# Polynomial Texture Maps

Tom Malzbender, Dan Gelb, Hans Wolters

Hewlett-Packard Laboratories[1]
http://www.hpl.hp.com/ptm

**Figure 1:** Top: Lighting changes across a polynomial texture map, bottom: across a conventional texture map.

## Abstract

In this paper we present a new form of texture mapping that produces increased photorealism. Coefficients of a biquadratic polynomial are stored per texel, and used to reconstruct the surface color under varying lighting conditions. Like bump mapping, this allows the perception of surface deformations. However, our method is image based, and photographs of a surface under varying lighting conditions can be used to construct these maps. Unlike bump maps, these Polynomial Texture Maps (PTMs) also capture variations due to surface self-shadowing and interreflections, which enhance realism. Surface colors can be efficiently reconstructed from polynomial coefficients and light directions with minimal fixed-point hardware. We have also found PTMs useful for producing a number of other effects such as anisotropic and Fresnel shading models and variable depth of focus. Lastly, we present several reflectance function transformations that act as contrast enhancement operators. We have found these particularly useful in the study of ancient archeological clay and stone writings.

**CR Categories:** I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.7 [Computer Graphics] Three-Dimensional Graphics and Realism - Color, shading, shadowing and texture; I.4.1 [Image Processing and Computer Vision] Digitization and Image Capture - Reflectance

**Keywords**: Graphics Hardware, Illumination, Image Processing, Image-Based Rendering, Reflectance & Shading Models, Texture Mapping

[1] {malzbend,dgelb,wolters}@hpl.hp.com

## 1 Introduction

Polynomial Texture Mapping enables greatly improved realism over conventional methods. Traditional texture mapping is used to give the impression of geometric detail in a model using an image. For example, a photograph of a brick wall may be used as a texture map on a planar surface to avoid modeling the complex surface detail of the brick. However, if the lighting in the synthetic environment where the texture map is used is different from the lighting the texture map was captured under, the resulting rendering will appear incorrect and unrealistic. Worse yet, when the texture is blended with the calculated lighting of a geometric surface, the resulting rendering will look very flat and smooth to the viewer.

Bump mapping [Blinn 78] is one proposed solution to this problem where the surface normals of underlying geometry are allowed to vary per texture element (texel). Introducing variations in the surface normals causes the lighting method to render the surface as though it had local surface variations instead of just a smooth surface. As a result, when the light is moved around the object, highlights appear due to the bump map and the surface appears to be rough, grooved or similarly modified as desired. Bump maps can be either hand modeled or, more typically, calculated procedurally. Creating a bump map to be used with real world textures from photographs is generally difficult. Methods have been developed that attempt to automatically generate bump maps from a set of input images under known light directions [Rushmeier 97]. These methods have difficulty with generating bump maps for objects with large surface variations that cause self-shadowing and intra-object interreflections. In addition, current bump map rendering techniques do not render shadows due to surface variation or brightened regions due to interreflections.

In contrast, our method is an image-based technique that requires no modeling of complex geometry or bump maps. The input data required is a set of images of the desired object to be used as a texture, each one under illumination from a different known direction, all captured from the same view point. The input light

directions need not be uniformly spaced in the hemispherical set of possible light directions. We choose to represent the variation in surface color for each pixel independently with a simple biquadratic polynomial. Although approximate, this representation is compact and allows fast color reconstruction during rendering. One implementation of our method uses a polynomial to approximate the luminance of each texel, keeping the chromaticity constant. The result of our method is a texture map that properly reproduces the effects of variations in the illuminant direction relative to the object, whether due to the surface orientation of the texture-mapped object, or to changing of the location of the source. Intensity and color variations that are also due to self shadowing, sub-surface scattering and interreflections are also captured and modeled by PTMs. Figure 1 compares our method to conventional texture mapping as lighting varies. Renderings using our method are very realistic, and require little or no user input once the input images are acquired.

## 2    Background and Previous Work

The characterization of surface reflectance properties is essential to achieving photorealistic renderings. The Bidirectional Reflectance Distribution Function [Nicodemus 77] characterizes the color of a surface as a function of incident light $(\Theta_i, \Phi_i)$ and exitant view $(\Theta_e, \Phi_e)$ directions.

$$BRDF(\Theta_i, \Phi_i, \Theta_e, \Phi_e, \lambda) \qquad (1)$$

The BRDF is the ratio of the reflected intensity in the exitant direction to the incident energy per unit area along the incident direction. It does contain a dependence on wavelength, $\lambda$, but in practice this is often approximated by independent BRDFs per color channel. [Marschner 99] presents an efficient method of collecting BRDFs from samples of curved surfaces with uniform reflectance properties. There have been a large number of techniques developed to accurately and compactly represent the 4D (per discrete wavelength) BRDF. These include linear basis functions such as spherical harmonics [Cabral 87], [Sillion 91], [Wong 97], physically based analytic models [He 91], [Stam 99], and empirical models [Phong 75], [LaFortune 97].

[Dana 99] defines the Bidirectional Texture Function (BTF) by allowing the BRDF to vary spatially across a surface parameterized by $u,v$.

$$BTF_{r,g,b}(\Theta_i, \Phi_i, \Theta_e, \Phi_e, u, v) \qquad (2)$$

The BTF as defined by [Dana 99] does not actually store the ratio of exitant to incident energy like the BRDF. Instead, the BTF captures the pre-integrated lighting condition for a particular light source. They provide samples of the BTF from photographs, but face two difficulties due to the high dimensionality of the model. First, each photograph of a texture patch can be seen spanning $u,v$, but only point sampling the remaining four dimensions. Numerous photographs will be required to adequately sample this space. Second, camera pose must be accurately calibrated to allow measurements across the viewing dimensions. We avoid these difficulties by holding two of these dimensions constant, namely the exitant direction. Each photograph now becomes a sample of a 2-dimensional space, and the need for camera calibration is avoided entirely since the viewpoint does not change.

This is similar to the approach taken by both [Debevec 00] and [Georghiades 99] where a per pixel reflectance model is acquired for a static scene, in particular human faces for those references. In our work we advocate polynomial models for these reflectance functions and their application in real-time rendering as texture maps. The per pixel reflectance maps that we collect have the following dependence,

$$I_{r,g,b}(\Theta_i, \Phi_i, u, v) \qquad (3)$$

namely two spatial coordinates $u,v$ and two parameters, $\Theta_i, \Phi_i$ encoding the direction of the incident illumination. By not including the exitant direction dependence, we sacrifice the ability to capture view dependent effects such as specularity, but retain the ability to represent arbitrary geometric shadowing and diffuse shading effects across a surface. For the remainder of the paper, we assume that the surfaces being photographed are either diffuse, or their specular contributions have been separated out through the use of polarizers on both the light sources and cameras [Debevec 00]. Although the acqusition methodology we will present is limited to diffuse objects, PTMs can be used to render specular as well as diffuse effects. This is described in Section 3.5.

Other image-based relighting methods have been developed that also allow a scene to be rendered under novel lighting conditions, based on a set of input images acquired under known illuminants. These methods take advantage of the linearity of light to generate output renderings. [Nimeroff 94] and [Teo 97] showed how basis images can be linearly combined to generate rerendered synthetic scenes. [Nishino 99] describes a method where input range and color images are compressed in eigenspace. Images under novel lighting conditions are generated by interpolating in eigenspace and subsequent mapping from eigenspace into image space. [Wood 00] applies surface light fields inferred from photographs and laser range scans to generate view dependent effects. [Georghiades 99] applies image-based relighting to real human faces, adding the constraint that the surface reflectance is Lambertian to eliminate the requirement that the input lighting conditions are known. [Debevec 00] is able to render faces with more general reflectance functions after acquiring a large number of images under known lighting. Employing high dynamic range imaging, this method is extended to render arbitrary objects under novel illumination conditions. [Wong 97] creates a 6D image-based representation, an extension to [Levoy 96][Gortler 96], to render objects with novel lighting from new viewpoints. [Epstein 95] points out that low dimensional lighting models are adequate to model many objects, a result confirmed in this work as well as [Ramamoorthi 01].

## 3    Polynomial Texture Maps

### 3.1  Photographic Acquisition of PTMs

As done by [Debevec 00], [Georghiades 99], [Epstein 96] and in the field of photometric stereo in general, we collect multiple images of a static object with a static camera under varying lighting conditions. Figure 2 shows two devices we have constructed to assist with this process. The first is a simple once-subdivided icosahedral template that assists in manually positioning a light source in 40 positions relative to a sample.
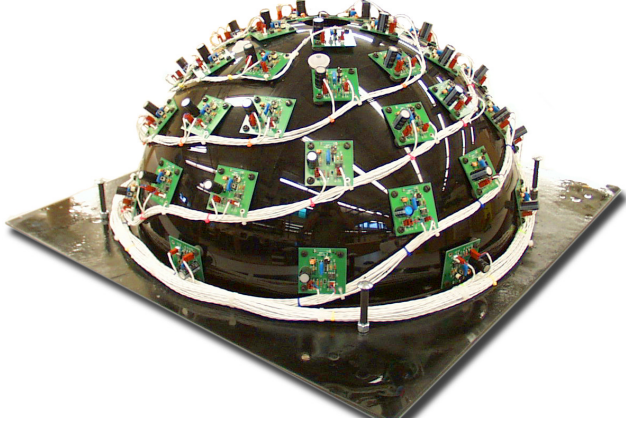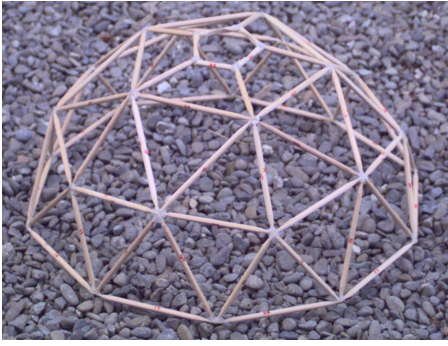
$$L(u,v;l_u,l_v) = a_0(u,v)l_u^2 + a_1(u,v)l_v^2 +$$
$$a_2(u,v)l_ul_v + a_3(u,v)l_u + a_4(u,v)l_v + a_5(u,v) \quad (5)$$

where $(l_u,l_v)$ are projections of the normalized light vector into the local texture coordinate system $(u,v)$ and $L$ is the resultant surface luminance at that coordinate. The local coordinate system is defined per vertex, based on the normal and on the tangent and binormal derived from the local texture coordinates. Coefficients $(a_0$-$a_5)$ are fit to the photographic data per texel and stored as a spatial map referred to as a Polynomial Texture Map. Given N+1 images, for each pixel we compute the best fit in the $L_2$ norm using singular value decomposition (SVD) [Golub 89] to solve the following system of equations for $a_0$-$a_5$.

$$\begin{bmatrix} l_{u0}^2 & l_{v0}^2 & l_{u0}l_{v0} & l_{u0} & l_{v0} & 1 \\ l_{u1}^2 & l_{v1}^2 & l_{u1}l_{v1} & l_{u1} & l_{v1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{uN}^2 & l_{vN}^2 & l_{uN}l_{vN} & l_{uN} & l_{vN} & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_5 \end{bmatrix} = \begin{bmatrix} L_0 \\ L_1 \\ \vdots \\ L_N \end{bmatrix} \quad (6)$$

Note that the SVD needs to be computed only once given an arrangement of light sources and then can be applied per pixel. The quadratic model proposed in Eq. 5 provides only an approximate fit to the observed color values. Note however that the resultant smoothing that occurs due to this approximation manifests itself only across the space of light directions and does not introduce any spatial blurring. This light space smoothing can have the effect of muting sharp specularities, softening hard shadows and essentially changing point light sources to area lights. However, arbitrarily high spatial frequencies in the original source photographs are preserved. Furthermore, we have verified that the general shape of the function described by the input data is well preserved. We have computed the root mean square error over all the pixels, and obtained a maximum error of roughly 10 per 8-bit color channel for typical examples such as the seeds shown in Figure 1 using the RGB PTM described below. Figures 5a-b and 6a-b show examples of a source photograph and a PTM reconstruction for the same light direction. Note the minimal loss in quality.

The representation just described is called an LRGB PTM since it explicitly separates and models luminance per texel. We have found that representing each color channel directly with a biquadratic polynomial is useful as well, especially for applications where we are modeling variations of pixel color due to other parameters besides incident lighting direction. We call this format an RGB PTM and a specific application is the depth of focus PTMs described in Section 4.2.

## 3.3 Scale and Bias

The results of the fitting operation described above are six floating-point coefficients per texel. We would like to store these as 8 bit integers for evaluation speed and so that all polynomial coefficients can be stored compactly. This is a non-trivial problem since there are typically several orders of magnitude difference between high and low order coefficients. To eliminate this problem we store 6 scale ($\lambda$) and bias ($\Omega$) values with each PTM, one for each coefficient. During reconstruction these values are applied to the stored 8 bit coefficients, $a_i'$, to recover their final values $a_i$:

$$a_i = \lambda(a_i' - \Omega) \quad (7)$$



**Figure 2:** Two devices for collecting PTMs.

Although very simple, this method is capable of achieving good results, such as the archeological samples shown later in the paper. The second device allows fully automated acquisition of 50 source images, each illuminated with an individual strobe light source. In both cases the camera (not shown) is mounted in the apex of the dome and samples are placed on the floor. In this manner multiple registered images are acquired with varying light source direction. Note that since the camera is fixed we avoid the need for any camera calibration or image registration.

## 3.2 Polynomial Color Dependence

Interpolating these input images to create textures from arbitrary light directions would be very costly both in memory and bandwidth. For each texel in our texture map we would be required to store a color sample for each input light position. One source of redundancy among these images is that the chromaticity of a particular pixel is fairly constant under varying light source direction; it is largely the luminance that varies. We take advantage of this redundancy by computing an unscaled color per texel $(R_n(u,v),G_n(u,v),B_n(u,v))$ that is modulated by a luminance model, L(u,v) again dependent on the texel:

$$R(u,v) = L(u,v)R_n(u,v);$$
$$G(u,v) = L(u,v)G_n(u,v); \quad (4)$$
$$B(u,v) = L(u,v)B_n(u,v);$$

We prefer this simple but redundant representation over color spaces such as LUV and $YC_bC_r$ due to the low cost of evaluation, although we have implemented the method in these color spaces as well. For diffuse objects, we have also found the dependence of luminance on light direction to be very smooth, even for very textured objects with high *spatial* frequencies. We choose to model this dependence with the following biquadratic per texel:

## 3.4 Hardware Implementations

PTMs were specifically designed to be implemented in VLSI. It is well known that fixed point operations generally require a smaller gate count than their floating point equivalents when the operands need limited precision. Similarly, multiplication and addition are straightforward and yield compact designs compared to the operations of division, square root, exponentiation, etc. It follows that evaluating low order polynomials consisting of fixed point coefficients and arguments can be done at low hardware cost. In particular, the per-pixel costs involved in evaluating the PTM for a given light direction consist of 11 multiplies and 5 adds[2], each of which can be done at low precision fixed point in parallel. This low complexity has associated speed advantages that allow PTM evaluations to be performed at high pixel fill rates.

We have developed an interactive software viewer with Intel MMX™ optimizations for interacting with PTMs directly. SIMD MMX™ multiply-add instructions allow four PTM coefficient terms to be evaluated simultaneously. We have also created an implementation that makes use of OpenGL hardware with extensions. Current consumer graphics cards have sufficient functionality in programmable texture operations and programmable vertex processing to perform our algorithm in hardware. The parameters $(l_u, l_v)$ are calculated in the vertex-processing pipeline, scan converted, and passed to the texture stage via the color channels. Two texture maps are used to store the coefficients of the polynomial in Eq. 5. The polynomial is evaluated per texel in the programmable texture stage.

## 3.5 Converting Bump Maps to PTMs

An alternative to photographically producing PTMs with the methods described above is to use existing bump maps and convert them into PTMs. This provides a method for rendering bump maps with any graphics hardware that renders PTMs directly. The approach we have developed uses lookup tables and is capable of converting bump maps as they are read from disk. We compute a single PTM from the bump map, which will be evaluated twice, first using the light direction, L, to yield the diffuse contribution and second using the halfway vector H (between the light source and the viewer) and exponentiation to yield the specular contribution.

Starting from the Phong Lighting equation:

$$I = I_a k_a + I_d k_d (N \cdot L) + I_s k_s (N \cdot H)^n \qquad (8)$$

we note that the diffuse and specular contributions at a texel are dependent exclusively on dot products which involve the normal $N$ at that particular location. Let us define a general function $F(\ )$ as the dot product between a surface normal vector $N$ and general normalized vector described by $(l_u, l_v)$:

$$F(l_u, l_v) = l_u N_u + l_v N_v + \sqrt{1 - l_u^2 - l_v^2}\, N_w \qquad (9)$$

The goal is to approximate this function with the biquadratic polynomial $L(l_u, l_v)$ introduced in Eq. 5. This means that we need to minimize the quantity

$$\int_{-1}^{1}\int_{-1}^{1} (L(l_u, l_v) - F(l_u, l_v))^2 dl_u dl_v \qquad (10)$$

which is the square of the continuous $L_2$ norm. Note that for simplicity, we are minimizing over the domain $[-1,1]^2$ even though we are only interested in the smaller domain $C = \{l_u, l_v : l_u^2 + l_v^2 <= 1\}$ of the unit circle. The reason is that the natural basis functions for domain $C$ are spherical harmonics [Born 80] which involve the computation of trigonometric functions. In order to minimize the expression above, we first transform the polynomial basis $B = \{1,\ l_u,\ l_v,\ l_u l_v,\ l_u^2,\ l_v^2\ \}$ into a basis $B'$ which is constructed to be orthonormal on $[-1,1]^2$.

$$B' = \{\frac{1}{2}, \frac{\sqrt{3}}{2}l_u, \frac{\sqrt{3}}{2}l_v, \frac{3}{2}l_u l_v, \frac{\sqrt{45}}{4}l_u^2 - \frac{\sqrt{45}}{12}, \frac{\sqrt{45}}{4}l_v^2 - \frac{\sqrt{45}}{12}\} \qquad (11)$$

By using fundamentals from Approximation Theory [Watson 80], we know that the coefficients $a'_i$ of the best approximant $L'$ can be computed directly as values of integrals:

$$a'_i = \int_{-1}^{1}\int_{-1}^{1} L'_i(l_u, l_v) F(l_u, l_v) dl_u dl_v \qquad (12)$$

where $L'_i$ is the $i$-th basis polynomial of $B'$. The final step then is to transform the $a'_i$ back into $a_i$, the PTM coefficients associated with a particular projected surface normal.

$$a_0 = \frac{1}{2}a_0' - \frac{\sqrt{45}}{12}(a_5' + a_4')$$
$$a_1 = \frac{\sqrt{3}}{2}a_1'$$
$$a_2 = \frac{\sqrt{3}}{2}a_2'$$
$$a_3 = \frac{3}{2}a_3' \qquad (13)$$
$$a_4 = \frac{\sqrt{45}}{4}a_4'$$
$$a_5 = \frac{\sqrt{45}}{4}a_5'$$

This computation only needs to be done once for all bump maps, and the results put into a lookup table which will convert bump map normals into PTMs. In summary the steps involved are:

**1) Precompute lookup table**. We sample the space of normals on the hemisphere by subdividing the range of latitude and longitude uniformly. Here $i$ denotes the index in the latitudinal direction and $j$ denotes the index in the longitudinal direction. For each normal $N_{i,j}$ we compute the polynomial coefficients as described above and store them in a look-up table. Note that this step is independent of the particular bump map and thus needs to be performed only once as a preprocess.

**2) Convert height field**: Given a height field bump map and user-defined scale factor, we compute surface normals per texel [Blinn 78].

**3) Convert normals to PTM**: Using the longitude and latitude of the surface normal as an index into the precomputed lookup table, we recover the PTM coefficients per texel. Here we make use of the fact that the polynomial is linear in its coefficients and hence we can perform six bilinear interpolations to compute the final polynomial.

---

[2] 8 multiplies and 5 adds for the luminance polynomial evaluation and 3 more multiplies to get color per pixel.
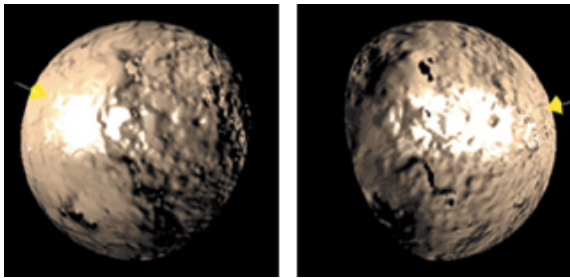
**Figure 3:** Bump Mapping with PTMs, two light positions.

**4) Render the PTM**: For each pixel we have computed a polynomial that approximates the dot product of the bump map normal with any other given normalized vector. That means that we can render the bump map by evaluating this polynomial for light vector (diffuse) as well as halfway vector (specular). This is illustrated in Figure 3. It shows two images of a bump map textured onto a sphere. The original data is a height field capturing data simulating the surface of Mars. The sphere is lighted with specular and diffuse terms under varying light positions.

## 3.6 PTM Filtering

One of the difficulties with using traditional bump maps to display surface detail is filtering them to match a particular sampling rate. Approximate solutions are discussed in [Kilgard 00]. The naive approach using mip-maps winds up smoothing the effective surface, removing the bumps themselves [Schilling 97]. The function that we would like to integrate over is the light radiated from a bumpy patch, instead of surface perturbations themselves. Since PTMs are image-based representations, we can achieve this effect by mip-mapping the polynomial coefficients ($a_0$-$a_5$) directly. At first this may seem odd, but a glance at Eq. 5 reveals that the light dependence is linear in the polynomial coefficients. Integrating over a patch $\Omega$ with $n$ samples, we have:

$$\frac{1}{n}\sum_{i,j\in\Omega}L_{r,g,b}(a_{0-5}(u_i,v_j)) = L_{r,g,b}(\frac{1}{n}\sum_{i,j\in\Omega}a_{0-5}(u_i,v_j))$$

(14)

This states that accumulating polynomial coefficients over a patch and then evaluating the sum is equivalent to accumulating the colors each texel contributes when independently evaluated. Not only does this allow us to mip-map the coefficients directly, but nonisotropic filtering techniques that yield improved image quality such as footprint assembly [Schilling 96] can be supported.

## 3.7 Anisotropic / Fresnel / Off-Specular

The light vector dependent PTMs that we have described can be combined with existing per-vertex or per-pixel lighting hardware to generate a variety of useful effects. One method is to modulate the specular component of standard lighting computations by the result of evaluating the PTM. The existing Phong lighting equation becomes:

$$I = I_a k_a + I_d k_d (N \cdot L) + I_s k_s (N \cdot H)^n PTM(u,v,l_u,l_v) \qquad (15)$$

In the trivial case where the PTM varies spatially but is constant for all incident light directions this is equivalent to specular maps or gloss maps [Blythe 99]. The dependence on light direction that is not available with standard specular maps could be used to

reproduce effects such as shadowing where texels do not exhibit specular highlights due to shadowing by small scale surface variations.

Many surfaces exhibit increased specularity for low angles of incidence due to decreases in the apparent roughness [Lafortune 97], a case of off-specular reflections. This can be reproduced using our technique by modulating a Phong specular lobe with a PTM whose magnitude increases as the incident angle approaches grazing.

Incident illumination dependence can also be used to approximate Fresnel effects. Fresnel effects are important for many surfaces, such as metals and glass where reflectance increases greatly for grazing incident angles. The reflectance of glass and other dielectrics are low when the incident illumination direction is near the surface normal. Metals also exhibit Fresnel reflectance, including wavelength dependence [Hall 89]. The color of reflected light changes as the incident angle changes due to reflectance variation for different wavelengths. These dependencies can be approximated by the polynomial function stored at each texel. If color changes over varying incident angles need to be captured, then RGB PTMs may be necessary. The combination of the illumination dependent effects that we described can all be stored in a single PTM. In addition, since texel polynomials in the PTM are independent, these properties can vary across the surface. A PTM can represent different materials in a single texture map.

Anisotropic materials can also be modeled using PTMs. Techniques to render anisotropy such as [Banks 94], [Stalling 97] and [Heidrich 99] define normal planes for surface points aligned with the direction of anisotropy. The projection of the incident light direction into the normal plane is then used in the lighting calculations. This then drops off as the incident light direction moves away from the preferred direction. For our technique each texel in the PTM stores the magnitude of the light vector projected onto the normal plane as a function of incident light direction. When the incident light direction is aligned near the preferred direction of anisotropy the evaluated PTM polynomial has a large magnitude. The result of evaluating the PTM is then modulated with calculated specular lighting so that specular highlights occur only in regions where the incident light direction aligns with the direction of anisotropy. Our technique allows us to render anisotropic surfaces under perspective views with local light sources, and spatially variant BRDFs in hardware. Figure 4 shows an example of a disc rendered with a synthetic anisotropic PTM.
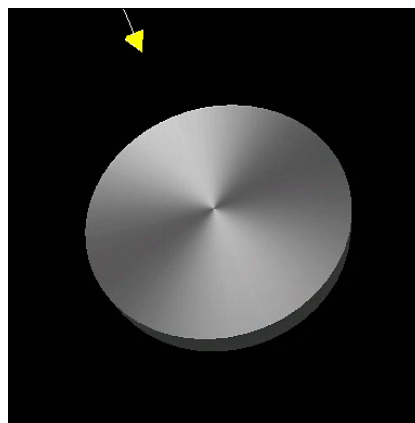


**Figure 4:** Anisotropic Disc

**Figure 5:** Specular enhancement: (A) Original Photograph (B) Reconstruction from PTM. (C) An image computed by extracting surface normals for each pixel and applying a specular lighting model per pixel. (D) Highlights computed in (C) added to (B). Light direction is the same for all four images. This artifact is a 4000 year old neo-Sumerian tablet.

## 4 2D Applications of PTMs

Although PTMs were developed as an extension to texture mapping, we have found them to have a number of uses in 2-dimensional applications where images are commonly employed. The next section discusses new contrast enhancement mechanisms and their application in the study of ancient writings. Additionally we demonstrate photographic depth of focus effects using PTMs and the mapping of short image sequences to PTMs.

## 4.1 PTMs for Contrast Enhancement

We have found that PTMs provide a valuable representation for the study and archiving of ancient artifacts, in particular early clay writings [Malzbender 00]. PTMs allow interactive control of lighting conditions that enable greatly increased perception of surface structure compared to photographs of these artifacts. In addition, we have developed three novel contrast operators, two of which rely on surface normals extracted from the coefficients themselves. Assuming a diffuse surface being photographed under the variable lighting apparatus shown in Figure 2, we can extract an estimate of the surface normal per pixel by solving for the

**Figure 6:** Diffuse gain shown on a 3000 year old Egyptian funerary statuette: (A) Original photograph. (B) Reconstruction from PTM. (C) Diffuse gain—a transformation that exaggerates the diffuse reflectance properties by a gain factor (g = 1.9 here), keeping the surface normal estimate constant. These images have identical light directions and intensities.

maximum luminance of Equation 5. Setting $\dfrac{\partial L}{\partial u} = \dfrac{\partial L}{\partial v} = 0$, we arrive at:

$$l_{u0} = \frac{a_2 a_4 - 2a_1 a_3}{4a_0 a_1 - a_2^{\,2}} \qquad (16)$$

$$l_{v0} = \frac{a_2 a_3 - 2a_0 a_4}{4a_0 a_1 - a_2^{\,2}} \qquad (17)$$

for the values of the projected surface normal $(l_{u0}, l_{v0})$ that maximizes the biquadratic of Eq. 5. The full estimated surface normal is then simply:

$$\bar{N} = (l_{u0}, l_{v0}, \sqrt{1 - l_{u0}^{\,2} - l_{v0}^{\,2}}) \qquad (18)$$

Photometric stereo methods from computer vision would provide an alternate approach to this stage of surface normal recovery [Rushmeier 97] and would also be affected by the presence of self-shadowing, as is our method.

**Method 1**: Specular enhancement. For every texel $(u,v)$, the normal vector can be used in a lighting equation (such as Eq. 8) to add either diffuse or specular shading effects to the image. Simulation of specularity is particularly effective at enhancing the perception of surface shape. We are free to change the properties of the surface (specularity $k_s$, specular exponent $n$) and simulated light source (position, color) under interactive control. Our method is demonstrated in Figure 5 on a 4000 year old neo-Sumerian tablet. We have effectively modified the reflectance properties of the tablet to enhance perception of surface shape, making the inscription more readable.

**Method 2**: Diffuse gain. For diffuse objects, the original photographed surfaces typically exhibit a gently varying change in surface intensity across light direction that we have fit with a 2-dimensional convex parabola. The Gaussian curvature of this parabola (its second spatial derivative) can be increased arbitrarily by a gain factor $g$ by computing new luminance coefficients using the following transformation:

**Figure 8:** A single PTM can be used to provide views at continuously variable focus depths.

$$a_0' = ga_0$$
$$a_1' = ga_1$$
$$a_2' = ga_2 \qquad\qquad (19)$$
$$a_3' = (1-g)(2a_0 l_{u0} + a_2 l_{v0}) + a_3$$
$$a_4' = (1-g)(2a_1 l_{u0} + a_2 l_{v0}) + a_4$$
$$a_5' = (1-g)(a_0 l_{u0}{}^2 + a_1 l_{v0}{}^2 + a_2 l_{u0} l_{v0}) +$$
$$(a_3 - a_3')l_{u0} + (a_4 - a_4')l_{v0} + a_5$$

This keeps the luminance maximum at the same location, namely $(l_{u0}, l_{v0})$, thereby retaining the surface normal direction. It also maintains the luminance at $(l_{u0}, l_{v0})$ retaining the surface albedo and color in this direction. However, the directional sensitivity of the surface to light is increased by this transformation, enhancing the reflectance properties of the sample. Figure 6 shows the method on a 3000 year old funerary statuette called an *ushabti* from ancient Egypt. Made out of a glazed frit material (faience), its purpose was to carry out work on behalf of the deceased in the netherworld. Note the enhanced definition of the inscription.

**Method 3**: Light direction extrapolation. Light source direction is a normalized quantity encoded by the projected light direction $l_u, l_v$. Physically realizable light sources are limited to

$$\sqrt{l_u{}^2 + l_v{}^2} \le 1 \qquad\qquad (20)$$

since the light direction is described by a normalized vector with three real components, as shown in Eq. 18. However, this constraint need not be applied when evaluating PTMs, allowing one to extrapolate the reflectance model beyond what is geometrically achievable. This extrapolation does not have an exact physical analog, but it provides lighting that is more oblique than any used in the original source images, providing enhanced contrast. An example is shown in Figure 7 on a trilobite fossil.
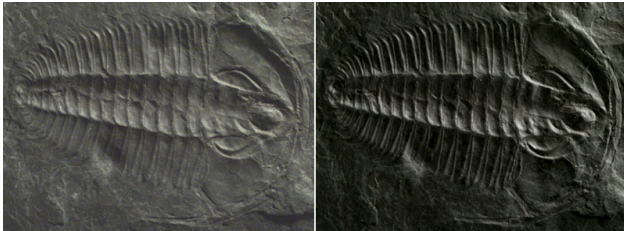


**Figure 7:** Light Direction Extrapolation. Left: Source photo. Right: Reconstruction with extrapolated light source direction.

## 4.2 Depth of Focus / Time Varying PTMs

The free parameters $l_u, l_v$ in Equation 5 can be used to interpolate between a number of related images. One useful application is representing a static scene under varying focus conditions. Figure 8 demonstrates this capability. First we photographed the scene 6 times across a range of focus depths. Next we assigned each image a value spanning $(-1.0 < l_u < 1.0)$ holding $l_v$ constant, and fit Eq. 5 for each color channel in each pixel. The resultant RGB PTM now offers a continuously variable depth of field. Note that in this case we have only used one of the two available free variables, $l_u$, hence fitting only a univariate quadratic PTM. We have found it useful to tie the other variable, $l_v$, to the camera aperture, allowing control of the spatial extent that is in focus as well as the specific depth that is maximally focused. This capability allows the graphics artist to specify the focus parameters of an image during final composition, not when it is originally photographed. We will see shortly that the additional storage required for this capability is not substantial compared with an individual conventional image.

This capability is not restricted to images that vary by focus parameters. Shown in Figure 9 are 3 reconstructions from a PTM whose input was a sequence of 40 images taken throughout the course of the day as the tide was going out. This RGB PTM is also computed as a univariate quadratic.
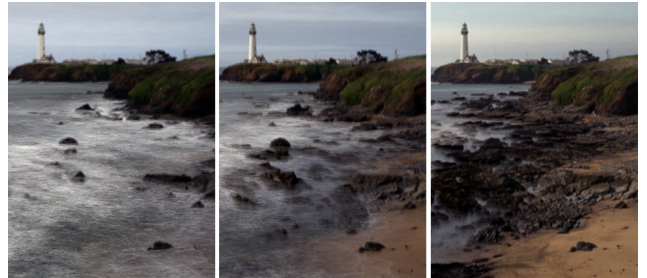


**Figure 9:** Three reconstructions from a single PTM. Input images from Sundial California published by Aladdin Systems Inc. Copyright © 1996 John M. Neil. Used with permission.

## 5 Compression / Palletization

The PTMs discussed in this paper are either LRGB or RGB PTMs. For LRGB PTMs we store 9 bytes per pixel (3 colors plus 6 polynomial coefficients) and for RGB PTMs we store 18 bytes per pixel (6 polynomial coefficients per color channel). We have found that both of these representations are highly compressible, due to both spatial correlation and correlations between byte-planes. Compression methods developed for multispectral images can be directly applied.

Our first compression method is to simply palletize the PTMs much as one palletizes images, by computing a color lookup table. Using K-means clustering we compute a lookup table for an LRGB PTM that includes the 6 polynomial coefficients per entry. We have found that a 256 entry table is sufficient to be indistinguishable from the original. This representation allows a one byte value to index the lookup table, with the 3 remaining bytes of color all packed into a 32 bit field. Since 24 bit images are often packed into a 32 bit format anyway, the size of this representation compared to a conventional image grows only by a small amount, the size of the lookup table (1.5 K). The analogous representation for RGB PTMs requires 18 fields (bytes) in the lookup table and we have found that 12 bit indices yield indistinguishable results compared to the original PTMs. Note that this method allows random access and individual PTM texels can be evaluated independently.

Although decoding is simple and fast with this lookup table approach, we have gotten better results by decorrelating the polynomial coefficients into planes and applying JPEG or JPEG-LS to the results [Motta 00]. Visual inspection of the polynomial coefficient planes and RGB planes makes it apparent that strong dependencies and correlations exist between these planes, and that further room for compression exists. Inter-plane prediction is employed to reduce this dependency. Given a reference plane, pixel values in planes being encoded are estimated, resulting in residual images. These are further compressed by intra-plane methods that estimate pixels values based on their causal neighbors. Table 1 below overviews the results for the lossless and perceptually lossless JPEG-LS cases for the intra-pane prediction. Note that this includes both color and polynomial coefficients.

| Dataset | lossless | loss = 1 | loss = 2 | loss = 4 |
|---|---|---|---|---|
| seeds | 30.3 bits | 19.2 bits | 15.7 bits | 12.6 bits |
| ushabti | 31.3 bits | 19.8 bits | 15.9 bits | 12.3 bits |
| tablet | 33.8 bits | 21.3 bits | 16.9 bits | 13.0 bits |
| lighthouse | 14.6 bits | 8.61 bits | 6.33 bits | 4.27 bits |
| trilobite | 34.0 bits | 21.6 bits | 17.1 bits | 13.0 bits |
| focus | 20.5 bits | 12.3 bits | 9.30 bits | 5.67 bits |
| **average** | **27.4 bits** | **17.1 bits** | **13.5 bits** | **10.1 bits** |

**Table 1:** PTM Compression: Number of bits per texel for varying amounts of imprecision tolerated in each color channel during reconstruction (out of 256 levels). These levels of loss are all imperceptible. Empirically we have found that visible artifacts start appearing at approximately 4 bits per pixel.

# 6   Conclusions

We have presented a method that requires only images to generate high quality photorealistic renderings of a textured surface. Our method captures light dependent effects, whether they are due to changes in the illuminant direction, or surface orientation of the texture mapped object. It can render changes in brightness due to shadows and indirect lighting that cannot be reproduced with existing bump mapping hardware, or Phong illuminated geometric objects. Our technique could be integrated into non-programmable graphics hardware with minor additional hardware requirements. It can be efficiently implemented using modern programmable shading hardware. Sample PTMs and an interactive browser program for viewing them are available for downloading at http://www.hpl.hp.com/ptm .

PTMs were originally developed to model the dependence of surface color on a parameterization of the light direction for interactive applications. We have shown extensions that allow Fresnel, anisotropic, off-specular, depth of focus and contrast enhancement effects to be realized. Several unexplored uses remain. One possibility is to allow rudimentary modeling of occlusion effects due to camera or object motion by indexing a PTM by the view vector instead of the light direction vector. Another is the application of PTMs to model surface opacity in addition to color. Both of these methods would allow modeling geometry in image-based manner. Lastly, the method presented here could be generalized to higher order polynomials or non-polynomial basis functions.

## References

[Banks 94] Banks, D.C., "Illumination in Diverse Codimensions", *Computer Graphics (SIGGRAPH 94 Proceedings)*, July 1994, pp. 327-334.

[Blinn 78] Blinn, J.F., "Computer Display of Curved Surfaces", Ph.D. Thesis, University of Utah, 1978.

[Blythe 99] Blythe, D., McReynolds, T., "Lighting and Shading Techniques for Interactive Applications", *Course Notes (Siggraph 99 Course 12)*, August, 1999, p. 101.

[Born 80] Born, Max, Wolf, Emil, "Principles of Optics", 6th edition, Appendix VII, Cambridge University Press, Cambridge, 1980.

[Cabral 87] Cabral, B., Max, N., Springmeyer, R., "Bidirectional Reflection Functions from Surface Bump Maps", *Computer Graphics (SIGGRAPH 87 Proceedings)*, July 1987, pp. 273-281.

[Dana 99] Dana, K., Van Ginneken, B., Nayar, S., Koenderink, J., "Reflectance and Texture of Real-World Surfaces", *ACM Transactions on Graphics*, Vol. 18, No. 1, January 1999, pp. 1-34.

[Debevec 00] Debevec, P., Hawkins,T., Tchou, C., Duiker, H., Sarokin, W., Sagar, M., "Acquiring the Reflectance Field of a Human Face*", Computer Graphics (SIGGRAPH 2000 Proceedings)*, July 2000, pp. 145-156.

[Epstein 95] Epstein, R., Hallinan, P., Yuille, A., "5 +/- 2 Eigenimages Suffice: An Empircal Investigation of Low-Dimensional Lighting Models, *IEEE Workshop on Physics-Based Vision*: 108-116, 1995.

[Epstein 96] Epstein, R., Yuille, A.L., Belhumeur, P.N., "Learning Object Representations from Lighting Variations", Object Representation in Computer Vision II Workshop, ECCV96, April 1996, pp.179-199.

[Georghiades 99] Georghiades, A., Belhumeur, P., Kriegman, "Illumination-Based Image Synthesis: Creating Novel Images of Human Faces Under Differing Pose and Lighting", *IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes*, 1999, pp. 47-54.

[Golub 89] Golub, G., van Loan, C., "Matrix Computations", Johns Hopkins University Press, Baltimore, 1989.

[Gortler 96] Gortler, S., Grzeszczuk, R., Szeliski, R., Cohen, M., "The Lumigraph", *Computer Graphics (SIGGRAPH 96 Proceedings),* August 1996, pp. 43-54.

[Hall 89] Hall, R. *Illumination and Color in Computer Generated Imagery*, Springer-Verlag New York Inc., New York, 1989, pp. 193-197.

[He 91] He, X., Torrance, K., Sillion, F., Greenberg, D., "A Comprehensive Physical Model for Light Reflection", *Computer Graphics (SIGGRAPH 91 Proceedings),* July 1991, pp.175-186.

[Heidrich 99] Heidrich, W., Seidel, H., "Realistic, Hardware-accelerated Shading and Lighting", *Computer Graphics (SIGGRAPH 99 Proceedings),* August 1999, pp.171-178.

[Kilgard 00] Kilgard. M.,"A Practical and Robust Bump-mapping Technique for Today's GPUs", Game Developers Conference (GDC) 2000: Advanced OpenGL, also available at nvidia.com.

[Lafortune 97] Lafortune, E., Foo, S.-C., Torrance, K., Greenberg, D., "Non-Linear Approximation of Reflectance Functions", *Computer Graphics (SIGGRAPH 97 Proceedings),* August 1997, pp. 117-126.

[Levoy 96] Levoy, M., Hanrahan, P., "Light Field Rendering", *Computer Graphics (SIGGRAPH 96 Proceedings),* August 1996, pp. 31-42.

[Malzbender 00] Malzbender, T., Gelb, D., Wolters, H., Zuckerman, B., "Enhancement of Shape Perception by Surface Reflectance Transformation", Hewlett-Packard Technical Report HPL-2000-38, March 2000.

[Marschner 99] Marschner, S., Westin, S., Lafortune, E., Torrance, K., Greenberg, D., "Image-Based BRDF Measurement Including Human Skin", *Rendering Techniques 99: Proceedings of the 10th Eurographics Workshop on Rendering*, June 1999, ISBN 3-211-83382-X, pp. 131-144.

[Motta 00] Motta, G., "Compression of Polynomial Texture Maps", Hewlett-Packard Laboratories Technical Report, HPL-2000-143, October 30, 2000.

[Nicodemus 77] Nicodemus, F.E., Richmond, J.C., Hsai, J.J., "Geometrical Considerations and Nomenclature for Reflectance", *U.S. Dept. of Commerce, National Bureau of Standards*, October 1977.

[Nimeroff 94] Nimeroff, J., Simoncelli, E., Dorsey, J., "Efficient Re-rendering of Naturally Illuminated Environments", *Eurographics Rendering Workshop Proceedings* 1994, pp. 359-374.

[Nishino 99] Nishino, K., Sato, Y., Katsushi, I., "Eigen-texture Method – Appearance Compression based on 3D Model", *IEEE Computer Vision and Pattern Recognition,* June 23-25 1999, Vol. 1, pp.618-624.

[Phong 75] Phong, B.-T., "Illumination for Computer Generated Images", Communications of the ACM 18, 6, June 1975, pp. 311-317.

[Ramamoorthi 01], Ramamoorthi, R. and Hanrahan, P., "An Efficient Representation for Environment Irradiance Maps", *Computer Graphics (SIGGRAPH 01 Proceedings),* August 2001.

[Rushmeier 97] Rushmeier, H., Taubin, G., Gueziec, A., "Applying Shape from Lighting Variation to Bump Map Capture", *Eurographics Rendering Workshop Proceedings* 1997, pp. 35-44, 1997.

[Schilling 96] Schilling, A., Knittel, G., Strasser, W., "Texram: A Smart Memory for Texturing", *IEEE Computer Graphics and Applications*, Vol. 16, No. 3, May 1996, pp. 32-41.

[Schilling 97] Schilling, A., "Towards Real-Time Photorealistic Rendering: Challenges and Solutions", *Proceedings of the 1997 Siggraph/Eurographics Workshop on Graphics Hardware*, Aug. 3-4, 1997, pp.7-15.

[Sillion 91] Sillion, F., Arvo, J., Westin, S., Greenberg, D., "A Global Illumination Solution for General Reflectance Distributions", *Computer Graphics (SIGGRAPH 91 Proceedings)*, July 1991, pp.187-196.

[Stalling 97] Stalling, D., Zöckler, M., Hege, H.-C., "Fast Display of Illuminated Field Lines", *IEEE Transactions on Visualization and Computer Graphics*, 3(2):118-128, 1997.

[Stam 99] Stam, J., "Diffraction Shaders", *Computer Graphics (SIGGRAPH 99 Proceedings),* August 1999, pp.101-110.

[Teo 97] Teo, P., Simoncelli, E., Heeger, D., "Efficient Linear Re-rendering for Interactive Lighting Design", Stanford Computer Science Department Technical Report STAN-CS-TN-97-60. October 1997.

[Watson 80] Watson, G.A., "Approximation Theory and Numerical Methods", A.J.Wiley & Sons, Chichester, 1980.

[Wood 00] Wood, D., Azuma, D., Aldinger, K., Curless, B., Duchamp, T., Salesin, D., Stuetzle, W., "Surface Light Fields for 3D Photography", *Computer Graphics (Siggraph 2000 Proceedings),* July 2000, pp. 287-296

[Wong 97] Wong, T., Heng, P, Or, S, Ng, W., "Image-based Rendering with Controllable Illumination", Rendering Techniques 97: *Proceedings of the 8th Eurographics Workshop on Rendering*, June 16-18, 1997, ISBN 3-211-83001-4, pp. 13-22.