SURVEY OF TEXTURE MAPPING

Paul S. Heckbert

Computer Graphics Lab
New York Institute of Technology †

ABSTRACT

Texture mapping is one of the most successful new techniques in high quality image synthesis. Its use can enhance the visual richness of raster scan images immensely while entailing only a relatively small increase in computation. The technique has been applied to a number of surface attributes: surface color, surface normal, specularity, transparency, illumination, and surface displacement, to name a few. Although the list is potentially endless, the techniques of texture mapping are essentially the same in all cases. We will survey the fundamentals of texture mapping, which can be split into two topics: the geometric mapping which warps a texture onto a surface, and the filtering which is necessary in order to avoid aliasing. An extensive bibliography is included.

KEYWORDS: texture mapping, texture filter, space variant filter, antialiasing.

INTRODUCTION

Why Map Texture?

In the quest for more realistic imagery, one of the most frequent criticisms of early synthesized raster images was the extreme smoothness of surfaces - they showed no texture, bumps, scratches, dirt, or fingerprints. Realism demands complexity, or at least the appearance of complexity. Texture mapping is a relatively efficient means to create the appearance of complexity without the tedium of modeling and rendering every 3-D detail of a surface.

The study of texture mapping is valuable because its methods are applicable throughout computer graphics and image processing. Geometric mappings are relevant to the modeling of parametric surfaces in CAD and to general 2-D image distortions for image restoration and artistic uses. The study of texture filters leads into the development of space variant filters, which are useful for image processing, artistic effects, depth-of-field simulation, and motion blur.

Definitions

We define a *texture* rather loosely: it can be either a texture in the usual sense (e.g. cloth, wood, gravel) - a detailed pattern which is repeated many times to tile the plane, or more generally, a multidimensional image which is mapped to a multidimensional space. The latter definition encompasses non-tiling images such as billboards and paintings.

Texture mapping means the mapping of a function onto a surface in 3-D. The domain of the function can be one, two, or three-dimensional, and it can be represented either by an array or by a mathematical function. For example, a 1-D texture can simulate rock strata; a 2-D texture can represent waves, vegetation [Nor82], or surface bumps [Per84]; a 3-D texture can represent clouds [Gar85], wood [Pea85], or marble [Per85a]. For our purposes textures will usually be 2-D arrays.

The source image (*texture*) is mapped onto a surface in 3-D *object space* which is then mapped to the destination image (*screen*) by the viewing projection. Texture space is labeled (u,v), object space is (x_o,y_o,z_o), and screen space is (x,y).

We assume the reader is familiar with the terminology of 3-D raster graphics and the issues of antialiasing [Rog85], [Fol82].

Uses for Texture Mapping

The possible uses for mapped texture are myriad. Some of the parameters which have been texture mapped to date are: surface color (the most common use) [Cat74], specular reflection [Bli76], normal vector perturbation ("bump mapping") [Bli78a], specularity (the glossiness coefficient) [Bli78b], transparency [Smi79], diffuse reflection [Mil84], surface displacement and mixing coefficients [Coo84b].

Illumination Mapping

Mapping specular and diffuse reflection is rather different from mapping other parameters, since these maps are not associated with a particular object in the scene, but to an imaginary infinite radius sphere, cylinder, or cube surrounding the scene [Gre86a]. Whereas standard texture maps are indexed by the surface parameters u and v, a specular reflection map is indexed by the reflected ray direction [Bli76] and the diffuse reflection map is indexed by the surface normal direction [Mil84]. The technique can be generalized for transparency as well, indexing by the refracted ray direction [Kay79]. In the special case that all surfaces have the same reflectance and they are viewed orthographically the total reflected intensity is a function of surface orientation only, so the diffuse and specular maps can be merged into one [Hor81].

Illumination mapping, as these techniques are called, facilitates the simulation of complex lighting environments, since the time required to shade a point is independent of the number of light sources. Other reasons for its recent popularity are: it is one of the few demonstrated techniques for antialiasing highlights [Wil83], it is an inexpensive approximation to ray tracing for mirror reflection, and to radiosity methods [Gor84] for diffuse reflection of objects in the environment. Efficient filtering is especially important for illumination mapping, where variations in surface curvature often necessitate broad areas of the sky to be averaged.

Since specular reflection varies as a function of the viewing direction, it is most conveniently computed on the fly, as in ray tracing. Diffuse reflection of the environment, however, has not yielded to ray tracing even when stochastic methods [Coo84a] are used. The problem is that diffuse reflection scatters light over an entire hemisphere, not a narrow cone, as does specular reflection. Fortunately diffuse reflection is independent of viewing direction, so the incident illumination at each surface point can be precomputed and treated as a texture [Coo84b]. Previous methods have approximated this using polygon subdivision to model hard shadows [Ath78], soft shadows [Nis83], beams of light [Hec84], or indirect illumination [Gor84]. With the development of more efficient algorithms for its computation, incident illumination promises to be a common use for textures in the future.

Even when direct support for illumination mapping is unavailable, tricks can be employed which give a visually acceptable approximation. Rather than calculate the exact ray direction at each pixel, one can compute the reflected or refracted ray direction only at polygon vertices and interpolate it, in the form of u and v texture indices, across the polygon using standard methods. This approximation is similar to that made by beam tracing [Hec84].

† Current address: Pacific Data Images, 1111 Karlstad Dr., Sunnyvale, CA 94089, USA.