISN method avoids these problems by mapping only from cylinder points near the object. If the object is a solid of revolution, then the result looks as though the pattern were mapped onto a cylinder of heat-shrinkable plastic, which was then shrunk. We call this mapping *shrinkwrap*.

Boxes

Because a box is closed and finite and topologically equivalent to a sphere, all mappings for boxes are relatively well behaved.

Spheres

All sphere mappings are well behaved. For spheres, the ISN mapping is equivalent to an intermediate surface centroid mapping, and so it is similar to an object centroid mapping. Thus, the ISN mapping for spheres can be eliminated.

Planes

Planes have the same topological problem as cylinders. Again, most of the object surface ends up mapped to distant points on the plane. For tesselations, this results in unacceptable distortion. For decals, all three mapping styles work about equally well. In particular, for small decals, the object centroid and intermediate surface normal methods are practically identical, since the directions from the object centroid are nearly parallel. For this particular case, both methods are single-valued, invertible, and continuous. The intermediate surface normal style, for this case, works much like projecting the texture pattern onto the surface with a slide projector. If the object is nearly planar, this slide-projector mapping is probably preferable because of its lower distortion.

We are left with five different mappings: shrinkwrap, centroid/box, ISN/box, centroid/sphere, and slide projector. These mappings are roughly suited to three different kinds of objects: solids of revolution, roughly spherical blobs, and planar shapes. In the next section we define these five different mappings formally. Then we will analyze the distortion of each. Finally, we discuss the interactive tools needed to make general-purpose mapping feasible.

Implementing the O mappings

Because we have chosen simple intermediate shapes, the O mappings can be described with simple equations. We give these equations here, for those interested in implementing them.

In describing the shrinkwrap, centroid/box, ISN/box, centroid/sphere, and slide projector mappings, we make the simplifying assumption that the centroid coordinate frame of the object and the origin of the intermediate surface are coincident, and that the symmetry axis of a cylinder is its y axis.

Recall that, while O^{-1} is single-valued, O is multivalued, so we express the mapping O as the equation of a ray R or a line L (represented parametrically as a base point p and a direction vector $\vec{d}$) that

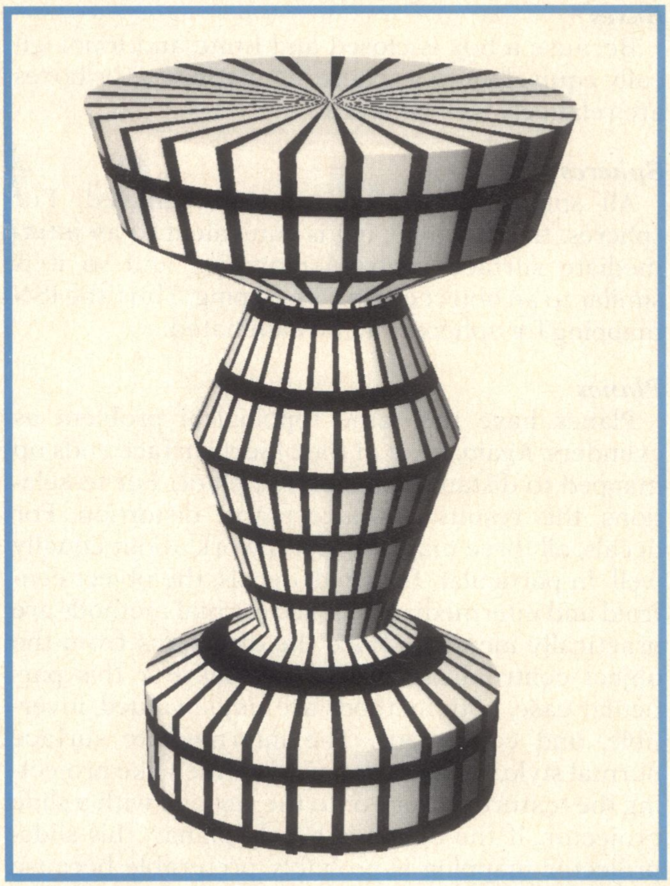


Figure 8. A grid pattern is mapped onto a solid of revolution using shrinkwrap mapping.

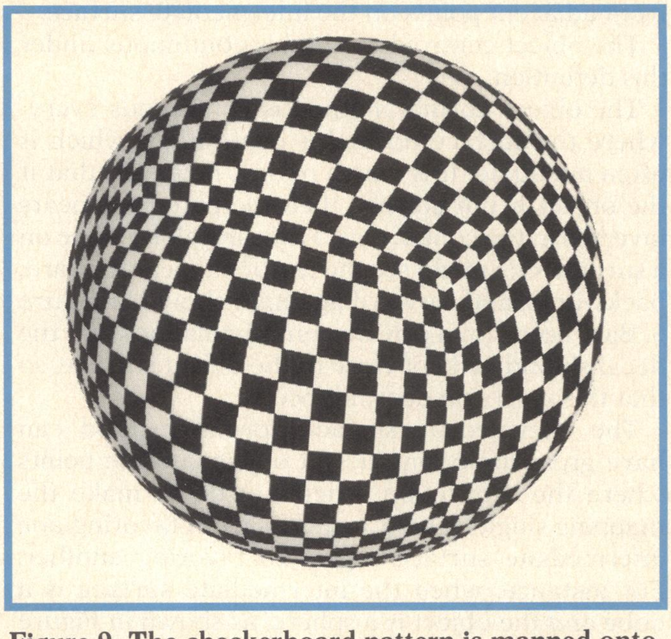


Figure 9. The checkerboard pattern is mapped onto a sphere with object centroid/box mapping. The distortion is low and none of the texture is lost. Compare this to Figure 7a, where texture is lost at each edge of the cube.

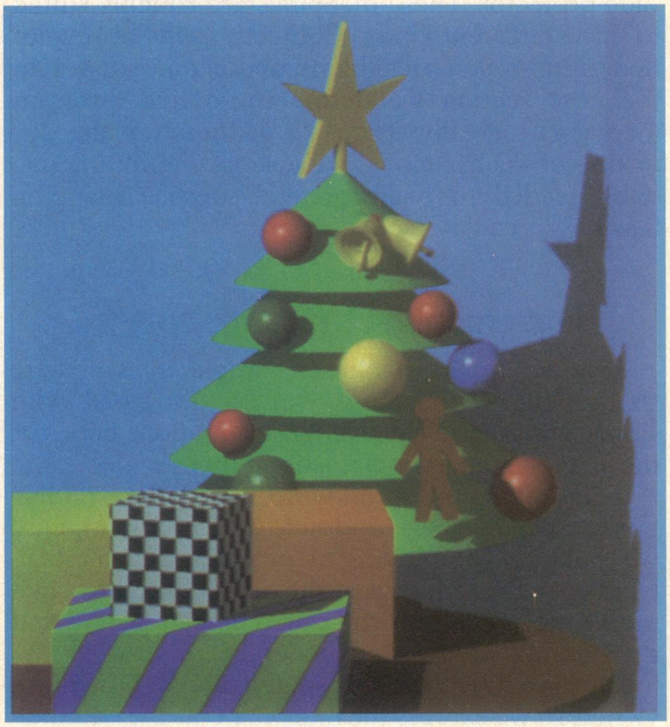


Figure 10. A Christmas tree scene where the boxes have been painted by ISN/box mapping.

includes all object points that will be painted by a particular intermediate surface point. For all ISN and centroid mappings, these rays or lines fill all the space, giving us a set of solid textures, which includes Peachey's projection mappings.[5]

The only values of interest in the *O* mapping are the intermediate surface point $[x_s,y_s,z_s]$, the object surface point $[x_o,y_o,z_o]$, and the size of the intermediate surface (radius *r* for cylinders, sides $[a,b,c]$ for boxes, radius *r* for spheres).

Shrinkwrap

Figure 2b and Figure 8 illustrate the shrinkwrap mapping onto solids of revolution. If we represent the cylinder as a set of points $[\theta,h]$ then the mapping is

$$\begin{aligned} O : [\theta, h] &\rightarrow R = \vec{p} + t\,\vec{d} \\ &\rightarrow [0, h, 0] + t\,[r\cos\theta, 0, -r\sin\theta] \\ O^{-1} : [x_o, y_o, z_o] &\rightarrow [\arctan(-z_o, x_o), y_o]. \end{aligned}$$

where arctan(*y*,*x*) is the common function that returns the angle whose tangent is *y*/*x*, whose sine has the sign of *y*, and whose cosine has the sign of *x*.

The problem with this mapping is the high distortion when the object narrows or widens rapidly (i.e., is more like a disk than a cylinder). Figure 8 shows a solid of revolution painted by the shrinkwrap mapping from a cylinder. Notice how the lines are stretched on the flatter cones and on the disks.

Centroid/box

If we represent the box as a set of points $[x_s, y_s, z_s]$ then the mapping is (numbering the faces in Figure 3 as 1, 2, 3, 4, left to right, 5 on the top, and 6 on the bottom)

$$\begin{aligned} O : [x_s, y_s, z_s] &\rightarrow \vec{R}(t) = \vec{p} + t\,\vec{d} \\ &\rightarrow [0, 0, 0] + t\,[x_s, y_s, z_s], \\ O^{-1} : [x_o, y_o, z_o] &\rightarrow \tfrac{1}{2}[cx_o/z_o, cy_o/z_o, c]. \end{aligned}$$

if the vector from [0, 0, 0] to $[x_o, y_o, z_o]$ points toward side 2, and similarly for the other five sides.

Figure 9 shows the object centroid mapping of a checkerboard pattern onto a sphere. Like Figure 7a, the length of the side of the cube intermediate surface was equal to the diameter of the sphere. The object centroid mapping has the advantage that no texture was lost in going from the box to the sphere.

ISN/box

Again let the box be a set of points $[x_s, y_s, z_s]$. For face 2 (which has surface normal [0, 0, 1] in the centroid coordinate frame), the mapping is

$$\begin{aligned} O : [x_s, y_s, z_s] &\rightarrow \vec{R} = \vec{p} + t\,\vec{d} \\ &\rightarrow [x_s, y_s, z_s] + t\,[0, 0, 1]. \end{aligned}$$

To perform O^{-1}: $[x_o, y_o, z_o]$, we must determine which face would paint $[x_o, y_o, z_o]$ if we were using O. Since all six faces want to paint $[x_o, y_o, z_o]$ this involves using our arbitration scheme. If the normal most nearly points in the direction of face 2's normal, then we have

$$O^{-1} : [x_o, y_o, z_o] \rightarrow [x_o, y_o, \tfrac{1}{2}c].$$

and similarly for the other five faces.

Two ISN mappings onto the sphere are shown in Figure 7. In both mappings there was a problem; either texture was lost, or some of the sphere was left unpainted.

It is clear that the easiest object to color using the ISN mapping is a box of exactly the same size and orientation as the intermediate surface. Figure 10 shows a Christmas tree scene in which the gift boxes have been wrapped in this manner. Of course, this is really a one-part mapping.

Centroid/sphere

Represent the sphere as the set of points $[x_s, y_s, z_s]$ such that $x_s^2 + y_s^2 + z_s^2 + r^2$. Then the mapping is simply

$$\begin{aligned} O : [x_s, y_s, z_s] &\rightarrow \vec{R}(t) = \vec{p} + t\,\vec{d} \\ &\rightarrow [0, 0, 0] + t\,[x_s, y_s, z_s], \\ O^{-1} : [x_o, y_o, z_o] &\rightarrow \frac{r\,[x_o, y_o, z_o]}{\|[x_o, y_o, z_o]\|}. \end{aligned}$$

Slide projector

If the plane is the z=0 plane in the centroid coordinate frame, then we have

$$\begin{aligned} O : [x_s, y_s, 0] &\rightarrow \vec{R}(t) = \vec{p} + t\,\vec{d} \\ &\rightarrow [x_s, y_s, 0] + t\,[0, 0, 1], \\ O^{-1} : [x_o, y_o, z_o] &\rightarrow [x_o, y_o, 0]. \end{aligned}$$

Distortion

When we combine our five interesting O mappings with the S mappings implied by their intermediate surfaces, we have five interesting M mappings, i.e., complete mappings from the artwork plane to an arbitrary object surface. We will analyze the distortion characteristics of these five M mappings.

In each case the distortion in the finished image depends on the final object. For shrinkwrap, if the object is a cylinder, there will be unbounded distortion at the ends. For both box mappings, if the object is a box of the same size and orientation as the intermediate box, these mappings will produce identical results with no distortion. However, each mapping will produce infinite distortion on a particular class of shapes. Hence, there is nothing we can say in general. However, our mappings do seem appropriate for particular kinds of objects—solids of revolution, roughly spherical blobs, and roughly planar shapes.

Since a sphere is both a solid of revolution and a roughly spherical blob, we analyze the distortion of the shrinkwrap, centroid/box, ISN/box, and centroid/sphere mappings when the object is a sphere. The slide projector mapping is more appropriate for roughly planar shapes, and is unchallenged in this domain, so we won't analyze it here.

In all cases, see Bier's master's thesis for a derivation.[7]

The sphere as a test case

What does it mean to wrap an image around a sphere with low distortion? Certainly the mappings should preserve adjacency (i.e., image features that

Table 3. Distortion characteristics of four two-part mappings.

	Homogeneity (1.0 is optimal)	Aspect Ratio (1.0 is optimal)
Shrinkwrap	0.0	0.0
ISN/Box	0.707	0.707
Centroid/Box	0.471	0.707
Centroid/Sphere (Stereographic Projection)	0.485	0.586

Sample derivation

In this example we find the distortion characteristics—homogeneity of resolution and worst-case aspect ratio—for the centroid/box mapping in the special case where the target object is a sphere, centered in the intermediate surface. The analysis of the shrinkwrap, ISN/box, and centroid/sphere mappings is similar.

To find the homogeneity of resolution. we assume a unit sphere, a cube of side 2, and that $dx_a = dx_s$ and $dy_a = dy_s$. Considering face 2 of the cube, from the formal definition of O^{-1}, we can find

$$dr_x = \sqrt{1 + y_a^2}/(1 + x_a^2 + y_a^2),$$
$$dr_y = \sqrt{1 + x_a^2}/(1 + x_a^2 + y_a^2).$$

The worst distortion occurs at the corners, giving a minimum to maximum ratio of .471.

The aspect ratio is 1 along both diagonals of the square. The ratio is worst at the four edge midpoints where it is .5/.707 approximately equal to .707.

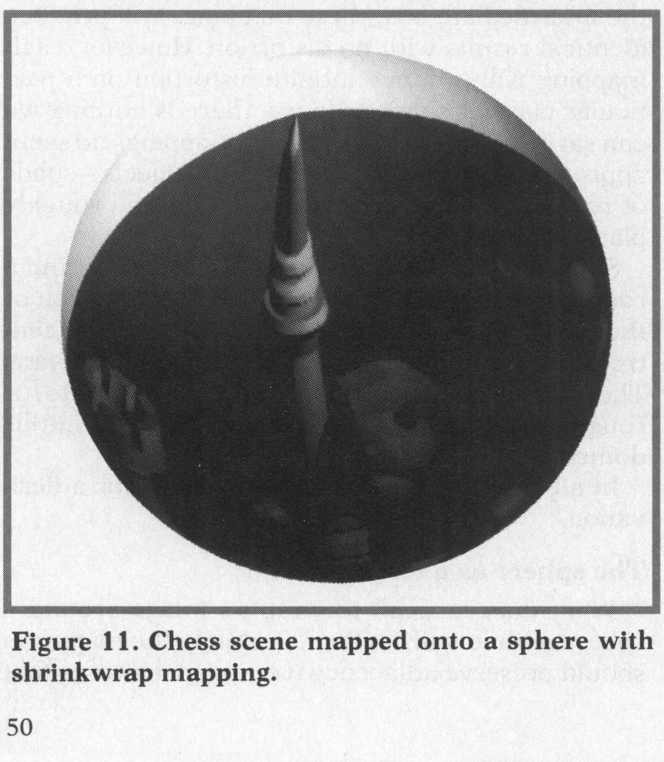


Figure 11. Chess scene mapped onto a sphere with shrinkwrap mapping.

are adjacent in the artwork should be adjacent on the surface of the sphere). Resolution should be homogeneous across the surface (e.g., a 1-unit-by-1-unit square portion of the artwork should map to the same size piece on the sphere, no matter where it is taken from on the artwork). Aspect ratio should be preserved (a feature 10 units high and 20 units wide in the artwork should be twice as wide as it is high on the surface of the sphere for some reasonable measure of width and height). None of the mappings above satisfy all these requirements for the sphere. However, we can quantify the homogeneity of resolution and the distortion of aspect ratio as described below.

Homogeneity of resolution

If we measure the arc-length dr_x traversed on the sphere as a result of a motion dx in the artwork (and the arc-length dr_y traversed on the sphere as a result of a motion dy in the artwork) in the neighborhood of some point $[x_a, y_a]$, we will have a quantization of the effective scaling of the mapping in the x and y directions at that point. Ideally, this scaling would be a constant over the sphere. However, each of the mappings deviates from this ideal. The ratio of the maximum value of dr_x to the minimum value of dr_x (and of the maximum value of dr_y to the minimum value of dr_y) is a measure of this deviation.

Aspect ratio

Assuming that the arcs corresponding to the dx and dy motion in the artwork are perpendicular, dr_y / dr_x gives the aspect ratio of the mapping onto the sphere at a given point $[x_a, y_a]$. These quantities are close enough to orthogonal for the mappings considered here. We will find the maximum value of this ratio (or its inverse, whichever is in the range $[0, \ldots, 1]$) for each mapping.

The results

The results are summarized in Table 3. A sample derivation is given in the box. The other derivations are similar.

The shrinkwrap mapping is the clear loser. Even when the object being painted is a solid of revolution (as a sphere is), infinite distortion of both types can occur. Figure 11 illustrates this effect; the neck of the chess king tapers to a point at the sphere's north pole (his head is too high to be included in the mapping at all). Also, recall the high distortion in Figure 8. Only shapes that are very close to cylindrical will have low distortion.

The ISN/box mapping is the winner. The reason is this: The ISN/box mapping does not map texture from points near the corners of the intermediate surface box. It is exactly at these corners that maximum distortion occurs in the centroid/box mapping. Likewise, in the stereographic mapping (whose distortion is all in the *S* mapping), the maximum distortion occurs on the artwork points farthest from the center of the artwork. The cube in the ISN/box mapping stays relatively close to the sphere, avoiding this kind of distortion.

There is a trade-off here. The ISN/box mapping, which has the worst discontinuity behavior, has the best distortion behavior.

The chief problem with the ISN mapping is that, no matter what size you make the box with respect to the sphere, there will always be either some box points that don't get mapped onto the sphere (box too large) or some sphere points that receive no texture from the box (box too small). Letting the side of the box be $\sqrt{2}$ times the radius of the sphere will probably produce the best results; all sphere points get their color from the box, and the points of the box that are not mapped are only those points on each face of the box that are outside the intersection of the ellipses $2x^2 + y^2 \leq 1$ and $x^2 + 2y^2 \leq 1$ (using an *xy*-face as an example). Figure 12 shows the ISN mapping with the side of the cube equal to $\sqrt{2}$ times the radius of the sphere. The artwork is the chess scene of Figure 3. Figure 12 is the lowest distortion mapping in this article.

Figure 13 shows the centroid/box mapping applied to the chess scene. This picture brings out another unpleasant characteristic of the centroid mapping: even artwork lines that cross box edges at right angles can end up with slope discontinuities (note that the base of the king has a kink in it). It is interesting that the centroid mappings and ISN mappings have opposite stretching characteristics around the middle of a cube edge. The centroid mapping stretches $\sqrt{2}$ more along the edge direction than transverse to it. The ISN mapping (with a cube of side $\sqrt{2}$) stretches $\sqrt{2}$ more in the transverse direction than along the edge. The different characteristics are particularly apparent if one compares the mane of the horse in Figure 12 to the mane in Figure 13.

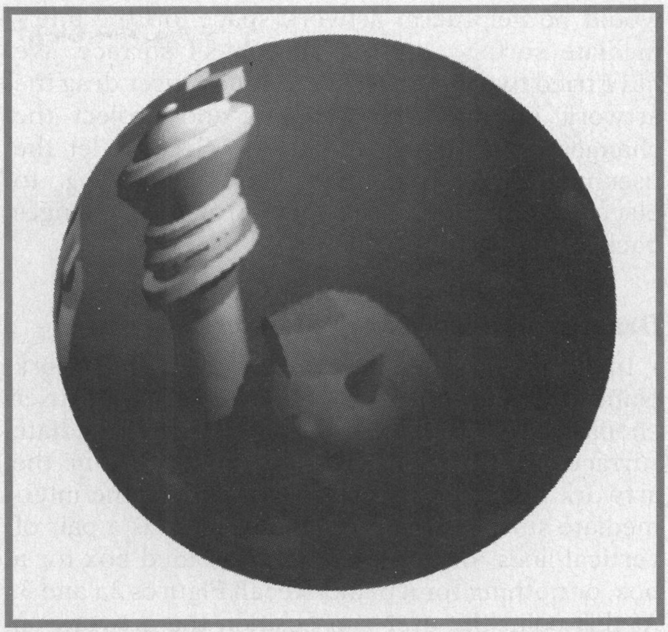


Figure 12. ISN/box mapping applied to the chess scene. The side of the simple surface cube is $\sqrt{2}/2$ times the diameter of the sphere. This cube size minimizes the amount of data lost, given that all sphere points are to receive color from the texture pattern. The distortion is hard to perceive.

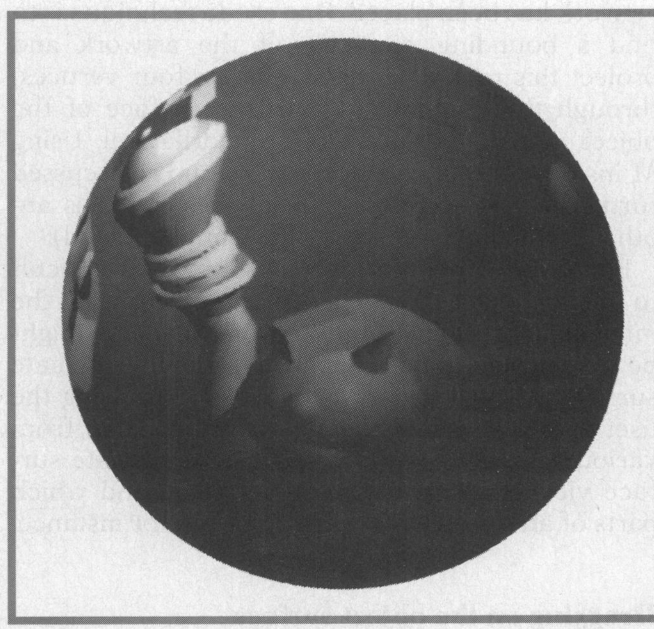


Figure 13. Centroid/box mapping of the chess scene onto a sphere.

User feedback

Several types of feedback can help the user position artwork on a surface. In his thesis, Blinn[4] mentions his problems with feedback: "Many of the pictures produced with texture mapping had to go through several iterations where the pattern came out reflected or rotated from what was desired." With two-part mappings there are three stages of the mapping process where artwork feedback would be helpful: in artwork space, on the intermediate surface, and on the object surface. We have tried two strategies: First, let the user drag the artwork in the artwork plane, and project the changes onto the object surface. Second, let the user point directly at the object surface (e.g., to place the center of the artwork), projecting changes back to the artwork plane.

Dragging in the artwork plane

In our system an interactive session with artwork plane dragging works as follows: First, the user chooses the size and shape of the intermediate surface. At this point the system draws, in the artwork plane, an "unfolded version" of the intermediate surface. This unfolded version is a pair of vertical lines for a cylinder, an unfolded box for a box, or nothing for a plane. Recall Figures 2a and 3. At this point the user can position the artwork on the intermediate surface by scaling, translating, and rotating the artwork in the artwork plane. This is very natural for the user, and easy to program.

An improved feedback system might also include projecting the object surface back onto the artwork. Each object might have an associated wireframe representing its "important features." Projecting these features to the artwork plane requires not only the existence of O^{-1} for points, but also for lines.

More helpful than seeing the object in the artwork plane, however, is seeing how the artwork looks on the final object. If rerendering the object is fast, then we are done. Otherwise, we may be willing to settle for a vague notion of where the artwork has been placed. One crude technique is to find a bounding rectangle of the artwork and project this rectangle (or at least its four vertices) through the mapping M onto the surface of the object, drawing the resulting quadrilateral. Using M, instead of M^{-1}, is called pushing and is discussed further below. Rendering at low resolution is another possibility (the one we have implemented).

For fairly complex S mappings it may be difficult to see the effects of S from a projection of the intermediate surface into artwork space. It might be worthwhile to be able to treat the intermediate surface as an object in its own right, allowing the user to display it, with the artwork mapped on, from various points of view. Such an intermediate surface view might help the user understand which parts of an unfolded box are adjacent, for instance.

Dragging on the object surface

If we have a way of pointing at the object's surface interactively (e.g., with the poke-at-it interaction technique suggested by Roth[10]), then we can point where we want the center of the artwork (or some other artwork feature) to go. The user will probably still want to see the artwork (and the unfolded intermediate surface) in the artwork plane, to understand why various discontinuities are occurring.

Sampled and synthetic artwork: pushing and pulling

As mentioned earlier, one of our goals is to be able to work with undistorted texture patterns. These texture patterns may be sampled images from a paint program or synthetic images from an illustration system that produces synthetic shapes (e.g., analytically represented lines and curves). All texture mapping known to us uses sampled images. Synthetic images raise some interesting issues.

When our rendering software encounters an object surface point that has been texture mapped, it computes the artwork point $[x_a,y_a]$ from which the color should come. At that time the system must determine the artwork color at $[x_a,y_a]$. For sampled images this requires indexing into the sample array. For synthetic images this requires hit testing, which is fairly time-consuming. The standard approach is to scan convert the synthetic image to make a sampled image and use that. For antialiasing, however, it might be preferable to use the synthetic image as it is, mapping the filter function into the artwork plane, and integrating over a region of the synthetic image.

Pushing from a synthetic image is even more interesting. As just described, we may wish to push the artwork from the artwork plane to the object surface for positioning feedback. This can probably be done more rapidly with a synthetic image than with a sampled image. Pushing is not trivial, however. The lines, splines, circles, etc., of the synthetic image must somehow be transformed by the mapping M and painted on the object. This is a topic for future research.

Conclusion

Mapping from planar artwork to an arbitrarily curved surface is a difficult proposition if we wish to use two-dimensional patterns that have not been predistorted to suit our choice of mapping. If this is not necessary, then such other techniques as manual predistortion, computed predistorted patterns, solid texturing, and environmental mapping (which computes the texture on the intermediate surface to begin with) are worth trying.

We have considered a set of 12 two-part mappings. Of these, only five are useful: the shrinkwrap mapping from a cylinder, the centroid and intermediate surface normal mappings from a box, the centroid mapping from a sphere, and the slide projector mapping from a plane. None of these mappings is perfect. The shrinkwrap mapping is most useful for solids of revolution that slope gradually (i.e., do not contain flat cones or disks). The box mappings are the best general-purpose mappings for texturing entire curved bodies. However, the artwork must be carefully cut up into an unfolded box shape to hide the inherent discontinuities of the giftwrap stage of the mapping. Any mapping with a sphere intermediate surface will involve a high-distortion first step to get the texture onto the sphere. Finally, the slide projector mapping can texture only half an object, so it is best reserved for small decals, where the plane is nearly tangent to the object surface.

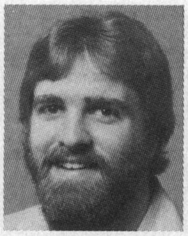
All these mappings are made more useful by interactive tools for positioning the artwork (both on the intermediate surface and on the target surface) and for previewing the current mapping. ■

Acknowledgments

We thank our reviewers for pointing us at related work, for useful suggestions on the organization of this article, and for describing their own uses of two-part mappings. Thanks also to the people of the Computer Science Lab of Xerox PARC for providing the computing environment that made the work possible. Special thanks go to Subhana Menis for a tasteful and thorough job of proofreading. This work was supported in part by a scholarship from AT&T Bell Laboratories and an internship with Xerox PARC.

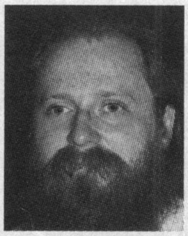
References

1. James F. Blinn and Martin E. Newell, "Texture and Reflection in Computer Generated Images," *Comm. ACM*, Vol. 19, No. 10, Oct. 1976, pp.542-547.
2. James F. Blinn, "Computer Display of Curved Surfaces," PhD dissertation, Univ. of Utah, Dec. 1978.
3. Edwin Catmull, *A Subdivision Algorithm for Computer Display of Curved Surfaces*, PhD dissertation, Univ. of Utah, rpt. no. UTEC-CSc-74-133, Dec. 1974, pp.36-37.
4. James F. Blinn, "Simulation of Wrinkled Surfaces," *Computer Graphics* (Proc. SIGGRAPH 78), Vol. 12, No. 3, Aug. 1978, pp.286-292.
5. Darwyn R. Peachey, "Solid Texturing of Complex Surfaces," *Computer Graphics* (Proc. SIGGRAPH 85), Vol. 19, No. 3, July 1985, pp.279-286.
6. Ken Perlin, "An Image Synthesizer," *Computer Graphics* (Proc. SIGGRAPH 85), Vol. 19, No. 3, July 1985, pp.287-296.
7. Eric A. Bier, "Solidviews: An Interactive Three-Dimensional Illustrator," Master's thesis, EECS Dept., MIT, Cambridge, Mass., May 1983.
8. Eliot A. Feibush, Marc Levoy, and Robert L. Cook, "Synthetic Texturing Using Digital Filters," *Computer Graphics* (Proc. SIGGRAPH 80), Vol. 14, No. 3, July 1980, pp.294-301.
9. Alan H. Barr, "Decal Projections," ACM SIGGRAPH 84, course 15: Mathematics of Computer Graphics, July 1984.
10. Scott D. Roth, "Ray Casting for Modeling Solids," *Computer Graphics and Image Processing*, Vol. 18, 1982, pp.109-144.



Eric A. Bier is a PhD candidate in electrical engineering and computer sciences at the University of California, Berkeley. Since 1980 he has been a part-time consultant to Xerox PARC. His research interests include interactive geometric design, solid modeling, and user interface management systems.

Bier received an SB and SM in computer science from the MIT. He is a member of IEEE, ACM, Tau Beta Pi, and Sigma Xi. From 1984 to 1985 he was an AT&T Bell Laboratories Scholar.



Kenneth R. Sloan, Jr., is assistant professor of computer science at the University of Washington, Seattle. From 1981 to 1984 he was assistant professor of computer graphics at Massachusetts Institute of Technology. From 1977 to 1981 he was research associate and instructor at the University of Rochester. He has worked for Western Electric and IBM. His research interests include computer graphics and animation, computer vision, artificial intelligence, and distributed computing.

Sloan received an ScB in applied mathematics from Brown University in 1970; an MS in computer science from Stevens Institute of Technology in Hoboken, New Jersey, in 1973; and a PhD in computer and information science from the University of Pennsylvania, Philadelphia, in 1977. He is a member of IEEE, ACM, AAAS, and AAAI.

The authors can be reached through Eric Bier at Xerox PARC, 3333 Coyote Hill Road, Palo Alto, CA 94304.