Thirteenth Eurographics Workshop on Rendering (2002)
P. Debevec and S. Gibson (Editors)

Acquisition and Rendering of Transparent and Refractive Objects

Wojciech Matusik [†] Hanspeter Pfister [‡] Remo Ziegler[‡] Addy Ngan[†] Leonard McMillan[†]

Abstract

This paper introduces a new image-based approach to capturing and modeling highly specular, transparent, or translucent objects. We have built a system for automatically acquiring high quality graphical models of objects that are extremely difficult to scan with traditional 3D scanners. The system consists of turntables, a set of cameras and lights, and monitors to project colored backdrops. We use multi-background matting techniques to acquire alpha and environment mattes of the object from multiple viewpoints. Using the alpha mattes we reconstruct an approximate 3D shape of the object. We use the environment mattes to compute a high-resolution surface reflectance field. We also acquire a low-resolution surface reflectance field using the overhead array of lights. Both surface reflectance fields are used to relight the objects and to place them into arbitrary environments. Our system is the first to acquire and render transparent and translucent 3D objects, such as a glass of beer, from arbitrary viewpoints under novel illumination.

Categories and Subject Descriptors (according to ACM CCS): I.3.2 [Computer Graphics]: Picture/Image Generation – Digitizing and Scanning, Viewing Algorithms I.3.6 [Computer Graphics]: Methodology and Techniques – Graphics Data Structures and Data Types

1. Introduction

Reproducing real objects as believable 3D models that can be placed into arbitrary synthetic environments has been a longstanding goal in computer graphics. Traditional 3D scanning techniques, although capable of accurately capturing 3D shape, are not well suited to capture complicated object appearances of, for example, highly specular, fuzzy, or transparent objects [32]. Recent advances in image-based 3D photography have made it possible to faithfully reproduce the view-dependent effects of specular surfaces with self-shadowing [39, 27, 5], to capture object reflectance fields for relighting [8, 15], and even to reproduce the complex visual silhouettes of objects with hair, leaves, or feathers [25].

While there has been much work on capturing and representing surface radiance and surface reflectance, very little attention has been paid to the scanning of transparent 3D objects and the capturing of light refraction. This is not surprising, since traditional 3D scanning devices are not capable of handling transparency. A notable exception are the techniques of environment matting [41, 6] that accurately acquire light reflection and refraction from a fixed viewpoint.

We have built an image-based 3D scanning system that is able to simultaneously acquire surface radiance, reflectance, and refraction of arbitrary 3D objects. Our method is based on an extension of environment matting to multiple viewpoints. Our hardware setup includes two turntables, lights, and background monitors to display multi-colored patterns. We acquire images with high-resolution digital cameras from many viewpoints around the object. At each viewpoint, we capture alpha mattes, where α represents partial coverage of the pixel. We also acquire environment mattes and reflectance images that capture how light is refracted and reflected by the foreground object.

Using the alpha mattes, we first construct the 3D opacity hull of the object [25]. The opacity hull is an extension of the visual hull [20] with view-dependent alpha at each surface point. Our object representation parameterizes the environment mattes and reflectance images onto the surface. During rendering, we project surface points into the closest k-images and use a variant of unstructured lumigraph interpolation [3] to compute the resulting color for that surface point.

† MIT, Cambridge, MA. Email: [matusik,addy]@graphics.lcs.mit.edu
‡ MERL, Cambridge, MA. Email: [pfister,ziegler]@merl.com

Our method is able to faithfully render transparent 3D objects from novel viewpoints and in novel synthetic environments (see Figure 1).

Figure 1: Rendering a scanned object from a new viewpoint and with a new environment.

The novel contributions presented in this paper include:

- A scanning system for transparent objects that is able to automatically capture 3D shape, surface reflectance fields, and fine scale silhouette features;
- A high- and low-resolution parameterization for surface reflectance fields on 3D surfaces;
- A view-dependent interpolation procedure for environment mattes of transparent objects;
- An efficient datastructure and rendering algorithm that allows transparent objects to be appropriately composited and relit in a novel environment;

Despite using only an approximate visual hull shape, our representation is capable of correctly reproducing very complicated surface appearance. Since environment matting assumes illumination from an infinitely distant environment, our method is not capable of correctly reproducing interactions with near elements, e.g., specular and transmissive effects between two nearby objects. However, it accurately captures self-reflections, self-refractions, and many surface scattering effects.

2. Previous Work

There are many approaches for acquiring high quality 3D shape of opaque objects with a diffuse surface [1]. However, many real world objects have specular surfaces or transmit light, and traditional passive or active light scanners cannot capture their shape [15, 32]. Most current 3D scanning approaches do not take self-shadowing and self-reflection into account, and none is capable of dealing with large scale light refraction.

Zongker et al. [41] developed the techniques of environment matting to capture mirror-like and transparent objects and to correctly composite them over arbitrary backgrounds. Their system is able to determine the direction and spread of the reflected and refracted rays by illuminating a shiny or refractive object with a set of coded light patterns. They efficiently parameterize surface reflectance into 2D environment mattes. Extensions to environment matting include a more accurate capturing method and a simplified and less accurate procedure for real time capture of moving objects [6]. However, their system only captures environment mattes for a fixed viewpoint, and they do not reconstruct the 3D shape of the object. We use environment matting to capture the scattering of light from multiple viewpoints. We substantially extend their method to construct a 3D model from the data that can be rendered from arbitrary viewpoints and with new illumination.

There are many parametric appearance models for light reflection and subsurface transport. Most models only approximate the bidirectional reflection distribution function (BRDF) [26], which assumes that light entering a surface leaves the surface at the same point. Quite often, these parametric models have been carefully crafted to fit some measured observations. For example, inverse rendering methods estimate a model of the surface BRDF from images and acquired geometry of the object. Sato et al.[34] and Yu et al. [40] assume that the specular part of the BRDF is constant over large regions of the object. Lensch et al. [21] fit a Lafortune BRDF model to each point on the object surface. More recently, Jensen et al. [19] proposed a parametric model for the bidirectional surface scattering reflection distribution function (BSSRDF) [26], which describes the light transport between any two rays on the object surface. Their model is applicable to homogeneous translucent material and does not capture the large-scale refractions seen in transparent objects.

Parametric appearance models are often incapable of representing the complexity and richness of material effects seen in the real world. Objects featuring glass, fur, hair, cloth, leaves, or feathers are very challenging or impossible to represent with parametric models [15]. More importantly, real objects are imperfect, rough, scratched, dirty, and light scattering is non-uniform and often unpredictable. We encourage you to carefully observe the many appearance variations of a glass of beer under direct illumination, many of which are due to complicated participating media effects and material imperfections. The intrinsic perfection of synthetic objects and parametric appearance models is often criticized as the "clean look" of computer graphics. There has been some work on surface weathering and aging [11, 28], but these methods are still limited. As observed by Shirley et al. [35], there is currently no general-purpose reflectance model that is suitable for all situations.

An alternative is to use image-based, non-parametric representations for object reflectance. Marschner et al. [23] use

a tabular BRDF representation and measure the reflectance properties of convex objects using a digital camera. Their method is limited to simple materials with uniform BRDFs. Dana et al. [7] have sampled the bidirectional texture function (BTF) for spatially-varying surface reflectance. They have measured the BTF for many small planar patches of different materials. Unfortunately, until now, none of these data-based methods are capable of capturing the complexity of large and small scale light-material interactions in the real world.

To bridge the gap between physically plausible and interactive rendering, Newell and Blinn [2] introduced environment mapping to quickly find reflections of distant objects from a perfectly mirrored surface. Cabral et al. [4] and Heidrich et al. [17] extended this work by pre-integrating a BRDF with the environment map. Hakura et al. [14] introduced parameterized and layered environment maps to simulate local reflections and self-shadowing. More recently, they use hybrid rendering and multiple layered environment maps to handle refractive objects [13]. However, all these methods require a parametric surface reflectance model, and they cannot easily render the object under novel illumination or in new environments.

To efficiently represent objects with arbitrary light reflectance we have adopted an image-based capturing and rendering approach. Image-based representations have the advantage of capturing and representing an object regardless of the complexity of its geometry and appearance. Because we use observed image data we are able to represent a much wider class of light transport effects than previously possible. The tradeoff is, of course, that our models contain an image database, which requires much more storage.

Very few image-based rendering approaches correctly deal with transmissive effects. The light field [22] and lumigraph [12] methods represent the light field as a 4D function using a 2-plane parameterization. Heidrich et al. [16] extended this parameterization by mapping incoming viewing rays to outgoing reflected and refracted rays. They did not extend the technique to acquired light fields of real objects, although in principle they could use a similar acquisition approach as presented in this paper. However, their method does not construct a full 3D geometric model and is limited to static illumination.

We use a similar mapping from incoming viewing rays to outgoing reflected and transmitted rays, but for points on the object surface. Instead of a view-based parameterization this is a surface-based parameterization, similar to surface light fields [39, 27, 5]. However, images generated from a surface light field always show the object under a fixed lighting condition, and surface light fields cannot handle transmission and refraction of light rays.

In order to properly relight the object, we use a monitor and rotating lights to acquire the *surface reflectance field*. Debevec et al. [8] define the reflectance field of an object as the radiant light from a surface under every possible incident field of illumination. They use their Light Stage system to capture the reflectance field of a human face [8] or of cultural artifacts [15]. However, their system is limited to renderings from a single viewpoint.

To interpolate between images from different viewpoints we are using the unstructured lumigraph interpolation by Buehler et al. [3]. This is an extension of the view-dependent texture mapping work by Pulli et al. [31] and Debevec et al. [10]. We extend unstructured lumigraph interpolation to render environment mattes of specular and transparent surfaces from arbitrary viewpoints.

3. Overview

We first develop our model for how light scatters at the object surface following the exposition of Chuang et al. [6] and Debevec et al. [8]. Assuming that the incident radiation originates infinitely far away from the object surface, the light arriving at a camera pixel can be described as:

$$C = \int_{\Omega} W(\omega_i) E(\omega_i) d\omega. \tag{1}$$

C is the recorded color value at each camera pixel, and $E(\omega_i)$ is the environment illumination from direction ω_i. $W(\omega_i)$ is a weighting function that comprises all means of light transport from the environment through the foreground object to the camera. It corresponds to what Debevec et al. [8] call the reflectance function $R(\omega_i)$. The integration is carried out over the entire hemisphere Ω and for each wavelength. We will drop the wavelength dependency in the rest of this paper, assuming that all equations are evaluated separately for R, G, and B.

Given a measured camera intensity C at a pixel and an environment E, we want to estimate the function W for the point on the object surface corresponding to the ray through that pixel. Our scanning system provides two different illumination fields for the environment: illumination from a high-resolution 2D texture map behind the object (displayed on the plasma monitors), and illumination by the overhead light array from a sparse set of directions on the remaining hemisphere. Figure 2 shows the basic setup.

We call the sector of the environment hemisphere covered by the high-resolution texture map Ω_h, and the remaining sector covered by the light array Ω_l. Furthermore, we are making the simplifying assumption that light transport in Ω_h can be described by two components (see Figure 2a). As shown in the figure, we are approximating the (potentially complicated) paths of light through the object by two straight light bundles from the ray-surface intersection to the background monitor. On the other hand, light from one or more directional light sources $L(\omega_i)$ in Ω_l is refracted and reflected by the object before arriving at pixel C (see Figure 2b). Here we assume that the incident light field in Ω_l can be sampled at substantially lower resolution than light

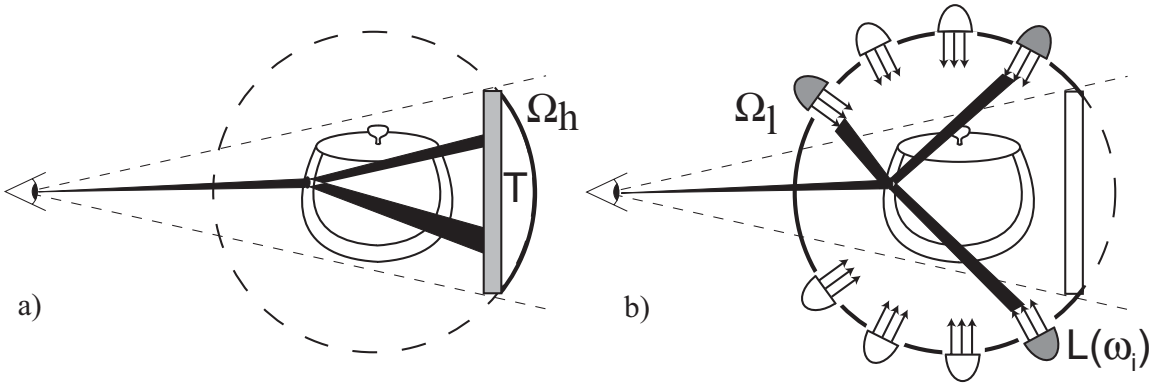

Figure 2: Illumination environment and light propagation model in our system. a) High-resolution sampling across Ω_h. b) Low-resolution sampling across Ω_l.

in Ω_h coming directly from behind the object. Thus Equation (1) becomes:

$$C = \int_{\Omega_h} W_h(x)T(x)dx + \int_{\Omega_l} W_l(\omega_i)L(\omega_i)d\omega. \quad (2)$$

As proposed by Chuang et al. [6], we are using a sum of Gaussians to describe $W_h(x)$. To find matching Gaussians during k-nearest neighbor interpolation, as described in Section 6, we restrict ourselves to a maximum of two Gaussians per surface point. Thus:

$$W_h(x) = a_1 G_1(x, C_1, \sigma_1, \theta_1) + a_2 G_2(x, C_2, \sigma_2, \theta_2). \quad (3)$$

G_1 and G_2 are elliptical, oriented 2D unit-Gaussians, and a_1 and a_2 are their amplitudes, respectively. x are the camera pixel coordinates, C_i the center of each Gaussian, σ_i are their standard deviations, and θ_i their orientations. See [6] for more details and figures.

Since we are sampling Ω_l with a discrete set of n light positions $L_i(\omega_i)$, we can rewrite Equation (2) as:

$$C = \int_{\Omega_h} (a_1 G_1 T(x) + a_2 G_2 T(x))dx + \sum_{i=1}^{n} R_i(\omega_i)L_i(\omega_i). \quad (4)$$

Using the environment matting and reflectance field procedures outlined in Sections 5.3 and 5.4, we estimate the parameters a, G (represented by C_i, σ_i, and θ_i) and R using observed data C. For each viewpoint, the estimated parameters are stored in an environment matte for (a_1, G_1, a_2, G_2), and n reflectance images for $R(\omega_i)$.

During rendering, we determine the visible surface points on the reconstructed shape S of the object. Each visible surface point is reprojected to look up the parameters a, G and R from the k-nearest environment mattes and reflectance images. We use unstructured lumigraph interpolation [3] to interpolate the parameters to the new viewpoint. The resulting parameters are then combined using Equation (4) to compute the new pixel color C_n.

Discussion Equation (4) is a compromise between high-quality environment matting [6] and the practical limitations of our 3D acquisition system. Ideally, one would surround the object with high-resolution monitors and acquire the parameters of an arbitrary number of weighting functions W distributed over multiple monitors. Instead, we assume that most of the refracted and reflected rays arriving at a pixel originate from the incident light field behind the object. This is true for most objects with a strong reflected component (mostly at grazing angles) and a strong refracted component. It is not necessarily correct for transparent objects with large-scale internal structure or surface facets, such as a crystal glass. However, in practice we found this approach to work reasonably well.

Using a high-resolution environment texture in viewing direction is superior to using only the light array to provide incident illumination [8, 25]. For example, Debevec et al. [8] use 2048 light positions, which corresponds to a 32×64 pixel environment map. Our previous approach [25] effectively only uses illumination from a 4×15 pixel environment map. These resolutions are not nearly enough to accurately capture and represent transmissive and refractive effects. For example, looking straight through a glass window shows the background in its full resolution. On the other hand, using a high-resolution illumination environment is only feasible with environment matting. The alternative would be to store a very large number of reflectance images for each viewpoint, which is impractical. Environment mattes are in essence a very compact representation for high-resolution surface reflectance fields.

It is important to note that, despite the term surface *reflectance* field, we are capturing a much wider array of effects, including refraction, dispersion, subsurface scattering, and non-uniform material variations. These effects, which are typically costly or impossible to simulate, can be rendered from our model in a reasonable amount of time. As noted by Debevec et al. [8], the surface reflectance field is al-

most equivalent to the BSSRDF. The main differences are that we do not know the exact physical location of a ray-surface intersection, and that the incoming direction of light is the same for any point on the surface. The first problem could potentially be addressed by improving the visual hull geometry using methods of stereopsis. Solving the second problem would require to illuminate the object with a dense set of laser point light sources, which is highly impractical.

Equation (4) differs from Equation (12) in Chuang et al. [6] by restricting the number of incoming ray bundles from the monitors to two, and by replacing the foreground color F with a sum over surface reflectance functions R_i. The first assumption is valid if reflection and refraction at an object causes view rays to split into two distinct ray bundles that strike the background (see Figure 2a). The second assumption results in a more accurate estimation of how illumination from Ω_l affects the objects foreground color. Chuang et al. [6] make up for this by capturing additional environment mattes using monitors on each side of the object.

This formulation of light transport improves our own previous work [25]. We are now correctly separating the effects of partial pixel coverage (α) and transparency. Previously, we used the monitors to capture alpha mattes (where alpha also captured transparency) and we only estimated the surface reflectance field using the overhead light array (i.e., $\Omega = \Omega_l$). By capturing high- and low-resolution surface reflectance fields we are now able to handle high frequency environment maps. The use of environment mattes, an improved interpolation scheme (see Section 6), and some improvements in alpha matte acquisition (see Section 5.1) allow us to acquire and render highly transparent objects that our previous system could not handle.

4. 3D Scanning System

We are using the hardware setup described by Matusik et al. [25] to acquire alpha mattes and the surface reflectance field from multiple viewpoints. Figure 3 shows an overview of our system. Objects are placed on a rotating turntable and a series of video cameras is pointed at the object from various angles. To facilitate consistent back lighting we mount the cameras roughly in the same vertical plane. A plasma monitor is placed directly opposite of the cameras. Another plasma monitor on the turntable illuminates the objects from below. The monitors are used to display color backdrops for alpha and environment matting.

An array of light sources is mounted on an overhead turntable. The lights are spaced roughly equally along the elevation angle of the hemisphere. We use the light sources to illuminate the object from the front and the sides in order to acquire the low-resolution surface reflectance field. We depend on good repeatability of the turntables to ensure that all images and light positions are well registered. Figure 4 shows a photograph of our system. More details about our hardware setup can be found in [25].

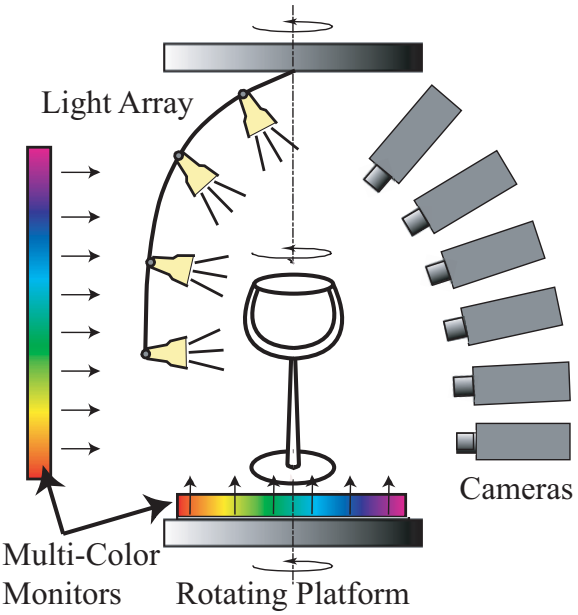

Figure 3: Our 3D digitizing system. Objects are rotated on a turntable with plasma display while images are acquired. The plasma monitors are used for environment matting. The overhead array of light sources can be rotated to acquire surface reflectance fields.

Figure 4: Photograph of our digitizing system.

5. Data Acquisition Process

After camera and system calibration [25], we acquire reference images from all viewpoints of the patterned backdrops displayed on the plasma monitor. These images only have to be acquired once after calibration and are stored for subsequent object scans.

Next, the object is placed on the turntable. For each viewpoint, we acquire the alpha and environment mattes as described in Sections 5.1 and 5.3, respectively. After the mattes are acquired, the array of lights is rotated around the object and we acquire the reflectance images as described in Section 5.4. Once all the images have been acquired for this turntable position, the object is rotated and the process repeats.

5.1. Alpha Mattes

Alpha mattes capture how light travels straight through pixels from the background map to the image [41, 6]. Ideally, each pixel's α value measures the partial occupancy of the object, and not the object's transparency. An accurate alpha matte allows the rendering system to accurately composite the object onto new backgrounds.

We have adapted the method of Smith and Blinn [36] to capture alpha mattes for each viewpoint. Of particular concern for specular and transparent objects is color *spill* [36], the reflection of backlight on the foreground object at grazing angles due to Fresnel effects. To minimize spill, we display a sequence of sinusoidal background patterns with different wavelengths. The patterns are displayed with and without the foreground object. If the observed color of a pixel is the same for each background, then we consider the pixel to map straight through. In total, we use six different backdrops to compute one final alpha matte per viewpoint. The details of this procedure are discussed by Matusik et al. [25].

To extract silhouette images for computation of the visual hull of the object, we use binary thresholding on the alpha mattes. We found that a threshold of $\alpha > \alpha_t = 0.05$ yields binary silhouette images that encompass all of the object.

However, we encountered one problem particular to very transparent objects: If light is not sufficiently refracted by the object (for example, for thin glass), pixels inside the object map straight through to pixels on the background. These pixels are assigned an alpha of zero, even though they are contained within the object silhouette. This leads to holes in the binary silhouette image, as shown in Figure 5a.

To correct this problem we use a simple greedy procedure that fills the holes. We call a pixel empty if its alpha value is below the threshold α_t. We fill regions of empty pixels if the number of connected empty pixels is below a user defined threshold n_e. For typical objects, n_e can range from 5 to 100. Figure 5b shows the resulting binary image.

Pixels inside the silhouette are assigned an alpha value of one. We assign the originally measured alpha values to pixels along the edges of the binary silhouette. Again, we use a simple greedy procedure. Pixels are considered to be inside the binary silhouette if there are no empty pixels surrounding them in some small neighborhood. We achieve good results with a four-pixel neighborhood. All other pixels are considered to be near the silhouette edge and are assigned the measured alpha values. Figure 5c shows the final alpha matte.

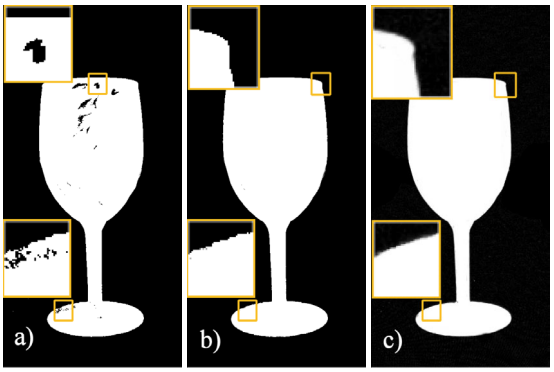

Figure 5: a) Silhouette image with holes due to transparency. b) Holes filled. c) Final alpha matte.

5.2. Opacity Hull Construction

Using the binary silhouette images of the object from various viewpoints, we first construct the image-based visual hull (IBVH) [24]. We re-sample the IBVH from three orthogonal directions into three layered depth images (LDIs). The LDIs are then used to construct an octree-based layered depth cube (LDC) tree [29]. To avoid self-occlusion, visibility of each surface point to each image is computed during IBVH construction using epipolar geometry [24]. Each surface point in the LDC tree stores depth, normal, and a camera-visibility bit vector. Each bit in the visibility vector corresponds to a viewpoint. A value of one indicates that the surface point is visible from that camera position.

To capture the view-dependent partial occupancy of a foreground object with respect to the background we use the opacity hull [25]. During rendering, each point on the visual hull surface is projected into the alpha mattes to determine its opacity. We use the procedure described in Section 6 to interpolate alpha for novel viewpoints.

5.3. Environment Mattes

To acquire environment mattes, we are using the high quality procedure by Chuang et al. [6]. The acquisition process involves taking multiple images of the foreground object in front of a backdrop with a 1D Gaussian profile that is swept over time in horizontal, vertical, and diagonal direction. Using the non-linear optimization procedure described

by Chuang et al. [6], we then solve for a and the parameters of the 2D Gaussians G.

To save storage and computation time for the non-linear parameter estimation, we identify and remove areas outside the object silhouette. The environment matte is subdivided into 8×8 pixel blocks. Each surface point on the opacity hull that is visible from this view is projected into the image. Only those blocks that contain at least one back-projected surface point are stored and processed.

For certain positions in the camera array, the rim of the plasma monitors is visible through transparent object, which makes much of the field of view unusable. Consequently, we only use the lower and the two upper most cameras for acquisition of environment mattes. The lower camera is positioned horizontally, directly in front of the background monitor. The two upper cameras are positioned above the monitor on the turntable. Using our environment matte interpolation (see Section 6), we can render plausible results for any viewpoint.

5.4. Reflectance Images

After acquisition of the alpha and environment mattes, we acquire the surface reflectance field [8, 15, 25]. To avoid specular reflections from the monitors we cover the vast majority of the display surface with black felt without upsetting the object position. For each viewpoint, the lights of the light array are turned on sequentially, then the array is rotated around the object. We use four lights and typically increment the rotation angle by 24°. Unlike our previous work [25], we only rotate the lights to cover Ω_l (see Figure 2b). We capture a total of 4×11 reflectance images for each camera position. This procedure is repeated for all viewpoints.

The raw reflectance image data would require about 76 GB of storage. Storing only the pixel blocks within the object silhouette still would require between 20 and 30 GB, depending on the size of the object. To make this data more manageable, we are using a simple compression scheme using principal component analysis (PCA). The reflectance images are subdivided into 8×8 image blocks. We apply PCA to all the blocks corresponding to the same viewpoint and varying illumination. We set a global threshold for the RMS reconstruction error. Each block is then stored as a variable number of eigenvalues and principal components. The average number of principal components is typically four to five per block if we set the global RMS reconstruction error to be within 1% of the average radiance values of all reflectance images. As shown by Matusik et al. [25], PCA analysis typically reduces the amount of reflectance data by a factor of 10.

6. Rendering Algorithm

We render the opacity hull models using elliptical weighted average (EWA) surface splatting [42]. A hierarchical forward-warping algorithm projects the surface points to screen space, where the EWA filter reconstructs the image. A modified A-buffer provides order-independent transparency blending and edge anti-aliasing.

Most of the rendering time is spent on computing the color for each point on the object surface. We start with a new environment map $\tilde{T}$, for example, a spherical high-dynamic range light probe image of a natural scene. We first reconstruct new reflectance images from the original viewpoints for this new illumination environment. Then we project the 2D Gaussians of the original environment mattes into the new environment map. (Note the difference between environment "mattes" and "maps".) To interpolate the alpha mattes, reflectance images, and environment mattes to new viewpoints, we use unstructured lumigraph interpolation [3]. The interpolated values are then used to evaluate Equation (4). We now discuss these steps in more detail.

6.1 Reflectance Image Reconstruction First, we compute new images from the acquired reflectance field data that show the object under the new light configuration. The new light colors and directions are specified by a low-resolution version of the environment map $\tilde{T}$. This low-resolution map must match the resolution of light positions in Ω_l [8]. In our case it contains only 4×11 pixels.

For each viewpoint we have a set of 8×8 reflectance blocks. Each block R is compressed using PCA analysis into:

$$R = \sum_{i=1}^{k} \gamma_i V_i, \tag{5}$$

where γ_i are the k eigenvalues we store for each block and V_i are its principal components. Given a new set of m directional lights $\tilde{L}_i$, we can compute the new colors C' for the block directly as a linear combination of the coefficients of the PCA basis [25]:

$$C' = \sum_{i=1}^{m} (\tilde{L}_i \sum_{j=1}^{k} \gamma_j V_j). \tag{6}$$

This direct computation avoids reconstruction of the reflectance data from the PCA basis. Note that we convert a set of reflectance images for each viewpoint into one radiance image that shows the object under the new illumination [8]. This computation is performed for each change of the environment map.

6.2. Environment Matte Projection Our acquired environment mattes are parameterized on the plane T of the background monitor. However, for rendering they need to be parameterized on the global environment map $\tilde{T}$. Figure 6 shows a 2D drawing of the situation. During system calibration we determine the position of each monitor plane T with respect to each viewpoint. This information is globally stored per viewpoint. $\tilde{T}$ is the parameterization plane of the new environment map. The mapping from T to $\tilde{T}$ may be

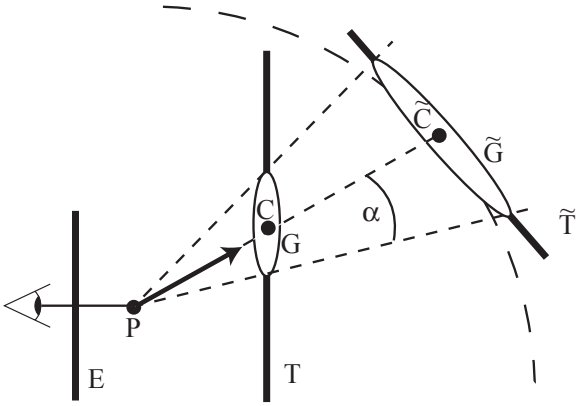

Figure 6: Reprojection of the environment matte Gaussian G from the monitor plane T into the environment map $\tilde{T}$.

non-linear, for example, for spherical environment maps. A 3D surface point P on the object is projected onto a pixel of the environment matte E, which stores the parameters of the 2D Gaussian G. We compute the Gaussian $\tilde{G}$ that best approximates the projected Gaussian G on the parameterized surface $\tilde{T}$.

We represent the new Gaussian $\tilde{G}$ using the following parameters: a (the amplitude of G), $\tilde{C}$ (a 3D vector), (α, β) (the opening angles), and $\tilde{\theta}$ (the new rotation angle). This projection is performed for each change of the environment map.

6.3. Alpha and Radiance Interpolation For each surface point, we compute the k-nearest ($k = 4$) visible viewpoints using the point's position and the global camera parameters. As mentioned in Section 5.2, visibility is determined during opacity hull construction and stored in the visibility vector. We compute the interpolation weights w_i for the four closest viewpoints according to unstructured lumigraph interpolation [3, 25]. The weights ensure continuous transitions between camera views and epipole consistency, i.e., rendering the object from original camera viewpoints exactly reproduces the original images.

Using the global camera parameters, each surface point is then projected into its four closest alpha mattes, reflectance images, and environment mattes. We use the interpolation weights w_i to interpolate the view-dependent alpha from the alpha mattes and the color radiance values from the reconstructed reflectance images.

6.4. Environment Matte Interpolation To compute the radiance contribution from the environment mattes involves two steps: interpolating new Gaussians $\hat{G}$, and convolving them with the environment map to compute the resulting colors.

We first interpolate the parameters of $k = 4$ reprojected Gaussians $\tilde{G}_i$. Using w_i, we compute linear combinations for the amplitudes a_i and the directional vectors $\tilde{C}_i$. The angular parameters (α_i, β_i) and $\tilde{\theta}_i$ are blended using quaternion interpolation. The result is a new Gaussian $\hat{G}$ that is an interpolated version of the Gaussians $\tilde{G}_i$, morphed to the new viewpoint.

Note that this interpolation needs to be performed on matching Gaussians from the environment mattes. Figure 7 shows a simplified 1D drawing of the matching process. We

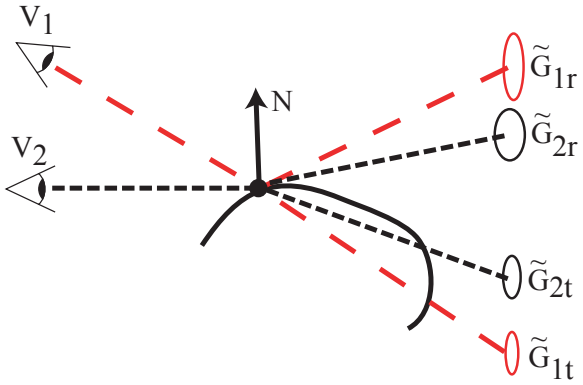

Figure 7: Matching of reflective and refractive Gaussians.

are only storing two Gaussians G_i per environment matte pixel, where each pixel corresponds to a viewpoint ray V_i in the figure. The two Gaussians per pixel are classified as *reflective* ($\tilde{G}_{ir}$) or *transmissive* ($\tilde{G}_{it}$). We compute the angle ϕ of their center vectors $\tilde{C}_{ir}$ and $\tilde{C}_{it}$ with the surface normal N. If $\phi > 90^o$, we classify the Gaussian as transmissive. If $\phi <= 90^o$, we classify it as reflective. If both Gaussians are reflective or refractive, we only store the one with the larger amplitude a. This computation has to be performed for each change of the environment map, after computing the reprojected Gaussians $\tilde{G}$.

During interpolation, we match up refractive and reflective Gaussians. In other words, new Gaussians $\hat{G}_r$ and $\hat{G}_t$ are interpolated from $\tilde{G}_{ir}$ and $\tilde{G}_{it}$, respectively. Note that this matching would be much more difficult if we had stored more than two Gaussians per environment matte pixel, as proposed by Chuang et al. [6].

To compute the color C'' for each viewing ray from the interpolated environment mattes we use:

$$C'' = a_r(\hat{G}_r \otimes \tilde{T}) + a_t(\hat{G}_t \otimes \tilde{T}), \qquad (7)$$

where $\otimes$ denotes convolution. The final pixel color C according to Equation (4) is the sum of the low-resolution reflectance field color C' (Equation 6) and the high-resolution reflectance field color C'' (Equation 7).

7. Results

Figures 8 and 9 show different models in new illumination environments. We used the light probe images available from Paul Debevec's web site as environment maps. All objects are rendered from novel viewpoints that are not part of the acquired image sequence. The rendering time per frame is about 5 minutes on a 2 GHz Pentium 4 PC with 1 GB of RAM.

Figure 8 (see color plates) shows renderings using only the environment mattes (left), using only the reflectance images (middle), and combining the two (right). Note that the environment mattes, storing the high-resolution reflectance field, mostly capture refractions, while the reflectance images, storing the low-resolution reflectance field, mostly capture reflections.

Figure 9 (see color plates) shows a few frames from an animation with a rotating viewpoint. Note how the specular highlights and refractive effects are accurately preserved by our interpolation procedure. The actual quality of the models and of our interpolation method can be observed in the companion animations.

The number of turntable positions for our scanner is user specified. It depends on the object shape, surface appearance, and the desired viewpoints for rendering. The visual hull does not capture large concavities, and we need to make up for this deficiency by capturing more images. We also need to capture many images to densely sample the surface reflectance and refraction fields for transparent objects and objects with high specularity. For the models shown in this paper we use six cameras, roughly spaced 10° apart in elevation, and 5° angular turntable increments. For a full 360° view of the object this corresponds to $6 \times 72 = 432$ viewing positions. To speed up acquisition and processing, we have the flexibility of only capturing views that are needed for the final animation.

For a full 360° acquisition of a model, we capture the following images:

Alpha Mattes: For each viewpoint we need $2 \times 3 = 6$ images for a total of $432 \times 6 = 2,592$ images. The acquisition time is about 1 hour.

Environment Mattes: We acquire 300 images for environment matting for each viewpoint. Because the aspect ratio of our monitor is 16:9, we use a different number of backdrop images for each direction: 125 in diagonal, 100 in horizontal, and 75 in vertical direction. As mentioned in Section 5.3, we capture environment mattes using only three cameras. All environment matte and reflectance images are captured using the high dynamic range (HDR) technique of Debevec et al. [9]. For each viewpoint, we take four pictures with different exposure times [25]. The total number of pictures is $3 \times 72 \times 300 \times 4 = 259,200$. The acquisition time is about 18 hours, or 5 minutes per viewpoint.

Reflectance Images: For each viewpoint, we acquire HDR reflectance images for 4×11 light directions, for a total of $432 \times 44 \times 4 = 76,032$ images. Uncompressed this would correspond to 371 GB of data. As shown in Table 1, the PCA compression achieves a reduction of at least a factor of 10. The acquisition time per turntable position with 6 cameras is about 24 minutes. The total acquisition time is 24×72 minutes or 28.8 hours.

The complete number of images is 337,824, and the entire digitizing process takes about 47.8 hours using a 2 GHz Pentium 4 processor with 1 GB of RAM. The whole process is fully automated without any user intervention. All of our models are created from a single scan.

The processing time to segment the images, compute the opacity hull, and build the point-based data structure is less than 30 minutes. The PCA analysis of the reflectance images takes about 20 to 40 hours, depending on the size of the object. The processing of the images for the environment mattes takes about 10 minutes per viewpoint, or 72 hours for the model. These times are total CPU hours. To speed up the computation, we are using multiple PCs to process different image sets in parallel.

Table 1 shows the size of some acquired models that we used for the images in this paper. *Views* refers to rotation angle range used during acquisition and rendering. *PCA* is the size of the PCA compressed reflectance images. *Mattes* is the size of all alpha and environment mattes. And *Hull* is the size of the opacity hull model. The total sizes are large, but manageable by today's standards.

Model	Views	PCA	Mattes	Hull	Total
Mug	180°	2 GB	1.4 GB	50 MB	3.5 GB
Horse	90°	0.7 GB	0.4 GB	20 MB	1.1 GB

Table 1: Size of some acquired models.

8. Future Work

We have only applied minimal lossy compression to the reflectance field data and no compression to the environment and alpha mattes. It seems feasible to apply traditional image compression tools, such as JPEG 2000 and MPEG-4. The temporal coherence of the acquired images should help in achieving high compression ratios.

One of the limitations of our approach is the assumption that reflectance at each point can be modeled by single refracted and reflected rays. This is clearly not completely general. Exploring the implications this has on factoring and interpolation is the subject of future work.

Our system only captures the effects of an infinitely distant environment illumination. We do not model the effect of

the object on near elements, such as shadows, reflections of the object onto other surfaces, caustics, etc. One way to approach this problem is to generate the image of the incoming radiance at every point in the surrounding scene (e.g., on a neighboring plane). Given the reflectance properties of these points (e.g., as a parametric BRDF model), we can then compute the outgoing radiance in any direction.

Other avenues for further work include real-time rendering of our point models, graphics hardware acceleration, and improvements to the scanner hardware. Since the low-resolution sampling of the reflectance field has limitations, we would like to replace it with a high-resolution illumination field, possibly by using multiple monitors around the object. Animation of the models, in particular with soft body dynamics, is a very challenging problem. It is not clear how to compute new surface reflectance field data for changing geometry.

9. Conclusions

In this paper we have presented a method for acquisition and rendering of transparent and refractive objects. Using an image-based 3D scanning system with color monitor backdrops, we are able to scan transparent objects that would be extremely difficult or impossible to scan with traditional 3D scanners. Our system is fully automated and very robust. We have shown that a parameterization of surface reflectance fields into high- and low-resolution areas offers a practical method to acquire high quality models. We believe our approach is the first to replace parametric appearance models for subsurface scattering and refraction with measured surface reflectance field data. Challenges for the future include efficient compression, real-time rendering, and animation.

Acknowledgments

Many thanks to Fredo Durand for his valuable feedback on parametric appearance models and global illumination methods. The light probe environment maps are courtesy of Paul Debevec.

References

1. *Active Optical Range Imaging Sensors*. Advances in Machine Vision. Springer Verlag, 1989.
2. J. Blinn and M. Newell. Texture and Reflection in Computer Generated Images. *Communications of the ACM (SIGGRAPH '76 Proceedings)*, 19(10):542–547, October 1976.
3. C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured Lumigraph Rendering. In *Computer Graphics*, SIGGRAPH 2001 Proceedings, pages 425–432. Los Angeles, CA, 2001.
4. B. Cabral, M. Olano, and P. Nemec. Reflection Space Image Based Rendering. In *Computer Graphics*, SIGGRAPH '99 Proceedings, pages 165–170. Los Angeles, CA, August 1999.
5. W.-C. Chen, R. Grzeszczuk, and J.-Y. Bouguet. Light Field Mapping: Efficient Representation and Hardware Rendering of Surface Light Fields. In *Computer Graphics*, To appear in the SIGGRAPH 2002 Proceedings. San Antonio, TX, July 2002.
6. Y.-Y. Chuang, D. Zongker, J. Hindorff, B. Curless, D. Salesin, and R. Szeliski. Environment Matting Extensions: Towards Higher Accuracy and Real-Time Capture. In *Computer Graphics*, SIGGRAPH 2000 Proceedings, pages 121–130. 2000.
7. K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and Texture of Real World Surfaces. *ACM Transactions on Graphics*, 1(18):1–34, 1999.
8. P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the Reflectance Field of a Human Face. In *Computer Graphics*, SIGGRAPH 2000 Proceedings, pages 145–156. July 2000.
9. P. Debevec and J. Malik. Recovering High Dynamic Range Radiance Maps From Photographs. In *Computer Graphics*, SIGGRAPH 97 Proceedings, pages 369–378. Los Angeles, CA, 1997.
10. P. Debevec, Y. Yu, and G. Borshukov. Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping. In *Proceedings of the 9th Eurographics Workshop on Rendering*, pages 105–116. Vienna, Austria, June 1998.
11. J. Dorsey, A. Edelman, J. Legakis, H. W. Jensen, and H. K. Pedersen. Modeling and Rendering of Weathered Stone. In *Computer Graphics*, SIGGRAPH '99 Proceedings, pages 225–234. Los Angeles, CA, August 1999.
12. S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The Lumigraph. In *Computer Graphics*, SIGGRAPH 96 Proceedings, pages 43–54. New Orleans, LS, August 1996.
13. Z. Hakura and J. Snyder. Realistic Reflections and Refractions on Graphics Hardware with Hybrid Rendering and Layered Environment Maps. In *Proceedings of the 12th Eurographics Workshop on Rendering*. London, UK, June 2001.
14. Z. Hakura, J. Snyder, and J. Lengyel. Parameterized Environment Maps. In *Symposium on Interactive 3D Graphics*, pages 203–208. March 2001.
15. T. Hawkins, J. Cohen, and P. Debevec. A Photometric Approach to Digitizing Cultural Artifacts. In *2nd International Symposium on Virtual Reality, Archaeology, and Cultural Heritage*. Glyfada, Greece, November 2001.
16. W. Heidrich, H. Lensch, M. Cohen, and H. Seidel. Light Field Techniques for Reflections and Refractions. In *Rendering Techniques '99*, pages 187–196. Springer, Wien, Granada, Spain, June 1999.
17. W. Heidrich and H.-P. Seidel. Realistic, Hardware-Accelerated Shading and Lighting. In *Computer Graphics*, SIGGRAPH '99 Proceedings, pages 171–178. Los Angeles, CA, August 1999.
18. H. W. Jensen. *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001. ISBN: 1568811470.
19. H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan.

A Practical Model for Subsurface Light Transport. In *Computer Graphics*, SIGGRAPH 2001 Proceedings, pages 511–518. Los Angeles, CA, August 2001.

20. A. Laurentini. The Visual hull Concept for Silhouette-Based Image Understanding. *PAMI*, 16(2):150–162, February 1994.

21. H. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel. Image-Based Reconstruction of Spatially Varying Materials. In *Proceedings of the 12th Eurographics Workshop on Rendering*. June 2001.

22. M. Levoy and P. Hanrahan. Light Field Rendering. In *Computer Graphics*, SIGGRAPH 96 Proceedings, pages 31–42. New Orleans, LS, August 1996.

23. S. Marschner, S. Westin, E. Lafortune, K. Torrance, and D. Greenberg. Image-based BRDF Measurement Including Human Skin. In *Proceedings of the 10th Eurographics Workshop on Rendering*, pages 139–152. Granada, Spain, June 1999.

24. W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-Based Visual Hulls. In *Computer Graphics*, SIGGRAPH 2000 Proceedings, pages 369–374. Los Angeles, CA, July 2000.

25. W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan. Image-Based 3D Photography using Opacity Hulls. In *Computer Graphics*, To appear in the SIGGRAPH 2002 Proceedings. San Antonio, TX, July 2002. Available as MERL TR after April 15.

26. F. Nicodemus, J. Richmond, J. Hsia, I. Ginsberg, and T. Limperis. *Geometric Considerations and Nomenclature for Reflectance*. National Bureau of Standards (US) Monograph 161, 1977.

27. R. Grzeszczuk (organizer). Acquisition and Visualization of Surface Light Fields. In *Course Notes*, SIGGRAPH 2001. 2001.

28. E. Paquette, P. Poulin, and G. Drettakis. Surface Aging by Impacts. In *Graphics Interface 2001*, pages 175–182. June 2001.

29. H. Pfister, M. Zwicker, J. van Baar, and M Gross. Surfels: Surface Elements as Rendering Primitives. In *Computer Graphics*, SIGGRAPH 2000 Proceedings, pages 335–342. Los Angeles, CA, July 2000.

30. M. Pharr and P. Hanrahan. Monte Carlo Evaluation of Non-Linear Scattering Equations for Subsurface Reflection. In *Computer Graphics*, SIGGRAPH 2000 Proceedings, pages 75–84. New Orleans, LS, July 2000.

31. K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, and W. Stuetzle. View-Based Rendering: Visualizing Real Objects from Scanned Range and Color Data. In *Eurographics Rendering Workshop 1997*, pages 23–34. June 1997.

32. H. Rushmeier. 3D Capture for Computer Graphics. In *Third International Conference on 3D Digital Imaging and Modeling*. Quebec City, Canada, May 28 – June 1 2001.

33. H. E. Rushmeier. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. Ph.d. thesis, Program of Computer Graphics, Cornell University, Ithaca, NY, 1988.

34. Y. Sato, M. D. Wheeler, and K. Ikeuchi. Object Shape and Reflectance Modeling from Observation. In *Computer Graphics*, SIGGRAPH 97 Proceedings, pages 379–387. 1997.

35. P. Shirley, H. Hu, B. Smits, and E. Lafortune. A Practitioners' Assessment of Light Reflection Models. In *Proceedings of Pacific Graphics*, pages 40–49. October 1997.

36. A. R. Smith and J. F. Blinn. Blue Screen Matting. In *Computer Graphics*, volume 30 of *SIGGRAPH 96 Proceedings*, pages 259–268. 1996.

37. J. Stam. Multiple Scattering as a Diffusion Process. In *Proceedings of the 6th Eurographics Workshop on Rendering*, pages 51–58. Dublin, Ireland, June 1995.

38. T. Whitted. An Improved Illumination Model for Shaded Display. *Communications of the ACM*, 6(23):343–349, June 1980.

39. D. Wood, D. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle. Surface Light Fields for 3D Photography. In *Computer Graphics*, SIGGRAPH 2000 Proceedings, pages 287–296. Los Angeles, CA, July 2000.

40. Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse Global Illumination: Recovering Reflectance Models of Real Scenes from Photographs. In *Computer Graphics*, SIGGRAPH 99 Proceedings, pages 215–224. Los Angeles, CA, August 1999.

41. D. Zongker, D. Werner, B. Curless, and D. Salesin. Environment Matting and Compositing. In *Computer Graphics*, SIGGRAPH 99 Proceedings, pages 205–214. August 1999.

42. M. Zwicker, H. Pfister., J. Van Baar, and M. Gross. Surface Splatting. In *Computer Graphics*, SIGGRAPH 2001 Proceedings, pages 371–378. Los Angeles, CA, July 2001.

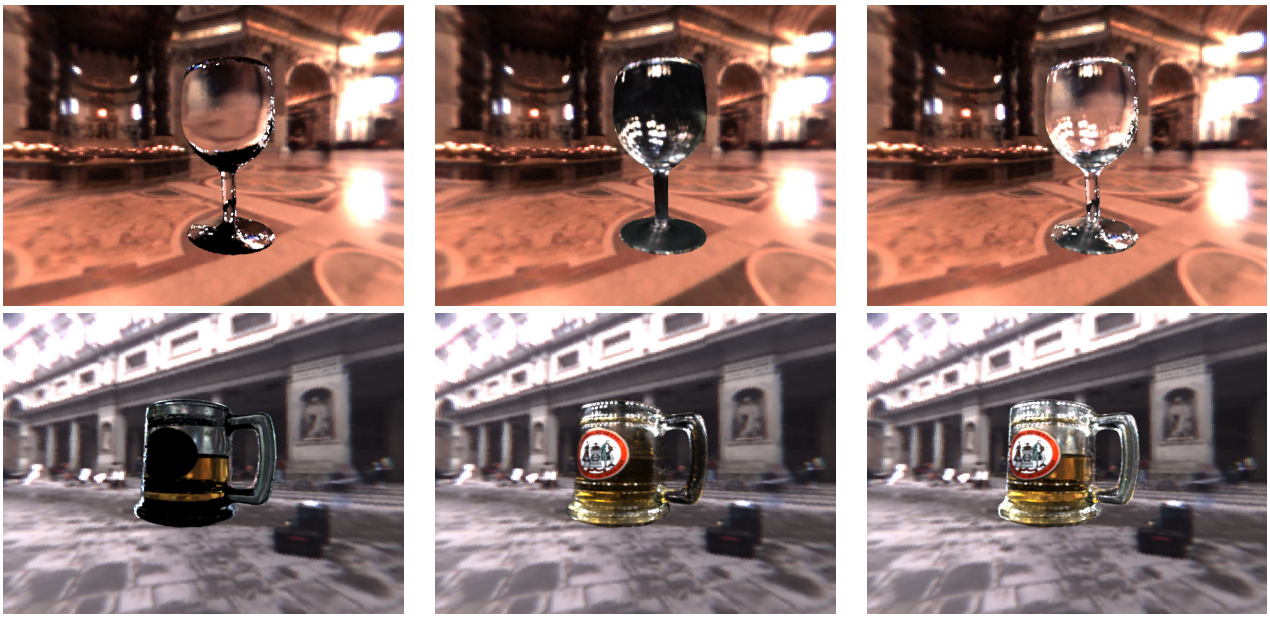

Figure 8: Left: High-resolution reflectance field from the environment mattes. Middle: Low-resolution reflectance field from the reflectance images. Right: Combined.

Figure 9: Frames from an animation with rotating viewpoint.