Capturing a Surface Light Field Under Virtual Illumination

Greg Coombe Jan-Michael Frahm Anselmo Lastra
University of North Carolina at Chapel Hill

Abstract

Surface light fields can be used to render the complex reflectance properties of a physical object. One limitation is that they can only represent the fixed lighting conditions of the environment where the model was captured. If a specific lighting condition is desired, then there are two options: either use a combination of physical lights as an approximation, or capture a full 6D surface reflectance field and only use the portion that corresponds to the desired lighting.

In this paper we present a method for capturing a surface light field using the virtual illumination from an environment map. We use a simple setup consisting of a projector, a camera, a pan-tilt unit, and tracking fiducials to recreate the desired lighting environment. To decrease noise and improve the quality of the capture under low- and high-dynamic range environment maps, we use an extended version of the multiplexed illumination algorithm. We show results from objects captured under different lighting environments.

1 Introduction

Surface Light Fields (SLFs) [14, 21] are image-based representations of lighting for the capture and display of complex, view-dependent illumination of real-world objects. SLFs are constructed by capturing a set of images from different positions around an object. These images are projected onto the surface of a known geometric model and compressed [3]. This parameterization results in a compact representation that can be rendered at interactive rates.

A SLF is a 4D function that represents the exitant radiance under fixed illumination conditions. However, it can only represent the lighting of the environment where the model was captured. This is problematic for synthetic environments such as games or virtual environments in which a specific illumination environment is desired. These virtual illumination environments could come from light probes captured in real locations, or from synthetic lighting environments created by artists. Examples of lighting environments are shown in Figure 2.

To achieve correct object appearance the lighting environments must be physically duplicated in the lab at the time of capture, which can be difficult. One approach is to use lights or projectors that are physically situated to mimic the virtual lighting positions and colors. This approach is constrained to the resolution of the physical lights, and is time-consuming to construct. Another approach is to collect the full 6D Bidirectional Texture Function, which enables the object to be rendered under arbitrary lighting conditions. This requires a significant increase in the amount of data that is acquired, most of which is unnecessary if the lighting environment is already known.

In this paper we describe a third approach: an efficient method for capturing a surface light field using the virtual illumination from an environment map. We use a simple setup consisting of a projector, a camera, tracking fiducials, and a pan-tilt unit to recreate the desired lighting environment. To decrease noise and improve the quality of the capture under low- and high-dynamic range environment maps, we use an extended version of the multiplexed illumination algorithm [19]. This results in a high-dynamic range SLF which accurately represents the interaction of the virtual illumination with the real object. Two examples of objects embedded into a virtual light environment are shown in Figure 1.

The remainder of the paper is organized as follows. The
2 Related work

In this section we briefly review related work in the areas of high-dynamic range illumination, the sampling of material properties from objects, and instruments similar to ours. We then give an overview of multiplexed illumination.

Since the range of illumination present in the world is much larger than the range that can be reproduced by displays or captured by cameras, we need to use high-dynamic range (HDR) imaging techniques. Research in this area was pioneered by Debevec [9] in a paper that described how to linearize the response of cameras and combine multiple exposures into a single HDR image. Debevec also describes a technique for illuminating synthetic objects under HDR illumination [9]. A recent book serves as an excellent reference to the body of work surrounding HDR imaging [18]. Examples of HDR light probes are shown in Figure 2.

There has been somewhat less work on the lighting of real objects under virtual, user-specified illumination. One way to do this is to capture the BRDF of the materials. A survey of this work was presented in [15]. Many of these approaches attempt to reconstruct a 6-dimensional (or higher) function, which requires complicated equipment and considerable time to sample. In this paper we capture the 4D surface light field [21], which naturally takes fewer samples to estimate.

Instruments designed to capture BRDFs and reflectance fields of objects can collect surface light fields also. Our system could also be used to capture BRDF and reflectance fields. The light stage presented in [8] used several cameras and a lamp mounted to a gantry that could move the lamp over the hemisphere around a person. The main advantage to the system we describe is cost; we estimate that a total of $4000 is enough to buy the required equipment, although in fact most labs already have this gear. Furthermore, our camera-based calibration does not require precise positioning, so we are able to use inexpensive tripods to hold the camera and the pan-tilt motor.

The system most similar to ours is that of Masselus, et al. [12]. They used two plasma panels, six cameras, and four Halogen lamps in their instrument. They fixed a camera to a turntable with an object at the center, and rotated both under a projector. Thus they were able to capture the response to a light field for a single view and then relight the object. The main limitation is that the object to camera relationship was fixed. They also lit the object directly from the projector, thus obtaining a very differing set of illumination rays than those obtained by our system.

2.1 Multiplexed Illumination

Capturing images under dim lighting is difficult due to the presence of camera CCD noise. This noise can significantly degrade the quality of the image due to the low signal-to-noise ratio. Schechner et al. [19] introduced a technique to significantly reduce the noise in the captured images with multiple low intensity light sources. Using $n$ light sources, we can increase the signal-to-noise ratio by up to $\sqrt{n}$ with the same number of images. We briefly describe the Multiplexed Illumination algorithm in order to provide background on our HDR approach.

Consider the problem of acquiring images of an object that are lit from a set of light sources. A reasonable approach would be to acquire one image of the object for each light source. For $n$ light sources, this means that each image receives only $1/n$-th of the total available light. In addition, when each of the light sources are dim, the noise from the CCD cameras can corrupt the images. Consequently, for many of the light probes that we use, the signal-to-noise ratio (SNR) of the captured images is very low.

Multiplexed illumination is a technique to improve the SNR of the images. Each image is captured using multiple lights, and a post-process is performed to demultiplex the contribution of each individual light source. The light are additive quantities and are linearly related by superposition

$$\begin{bmatrix} a_0(x,y) \\ a_1(x,y) \\ \vdots \\ a_m(x,y) \end{bmatrix} = W \begin{bmatrix} i_0(x,y) \\ i_1(x,y) \\ \vdots \\ i_m(x,y) \end{bmatrix}$$

where $a_k(x,y)$ is the light observed at pixel $(x,y)$ under the set of lights $\zeta_k$ and $i_l(x,y)$ is the energy contributed by light source $l$ at pixel $(x,y)$. The multiplexing matrix $W$ for

![Figure 2. Sample light probes. Left and middle: Real light probes captured from St. Peter's Cathedral and Uffizi Gallery, courtesy of Paul Debevec (from debevec.org). Right: Synthetic light probe created by the artist Crinity.](image-url)
the light sources $l = 1, \ldots, m$ describes which light sources illuminate the scene. An element $W_{i,j}$ is one if the light $j$ is illuminated in image $i$, and zero if the light was “off”. The sets $\zeta_k$ consists of all of the lights in row $k$ which are “on”.

In order to recover the images $i_l(x, y)$ as lit under a single light source $l$, we demultiplex the observed images $a(x, y)$ by inverting the matrix $W$:

$$i_l(x, y) = W^{-1}a(x, y)$$

It’s important to note that multiplexed illumination does not require taking any more images than single light illumination. The only added computation is the post-process demultiplexing step.

3 System Overview

Our proposed method to capture a SLF under virtual lighting requires modest equipment and infrastructure. A diagram of our setup is shown in Figure 3. Light is projected onto a screen and reflected onto the object, which mimics the light from the light probe falling onto the object. In this way the projector and screen act as large area programmable light source. This is in contrast to a system which directly illuminates the object with the projector, as such a system would only represent a small portion of the lighting environment. This is because the projector essentially becomes a point light source, as shown in Figure 3. In addition, this configuration would require a rig to move the projector around the object to approximate the desired spherical lighting environment [12].

A photograph of our laboratory setup in shown in Figure 4. While the screen covers a large area of the lighting environment, it does not cover the entire environment. In order to recreate a given virtual lighting environment, the light source must illuminate the object from every possible direction. This would require moving the projector, screen and the camera around the object as we acquire images. However, instead of moving the camera as in similar systems [5], we move the object itself using a programmable pan/tilt unit. Moving the object simplifies the setup since the screen and projector can be fixed in place and calibrated once. A static setup of camera and screen also simplifies the problem of the camera (and user) occluding the light.

As the object rotates and tilts, the corresponding portion of the lighting environment changes. From the point-of-view of the object, it is as if a “window” is moving around it, allowing light from the environment to hit the object. In order to maintain a fixed orientation in the fixed lighting environment, we need to track the object’s orientation relative to the camera. This information is also needed to correctly composit the images together. We use fiducials to track the orientation, which can be seen in Figure 4.

The lighting environment can be arbitrarily complex, and can be either fully synthetic or captured from a real scene as a light probe [9]. Examples of light probes that we used in our experiments are shown in Figure 2. Since these light probes often have dramatic contrast between the darkest and brightest areas, we developed a novel approach to create a high dynamic lighting environment using a low dynamic range camera and a low dynamic range projector. Our technique is an extension of Multiplexed Illumination [19], which was developed in the computer vision community to reduce the noise in acquired images. We describe this technique in Section 6, after we discuss the calibration and registration of the system.

4 System Calibration

To successfully illuminate a real object with a virtual lighting environment, the mapping between the physical setup and the virtual lighting environment must be determined. This requires calibrating two transformations; first, the static transformation between light rays that are emitted from the projector and reflected off the screen onto the lighting stage. The second transformation is a dynamic correspondence between the camera and the object on the pan-tilt unit. In this section, we describe the first correspondence,
which depends mainly on the projector characteristics and the screen geometry. The result of this calibration is a mapping between pixels on the screen and points on the surface of the object. This mapping is used to determine how to display the lighting environment on the screen to achieve correct illumination in the desired lighting environment, as shown in Figure 5.

The calibration of the system registers the virtual lighting environment to the real physical object. This involves four components: the camera, the tracking fiducials, the physical object, and the projected image on the screen. The camera is fixed in relation to the screen, and the fiducials are fixed in relation to the object.

Figure 4. A picture of the Surface Light Field capture system. The light is projected onto the screen and reflects down onto the object. The object is mounted on a tracking board, which is mounted on a pan/tilt device. The fiducial markers are used to estimate the object’s position and orientation relative to the screen. The black curtains on the walls minimize light scattering in the lab.

One of the challenges of this calibration method is that we would like to be able to use arbitrary screen geometries. For example, consider projecting images into the corner of a room. Since the screen covers a larger solid angle above the object, we can reduce the number of cameras required to capture the full hemisphere of incident light. These considerations led us to look for a general calibration method that does not make assumptions about the geometry of the projected image.

The general screen geometry poses a significant challenge for the calibration process. Further problems arise from the deviation of the projector from an ideal projection due to aperture and lens distortions. These factors imply that physically measuring the system is difficult and often inaccurate. To account for all these effects we need to choose a fully automatic calibration technique, as potentially every ray illuminating the object needs to be calibrated separately.

Our method addresses these considerations by using a reflective sphere to register the pixels from the projector with a set of tracking fiducials from ARToolkit [11]. This procedure takes advantage of the property that light probes are independent of translation. Thus the calibration procedure needs to only compute the rays emanating from the object, and does not need to compute the translation information. This simplifies the calibration procedure to one of determining the relation between a pixel on the screen and a ray in object space.

We place a mirrored sphere with a measured radius in approximately the same location as the object. This mirrored sphere reflects the illuminated points on the screen back to the camera. Using the known projector pixel and the reflection of this point into the camera, the ray associated with each projector pixel can be computed. This is shown in Figure 6.

The process works as follows. A small block of pixels is projected onto the screen, which reflects off the mirrored sphere and back to the camera. Using the position of the pixels in the image, the ray through the camera plane into the scene can be computed from the pose of the tracking board. This ray is then traced into the scene and intersected with the sphere. At the intersection point, the normal is computed and a reflected ray is generated. This reflected ray is the ray in world space that corresponds to the projector pixel.

This automatically establishes a correspondence between projector pixels and rays in the scene. For rendering, these rays are rotated according to the delta rotation of the tracking board, and used to index into the environment map. The resulting images for the projector are shown in

Figure 5. Mapping the virtual illumination environment to our physical setup. Left: The virtual lighting environment. Right: The image that is projected onto the screen. Note that the image is stretched upward and outward to represent the rays of light from the virtual illumination environment.
Figure 6. The reflective object calibration process, shown for a planar screen on the left and a corner screen on the right. Top: A reflective calibration object is placed in the same location as the object. Pixels are projected onto the screen and segmented from the images. Bottom: The reflected rays are used to index into the environment map, producing an image which is correct from the point-of-view of the object.

Figure 6. Alternatively a more advanced structured light pattern can be used to determine all correspondences from very few images.

4.1 Projector Color Calibration

One of the sources of error in our setup is the color mapping of the projector, which is composed of aperture, lens, and CCD mapping of the colors. It is not correct to assume that the light coming from the projector is linearly correlated with the values that are sent, even with the gamma set to 1.0 and the controls adjusted. This is important because both the HDR exposures and the Multiplexed Illumination assume that the light can be linearly combined (discussed in Section 2). Without correcting for this non-linearity, the color of the resulting images would be incorrect.

We first calibrated the color response of the camera using the Camera Color Calibration Toolkit [10]. We established the reference colors by placing the Macbeth color checker in the direct light of the projector. Once the camera was calibrated for this light level, we then displayed a series of pure and mixed colors and captured images. By averaging over a large area of these images, we obtained a set of input color levels and their corresponding measured output color levels. Inverting this mapping gives us the necessary input color to produce a desired output color. Using these values, we fit a curve of the form \( c = \alpha_1 x^2 + \alpha_2 x + \alpha_3 \) to each RGB channel separately. A quadratic curve was chosen because it is simple to fit and can be inverted in graphics hardware by selecting the positive root of the quadratic equation. We first attempted to fit an exponential curve to the data, but were unable to obtain good results due to a knee in the data at the lower values.

5 Multiple Cameras

In order to properly capture the appearance of the object under the synthetic lighting conditions, the projector-camera system must be able to display the entire hemisphere of incoming light. However, due to the limited field of view, a single camera can only capture a section of the hemisphere, as shown in Figure 7. Any object captured under this physical setup would only be lit by a portion of the environment.

5.1 Coverage

To capture the full environment, we need to add more cameras, which are situated so that they can capture a broader portion of the hemisphere. In Figure 7 four cameras have been placed around the object to capture more light. In essence, each camera is capturing the light which has bounced off the object in a different direction. We initially used the visualization in Figure 7 to attempt to manually position the cameras for the optimum coverage, but we determined that it was easier to casually align the cameras and account for the positioning in software.

5.2 Integrating Multiple Cameras

Once a set of images has been captured for each camera position, they need to be combined to generate a single surface light field. If the portions of the hemisphere do not overlap, the light is independent and can be simply summed to get the final result. However, since the cameras are located in different positions it is unlikely that their projected positions will line up.

Instead, we combine the images in the resampling stage, which is performed later in the pipeline. After the visibility has been computed the camera positions are projected onto the coordinate system of each vertex and the points are resampled using a Delaunay triangulation. The values are resampled onto a fixed grid, which is the same for every camera. At this stage we can add the values from the different cameras together without incurring any more projection error than the resampling already creates.
6 High-Dynamic Range

To accurately model an object under a virtual illumination environment, we need to be able to handle the high-dynamic range of light. This requires mapping the high-dynamic range values from the light probe onto the low-dynamic range of the projector. A common technique is to split the high-dynamic range light probe into multiple exposure levels, which can then be recombined to form the full dynamic range. We use the technique by Cohen et al. [4] and split the energy into discrete levels, each of which fits within the 8-bit range of the projector. We use the function

$$\text{Color}_E = \frac{1}{10} \left\lfloor \frac{\text{Color}}{10E} \right\rfloor$$

with the exposure level $E \in [-2...3]$. Any color values outside the range $[0...1]$ are ignored. This mapping enables us to reconstruct the full dynamic range of the light probe as a sum of scalar multiples of the exposure levels.

Once this mapping is established, we can capture a set of images using the different exposure levels and scale the images to get the high-dynamic range result. However, the problem with this approach is that most of the light energy is concentrated at a few pixels in the light probe (the bright light sources). This means that very little light is falling on our object for the high-intensity light levels. If we capture an image under this light, we need to scale it by a large multiplier to get a high-dynamic range result. This also scales the CCD noise, which causes serious artifacts in the final image. A similar problem was addressed in the multiplexed illumination [19] work. In this section we describe how to extend multiplexed illumination to high-dynamic range lighting.

6.1 Multiplexed illumination for High-Dynamic Range Images

In this section we discuss how to apply multiplexed illumination to high-dynamic range images. The approach from [19] can only be applied to one exposure level at a time since the relationship across exposure levels is nonlinear (Equation 1).

The main idea of multiplexed illumination is to reduce the additive independent noise of the camera’s CCD array in low-light situations by collecting multiple measurements per image. Our approach tiles the projector screen into multiple regions $i_0, \ldots, i_m$ where each region displays a different exposure level (see Figure 8). As previously mentioned, much of the high-intensity lighting is concentrated in a small number of pixels. By splitting the screen into multiple regions, we can balance the amount of light that falls on the object, which improves the quality of the acquired images.
Each region on the screen is a light source $l$ displaying a part of the light probe with exposure level $E_t$. Demultiplexing these images requires that we account for light sources that have different exposure levels. However, this means that the coefficients $W_{k,l}$ of the weight matrix $\tilde{W}$ for multiplexed high-dynamic range illumination are no longer binary. They now model the exposure level of the region $W_{k,l} = 10^{-E_t}$ with $t = 0, \ldots, h - 1$ enumerating the different exposures $E_t$

$$a(x, y) = \tilde{W} \begin{bmatrix} i_{0,E_0}(x, y) & \cdots & i_{m,E_0}(x, y) & \cdots & i_{m,E_{h-1}}(x, y) \end{bmatrix}^T,$$

(2)

where $i_{t,E_t}(x, y)$ with $t = 0, \ldots, h - 1$ is pixel $(x, y)$ in the image of light source $l$ with exposure level $E_t$ and $a_k(x, y)$ is the $k$-th image. Hence the contribution of each light source at all exposure levels $E_t$ at pixel $(x, y)$ can be computed by

$$i_h(x, y) = \tilde{W}^{-1} a(x, y).$$

Equation (3) is an extension to handle multiple dynamic ranges of the light sources. A sample image used for high-dynamic range illumination is shown in Figure 8.

To compute the contribution of each light source from the captured images, the weight matrix $\tilde{W}$ has to be inverted. The efficient inversion from [19] can not be directly applied to the weight matrix $\tilde{W}$ for the high-dynamic range illumination since these values are no longer binary. Additionally the high range of entries $W_{k,l}$ of the exposure levels for high dynamic range images leads to a high condition number of $\tilde{W}$. This poses numerical problems for a direct inversion of $\tilde{W}$.

To invert this matrix, we note that the illuminations of the light sources within each exposure level are additive quantities, but they are not additive across exposure levels. Hence the matrix $\tilde{W}$ consists of blocks corresponding to the different exposure levels. Each of the blocks can be written as

$$\tilde{W}_{tm:(t+1)m-1,th:(t+1)h-1} = 10^{-E_t} W,$$

(4)

where $\tilde{W}_{tm:(t+1)m-1,th:(t+1)h-1}$ is the sub matrix of $\tilde{W}$ from row $tm$ to $(t+1)m-1$ and columns $th$ to $(t+1)h-1$. According to (4) the inversion scheme from [19] can be applied to each of the sub matrices $W_{tm:(t+1)m-1,th:(t+1)h-1}$

$$\tilde{W}^{-1}_{tm:(t+1)m-1,th:(t+1)h-1} = 10^{E_t} W^{-1}.$$  

(5)

This re-ordering of the matrix results in a new weight matrix $\tilde{W}$ for high dynamic range illumination that can be inverted as efficiently as before. To simplify the demultiplexing computation, the same sequence of exposure levels is displayed for each position. This means that the weight matrix $\tilde{W}$ only has to be inverted once. Furthermore the computation can be done online after the capture of image $a_k$ by multiplying the $k$-th column of $W^{-1}$ with $a_k$. This means that we can stream the images $a_k$ through memory and never have to store them.

7 Implementation and Results

We use a 1024x768 resolution projector pointed at a 4x3 foot white screen as light source. The printed fiducial board is placed atop a pan-tilt unit from Directed PerceptionTM. A photograph of our laboratory setup is shown in Figure 4. The geometry of the object is captured as a preprocess using a FaroTM digitizing arm. The 3D point samples (usually around 1000-3000 samples) are triangulated using the constrained 3D Delaunay mesh generator Triangle [20]. This triangulation is then loaded into Blender [1] or MeshLab [13] for cleanup and refinement. In addition to the 3D

Figure 8. Multiplexing the St. Peter’s Cathedral light probe (light probe courtesy of debevec.org). Left: Low-dynamic range multiplexing [19]. The screen is divided into 9 regions, where each region is either “on” or “off”. Approximately half of the regions are “on” at a time. Right: High-dynamic range multiplexing. Each region represents a different exposure level.

Figure 9. Different HDR levels of the heart model illuminated with the St. Peter’s Cathedral light probe.
model points, the 3D locations of the fiducial markers are also acquired. This enables the model to be registered with the tracking system.

The imaging device that we use is a Point Grey Research Flea™ video camera. This camera captures 1024 × 768 color images at 30 frames per second. The advantage of a video camera over a digital still camera is the higher speed of data transfer. The camera was calibrated with Bouguet’s Camera Calibration Toolbox [2], and images are rectified using Intel’s Open Source Computer Vision Library [16].

The Surface Light Field processing was implemented in an open source software package called OpenLF [17], developed by Intel and based on the research by Chen et al. [3]. Since these components were developed for standard 8-bit images, the extension to high-dynamic range floating-point images required extensive software modifications.

7.1 Results

We captured two objects, a pitcher model and a heart model on our capture rig under a number of different illumination environments, as shown in Figures 1 and 9. Capturing a low-dynamic range surface light field from 80 viewpoints took about 40 minutes, and OpenLF processing took another 20 minutes. The high-dynamic range capture multiplies the capture time by the number of exposure levels (we used five exposure levels in our experiments). For the heart model shown in Figure 9, the capture took about 4 hours. The light field processing time is only slightly changed, as the demultiplexing recombines these exposure levels into a single high-dynamic range image. There is a small added computational burden as the HDR images are stored as 32-bit floating-point rather than 8-bit fixed-point images. These timings are approximate, as we did not optimize the application for speed. The measurement was dominated by exposure times (typically about 0.5s per image) and movement of the pan-tilt unit. Using multiple cameras would avoid duplicating much of this work.

As discussed in [6], the OpenLF system has problems with missing data. In the virtual illumination system, this problem is exacerbated by the multiple camera positions. A triangle which is not visible in one camera view may be visible in another, which results in a data mismatch which must be corrected in a post-process. Using the Incremental WLS system [6] would address this problem.

7.2 Error

There are two dominant sources of error that we have measured in our system. The first is error from the calibration system, and the second is missing light from the user or camera blocking the projector.

<table>
<thead>
<tr>
<th>Pixel</th>
<th>-2%</th>
<th>-1%</th>
<th>+1%</th>
<th>+2%</th>
</tr>
</thead>
<tbody>
<tr>
<td>(34, 131)</td>
<td>0.4064°</td>
<td>0.2011°</td>
<td>0.1971°</td>
<td>0.3902°</td>
</tr>
<tr>
<td>(151, 63)</td>
<td>2.0260°</td>
<td>0.9988°</td>
<td>0.9719°</td>
<td>1.9179°</td>
</tr>
<tr>
<td>(-18, 156)</td>
<td>0.0919°</td>
<td>0.0455°</td>
<td>0.0446°</td>
<td>0.0883°</td>
</tr>
<tr>
<td>(-72, 154)</td>
<td>0.7546°</td>
<td>0.3733°</td>
<td>0.3655°</td>
<td>0.7235°</td>
</tr>
</tbody>
</table>

Table 1. Angular error vs. relative radius error

7.2.1 Calibration

The primary source of error in this calibration procedure is from the physical measurement of the sphere. We measured the error in the reflected rays as a function of the error in the radius of the sphere for several projector pixels. These results are shown in Table 1. Pixels near the edge of the sphere are more susceptible to error, since the normals change more quickly. In general, there is about 1° of angular error for 1% radial error. Other sources of error which we did not measure are camera pose estimation and segmentation of the illuminated pixels from the reflection on the sphere.

Currently, we use a 9x9 sampling of the screen (100 calibration points) aligned on a regular grid. For screens which are mostly planar (including our corner screen), this is sufficient. However, for more complex screens a denser sampling would be needed.

7.2.2 Blocked Light

Figure 10 shows the camera blocking part of the screen, thereby reducing the amount of light that falls on the object. This is a common problem in Image-Based Modeling since the acquisition devices often interfere with the lighting [8]. To quantify this error, we measured the area of the image that is covered by the camera, and compared it to the area that the screen covers. In the 1024 × 768 image shown, over 3.1 × 10^5 pixels are screen pixels, and 1.6 × 10^4 are camera pixels. This means that the camera covers about 5.2% of the light from the screen.

The image shown is the worst-case for camera position. When the camera is positioned to the sides or back, it does not block the light from the projector and the screen. For our system, it might be possible to mount the camera behind the screen so that it peeks through a small hole. This would minimize the light interference.

8 Conclusion

In this paper, we described a cost-efficient system which captures the surface light field of an object under virtual illumination from a light probe. It consists of a projector which shines light onto a screen and reflects onto the ob-
ject. This configuration enables us to capture a larger portion of the hemisphere than shining the projector directly on the object. To model the physical world, we used high-dynamic range light probes and mapped the values onto the 8-bit levels of the projector. To avoid amplifying noise, we extended the multiplexed illumination algorithm to high-dynamic range imagery.

8.1 Future Work

All of the multiplexing in our system is done in screen space. Another way to multiplex is to decompose the light probe into several light sources using an algorithm such as the Median Cut algorithm [7]. This would allow us to treat the bright spots in the light probe differently than the background areas. It also moves the demultiplexing from images into model space, which may result in fewer acquired images. This could be combined with an incremental Surface Light Field technique [5] to speed up the acquisition process.

There are several simplifying assumptions that we made. One of them is that the size of the object is small in relation to the screen. This constrains the capture to small objects (or requires large screens). This assumption was made to avoid calculating the individual rays from each point on the surface of the mesh model. However, since we have a coarse 3D model of the objects, it is possible to directly calculate these rays. It would be interesting to see whether the flexibility of capturing larger objects justifies the increased computational cost, or whether a different approach is needed. In the other direction, future research could extend the system to even lower-cost capture employing monitors and webcams.

References