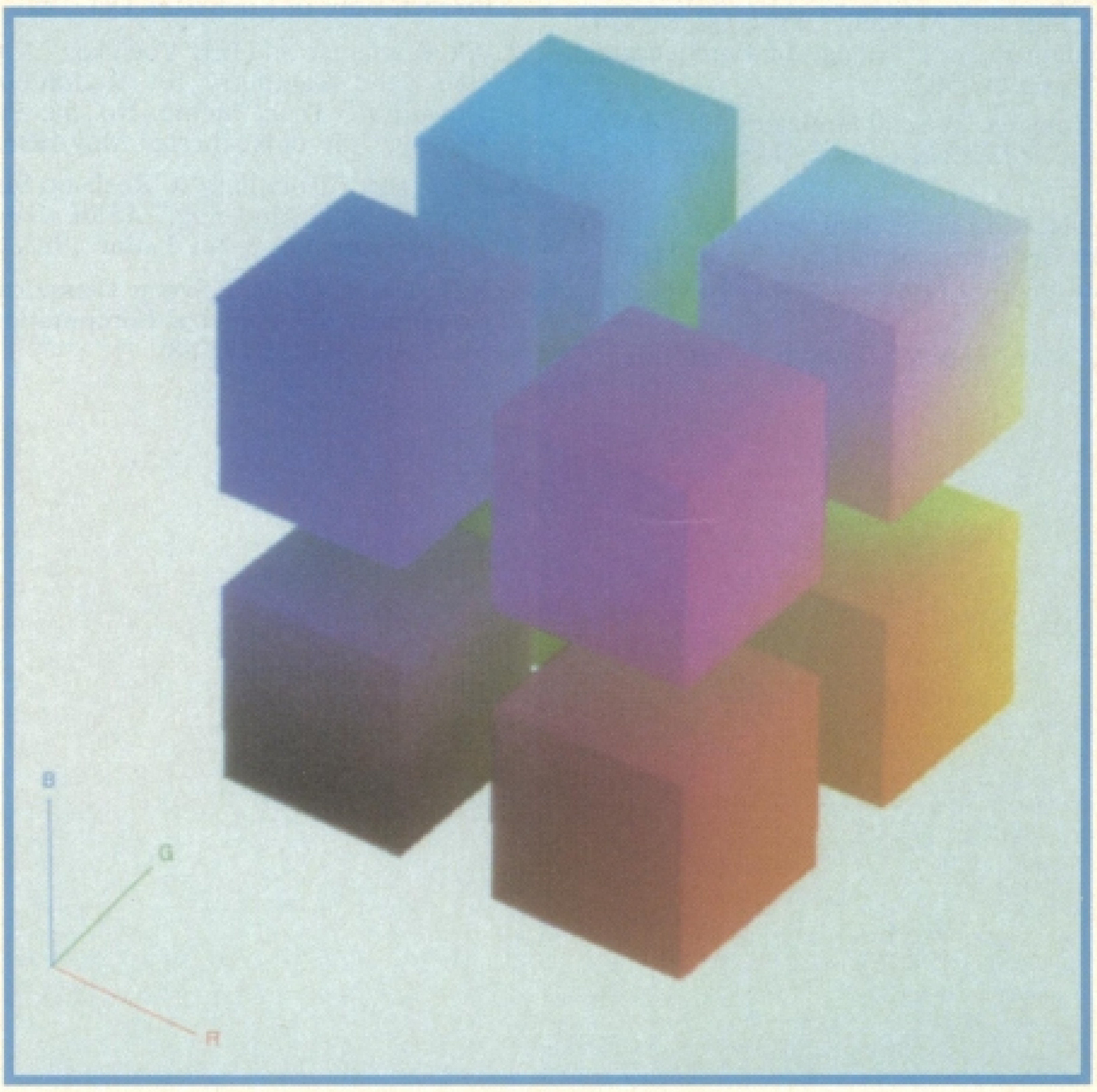
# Two-Part Texture Mappings

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 microstructure (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  

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