Vision, Modeling, and Visualization (2010)

Relighting Spherical Light Fields with Polynomial Texture Mapping

Lisa Brückbauer[1] and Christof Rezk-Salama[2] and Andreas Kolb[1]

[1]Computer Graphics Group, Institute for Vision and Graphics (IVG), University of Siegen, Germany
[2] Mediadesign Univ. of Applied Sciences, Düsseldorf, Germany

Abstract

We present a novel image-based rendering (IBR) technique based on spherical light fields, which makes it possible to relight the captured object for arbitrary viewing positions. This approach incorporates view-dependent effects such as self-shadowing and inter-reflections. For this, we apply Polynomial Texture Maps (PTMs) to 3D objects. Once acquired, a light field representation of an object can be relit at low computation costs due to the efficiency of the PTM approach. The relighting process makes even small lighting changes visible and retains surface appearance even on a meso-scale level. Furthermore, we present a simple method to adopt specular reflections captured in the PTM to novel viewing directions.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Image-Based Rendering—

1. Introduction

Adding meso-scale visual information into image synthesis has been a research focus for some decades. Different approaches have been presented in order to prevent the explicit geometrical representation of meso-scale geometry, i.e. geometry which is too fine to render efficiently using polygons, but containing visible information like wrinkles on a face, cloth structure or engravings in stone or metal. Traditionally, techniques like texture mapping and bump mapping [Bli78] have been used to address this problem (see [AMHH08] for an overview).

Image-based rendering (IBR) techniques cope with the quest for the incorporation of meso-scale information by synthesizing images from pre-recorded sample images, which are taken from real or synthetic objects and scenes, mainly under known illumination conditions like a light stage [DHT*00]. IBR techniques are able to capture many complex visual effects like subsurface scattering or self-shadowing without adding additional complexity to the image synthesis process, which makes it also appealing to apply them to exhaustive rendering processes. However, IBR techniques do not yet provide all degrees of freedom regarding the variation of viewing positions, the change of lighting conditions (relighting) and the scene dynamic that are common to classical polygon-based rendering approaches.

Light fields, introduced to computer graphics by Levoy [LH96] and Gortler [GGSC96], focus on the synthesis of static scenes under static illumination conditions, but with a nearly unrestricted selection of the viewing position. Different setups for the representation of light fields have been proposed. Commonly, a rather large number of images are required to reduce ghosting artifacts. Additional depth information, however, can resolve some of these artifacts in case of a moderate number of acquisition view points, e.g. using depth maps [TRSKK08].

Another class of IBR techniques tackles the relighting problem. These methods are commonly applied for static scenes and for a single, fixed view point. Debevec et al. [DHT*00] introduced the reflectance function, which represents the light transfer between incident light to outgoing light with respect to a volume. Malzbender et al. [MGW01] introduced *Polynomial Texture Maps (PTMs)*, a per-pixel representation for the reflectance function, resulting in an efficient, yet view dependent relighting approach. PTMs provide variations in surface details without using explicit 3D geometry, which can be applied for product presentations, archeological applications, compositing in film and arts, materials science, geology, biology, for example.

In this paper, we introduce a novel technique for relight-

DOI: 10.2312/PE/VMV/VMV10/073-080

ing light fields using PTMs. In order to cope with the memory demands of light fields and of image-based relighting techniques, we combine the spherical light field acquisition and rendering method used by Todt et al. [TRSKK08] with Malzbender et al. 's PTM approach, which are both memory efficient. The major contributions of this paper are:

- A novel IBR technique, that allows synthesized views of arbitrarily relit objects while maintaining the ability to choose the view within 6 DOF, thus realizing view and light dependent effects such as self-shadowing and interreflections for meso-scale structures.
- An approach to apply PTMs to 3D objects for arbitrary viewing directions. Especially, we introduce a 360° light representation for PTMs.
- We introduce a simple method to adopt specular reflections captured in the PTM to novel viewing directions.

Currently, our technique is restricted in two ways: First, we apply our method only to synthetic data, which we acquire with our 3D PTM renderer. Second, we apply only directional light sources to our light fields. While the first restriction is mainly a technical issue in setting up a rather complex capturing system, the second point is a major challenge for future research.

The remainder of this paper is structured as follows. After a summary of prior work in the field (Sec. 2), we give an overview of our system in Sec. 3, including an introduction to PTMs. Sec. 4 describes the 3D PTM rendering approach and Sec. 5 explains the representation and rendering of PTM-based relightable light fields. Sec. 6 shows the results of our technique.

2. Prior Work

2.1. Light Fields

Adelson and Bergen [AB91] introduced the *plenoptic function* which represents the radiance along all rays in a region of 3D space illuminated by a static light setup. Using the fact that the radiance along a ray remains constant and restricting ourselves to viewing locations outside the object's convex hull, the *plenoptic function* is a 4D function, parametrizable using two intersection points of each ray with the convex hull. These functions have been introduced as *light fields* to computer graphics. Both initial works by Levoy and Hanrahan [LH96] and Gortler et al. [GGSC96] use regular structures to represent the rays in 3D space, i.e. camera and image planes (*2-plane parametrization*) from one major direction, yielding a non-uniform distribution of the sampled light field, especially in the case of 360° light field setup, which requires 6 pairs of camera- and image-planes. Furthermore, Gortler et al. [GGSC96] added a coarse polygonal model in order to improve the quality of radiance interpolation and to reduce the number of input samples significantly.

Spherical sampling setups try to improve the sampling uniformity and thus reduce the amount of discontinuities resulting from non-uniform sampling. *Spherical light fields*, as proposed by Ihm et al. [IPL97], use intersections with a positional sphere and a second directional sphere located at the intersection point with the positional sphere. Alternative setups have been presented by Camahort et al. [CLF98], such as the *Sphere-Sphere* parametrization, where rays are determined by intersecting the same sphere twice, and the *Sphere-Plane* parametrization. Todt et al. [TRSKK08] introduce a spherical light field representation using a parabolic mapping of rays hitting the opposite hemisphere with respect to the camera sample position. Using depth maps as implicit geometry representation, Todt et al. improved the rendering quality while reducing the required number of sample positions at the same time.

For many practical applications, unstructured light fields [BBM*01] are superior to structured setups, since only tracking of the (handheld) camera is required and no further steering to specific capturing locations. However, they are hard to use for relighting purposes, since varying lighting conditions would be acquired with different camera locations.

2.2. Image Based Relighting

Debevec et al. [DHT*00] introduced the *reflectance field*, which is the basic formulation for image-based relighting. The reflectance function is an 8D-function, determining the light transfer between light entering a volume of interest at a specific point and direction and a light leaving the volume at another point and direction. Relighting research focuses on setups for the data acquisition, the separation and replacement of illumination from the acquired data and the method for interpolating the reflectance function for re-sampling.

Debevec et al. [DHT*00] focus on relighting of a human face. They use a dense set of directional light sources and a sparse set of camera positions to reduce the amount of required data. Relighting is accomplished by acquiring the illumination environment with a mirror ball and linearly transferring this information with the reflectance function to the given viewer position. Applying a skin model and a scanned geometry of a face, the acquired reflectance function of the face can be transformed and rendered to novel view points.

Masselus et al. [MPDW03] model a 6D reflectance field for a single view point. They approximate the light field incident to the observed object using Haar-Wavelet-like illumination structured light patterns which are explicitly captured. Other acquisition approaches use indirect environment illumination from arbitrary directions [FBS05] or even fully uncontrolled lighting situations [MLP04].

Whereas many approaches use rather simple linear interpolation techniques in order to re-sample the reflectance function, Fuchs et al. [FLBS07] use a non-linear superresolution technique to reduce the interpolation problem by syn-

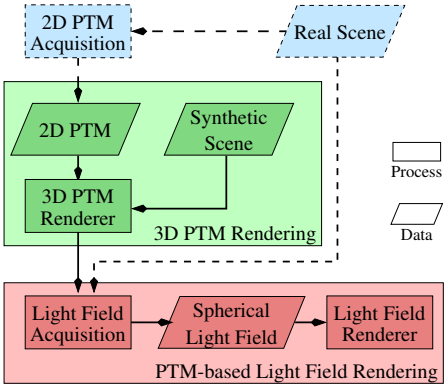

Figure 1: *Light field rendering using PTM-based relighting. Dashed components are not further discussed in this paper.*

thesizing plausible intermediate images at a much higher density in the domain of light directions.

Malzbender et al. [MGW01] also use a dense set of light directions and a single viewing position to acquire 2D surface reflectance functions. Their major goal is to capture meso-structures of objects. A per-pixel representation of the reflectance function using low-dimensional polynomials yields an efficient, yet easy way to store illumination information dependent on the two light parameters (see Sec. 3.2 for more details).

In general, all works mentioned so far have gone a similar path to Debevec et al. [DHT*00], using a fixed viewing position and varying setups for the acquisition of the reflectance function for a scene and the illumination environment. Lin et al. [LWS02] use a 3-plane parametrization for light fields, adding a light source plane to the classical 2-plane light field parametrization. This allows for the change of the viewing position using a simple linear interpolation on a per-pixel level. This work is the closest approach to ours. However, our techniques allows for seamless 360° viewer positioning and relighting using efficient techniques for consistent light field and relighting representation.

3. Spherical Light Fields With Dynamic Lighting

In this section, we give an overview of our system components involved in the acquisition and rendering of relightable light fields (Sec. 3.1). In Sec. 3.2 we briefly introduce PTMs, since we use them in different components in our systems.

3.1. System Overview

The general system setup, consisting of the two major subsystems *3D PTM Rendering* and *PTM-based Light Field Rendering*, is depicted in Fig. 1. Currently, we only support synthetic scenes, consisting of polygonal objects, which are enhanced with PTMs. Whereas the PTM acquisition from real scenes is a standard 2D process with fixed viewing position (components *2D PTM Acquisition* and *2D PTM*), we require a 3D rendering of the PTM under arbitrary viewing directions. The details of the corresponding *3D PTM Renderer* are described in Sec. 4.

Our light field rendering technique is based on spherical light fields introduced by Todt et al. [TRSKK08]. This method uses depth maps as implicit, image based geometry representation. The advantage of this approach is the high visual quality even for a relatively small number of sampling cameras. We extend the spherical light field approach by incorporating PTMs as image-based relighting information. The technical details are described in Sec. 5.

3.2. Polynomial Texture Mapping (PTM)

Polynomial Texture Maps, as introduced by Malzbender et al. [MGW01], store the reflectance information of an object while varying the position of a point light source on a hemisphere over the object. With the arrangement of known normalized light directions $(x_i, y_i, z_i)^T$ on the hemisphere, the reflectance $\mathbf{R}$ for each pixel is coded in the observation image using the x,y-components of the light direction as lighting parameters. We separate the luminance L from the chromaticity C by Eq. 1.

$$L = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{1}$$

The chromaticity channels R,G,B are assumed to be constant for each pixel, and we get the following lighting dependency:

$$\mathbf{R}(x,y) = L(x,y) \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \tag{2}$$

The luminance is then modeled as bi-quadratic polynomial

$$L(x,y) = ax^2 + by^2 + cxy + dx + ey + f. \tag{3}$$

For one camera and $N > 6$ light sources, the resulting over constrained linear system

$$\underbrace{\begin{pmatrix} x_0^2 & y_0^2 & x_0y_0 & x_0 & y_0 & 1 \\ x_1^2 & y_1^2 & x_1y_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^2 & y_N^2 & x_Ny_N & x_N & y_N & 1 \end{pmatrix}}_{=:M} \underbrace{\begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix}}_{=:\vec{\mathbf{c}}} = \underbrace{\begin{pmatrix} L_0 \\ L_1 \\ \vdots \\ L_N \end{pmatrix}}_{=:\vec{\mathbf{L}}} \tag{4}$$

can be solved for $\vec{\mathbf{c}}$ using singular value decomposition (SVD) [GVL96]. Relighting is then done by evaluating Eqs. 2 and 3 for a given light direction (x,y).

PTMs consist of 9 parameters per pixel, 3 for the chromaticity and 6 for the luminance polynomial. All parameters are stored in 8-bit textures, and global scale and bias values are applied to each of the polynomial coefficients in order to cope with different dynamics in their values.

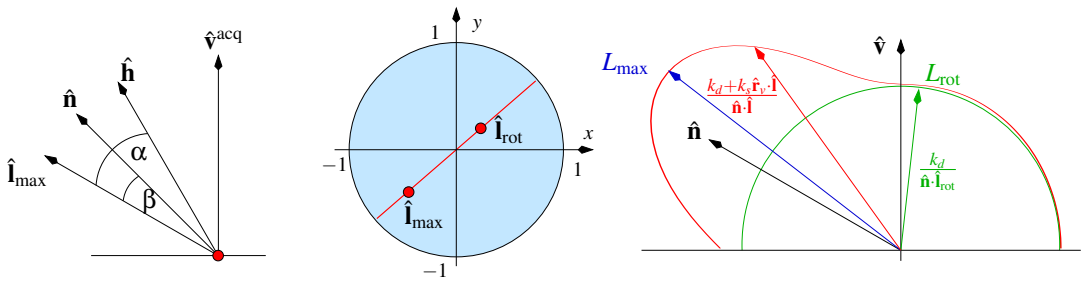

Figure 2: *Estimating $\hat{\mathbf{n}}, k_d, k_s$ for the Blinn-Phong model based on the bi-quadratic PTM reflectance function: The angular representation of the Blinn-Phong model (left), PTM parameters of the light directions with maximum reflectance and the rotated light direction (middle) and the reflectance distribution (right).*

4. 3D PTM Rendering for Polygonal Objects

We consider PTMs as 2D textures (*2D PTMs*) which may be mapped onto an arbitrary polygonal geometry. Our 3D PTM rendering approach is similar to that of Malzbender et al. [MGW01] since we use a single light direction. However, there are two challenges of rendering texture mapped PTMs (*3D PTMs*):

- The light direction is defined in world or in view space and needs to be transformed in local reference frames per pixel (see Sec. 4.1), and
- 2D PTMs are only valid for a single, orthogonal viewing direction, which is the view direction during acquisition of the images (see Sec. 4.2).

Ignoring the later aspect automatically yields a view-independent, i.e. a diffuse-like rendering, where the appearance of the object surface does not change under varying viewing directions. We include view-dependent effects by replacing the specular component from the original PTM with a specular contribution resulting from the viewing direction during rendering.

4.1. Local PTM Coordinates

Mapping PTMs onto polygons is done using standard methods, as used e.g. for normal mapping [COM98], which are modified to work on GPUs using standard graphics programming APIs like GLSL (see [Len04] for further technical details).

Having texture coordinates (u,v) associated to the triangle vertices, the local tangent frame is aligned with the $u-$ and $v-$directions of the texture map. The resulting local coordinate frame is computed for each vertex in a geometry shader. After the perspective interpolation, performed by the rasterization unit, we obtain a local frame $\{\hat{\mathbf{t}},\hat{\mathbf{b}},\hat{\mathbf{n}}\}$ representing the texture orientation in 3-space. Now, the global light direction $\hat{\mathbf{l}}$ can be projected into the local coordinate frame: $\hat{\mathbf{l}}^{loc} = \left(\hat{\mathbf{t}}\cdot\hat{\mathbf{l}}, \hat{\mathbf{b}}\cdot\hat{\mathbf{l}}, \hat{\mathbf{n}}\cdot\hat{\mathbf{l}}\right)$. The resulting tangential coordinates of the light vector $\hat{\mathbf{l}}^{loc}$ are used to compute the reflectance according to Eqs. 2 and 3.

4.2. Including View Dependency for Specular Reflections

As mentioned above, applying PTMs in the same way as normal maps automatically yields a diffuse rendering, since no view-dependent effect is taken into account. Nevertheless, the PTM data include a specular contribution, but this is only valid for the viewing direction at acquisition time. To include view-dependency at rendering time, we isolate the fixed specular part from the PTM and replace it with a new one using the specular component with respect to the current viewing direction. Assuming a Phong-like illumination model for the luminance, we can describe this by

$$L^{new} = L^{acq} - L^{acq}_{spec} + L^{curr}_{spec}. \tag{5}$$

Assuming a modified Blinn-Phong illumination [AMHH08] model, i.e.

$$L = L_{diff} + L_{spec} = \hat{\mathbf{n}}\cdot\hat{\mathbf{l}}\left[k_d + k_s(\hat{\mathbf{n}}\cdot\hat{\mathbf{h}})^n\right],\ \hat{\mathbf{h}} = (\hat{\mathbf{l}}+\hat{\mathbf{v}})/\left\|\hat{\mathbf{l}}+\hat{\mathbf{v}}\right\|, \tag{6}$$

we need to estimate the normal $\hat{\mathbf{n}}$ and the reflection coefficients k_d, k_s of the acquired object in order to replace the specular part. In contrast to exhaustive optimization approaches on the basis of the reflectance samples, like in Schirmacher et al. [SHR*99], we directly work with the bi-quadratic PTM reflectance approximation. The only user parameter is the specular exponent n.

Before estimating k_d, k_s and $\hat{\mathbf{n}}$, we compute the direction of maximum reflectance based on the biquadratic PTM approximation of the reflectance function [MGW01]:

$$\text{grad}\,[L](x_0,y_0) = \begin{pmatrix}\frac{\partial L}{\partial x}\\ \frac{\partial L}{\partial y}\end{pmatrix} = \begin{pmatrix}2a & c\\ c & 2b\end{pmatrix}\cdot\begin{pmatrix}x_0\\ y_0\end{pmatrix} + \begin{pmatrix}d\\ e\end{pmatrix} = \begin{pmatrix}0\\ 0\end{pmatrix}$$

$$\Rightarrow \begin{pmatrix}x_0\\ y_0\end{pmatrix} = \frac{1}{4ab-c^2}\begin{pmatrix}ce-2bd\\ cd-2ae\end{pmatrix}. \tag{7}$$

The light vector resulting in a maximum reflection is given as $\hat{\mathbf{l}}_{max} = \left(x_0, y_0, \sqrt{1-x_0^2-y_0^2}\right)^T$.

Estimating $\hat{\mathbf{n}}$: Assuming diffuse reflection, we would get

$\hat{\mathbf{n}} = \hat{\mathbf{l}}_{\max}$. On the other hand, in case of purely specular reflection, $\hat{\mathbf{n}}$ is the halfway vector $\hat{\mathbf{h}}$ between $\hat{\mathbf{l}}_{\max}$ and the fix viewing vector $\hat{\mathbf{v}}^{\text{acq}}$ during the PTM acquisition. In general, both, diffuse and specular components are present, so the desired $\hat{\mathbf{n}}$ must lie somewhere in between those two (see Fig. 2, left). We express the modified Blinn-Phong model of Eq. 6 in angular terms, yielding

$$L^{\text{acq}}(\beta) = \cos(\beta)(k_d + k_s \cos^n(\alpha - \beta)).$$

As the maximum reflectance vector $\hat{\mathbf{l}}_{\max}$ and the viewing vector $\hat{\mathbf{v}}^{\text{acq}}$ are known, searching for the maximum of $L^{\text{acq}}(\beta)$ by varying β yields the desired normal $\hat{\mathbf{n}}$. Since there is no closed solution to this inverse problem, we apply a bisection search to $\beta \in [0, \alpha]$.

Estimating k_d, k_s: For the PTM light direction $\hat{\mathbf{l}}_{\max}$ from Eq. 7 with maximum luminance $L_{\max}$, diffuse and specular components superimpose. In order to estimate k_d, we assume to observe a nearly diffuse reflection when we rotate $\hat{\mathbf{l}}_{\max}$ by about 60° (see Fig. 2, middle and right), i.e. $\hat{\mathbf{n}} \cdot \hat{\mathbf{h}}_{\text{rot}} \approx 0$. Thus with Eq. 6 and the assumption $\hat{\mathbf{n}} \cdot \hat{\mathbf{h}}_{\max} \approx 1$ we get

$$\frac{L_{\text{rot}}}{L_{\max}} \approx \frac{\hat{\mathbf{n}} \cdot \hat{\mathbf{l}}_{\text{rot}} k_d}{k_d + k_s} \tag{8}$$

With $\cos(60^\circ) = \frac{1}{2}$ and assuming $k_d + k_s = 1$, we get $k_d = \frac{L_{\text{rot}}}{2L_{\max}}$.

Finally, we compute both specular reflections $L^{\text{acq}}_{\text{spec}}, L^{\text{curr}}_{\text{spec}}$ and apply k_s, the estimated normal $\hat{\mathbf{n}}$ and the current viewing direction $\hat{\mathbf{v}}^{\text{curr}}$ to the specular correction from Eq. 5.

5. PTM-based Light Field Rendering

The following sections will focus on the acquisition and the rendering process of a light field with controllable illumination. In Sec. 5.1, we present our setup for acquiring a 3D PTM-based light field. Sec. 5.2 discusses the rendering and relighting stage. Sec. 5.3 focuses on the difficulties arising from having a spherical distribution of light sources in contrast to a half sphere as in [MGW01].

5.1. Acquiring Input Images

In order to generate a spherical light field with dynamic lighting, we first choose the number of cameras which we distribute uniformly over the unit sphere. The density of the camera distribution determines the sampling rate of the light field. For each camera, we append an arrangement of light sources. For a single image, our acquisition setup is similar to that of Malzbender et al. [MGW01] in that the light directions are defined in the local reference frame of each camera. In contrast to their setup, where the arrangement of light sources only covers a hemisphere, we use a full 360° distribution of light sources. We benefit from this as we are able to capture specular reflections observed from a flat viewing angle of a surface.

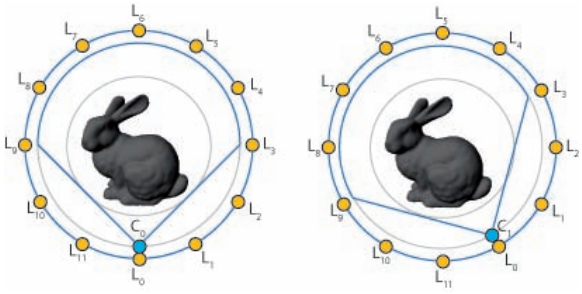

Figure 3: *Setup for the acquisition of input images. Shown are two subsequent acquisition stages for two cameras. Note that the number of cameras and the number of lights do not need to be equal.*

Our setup is based on the acquisition setup used by [TRSKK08], which acquires parabolically distorted images due to the spherical nature of the rendering process. To this setup, we add a number of light sources and distribute them over the sphere (see image 3).

For every camera on the sphere, we apply 3D PTM rendering for polygonal objects as described in Sec. 4 (`computeImage()`-function in the pseudo-code below). Given an arrangement of n cameras, this yields n sets of RGB + coefficient data. We store these data as RGB textures for each camera. In addition to this, we store a depth map of the object per camera view, which we use in the light field synthesis step to reconstruct surface points. The acquisition process of a light field with dynamic lighting can roughly be summarized as follows:

```
computeFittingMatrixForLightGeometry();
loadObjectGeometry();
for each camera{
   for each light{
      computeImage();
      addImageToImageStackForCurrentCam();
   }
   computePTMDataForImageStack();
   addPTMDataToLightfield();
}
```

With the current camera being the reference frame for the light sphere, the projected light direction of a light source onto a given image plane of a camera is always the same. We benefit from this, as we have to compute the fitting matrix M, which we use in the coefficient computation step, only once for a given light source geometry (see Eq. 4). This can be done as a pre-computing step.

As already mentioned, our light source can reside anywhere on the full sphere surrounding the object. Remember that the geometry of which we want to record a light field is textured with a PTM texture. With only a hemispherical light distribution, we could not achieve angles of more than

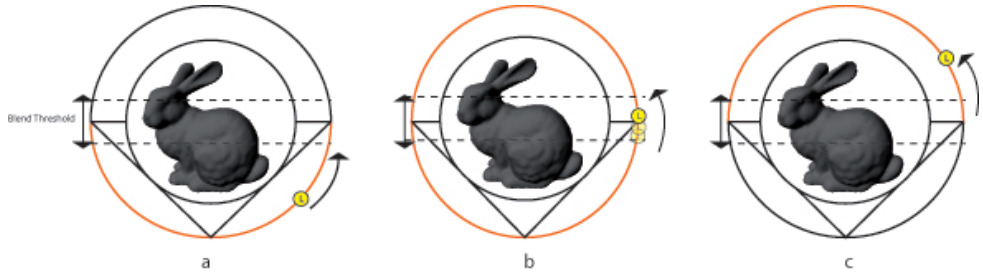

Figure 4: *Light positions on the front hemisphere (a) and the back hemisphere (c) close to the seam between both. Blending between the front and the back hemisphere (b).*

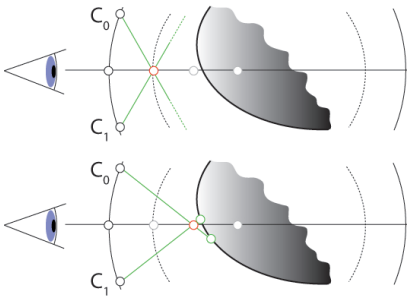

Figure 5: *The light field raycasting approach. Image courtesy of Todt et al. [TRSKK08]*

90 degrees between incoming light and view direction. For PTM-based relighting, this would suffice only in the case of triangles being perpendicular to the viewing direction, which most probably will not be the case for most of the triangles. Thus, as a logical consequence of our 3D PTM rendering approach, we use a full light sphere. With a single bivariate polynomial, however, we cannot distinguish between light coming from the back hemisphere and from the front hemisphere. We chose to compute two fitting matrices, one for each hemisphere, which results in having two sets of PTM coefficients.

5.2. PTM Light Field Rendering and Relighting

Our light field is reconstructed by rendering the front faces of the smooth shaded spherical camera geometry from an arbitrary viewpoint by a customized fragment program. The rendering process is based on a ray casting approach described by Todt et al. [TRSKK08], see Fig. 5 for an illustration. For each fragment, we cast a ray into the scene. As done by Todt et al. , we determine the intersection point between the viewing ray and the surface of the object represented by the light field. With the depth textures bound to the three adjacent cameras C_0, C_1 and C_2 of the fragment, each sample point on the ray is tested for the assumption that it is an actual point on the object. When the intersection point on the surface of the object is found, three chromaticity values, RGB_0, RGB_1 and RGB_2 can be determined by sampling the RGB textures attached to the cameras. The final color of the fragment is then determined in the relighting step. The reconstruction of the correct surface point automatically yields the correct texture coordinates for sampling the textures containing the polynomial coefficients for the respective pixel. For each camera, we sample the coefficient textures and set up the polynomial to determine the luminance of the pixel. The input for the polynomial consists of the current light direction transferred into the local coordinate systems of the cameras and projected onto their image planes, yielding (x_0, y_0), (x_1, y_1) and (x_2, y_2). Evaluating the polynomials produces three luminance values L_0, L_1 and L_2, and together with the chromaticity values the final fragment RGB_{final} can be evaluated by

$$RGB_{final} = w_0 L_0 RGB_0 + w_1 L_1 RGB_1 + w_2 L_2 RGB_2 \quad (9)$$

with w_n being the barycentric weights of the current fragment in the triangle.

5.3. Spherical vs. Hemispherical Light Distribution

Recall, that we compute and store the PTM for the front and the back hemispheres separately. This leads to noticeable flickering during the relighting as the light vector switches from the front to the back hemisphere or vice versa. This is due to discontinuities between the two functions approximating the illumination of the scene, since there is no warranty for a smooth reconstruction of surface luminance in these separate PTMs (see Fig. 4). We solve this by choosing a threshold τ for the light position's z-value. Beyond this threshold, we sample both of the coefficient textures and blend the coefficient vectors $\vec{\mathbf{c}}_{\text{front}}$ and $\vec{\mathbf{c}}_{\text{back}}$

$$\vec{\mathbf{c}} = lerp(\vec{\mathbf{c}}_{\text{frontSide}}, \vec{\mathbf{c}}_{\text{backSide}}, \alpha) \quad (10)$$

with $\alpha = (1 \pm \frac{\hat{\mathbf{l}}_z}{\tau})/2$. As a result, we achieve a smooth interpolation as soon as the light source approaches the seam from one direction or the other.

6. Results

6.1. PTM Light Field Acquisition

In the acquisition setup we used 12 (or 42) camera positions and 12 (or 42) light positions, respectively. The numbers derive from the vertices of a hierarchically subdivided icosahedron, which is used to approximate the sphere around the object. Any other light source distribution can also be used, given that the light's positions are known. Fig. 6 shows a sample 3D PTM from the acquisition module with varying light positions. Result images of our 3D PTM renderer with

Figure 6: *A view of a model using a 3D PTM with varying light source positions.*

and without specular correction can be seen in Fig. 7. The

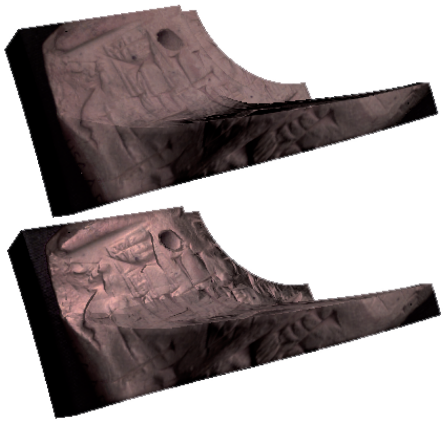

Figure 7: *Views on a model using 3D PTM without (top) and with (bottom) specular shading correction.*

acquisition of all data for a 12×12 PTM light field takes about one minute, including the initial computation for the fitting matrix. The overall file size for this parametrization amounts to about 224.5 MB. For all other parametrization combinations see table 1. Note that, due to the PTM approximation, the file size is independent of the number of initial light sources.

Cameras	Lights	Acquisition Time	File Size
12	12	$\approx$ 1 min	224.3 MB
12	42	$\approx$ 3 min	224.3 MB
42	12	$\approx$ 6 min	784.9 MB
42	42	$\approx$ 10 min	784.9 MB

Table 1: *Acquisition time and file size for sample parametrizations.*

6.2. PTM Light Field Rendering

Fig. 8 shows a 3D PTM sample image from the acquisition module (left) and the same view on the object which was synthesized by the PTM light field renderer using the same light direction (right). Fig. 9 shows a PTM light field

Figure 8: *PTM-texture-mapped geometry (left), PTM light field reconstruction (right).*

acquired by a camera and light setup of 42×42. The light position here is fixed to the light field, only the viewpoint changes. Fig. 10 shows the same PTM light field. Here, the

Figure 9: *$42\times$ 42 PTM light field with fixed light position and changing viewpoint.*

viewing position is fixed while the light position moves from

the left to the right side of the light field representation of the bust. Using a 512×512 viewport, our PTM light field ren-

Figure 10: *42× 42 PTM light field with fixed view direction and light position moving from left to right.*

derer renders a relightable light field of 42 cameras at frame rates of 20fps and a 12 camera light field at frame rates of 29fps. The platform used for this metrics was an Intel i7 920 2.67 GHz machine equipped with an NVidia GeForce GTX 8800 with 768MB of VRAM. Changing the light position during rendering did not lead to noticeable changes in the frame rate.

7. Conclusion And Future Work

We have presented a novel technique for relighting spherical light fields, using polynomial texture maps to store the illumination dependent information. For the light field acquisition, we provide polygonal objects with meso-scale surface detail without elaborate modeling. Small lighting changes reveal structural changes on the object's surface, retaining natural reflections, inter-reflections and self-shadowing. By adding view-dependency, we can model specular reflections and even enhance or attenuate them. Future work will be dedicated to acquire view-dependent PTM light fields of real objects. Furthermore, we will work on efficiency and compression of our data structures.

Acknowledgments

This work has been partially supported by the German Research Foundation (DFG) under grant number Ko-2960-6/1,2. Furthermore, we would like to thank Tom Malzbender for providing the PTM datasets on his website. All PTM datasets used in this work are taken from the HP Labs website [HP].

References

[AB91] ADELSON E. H., BERGEN J. R.: The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, Landy M. S., Movshon J. A., (Eds.). MIT Press, 1991, pp. 3–20. 2

[AMHH08] AKENINE-MÖLLER T., HAINES E., HOFFMAN N.: *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., 2008. 1, 4

[BBM*01] BUEHLER C., BOSSE M., MCMILLAN L., GORTLER S., COHEN M.: Unstructured lumigraph rendering. In *Proc. ACM SIGGRAPH* (2001), pp. 425–432. 2

[Bli78] BLINN J.: Simulation of wrinkled surfaces. In *Proc. ACM SIGGRAPH* (1978), pp. 286–292. 1

[CLF98] CAMAHORT E., LERIOS A., FUSSELL D.: *Uniformly Sampled Light Fields*. Tech. rep., University of Texas at Austin, Austin, TX, USA, 1998. 2

[COM98] COHEN J., OLANO M., MANOCHA D.: Appearance-preserving simplification. In *Proc. ACM SIGGRAPH* (1998), pp. 115–122. 4

[DHT*00] DEBEVEC P., HAWKINS T., TCHOU C., DUIKER H.-P., SAROKIN W., SAGAR M.: Acquiring the reflectance field of a human face. In *Proc. ACM SIGGRAPH* (2000), pp. 145–156. 1, 2, 3

[FBS05] FUCHS M., BLANZ V., SEIDEL H.-P.: Bayesian relighting. In *Proc. Euographics Symposium on Rendering Techniques* (2005), pp. 157–164. 2

[FLBS07] FUCHS M., LENSCH H., BLANZ V., SEIDEL H.-P.: Superresolution reflectance fields: Synthesizing images for intermediate light directions. *J. Computer Graphics Forum* (2007), 447–456. 2

[GGSC96] GORTLER S., GRZESZCZUK R., SZELISKI R., COHEN M.: The lumigraph. In *Proc. ACM SIGGRAPH* (1996), pp. 43–54. 1, 2

[GVL96] GOLUB G. H., VAN LOAN C. F.: *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. 3

[HP] http://www.hpl.hp.com/research/ptm/. 8

[IPL97] IHM I., PARK S., LEE R. K.: Rendering of spherical light fields. In *Proc. Pacific Graphics* (1997), pp. 59–68. 2

[Len04] LENGYEL E.: *Mathematics for 3D Game Programming and Computer Graphics*. Charles River Media Inc., 2004, ch. 6.8 Bump Mapping. 4

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proc. ACM SIGGRAPH* (1996), pp. 31–42. 1, 2

[LWS02] LIN Z., WONG T.-T., SHUM H.-Y.: Relighting with the reflected irradiance field: Representation, sampling and reconstruction. *Int. J. Comput. Vision 49*, 2-3 (2002), 229–246. 3

[MGW01] MALZBENDER T., GELB D., WOLTERS H.: Polynomial texture maps. *Proc. ACM SIGGRAPH 20*, 3 (2001), 519–528. 1, 3, 4, 5

[MLP04] MATUSIK W., LOPER M., PFISTER H.: Progressively-refined reflectance functions from natural illumination. In *Proc. Euographics Symposium on Rendering Techniques* (2004), pp. 299–308. 2

[MPDW03] MASSELUS V., PEERS P., DUTRÉ P., WILLEMS Y.: Relighting with 4d incident light fields. *Proc. ACM SIGGRAPH 22*, 3 (2003), 613–620. 2

[SHR*99] SCHIRMACHER H., HEIDRICH W., RUBICK M., SCHIRON D., SEIDEL H.-P.: Image-based brdf reconstruction. In *Proc. Vision, Modeling and Visualization* (1999). 4

[TRSKK08] TODT S., REZK-SALAMA C., KOLB A., KUHNERT K.-D.: GPU-based spherical light field rendering with per-fragment depth correction. *J. Computer Graphics Forum27*, 8 (2008), 2081–2095. 1, 2, 3, 5, 6