Environment Matting Extensions: Towards Higher Accuracy and Real-Time Capture

Yung-Yu Chuang[1] *Douglas E. Zongker*[1] *Joel Hindorff*[1] *Brian Curless*[1] *David H. Salesin*[1,2] *Richard Szeliski*[2]

[1]University of Washington [2]Microsoft Research

Abstract

Environment matting is a generalization of traditional bluescreen matting. By photographing an object in front of a sequence of structured light backdrops, a set of approximate light-transport paths through the object can be computed. The original environment matting research chose a middle ground—using a moderate number of photographs to produce results that were reasonably accurate for many objects. In this work, we extend the technique in two opposite directions: recovering a more accurate model at the expense of using additional structured light backdrops, and obtaining a simplified matte using just a single backdrop. The first extension allows for the capture of complex and subtle interactions of light with objects, while the second allows for video capture of colorless objects in motion.

CR Categories: I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding — modeling and recovery of physical attributes; I.3.3 [**Computer Graphics**]: Picture/Image Generation — display algorithms; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism — color, shading, shadowing, and texture

Keywords: Alpha channel, augmented reality, blue-screen matting, blue spill, clip art, colored transparency, environment map, environment matte, image-based rendering, real-time capture, reflection, refraction.

1 Introduction

Conventional *matting* consists of filming a foreground object against a known background and determining the foreground color and opacity at each image pixel. Conventional *image compositing* simply layers the foreground over an arbitrary background using the opacity to control the relative contributions at each pixel. *Environment matting* and *compositing* generalize the conventional methods by modeling arbitrary transport paths from the background through the foreground object to the camera. After making a set of approximations, Zongker *et al.* [20] demonstrate the ability to capture and render the effects of reflection, refraction, scatter, and colored filtering of light from a background. These effects, none of which are modeled with conventional matting and compositing, make a dramatic contribution to the visual realism of the final image.

The original environment matting method employs a sequence of structured backdrops to estimate mappings from the background through the foreground object. These backdrops consist of a hierarchy of finer and finer horizontal and vertical square-wave stripes from which the matte can be extracted with $O(\log k)$ images for an $k \times k$ pixel grid. This choice of backdrops is inspired by a related technique developed for 3D range scanning [1]. In practice, however, this approach has a number of shortcomings.

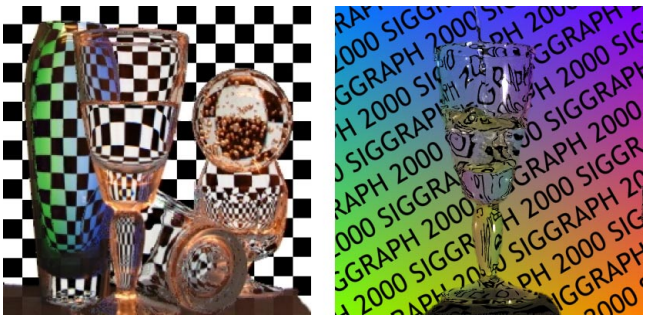


Figure 1 Sample composite images constructed with the techniques of this paper: slow but accurate on the left, and a more restricted example acquired at video rates on the right.

The previous approach of Zongker *et al.* assumes, first of all, that each image pixel collects light from a single region of the background, augmented with an alpha component for straight-through partial coverage. This assumption fails when we consider the effects of simultaneous reflection and refraction at a dielectric. Second, the method is tuned to capturing highly specular interactions, but breaks down in the presence of surfaces that are even moderately rough. Third, the mappings that are captured assume axis-aligned filtering of background pixels. At pixels where this assumption does not hold, this axis alignment results in excessive blurring and degrades the quality of the final composite. Fourth, the original method accounts for colored filtering of light, but does not account for the effects of dispersion, which necessitate different mappings per color channel and give rise to prismatic rainbowing effects. Finally, the number of images required is typically about 20 per matte, and thus the technique does not lend itself to real-time acquisition.

In this paper, we address each of these limitations. Again taking inspiration from the 3D range-scanning literature, we follow two distinct paths. In the first approach, the background consists of a single stripe swept over time in the vertical, horizontal, and two diagonal directions. In each case, the stripe orientation is perpendicular to the sweeping direction. By combining sweeps with stripes of varying widths and intensity profiles, we demonstrate an efficient method for extracting the highest quality environment mattes to date. This method, however, requires $O(k)$ images and is even less suitable for real-time acquisition than the previous method.

Our second approach utilizes a *single* background image consisting of a color ramp. Through careful simplification of the governing equations, we demonstrate a method capable of extracting pure specular refractive and reflective mappings through a moving, deforming colorless object, such as water pouring into a glass. This technique is real-time in the sense that the data it requires can be collected in real-time, though the analysis and extraction of the matte must still be performed offline.

1.1 Related work

Environment matting [20] builds upon and substantially extends research and practice in blue screen matting. Pioneered by Vlahos [16], conventional (and commercial) blue screen matting relies on a single-color background sufficiently different from the foreground objects to extract an alpha and foreground color at each pixel. Smith and Blinn [16] use two backdrops to lift restrictions on the color of foreground objects. Neither of these techniques models transport paths beyond scalar (non-color) attenuation and partial pixel coverage. In addition, the Smith and Blinn technique does not easily extend to real-time capture, given the two-frame requirement. Our real-time method for capturing environment mattes is actually similar in spirit to the Vlahos work in that we must place constraints on the object and lighting in order to achieve our results.

The structured-light range-scanning literature suggests many possible ways to capture spatially varying properties of an object [1]. We should note that the end goals are substantially simpler in the range-scanning case. Range scanners attempt to recover just a handful of parameters per pixel: primarily, depth and reflectance. Environment-matting procedures, on the other hand, generally need to recover a continuous, wavelength-dependent mapping from a background to the image plane. Even with the approximations described in Section 3, we must estimate at least 21 parameters per pixel, and sometimes many more.

Despite the dissimilarities, the range-scanning *illumination patterns* do inspire a number of techniques for environment matting.

The most "brute force" range-scanning method is to sweep a beam of light over an object in a raster pattern. Such an approach, while $O(k^2)$ in time since each range-image pixel is acquired sequentially, is actually practical for triangulation and imaging radar systems [1], since the reflected light seen by the sensor is known to have followed a straight line from the object. As a result, objects can be imaged with fast 1D (triangulation) or 0D (imaging radar) sensors. These faster sensors make the acquisition speeds comparable to, and in some cases better than, the $O(k)$ swept-stripe techniques described below. By contrast, typical objects used in environment matting will cause light from the background to bend through or reflect off of the object in unpredictable ways, thus requiring a full 2D sensor array to capture the light. In this case, the $O(k^2)$ penalty is prohibitive.

Using a swept plane of light, $O(k)$ images can provide shape information through optical triangulation [1]. The first environment matting technique described in Section 3 uses such a pattern, though multiple oriented sweeps are required to capture all the parameters. Note that this particular environment matting technique bears some resemblance to the space-time analysis described by Kanade *et al.* [9] and Curless and Levoy [4], in which the authors study the time evolution of reflected light and triangulate over space and time.

By projecting a hierarchy of progressively finer stripe patterns, the required number of images for optical triangulation can be reduced to $O(\log k)$ [14]. Zongker *et al.* [20] use such a stripe hierarchy with some accompanying compromises over the swept-stripe technique.

Finally, using a color ramp, researchers have demonstrated single-frame triangulation, that is, an acquisition with time complexity $O(1)$, albeit with increased susceptibility to noise [6, 15]. In Section 4, we apply a similar pattern to extract environment mattes in real time. To combat the effects of noise, we apply a non-linear, discontinuity-preserving filter [11] to the resulting matte sequence.

Hybrid stripe-ramp methods have been proposed to manage the trade-off between number of images and susceptibility to noise [3, 8]. We have not explored such methods for environment matting, as they will likely yield results of lower quality than the swept-stripe method we describe here, and will still require multiple frames, making them unsuitable for real-time capture.

Our work also has some connection to BRDF acquisition. Though we do not explicitly solve for the BRDF, one could certainly imagine using environment matting to capture reflection functions over a uniformly coated surface of known geometry, such as a sphere. A more direct connection lies in the BRDF fitting work of Ward [18]. Using an elliptical Gaussian model for rough specular reflection, he achieves excellent matches to goniometric samples. This model is the motivation for our choice of oriented, elliptical, Gaussian weighting functions described in Section 3.

1.2 Overview

In Section 2 we describe the general environment matting model. The two following sections describe the two extensions we have developed—higher accuracy mattes in Section 3 and real-time-capture mattes in Section 4. Each of Sections 3 and 4 describes the assumptions made to reduce the general matting equation to something that can be captured, then describes the experimental procedure used and shows results. We conclude in Section 5 with a summary and ideas for future work.

2 The environment matting equation

We begin by developing a general expression for the environment matting equation and showing how it reduces to the traditional compositing equation, as well as the equation developed by Zongker *et al.* [20].

An imaging system, such as a CCD camera, records a discrete set of samples over an image plane. Let's assume for the moment that we have a camera that measures the irradiance at each wavelength separately. Then, for a given pixel, the camera records a value C for each wavelength.[1] Following the environment mapping work of Blinn and Newell [2], we can express this color in terms of an infinitely distant environment illumination $\boldsymbol{E}(\boldsymbol{\omega})$:

$$C \;=\; \int \boldsymbol{W}(\boldsymbol{\omega})\, \boldsymbol{E}(\boldsymbol{\omega})\, d\boldsymbol{\omega}\,. \tag{1}$$

The weighting function $\boldsymbol{W}$ comprises all means of transport of environment lighting from all directions $\boldsymbol{\omega}$ through a foreground object to the camera, including any blurring due to the camera optics and area integration at a sensor cell. This equation holds under the assumption that none of the materials that are scattering light from the environment exhibit any wavelength coupling (e.g., fluorescence).

Next, we rewrite this equation as a spatial integral over a bounding surface (e.g., an environment map). Further, we augment the equation to include an additive foreground color F. This foreground color is typically due to some additional lighting that is separate from the environment map, though it could encompass object emissivity as well. Under these assumptions, our equation becomes

$$C \;=\; F \;+\; \int \boldsymbol{W}(\boldsymbol{x})\, \boldsymbol{T}(\boldsymbol{x})\, d\boldsymbol{x}\,. \tag{2}$$

From this equation, we can develop a series of approximations that allow us to embed a foreground object in a new environment with varying degrees of quality.

[1] Throughout this paper, we use ordinary italics for scalar quantities (e.g., a position x); bold-italics for functions of more than one spatial parameter (e.g., an area $\boldsymbol{A}$); colored italics for functions of wavelength (e.g., a color C); and colored bold-italics for functions of both wavelength and more than one spatial parameter (e.g., a texture map $\boldsymbol{T}$).

To arrive at the traditional image compositing equation [12], we assume that the straight-through background pixel is the only environment sample that affects the camera pixel. Let $\boldsymbol{P}$ be the rectangular-area support of the pixel p on the background. Then we describe the (in this case, monochromatic) weighting function as

$$\boldsymbol{W}(\boldsymbol{x}) \;=\; (1-\alpha)\,\boldsymbol{\Pi}(\boldsymbol{x};\boldsymbol{P})\,, \tag{3}$$

where α represents the foreground's transparency or partial pixel coverage, and $\boldsymbol{\Pi}(\boldsymbol{x};\boldsymbol{A})$ is the box function of unit volume supported over an arbitrary axis-aligned area $\boldsymbol{A}$. Next we define $\mathcal{M}(\boldsymbol{T},\,\boldsymbol{A})$ as the "texture-mapping operator" that performs the area integral and returns the average value of the texture $\boldsymbol{T}$ over region $\boldsymbol{A}$:

$$\mathcal{M}(\boldsymbol{T},\,\boldsymbol{A}) \;\equiv\; \int \boldsymbol{\Pi}(\boldsymbol{x};\boldsymbol{A})\,\boldsymbol{T}(\boldsymbol{x})\,d\boldsymbol{x}\,. \tag{4}$$

Finally, defining the filtered background B to be the integral over the pixel's support

$$B \;\equiv\; \mathcal{M}(\boldsymbol{T},\,\boldsymbol{P})\,, \tag{5}$$

and substituting the previous three equations into Equation 2, gives the traditional compositing equation:

$$C \;=\; F \;+\; (1-\alpha)\,B\,. \tag{6}$$

Note that α does not have any wavelength dependence and thus cannot model color-filtered transparency. In addition, F is a measured quantity that is added directly to the attenuated background—in effect, it is pre-multiplied by α.

Zongker *et al.* model more complex lighting effects by approximating the environment as a set of m texture maps $\boldsymbol{T}_i(\boldsymbol{x})$ (the six sides of a bounding cube for instance), and by using more general light transport paths. Their weighting function is

$$\boldsymbol{W}(\boldsymbol{x}) \;=\; (1-\alpha)\,\boldsymbol{\Pi}(\boldsymbol{x};\boldsymbol{P}) \;+\; \sum_{i=1}^{m} R_i\,\boldsymbol{\Pi}(\boldsymbol{x};\boldsymbol{A}_i)\,. \tag{7}$$

In their formulation, the $\boldsymbol{A}_i$ represent various axis-aligned regions, each lying on a different texture map (corresponding, typically, to a different face of the environment cube). The R_i are reflectance coefficients describing the amount of light from the designated area of texture map i that is reflected or transmitted by the object at a given wavelength. In this formulation, R_i captures color-filtered transparency, and α represents only partial pixel coverage of the object. Substituting this weighting function into Equation 2 gives the environment matting equation used by Zongker *et al.*:

$$C \;=\; F \;+\; (1-\alpha)\,B \;+\; \sum_{i=1}^{m} R_i\,\mathcal{M}(\boldsymbol{T}_i,\,\boldsymbol{A}_i)\,. \tag{8}$$

Note that this approach not only permits colored filtering of light, but also enables effects such as reflection and refraction since the light contributing to a pixel can be scattered from parts of the environment other than just the pixel directly behind the object. This approach, however, does have several distinct limitations. First, the components of the weighting function are assumed to be separable products of wavelength functions R_i and spatial functions $\boldsymbol{\Pi}(\boldsymbol{x};\boldsymbol{A}_i)$. Thus, phenomena such as dispersion are not handled, since these require the weighting functions to shift spatially with wavelength. Second, the axis-aligned rectangle weighting functions do not simulate the effects of, for example, smooth BRDF's, which when mapped onto a background have a smooth, oriented footprint. Finally, other than the straight-through α-component, the approach models only a single mapping from a texture face to the camera. In reality, multiple mappings to the same face can and do happen and must be modeled, for example, when reflection and refraction at an interface cause view rays to split into distinct groups that strike the same backdrop.

Our first objective, then, is to choose a different model for the weighting function that is more physically motivated and whose parameters are still easy to acquire using a simple apparatus.

3 Towards higher accuracy

To address the limitations of the weighting function described in Zongker *et al.*, we generalize it to a sum of Gaussians:

$$\boldsymbol{W}(\boldsymbol{x}) \;=\; \sum_{i=1}^{n} R_i\,\boldsymbol{G}_i(\boldsymbol{x})\,. \tag{9}$$

In our formulation, we allow any number of contributions from a single texture map. Here, R_i is an attenuation factor, and each $\boldsymbol{G}_i$ is the unit-area, elliptical, oriented $2D$ Gaussian:

$$\boldsymbol{G}_i(\boldsymbol{x}) \;\equiv\; \boldsymbol{G}_{\text{2D}}(\boldsymbol{x};\;\boldsymbol{c}_i,\,\boldsymbol{\sigma}_i,\,\theta_i)\,, \tag{10}$$

where $\boldsymbol{G}_{\text{2D}}$ is defined as

$$\boldsymbol{G}_{\text{2D}}(\boldsymbol{x};\boldsymbol{c},\boldsymbol{\sigma},\theta) \;\equiv\; \frac{1}{2\pi\sigma_u\sigma_v}\exp\left[-\frac{u^2}{2{\sigma_u}^2}-\frac{v^2}{2{\sigma_v}^2}\right] \tag{11}$$

with

$$\begin{aligned} u \;&=\; (x-c_x)\cos\theta-(y-c_y)\sin\theta \\ v \;&=\; (x-c_x)\sin\theta+(y-c_y)\cos\theta. \end{aligned}$$

Here, $\boldsymbol{x}=(x,y)$ are the pixel coordinates, $\boldsymbol{c}=(c_x,c_y)$ is the center of each Gaussian, $\boldsymbol{\sigma}=(\sigma_u,\sigma_v)$ are the "unrotated" widths (a.k.a. standard deviations) in a local uv-coordinate system, and θ is the orientation. Figure 2 illustrates these parameters. Thus, our weighting function is some n-modal Gaussian with each term contributing a reflective or refractive effect from the object. Substituting into Equation 2, we arrive at a new form of the matting equation:

$$C \;=\; F \;+\; \sum_{i=1}^{n} R_i \int \boldsymbol{G}_{\text{2D}}(\boldsymbol{x};\;\boldsymbol{c}_i,\,\boldsymbol{\sigma}_i,\,\theta_i)\,\boldsymbol{T}(\boldsymbol{x})\,d\boldsymbol{x}\,. \tag{12}$$

(In this equation, we use $\boldsymbol{T}(\boldsymbol{x})$ to represent the set of all texture maps. The n modes of the weighting function are distributed over m textures, where n may be larger than m in general. The choice of the particular texture map used in computing a given Gaussian contribution i should be assumed to be implicitly controlled by the position $\boldsymbol{c}_i$ of the Gaussian weighting function.)

The key advantages of this weighting function over the one used by Zongker *et al.* are that: (1) the spatial variation can be coupled with wavelength to permit modeling of dispersion; (2) it supports multiple mappings to a single texture; and (3) it approximates the behavior of BRDF's more closely (by using oriented Gaussian weighting functions rather than box functions).

In practice, each of the "colored" values $C, F, R_i, \boldsymbol{c}_i, \boldsymbol{\sigma}_i, \theta_i$ and $\boldsymbol{T}$ in Equation 12 is implemented as an *rgb* vector. So, in practice, this equation actually represents three independent equations, one for each of the color components. Our unknowns are $F, R_i, \boldsymbol{c}_i, \boldsymbol{\sigma}_i, \theta_i$, which means that each pixel encodes $3+18n$ parameters.

3.1 Swept Gaussians for environment matting

Recovering the environment matte requires taking a set of images of an object in front of a sequence of backdrops. Our method consists of three steps: (1) identifying pixels outside the object silhouette, (2) recovering the foreground color, and (3) applying a set of novel

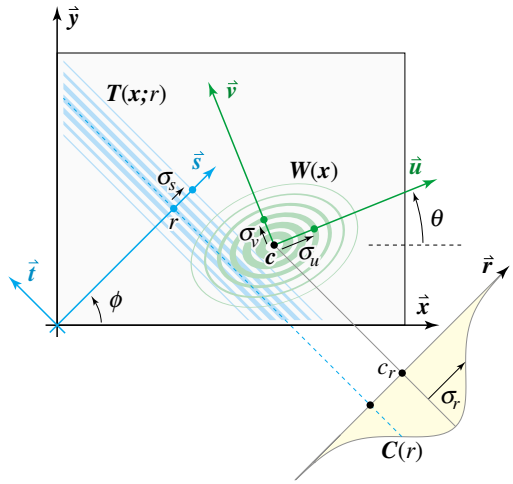


Figure 2 Illustration of the variables used in recovering an unknown elliptical, oriented Gaussian by sweeping out convolutions with known Gaussian stripes. As a tilted stripe $\boldsymbol{T}(\boldsymbol{x};r)$ of width σ_s and position r sweeps across the background in direction s, it passes "under" the elliptical Gaussian weighting function $\boldsymbol{W}(\boldsymbol{x})$ associated with a single camera pixel. The camera records the integral of the product of the stripe and the weighting function, which describes a new, observed function $C(r)$ as the stripe sweeps. The center c_r and width σ_r of this observed function are related to the center $\boldsymbol{c}$ and width $\boldsymbol{\sigma} = (\sigma_u, \sigma_v)$ of the weighting function and the width of the stripe through Equations 20 and 21.

background *stimulus functions* to estimate the remaining parameters in the matte.

In the first step of our high-accuracy matting method, we identify pixels that are outside the silhouette of the object. This step is desirable for two reasons: it saves us the computational effort of estimating the matte parameters at these pixels, and it prevents us from making potentially noisy estimates of how straight-through background pixels map to the image, which would result in shimmering artifacts when rendering. To identify these pixels, we use the method of Zongker *et al.* In particular, we display a coarse-to-fine sequence of horizontal and vertical square-wave background patterns with and without the object. If we measure the same color (within a user-specified tolerance) at a pixel both with and without the object for each background, then we consider the pixel to map straight through. The overhead of taking these additional images is small compared to the total acquisition time.

To recover the foreground color, we photograph the object against two solid backgrounds. Replacing $\boldsymbol{T}(\boldsymbol{x})$ in Equation 12 with a single backdrop of constant color T and integrating, we get

$$C \;=\; F \;+\; RT \tag{13}$$

where $R \equiv \sum_{i=1}^{n} R_i$. Given the two images, we have two equations in two unknowns for each color channel, i.e., the foreground color F and the aggregate attenuation factor R. Solving the system of equations yields the foreground color.

Once we have the silhouette mask and the foreground color, we can solve for the remaining parameters of Equation 12 using a large set of controlled backdrops (i.e., stimulus functions). Zongker *et al.* use a hierarchical set of square-wave stripe patterns in both the vertical and horizontal directions. They encounter difficulties with this method for two reasons: (1) the square waves are not good stimuli for recovering smooth functions, and (2) there is no obvious way to recover multiple mappings to the backdrop using these stimuli. To combat the first problem, we choose a smooth set of stimulus functions. To address the second, we constrain the stimuli to be narrow in one dimension, sweeping over time to reveal multiple mappings to the same background. Our choice of stimulus function, then, is a set of swept Gaussian stripes.

Let's see how we can use sweeping stripes to recover some of the parameters of our weighting functions. To begin, let us assume that the weighting function is unimodal and axis-aligned ($n = 1$ and $\theta = 0$). Under these assumptions, we can omit the summation and the subscript i in Equations 9 and 10 and then decompose the 2D Gaussian weighting function into two 1D components:

$$\boldsymbol{W}(\boldsymbol{x}) \;=\; R\,\boldsymbol{G}_{1\mathrm{D}}(x;c_x,\sigma_u)\,\boldsymbol{G}_{1\mathrm{D}}(y;c_y,\sigma_v)\,, \tag{14}$$

where

$$\boldsymbol{G}_{1\mathrm{D}}(x;c,\sigma) \;=\; \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x-c)^2}{2\sigma^2}\right]. \tag{15}$$

Our first stimulus function will be a vertical stripe that is constant in y and has a 1D Gaussian intensity profile in x with width σ_s:

$$\boldsymbol{T}(\boldsymbol{x}) \;=\; \boldsymbol{G}_{1\mathrm{D}}(x;0,\sigma_s)\,. \tag{16}$$

Now consider sweeping the stripe horizontally, displacing it at each step by some offset r:

$$\boldsymbol{T}(\boldsymbol{x};r) \;=\; \boldsymbol{G}_{1\mathrm{D}}(x-r;0,\sigma_s) \;=\; \boldsymbol{G}_{1\mathrm{D}}(r-x;0,\sigma_s)\,. \tag{17}$$

The camera observation at a pixel is then given by:

$$\begin{aligned}
C(r) &= \int \boldsymbol{W}(\boldsymbol{x})\,\boldsymbol{T}(\boldsymbol{x};r)\,d\boldsymbol{x} \\
&= \iint R\,\boldsymbol{G}_{1\mathrm{D}}(x;c_x,\sigma_u)\,\boldsymbol{G}_{1\mathrm{D}}(y;c_y,\sigma_v)\,\boldsymbol{G}_{1\mathrm{D}}(r-x,0,\sigma_s)\,dx\,dy \\
&= \int R\,\boldsymbol{G}_{1\mathrm{D}}(x;c_x,\sigma_u)\,\boldsymbol{G}_{1\mathrm{D}}(r-x;0,\sigma_s)\,dx \cdot \int \boldsymbol{G}_{1\mathrm{D}}(y;c_y,\sigma_v)\,dy \\
&= R\,\boldsymbol{G}_{1\mathrm{D}}(r;c_x,\sigma_u) * \boldsymbol{G}_{1\mathrm{D}}(r;0,\sigma_s)\cdot 1 \\
&= R\,\boldsymbol{G}_{1\mathrm{D}}(r;c_x,\sqrt{{\sigma_u}^2+\sigma_s^2}).
\end{aligned} \tag{18}$$

Thus, at each pixel, we expect to record a Gaussian evolving over time. Given an illumination stripe of known width, we can now estimate the *rgb* parameters c_x and σ_u using the procedure described below in Section 3.2. By symmetry, we can recover the vertical center coordinate and width by sweeping a horizontal Gaussian stripe in the vertical direction behind the foreground object. Thus, for the case of a single, unoriented Gaussian weighting function, a horizontal and a vertical swept Gaussian stripe are enough to estimate all the remaining parameters of the environment matte.

Figure 2 illustrates the more general case of a sweeping stripe that is constant in the t-direction and has Gaussian profile in the s-direction. This stripe is oriented at an angle ϕ with respect to the xy-coordinate system and travels in the s-direction. Under these circumstances, it is straightforward to show that the observation at a pixel will be:

$$C(r) \;=\; R\,\boldsymbol{G}_{1\mathrm{D}}(r;c_r,\sigma_r)\,, \tag{19}$$

where

$$c_r \;=\; c_x\cos\phi + c_y\sin\phi \tag{20}$$

$$\sigma_r \;=\; \sqrt{{\sigma_u}^2\cos^2(\phi-\theta) + {\sigma_v}^2\sin^2(\phi-\theta) + \sigma_s^2} \tag{21}$$

Here, c_r is the center of the weighting function projected onto the r-axis, and σ_r is the projected, convolved standard deviation of the observed Gaussian.

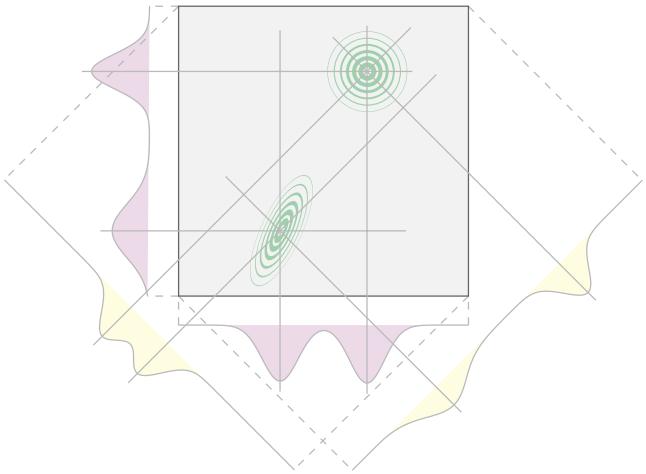


Figure 3 The green concentric rings depict a possible bimodal weighting function. The curves around the image indicate the convolved projections resulting from sweeping horizontal, vertical, and left and right diagonal Gaussian stripes across the screen independently. The horizontal and vertical sweeps alone (purple) are insufficient to determine the mode positions, but by adding additional diagonal sweeps (yellow) the correct modes can be determined.

Horizontally and vertically swept stripes alone ($\phi = 0°$ and $90°$, respectively) are not enough to determine the weighting function, so we introduce two diagonal passes at $\phi = 45°$ and $-45°$. The additional oriented stripes serve another purpose: disambiguating multiple mappings to the backdrop. As shown in Figure 3, a bimodal weighting function results in two Gaussian images over time at a pixel as the stripe sweeps across. If we use just the horizontal and vertical stripes, the two modes recorded in each sweep yield multiple indistinguishable interpretations of the bimodal weighting function. The oriented stripes can be used to disambiguate these choices, as described below.

3.2 Estimating the matte parameters

In practice, our acquisition process entails stepping each Gaussian stripe across a computer screen and recording a set of samples for each sweep and for each color channel. Given this data, we seek the best set of parameters that explain the measurements. We estimate these parameters (separately for each color channel) in four steps: (1) identifying the number of Gaussians modes in the response, (2) solving for the projected centers and widths associated with each Gaussian mode, (3) intersecting the centers to localize the Gaussian modes, and (4) computing the parameters for each Gaussian mode.

To identify the number of Gaussians for the response to a given stripe sweep, we search for a series of peaks above the noise floor of the sensor. To make this process more robust, we first filter the 1D response function, and then identify the peaks. The locations of the peaks are the starting points for the projected centers of the projected modes. If the projected modes are clearly separated, we also estimate the projected widths by examining the extent of the signal that is above the noise floor. For two overlapping modes, we compute the distances from the left mode to the left extent and the right mode to the right extent and then estimate widths accordingly. For more overlapping modes, we compute the total width and divide by the number of modes. In any case, these center and width estimates are simply starting points for a Levenburg-Marquardt optimization procedure [13] that takes the original data, the number of Gaussians, and the initial center and width estimates in order to find the best centers and widths that explain the data.

Next, we use the sets of projected centers to choose the most likely locations of the Gaussian modes $\boldsymbol{G}_i$ (from Equation 10). The centers computed in the previous step should each correspond to the center of a Gaussian mode as projected onto the axis defined by the stripe. We then construct a line passing through each projected center point running parallel to the stripe's t direction. We consider all 4-tuples of horizontal, vertical, and two diagonal lines, and hypothesize their intersections by computing the point closest to each set of four lines. We measure the distance of that point to each of the lines, and apply a user-specified tolerance to reject or accept the purported intersection.

Figure 4 A photograph of our experimental setup. In this instance an accurate environment matte is being acquired.

Finally, given the set of Gaussian modes selected by the intersection process, we determine the parameters of each Gaussian mode. For each identified mode, we have estimates of the convolved, projected parameters according to Equations 20 and 21. We compute R as the average of individually computed R's. The center $\boldsymbol{c}_i$ of $\boldsymbol{G}_i$ is simply the closest point to the lines as described in the previous step. Finally, we have four equations that relate the width and orientation of each Gaussian mode to the four measured widths. We solve this over-constrained, non-linear system of equations by finely sampling the space of possible orientations, solving for the linear-least-squares-best width parameters, and then choosing the orientation and widths that yield the lowest overall error.

The result of this sequence of steps is a reasonable estimate for the number of Gaussian modes and their parameters. As a final step, we apply a full Levenburg-Marquardt optimization to find the best $\boldsymbol{c}_i$, $\boldsymbol{\sigma}_i$, and θ_i that explain all of the measurements.

3.3 Results of accurate matting

Figure 4 shows our experimental setup. A Sony DCR-TRV900 digital video camera records images of an object as one of three monitors presents a sequence of stimulus functions. We correct for nonlinearities in the video camera using Debevec and Malik's method [5]. To calibrate each monitor's brightness settings, we display a sequence of solid gray images and record them with the radiometrically corrected camera. After averaging the gray values within each image, we have a mapping between gray values on the computer and displayed radiance. Each stripe image is adjusted so that the profile is Gaussian in radiance space.

After calibration, we begin imaging the object by extracting the foreground color and silhouette mask as described in the previous section. Next, we display the sequence of background patterns. We translate each Gaussian stripe across the screen in steps of $\sigma_s/2$ to ensure enough samples for the estimation procedure. We typically

use $\sigma_s = 2$ or 4 (measured in camera pixels) requiring about 300 or 150 stripe positions, respectively, per horizontal, vertical, or diagonal sweep. Due to the lack of synchronization between the monitor refresh and the camera, we are unable to capture at video rates; instead, a typical capture plus digital video transfer requires roughly 30 minutes. (With a synchronized system and real-time transfer to PC memory, we expect acquisition could take less than a minute.) Processing time for an environment matte is typically about 20 minutes on a 400 MHz Pentium II PC with 128 MB of RAM.

We demonstrate the accuracy of our new environment matting algorithm on three objects. For each example, we render the matte by explicitly integrating the oriented Gaussian filters over the background. The results are shown in Figures 5 and 6.

The first object is a crystal in the shape of a regularly triangulated sphere, shown in Figure 5(a). The planar facets give rise to prismatic rainbowing effects due to dispersion. This effect is captured by our new matting algorithm because we estimate a different Gaussian weighting function (with a different center) for each color channel. This effect is not modeled by the old matting algorithm, which breaks down even further due to the multiple mappings at pixels that straddle crystal facets.

The multiple mapping problem is more clearly demonstrated by our next object, a beer glass laid on its side (Figure 5(b)). Due to the grazing angle, simultaneous reflection and refraction at the top of the glass results in bimodal mappings to the background. The old method simply cannot handle this phenomenon, whereas the new method captures the effect realistically.

Finally, we captured an environment matte for a pie tin with a rough surface, oriented to cause tilted reflections from the backdrop. Figure 5(c) demonstrates the failure of the old method to capture the large, smooth weighting function indicative of surface roughness, in contrast to the new method's success.

Figure 6 demonstrates the importance of capturing the orientation of the weighting function. When we apply the new method without estimating orientation (i.e., by simply using the widths determined by the horizontal and vertical sweeps), the texture lines running at 25° off of vertical are significantly blurred. After estimating the orientation, we obtain a matte that faithfully preserves these details.

4 Towards real-time capture

In the previous section, we considered ways to increase the accuracy of composites produced with the environment matting technique, at the expense of increasing the number of input images required. Now we will attempt to go the other way—to see how much realism can be maintained when we restrict the input to just a single image of the object.

The single-image case is interesting for two reasons. First, it represents a definite, extreme end of the image-count-versus-accuracy spectrum—a sort of lower bound on the quality of the whole environment matting technique. Secondly, unlike a solution that requires even two or three images of the object, a single-image solution makes it straightforward to capture video environment mattes of *moving* objects in front of a still, structured background. The original environment matting work [20] produced video using a sort of stop-motion technique, where a rigid object was placed on a motion platform, allowing multiple photographs to be taken of each pose. This severely limited the kinds of motions that could be captured. In contrast, a method utilizing just a single background image could be used to capture breaking glass, sloshing liquids, and other kinds of non-repeatable, uninterruptible motions. Although the data capture itself is real-time, the matte extraction process is performed off-line at a slower speed. The matte extraction process analyzes the frames of a captured video and constructs a matte for each frame.

4.1 Simplifying the matting equation

Using the high-accuracy matting process, we enjoy the luxury of having many samples per pixel. In the current process, we only have three samples per pixel: red, green, and blue. Our first objective, then, is to simplify the problem just enough to be solvable—that is, so that there will be only three unknowns remaining. We choose these simplifications to maximize the visual impact of the final matte. Then, using some carefully designed heuristics, we attempt to recover more variables in order to significantly improve visual appearance.

We begin by examining the original environment matting equation [20], written here as Equation 8. Assuming a single backdrop texture, we drop the summation and the subscript i:

$$C \;=\; F \;+\; (1-\alpha)\,B \;+\; R\,\mathcal{M}(\boldsymbol{T},\,\boldsymbol{A})\,. \tag{22}$$

We can now count variables, keeping in mind that where we see a wavelength dependence, we can assume an *rgb* vector in practice. The unknowns are then F and R (*rgb* vectors), α, and $\boldsymbol{A}$. The $\boldsymbol{A}$ term can be broken into $\{\boldsymbol{c}, \boldsymbol{w}\}$, where $\boldsymbol{c} = (c_x, c_y)$ is the center of the area and $\boldsymbol{w} = (w_x, w_y)$ is the width in x and y. Thus, we have eleven variables.

The properties of the matte that we would most like to preserve are, in order: (1) the capacity to refract, reflect, and attenuate the background ($\boldsymbol{A}, R$); (2) smooth blending with the background on silhouettes (α); and (3) specular highlights due to foreground lighting (F).

Let's focus on preserving the first property, which still has seven unknowns. We can simplify this set to three unknowns under the assumption that the object is both colorless and specularly reflective and refractive (i.e., has no roughness or translucency). If the object is colorless, then R becomes a scalar ρ, with no wavelength dependence. In addition, pure specularity implies that neighboring pixels do not have overlapping support in their weighting functions. Thus, $\boldsymbol{c}$ is now an image warping function, and $\boldsymbol{w}$, derived from the warping function, indicates the size of the filter support for proper antialiasing [19]. We have found the following approximation for $\boldsymbol{w}$ to work well in practice:

$$w_x \;\equiv\; \tfrac{\partial}{\partial x} c_x \;\approx\; \tfrac{1}{2}\left[c_x(x+1,y) - c_x(x-1,y)\right] \tag{23}$$

$$w_y \;\equiv\; \tfrac{\partial}{\partial y} c_y \;\approx\; \tfrac{1}{2}\left[c_y(x,y+1) - c_y(x,y-1)\right]. \tag{24}$$

Thus, our *rgb* environment matting equation becomes

$$C \;=\; \rho\,\mathcal{M}(\boldsymbol{T},\boldsymbol{A}) \tag{25}$$

4.2 Single image matte recovery

What kind of stimulus function could we use to recover all the parameters in this problem? A logical choice would be a smooth function, where we define smoothness as

$$\mathcal{M}(\boldsymbol{T},\boldsymbol{A}) \;\approx\; \boldsymbol{T}(\boldsymbol{c}) \tag{26}$$

for any area A. Such a function would have the property under our simplified model that

$$C \;\approx\; \rho\,\boldsymbol{T}(\boldsymbol{c})\,. \tag{27}$$

Treating this equation as three equations in r, g, and b, we can easily solve for the three unknowns ρ, c_x, c_y. Backgrounds that are smooth according to the definition in Equation 26 include constant color functions and linear color ramps. However, our function must

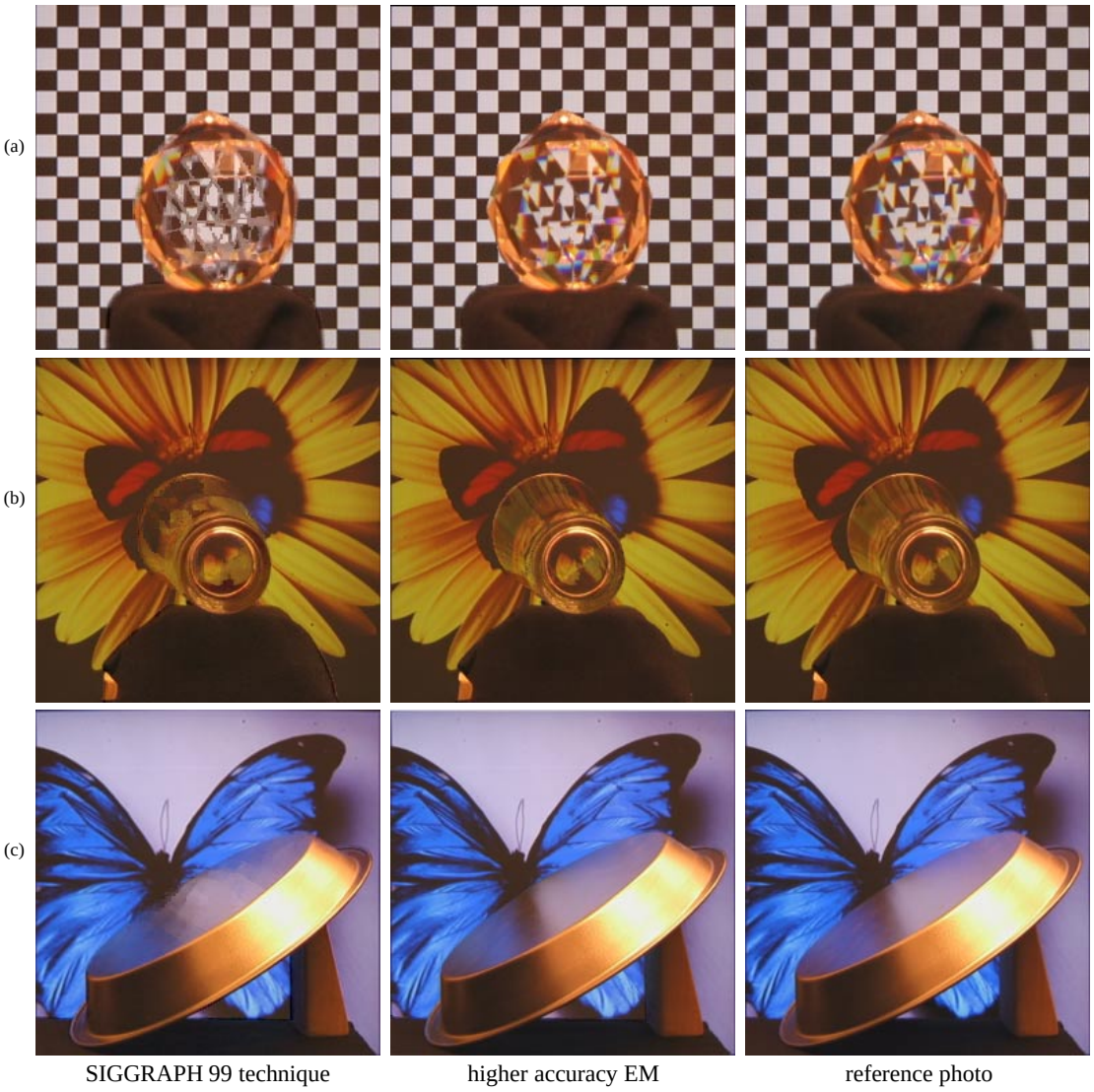


Figure 5 Comparisons between the composite results of the previously published algorithm, the higher accuracy environment matting technique described here, and reference photographs of the matted objects in front of background images. Lighting in the room contributed a yellowish foreground color F that appears, e.g., around the rim of the pie tin in the bottom row. (a) A faceted crystal ball causes rainbowing due to prismatic dispersion, an effect successfully captured by the higher accuracy technique since shifted Gaussian weighting functions are determined for each color channel. (b) Light both reflects off and refracts through the sides of a glass. This bimodal contribution from the background causes catastrophic failure with the previous unimodal method, but is faithfully captured with the new multi-modal method. (c) The weighting functions due to reflections from a roughly-textured pie tin are smooth and fairly broad. The new technique with Gaussian illumination and weighting functions handles such smooth mappings successfully, while the previous technique based on square-wave illumination patterns and rectangular weighting functions yields blocky artifacts.

Figure 6 Oriented weighting functions reflected from a pie tin. (a) As in Figure 5, the previous method yields blocky artifacts for smooth weighting functions. (b) Using the higher accuracy method with unoriented Gaussians ($\theta = 0$) produces a smoother result. (c) Results improve significantly when we orient the Gaussians and solve for θ. In this case, $\theta \approx 25^\circ$ over most of the bottom surface (facing up) of the pie tin. (d) Reference photograph.

also be invertible, so that we can identify which pixel we are seeing from its color. Obviously, a constant color function does not fill this requirement. Instead, we display a color ramp background which is a slice through the *rgb* cube.

Before extracting the environment matte against a ramp background, we add a step that will allow us to recover an estimate of α, the second desirable property on our list. In Section 3, we described a method for classifying pixels as to whether they belong to the object. We can think of this classification as choosing between either $\alpha = 1$ (i.e., the pixel belongs to a foreground object) or $\alpha = \rho = 0$ (i.e., the pixel belongs to the background).

To classify pixels, we first take a series of pictures of the background without the object and average them together to give us a low noise estimate of the ramp background. Once we begin recording video of the object, we apply a simple difference threshold to each frame, comparing the image of the object to the image of the background alone. This step separates foreground and background pixels. We then use some morphology operations (dilation followed by hole-filling followed by erosion) to clean up this binary map, giving us a reasonably accurate mask of the pixels covered by the object. To avoid a sharp discontinuity, we slightly feather the alpha at the boundaries of the object as a post-processing step. Thus, we arrive at the improved matting equation:

$$C \;=\; (1-\alpha)B \;+\; \rho\,\mathcal{M}(\boldsymbol{T},\boldsymbol{A}) \tag{28}$$

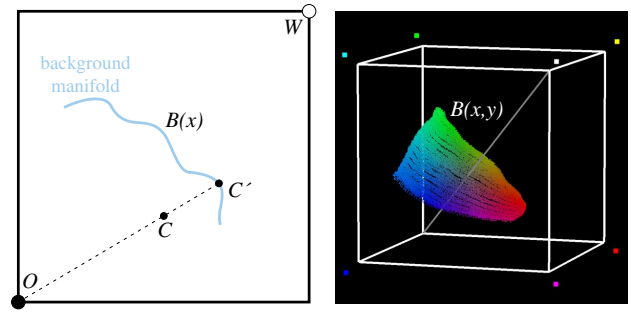
which, for a smooth background, reduces to

$$C \;=\; (1-\alpha)\,B \;+\; \rho\,\boldsymbol{T}(\boldsymbol{c})\,. \tag{29}$$

We can now begin to recover an environment matte. Because we assume that $F = 0$ everywhere, we photograph the object in a dark room, lit only by the structured backdrop. The structured background is a smoothly-varying wash of color, in particular, a planar slice through the *rgb* cube. Due to non-linearities in the system, including crosstalk between the spectra of the monitor phosphors and the CCD elements, the gamma of the backdrop display, and processing in the camera's electronics, this plane in color space will be distorted, becoming a curved 2D manifold lying within the *rgb* cube, as in Figure 7(b).

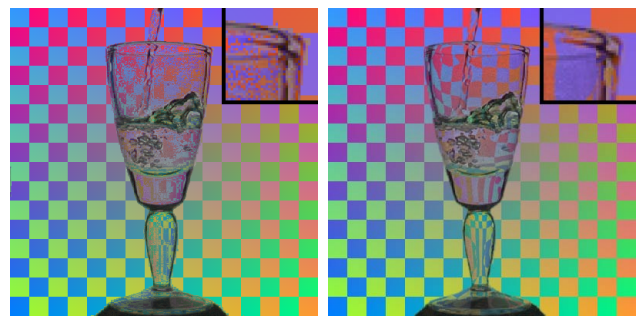
To extract matte parameters at each pixel, we consider the line joining the observed color and the black point in *rgb* space. The point where this line intersects the background-color manifold gives us the point $\boldsymbol{c}$, and the fractional distance of the observed color to the manifold gives us ρ, as illustrated in Figure 7(a). The manifold is difficult to characterize algebraically, so rather than projecting the observed color onto it, we do a multiresolution search to find the point on the manifold closest to the construction line.

A single frame of video captured with a CCD camera will have considerable noise. While we can capture several seconds of the empty background and average frames to create a nearly noise-free reference, we get only one frame of the object in front of the backdrop. This leads to grainy composite images, as seen in Figure 8(a). One way to combat this effect is to filter the input images to smooth out noise before matte extraction. However, our extraction process is so sensitive to noise that we have not been successful in obtaining smooth mappings without also significantly smoothing away detail in the images. Instead, we find that directly smoothing the extracted warping function, $\boldsymbol{c}$, is most effective. To this end, we apply the edge-preserving smoothing operator of Perona and Malik [11] to the c_x and c_y channels. This operator averages each pixel with its neighborhood, with unequal contributions from neighboring pixels. The relative contributions are determined by the difference of the pixels' values, so that similarly-valued pixels affect each other more. This filter smoothes out regions with low-to-moderate noise levels while preventing significant energy



(a) (b)

Figure 7 (a) 2D version of the algorithm for constructing environment mattes from a single image, assuming that $F = 0$. The observed color C is projected from black onto the background manifold. The position of point C' on the manifold gives the position x, and $\rho = \overline{OC}/\overline{OC'}$. (b) In reality, the background colors lie on a 2D manifold within the *rgb* cube, and we recover an (x, y) position.



(a) (b)

Figure 8 A composite created from a single-frame environment matte. Part (a) shows the results when no filtering is applied. Filtering the matte both spatially and temporally reduces noise in the composite considerably, as seen in part (b).

transfer across sharp edges. For video, the best results are obtained when this operator is applied temporally as well as spatially, giving frame-to-frame coherence, which is especially important in areas of the object that are not moving.

4.3 Heuristics for specular highlights

The most noticeable visual effect of this restricted model is the loss of specular highlights. The objects we capture are typically curved glass, and highlights are both important for communicating the shape of the object and for making it visually appealing. In this section, we develop a method for recovering the intensity of the foreground color F, under the restriction that it is white. Thus, $F = f\boldsymbol{W}$ where $\boldsymbol{W} = (1,1,1)$, so that only one additional parameter, f is added to the matting equation. This new single-image environment matting equation then becomes

$$C \;=\; f\,\boldsymbol{W} \;+\; (1-\alpha)\,B \;+\; \rho\,\mathcal{M}(\boldsymbol{T},\boldsymbol{A}) \tag{30}$$

or, for a smooth background:

$$C \;=\; f\,\boldsymbol{W} \;+\; (1-\alpha)\,B \;+\; \rho\,\boldsymbol{T}(\boldsymbol{c})\,. \tag{31}$$

We extend our simple model by allowing the objects to be photographed with bright, near-point light sources. Because the surfaces of our objects are curved, such light sources primarily create bright spots and highlight contours where the normal is equal to the halfway vector between the viewing and lighting rays.

When applied to images taken with lighting other than the backdrop, the recovery algorithm of the last section will discover some points where $\rho > 1$, i.e., where the observed color point lies on the side of the background manifold closer to white. The theory of

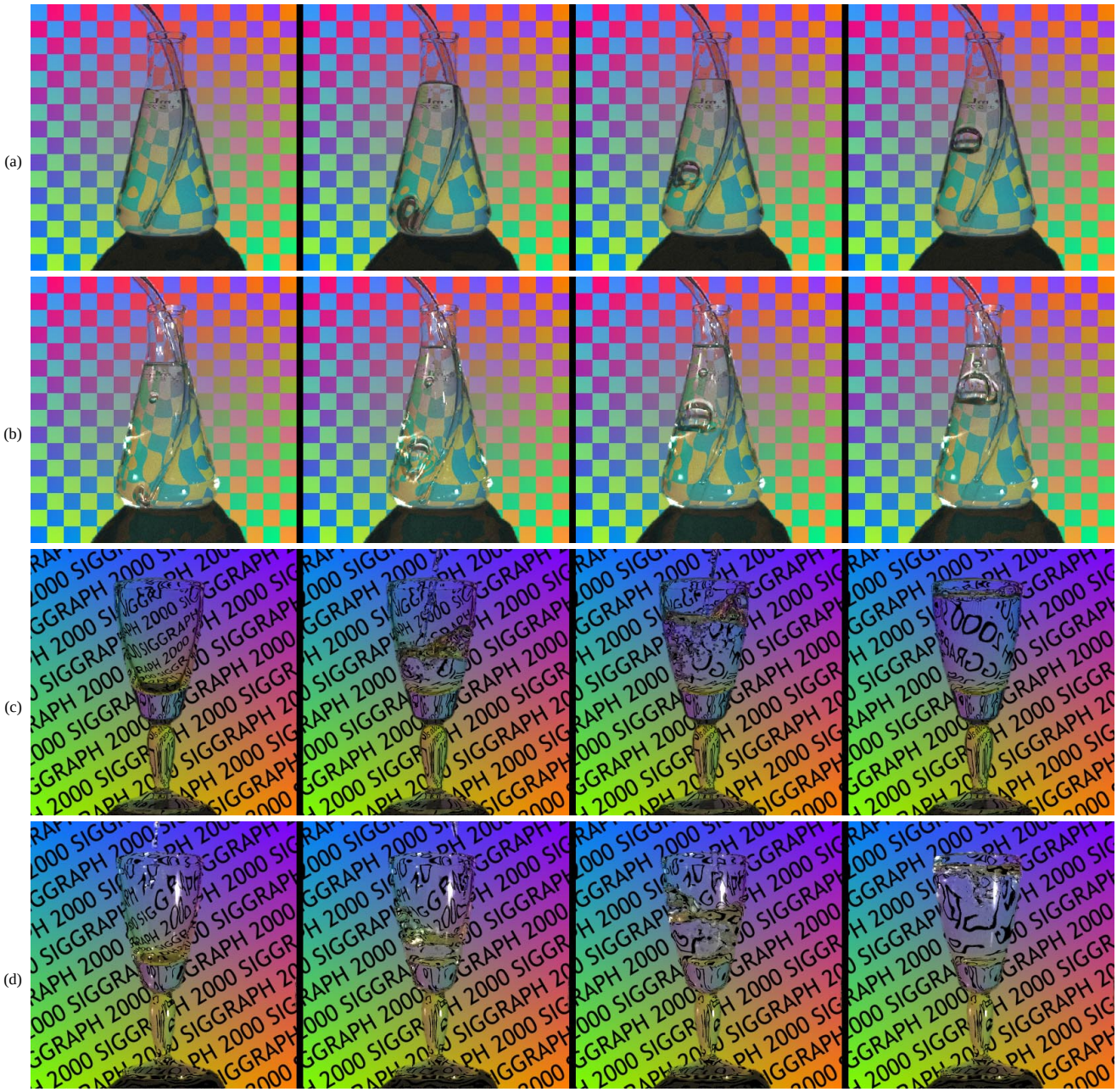


Figure 9 Sample frames from four environment matte video sequences. Rows (a) and (b) show bubbles being blown in an Ehrlenmeyer flask filled with glycerin, while rows (c) and (d) show a glass being filled with water. Sequences (a) and (c) were captured with no lighting other than the backdrop, so that the foreground color is zero. Sequences (b) and (d) were captured separately, shot with the lights on, and the foreground estimation technique is used to recover the highlights.

light transport (see Appendix A) tells us that this should not happen when the object is lit only by the backdrop, so we assume that wherever ρ exceeds unity there must be some F-term contribution to the pixel. (In practice, sensor noise means that ρ can occasionally exceed unity even where there is no highlight. Our algorithm actually looks for highlights only in regions where $\rho > 1 + \delta$, and clamps smaller ρ values to the range $[0, 1]$. For our (fairly noisy) video camera, a δ in the range 0.03–0.10 is typically used.)

Since the highlights we observe will be small or narrow, we make the assumption that the refraction direction and transparency will be smoothly varying in the area of the highlight. We estimate the parameters ρ, c_x, and c_y for the neighborhood around the highlight by interpolating from the values at nearby pixels outside the highlight area and then applying Gaussian smoothing. Once we have estimates for these parameters, we can use Equation 31 to compute f independently for each color channel, e.g.:

$$f_g \;=\; C_g \;-\; (1-\alpha)\, B_g \;-\; \rho\, T_g(\boldsymbol{c})\,, \tag{32}$$

where the g subscript on each variable represents the green color component. Similarly, we compute f_r and f_b for the red and blue color components, respectively, and then combine them to estimate f:

$$f \;=\; \max\{f_r,\, f_g,\, f_b\}\,. \tag{33}$$

In principle, f_r, f_g, and f_b should all be the same, but in practice some or all channels of the observed color C can be clipped, resulting in an artificially low f value. We compute f using each channel separately and take the maximum to counter the effects of such clipping. In practice, the resulting f values may still appear too dim,

(a) (b)

Figure 10 Image (a) is an image of a glass fish sculpture captured in front of a photograph displayed on a computer monitor. Image (b) shows a single frame environment matte composite—the same fish photographed once in front of a color wash and digitally composited onto the fruit image. The background image was darkened using the color histogram matching technique of Heeger and Bergen [7] to approximate the darkening produced by photographing the image on a monitor.

so we scale them up when compositing to produce a more vivid highlight. In addition, we currently solve for f before feathering α, though it would be straightforward to solve for f after feathering.

Figure 9 shows some still images taken from an environment matte video captured with this method. While this technique is most effective for video, still images can also be used for gauging the quality of the extracted mattes. The first four rows all show examples of liquids in motion, which could not be captured by a technique requiring multiple images of the same object pose. Figure 10 compares the results of video environment matting to a photograph. Some details, such as the ridges on the fish's dorsal fin, which are clearly visible in the photograph, have become indistinct with the environment matte, but the environment matte has done a reasonable job of capturing the gross refraction pattern of the object.

5 Conclusion

Environment matting involves an inherent tradeoff: the amount of input data required versus the quality of the resulting matte. The original environment matting and compositing paper by Zongker *et al.* [20] provided one data point in this space—a reasonably accurate model obtained using a reasonably small (logarithmic) number of photographs. In this work, we have presented two additional data points in the environment matting design space. The first captures a higher quality model in which each pixel can see one or more different Gaussian regions of the environment on a per-channel basis. This approach allows for accurate capture of objects with multimodal refraction and reflection qualities, or with prismatic color dispersion. In the second approach, we limited ourselves to a single input image to see how much quality could be retained. While the range of modeled effects must be severely pared down, in many interesting situations the composite images created are still quite convincing. The effect is greatly enhanced by matting an object in motion, an effect made possible with a single-frame solution.

One general area of future work is to develop more sophisticated mathematical tools for extracting environment mattes from our input data. For the higher accuracy method, we are developing a sensitivity analysis of our parameter estimation process in hopes of selecting a new, smaller set of basis functions that exhibit greater noise immunity. For the fast, lower accuracy method, we are researching a more principled Bayesian approach to fitting matte parameters given noisy image streams.

Finally, our accurate environment matting methods should enable us to capture the behaviors of surfaces with bimodal BRDF's, e.g., having specular and diffuse components. Our initial experiments in this direction have yielded promising results. However, we have found that monitor illumination is too weak when reflected off of a diffuse surface. By acquiring high dynamic range radiance maps [5], we hope to solve this problem and demonstrate interactive lighting of these more complex environment mattes. We also expect that Gaussian weighting functions will not accurately model diffuse reflection. Choosing new sets of weighting functions to handle such cases is another area for future work.

Acknowledgements

We would like to thank Eric Veach for his insights on light transport. This work was supported by NSF grants 9803226 and 9875365, and by industrial gifts from Intel, Microsoft, and Pixar, and by the Intel Fellowship program.

References

[1] Paul Besl. Active optical range imaging sensors. In Jorge L.C. Sanz, editor, *Advances in Machine Vision*, chapter 1, pages 1–63. Springer-Verlag, 1989.

[2] J. F. Blinn and M. E. Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19:542–546, 1976.

[3] G. Chazan and N. Kiryati. Pyramidal intensity ratio depth sensor. Technical Report 121, Center for Communication and Information Technologies, Department of Electrical Engineering, Technion, Haifa, Israel, October 1995.

[4] B. Curless and M. Levoy. Better optical triangulation through spacetime analysis. In *Proceedings of IEEE International Conference on Computer Vision*, pages 987–994, June 1995.

[5] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH 97*, pages 369–378, August 1997.

[6] Gerd Häusler and Dieter Ritter. Parallel three-dimensional sensing by color-coded triangulation. *Applied Optics*, 32(35):7164–7169, December 1993.

[7] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. *Proceedings of SIGGRAPH 95*, pages 229–238, August 1995.

[8] Eli Horn and Nahum Kiryati. Toward optimal structured light patterns. In *Proceedings of the International Conference on Recent Advances in Three-Dimensional Digital Imaging and Modeling*, pages 28–35, 1997.

[9] T. Kanade, A. Gruss, and L. Carley. A very fast VLSI rangefinder. In *1991 IEEE International Conference on Robotics and Automation*, volume 39, pages 1322–1329, April 1991.

[10] F. E. Nicodemus, editor. *Self-Study Manual on Optical Radiation Measurements: Part I—Concepts*, chapter 1–3. National Bureau of Standards, 1976. Technical Note 910–1.

[11] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[12] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of SIGGRAPH 84*, volume 18, pages 253–259, July 1984.

[13] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.)*. Cambridge University Press, 1992.

[14] K. Sato and S. Inokuchi. Three-dimensional surface measurement by space encoding range imaging. *Journal of Robotic Systems*, 2:27–39, 1985.

[15] Erhard Schubert. Fast 3d object recognition using multiple color coded illumination. In *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing*, pages 3057–3060, 1997.

[16] Alvy Ray Smith and James F. Blinn. Blue screen matting. In *Proceedings of SIGGRAPH 96*, pages 259–268, August 1996.

[17] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, December 1997.

[18] Gregory J. Ward. Measuring and modeling anisotropic reflection. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 265–272, July 1992.

[19] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.

[20] Douglas E. Zongker, Dawn M. Werner, Brian Curless, and David H. Salesin. Environment matting and compositing. In *Proceedings of SIGGRAPH 99*, pages 205–214, August 1999.

A Bounding the attenuation coefficient

In Section 4, we claim that the attenuation coefficient ρ is always less than or equal to 1. Here we present an argument (due to Eric Veach) that justifies this claim in terms of known physical laws. Our claim is equivalent to showing that the spectral radiance of any ray entering the camera is bounded above by the maximum spectral radiance of rays leaving the background monitor; that is, the intervening objects cannot somehow increase the spectral radiance by reflection or transmission. Since radiance can in fact be increased by refraction, instead we work with a quantity known as *basic radiance*, defined as radiance divided by the square of the refractive index of the surrounding medium [10]. It is well known in classical optics that basic radiance cannot be increased at a refractive surface, a fact known as *Abbe's law*. Although this law is not directly applicable in our case (since it applies only to perfect mirrors and lenses), a similar result can be derived for general reflecting and transmitting surfaces [17]. This result also holds for *basic spectral radiance* under the assumption that surfaces do not exhibit wavelength coupling (e.g., fluorescence).

We now show that the basic spectral radiance along any ray cannot exceed that of the emitting surfaces. The light transport equation can be written as $L = L_e + \mathbf{T}L$, where L is the equilibrium basic radiance distribution, L_e is the emitted basic radiance distribution, and $\mathbf{T}$ is the light transport operator. We wish to show that $\|L\| \leq \|L_e\|$, where all norms are infinity norms. We do this by induction, starting with $L_0 = L_e$ and considering a sequence $L_{i+1} = L_e + \mathbf{T}L_i$ whose limit is L. Veach [17] shows that the norm of $\mathbf{T}$ is always less than one, i.e. $\|\mathbf{T}L_i\| < \|L_i\| \leq \|L_e\|$. If we now assume that the emitting surfaces do not reflect light at the wavelengths they emit (which in fact is a well-known property of emitters), then L_e and $\mathbf{T}L_i$ are non-zero on disjoint sets of rays. For the infinity norm this implies that $\|L_e + \mathbf{T}L_i\| \leq \|L_e\|$, and so by induction $\|L\| \leq \|L_e\|$. Finally, assuming that the monitor and the camera are in the same medium (such as air), this implies that the sensor will never measure a value at any pixel greater than the maximum emitted radiance.