Eurographics Symposium on Rendering (2004)
H. W. Jensen, A. Keller (Editors)

Progressively-Refined Reflectance Functions from Natural Illumination

Wojciech Matusik†, Matthew Loper, and Hanspeter Pfister

Mitsubishi Electric Research Laboratories, Cambridge, MA.

Abstract

In this paper we present a simple, robust, and efficient algorithm for estimating reflectance fields (i.e., a description of the transport of light through a scene) for a fixed viewpoint using images of the scene under known natural illumination. Our algorithm treats the scene as a black-box linear system that transforms an input signal (the incident light) into an output signal (the reflected light). The algorithm is hierarchical – it progressively refines the approximation of the reflectance field with an increasing number of training samples until the required precision is reached. Our method relies on a new representation for reflectance fields. This representation is compact, can be progressively refined, and quickly computes the relighting of scenes with complex illumination. Our representation and the corresponding algorithm allow us to efficiently estimate the reflectance fields of scenes with specular, glossy, refractive, and diffuse elements. The method also handles soft and hard shadows, inter-reflections, caustics, and subsurface scattering. We verify our algorithm and representation using two measurement setups and several scenes, including an outdoor view of the city of Cambridge.

Categories and Subject Descriptors (according to ACM CCS): I.2.10 [Artificial Intelligence]: Vision and Scene Understanding; I.3.3 [Computer Graphic]: Picture/ Image Generation

1. Introduction

We are interested in acquiring a representation of arbitrary objects and scenes that allows us to render new images of the scene using natural illumination captured in real-world environments. Such a representation has many applications in realistic image synthesis (offline or real-time) and scene analysis, such as 3D scanning of cultural artifacts, object and face recognition, or relighting of image-based models. Current approaches require devices that generate controlled incident illumination (either point light sources or structured light). Even for small objects these devices are typically large, expensive, and not very portable.

We propose a novel approach that works with known natural illumination and requires a small number of measurements to compute a good estimation of the scene reflectance. Note that we use the term natural illumination freely, since images of natural scenes can also be displayed on a monitor that illuminates the scene. Figure 1 shows a few results from our method.

The problem of natural scene acquisition has received a lot of attention in computer graphics and computer vision. In general, we would like to acquire the *reflectance field* [DHT*00], also known as the bidirectional scattering-surface reflectance distribution function (BSSRDF) [NRH*77]. It is defined as the ratio of incoming to outgoing radiance at two different scene points. Formally we use $f_r(\omega_i, \vec{x}_i, \omega_o, \vec{x}_o)$, where ω_i is the direction of the incident illumination at point $\vec{x}_i$, and ω_o is the observation direction of radiance emitted at point $\vec{x}_o$. The function is eight-dimensional, assuming a two-dimensional parameterization for the points in the scene.

Because sampling of an eight-dimensional function is challenging, we focus on a four-dimensional subspace or 4D slice of the reflectance field. First, we assume that the incident illumination is far away and arrives at scene points from direction ω_i. Second, we acquire the reflectance field for each image pixel (x,y), which implies both scene position x_o and viewing direction ω_o. The resulting 4D

† [matusik,pfister]@merl.com

Figure 1: Top row: Actual image of a scene under new natural illumination. Bottom row: Prediction using our method.

function $f_w(\omega_i, x, y)$ has been called the *weighting function* [ZWCS99] or *reflectance function* [DHT*00] of light transport from distant illumination to the observation point. It is a view-dependent slice of the full 8D reflectance field.

However, robustly acquiring this 4D reflectance function for arbitrary scenes is also difficult. Consider a scene where the surface coordinates are discretized into 1000×1000 points and the incident illumination is represented by a 1000×1000 pixel image. Sampling and tabulating the reflectance function directly requires 10^{12} values, posing a challenge for storage. Moreover, the acquisition time would be prohibitively long to take 10^6 high-resolution photographs.

In practice, most direct sampling approaches use only low-resolution (low-frequency) incident lighting. Consequently, they cannot represent high-frequency effects, such as specularities, refraction, and shadows. On the other hand, environment matting techniques focus on reflectance functions for specular, refractive, and glossy materials, but they have difficulty representing hard shadows or a combination of diffuse and specular materials.

We propose a novel approach for reflectance function estimation and representation with the following properties:

Natural Illumination Input: Our algorithm does not require structured illumination patterns or point light source input. It works with arbitrary (known) natural illumination.

All-Frequency Robustness: Our algorithm is equally efficient for both the high frequency (e.g., specular) and low frequency (e.g., diffuse) components of reflectance fields. It also handles scenes with discontinuous reflectance fields (e.g., hard shadows).

Compact Representation: Our underlying representation of reflectance fields is compact. One 4D slice of a reflectance field can be well represented with the storage required for a few 2D slices.

Progressive Refinement: Our algorithm improves the approximation of the reflectance field as more measurements are made.

Fast Evaluation: Our representation of reflectance fields allows us to rapidly perform the integration with complex incident illumination.

Simplicity: The algorithm is very simple and can be quickly implemented.

2. Previous Work

As mentioned above, a reflectance function $f_w(\omega_i, x, y)$ relates incoming illumination to observed radiance. We can classify the methods for estimating reflectance functions into *forward* and *inverse* methods.

Forward Methods Most forward methods sample the reflectance functions exhaustively and tabulate the results. For each incident illumination direction they store the reflectance function weights for a fixed observation direction. In practice, only low-resolution incident illumination can be used, since one reflectance table has to be stored per scene point. Debevec et al. [DHT*00] use the highest resolution incident illumination with roughly 2000 directions. Polynomial texture maps [MGW01] improve the compactness of the representation by expressing each reflectance field table with a bivariate polynomial. These direct approaches work very well for diffuse or glossy objects, such as human faces [GBK99, DHT*00], cultural artifacts [HCD01, MGW01, MDA02], and other objects with complex appearance [MPN*02]. However, they can not represent high-frequency phenomena (such as refractions) or discontinuities in the reflectance field (such as hard shadows). Furthermore, these methods do not provide progres-

sive refinement of the approximation with an increasing number of samples.

Wavelet environment matting [PD03] addresses some of these shortcomings. Images with wavelets patterns are used as incident illumination, and a feedback loop determines the next pattern to emit based on the error of the current approximation. The reflectance function is progressively refined as more measurements are made. However, this algorithm requires many wavelet patterns for highly specular and refractive materials or scenes with hard shadows (up to 2400 pattern images are reported by Peers and Dutré [PD03]). Since each pixel stores the coefficients of the corresponding wavelet patterns, this representation becomes rather large (2.5 GB per environment matte with lossless compression). Rendering is also expensive, since it requires a weighted sum of many wavelet-basis images of the new incident illumination.

Masselus et al. [MPDW03] measure six-dimensional slices of the eight-dimensional reflectance field by varying both the position and direction of the incident illumination for a fixed viewpoint. They use a projector system for scene acquisition and tabulate the results.

Inverse Methods Inverse methods observe an output and compute the probability that it came from a particular region in the incident illumination domain. The incident illumination is typically represented by a bounded region, such as an environment map [BN76], which is then modeled as a sum of basis functions (e.g., rectangular [ZWCS99] or Gaussian kernels [CZH*00]). The inverse problem can then be stated as follows: Given an observation (e.g., an image pixel), what are the weights and parameters of the basis functions that best explain the observation?

The initial environment matting technique [ZWCS99] describes the observation in each pixel using only one 2D box function. This leads to a very compact representation, and the algorithm works well for specular and refractive objects. In principle, rendering with complex illumination could be performed fast, but this is not demonstrated in the paper. The algorithm works offline and does not progressively refine the approximation errors. Furthermore, one 2D box function is sometimes not expressive enough to capture the properties of complex reflectance functions.

High-quality extensions to environment matting [CZH*00] achieve excellent results for specular, refractive, and glossy materials. The box functions are replaced by oriented 2D Gaussian kernels, which also lead to a very compact representation. However, rendering with complex incident illumination is slow, since the Gaussian kernels have to be convolved with the new illumination. This algorithm also cannot progressively refine the approximation. Moreover, the 2D Gaussian kernels have difficulty representing discontinuous or more complex reflectance functions, such as a small specular lobe inside a larger diffuse component.

Wexler et al. [WFZ02] were the first to use natural illumination to estimate environment mattes. However, their method can only capture very specular and refractive scenes. Our method works for scenes with arbitrary materials.

Hybrid Methods Matusik et al. [MPZ*02] combine a forward method [DHT*00] for low-frequency components and high-quality environment matting [CZH*00] for high-frequency reflections and refractions. The low-frequency data is compressed using principal component analysis (PCA). However, their representation is not compact, and the approach does not address any of the problems mentioned above.

Pre-Computed Radiance Transport Reflectance functions have become very popular as a way to increase the realism in real-time rendering, where they are known as radiance transport functions. In these applications, the radiance transport is pre-computed using a detailed model of the scene [SKS02]. To improve the rendering performance, the incident illumination can be represented using spherical harmonics [RH01, KPS02, SKS02] or wavelets [NRH03]. The reflectance field (typically stored per vertex as a transfer matrix) can be compressed using PCA [LK03, SHHS03] or wavelets [NRH03]. Ng et al. [NRH03] introduce the term *all-frequency robustness* with respect to illumination discontinuities such as shadows.

We first describe our novel representation of the reflectance function based on hierarchical rectangular basis functions. We then describe a novel inverse algorithm to estimate the reflectance function from input images of the scene (Section 5). We discuss details of our experimental setups and acquisition procedure (Section 6) and finally show some results (Section 7).

3. Representation

In the following we use the notation of [NRH03] and [PD03]. We assume that the incident illumination is 2D parameterizable (i.e., can be represented by an environment map). If we consider the problem as a black box linear system, we can write:

$$B = TL, \tag{1}$$

where L is the incident illumination (stacked in a vector), B is the resulting relit image (stacked in a vector), and T is the light transport matrix (or reflectance field). Each row in T represents a reflection function T_i, thus $T = [T_0, \ldots, T_n]$. The pixel value of a single pixel $B = [b_i]$ is:

$$b_i = T_i \cdot L. \tag{2}$$

Note that the pixel value b_i is the inner product of two vectors T_i (reflectance function) and L (incident illumination).

We approximate T_i in (2) as a weighted sum of several 2D *rectangular kernels*, such that:

$$T_i \approx \sum_k w_{k,i} R_{k,i}, \tag{3}$$

where $w_{k,i}$ are the weights of each 2D rectangular box $R_{k,i}$. The kernels are constructed using an even-split kd-tree subdivision of the 2D image (see Figure 3). Figure 2 shows an example of the axis-aligned 2D kernels. Zongker

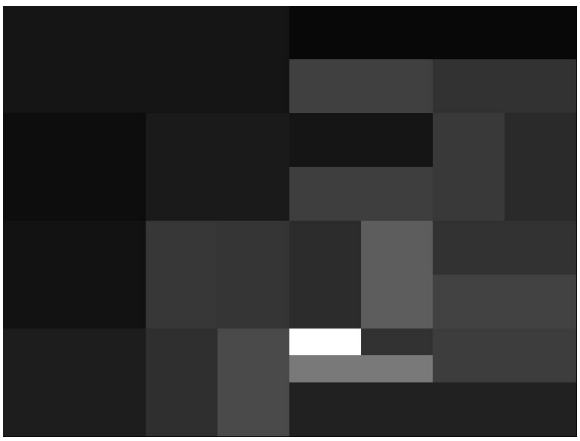

Figure 2: Example rectangular kernels.

et al. [ZWCS99] also use rectangular axis-aligned kernels. However, they considered only *one* kernel per reflectance function, while we consider many kernels of varying size and position.

Assuming that light transport through a scene is linear and additive, Equation (2) becomes:

$$b_i \approx \sum_k w_{k,i} R_{k,i} \cdot L. \tag{4}$$

The term $(R_{k,i} \cdot L)$ computes the contribution of the incident illumination L coming from the region of the corresponding 2D kernel $R_{k,i}$. We use summed-area tables to rapidly evaluate $(R_{k,i} \cdot L)$ for any of the kernels. This leads to a very efficient rendering algorithm.

This representation is very efficient. For each pixel b_i we store a vector with weights $w_{k,i}$ and the position and size of the associated kernels. We found that storing 25 kernels per pixel yields high all-frequency robustness (see Section 7).

4. Assumptions

As stated above, we assume that the light transport through the scene is a linear process. We also assume that the response curve of the device that generates light (e.g., a monitor) and of the measurement device (e.g., a camera) can be linearized. Furthermore, we assume that the scene elements are static and that the incident illumination is 2D parameterizable.

5. Estimation Algorithm

The problem of reflectance field acquisition is now equivalent to estimating the impulse response function (or point spread function) of a linear system with multiple inputs (the incident illumination L) and multiple outputs (the observation B).

5.1. Algorithm Input and Output

The recording process can be described as:

$$B_j = TL_j, \tag{5}$$

where L_j is a set of different natural incident illumination (the *input images*) and B_j is the resulting set of observed photographs of the scene (the *observed images*). We typically use between 100 and 200 input and observed images.

We denote a pixel $b_{i,j}$ as pixel i in the j-th observed image. Given L_j and $b_{i,j}$, the algorithm estimates the weights $w_{k,i}$, positions, and sizes of the 2D rectangular kernels for each pixel b_i in the final *reflectance field* image. As we increase the number n of input and observed images, we refine the kernels to improve the estimate of the reflectance field.

Note that we compute the weights independently for each color channel (RGB). However, we only store one set of 2D kernel positions per pixel. This means we do not model dispersion, where one color channel may influence the others, although this may be possible in principle. This also assumes that there is no color cross-talk between monitor spectra and camera pixel filters.

The algorithm has two major steps. First, for each pixel b_i in the reflectance field, we estimate its kernels using an even-split kd-tree subdivision of the incident (2D) illumination map. In the second step we further improve the result using the estimates of other pixels in a small neighborhood around pixel b_i.

5.2. Kernel Subdivision

We observe that the problem can be solved independently for each pixel b_i. We solve the following optimization problem:

$$\begin{aligned} &argmin_W \ ||A_i W_i - B_i||^2 \\ &s.t.\ W \geq 0. \end{aligned} \tag{6}$$

W_i is the stacked vector of weights $w_{k,i}$ to be computed (for a single pixel b_i). $A_{i_{k,j}} = [R_{k,i} \cdot L_j]$, where L_j is one of the input natural illumination images, and $B_i = b_{i,j}$. We found through experimentation that constraining the kernel weights to be positive is very important for the stability of the solution.

Equation (6) is equivalent to the following linear system:

$$\begin{aligned} &argmin_W \ 0.5 W_i^T A_i^T A_i W_i - W_i^T A_i^T B_i \\ &s.t.\ W \geq 0. \end{aligned} \tag{7}$$

This over-constrained system can be efficiently solved using

quadratic programming. It estimates the weights that satisfy the system best in the least-squares sense.

To find the kernel positions and sizes for pixel b_i, we use an even-split kd-tree subdivision of the input image to find them (see Figure 3). We first start with a kernel that occupies

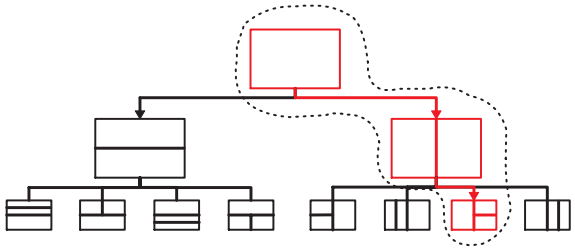

Figure 3: Example kd-tree subdivision of a kernel. The indicated path shows the subdivision that yields the least error.

the whole image. We split it into two equal-size rectangles. Having the option of splitting the kernel horizontally or vertically, we choose the subdivision that yields the lowest error in Equation (6). To compute this error we solve for the kernel weights using Equation (7). In the second iteration we have four possible splits of the two kernels from the first iteration, and so on. The recursive subdivision stops if K kernels are computed (or $K-1$ subdivisions have been made). We found that $K = 25$ yields a good quality-vs.-time tradeoff (see Section 7 and Figure 8).

In general, we have $2 * k$ possible splits in the k^{th} iteration. Thus in the k^{th} iteration we would need to perform $O(k^2)$ quadratic programming optimizations using Equation (7). Instead of trying all possible equal-size splits, we found that we can split the kernel with the highest energy, which we define as area $\times$ weight. This means we only perform $O(k)$ quadratic programming optimizations per pixel, which improves the performance by a factor of 10 on our datasets.

The even-split kernel subdivision is not optimal. We believe that subdividing kernels into non-equal parts would yield better results, at the price of slightly increased storage. In addition, our energy measure is not perfect, and we may miss the optimal split for a given kernel. Both issues lead to sub-optimal estimations that need to be spatially corrected.

5.3. Spatial Correction

We perform the spatial correction for pixels b_i whose error (Equation (6)) is bigger than a threshold τ. We observe that the kernels for neighboring pixels are usually similar.

Our algorithm proceeds as follows: For each pixel b_i, it finds the kernel locations from each neighboring pixel and solves for the new weights using Equation (7). It then computes the new error using Equation (6) and checks if it is smaller than the current error. If it is, it copies the kernel position and sizes from the neighbor that yields the lowest error. We iteratively repeat this procedure for all the pixels with high error until the error falls below a threshold. This assures spatial coherence and reduces the overall error. Next, we optimize the position of all the new kernels by simultaneously moving them one pixel in each direction until the error does not decrease any more.

Spatial correction improves the estimation of the reflectance function, especially when the number of input images is small (see Figure 1 top). It typically does not need to be performed if the number of input images is large.

6. Experimental Setups and Acquisition

We have built two experimental setups, one for indoor and one for outdoor scenes. The indoor system includes a camera that measures the reflected radiance; a monitor that generates the incident illumination; and the scene for which we want to estimate the light transport. Figure 4 shows a photo of the system. We use a Dragonfly camera from PointGrey

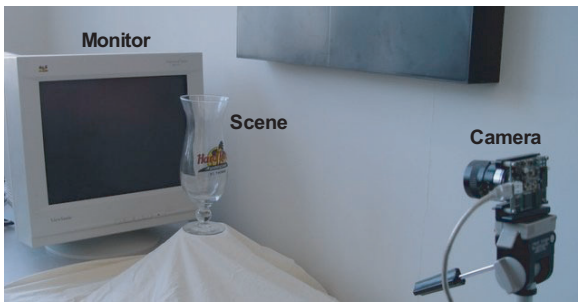

Figure 4: Acquisition system for indoor scenes.

Research (www.ptgrey.com) with a 1024×768 pixel CCD sensor and an IEEE-1394 (FireWire) output. The imager has 8 bits of precision per pixel and uses a Bayer pattern for color acquisition. The monitor is a ViewSonic PF775 CRT set to 800×600 output resolution at 85 Hz refresh rate. Monitor and camera are connected to a 2.8 GHz Pentium 4 PC with 2 GB of RAM.

As mentioned in Section 5, we assume that both the output device (the monitor) and the input device (the camera) have linear response. We confirmed that the relationship between exposure time and radiance values is linear over most of the camera's operating range. We use the following simple calibration procedure to correct for the non-linearity of the CRT monitor. We display images with uniform color on the monitor and acquire corresponding images with the camera. The images are created by varying the framebuffer values of each color channel between 0 and 255 in increments of one, keeping the other two channels at a fixed value (e.g., 128). The corresponding acquired images give us a mapping between framebuffer values and observed radiance. We use this table to linearize the input images during our optimization procedure.

One single image usually cannot capture the full dynamic range of the scene – some areas may be too too dark, others may be over-exposed. Therefore, we take all images using high-dynamic range (HDR) imaging [DM97]. For each input image, we take four pictures with exponentially increasing exposure times and use a least-squares linear fit to determine the response line. We store the slope of the response line as one floating point radiance measurement per pixel.

Before making any measurements, we acquire one HDR image with the ambient light in the room and subtract it from each observed image. In order to minimize the amount of computation and storage, we determine which observed pixels show scene elements of interest in a pre-processing step. This can be done either manually, by painting a mask over an image, or automatically, through simple background subtraction.

The input images that are displayed on the monitor include indoor and outdoor scenes – such as kitchens, gardens, cities, etc. Alternatively, we use a sequence of images from rotated versions of a panorama (environment map). We found that either method leads to good results, provided that the input images have sufficient variation and some high-frequency content. Ideally, the system would generate high-resolution incident illumination from all directions and from far away. Our simple acquisition setup certainly does not do this. However, we believe that achieving good results despite this limitation demonstrates one of the strengths of the algorithm.

To prove that our method also works with real outdoor illumination, we have built the system shown in Figure 5. The

Figure 5: Acquisition system for an outdoor (city) scene.

inset on the top-right shows the first camera that is pointed at the city of Cambridge from an office window in our building. The rest of the image shows the second camera that captures the incident illumination on the rooftop of the building using a chrome sphere. The two cameras are connected to two PCs that have approximately synchronized clocks. For our experiment, both cameras simultaneously captured HDR images of the city and the sky every two minutes over a period of three days.

To reduce the brightness of the sun we use three consecutive neutral-density filters for the camera on the roof. The reduced brightness lead to acquisition times of approximately 30 seconds for the HDR images. Almost all of the energy in the sky comes from the sun, which is practically a very bright point light source. For the reflectance field estimation we discarded all of the sky images where only the sun was visible and used 97 images with various cloud covers. We automatically fill in the region where the camera is visible with neighboring pixels.

7. Results

Our interactive renderer for reflectance fields has been implemented entirely in software. The new (input) illumination is loaded and converted into a summed area table. For each pixel in the reflectance field, we evaluate Equation (4) to compute the new output color. We can render about 0.5 frames per second for a reflectance field resolution of 1024×768. The frame rate increases proportionally with the number of displayed pixels.

We first show the results of our algorithm on a variety of indoor scenes. The scenes included specular, refractive, diffuse, and glossy objects. We also show scenes with soft shadows, hard shadows, and subsurface scattering. Unless noted otherwise, we used 180 natural input images to compute our predictions. Our input resolution is 800×600 pixels, and our reflectance field resolution is 1024×768. All results were computed using HDR images. The number of kernels per pixel is 25, which proved to work well for all scenes. The computation time is about two hours per reflectance field on a 2.8 GHz Pentium 4 PC with 2 GB of RAM. In addition to our set of input images – the training set – we have a test set of natural images that we use for the validation of the reflectance function estimation.

As a standard test of the quality of reflectance field estimation, we first compare the actual picture of the scenes under new illumination with the prediction obtained using our estimated reflectance field (see Figure 1). In general, our prediction works very well, for glossy, diffuse, and transparent objects. The hard shadows (near the objects) and soft shadows (farther away from the objects) are well reproduced, and the inter-reflection of the gold teacup on the table (right) is reproduced correctly. Diffuse elements (e.g., the label on the molasses jar) are reproduced correctly. The Santa model, which is hollow, shows both subsurface scattering and refractions.

Since our algorithm depends entirely on natural images, the most difficult test for our reflectance field prediction is to

Figure 6: Images of the scenes under synthetic illumination (vertical white bar) that is radically different than any image in the input set.

use synthetic input illumination with very high frequencies, such as images with white bars on a black background. Figure 6 show our predictions of the scenes with a vertical white bar, 100 pixels wide, sliding across the illumination domain. The glass clearly shows the greasy fingerprints, which corresponds well to reality. However, the video of this scene contains a slight discontinuity due to the vertical subdivision used for the kernels.

The molasses scene accurately reproduces specular reflections and the soft shadow across the table. Note the refraction on the bottom of the glass when the light is on the opposite side. Some noise in the form of discoloration is visible in the shadow areas. We believe it is due to non-optimal kernel subdivision. The tea set scene also shows soft shadows and moving reflections of the specular objects on the table. Note how the superposition of shadows and specular reflections are correctly handled in Figure 6.

Figure 7 shows how the quality of the prediction gets worse with a decreasing number of input images (from left to right: actual (top), 100, 75, 50, and 25). There are many visible errors when we use 25 input images (top right). However, using 75 input images (top middle) yields results that are hard to distinguish from the original. Note that fingerprints on the glass do show up in all of the images. The bottom row shows the output of the subdivision algorithm before doing any spatial correction. Even with 100 images there are differences between the non-corrected and the corrected predictions. One notable feature of this glass is that the bottom part magnifies the background, while the top part minifies it.

Next we show images of the progressive subdivision of the kernels for different scene points indicated by the red arrows (see Figure 8). The kernels progress from left to right with 1, 2, 3, 4, 5, 10, 20, and 24 subdivisions. The grey-scale colors of the kernels signify their weight (white is highest). Intuitively, these images show which light from the background is transported towards the pixel. The first row shows how our algorithm quickly (after 10 subdivisions) identifies the small illumination kernel that is refracted by the glass. We identify one pixel (from the 800×600 input image) after 24 kd-tree subdivisions, which is quite remarkable. The second row shows a primarily diffuse point in a hard shadow area. Note how the light transport comes mostly from the right as expected, and how a small reflective component is visible as well. The third row shows a point with subsurface scattering. Light from a large area is scattered towards the viewer. Finally, the last row shows a glossy point. The illumination enters the scene from the left (i.e., where the monitor was placed during acquisition). Note how the light is transported from almost concentric circular areas towards the incoming illumination.

Scene	Nonzero	99%	95%
Hardrock	77.4%	71.0%	60.8%
Molasses	97.8%	93.6%	83.9%
Santa	96.2%	92.4%	83.8%
Teacups	99.3%	97.3%	88.8%

Table 1: Percentage of total number of kernels. Nonzero: total number of kernels with non-zero weights. 99% / 95%: percentage of kernels necessary to capture 99% and 95% of the energy (weight $\times$ area).

Table 1 shows statistics for the kernels of the different scenes. The first column shows the percentage of kernels with non-zero weights. It clearly shows that a reflective and refractive scene (hardrock glass) contains fewer significant kernels. The following two columns show how the energy (kernel weight $\times$ area) in each scene is captured by the kernels. For example, 99% of the energy in the hardrock glass scene is concentrated in 71% of the kernels.

Figure 9 shows results of relighting the city of Cambridge. The figure shows example input and observed images of sky and city, respectively. The sky images were captured as described in Section 6 and are resampled into latitude-longitude maps. The bottom two rows are our predictions of relighting the city with a vertical white bar, 100 pixels wide,

Figure 7: Comparison of the quality with different number of input images. Top row: With spatial correction. Bottom row: Without spatial correction. Left to right: Actual (top), 100, 75, 50, and 25 input images.

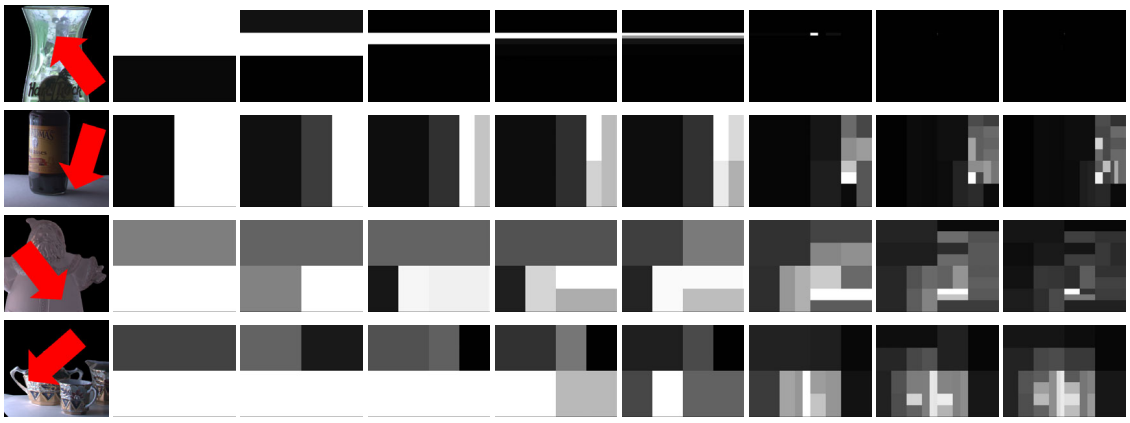

Figure 8: Kernel subdivisions for different scene points. From the top (row by row): specular and refractive; hard shadow; subsurface scattering; glossy.

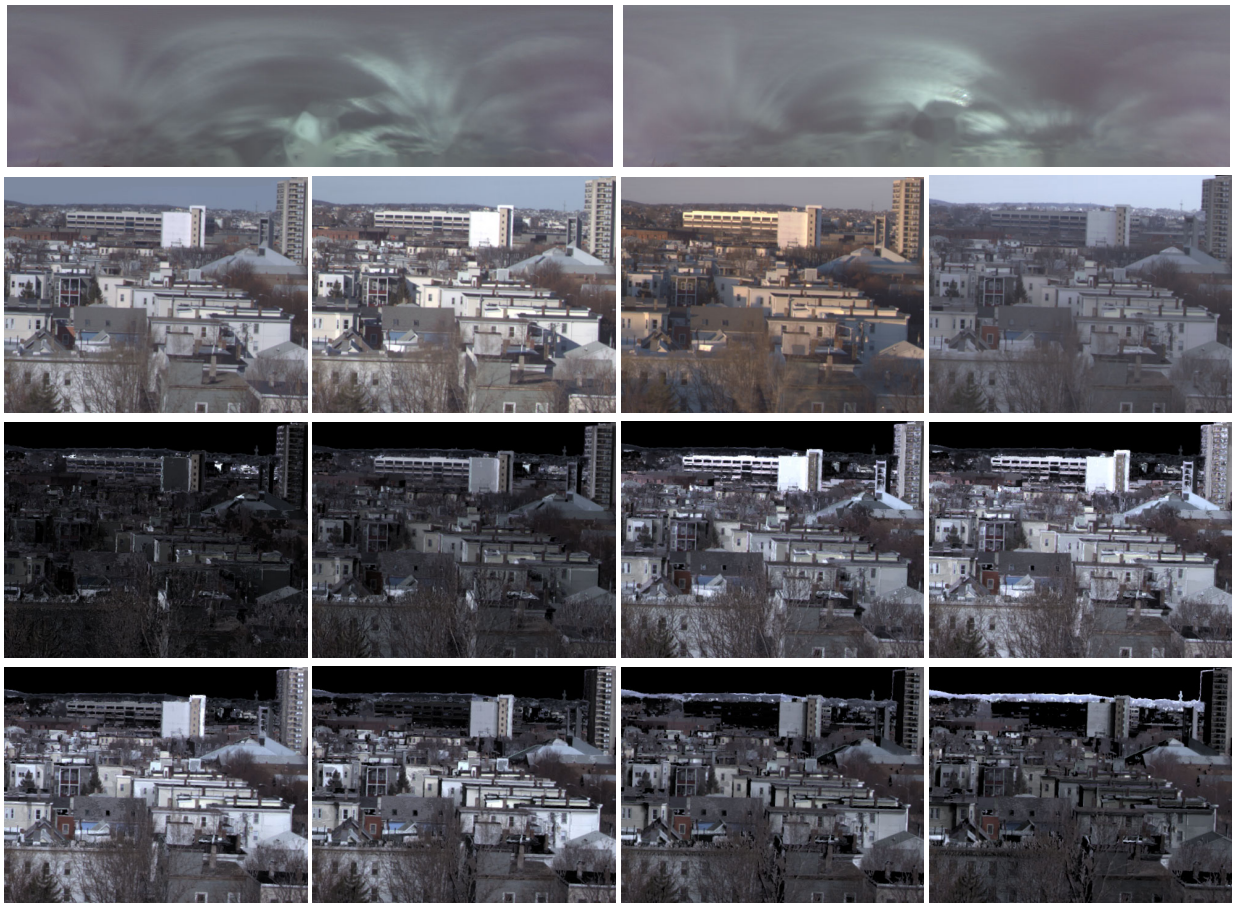

Figure 9: Relighting the city of Cambridge. Top row: Two example input latitude-longitude maps of the sky. Second row: Example observed images of the city. Bottom rows: The city relit with synthetic illumination (vertical white bar, moving from left to right). Note the moving shadows.

sliding left to right across the illumination domain (i.e., the latitude-longitude map). Note how the shadows on the buildings move. We are obviously not able to compare the predictions with real observations. Actually, to the casual observer these images look strange, since we are used to seeing outdoor scenes under full-hemisphere natural illumination. As far as we know, this is the first time that a city can be relighted arbitrarily without any scene geometry.

8. Future Work and Conclusions

We presented a novel representation and method for the acquisition of reflectance functions. Our algorithm works with natural incident illumination. It is simple, works robustly for all light frequencies, and progressively refines the approximation. Moreover, the underlying representation is compact and it allows for fast rendering with complex illumination. We tested our algorithm using two setups for indoor and outdoor scenes, respectively.

We envision different acquisition systems. For example, a single camera mounted on a rotating platform. The camera first acquires images of the incident illumination while rotating around a fixed point. During another rotation the camera acquires images of an object that has been placed onto the rotating platform. Another possible acquisition system consists of a dome (or cube) that is capable of generating high frequency illumination from all directions using, for example, several projectors.

An alternative to spatial correction would be to estimate the reflectance function on a low-resolution image of the scene, and propagating the kernel information to higher-resolution images as the algorithm proceeds.

The reflectance function does not include any dependence on the observation direction, which means it can not represent view-dependent effects such as specular highlights. However, Matusik et al. [MPN*02, MPZ*02] have shown that a combination of approximate geometry, reflectance functions from multiple viewpoints, and image-based ren-

dering can effectively be used to represent 3D objects of arbitrary complexity and appearance. We believe that one could also apply the method described in this paper to multiple viewpoints for the relighting of 3D image-based models.

9. Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments, Jan Kautz and Barb Cutler for comments on an early version of this paper, and Jennifer Roderick Pfister for proofreading the paper.

References

[BN76] BLINN J., NEWELL M.: Texture and reflection in computer generated images. *Communications of the ACM (SIGGRAPH 76 Proceedings) 19*, 10 (Oct. 1976), 542–547. 3

[CZH*00] CHUANG Y.-Y., ZONGKER D., HINDORFF J., CURLESS B., SALESIN D., SZELISKI R.: Environment matting extensions: Towards higher accuracy and real-time capture. In *Computer Graphics* (2000), SIGGRAPH 2000 Proceedings, pp. 121–130. 3

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

[DM97] DEBEVEC P., MALIK J.: Recovering high dynamic range radiance maps from photographs. In *Computer Graphics* (Los Angeles, CA, 1997), SIGGRAPH 97 Proceedings, pp. 369–378. 6

[GBK99] GEORGHIADES A., BELHUMEUR P., KRIEGMAN D.: Illumination-based image synthesis: Creating novel images of human faces under differing pose and lighting. In *IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes* (Apr. 1999), pp. 47–54. 2

[HCD01] HAWKINS T., COHEN J., DEBEVEC P.: A photometric approach to digitizing cultural artifacts. In *2nd International Symposium on Virtual Reality, Archaeology, and Cultural Heritage* (Glyfada, Greece, Nov. 2001). 2

[KPS02] KAUTZ J., P.SLOAN, SNYDER J.: Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. In *Eurographics Workshop on Rendering* (2002), pp. 291–296. 3

[LK03] LEHTINEN J., KAUTZ J.: Matrix radiance transfer. In *Symposium on Interactive 3D Graphics* (Apr. 2003), pp. 59–64. 3

[MDA02] MASSELUS V., DUTRÉ P., ANRYS F.: Free-form light stage. In *Proceedings of the Eurographics Workshop on Rendering* (Pisa, Italy, June 26–28 2002), Debevec P., Gibson S., (Eds.), pp. 247–255. 2

[MGW01] MALZBENDER T., GELB D., WOLTERS H.: Polynomial texture maps. In *Computer Graphics* (Los Angeles, CA, 2001), SIGGRAPH 2001 Proceedings, pp. 519–528. 2

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

[MPN*02] MATUSIK W., PFISTER H., NGAN A., BEARDSLEY P., ZIEGLER R., MCMILLAN L.: Image-based 3D photography using opacity hulls. *ACM Transaction on Graphics 21*, 3 (July 2002), 427–437. 2, 9

[MPZ*02] MATUSIK W., PFISTER H., ZIEGLER R., NGAN A., MCMILLAN L.: Acquisition and rendering of transparent and refractive objects. In *Proceedings of the 13th Eurographics Workshop on Rendering* (Pisa, Italy, June 2002). 3, 9

[NRH*77] NICODEMUS F., RICHMOND J., HSIA J., GINSBERG I., LIMPERIS T.: Geometric considerations and nomenclature for reflectance. Monograph 160, National Bureau of Standards (US), October 1977. 1

[NRH03] NG R., RAMAMOORTHI R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics 22*, 3 (2003), 376–381. 3

[PD03] PEERS P., DUTRÉ P.: Wavelet environment matting. In *Proceedings of the Eurographics Symposium on Rendering* (2003), Eurographics Association, pp. 157–166. 3

[RH01] RAMAMOORTHI R., HANRAHAN P.: An efficient representation for irradiance environment maps. In *Proceedings of SIGGRAPH 2001* (2001), pp. 497–500. 3

[SHHS03] SLOAN P.-P., HALL J., HART J., SNYDER J.: Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics 22*, 3 (2003), 382–391. 3

[SKS02] SLOAN P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transaction on Graphics 21*, 3 (July 2002), 527–536. 3

[WFZ02] WEXLER Y., FITZGIBBON A., ZISSERMAN A.: Image-based environment matting. In *Proceedings of the Eurographics Workshop on Rendering* (Pisa, Italy, June 26–28 2002), Debevec P., Gibson S., (Eds.), pp. 289–299. 3

[ZWCS99] ZONGKER D., WERNER D., CURLESS B., SALESIN D.: Environment matting and compositing. In *Computer Graphics* (Aug. 1999), SIGGRAPH 99 Proceedings, pp. 205–214. 2, 3, 4